
draft-bannister-dbis-devices-04

dbis : devices

Ce document étend DBIS pour supporter les bases ethers et bootparams. Les schémas de base de données devraient être compatible avec NIS, mais stockés dans les entrées X.500 pour pouvoir être résolus avec LDAP.

Une base ethers map les adresses Ethernet 48bits à des adresses IP ou des noms d'hôte, et bootparams map les hôte aux paramètres de boot du kernel.

Ce document décrit les classes et attributs LDAP requis pour étendre les entrées hosts pour supporter les paramètres pour les map ethers et bootparams.

Maps de configuration

Les bases ethers et bootparams utilisent les maps de configuration standard définies dans draft-bannister-dbis-mapping. Additionnellement, les entrées dbisMapConfig pour les bases ethers devraient avoir la classe dbisEtherConfig, et les entrées pour les bases bootparams devraient avoir la classe dbisBootConfig.

Il est recommandé que l'entrée dbisMapConfig pour une base ethers ou bootparams ait l'attribut dbisMapFilter définis en accord avec la tables suivante :

```
-----  
Database- - - dbisMapFilter  
-----
```

```
ethers- - - objectClass=ieee802Device  
bootparams- - objectClass=bootableDevice  
-----
```

Exemple d'entrée de map de configuration

L'exemple suivant montre une entrée de map de configuration pour une base ethers

```
dn: cn=ethers,en=sales.corp,ou=domain-mappings,o=infra  
objectClass: top  
objectClass: dbisMapConfig  
objectClass: dbisEtherConfig  
cn: ethers  
dbisMapDN: ou=hosts,o=infra  
dbisMapDN: ou=lab,ou=hosts,o=infra  
dbisMapFilter: objectClass=ieee802Device  
profileTTL: 900  
description: Primary ethers database
```

L'exemple suivant montre en entrée de map de configuration pour une base bootparams

```
dn: cn=bootparams,en=sales.corp,ou=domain-mappings,o=infra  
objectClass: top  
objectClass: dbisMapConfig
```

```
objectClass: dbisBootConfig
cn: bootparams
dbisMapDN: ou=hosts,o=infra
dbisMapDN: ou=lab,ou=hosts,o=infra
dbisMapFilter: objectClass=bootableDevice
profileTTL: 900
description: Primary bootparams database
```

Base ethers

Une base ethers contiens les champs suivants :

- Adresse ethernet 48-bits
- Un nom d'hôte

ObjectClass

Une entrée dbisMapConfig pour une base ethers devrait avoir la class dbisEtherConfig. Une entrée host, définie par la classe ipv4HostObject ou ipv6HostObject, peut être augmentée par la classe ieee802Device pour ajouter des informations pour la map ethers.

dbisEtherConfig

La classe dbisEtherConfig est définie comme suit :

```
objectclass ( 1.3.6.1.4.1.23780.219.1.37 NAME 'dbisEtherConfig' DESC 'DBIS ethers configuration map' SUP
dbisMapConfig STRUCTURAL )
```

ieee802Device

La classe ieee802Device est définie comme suit :

```
objectclass ( 1.3.6.1.1.1.2.11 NAME 'ieee802Device' DESC 'A device with a 48-bit Ethernet address' SUP top
AUXILIARY MAY macAddress )
```

C'est une class auxiliaire et il est recommandé qu'elle soit associé avec des entrées ipv4HostObject ou ipv6HostObject. Cependant, il est préférable de conserver les adresses Ethernet dans des entrées séparée qui peuvent être associés avec des classe device à la place.

Attributs

macAddress

L'adresse 48-bits est stockée dans l'attribut `macAddress` qui peut être associé avec une entrée `ieee802Device`
attributetype (1.3.6.1.1.1.1.22 NAME ('macAddress') DESC 'MAC address in maximal, colon separated hex
notation, eg. 00:00:92:90:ee:e2' EQUALITY caseIgnoreIA5Match SYNTAX 1.3.6.1.4.1.1466.115.121.1.26)

Exemple d'entrée `ieee802Device`

L'exemple suivant est un exemple d'une entrée `ipv4HostObject` au format `ldif` avec une class `ieee802Device` :

```
dn: rn=kilcher,ou=hosts,o=infra
objectClass: top
objectClass: ipHostObject
objectClass: ipv4HostObject
objectClass: ieee802Device
rn: kilcher
ipv4Address: 10.11.12.13
macAddress: 08:00:27:00:50:f2
```

base bootparams

Une base `bootparams` contient les champs suivants :

- Un nom d'hôte
- Des paramètres de boot

Les paramètres de boot sont interprétés par le kernel et varie d'une plate-forme à l'autre. Ce schéma ne tente pas de définir des attributs unique pour chaque paramètre.

ObjectClass

Une entrée `dbisMapConfig` pour une base `bootparams` devrait avec la class `dbisBootConfig`. Une entrée `host`, définie par la classe `ipv4HostObject` ou `ipv6HostObject`, peut être augmentée par la classe `bootableDevice` pour ajouter des informations pour la map de `bootparams`, qui fournis des informations de configuration pour `rpc.bootparamd`.

dbisBootConfig

La classe `dbisBootConfig` est définie comme suit :

```
objectclass ( 1.3.6.1.4.1.23780.219.1.38 NAME 'dbisBootConfig' DESC 'DBIS bootparams configuration map' SUP
dbisMapConfig STRUCTURAL )
```

bootableDevice

La classe `bootableDevice` est définie comme suit :

```
objectclass ( 1.3.6.1.1.1.2.12 NAME 'bootableDevice' DESC 'A device with boot parameters' SUP top AUXILIARY
MAY ( bootFile $ bootParameter ) )
```

C'est une classe auxiliaire et il est recommandé qu'elle soit associée avec des entrée ipv4HostObject ou ipv6HostObject. Cependant, il est préférable de conserver les adresses Ethernet dans des entrées séparée qui peuvent être associés avec des classe device à la place.

Attributs

bootFile

Nom de l'image de boot est stocké dans l'attribut bootFile qui peut être associé avec une entrée bootableDevice :

```
attributetype ( 1.3.6.1.1.1.1.24 NAME 'bootFile' DESC 'Boot image name' EQUALITY caseExactIA5Match SYNTAX
1.3.6.1.4.1.1466.115.121.1.26 )
```

bootParameter

Les paramètres de boot sont stockés comme "clé=valeur" dans l'attribut bootParameter qui peut être associé avec une entrée bootableDevice :

```
attributetype ( 1.3.6.1.1.1.1.23 NAME 'bootParameter' DESC 'rpc.bootparamd parameter' EQUALITY
caseExactIA5Match SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
```

Exemple d'entrée avec bootableDevice

L'exemple suivant montre une entrée ipv4HostObject au format ldif avec une classe bootableDevice :

```
dn: rn=kilcher,ou=hosts,o=infra
objectClass: top
objectClass: ipHostObject
objectClass: ipv4HostObject
objectClass: ieee802Device
objectClass: bootableDevice
rn: kilcher
ipv4Address: 10.11.12.13
macAddress: 08:00:27:00:50:f2
bootParameter: root=alaska:/export/client/root
bootParameter: domain=country.music.edu
```

Syntaxe d'attribut

Les syntaxes suivante sont utilisée par les attributs définis dans ce document :

```
- - - - -
Syntax OID- - - - - -Value- - - - - -Reference
```

1.3.6.1.4.1.1466.115.121.1.26 -IA5 String - - - -[RFC4517]

Mappage de champs NIS

Tous les champs qui sont requis pour générer des formats de base ether et bootparams NIS existent dans ce schéma et peut être mappés aux types d'attribut en utilisant les production ABNF décrites dans draft-bannister-dbis-netgroup.

ethers

Les champs de base ethers NIS sont mappés comme suit :

```
ether-addr = macAddress  
hostname = rn / en ; depending on class, see below  
  
ethers-entry = ether-addr SPACE hostname
```

- hostname vient de l'attribut rn si la classe ipv4HostObject ou ipv6HostObject est utilisée. Si la classe ieee802Device est associée avec un objet avec la classe device, le nom d'hôte vient de l'attribut cn.

bootparams

Les champs de base bootparams NIS sont mappés comme suit :

```
hostname = rn / en ; depending on class, see below  
params = bootParameter *(SPACE bootParameter)  
  
bootparams-entry = hostname SPACE params
```

- hostname vient de l'attribut rn si la classe ipv4HostObject ou ipv6HostObject est utilisée. Si la classe bootableDevice est associée avec un objet avec la classe device, le nom d'hôte vient de l'attribut cn.

Filtres de recherche communs

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

Si une entrée hosts a une adresse ethernet "ether", sa définition est localisée en utilisant le filtre de recherche suivant :

```
(&(dbisMapFilter)(!(disableObject=TRUE))(objectClass=ieee802Device)(macAddress=ether))
```