draft-bannister-dbis-automounter-04

dbis : maps d'auto-montage

Ce document étends DBIS pour supporter les bases automount. Le schéma de base automount devrait être compatible avec NIS, mais stocké dans des entrées X.500 pour qu'ils puissent être résolus via LDAP. Une base automount contient les informations sur les points de montage du système qui sont mappés par l'automonteur.

Maps de configuration

La base automount utilise les maps de configuration standard définis dans draft-bannister-dbis-mapping. Additionnellement, les entrées dbisMapConfig pour les base automount devraient avoir la classe dbisAutomountConfig. Il est recommandé que l'entrée dbisMapConfig pour une base automount ai d'attribut dbisMapFilter définis en accord avec la table suivante :

```
- - - - - - - - - - - - - - - - - - - - - -
Database- - - dbisMapFilter
- - - - - - - - - - - - - - - - - - - - - -
automount- - objectClass=automountMapObject
- - - - - - - - - - - - - - - - - - - - - -
```

Exemple d'entrée de map de configuration

L'exemple suivant donne un exemple d'une entrée de map de configuration pour une base automount :

```
dn: cn=automount,en=sales.corp,ou=domain-mappings,o=infra
objectClass: top
objectClass: dbisMapConfig
objectClass: dbisAutomountConfig
cn: automount
dbisMapDN: cn=automount,ou=dbis,o=infra
dbisMapFilter: objectClass=automountMapObject
profileTTL: 900
description: Primary automount database
```

Base de données

Une base automount contient 3 types d'entrées différentes :

- Des entrées de map master
- Des entrées de map Direct
- Des entrées de map Indirect

Une entrée master contient les informations suivantes :

- Le chemin à monter, ou /- pour les maps directs

- Le nom de map contenant les informations de montage pour ce chemin
- Les options de montage

Une entrée directe contient :

- Le nom complet du point de montage
- Les options de montage
- Un ou plusieurs emplacement de système de fichier

Une entrée indirecte contient :

- La clé utilisé pour déterminer le nom de la feuille de point de montage
- Les options de montage
- Un ou plusieurs emplacement de système de fichier

Les maps directe et indirecte peuvent également inclure des entrées d'autre maps nommées. Une entrée à montage multiple définis plusieurs points de montage pour une seule clé dans une map indirect.

Les options de montage, les symboles de substitution et wildcard peuvent être supportés par une implémentation mais sont au-delà du scope de ce document. Ce schéma n'applique aucune restriction sur l'utilisation d'options ou de symboles spéciaux, ni ne tente de les définir. À la place, ce document recherche à décrire un schéma et une structure qui supporte toute implémentation.

ObjectClass

Une entrée dbisMapConfig pour une base automount devrait avoir la classe dbisAutomountConfig. Une entrée de map master devrait être définie par une entrée avec la classe automountMaster, qui est un sous-classe de automountMapObject. Une map directe ou indirecte devrait être identifiée par la classe automountMapObject. Les entrées dans une map donnée doivent être des objets enfant dans le DIT de la map à laquelle elle appartient et utiliser la classe automountEntry. Il n'y a pas de telles restriction d'emplacement sur les entrées automountMapObject, qui peuvent être localisées n'importe où dans le DIT tant qu'ils sont localisé par dbisMapDN et dbisMapFilter.

La classe automountMulti devrait être utilisée pour identifier l'objet top-level dans une entrée de montage multiple. Chaque entrée dans le montage multiple devrait être un objet enfant dans le DIT avec la classe automountEntry. Un DUA peut choisir de ne pas supporter les entrées de montage multiples.

Une map directe ou indirecte peut inclure les entrées d'une autre map par une entrée avec la classe automountInclude, qui devrait être un objet enfant dans le DIT d'un automountMapObject ou un automountMulti.

dbisAutomountConfig

La classe dbisAutomountConfig est définie comme suit :

```
objectclass ( 1.3.6.1.4.1.23780.219.1.27 NAME 'dbisAutomountConfig' DESC 'DBIS automount configuration map'
SUP dbisMapConfig STRUCTURAL )
```

automountMapObject

La classe automountMapObject est définie comme suit :

```
objectclass ( 1.3.6.1.4.1.23780.219.1.28 NAME 'automountMapObject' DESC 'Top-level of an automount map,
entries are child nodes' SUP top STRUCTURAL MUST en MAY ( description $ manager $ disableObject ) )
```

automountMaster

La classe automountMaster est définie comme suit :
```
objectclass ( 1.3.6.1.4.1.23780.219.1.29 NAME 'automountMaster' DESC 'Automounter master map entry' SUP
automountMapObject STRUCTURAL MUST automountUseMap MAY automountOption )
```

automountEntry

La classe automountEntry est définie comme suit :
```
objectclass ( 1.3.6.1.4.1.23780.219.1.31 NAME 'automountEntry' DESC 'Automounter direct or indirect map
entry' SUP top STRUCTURAL MUST ( en $ automountLocation ) MAY ( automountOption $ description $ manager $
disableObject ) )
```

Les entrées de cette classe devraient être des objets enfant dans le DIT de l'automountMapObject auquel il appartient, sauf si elles font partie d'une montage multiple auquel cas elles doivent être enfant de leur entrée automountMulti correspondante.

automountInclude

La classe automountInclude est définie comme suit :
```
objectclass ( 1.3.6.1.4.1.23780.219.1.32 NAME 'automountInclude' DESC 'Include another map within an
automounter direct map, indirect map or multiple mount entry' SUP top STRUCTURAL MUST en MAY ( description $
manager $ disableObject ) )
```

Le nom de la map à inclure devrait être fournie dans l'attribut en. Les règles pour localiser une entrée automountInclude dans le DIT sont les mêmes que pour une entrée automountEntry.

Attributs

en

Le chemin à monter, la clé utilisée dans la map indirecte, ou le nom d'une map à inclure lorsqu'utilisé avec la classe automountInclude, est stocké dans l'attribut en, qui est définis dans draft-bannister-dbis-mapping. Ce attribut doit être associé avec les entrées automountMapObject, automountMaster, automountMulti, automountEntry et automountInclude et devrait former le RDN dans tous les cas.

automountUseMap

Le nom de la map d'auto-montage à utiliser avec l'entrée de map master est stocké dans l'attribut automountUseMap qui doit être assigné à une entrée automountMaster :
```
attributetype ( 1.3.6.1.4.1.23780.219.2.31 NAME 'automountUseMap' DESC 'Name of automount map associated with
master map entry' EQUALITY caseExactIA5Match SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
```

automountOption

Chaque option d'auto-montage est stocké dans un ou plusieurs attributs automountOption qui peuvent être assignés à une entrée automountMaster, automountEntry ou automountMulti :

```
attributetype ( 1.3.6.1.4.1.23780.219.2.32 NAME 'automountOption' DESC 'Automounter option' EQUALITY
caseExactIA5Match SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
```

Les options disponibles sont spécifique à l'implémentation. Les options d'auto-montage ne sont pas additifs, par exemple, les options d'un automountEntry, si fournis, devraient écraser les options dans une entrée automountMulti ou automountMaster. Les options d'une entrée automountMulti devraient écraser les options d'une entrée automountMaster.

automountLocation

Le serveur de fichier et le chemin sont fournis dans un ou plusieurs attributs automountLocation qui doivent être assignés à un automountEntry :

```
attributetype ( 1.3.6.1.4.1.23780.219.2.33 NAME 'automountLocation' DESC 'Automounter location e.g.
server:path' EQUALITY caseExactIA5Match SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
```

Le format est spécifique à l'implémentation, mais typiquement "server :path". Le composant serveur peut être optionnel mais l'option (:) et le composant chemin ne le sont pas. Certaines implémentations supportent également plusieurs serveurs séparés par des virgules, ex, serverA,serverB :path, et des options de poids, ex, serverA(2),serverB(1) :path, pour spécifier une préférence pour le second serveur dans la liste.

description

L'attribut description peut être associé avec une entrée pour fournir une description arbitraire de l'entrée.

manager

L'attribut manager peut être associé avec une entrée pour fournir un ou plusieurs DN d'individus, groupes ou systèmes qui sont responsables de la maintenance de l'entrée.

disableObject

Une entrée peut être désactivée en définissant disableObject à TRUE. Si une entrée est désactivée, le DUA devrait se comporter comme si l'entrée n'existait pas. Le DUA peut optionnellement fournir un mécanisme séparé pour lister les entrées désactivé, mais doivent les marquer clairement comme désactivé pour éviter toute confusion.

Exemple d'entrées d'auto-montage

L'exemple suivant montre des entrées automountMaster au format LDIF :

```
dn: en=/home,ou=master,ou=automount,o=infra
```

```
objectClass: top
objectClass: automountMapObject
objectClass: automountMaster
en: /home
automountOption: nobrowse
automountUseMap: auto_home
description: Master entry for /home, see auto_home map

dn: en=/qa,ou=master,ou=automount,o=infra
objectClass: top
objectClass: automountMapObject
objectClass: automountMaster
en: /qa
automountUseMap: qa
description: Master entry for /qa, see qa map

dn: en=/media,ou=master,ou=automount,o=infra
objectClass: top
objectClass: automountMapObject
objectClass: automountMaster
en: /media
automountUseMap: media
description: Master entry for /media, see media map

dn: en=/-,ou=master,ou=automount,o=infra
objectClass: top
objectClass: automountMapObject
objectClass: automountMaster
en: /-
automountUseMap: auto_direct
description: Master entry for direct maps, see auto_direct
```

L'exemple suivant montre des entrées de map direct :

```
dn: en=auto_direct,ou=maps,ou=automount,o=infra
objectClass: top
objectClass: automountMapObject
en: auto_direct
description: auto_direct map entries are child objects

dn: en=/usr/install,en=auto_direct,ou=maps,ou=automount,o=infra
objectClass: top
objectClass: automountEntry
en: /usr/install
automountOption: ro
automountLocation: esher,kingston(1):/export/install
automountLocation: hampton(3):/usr/install
description: Mount /usr/install from three possible servers
```

L'exemple suivant montre des entrées de map indirect :

```
dn: en=auto_home,ou=maps,ou=automount,o=infra
objectClass: top
objectClass: automountMapObject
en: auto_home
description: auto_home map entries are child objects

dn: en=fred,en=auto_home,ou=maps,ou=automount,o=infra
objectClass: top
```

```
objectClass: automountEntry
en: fred
automountLocation: surbiton:/export/home/&
description: /home/fred

dn: en=sheila,en=auto_home,ou=maps,ou=automount,o=infra
objectClass: top
objectClass: automountEntry
en: sheila
automountLocation: surbiton:/export/home/&
description: /home/sheila

dn: en=*,en=auto_home,ou=maps,ou=automount,o=infra
objectClass: top
objectClass: automountEntry
en: *
automountLocation: ditton:/export/home/&
description: Catch-all for user home directories not listed

dn: en=surrey_home,en=auto_home,ou=maps,ou=automount,o=infra
objectClass: top
objectClass: automountInclude
en: surrey_home
description: Include entries from surrey_home map
```

L'exemple suivant montre un autre entrée de map indirect :

```
dn: en=media,ou=maps,ou=automount,o=infra
objectClass: top
objectClass: automountMapObject
en: media
description: media map entries are child objects

dn: en=cdrom,en=media,ou=maps,ou=automount,o=infra
objectClass: top
objectClass: automountEntry
en: cdrom
automountOption: fstype=hsfs
automountOption: ro
automountLocation: :/dev/sr0
description: /media/cdrom
```

L'exemple suivant montre une entrée à montage multiple :

```
dn: en=qa,ou=maps,ou=automount,o=infra
objectClass: top
objectClass: automountMapObject
en: qa
description: qa map entries are child objects

dn: en=qa_root,en=qa,ou=maps,ou=automount,o=infra
objectClass: top
objectClass: automountMulti
en: qa_root
automountOption: ro
description: qa_root multi-mount entries are child objects

dn: en=/,en=qa_root,en=qa,ou=maps,ou=automount,o=infra
objectClass: top
```

```
objectClass: automountEntry
en: /
automountLocation: esher:/export/qa
description: /qa

dn: en=/docs,en=qa_root,en=qa,ou=maps,ou=automount,o=infra
objectClass: top
objectClass: automountEntry
en: /docs
automountLocation: surbiton:/export/qa/docs
description: /qa/docs

dn: en=/tmp,en=qa_root,en=qa,ou=maps,ou=automount,o=infra
objectClass: top
objectClass: automountEntry
en: /tmp
automountOption: rw
automountLocation: surbiton:/export/qa/tmp
description: /qa/tmp
```

Syntaxe d'attributs

Les syntaxes d'attribut suivant sont utilisés par les attributs définis dans ce document :

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Syntax OID - - - - - - - - - -Value - - - - - - Reference
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1.3.6.1.4.1.1466.115.121.1.12 DN - - - - - - - -[RFC4517]
1.3.6.1.4.1.1466.115.121.1.26 IA5 String - - - -[RFC4517]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Mappage des champs NIS

Tous les champs qui sont requis pour générer des format de base d'auto-montage NIS existent dans ce schéma et peuvent être mappés en utilisant les production ABNF décrites dans draft-bannister-dbis-netgroup.

Les champs de map master sont mappés comme suit :

```
mount-point = en
map-name = automountUseMap
mount-option = automountOption

master-entry = mount-point SPACE map-name *(SPACE mount-option)
```

Les champs de map directe sont mappés comme suit :

```
mount-point = en ; from automountEntry
mount-option = automountOption ; from automountEntry
location = automountLocation ; from automountEntry
include-map = en ; from automountInclude

mount = mount-point *(SPACE mount-option)
  1*(SPACE location)
include = PLUS include-map
```

```
direct-entry = mount / include
```

Les champs de map indirecte sont mappés comme suit :

```
mount-point = en ; from automountEntry
mount-option = automountOption ; from automountEntry
location = automountLocation ; from automountEntry
multi-key = en ; from parent automountMulti
multi-option = automountOption ; from parent automountMulti
include-map = en ; from automountInclude

mount = mount-point *(SPACE mount-option) 1*(SPACE location)
multi-mount = multi-key *(SPACE multi-option) 1*(LF SPACE mount-point *(SPACE mount-option) 1*(SPACE
location))
include = PLUS include-map

direct-entry = mount / multi-mount / include
```

Filtres de recherche communs

Cette section fournis des exemples de filtre de recherche pour obtenir des entrées de base avec des critères communément utilisés.

Pour simplifier les exemples, toutes les base sont assumées être définies avec une simple map de configuration. Cependant, dbis permet plusieurs entrées, donc une implémentation doit le supporter, augmentant le nombre de recherches nécessaires pour localiser toutes les entrées de base.

Le DN de base utilisé dans les opérations de recherche décrites dans cette section provient de l'attribut dbisMapDN assigné à l'entrée dbisMapConfig. Noter qu'une entrée dbisMapConfig peut en avoir plus d'une.

Quand il apparaît dans les filtres de recherche ci-dessous, le texte "dbisMapFilter" réfère à la valeur assignée à l'attribut de même nom dans l'entrée dbisMapConfig correspondante. Les noms de classe et attribut utilisés dans ces filtres de recherche peuvent être modifiés par dbisMapClass et dbisMapAttr assignés à l'entrée dbisMapConfig.

Pour localiser la map de configuration d'un domaine DBIS :
(&(objectClass=dbisAutomountConfig)(!(disableObject=TRUE)))
Les maps master sont énumérées en appliquant le dbisMapFilter comme suit :
(&(dbisMapFilter)(objectClass=automountMaster)(!(disableObject=TRUE)))
Les maps directes et indirectes sont énumérées par le filtre suivant :
(&(dbisMapFilter)(!(objectClass=automountMaster))(!(disableObject=TRUE)))
Les entrées de map sont énumérée par le filtre suivant :
(&(objectClass=automountEntry)(!(disableObject=TRUE)))
Pour différencier entrée une entrée de map simple et une entrée à montage multiple, l'objectClass dans l'entrée parent doit être testée. Si la classe est automountMulti, alors cette entrée fait partie d'une entrée à montage multiple. Dans ce cas, le DUA devrait tester disableObject=TRUE sur l'objet automountMulti. Si cet objet est marqué désactivé, les entrées enfant devraient également être traités comme désactivés.

Les entrées à montage multiples sont énumérées par le filtre suivant :
(&(objectClass=automountMulti)(!(disableObject=TRUE)))
Pour localiser une map spécifique appelée "name", utiliser le filtre suivant :
(&(dbisMapFilter)(!(disableObject=TRUE))(en=name))
Pour lister toutes les maps à inclure par une map spécifique appelée "name", utiliser le filtre précédent pour localiser le DN de la map, puis utiliser sur tous les objets enfants :
(&(objectClass=automountInclude)(!(disableObject=TRUE)))