

---

## Configure Variant

### Attribute sharing

It is not completely rare that, when managing a complex directory, some values need to be shared between entries.

As LDAP is not a protocol that has a relational data model (e.g. being able to answer queries like `"list entries having an attribute a with value x and whose parent doesn't have attribute b present"`), and extending it this way would probably lead to losing a lot of the benefits, especially performance-wise. It is for this reason that many varying approaches have been devised to help with this:

- aliases: will completely replace the entry with another
- referral: similar to aliases
- [collective attributes](<https://tools.ietf.org/html/rfc3671>): an attribute is propagated to child entries
- [dereference control](<https://tools.ietf.org/html/draft-masarati-ldap-deref-00>): another set of values can be returned as part of the search query based on a DN-valued attribute
- ``slapo-dynlist``: an attribute specifying an additional entry is referenced to retrieve certain values

Obviously, each of them is more suited for a different use case, aliases and referrals will only replace an entry with another, no mixing and matching is possible. Not all collective attribute implementations allow for the collective attribute to be used in a filter and the modification never propagates to the source entry. Likewise with dereference control or ``slapo-dynlist``, the values can only be returned as part of a search response, no filtering on the referenced values or modification is permitted or defined.

If there is a need to share only some attributes, yet allow modification to happen, the fallback has always been to bake this into the provisioning systems or setting up a reactive process to propagate that when a change is detected.

Yet another option is to leverage the server and implement the business logic into it. While this will never approach the flexibility of a relational database, it might still get close enough to solving the issue to be useful.

### Variant overlay

One implementation that attempts to cover some areas where the above standard approaches were falling short is the variant overlay for OpenLDAP. Using this overlay, an attribute and its values can be seamlessly shared between several entries at the same time, with modifications taking effect instantaneously.

The overlay allows some entries in the database to be configured as so-called variant entries. These entries can inherit any of their attributes from other entries (alternative entries), possibly under a different name. Whenever that entry is being considered, it is as if the attribute was a part of it, for the purposes of filtering, access control and modification.

Internally, the values are always stored in the alternative entry and should the variant entry already contain the attribute, it will not be shown. Attempting to add an entry configured as a

variant with a variant attribute will be rejected, modifying the variant attributes will perform the necessary modifications on the respective alternative entries.

### Regex variants

Variant entries can also be defined more generally specifying a regular expression to be tested against their DN. These entries are, however, not as transparent to the client as regular variant entries, see below.

With regex variants, the regular expression can contain capture patterns and anything matched by them is then available when locating the alternative entry for each attribute. This is great when the alternative entry might be different for each variant entry and there are easy to describe rules to locate it.

When compared to the regular variants, a drawback is that searches for regex entries will only be processed as such if requested specifically and with scope set to `base`.

### Configuration

The variant overlay is available in Symas OpenLDAP builds from version **\*\*FIXME\*\*** onwards.

1. Before you can use it, instruct the server to load the module, then add it to the database you intend to use it with. If you replicate that database, you want this overlay to be configured on each server and consult its manual page (`man 5 slapo-variant`) for more information.

```
ldapmodify -x -D cn=config -W -H ldap://localhost
dn: cn=module{0},cn=config
changetype: modify
add: olcModuleLoad
olcModuleLoad: variant.la

dn: olcOverlay=variant,$DATABASE
changetype: add
objectClass: olcVariantConfig
```

2. Now you can specify that an entry is to be treated as a variant, suppose we want `dc=example,dc=com` to be a regular variant entry and ``cn=(.*)],[^,]*),dc=example,dc=com`` to be a regex variant:

```
ldapmodify -x -D cn=config -W -H ldap://localhost
dn: name=example,olcOverlay={x}variant,$DATABASE
changetype: add
objectClass: olcVariantVariant
olcVariantEntry: dc=example,dc=com

dn: name=example 2,olcOverlay={x}variant,$DATABASE
changetype: add
objectClass: olcVariantRegex
olcVariantEntry: cn=(.)*],[^,]*),dc=example,dc=com
```

3. If we want the first entry to share its address with where its headquarters are, we might configure the attribute like this:

```
dn:
olcVariantVariantAttribute=postaladdress,name={0}example,olcOverlay={x}variant,$DATABASE
objectClass: olcVariantVariantAttribute
olcVariantVariantAttribute: postaladdress
olcVariantAlternativeAttribute: postaladdress
olcVariantAlternativeEntry: ou=Headquarters,dc=example,dc=com
```



# Symas OpenLDAP

## How-To Guides

4. The phone number recorded should be the same as the CEO's home phone, we make sure that the number is taken from the correct attribute (homephone) on the alternative entry:  
dn:  
    olcVariantVariantAttribute=telephonenumber,name={0}example,olcOverlay={x}  
    }variant,\$DATABASE  
objectClass: olcVariantVariantAttribute  
olcVariantVariantAttribute: telephonenumber  
olcVariantAlternativeAttribute: homephone  
olcVariantAlternativeEntry:cn=John Doe,ou=People,dc=example,dc=com
5. If we stored some attributes elsewhere, the regex attribute can help locate them for us:  
dn: olcVariantVariantAttribute=location,name={1}example  
2,olcOverlay={x}variant,\$DATABASE  
objectClass: olcVariantAttributePattern  
olcVariantVariantAttribute: location  
olcVariantAlternativeAttribute: location  
olcVariantAlternativeEntryPattern:  
name=\$1,ou=alternative,\$2,dc=another,dc=com

### Complete cn=config Example

```
dn: cn=module{0},cn=config
changetype: modify
add: olcModuleLoad
olcModuleLoad: variant.la

dn: olcOverlay={0}variant,olcDatabase={1}mdb,cn=config
changetype: add
objectClass: olcOverlayConfig
objectclass: olcVariantConfig

dn: olcOverlay={0}variant,olcDatabase={1}mdb,cn=config
changetype: modify
replace: olcVariantPassReplication
olcVariantPassReplication: TRUE

dn: name={0}variant,olcOverlay={0}variant,olcDatabase={1}mdb,cn=config
changetype: add
objectclass: olcVariantVariant
olcVariantEntry: ou=People,dc=example,dc=com

# a basic variant
dn:
olcVariantVariantAttribute=description,name={0}variant,olcOverlay={0}variant,o
lcDatabase={1}mdb,cn=config
changetype: add
objectclass: olcVariantAttribute
olcVariantAlternativeAttribute: description
olcVariantAlternativeEntry: dc=example,dc=com

# a nonexistent alternate
dn:
olcVariantVariantAttribute=seealso,name={0}variant,olcOverlay={0}variant,olcDa
tabase={1}mdb,cn=config
```





# Symas OpenLDAP

## How-To Guides

```
changetype: add
objectclass: olcVariantAttribute
olcVariantAlternativeAttribute: seealso
olcVariantAlternativeEntry: ou=Societies,dc=example,dc=com

dn: name={1}variant,olcOverlay={0}variant,olcDatabase={1}mdb,cn=config
changetype: add
objectclass: olcVariantVariant
olcVariantEntry: ou=Groups,dc=example,dc=com

# recursive retrieval is not done
dn:
olcVariantVariantAttribute=description,name={1}variant,olcOverlay={0}variant,olcDatabase={1}mdb,cn=config
changetype: add
objectclass: olcVariantAttribute
olcVariantAlternativeAttribute: description
olcVariantAlternativeEntry: ou=People,dc=example,dc=com

# a variant taking data from a different attribute (after the changes below)
dn:
olcVariantVariantAttribute=st,name={1}variant,olcOverlay={0}variant,olcDatabase={1}mdb,cn=config
changetype: add
objectclass: olcVariantAttribute
olcVariantAlternativeAttribute: st
olcVariantAlternativeEntry: cn=Manager,dc=example,dc=com

# configuration changes
dn:
olcVariantVariantAttribute={1}st,name={1}variant,olcOverlay={0}variant,olcDatabase={1}mdb,cn=config
changetype: modify
replace: olcVariantAlternativeAttribute
olcVariantAlternativeAttribute: ou
-
replace: olcVariantAlternativeEntry
olcVariantAlternativeEntry: ou=Alumni Association,ou=People,dc=example,dc=com
-

# a regex variant
dn: name={2}regex,olcOverlay={0}variant,olcDatabase={1}mdb,cn=config
changetype: add
objectclass: olcVariantRegex
olcVariantEntryRegex: (.*) ,(ou=.*technology.*) (,)dc=example,dc=com

dn:
olcVariantVariantAttribute=ou,name={2}regex,olcOverlay={0}variant,olcDatabase={1}mdb,cn=config
changetype: add
objectclass: olcVariantAttributePattern
olcVariantAlternativeAttribute: ou
olcVariantAlternativeEntryPattern: $2$3dc=example$3dc=com
```





# Symas OpenLDAP

## How-To Guides

```
# Duplicate description into title
dn:
olcVariantVariantAttribute=title,name={2}regex,olcOverlay={0}variant,olcDatabase={1}mdb,cn=config
changetype: add
objectclass: olcVariantAttributePattern
olcVariantAlternativeAttribute: description
olcVariantAlternativeEntryPattern: $0

# everything
dn: name={3}regex,olcOverlay={0}variant,olcDatabase={1}mdb,cn=config
changetype: add
objectclass: olcVariantRegex
olcVariantEntryRegex: .*

dn:
olcVariantVariantAttribute=1,name={3}regex,olcOverlay={0}variant,olcDatabase={1}mdb,cn=config
changetype: add
objectclass: olcVariantAttributePattern
olcVariantAlternativeAttribute: 1
olcVariantAlternativeEntryPattern: dc=example,dc=com
```

### Complete slapd.conf Example

```
include          ./schema/core.schema
include          ./schema/cosine.schema
include          ./schema/inetorgperson.schema
include          ./schema/openldap.schema
include          ./schema/nis.schema

pidfile          slapd.pid
argsfile         slapd.args

moduleload       back_mdb.la
moduleload       variant.la

database         monitor

database         mdb
suffix           "dc=example,dc=com"
rootdn           "cn=Manager,dc=example,dc=com"
rootpw           secret
directory        ./db
index            objectClass      eq
index            cn,sn,uid         pres,eq,sub

overlay          variant
passReplication TRUE

variantDN        ou=People,dc=example,dc=com
variantSpec      seealso seealso ou=Societies,dc=example,dc=com
```





# Symas OpenLDAP

## How-To Guides

```
variantSpec description description dc=example,dc=com
```

```
variantDN ou=Groups,dc=example,dc=com
```

```
variantSpec st ou "ou=Alumni Association,ou=People,dc=example,dc=com"
```

```
variantSpec description description ou=People,dc=example,dc=com
```

```
variantRegex "(.*) ,(ou=.*technology.*)" (,)dc=example,dc=com"
```

```
variantRegexSpec title description $0
```

```
variantRegexSpec ou ou "$2$3dc=example$3dc=com"
```

```
variantRegex .*
```

```
variantRegexSpec l l dc=example,dc=com
```