OPENLDAPのバックエンドをbdbからmdbに入れ替えついでにOLC(cn=config)対応する

2016/10/07
OPENLDAP 2.4.44 でbdb非推奨になったのを受け、mdbへの移行を行う自分用メモ。
slapd.conf はまだ使えるけど後で泣いても知らないよー、と脅しがあったのでon-line configuration (OLC) へ移行を行う自分用メモ。

2016/10/07-2
実運用環境で mdb に移るときは maxsize の指定を必ずやっとくように。 私が対応してきた環境ではエントリーが5万を超えた辺りで“index generation failed”とか言われてldapaddコマンドで1エントリーずつしか追加できなくなってしまった。 どうもmaxsizeの指定が無いとデフォルトの小さなサイズ(10485760B→10MB)で処理を行おうとしてエントリーの追加についていけなくなってしまうみたい。 たぶん1エントリーずつなら出来ていた追加もすぐ限界が来ると思う。サンプルにあったサイズ(1073741824B→1GB)なら問題なく処理できた。

環境は FreeBSD/amd64 10.3-RELEASE□実環境で実行する前にプライベート環境で実験している。

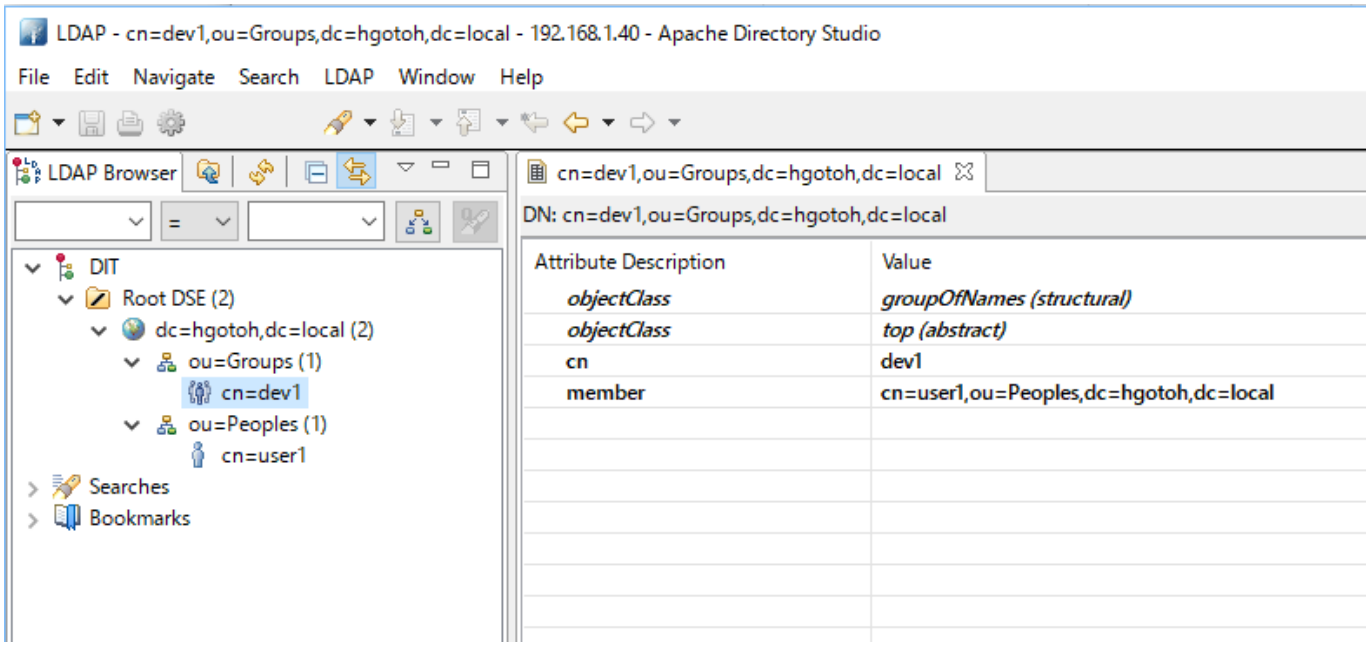
バックエンドデータベースを bdb から mdb へ変更する

いつの頃からかバックエンドデータベースとして bdb が非推奨になりPortsのデフォルトオプションでも bdb が外されていたので mdb に変更する。

まだ指定すれば使えるけど□DEPRECATED(非推奨)になっちゃってる。

bdb稼働状態のOPENLDAPからLDIF形式でデータをダンプする

bdbからmdbへの変換パスは無い様なので、バックアップ・リストアの手順を踏むことになる Apache Directory Studio で参照するとこんなツリーをダンプする。



slapcatコマンドを使ってLDIF形式でダンプを取得した。

```
root@openldap:/usr/local/etc/openldap # slapcat -b "dc=hgotoh,dc=local" -l
backup.ldif
57f66986 bdb_db_open: warning - no DB_CONFIG file found in directory
/var/db/openldap-data: (2).
Expect poor performance for suffix "dc=hgotoh,dc=local".
root@openldap:/usr/local/etc/openldap #
```

取得したLDIF形式ダンプから特定のアトリビュート指定を取り除く

ここで取得したLDIF形式ダンプをldapaddコマンドで投入し直すことになるが、このままではエラーとなる。 以下はバックエンドをmdbに入れ替えしたOPENLDAPに投入したところ発生したエラーの一部。

```
root@openldap:/usr/local/etc/openldap # ldapadd -x -D
"cn=Manager,dc=hgotoh,dc=local" -W -f backup.ldif
Enter LDAP Password:
adding new entry "dc=hgotoh,dc=local"
ldap_add: Constraint violation (19)
        additional info: structuralObjectClass: no user modification allowed

root@openldap:/usr/local/etc/openldap #
```

各エントリーにあるアトリビュート指定行をいくつか削除しておく必要があった。 今回の作業では以下の行を各エントリーの定義から削除した。もっと種類があるのかもしれないが私にはわからない。

- structuralObjectClass
- entryUUID
- creatorsName
- createTimestamp
- entryCSN
- modifiersName
- modifyTimestamp

変更前	変更後
dn: dc=hgotoh,dc=local objectClass: organization objectClass: dcObject dc: hgotoh o: hgotoh inc structuralObjectClass: organization entryUUID: ae888424-2024-1036-9c5c-27e7a8b8c544 creatorsName: cn=Manager,dc=hgotoh,dc=local createTimestamp: 20161006152413Z entryCSN: 20161006152413.646899Z#000000#000#000000 modifiersName: cn=Manager,dc=hgotoh,dc=local modifyTimestamp: 20161006152413Z dn: ou=Peoples,dc=hgotoh,dc=local objectClass: organizationalUnit ou: Peoples structuralObjectClass: organizationalUnit entryUUID: ae88b7c8-2024-1036-9c5d-27e7a8b8c544 creatorsName: cn=Manager,dc=hgotoh,dc=local createTimestamp: 20161006152413Z entryCSN: 20161006152413.648232Z#000000#000#000000 modifiersName: cn=Manager,dc=hgotoh,dc=local modifyTimestamp: 20161006152413Z dn: ou=Groups,dc=hgotoh,dc=local objectClass: organizationalUnit ou: Groups structuralObjectClass: organizationalUnit entryUUID: ae88e25c-2024-1036-9c5e-27e7a8b8c544 creatorsName: cn=Manager,dc=hgotoh,dc=local createTimestamp: 20161006152413Z entryCSN: 20161006152413.649325Z#000000#000#000000 modifiersName: cn=Manager,dc=hgotoh,dc=local modifyTimestamp: 20161006152413Z dn: cn=user1,ou=Peoples,dc=hgotoh,dc=local objectClass: top objectClass: person cn: user1 sn: user1 structuralObjectClass: person entryUUID: ae8911dc-2024-1036-9c5f-27e7a8b8c544 creatorsName: cn=Manager,dc=hgotoh,dc=local createTimestamp: 20161006152413Z entryCSN: 20161006152413.650540Z#000000#000#000000 modifiersName: cn=Manager,dc=hgotoh,dc=local modifyTimestamp: 20161006152413Z dn: cn=dev1,ou=Groups,dc=hgotoh,dc=local objectClass: top objectClass: groupOfNames cn: dev1 member: cn=user1,ou=Peoples,dc=hgotoh,dc=local structuralObjectClass: groupOfNames entryUUID: ae893ff4-2024-1036-9c60-27e7a8b8c544 creatorsName: cn=Manager,dc=hgotoh,dc=local createTimestamp: 20161006152413Z entryCSN: 20161006152413.651718Z#000000#000#000000 modifiersName: cn=Manager,dc=hgotoh,dc=local modifyTimestamp: 20161006152413Z	dn: dc=hgotoh,dc=local objectClass: organization objectClass: dcObject dc: hgotoh o: hgotoh inc dn: ou=Peoples,dc=hgotoh,dc=local objectClass: organizationalUnit ou: Peoples dn: ou=Groups,dc=hgotoh,dc=local objectClass: organizationalUnit ou: Groups dn: cn=user1,ou=Peoples,dc=hgotoh,dc=local objectClass: top objectClass: person cn: user1 sn: user1 dn: cn=dev1,ou=Groups,dc=hgotoh,dc=local objectClass: top objectClass: groupOfNames cn: dev1 member: cn=user1,ou=Peoples,dc=hgotoh,dc=local

OPENLDAP停止

以下のコマンドで停止。

```
service slapd stop
```

slapd.conf書き換え

slapd.confの該当する記述をmdb用に書き換えする。

変更前	変更後
moduleload back_bdb	moduleload back_mdb
database bdb	database mdb maxsize 1073741824

データベースディレクトリのバックアップ

bdbのデータベースファイルがディレクトリ /var/db/openldap-data に格納されているので(slapd.confに記述がある)、このディレクトリを /var/db/openldap-data.backup にリネームする。mdbバックエンドで正しく動作しているようならあとで削除する。

ディレクトリはmdbバックエンドでの初回起動時に作成されるので、作り直さなくてもよい。

OPENLDAP開始

以下のコマンドで開始。

```
service slapd start
```

LDIF形式ダンプを取り込む

先に編集を済ませたLDIF形式ダンプをldapaddコマンドで取り込む。

```
root@openldap:/usr/local/etc/openldap # ldapadd -x -D
"cn=Manager,dc=hgotoh,dc=local" -W -f backup.ldif
Enter LDAP Password:
adding new entry "dc=hgotoh,dc=local"

adding new entry "ou=Peoples,dc=hgotoh,dc=local"

adding new entry "ou=Groups,dc=hgotoh,dc=local"

adding new entry "cn=user1,ou=Peoples,dc=hgotoh,dc=local"

adding new entry "cn=dev1,ou=Groups,dc=hgotoh,dc=local"
```

```
root@openldap:/usr/local/etc/openldap #
```

再度Apache Directory StudioでOPENLDAPに接続し、参照できることを確認した。

注意

createTimestampやmodifyTimestampといったオペレーショナルなアトリビュートを移せないので、これらをあてにしているアプリケーションは問題を起こすだろう。 必要であれば他の方法を探すこと。

slapd.confからOLC(cn=config)へ移行する

最近ではOPENLDAPの設定記事でこちらの説明をしている記事が増えてきている。また、はやいうちにOLCに移行した方がいいぞー、と脅してるブログ記事があったり。

小心者なのでやれるうちにやっておくことにする。

OPENLDAP停止

以下のコマンドで停止。

```
service slapd stop
```

/etc/rc.conf に追加

FreeBSDの場合、/etc/rc.conf にOLCを使って起動するよう記述の追加が必要。

```
slapd_cn_config="YES"
```

/usr/local/etc/openldap/slapd.conf を編集

次の3行を、database mdbの記述行より前に記述。

```
database        config
rootdn          "cn=admin,cn=config"
rootpw          config
```

database mdb記述行以降を削除しないこと。slaptestコマンドがマイグレーションを行う時の参照情報の為。

slaptestコマンドでマイグレーションする

slapd.confの内容からマイグレーション用ファイルが生成され、ディレクトリ /usr/local/etc/openldap/slapd.d の中に格納される。

```
root@openldap:/usr/local/etc/openldap # cp slapd.conf slapd.conf.backup
root@openldap:/usr/local/etc/openldap # mkdir slapd.d
root@openldap:/usr/local/etc/openldap # slaptest -f slapd.conf -F slapd.d
config file testing succeeded
root@openldap:/usr/local/etc/openldap # ls -l slapd.d
total 8
drwxr-x---  3 root  wheel   512 10月  6 15:40 cn=config/
-rw-------  1 root  wheel  1030 10月  6 15:40 cn=config.ldif
root@openldap:/usr/local/etc/openldap #
```

直接書き換えるとCRCエラーになるのでやっちゃダメ。

OPENLDAP開始

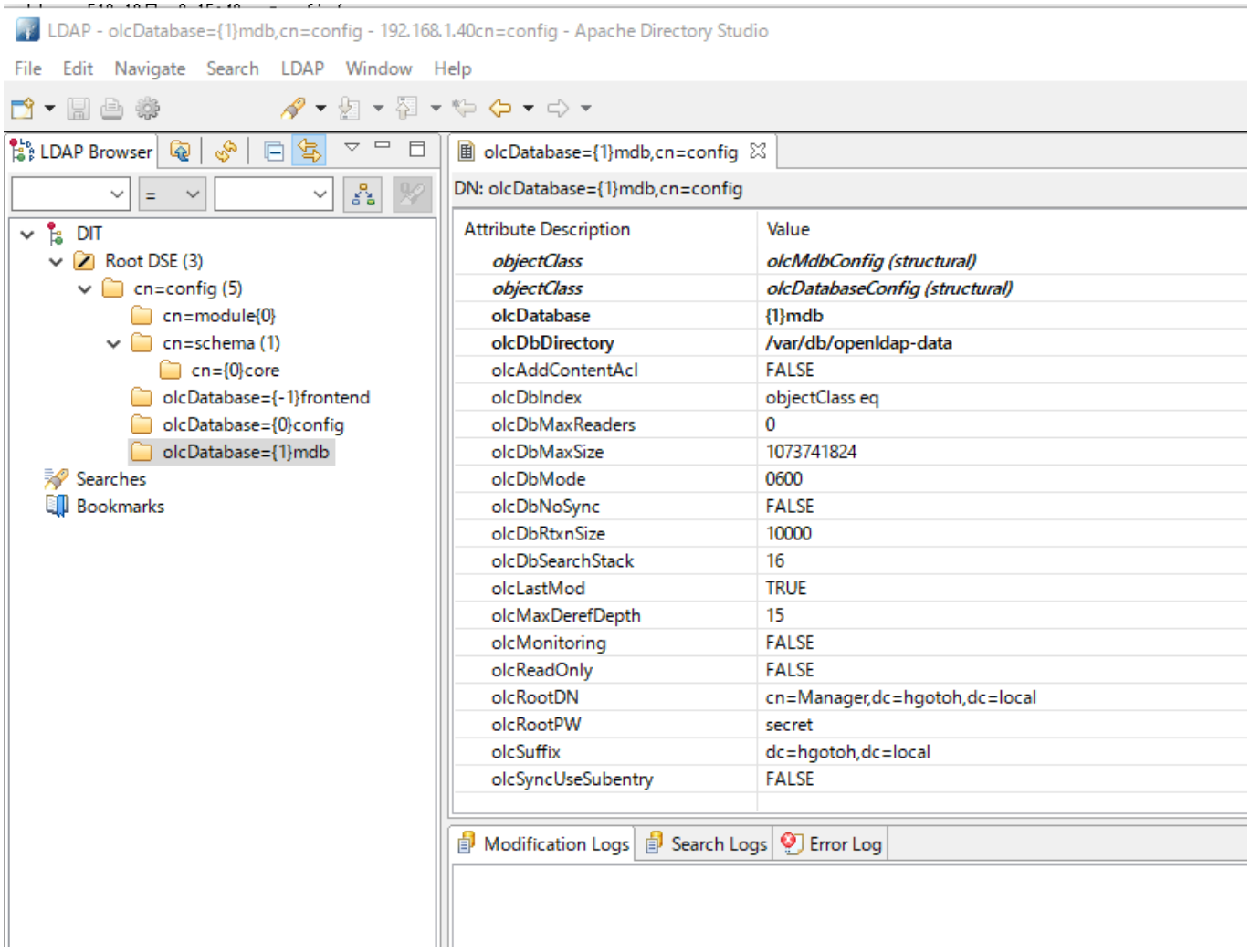
以下のコマンドで開始。

```
service slapd start
```

再度Apache Directory StudioでOPENLDAPに接続し、参照できることを確認した。

cn=configを参照してみる

slapd.conf に記述した database config はcn=config のスキーマに対応する指定Apache Directory Studio でこのスキーマを参照してみる。バインドするユーザは cn=admin,cn=config でパスワードは config になるslapd.conf に記述したあれと一緒。



slapd.conf に記述していた database mdb の内容がLDAPのツリーから参照できている。

slapd.confを変更した場合はサービス再起動が必要だったが、OLCならその必要はない。LDAP上のエントリを書き換えするとそのまま内容が反映される。はず。 そしてその変更はディレクトリ slapd.d 以下のファイルに反映される。

一度OLCで稼働すれば slapd.conf は不要な筈だけど、他の記事では消すことをしていないような……気持ちはわかる。

LDAP, OPENLDAP, FreeBSD, cn=config, OLC, 技術資料