

---

# draft-bannister-dbis-mapping-06

## Directory-Based Information Services

Ce document fait partie de la série de documents qui décrivent les composants dans DBIS (Directory-Based Information Services). DBIS fournit un framework pour la représentation des données lié à TCP/IP et le système UNIX dans des entrées X.500 qui ont été précédemment stockés dans NIS ; pour qu'ils puissent être résolus avec le protocole LDAP.

Le but de DBIS est d'étendre, et remplacer le protocole NIS et expérimental pour utiliser LDAP comme service d'information réseaux..

DBIS consiste d'un schéma LDAP, de conventions de nommage et de protocoles pour décrire son utilisation par les DUA nécessitant des informations de service réseaux. La communication client/serveur et les opérations côté serveur sont entièrement contenus dans le domaine de LDAP.

Les aspects clé de DBIS et des améliorations par rapport à RFC2307 sont :

- le schéma est compatible avec NIS, incluant la sensibilité à la casse des noms clé.
- Standardisation des informations de mappage pour augmenter la portabilité des implémentations DUA et pour réduire la duplication des informations de configuration du client.
- Fonctionnalités ajoutées pour augmenter la flexibilité dans de grands environnements :
  - les maps peuvent être jointes depuis les données localisées dans différents endroits du DIT.
  - Les groupes de DUA peuvent avoir de variations dans leur données en fonction de leur appartenance aux netgroup d'hôte.
- Conception modulaire pour permettre de séparer les parties du système à remplacer, améliorer et augmenter séparément dans le future.
- Support ajouté pour les maps d'automontage

Ce document décrit le mappage d'objets utilisés par DBIS pour localiser et transformer les informations de service stockés dans le DIT.

## Databases

Le rôle de DBIS est de fournir un framework qui fournit des informations de configuration, principalement des services de nom tel que les comptes, groupes et hôtes/réseaux, et toutes les informations traditionnellement fournies par NIS. Chaque type d'information est appelée une base de donnée, vu que c'est une collection d'entrées liées dans le DIT. Le format des entrées est spécifique à chaque type de base et n'est pas défini dans ce document. Chaque base est configurée séparément en utilisant des maps de configuration qui décrivent où localiser les entrées concernées dans le DIT. Le format des maps de configuration est définie dans ce document, bien qu'il peut être étendu par d'autres documents.

## Alias

Quand une base supporte les entrées alias, ils doivent être configurés comme décrits dans la rfc4512. Un DUA devrait effectuer le déréférencement d'alias dans ces bases.

## Domaine : définition

---

Le mappage d'objets définis les composants qui font un domaine DBIS. Un domaine DBIS, est un groupement logique de services d'informations requis par une collection commune de DUA, de la même manière que le domaine NIS contient toutes les maps NIS requises pour les opérations d'un groupe de machines. Un domaine DBIS devrait être identifié par une entrée LDAP avec l'objectClass dbisDomainObject. Les maps de configuration pour le domaine sont contenus dans les entrées qui devraient être localisés sous l'entrée dbisDomainObject dans le DIT.

## dbisDomainObject

Cette classe est définie comme suit :

```
objectclass ( 1.3.6.1.4.1.23780.219.1.1 NAME 'dbisDomainObject' DESC 'Defines a top-level mapping object for a DBIS domain' SUP top STRUCTURAL MUST en MAY ( profileTTL $ negativeTTL $ description $ manager ) )
```

## Attributs de domaine

**en** Le nom du domaine, identique au format dans un domaine NIS, qui doit être associé avec une entrée dbisDomainObject et devrait former le RDN.

**profileTTL** Valeur TTL par défaut pour la pertinence des données de configuration (définis dans la rfc4876) qui peut être associé avec une entrée dbisDomainObject. Les DUA devraient conserver une copie locale des données obtenues. Si profileTTL vaut 0, les DUA peuvent conserver les copies locales indéfiniment. Si profileTTL n'est pas présent, les DUA devraient se comporter comme s'il était définis à 0.

**negativeTTL** Identique à profileTTL, excepté pour les entrées qui n'existent pas. Les DUA devraient conserver une copie locale des recherches qui n'existent pas mais ne doivent pas utiliser cette donnée après le nombre de secondes spécifiée.

```
attributetype ( 1.3.6.1.4.1.23780.219.2.36 NAME 'negativeTTL' DESC 'Time to live, in seconds, for missing entries' EQUALITY integerMatch ORDERING integerOrderingMatch SINGLE-VALUE SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 )
```

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

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

## Alias de domaine

Si les alias noms de domaine sont requis ils doivent être configurés comme décrits dans la rfc4512. Un DUA doit déréférencer les alias.

## Exemple d'entrée de domaine

L'exemple suivant est une entrée dbisDomainObject :

```
dn: en=sales.corp,ou=domain-mappings,o=infra
objectClass: top
objectClass: dbisDomainObject
en: sales.corp
profileTTL: 900
```

---

negativeTTL: 300  
description: Sales Workforce

## Maps de configuration

Une map de configuration DBIS instruit un DUA sur l'emplacement des entrées dans le DIT pour une base particulière. Elle décrit comment trouver les entrées et optionnellement quel sous-jeu de DUA devraient utiliser ces entrées (basé sur l'appartenance de netgroup). Ce document ne définit aucune map de configuration spécifique, mais un framework qui doit être suivi pour la spécification de tels maps. Les maps de configuration devraient être évaluées par un DUA dans l'ordre lexicographique de leur attribut cn. L'ordre d'évaluation de ces entrées de configuration déterminent également l'ordre dans lequel les entrées de base apparaissent si plusieurs emplacements sont trouvés. L'ordre est également important pour s'assurer que les netgroups corrects sont disponibles pour tester si la configuration est restreinte par l'appartenance à un netgroup en utilisant soit l'attribut exactNetgroup ou notNetgroup.

## dbisMapConfig

Une map pour une base est optionnelle et devrait être identifiée par une ou plusieurs entrées LDAP localisées sous l'entrée dbisDomainObject dans le DIT. Le comportement du DUA si une entrée depuis une base est demandée et n'a pas de map de configuration correspondant est indéfinie. Les entrées de map de configuration assignées, ou une sous-classe :

```
objectclass ( 1.3.6.1.4.1.23780.219.1.2 NAME 'dbisMapConfig' DESC 'DBIS configuration map for a specific database' SUP top STRUCTURAL MUST ( cn $ dbisMapDN ) MAY ( dbisMapFilter $ dbisMapClass $ dbisMapAttr $ dbisTransAttr $ exactNetgroup $ notNetgroup $ profileTTL $ negativeTTL $ description $ manager $ disableObject ) )
```

Un DUA devrait supporter plusieurs entrées de map de configuration pour une simple base. Une base devrait nécessiter au moins un objectClass additionnel assigné à sa configuration, qui est utilisé pour identifier de manière unique le type de base pour lequel les entrées appartiennent.

## cn

L'attribut cn doit être utilisé pour former le RDN d'une entrée dbisMapConfig. C'est un nom arbitraire qui n'a pas de signification spéciale dans DBIS, mais qui identifie de manière unique l'entrée dbisMapConfig. Comme vu plus haut, les entrées sont évaluées dans l'ordre lexicographique de leur attribut cn.

## dbisMapDN

Un ou plusieurs DN's localisant la base de recherche des entrées de la base dans le DIT sont donnés dans l'attribut dbisMapDN qui doit être assigné à une entrée dbisMapConfig :

```
attributetype ( 1.3.6.1.4.1.23780.219.2.1 NAME 'dbisMapDN' DESC 'DN of search base for DBIS database entries' EQUALITY distinguishedNameMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 )
```

## dbisMapFilter

---

Un filtre de recherche LDAP utilisé pour localiser les entrées de la base sous lequel chaque dbisMapDN est donné dans l'attribut dbisMapFilter qui peut être assigné à une entrée dbisMapConfig :

```
attributetype ( 1.3.6.1.4.1.23780.219.2.2 NAME 'dbisMapFilter' DESC 'LDAP search filter for DBIS database entries' EQUALITY caseIgnoreIA5Match SINGLE-VALUE SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
```

## dbisMapClass

Les objectClass utilisés pour identifier les entrées pour une base peut être changé du défaut par l'attribut dbisMapClass qui peut être assigné à une entrée dbisMapConfig :

```
attributetype ( 1.3.6.1.4.1.23780.219.2.3 NAME 'dbisMapClass' DESC 'LDAP class mapping for DBIS database entries' EQUALITY caseIgnoreIA5Match SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
```

La représentation chaîne de l'attribut dbisMapClass est définie par la grammaire suivante, qui utilise la notation ABNF. Les productions utilisées qui ne sont pas définies ici sont définies dans la rfc4512.

```
from_class = keystack  
to_class = keystack  
dbisMapAttr = to_class EQUALS from_class
```

Si l'attribut dbisMapClass est manquant de l'entrée dbisMapConfig, le DUA devrait continuer avec les classes par défaut pour la base. Changer cet attribut n'a pas d'effet sur le dbisMapFilter, qui doit être ajusté indépendamment.

## dbisMapAttr

Les attributs utilisé pour stocker les clés et valeurs de l'entrée de la base peuvent être changés par l'attribut dbisMapAttr qui peut être assigné à une entrée dbisMapConfig :

```
attributetype ( 1.3.6.1.4.1.23780.219.2.4 NAME 'dbisMapAttr' DESC 'LDAP attribute mapping for DBIS database entries' EQUALITY caseIgnoreIA5Match SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
```

La représentation chaîne de l'attribut dbisMapAttr est définie par la grammaire suivante, qui utilise la notation ABNF. Les productions utilisées qui ne sont pas définies ici sont définies dans la rfc4512.

```
from_attr = keystack  
to_attr = keystack  
dbisMapAttr = to_attr EQUALS from_attr
```

L'attribut utilisé dans la base est identifiée par from\_attr et devrait être ré-écrit par le DUA à l'attribut to\_attr. Si l'attribut dbisMapAttr est manquant de l'entrée dbisMapConfig, le DUA devrait continuer avec les attributs par défaut pour la base. Changer cet attribut n'a pas d'effet sur le dbisMapFilter ni dbisTransAttr, qui doivent être ajustés indépendamment.

## dbisTransAttr

Les valeurs d'attribut utilisés par les entrées de base peuvent être transformés par l'attribut dbisTransAttr qui peut être assigné à une entrée dbisMapConfig :

```
attributetype ( 1.3.6.1.4.1.23780.219.2.4.1 NAME 'dbisTransAttr' DESC 'LDAP attribute transformation for DBIS database entries' EQUALITY caseIgnoreIA5Match SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
```

---

La représentation chaîne de dbisTransAttr est définie par la grammaire suivante, qui utilise la notation ABNF. Les productions qui ne sont pas définies ici sont définies dans draft-bannister-dbis-netgroup-XX :

```
attrname = keystring
prefix = keystring
suffix = SLASH keystring
incr = PLUS number
decr = HYPHEN number
trans = prefix / suffix / incr / decr
dbisTransAttr = attrname EQUALS trans
```

La valeur de l'attribut attrname apparaissant dans les entrées de base devraient être ré-écrites par le DUA tel sorte qu'il porte le nouveau préfix et/ou suffix. Alternativement, si la valeur de l'attribut est numérique, il doit être incrémenté ou décrétementé en ajoutant ou supprimant le nombre donné. Si l'attribut dbisTransAttr est manquant, le DUA devrait continuer avec les valeurs non-éditées pour la base.

## exactNetgroup

Un ou plusieurs noms netgroup identifiant les noms d'hôte des DUA qui devraient appliquer la map de configuration donnée dans l'attribut exactNetgroup qui peut être assigné à une entrée dbisMapConfig. Si l'attribut exactNetgroup est manquant, le DUA devrait appliquer cette entrée de map de configuration. Si l'attribut existe, le DUA devrait appliquer l'entrée seulement si l'hôte sur lequel il tourne est membre du netgroup donné. Si une entrée matchant est trouvée, le DUA devrait utiliser cette configuration, sinon le DUA doit ignorer cette entrée. La seule exception à ces règles est si le DUA est membre d'un netgroup identifié par l'attribut notNetgroup, qui a précédence.

## noNetgroup

Un ou plusieurs noms netgroup identifiant les noms d'hôte des DUA qui ne devraient pas appliquer la map de configuraion sont donnés dans l'attribut notNetgroup qui peut être assigné à une entrée dbisMapConfig. Cela permet aux entrées de map de configuration d'être exclus d'un groupe d'hôtes particuliers. Le DUA devrait exclure cette entrée de map de configuration si le DUA est membre du netgroup donné, même s'il est également membre d'une attribut exactNetgroup.

## profileTTL

Une valeur TTL peut être assignée à une entrée dbisMapConfig dans l'attribut profileTTL définis dans la rfc4876. Les DUA devraient considérer cet attribut comme ayant précédence au profileTTL fournis dans l'entrée dbisMapConfig, avec le scope limité à cette entrée et toutes entrées qu'elle référence. Si profileTTL est à 0, le DUA peut conserver les copies indéfiniment ou jusqu'à ce que d'autre périodes localement définies soient passées. Si profileTTL est omis de dbisMapConfig, le profileTTL par défaut fournis dans le dbisDomainObject devrait prévaloir.

## negativeTTL

Identique à profileTTL, excepté qu'il est utilisé pour les entrées qui n'existent pas.

## description

---

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

## manager

L'attribut manager peut être associé avec une entrée dbisMapConfig pour fournir un ou plusieurs DN des individus, groupes ou systèmes qui sont responsable pour maintenir l'entrée.

## disableObject

L'attribut disableObject peut être associé avec une entrée dbisMapConfig pour désactiver ce composant de configuration, et est définis comme suit :

```
attributetype ( 1.3.6.1.4.1.23780.219.2.5 NAME 'disableObject' DESC 'TRUE if the entry is disabled' EQUALITY
booleanMatch SINGLE-VALUE SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 )
```

Un DUA devrait ignorer les entrées qui ont l'attribut disableObject mis à TRUE.

## Attributs communs

Les attribut additionnels qui sont soit utilisés dans ce document ou requis par d'autres documents en utilisant le schéma de mappage DBIS sont définis ou référencé ci-dessous.

### en (exactName)

L'attribut en peut être utilisé à la place de cn où la sensibilité à la casse est requise, et est définie comme suit :

```
attributetype ( 1.3.6.1.4.1.23780.219.2.6 NAME ( 'en' 'exactName' ) DESC 'Exact name by which the entity is
known' EQUALITY caseExactMatch SINGLE-VALUE SUBSTR caseExactSubstringsMatch SYNTAX
1.3.6.1.4.1.1466.115.121.1.15{32768} )
```

L'attribut en est identique à l'attribut cn à l'exception qu'il est sensible à la casse et est SINGLE-VALUE.

### rn (regularName)

L'attribut rn peut être utilisé à la place du cn où la casse n'est pas importante mais une seule valeur est permise :

```
attributetype ( 1.3.6.1.4.1.23780.219.2.7 NAME ( 'rn' 'regularName' ) DESC 'Regular name by which the entity
is known' EQUALITY caseIgnoreMatch SINGLE-VALUE SUBSTR caseIgnoreSubstringsMatch SYNTAX
1.3.6.1.4.1.1466.115.121.1.15{32768} )
```

L'attribut rn est identique à l'attribut cn à l'exception qu'il est SINGLE-VALUE.

---

# Syntaxe d'attribut

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

Syntax_OID	Value	Reference
1.3.6.1.4.1.1466.115.121.1.7	Boolean	[RFC4517]
1.3.6.1.4.1.1466.115.121.1.12	DN	[RFC4517]
1.3.6.1.4.1.1466.115.121.1.15	Directory_String	[RFC4517]
1.3.6.1.4.1.1466.115.121.1.26	IA5_String	[RFC4517]

## Caches

Il est commun pour les systèmes d'exploitation d'implémenter leur propres algorithmes de cache de service de noms, par exemple nscd, qui a sont propre TTL. Tou DUA implémentant DBIS devrait honorer le profileTTL et negativeTTL au niveau du domanie et des entrées de map de configuration qui doivent avoir précédence sur les paramètres locaux. Cela peut résulter en différents TTL pas seulement pour les base individuelles mais potentiellement pour les sous-jeux des entrées dans une simple base.