Directory-Based Information Services:
Mapping Objects

Status of this Memo

Distribution of this memo is unlimited.

This Internet-Draft is submitted in full conformance with the
provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering
Task Force (IETF), its areas, and its working groups. Note that
other groups may also distribute working documents as Internet-
Drafts.

Internet-Drafts are draft documents valid for a maximum of six months
and may be updated, replaced, or obsoleted by other documents at any
time. It is inappropriate to use Internet-Drafts as reference
material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at

This Internet-Draft will expire on January 25, 2016.

Comments are solicited and should be addressed to the author.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the
document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal
Provisions Relating to IETF Documents
(http://trustee.ietf.org/license-info) in effect on the date of
publication of this document. Please review these documents
carefully, as they describe your rights and restrictions with respect
to this document.

Abstract
This is one of several documents that describe the components within Directory-Based Information Services (DBIS). DBIS provides a framework for the representation of data relating to TCP/IP and the UNIX system within [X.500] entries that have previously been stored in the Network Information Service [NIS]; so that they may be resolved with the Lightweight Directory Access Protocol [RFC4510].

The intention of DBIS is to extend, and thereby replace both NIS and the experimental protocol for using LDAP as a Network Information Service (RFC2307), which have both achieved widespread adoption.

DBIS consists of an LDAP schema, naming conventions and protocols to describe its use by DUAs requiring network service information. Client/server communication and server-side operations are entirely contained within the domain of LDAP.

Key aspects of DBIS and improvements over RFC2307 are:

- Schema is backwards compatible with NIS, including case sensitivity of key names.
- Standardisation of mapping information to increase portability of DUAs implementations and to reduce duplication of client configuration data.
- Features added to increase flexibility in large complex environments:
  o Maps may be joined from data located in different areas of the Directory Information Tree (DIT).
  o Groups of DUAs may have variances in their data depending upon their host netgroup membership.
- Modular design to allow separate parts of the system to be replaced, improved or augmented separately in the future.
- Support added for automounter maps [draft-bannister-dbis-automounter-00].

This document describes mapping objects used by DBIS to locate and transform service information stored within the DIT.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED" and "MAY" in this document are to be interpreted as described in [RFC2119].

Table of Contents
1. Concepts

2. Domain

3. Configuration Maps

4. Common Attributes

5. Attribute Syntax

6. Implementation Notes

7. Security Considerations

8. References

9. Author’s Address
1.1. Databases

The role of DBIS is to provide a framework that supplies configuration information, chiefly name service data such as login accounts, user groups and host/network lookup information, and any data traditionally provided by [NIS].

Each different type of information is called a "database", as it is a collection of related data entries stored in the DIT. The format of database entries is specific to each type of database and is not defined in this document.

Each database is separately configured using configuration maps that describe where to locate the relevant entries in the DIT. The format of the configuration map is defined in this document, although it may be extended by other documents.

1.2. Aliases

When a database supports alias entries, they are to be configured as described in section 2.6 of [RFC4512]. A DUA SHALL perform alias dereferencing on these databases.

1.3. Exceptions

Except where otherwise noted the behaviour of the DUA is undefined if an attribute used in this document contains a value that does not comply with the format mandated herein.

2. Domain

2.1. Definition

DBIS mapping objects define the components that make up a DBIS domain. A DBIS domain (or "domain"), is a logical grouping of information services required by a common collection of DUAs, in the same way that a NIS domain contains all of the NIS maps required for the correct operation of a group of computers.

A DBIS domain SHALL be identified by an LDAP entry with the object class dbisDomainObject.

Configuration maps for the domain are contained in entries that SHALL be located underneath the dbisDomainObject entry within the DIT.

2.2. Domain Object Classes

2.2.1. dbisDomainObject
The dbisDomainObject class is defined as follows:

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

2.3. Domain Attributes

2.3.1. en

The name of the domain, identical in format to a NIS domain, is stored in the LDAP attribute en which MUST be associated with a dbisDomainObject entry and SHALL form the RDN. The en attribute is defined in section 4.2 of this document.

2.3.2. profileTTL

The default time-to-live value for configuration data pertaining to the domain is set in the profileTTL attribute defined in [RFC4876] which MAY be associated with a dbisDomainObject entry. DUA\S SHOULD keep a local copy of any configuration data obtained from the dbisDomainObject entry and its children, and any data those entries refer to, and MUST NOT use configuration contained in its local copy after the number of seconds defined in the profileTTL have elapsed since the data was obtained, instead obtaining a new copy from the DSA.

If the value of the profileTTL attribute is 0, then the DUA MAY keep its local copies indefinitely or until some other locally defined time period has elapsed. If the dbisDomainObject entry has no profileTTL attribute then the DUA SHALL behave as if the profileTTL was set to 0.

Child entries (dbisMapConfig) underneath the dbisDomainObject MAY possess their own profileTTL attributes, which SHALL override any default profileTTL set on the dbisDomainObject entry both for the child entry and for any configuration data to which that entry refers.

2.3.3. negativeTTL

Identical to a profileTTL attribute, except for entries that do not exist. DUA\S SHOULD keep a local copy of lookups that did not exist but MUST NOT use this data after the number of seconds defined in the negativeTTL have elapsed since the lookup failed.
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 )

2.3.4. description

The description attribute MAY be associated with a dbisDomainObject entry to provide an arbitrary description of the entry.

2.3.5. manager

The manager attribute MAY be associated with a dbisDomainObject entry to provide one or more DNs of the individuals, groups or systems that are responsible for maintaining the entry.

2.4. Domain Aliases

If alias domain names are required then these are configured as described in section 2.6 of [RFC4512]. A DUA SHALL perform alias dereferencing.

2.5. Example Domain Entry

The following is an example of a dbisDomainObject entry in LDIF format [RFC2849]:

dn: en=sales.corp,ou=domain-mappings,o=infra
objectClass: top
objectClass: dbisDomainObject
en: sales.corp
profileTTL: 900
negativeTTL: 300
description: Sales Workforce

3. Configuration Maps

3.1. Definition

A DBIS configuration map instructs a DUA on the location of entries within the DIT for a particular database. It describes how to find the database entries and optionally which subset of DUAs should use those entries (based on netgroup membership).

This document does not define any specific configuration maps, rather
it defines a framework that MUST be followed for the specification of such maps.

Configuration maps SHALL be evaluated by a DUA in lexicographical order of their cn attribute. The order that configuration map entries are evaluated also determines the order in which database entries appear if being sourced from multiple locations. Ordering is also important to ensure that the correct netgroups are available for testing if configuration maps are being restricted by netgroup membership using either the exactNetgroup or notNetgroup attribute.

3.2. Object Classes

3.2.1. dbisMapConfig

A map for any database is optional and SHALL be identified by one or more LDAP entries located underneath the dbisDomainObject entry in the DIT. The behaviour of the DUA if an entry from a database is requested that has no corresponding configuration map is undefined.

Configuration map entries for a single database MUST have the following object class assigned, or a subclass of it:

```
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 ) )
```

A DUA SHALL support multiple configuration map entries for a single database. A database SHALL require at least one additional object class to be assigned to its configuration map entries, which is used to uniquely identify the type of database for which the entries belong.

3.3. Attributes

3.3.1. cn

The cn attribute MUST be used to form the RDN of a dbisMapConfig entry. This is an arbitrary name that has no special meaning within DBIS, but which uniquely identifies the dbisMapConfig entry.

As discussed in section 3.1, configuration map entries are evaluated in lexicographical order of their cn attribute.
3.3.2. dbisMapDN

One or more DNs locating the search base of the database entries in
the DIT are given in the dbisMapDN attribute which MUST be assigned
to a dbisMapConfig entry:

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 )

3.3.3. dbisMapFilter

An LDAP search filter [RFC4515] used for locating the database
entries underneath each dbisMapDN is given in the dbisMapFilter
attribute which MAY be assigned to a dbisMapConfig entry:

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 )

If the dbisMapFilter attribute is missing from the dbisMapConfig
entry then the DUA SHALL use the default filter ‘objectClass=*’.

3.3.4. dbisMapClass

The object classes used to identify the entries for a database can be
changed from the default by the dbisMapClass attribute which MAY be
assigned to a dbisMapConfig entry:

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 )

The string representation of the dbisMapClass attribute is defined by
the following grammar, which uses the ABNF notation defined in
[RFC5234]. The productions used that are not defined here are defined
in section 1.4 of [RFC4512]:

    from_class  = keystring
    to_class    = keystring
    dbisMapAttr = to_class EQUALS from_class

If the dbisMapClass attribute is missing from the dbisMapConfig entry
then the DUA SHALL continue with the default classes for the
database.
Changing this attribute has no effect on the dbisMapFilter, which must be adjusted independently.

3.3.5. dbisMapAttr

The attributes used for storing the database entry’s key and values can be changed from the default by the dbisMapAttr attribute which MAY be assigned to a dbisMapConfig entry:

```
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 )
```

The string representation of the dbisMapAttr attribute is defined by the following grammar, which uses the ABNF notation defined in [RFC5234]. The productions used that are not defined here are defined in section 1.4 of [RFC4512]:

```
from_attr   = keystring
to_attr     = keystring
dbisMapAttr = to_attr EQUALS from_attr
```

The attribute used in the database is identified by from_attr and this SHALL be rewritten by the DUA to the attribute to_attr.

If the dbisMapAttr attribute is missing from the dbisMapConfig entry then the DUA SHALL continue with the default attributes for the database.

Changing this attribute has no effect on the dbisMapFilter nor dbisTransAttr, which must be adjusted independently.

3.3.6. dbisTransAttr

Attribute values used by the database entries may be transformed by the dbisTransAttr attribute which MAY be assigned to a dbisMapConfig entry:

```
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 )
```

The string representation of the dbisTransAttr attribute is defined by the following grammar, which uses the ABNF notation defined in [RFC5234]. The productions used that are not defined here are defined in section 1.2 of [draft-bannister-dbis-netgroup-00]:
attrname      = keystring
prefix        = keystring
suffix        = SLASH keystring
incr          = PLUS number
decr          = HYPHEN number
trans         = prefix / suffix / incr / decr
dbisTransAttr = attrname EQUALS trans

The value of the attribute attrname wherever it appears in the
database entries SHALL be rewritten by the DUA such that it bears the
new string prefix and/or suffix.  Alternatively, if the attribute
value is numeric, then it may be incremented or decremented by adding
or subtracting the given number.

If the dbisTransAttr attribute is missing from the dbisMapConfig
entry then the DUA SHALL continue with the unedited values for the
database.

3.3.7. exactNetgroup

One or more netgroup names identifying the host names of the DUAs
that should apply the configuration map are given in the
exactNetgroup attribute [draft-bannister-dbis-netgroup-00] which MAY
be assigned to a dbisMapConfig entry.

If the exactNetgroup attribute is missing from the dbisMapConfig
entry then the DUA SHALL apply this configuration map entry. If the
attribute exists then the DUA SHALL apply the entry only if the host
on which the DUA is running is a member of the given netgroup.

If a matching entry is found then the DUA SHALL use this
configuration map entry, otherwise the DUA MUST ignore this
configuration map entry.

The only exception to these rules is if the DUA is a member of a
netgroup identified by the notNetgroup attribute, which has
precedence.

3.3.8. notNetgroup

One or more netgroup names identifying the host names of the DUAs
that should NOT apply the configuration map are given in the
notNetgroup attribute [draft-bannister-dbis-netgroup-00] which MAY be
assigned to a dbisMapConfig entry.

This allows configuration map entries to be excluded from particular
groups of hosts.  The DUA SHALL exclude this configuration map entry
if the DUA is a member of the given netgroup, even if the DUA is also
a member of any given exactNetgroup attributes.

3.3.9. profileTTL

A time-to-live value MAY be assigned to a dbisMapConfig entry in the profileTTL attribute defined in [RFC4876]. DUAs SHALL take any such attribute as an override to the profileTTL provided on the dbisDomainObject entry, with the scope limited to this configuration map entry and any entries to which it refers.

If the profileTTL attribute is 0 then the DUA MAY keep its local copies indefinitely or until some other locally defined time period has elapsed. If the profileTTL attribute is omitted from the dbisMapConfig entry then the default profileTTL provided on the dbisDomainObject entry SHALL prevail.

3.3.10. negativeTTL

Identical to a profileTTL attribute, except for entries that do not exist. DUAs SHALL take any such attribute as an override to the negativeTTL provided on the dbisDomainObject entry, with the scope limited to the configuration map entry and any entries to which it refers.

3.3.11. description

The description attribute MAY be associated with a dbisMapConfig entry to provide an arbitrary description of the entry.

3.3.12. manager

The manager attribute MAY be associated with a dbisMapConfig entry to provide one or more DNs of the individuals, groups or systems that are responsible for maintaining the entry.

3.3.13. disableObject

The disableObject attribute MAY be associated with a dbisMapConfig entry to disable this configuration component, and is defined as follows:

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 )

A DUA SHALL ignore entries that have the disableObject attribute set
to TRUE.

4. Common Attributes

4.1. Scope

Additional attributes that are either used within this document or required by other documents using the DBIS mapping scheme are defined or referenced below.

4.2. en (exactName)

The en attribute may be used in place of cn where case sensitivity is required, and is defined as follows:

```
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} )
```

The en attribute is identical to the cn attribute defined in [RFC4519] with the exception that it is case sensitive and SINGLE-VALUE. If multiple names, or aliases, are required for an entry then these are configured as described in section 2.6 of [RFC4512].

4.3. rn (regularName)

The rn attribute may be used in place of cn where case is not important but only a single value is allowed:

```
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} )
```

The rn attribute is identical to the cn attribute defined in [RFC4519] with the exception that it is SINGLE-VALUE. If multiple names, or aliases, are required for an entry then these are configured as described in section 2.6 of [RFC4512].

5. Attribute Syntax

The following syntaxes are used by the attributes defined in this document:
6. Implementation Notes

6.1. Caching

It is common for operating systems to implement their own name service caching algorithms, for example the name service caching daemon (nscd), which have their own TTL configurations for the name service databases. Any DUA implementing DBIS SHALL honour the profileTTL and negativeTTL attribute settings both at the domain level as well as on individual configuration map entries which MUST override any local TTL settings. This can result in different TTLs not just for individual databases but potentially for subsets of entries within a single database.

7. Security Considerations

As this document describes an LDAP schema and a DIT layout it is necessary to ensure that the LDAP entries referred to herein are suitably secured so that only the appropriate administrators for the domain are able to modify entries.

Because of the distributed and modular nature of DBIS configuration maps and their database entries, one has to ensure that referenced DNs are as secure as the domain objects that reference them.

8. References

8.1. Normative References


8.2. Informative References


Author’s Address

Mark R. Bannister
Prose Consulting Ltd.
73 Claygate Lane
Esher, Surrey, KT10 0BQ
United Kingdom