# 静的グループを動的グループに自動反映

## 概要

工学研究科LDAPのグループ（ou=groups配下）を動的なグループに反映する。

ou=groups以下にメンバーやオーナーのdnを定義してグループを表すとき、これらを静的グループと呼ぶことにする。一方、各個人エントリやその他のエントリに、特定の属性を持たせ、そこにグループへのdnを記述することでグループを作成することもできる。これを動的グループと呼ぶ。

このドキュメントは以下のサーバにおいて、静的グループとして定義されたものを、動的なグループに自動で反映する処理について記述する。

- 対象サーバ
  ldap1.kogaku.kyoto-u.ac.jp
  ldap2.kogaku.kyoto-u.ac.jp

## 背景

静的なグループのuniqueMemberやownerに、dnを登録した場合、それらのグループに属するdnを検索して、そのエントリ詳細を得ようとした場合、単純なldapフィルタだけでは書き表せないという問題がある。

- 例
  静的グループに属する各メンバのメールアドレス(mail属性値)一覧は、(1つの)ldapフィルタでは取り出せない。
  これはldapフィルタはあるツリーを対象として、各エントリの持つ属性値をもとに検索するため、静的なグループを参照しながら、各エントリの詳細を取り出す、というような動作はできないためである。そのため、ldapフィルタを使用する場合、静的なグループでは困る場合が多い。

これを解消するために、静的なグループに登録された情報を基に、そのuniqueMemberやownerのdnに対し、専用属性を持たせ、それぞれのエントリから静的なグループへのリンク(memberOf,ownerOf)を持たせるようにする必要がある。このように、メンバーのエントリ自体に、グループへのリンクを示す属性を持たせて識別するグループのことを動的グループと呼ぶ。

## 動作仕様

ldapに登録されたou=groups配下の各エントリのうち、動的に反映する必要のあるものには、特別なobjectClassを付与するものとする。このobjectClassが付与されたエントリに対し、動的グループへの反映処理を行なう。付与されていないものについては反映処理を行なわない。
また、ou=groups配下のエントリには、uniqueMemberやownerの属性を持つグループが定義されるが、uniqueMemberとownerのいずれか一方、あるいは両方を反映するという指定が可能となるよう、objectClassも2つ用意する。

- 静的グループについて（反映元）
  - 反映される静的グループのベースエントリ
    ou=groups,o=kogaku,dc=kyoto-u,dc=ac,dc=jp
  - uniqueMember属性を反映させるときに付与するobjectClass
    objectClass: kyoto-uDynGroupOfUniqueMember

    - owner属性を反映させるときに付与するobjectClass
      objectClass: kyoto-uDynGroupOfOwner
- 動的グループについて（反映先）
    - 反映される動的グループのベースエントリ
      o=kogaku,dc=kyoto-u,dc=ac,dc=jp
    - uniqueMemberを表す属性
      kyoto-uEduPersonUniqueMemberOf: （元の静的グループへのdn）
    - ownerを表す属性
      kyoto-uEduPersonOwnerOf: （元の静的グループへのdn）
      なお、反映されるエントリには必ずobjectClass: kyoto-uEduPersonが付与されている必要がある。これをもたないエントリには反映されない。

# サーバ設定

## ldapのschema設定

/etc/openldap/schema/kyoto-uEduPerson.schemaを編集して、以下のattributeおよびobjectClassを追加する。

- attribute

```
attributetype ( feideAttribute:8
        NAME 'kyoto-uEduPersonUniqueMemberOf'
        DESC 'Group DN which containing this entry as uniqueMember.'
        EQUALITY distinguishedNameMatch
        SYNTAX '1.3.6.1.4.1.1466.115.121.1.12' )

attributetype ( feideAttribute:9
        NAME 'kyoto-uEduPersonOwnerOf'
        DESC 'Group DN which containing this entry as a owner.'
        EQUALITY distinguishedNameMatch
```

- objectClass

```
objectclass ( feideObjClass:3
        NAME 'kyoto-uEduPerson'
        SUP eduPerson
        AUXILIARY
        MAY ( kyoto-uEduPersonOIN $ kyoto-uEduPersonBirthDate $
              kyoto-uEduPersonLIN $ kyoto-uEduPersonPrivate $
              kyoto-uEduPersonUniqueMemberOf $ kyoto-uEduPersonOwnerOf) )

objectclass ( feideObjClass:4
        NAME 'kyoto-uDynGroupOfUniqueMember'
        SUP top
        AUXILIARY
        MUST ( uniqueMember ) )

objectclass ( feideObjClass:5
        NAME 'kyoto-uDynGroupOfOwner'
        SUP top
        AUXILIARY
        MUST ( owner ) )
```

## 反映用ldapユーザ作成

ldapエントリのou=users配下に反映用ldapユーザを追加する。またadminグループにそのユーザを所属させて、任意のエントリについて読み書き可能とする。

- 追加ldapユーザ

```
dn: cn=dyngroup,ou=users,o=kogaku,dc=kyoto-u,dc=ac,dc=jp
objectClass: applicationProcess
objectClass: shadowAccount
cn: dyngroup
uid: dyngroup
userPassword: secret
description: static group to dynamic group
description;lang-ja: 静的グループを動的グループに反映するためのユーザ
```

- adminグループに追加

```
dn: cn=admin,ou=groups,o=kogaku,dc=kyoto-u,dc=ac,dc=jp
objectClass: groupOfUniqueNames
cn: admin
（省略）
uniqueMember: cn=dyngroup,ou=users,o=kogaku,dc=kyoto-u,dc=ac,dc=jp
（省略）
```

ldapサーバを再起動する。

```
/sbin/service ldap restart
```

## 反映スクリプト作成

自動反映するスクリプトを作成する。まずスクリプト格納用のディレクトリを作成する。

```
[root@ldap1 home]# ls -ld /home/ldap
drwx------  2 ldap ldap 4096  9月 13 15:11 /home/ldap
```

/home/ldap配下に以下の3ファイルを作成する。（内容は dyngroup.tar.gz参照）

```
-rwx------  1 ldap ldap 1070  9月 13 15:10 chk_modify_ldap
-rwx------  1 ldap ldap 7454  9月 13 15:10 dyngroup
-rw-------  1 ldap ldap   20  9月 13 15:21 dyngrp_date.txt
```

- 反映スクリプト
  /home/ldap/dyngroupとして作成する。反映処理自体を行なう。直接は呼び出さない。
- ldap更新チェックスクリプト
  /home/ldap/chk_modify_ldapとして作成する。ldapデータベースの更新日付（/var/lib/ldap以下ファイルの最も新しい更新日付）を取得し、前回更新時の日付と比較する。前回更新の後にldapデータベースが更新されていれば、dyngroupを呼び出して、反映処理を行なう。
- 前回反映時刻保存ファイル
  dyngrp_date.txtとしてtouchしておく。chk_modify_ldapから、このファイルを読み書きする。

## 自動実行登録

crontabに登録する。実行ユーザはldapとする。
rootになり、

```
sudo -u ldap crontab -e
```

で編集できる。以下の内容を登録する。

```
0-59/2 * * * * /home/ldap/chk_modify_ldap &
```

# 実行例

以下、静的グループへの登録と動的グループへの反映例を示す。

- 静的グループ（cn=test,ou=groups...）を以下のように設定

  ```
  dn: cn=test,ou=groups,o=kogaku,dc=kyoto-u,dc=ac,dc=jp
  objectClass: groupOfUniqueNames
  objectClass: kyoto-uDynGroupOfUniqueMember
  objectClass: kyoto-uDynGroupOfOwner
  cn: test
  uniqueMember: cn=foo,ou=users,o=kogaku,dc=kyoto-u,dc=ac,dc=jp
  uniqueMember: uid=bar,ou=people,o=kogaku,dc=kyoto-u,dc=ac,dc=jp
  owner: cn=foo,ou=users,o=kogaku,dc=kyoto-u,dc=ac,dc=jp
  owner: uid=bar,ou=people,o=kogaku,dc=kyoto-u,dc=ac,dc=jp
  ```

  objectClassとして「kyoto-uDynGroupOfUniqueMember」「kyoto-uDynGroupOfOwner」を持っているので、「uniqueMember」「owner」属性について、それぞれ動的グループへと反映される。
- 反映後の各エントリ
  cronにより、1分以内に反映処理が行なわれる。反映後、それぞれのエントリは以下のとおりとなる。
  （反映される例）

  ```
  dn: uid=bar,ou=people,o=kogaku,dc=kyoto-u,dc=ac,dc=jp
  uid: bar
  objectClass: person
  objectClass: organizationalPerson
  objectClass: inetOrgPerson
  objectClass: eduPerson
  objectClass: kyoto-uEduPerson
  （省略）
  kyoto-uEduPersonUniqueMemberOf: cn=test,ou=groups,o=kogaku,dc=kyoto-u,dc=ac,dc=jp
  kyoto-uEduPersonOwnerOf: cn=test,ou=groups,o=kogaku,dc=kyoto-u,dc=ac,dc=jp
  ```

  （反映されない例：objectClass: kyoto-uEduPersonを持たないため）

  ```
  dn: cn=foo,ou=users,o=kogaku,dc=kyoto-u,dc=ac,dc=jp
  objectClass: applicationProcess
  objectClass: shadowAccount
  cn: foo
  uid: foo
  （省略）
  ```

- 注意
  - 動的グループは常に静的グループと対応するように反映される。
    したがって静的グループのuniqueMemberやowner属性からメンバーを削除したときも、対応するエントリのkyoto-uEduPersonUniqueMemberOfやkyoto-uEduPersonOwnerOfの属性が削除される。
    （正確には、削除された静的グループへのdnを属性値として持つ属性と属性値の組が削除される）
  - 静的グループ自体が削除されたとき
    静的グループ自体が削除されたときは、削除されたグループに属するように反映されていた各エントリから、対応する属性と属性値が削除される。
  - 静的グループのobjectClassが変更されたとき
    静的グループのobjectClassのうち、「kyoto-uDynGroupOfUniqueMember」または「kyoto-uDynGroupOfOwner」が削除された場合、対応する属性について、このグループに属するように反映されていた各エントリから、対応する属性と属性値が削除される。