Introduction to LDAP

Brad Marshall

bmarshal@pisoftware.com

Plugged In Software

History of LDAP

- Originally started as a front end to X.500
- Provides much of X.500’s functionality at a lower implementation cost
- Removed redundant and rarely used operations
- Uses TCP rather than OSI stack
- Univerity of Michigan wrote first LDAP implementation
- Most early LDAP implementations were based on it
- U.Mich eventually realised didn’t need X.500 and wrote lightweight server
- Meant it was easier to deploy, and more people started using it

What is LDAP?

- LDAP = Lightweight Directory Access Protocol
- Based on X.500
- Directory Service (RFC1777)
- Stores attribute based data
- Data generally read more than written to
  - No transactions
  - No rollback
- Client-server model
- Based on entries
  - Collection of attributes
  - Has a distinguished name (DN) - like domain name

Why use LDAP

- Centrally manage users, groups and other data
- Don’t have to manage separate directories for each application - stops the “N + 1 directory problem”
- Distribute management of data to appropriate people
- Allow users to find data that they need
- Not locked into a particular server
- Ability to distribute servers to where they are needed

LDAP vs Databases

- Read-write ratio - LDAP is read optimised
- Extensibility - LDAP schemas are more easily changed
- Distribution - with LDAP data can be near where it is needed
- Replication - with LDAP data can be stored in multiple locations
- Different performance - databases are generally deployed for limited amount of applications

LDAP vs Databases cont

- Transaction model - LDAP transactions are simple - usually changing one entry, databases can modify much more
- Size of information - LDAP is better at storing small bits of information
- Type of information - LDAP stores information in attributes
- Standards are more important for directories - LDAP clients can talk to any LDAP server, but database client can only talk to the database it was designed for

LDAP vs NIS

- Uses arbitrary ports
- No data encryption
- No access-control mechanism
- Uses a flat (non scalable) namespace
- Uses a single-key database (providing only basic searching abilities)
- All changes had to be made by the superuser on the domain master
- Does not provide directory services for non nameservice applications

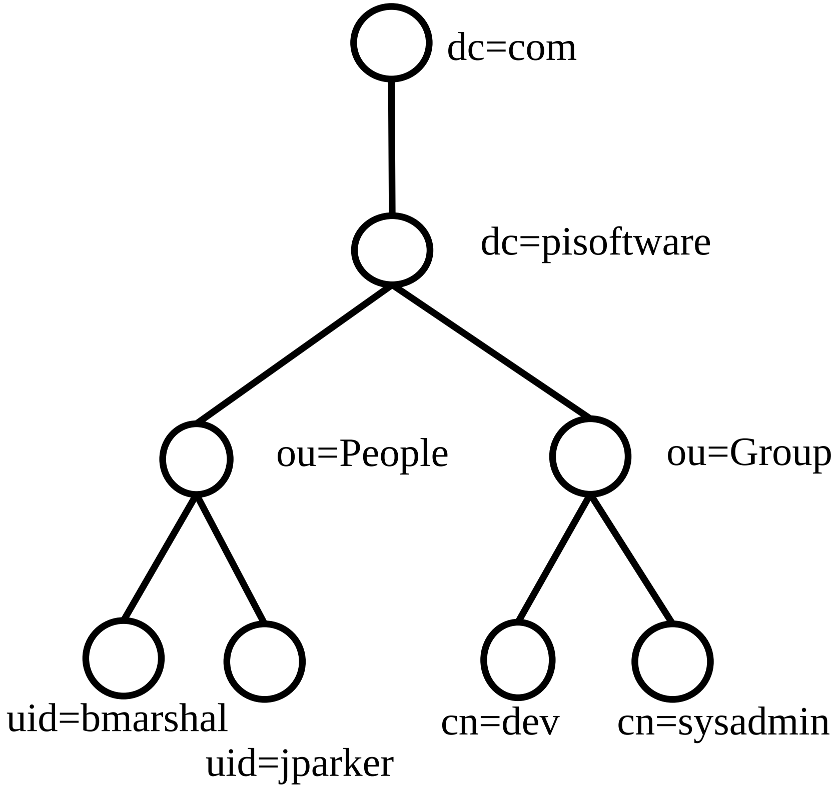
Acronym

LDAP	Lightweight Directory Access Protocol
DN	Distinguish Name
RDN	Relative Distinuished Name
DIT	Directory Information Tree
LDIF	LDAP Data Interchange Format
OID	Object Identifier

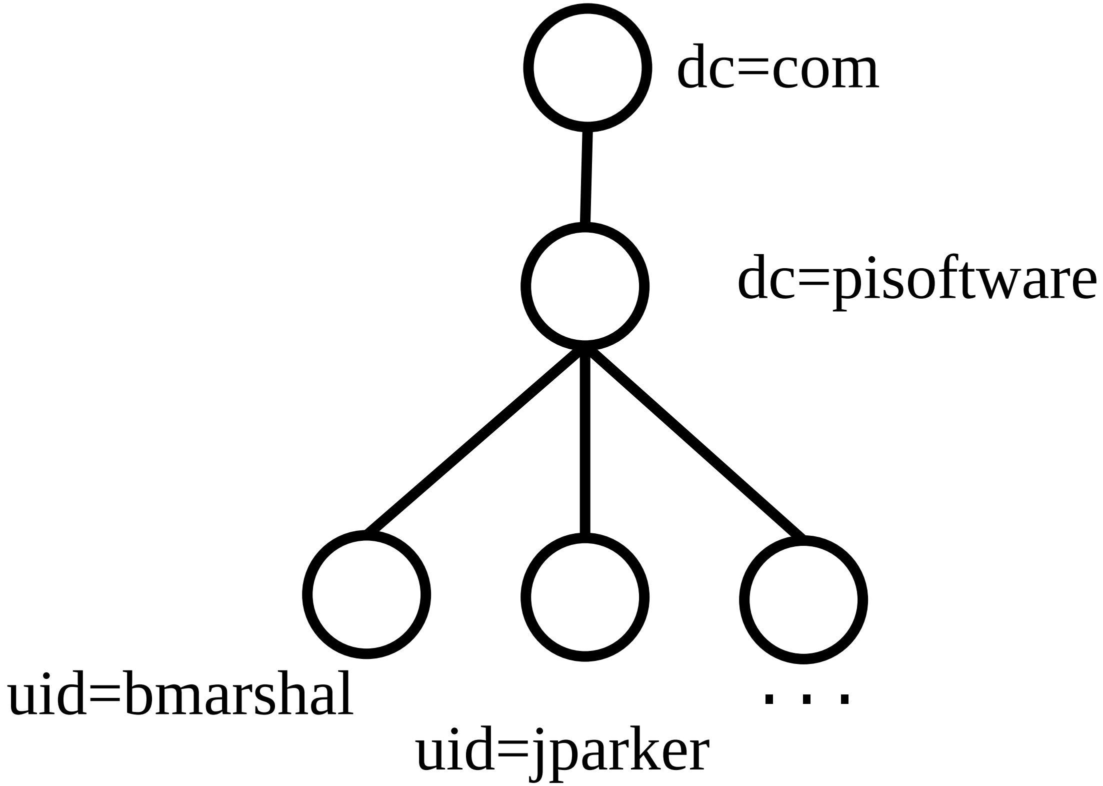
Namespaces

- Hierarchical data structure
  - Entries are in a tree-like structure called Directory Information Tree (DIT)
- Consistent view of data - uniform namespace
  - Answers request
  - Refer to server with answer

Namespaces - Hierarchal



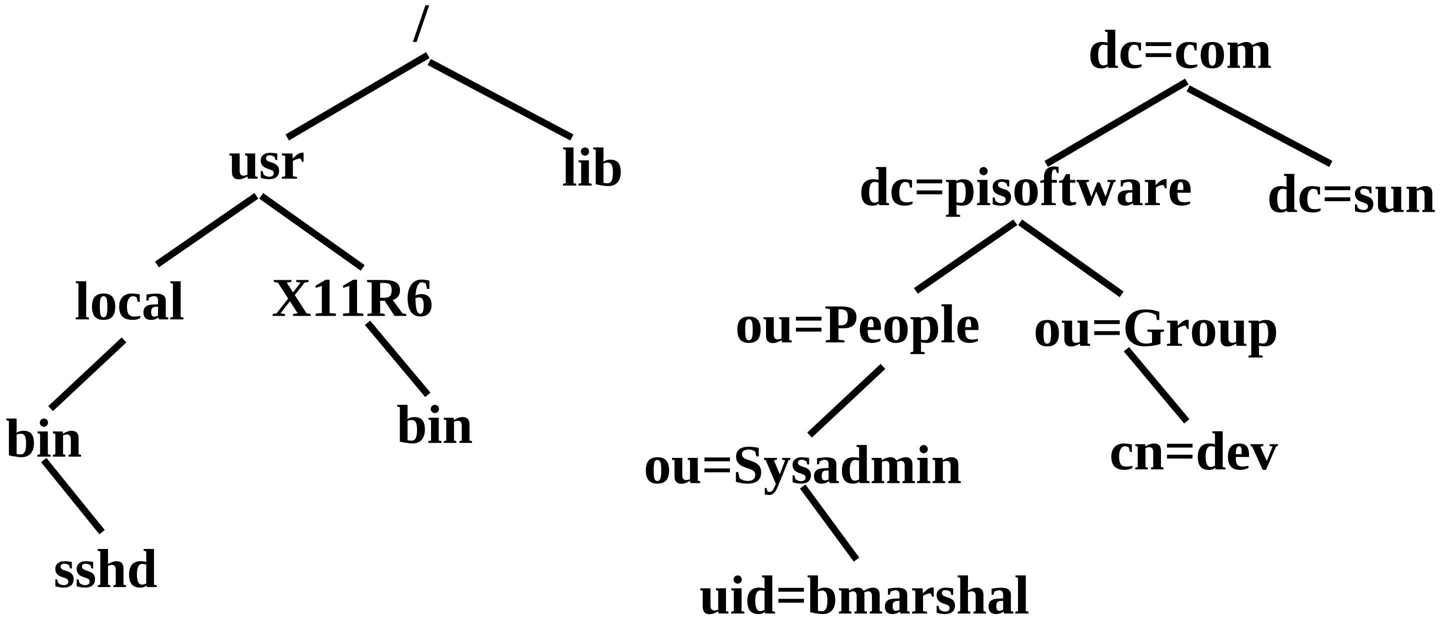
Namespaces - Flat



Namespaces cont

- Directory tree is similar to unix file system
  - No root entry in ldap
  - Each entry in ldap can both contain data and be a container
    - In unix, an entry is either a file or a directory - not both
  - LDAP distinguished names are read from bottom to top, unix file systems from top to bottom

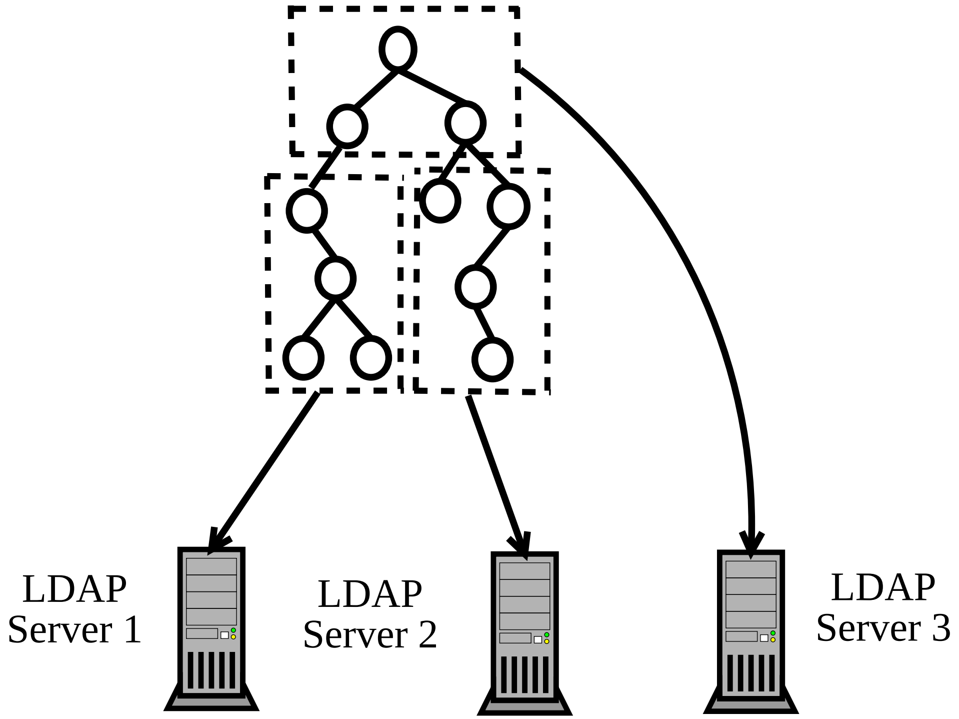
Namespaces cont



Namespace Design

- Designing a namespace is Hard
- Requires indepth knowledge of what the directory will be used for
- Hard to reorganise once data is put in - requires downtime, etc
- Needs to support applications that want to use it - be aware of existing standards
- Need to partition up data for access control and replication
- Try not to break out into different departments - what happens when person moves?
- Don’t go overboard - too much hierachy can get confusing

Global View



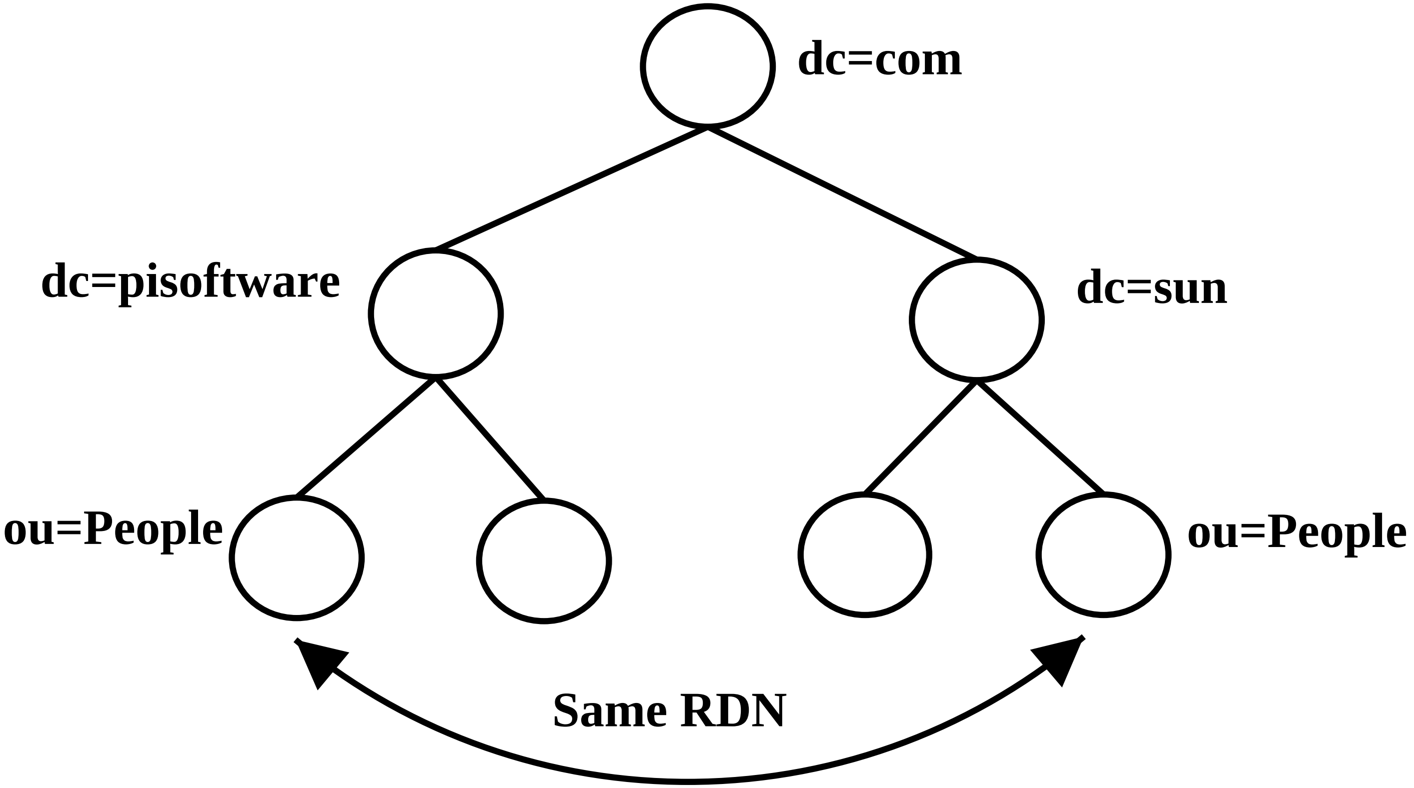
Note each server must contain a subtree

Distinguished Names

- Built up by starting at the bottom, and connecting each level together with commas
- Contain two parts
  - Left most part is called relative distinguished name
  - Remainder is base distinguished name
- Eg: uid=bmarshal,ou=People,dc=pisoftware,dc=com
  - RDN is uid=bmarshal
  - Base DN is ou=People,dc=pisoftware,dc=com

Distinguished Names cont

- In each base DN, each RDN is unique
  - This ensures no two entries have the same DN



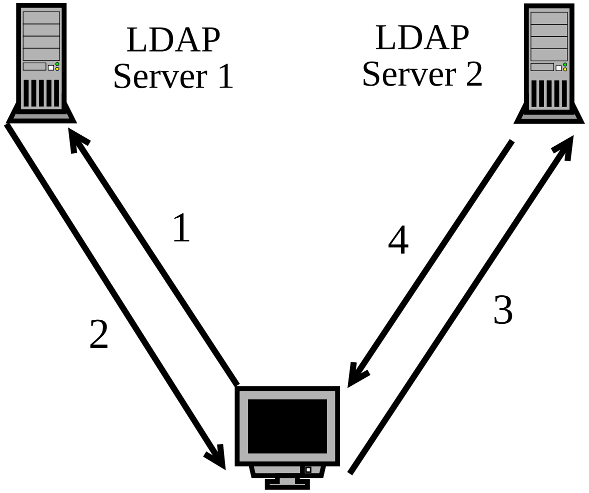
Distinguished Names cont

- Use DNS name to generate base DN
- See RFC2377 for more details - "Naming Plan for Internet Directory-Enabled Applications"
- example.com gives dc=example,dc=com
- Already globally unique
- Already registered
- Can trace back to who owns it easily

LDAP Entry

- Entries are composed of attributes
- Attributes consist of types with multiple values
- Type describes what the information is
- Value is the actual information in text format
- Attributes have a syntax which specifies what type of data - see Schema later on

Referrals



1. Client requests information
2. Server 1 returns referral to server 2
3. Client resends request to server 2
4. Server 2 returns information to client

Aliases

- Aliases are used to point one LDAP entry to another
- Allows you to have structures that aren’t hierarchal
- Similar in sense to using a symlink in unix
- Not all LDAP servers support aliases - big performance hit

Aliases cont

- Created by:
  - Entry with object class of alias
  - Attribute named aliasedObjectName that points to DN of the alias
- Can use either referrals or putting a LDAP url in an entry

Schema

- Set of rules that describes what kind of data is stored
- Helps maintain consistency and quality of data
- Reduces duplication of data
- Ensures applications have consistent interface to the data
- Object class attribute determines schema rules the entry must follow

Schema cont

- Schema contains the following:
  - Required attributes
  - Allowed attributes
  - How to compare attributes
  - Limit what the attributes can store - ie, restrict to integer etc
  - Restrict what information is stored - ie, stops duplication etc

Objectclass

- Used to group information
- Provides the following rules:
  - Required attributes
  - Allowed attributes
  - Easy way to retrieve groups of information
- Entries can have multiple object classes
  - Required and allowed attributes are the union of the attributes of each of the classes

Objectclass inheritance

- Object classes can be derived from others
- Extends attributes of other objectclass
- No multiple inheritance
- Can’t override any of the rules
- Special class called top - all classes extend
  - Only required attribute is objectclass
  - Ensures all entries have a objectclass

Attributes

Attributes have:

- Name - unique identifier, not case sensitive
- Object identifier (OID) - sequence of integers separated by dots
- Attribute syntax:
    - Data attributes can store - eg integer, string etc
    - How comparisons are made
- If multivalued or single valued

Attributes

See RFC2256

uid	User id
cn	Common Name
sn	Surname
l	Location
ou	Organisational Unit
o	Organisation
dc	Domain Component
st	State
c	Country

LDIF

- LDAP Data Interchange Format
  - Represents LDAP entries in text
  - Human readable format
  - Allows easy modification of data
  - Useful for doing bulk changes
    - dump db, run a script over, import back
  - Can use templates for additions
  - Good for backups and transferring data to another system
- Utilities to convert from database to ldif and back
  - ldbmcat & slapcat: ldbm database to ldif
  - ldif2ldbm & slapadd: ldif to ldbm database

LDIF Example

```
dn: uid=bmarshal,ou=People,
         dc=pisoftware,dc=com
uid: bmarshal
cn: Brad Marshall
objectclass: account
objectclass: posixAccount
objectclass: top
loginshell: /bin/bash
uidnumber: 500
gidnumber: 120
homedirectory: /mnt/home/bmarshal
gecos: Brad Marshall,,,,
userpassword: {crypt}KDnOoUYN7Neac
```

Search Filters

- Criteria for attributes that must be fulfilled for entry to be returned
- Base dn = base object entry search is relative to
- Prefix notation
- Standards
  - RFC 1960: LDAP String Representation of Search Filters
  - RFC 2254: LDAPv3 Search Filters

Search Filters Operators

&	and
|	or
!	not
~=	approx equal
>=	greater than or equal
<=	less than or equal
*	any

Search Filters Examples

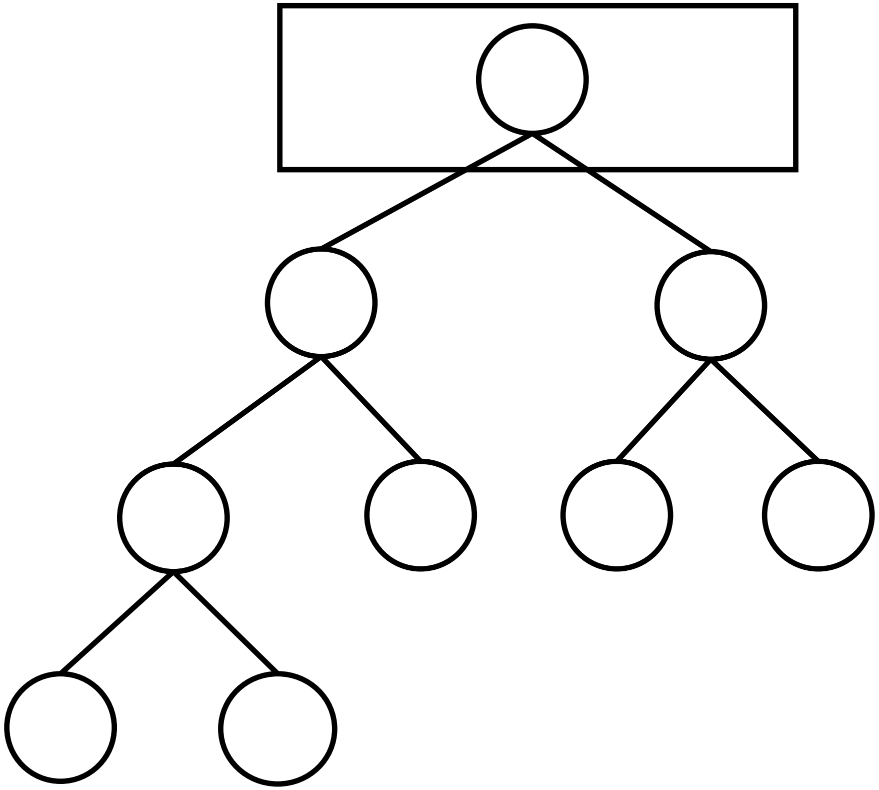
- (objectclass=posixAccount)
- (cn=Mickey M*)
- (|(uid=fred)(uid=bill))
- (&(|(uid=jack)(uid=jill))(objectclass=posixAccount))

Search Scope

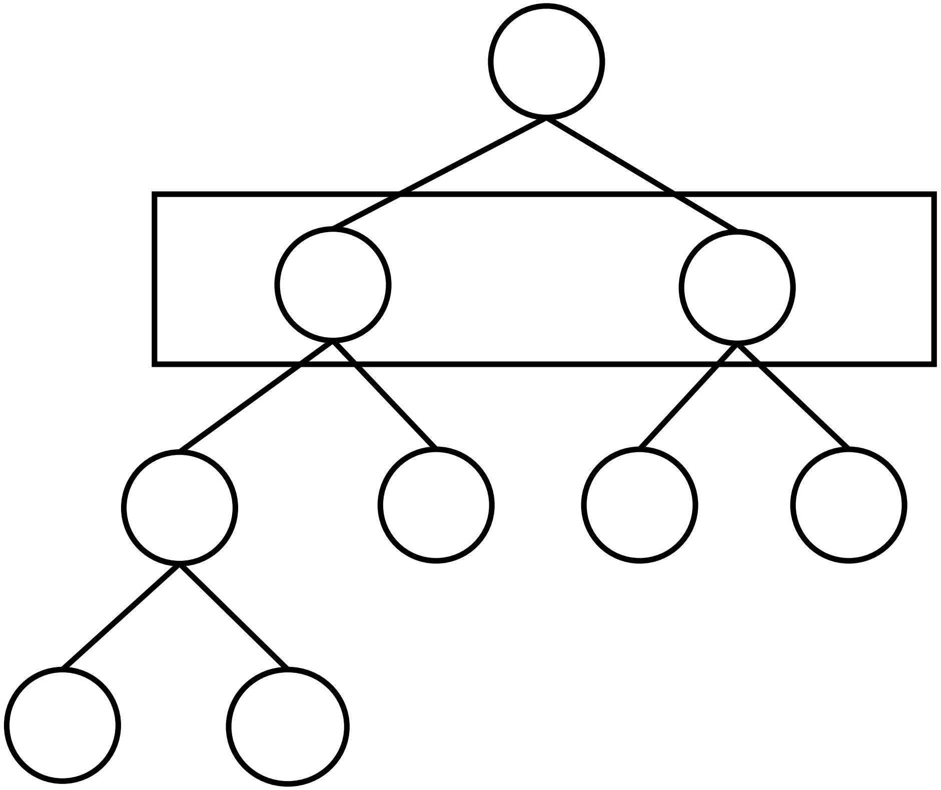
3 types of scope:

base	limits to just the base object
onelevel	limits to just the immediate children
sub	search the entire subtree from base down

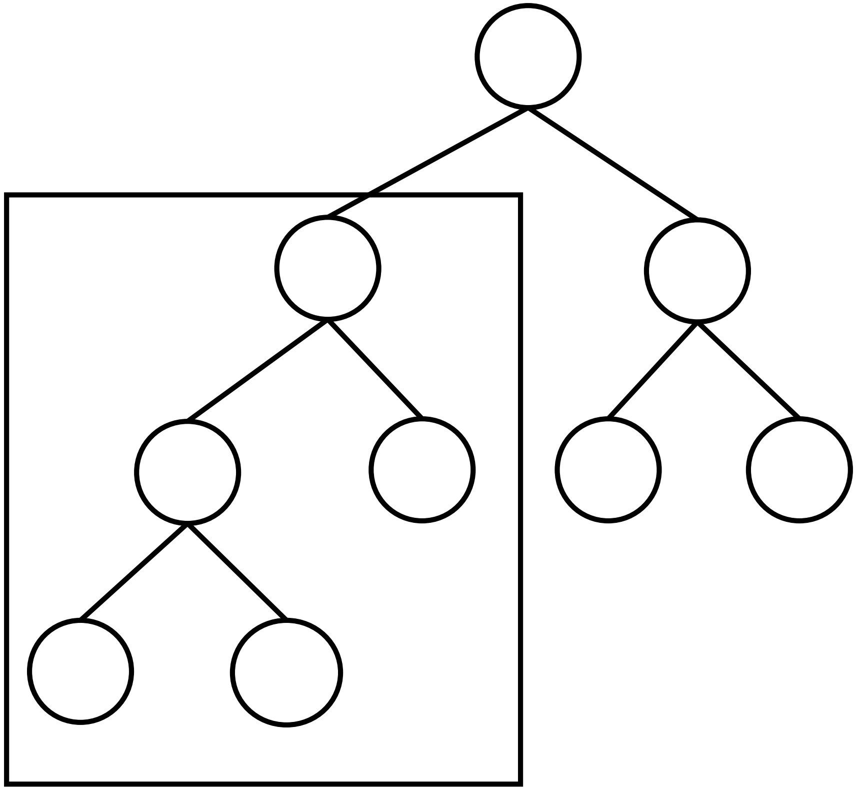
Base Scope



One Level Scope



Subtree Scope



LDAP URLs

Definition taken from RFC1959

```
<ldapurl> ::= "ldap://" [ <hostport> ]
         "/" <dn> [ "?" <attributes>
         [ "?" <scope> "?" <filter> ] ]
<hostport> ::= <hostname>
           [ ":" <portnumber> ]
<dn> ::= a string as defined in RFC 1485
<attributes> ::= NULL | <attributelist>
<attributelist> ::= <attributetype>
                | <attributetype>
                [ "," <attributelist> ]
<attributetype> ::= a string as defined
                      in RFC 1777
<scope> ::= "base" | "one" | "sub"
<filter> ::= a string as defined in RFC 1558
```

LDAP URLs

DN Distinguished name

Attribute list List of attributes you want returned

Scope

base	base object search
one	one level search
sub	subtree search

Filter Standard LDAP search filter

LDAP URL examples

- ldap://foo.bar.com/dc=bar,dc=com
- ldap://argle.bargle.com/dc=bar, dc=com??sub?uid=barney
- ldap://ldap.bedrock.com/dc=bar, dc=com?cn?sub?uid=barney

LDAPv3

- Internationalisation - using UTF-8
- Referrals
- Security
- Extensibility
- Feature and schema discovery
  - LDAPv3 servers have a directory entry called root DSE (Directory Server Entry)
  - Contains: protocol supported, schemas, other useful info

LDAP Servers

- Slapd
  - University of Michigan
  - Openldap
- Netscape Directory Server
- Microsoft Active Directory (AD)
- Microsoft Exchange (interface only)
- Novell Directory Services (NDS)
- Lotus Domino (interface only)
- Sun Directory Services (SDS)
- Lucent’s Internet Directory Server (IDS)

Openldap

- Based on UMich ldap server
- Available from http://www.openldap.org/
- Versions:
  - Historic: 1.2.13 - implements LDAPv2
  - Stable: 2.0.25 - implements LDAPv3
  - Release: 2.1.12 - implements LDAPv3 and other features

Openldap 2.1 features

OpenLDAP 2.1 was released June 2002 Functional enhancements and improved stability (from web site):

- Transaction oriented database backend
- Improved Unicode/DN Handling
- SASL authentication/authorization mapping
- SASL in-directory storage of authentication secrets
- Enhanced administrative limits / access controls
- Enhanced system schema checking
- LDAP C++ API
- Updated LDAP C & TCL APIs

Openldap 2.1 features cont

- LDAPv3 extensions:
  - Enhanced Language Tag/Range option support
  - objectClass-based attribute lists
  - LDAP Who ami I? Extended Operation
  - LDAP no-op Control
  - Matched Values Control
  - Misc LDAP Feature Extensions
- Meta Backend
- Monitor Backend
- Virtual Context "glue" Backend

Openldap LDAPv3 Support

OpenLDAP LDAPv3 support includes:

- SASL Bind (RFC 2829)
- Start TLS (RFC 2830)
- LDIFv1 (RFC 2849)

LDAPv3 supported extensions include:

- Language Tag Options (RFC 2596)
- Language Range Options
- DNS-based service location (RFC 2247 & RFC 3088)
- Password Modify (RFC 3062)
- Named Referrals / ManageDSAit (I-D namedref)
- Matched Values Control
- All Operational Attributes ("+")

Openldap LDAPv3 Not Supports

Does not support:

- DIT Content Rules
- DIT Structure Rules
- Name Forms
- Schema updates (using LDAP)
- Subtree rename

LDAPv3 unsupported extensions include:

- Dynamic Directory Services (RFC 2589)
- Operational Signatures (RFC 2649)
- Simple Paged Result Control (RFC 2696)
- Server Side Sorting of Search Results (RFC 2891)

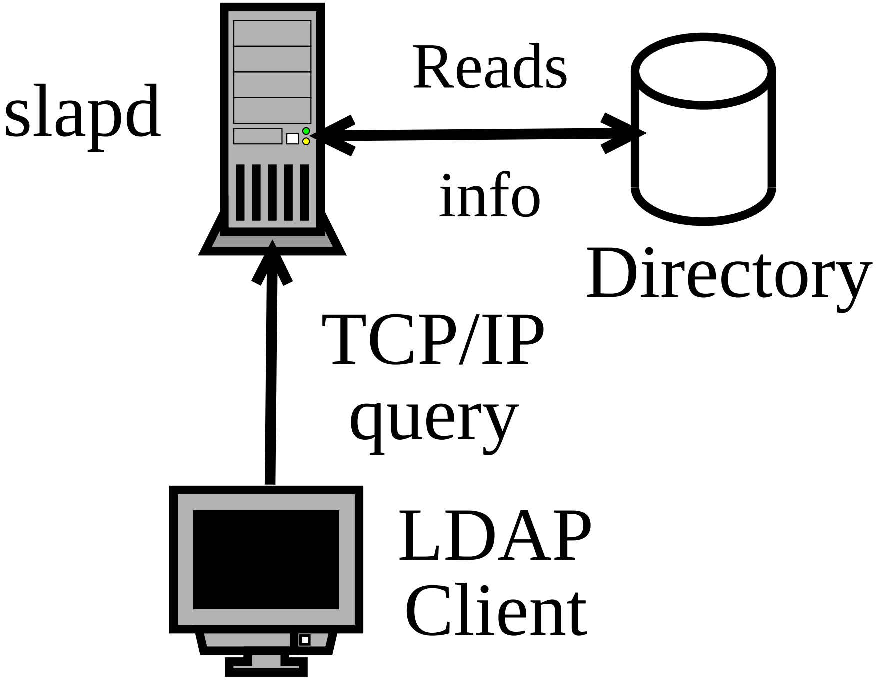
Openldap Platforms

- Runs on:
  - FreeBSD
  - Linux
  - NetBSD
  - OpenBSD
  - Most commercial UNIX systems
- Ports in progress:
  - BeOS
  - MacOS
  - Microsoft Windows NT/2000

LDAP slapd architecture

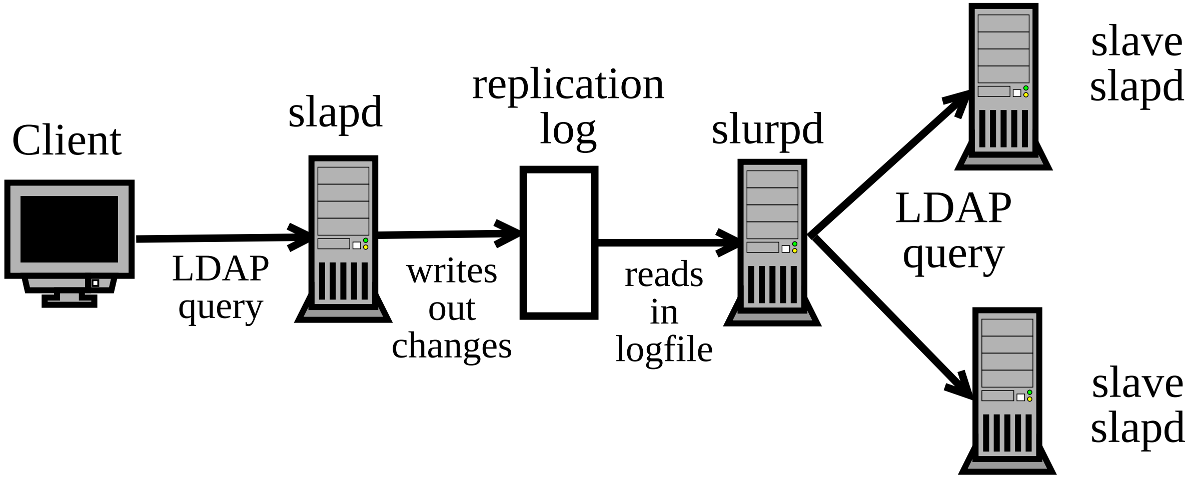
- LDAP daemon called slapd
  - Choice of databases
    - LDBM - high performance disk based db
    - SHELL - db interface to unix commands
    - PASSWORD - simple password file db
    - SQL - mapping sql to ldap (in OpenLDAP 2.x)
  - Multiple database instances
  - Access control
  - Threaded
  - Replication

LDAP slapd architecture



LDAP slurpd architecture

- Replication daemon called slurpd
  - Frees slapd from worrying about hosts being down etc
  - Communicates with slapd through text file



Slurpd Replication Log File

Slapd writes out a replication log file containing:

- Replication host
- Timestamp
- DN of entry being modified
- List of changes to make

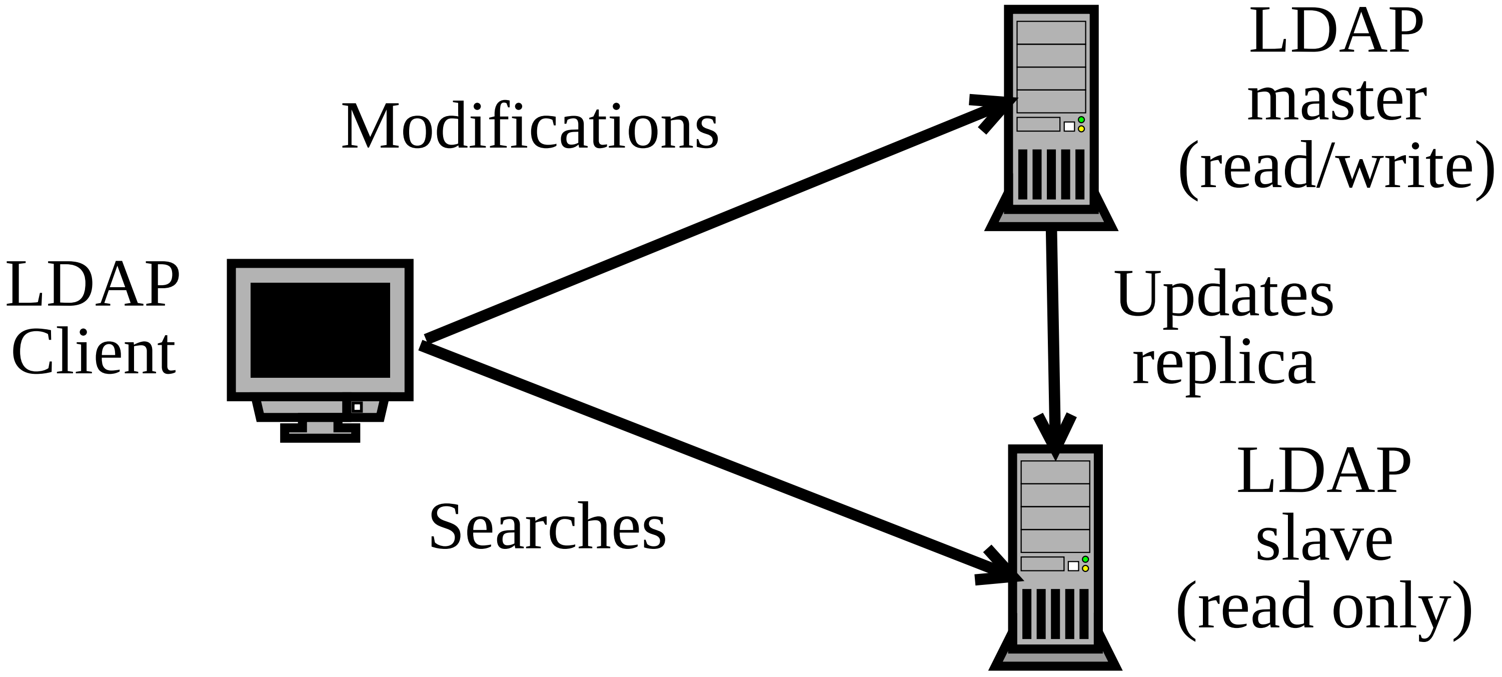
Slurpd Replication Log File Example

```
replica: slave.pisoftware.com:389
time: 93491423
dn: uid=bmarshal,ou=People,
           dc=pisoftware,dc=com
changetype: modify
replace: multiLineDescription
description: There once was a sysadmin...
-
replace: modifiersName
modifiersName: uid=bmarshal,ou=People,
           dc=pisoftware,dc=com
-
replace: modifyTimestamp
modifyTimestamp: 20010606122901Z
-
```

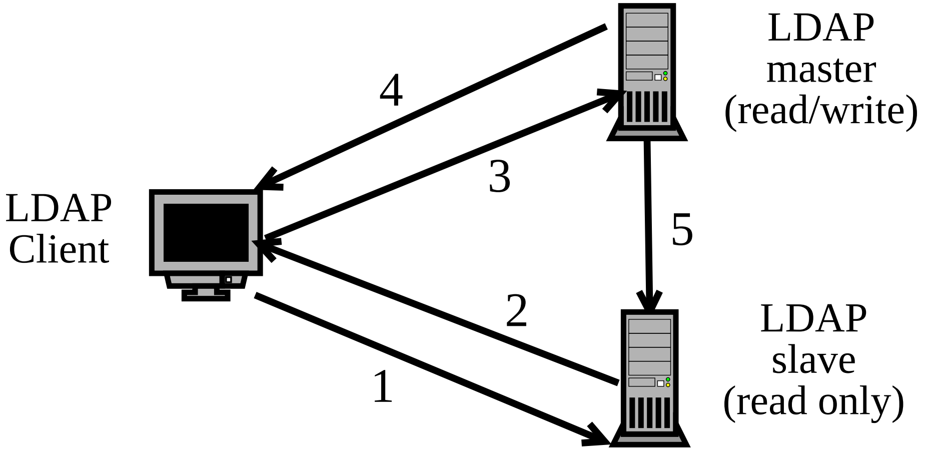
Replication

- Increases:
  - Reliability - if one copy of the directory is down
  - Availability - more likely to find an available server
  - Performance - can use a server closer to you
  - Speed - can take more queries as replicas are added
- Temporary inconsistencies are ok
- Having replicas close to clients is important - network going down is same as server going down
- Removes single point of failure

Replication Options - Mods to Master

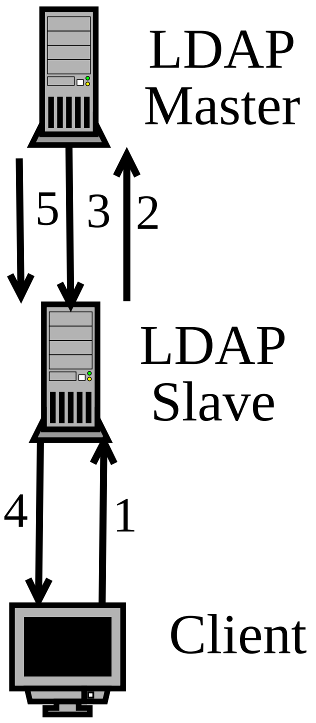


Replication Options - Referrals



1. Client sends modification to replica
2. Replica returns referral to master
3. Client resubmits modification to master
4. Master returns results to client
5. Master updates replica with change

Replication Options - Chaining



1. Client sends modification to replica
2. Replica forwards request to master
3. Master returns result to replica
4. Replica forwards result to client
5. Master updates replica

Slapd.conf Example

```
#
# See slapd.conf(5) for details
#   on configuration options.
# This file should NOT be world readable.
#
include         /etc/openldap/slapd.at.conf
include         /etc/openldap/slapd.oc.conf
schemacheck     off

pidfile         /var/run/slapd.pid
argsfile        /var/run/slapd.args

defaultaccess read
```

Slapd.conf Example cont

```
access to attr=userpassword
   by self write
   by * read

access to *
   by self write
   by dn=".+" read
   by * read
```

Slapd.conf Example cont

```
######################################
# ldbm database definitions
######################################
database  ldbm
suffix    "dc=pisoftware, dc=com"
rootdn    "cn=Manager,dc=pisoftware,dc=com"
rootpw    {crypt}lAn4J@KmNp9
replica host=replica.bne.pisoftware.com:389
    binddn="cn=Manager,dc=pisoftware,dc=com"
    bindmethod=simple credentials=secret
    replogfile /path/to/replication.log
# cleartext passwords, especially for
# the rootdn, should be avoid.  See
# slapd.conf(5) for details.
directory       /var/lib/openldap/
```

ACLs

Can restrict by:

- Distinguished Name
- Filter that matches some attributes
- Attributes

ACLs cont

Can restrict with:

- Anonymous users
- Authenticated users
- Self - ie, user who owns the entry
- Distinguished name
- IP address or DNS entry

ACLs cont

Access control priority:

- Local database
- Global rules
- Runs thru in order the rules appear in the config file
- First matching rule is used

ACL examples

```
access to attribute=userpassword
    by dn="cn=Manager,dc=pisoftware,
         dc=com" write
    by self write
    by * read

access to dn="(.*,)?dc=pisoftware,dc=com"
         attr=homePhone
    by self write
    by dn="(.*,)?dc=pisoftware,dc=com" search
    by domain=.*\.pisoftware\.com read
    by anonymous auth
```

Slapd and TLS

To generate a certificate:

```
$ openssl req -newkey rsa:1024 -keyout
    server.pem -nodes -x509 -days 365
    -out server.pem
```

Assuming that the slapd.conf file is properly configured, the following additions are required:

```
TLSCertificateFile    /usr/lib/ssl/misc/server.
TLSCertificateKeyFile /usr/lib/ssl/misc/server.
TLSCACertificateFile  /usr/lib/ssl/misc/server.
replica  host=hostname:389
   tls=yes
   binddn="normal bind parameters"
   bindmethod=simple
   credentials=password
```

Slapd and TLS cont

Configure your slapd init scripts to run with the following options:

```
  slapd -h "ldap:/// ldaps:///"
```

To confirm that it is listening, run the following:

```
$ sudo netstat --inet --l -p | grep slapd
tcp  0   0 *:ldap   *:*  LISTEN  17706/slapd
tcp  0   0 *:ldaps  *:*  LISTEN  17706/slapd
```

To check the certificate:

```
$ openssl s_client -connect localhost:636 \
               -showcerts
```

Referral Config

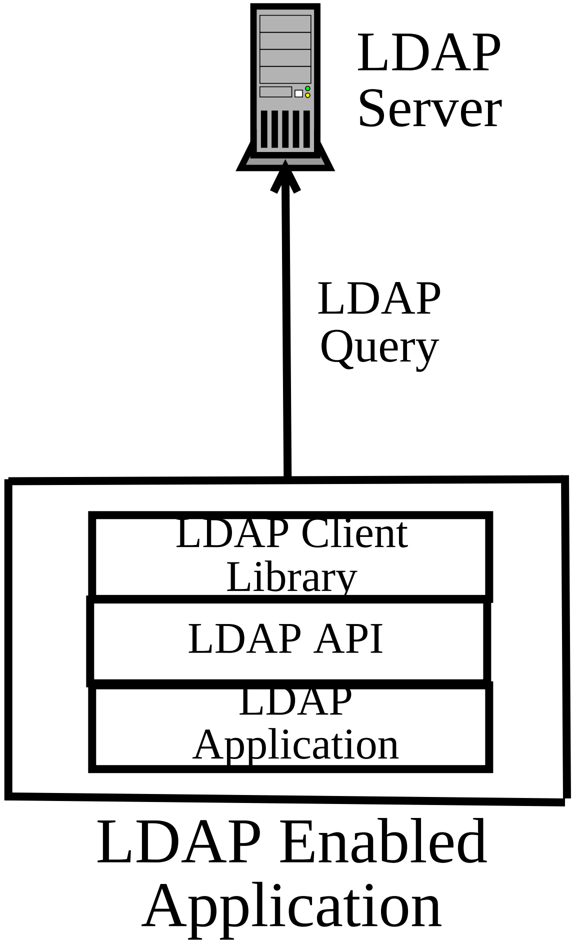
To delegate a subtree to another server, use the ref attribute to specify the ldap url to follow.

```
dn: dc=subtree, dc=example, dc=net
objectClass: referral
objectClass: extensibleObject
dc: subtree
ref: ldap://b.example.net/dc=subtree,
               dc=example,dc=net/
```

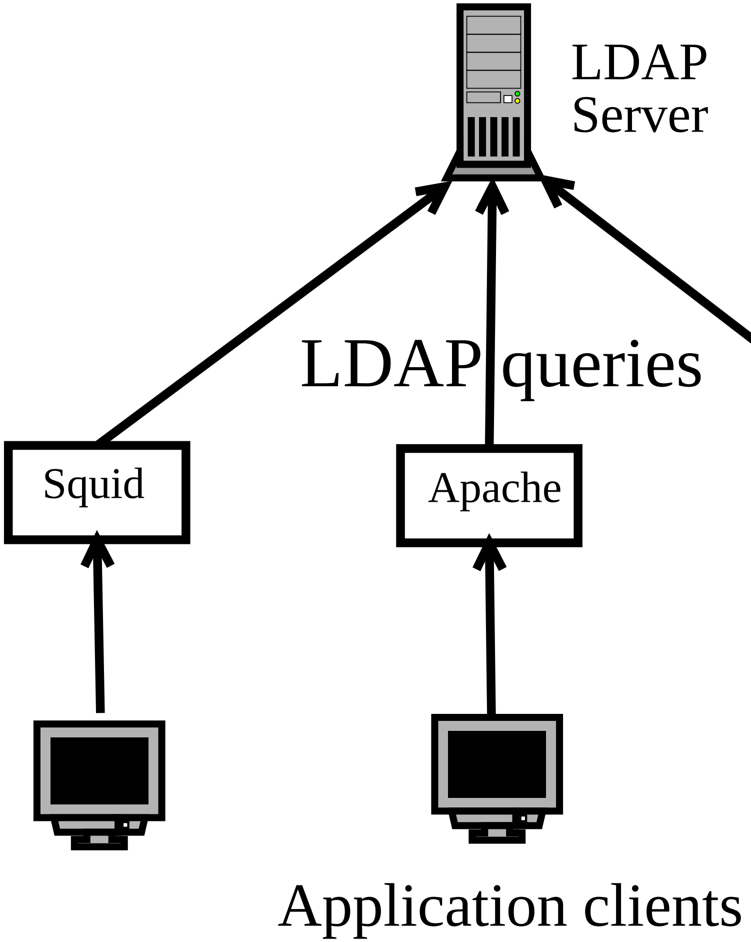
To specify another ldap server to go to if the current server can’t answer, use the referral directive.

```
referral                ldap://root.openldap.org/
```

Using LDAP in Applications



Using Multiple Applications



Linux Authentication

- Consists of two main parts
  - PAM - Pluggable Authentication Modules
  - NSS - Name Service Switch

PAM

- Allows sysadmin to choose how applications authenticate
- Consists of dynamically loadable object files - see dlopen(3)
- Modules stored in /lib/security/pam_modulename.so
- Seperates development of applications from developing of authentication schemes
- Allows changing of authentication schema without modifying applications

PAM cont

- Remember in early days when Linux changed to shadow passwords
    - Used to have hard coded authentication method - /etc/passwd
    - Needed to recompile any programs that authenticated
    - Very frustrating for most users
- Can have different apps auth against different databases
- Can also do restrictions on various things - eg login time, resources used

PAM Config files

- Each application has a (hard coded) service type
- Config files can be kept in:
  - /etc/pam.conf
  - /etc/pam.d, with a seperate file per service type
- Format for /etc/pam.conf:

  ```
  service module-type control-flag
          module-path arguments
  ```

- Format for /etc/pam.d/service:

  ```
  module-type control-flag
          module-path arguments
  ```

- Can have multiple entries for each module-type - known as stacking modules

PAM Module Types

- Authentication
  - Establishes the users is who they say they are by asking for password (or some other kind of authencation token)
  - Can grant other privileges (such as group membership) via credential granting
- Account
  - Performs non-authentication based account management
  - Restrict access based on time of day, see if accounts have expired, check user and process limits etc

PAM Module Types cont

- Session
  - Deals with things that have to be done before and after giving a user access
  - Displaying motd, mounting directories, showing if a user has mail, last login, updating login histories etc
- Password
  - Updating users authentication details - ie, changing passwords

Name Service Switch (NSS)

- Provides more information than just username and password
- Originally done by changing the C library
- Now done using dynamic loadable modules
- Follows design from Sun Microsystems
- Can get this information from places such as LDAP
- Modules stored in /lib/libnss_name.so
- Configuration file is /etc/nsswitch.conf

System Authentication

- Uses RFC2307
- Provides a mapping from TCP/IP and unix entities into LDAP
- Gives a centrally maintained db of users
- Can create own tools to maintain, or use ready made ones
- Could dump out to locally files - not ideal
- Use PADL's nss_ldap and pam_ldap tools

System Authentication Migration

Used PADLs MigrationTools

Script	**Migrates**
migrate_fstab.pl	/etc/fstab
migrate_group.pl	/etc/group
migrate_hosts.pl	/etc/hosts
migrate_networks.pl	/etc/networks
migrate_passwd.pl	/etc/passwd
migrate_protocols.pl	/etc/protocols
migrate_rpc.pl	/etc/rpc
migrate_services.pl	/etc/services

System Authentication Migration cont

These scripts are called on the appropriate file in /etc in the following manner:

```
# ./migrate_passwd.pl /etc/passwd
                ./passwd.ldif
```

The migration tools also provide scripts to automatically migrate all configuration to LDAP, using migrate_all_online,offline.sh. See the README distributed with the package for more details.

Example user LDIF

```
dn: uid=bmarshal,ou=People,
        dc=pisoftware,dc=com
uid: bmarshal
cn: Brad Marshall
objectclass: account
objectclass: posixAccount
objectclass: top
loginshell: /bin/bash
uidnumber: 500
gidnumber: 120
homedirectory: /mnt/home/bmarshal
gecos: Brad Marshall,,,,
userpassword: {crypt}aknbKIfeaxs
```

Example group LDIF

```
dn: cn=sysadmin,ou=Group,
       dc=pisoftware,dc=com
objectclass: posixGroup
objectclass: top
cn: sysadmin
gidnumber: 160
memberuid: bmarshal
memberuid: dwood
memberuid: jparker
```

Server Configuration

/etc/openldap/slapd.conf

```
include         /etc/openldap/slapd.at.conf
include         /etc/openldap/slapd.oc.conf
schemacheck     off

pidfile         /var/run/slapd.pid
argsfile        /var/run/slapd.args

defaultaccess read
```

Server Configuration cont

```
access to attr=userpassword
   by self write
   by * read

access to *
   by self write
   by dn=".+" read
   by * read
```

Server Configuration cont

```
############################
# ldbm database definitions
############################

database    ldbm
suffix      "dc=pisoftware, dc=com"
rootdn      "cn=Manager, dc=pisoftware, dc=com"
rootpw      {crypt}lAn4J@KmNp9
replica host=replica.pisoftware.com:389
   binddn="cn=Manager,dc=pisoftware,dc=com"
   bindmethod=simple credentials=secret
   replogfile /var/lib/openldap/replication.log
# cleartext passwords, especially for the
# rootdn, should be avoid.  See slapd.conf(5)
# for details.
directory       /var/lib/openldap/
```

PAM Configuration

/etc/pam_ldap.conf - See actual file for more details

```
# Your LDAP server.
# Must be resolvable without using LDAP.
host 127.0.0.1

# The distinguished name of the search base.
base dc=pisoftware,dc=com

# The LDAP version to use (defaults to 3
# if supported by client library)
ldap_version 3

# The port.
# Optional: default is 389.
#port 389
```

PAM Configuration cont

```
# Hash password locally; required for
# University of Michigan LDAP server,
# and works with Netscape Directory
# Server if you’re using the UNIX-Crypt
# hash mechanism and not using the NT
# Synchronization service. This is the
# default.
pam_password crypt

# Use nds for Novell Directory
# Use ad for Active Directory
# Use exop for Openldap password
# change extended operations
```

pam.d configuration

/etc/pam.d/ssh

```
#%PAM-1.0
auth     required   pam_nologin.so
auth     sufficient pam_ldap.so
auth     required   pam_unix.so try_first_pass
auth     required   pam_env.so # [1]

account  sufficient pam_ldap.so
account  required   pam_unix.so
```

pam.d configuration cont

```
session  sufficient pam_ldap.so
session  required   pam_unix.so
session  optional   pam_lastlog.so # [1]
session  optional   pam_motd.so # [1]
session  optional   pam_mail.so standard noenv
session  required   pam_limits.so

password sufficient pam_ldap.so
password required   pam_unix.so try_first_pass
```

NSS configuration

/etc/libnss_ldap.conf - see local file for more details

```
# Your LDAP server.
# Must be resolvable without using LDAP.
host 127.0.0.1

# The distinguished name of the search base.
base dc=pisoftware,dc=com

# The LDAP version to use (defaults to 2)
ldap_version 3

# The port.
# Optional: default is 389.
#port 389
```

NSS configuration - nsswitch.conf

/etc/nsswitch.conf

```
passwd:         compat ldap
group:          compat ldap
shadow:         compat ldap
```

Note that the order of the nss sources will modify which source is canonical. That is, if you list ldap first, it will be checked first.

System Auth - Usage

- ldappasswd

```
ldappasswd -W -D ’uid=bmarshal,ou=People,
        dc=pisoftware,dc=com’ ’uid=bmarshal’
```

- ldapsearch

```
ldapsearch -L ’uid=*’
ldapsearch -L ’objectclass=posixGroup’
ldapsearch -L ’objectclass=posixAccount’
ldapsearch -D ’uid=bmarshal,ou=People,
     dc=pisoftware,dc=com’ -W -L
     ’uid=bmarshal’
```

- ldapmodify (where bmarshal.ldif is ldapsearch -L ’uid=bmarshal’)

```
ldapmodify -W -r -D "cn=Manager,
    c=pisoftware,dc=com" < bmarshal.ldif
```

Sendmail and LDAP

- Sendmail traditionally uses flat files stored on the server
- Reduces need to manually sync data across multiple servers
- Allows cross-platform, standardised, centralised repository of user data
- Can use data in multiple applications - internal email directory etc

Sendmail and LDAP compiling

To check that sendmail has LDAP support, run:

```
sendmail -d0.1 -bv root
```

The output should contain:

```
Compiled with: LDAPMAP
```

To compile sendmail with LDAP support:

```
APPENDDEF(‘confMAPDEF’, ‘-DLDAPMAP’)
APPENDDEF(‘confINCDIRS’,
    ‘-I/path/to/openldap-1.2.11/include’)
APPENDDEF(‘confLIBSDIRS’,
    ‘-L/path/to/openldap-1.2.11/libraries’)
APPENDDEF(‘confLIBS’, ‘-lldap -llber’)
```

Now you can rebuild as normal.

Sendmail and LDAP config

The base config that you need to add to sendmail.mc is:

```
LDAPROUTE_DOMAIN('example.com')dnl
define(confLDAP_DEFAULT_SPEC,
         -h ldap.example.com
         -b dc=example.com)
```

To define a group of hosts, use:

```
define(‘confLDAP_CLUSTER’, ‘Servers’)
```

To enable LDAP aliases:

```
define(‘ALIAS_FILE’, ‘ldap:’)
```

To enable other lookups, use:

```
FEATURE(‘access_db’, ‘LDAP’)
FEATURE(‘virtusertable’, ‘LDAP’)
```

To enable classes:

```
RELAY_DOMAIN_FILE(‘@LDAP’)
```

Sendmail LDAP Map Values

FEATURE()	**sendmailMTAMapName**
access_db	access
authinfo	authinfo
bitdomain	bitdomain
domaintable	domain
genericstable	generics
mailertable	mailer
uucpdomain	uucpdomain
virtusertable	virtuser

Sendmail Alias LDIF example

```
dn: sendmailMTAKey=postmaster,
      dc=pisoftware, dc=com
objectClass: sendmailMTA
objectClass: sendmailMTAAlias
objectClass: sendmailMTAAliasObject
sendmailMTAAliasGrouping: aliases
sendmailMTACluster: Servers
sendmailMTAKey: postmaster
sendmailMTAAliasValue: bmarshal
```

Sendmail Mailertable LDIF example

Group LDIF:

```
dn: sendmailMTAMapName=mailer,
      dc=pisoftware, dc=com
objectClass: sendmailMTA
objectClass: sendmailMTAMap
sendmailMTACluster: Servers
sendmailMTAMapName: mailer
```

Sendmail Mailertable LDIF example cont

Entry LDIF:

```
dn: sendmailMTAKey=example.com,
       sendmailMTAMapName=mailer,
       dc=pisoftware, dc=com
objectClass: sendmailMTA
objectClass: sendmailMTAMap
objectClass: sendmailMTAMapObject
sendmailMTAMapName: mailer
sendmailMTACluster: Servers
sendmailMTAKey: example.com
sendmailMTAMapValue: relay:[smtp.example.com]
```

Sendmail LDAP Classes Values

Command	sendmailMTAClassN
CANONIFY_DOMAIN_FILE()	Canonify
EXPOSED_USER_FILE()	E
GENERICS_DOMAIN_FILE()	G
LDAPROUTE_DOMAIN_FILE()	LDAPRoute
LDAPROUTE_EQUIVALENT_FILE()	LDAPRouteEquiv
LOCAL_USER_FILE()	L
MASQUERADE_DOMAIN_FILE()	M
MASQUERADE_EXCEPTION_FILE()	N
RELAY_DOMAIN_FILE()	R
VIRTUSER_DOMAIN_FILE()	VirtHost

Sendmail Classes LDIF example

```
dn: sendmailMTAClassName=R,
    dc=pisoftware, dc=com
objectClass: sendmailMTA
objectClass: sendmailMTAClass
sendmailMTACluster: Servers
sendmailMTAClassName: R
sendmailMTAClassValue: pisoftware.com
sendmailMTAClassValue: example.com
sendmailMTAClassValue: 10.56.23
```

Apache and LDAP

- Allows you to restrict access to a webpage with data from LDAP
- Download mod_auth_ldap.tar.gz from http://www.muquit.com/muquit/software/mod_auth_ldap/mod_auth_ldap.html
- Install either as a DSO or by compiling in - see webpage for more details

Apache and LDAP cont

- Add the following to httpd.conf:

```
<Directory "/var/www/foo">
Options Indexes FollowSymLinks
AllowOverride None
order allow,deny
allow from all
AuthName "RCS Staff only"
AuthType Basic
```

Apache and LDAP cont

```
LDAP_Server ldap.server.com
LDAP_Port 389
Base_DN "dc=server,dc=com"
UID_Attr uid
#require valid-user
require user foo bar doe
#require roomnumber "C119 Center Building"
#require group
#  cn=sysadmin,ou=Group,dc=server,dc=com
</Directory>
```

Squid and LDAP

- Allows you to restrict access to Squid via ldap
- Add the following to the configure line:
  –enable-auth-modules=LDAP
- See documentation at http://orca.cisti.nrc.ca/ gnewton/ opensource/squid_ldap_auth/
- Add the following to squid.conf:

```
authenticate_program /path/to/squid_ldap_auth
     -b dc=yourdomain,dc=com ldap.yourdomain
acl ldapauth proxy_auth REQUIRED
#acl ldapauth proxy_auth bmarshal dwood pag
```

- Restart squid

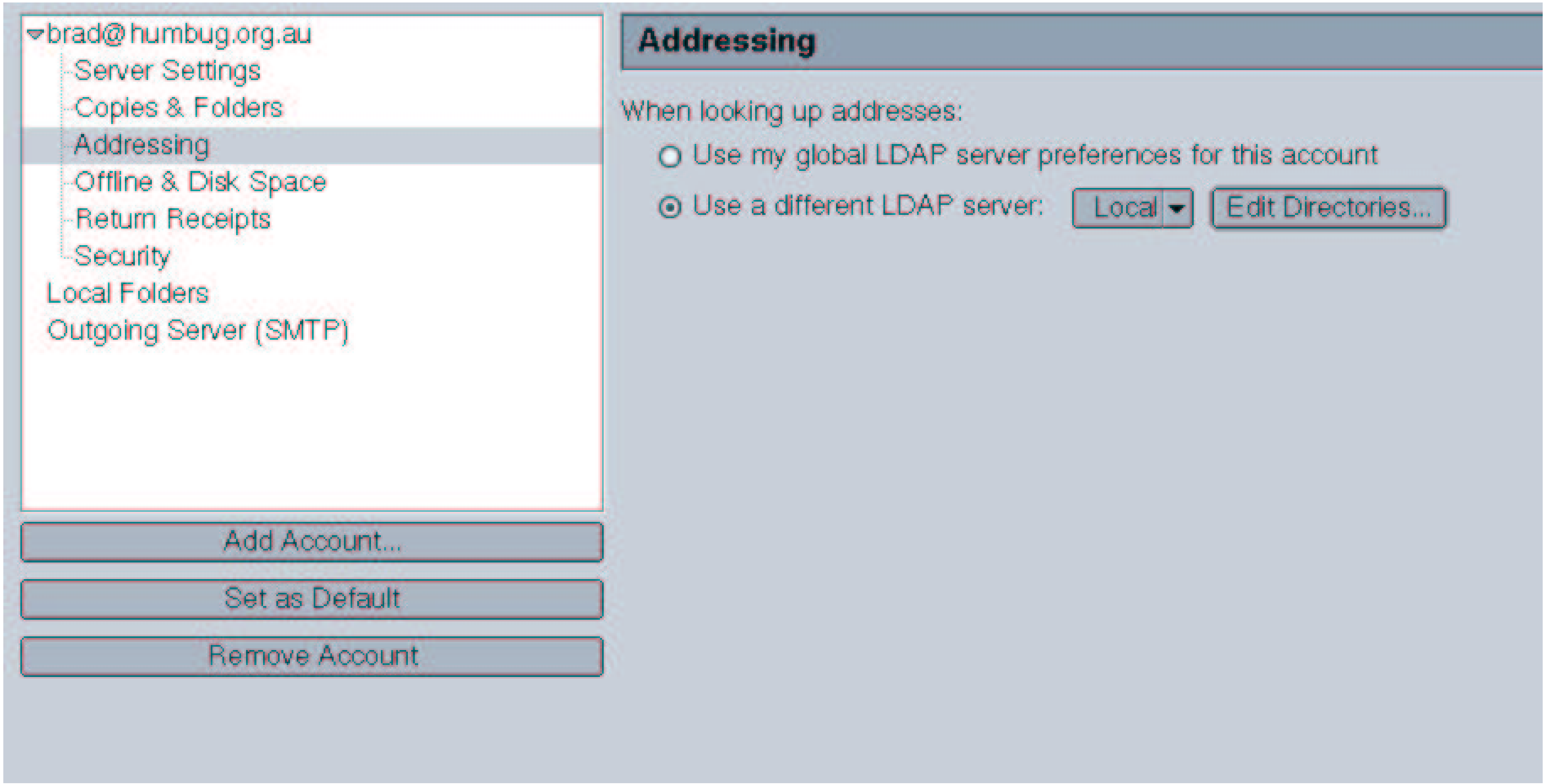
Netscape Addressbook and LDAP

Go to:

- Edit | Mail & Newsgroup Account Setup | Addressing
- Click on Edit Directories | Add
- Fill out hostname, base DN etc

Now when you compose a message, it will search your ldap server.

Netscape Addressbook Adding



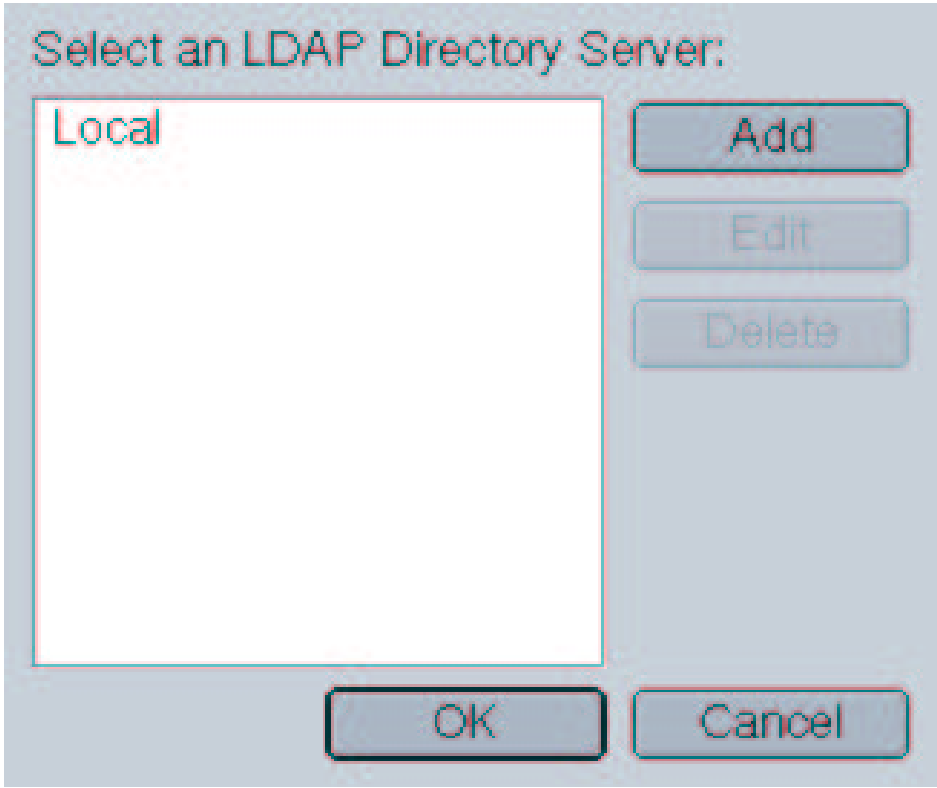
Addressing

When looking up addresses:

○ Use my global LDAP server preferences for this account

● Use a different LDAP server: Local

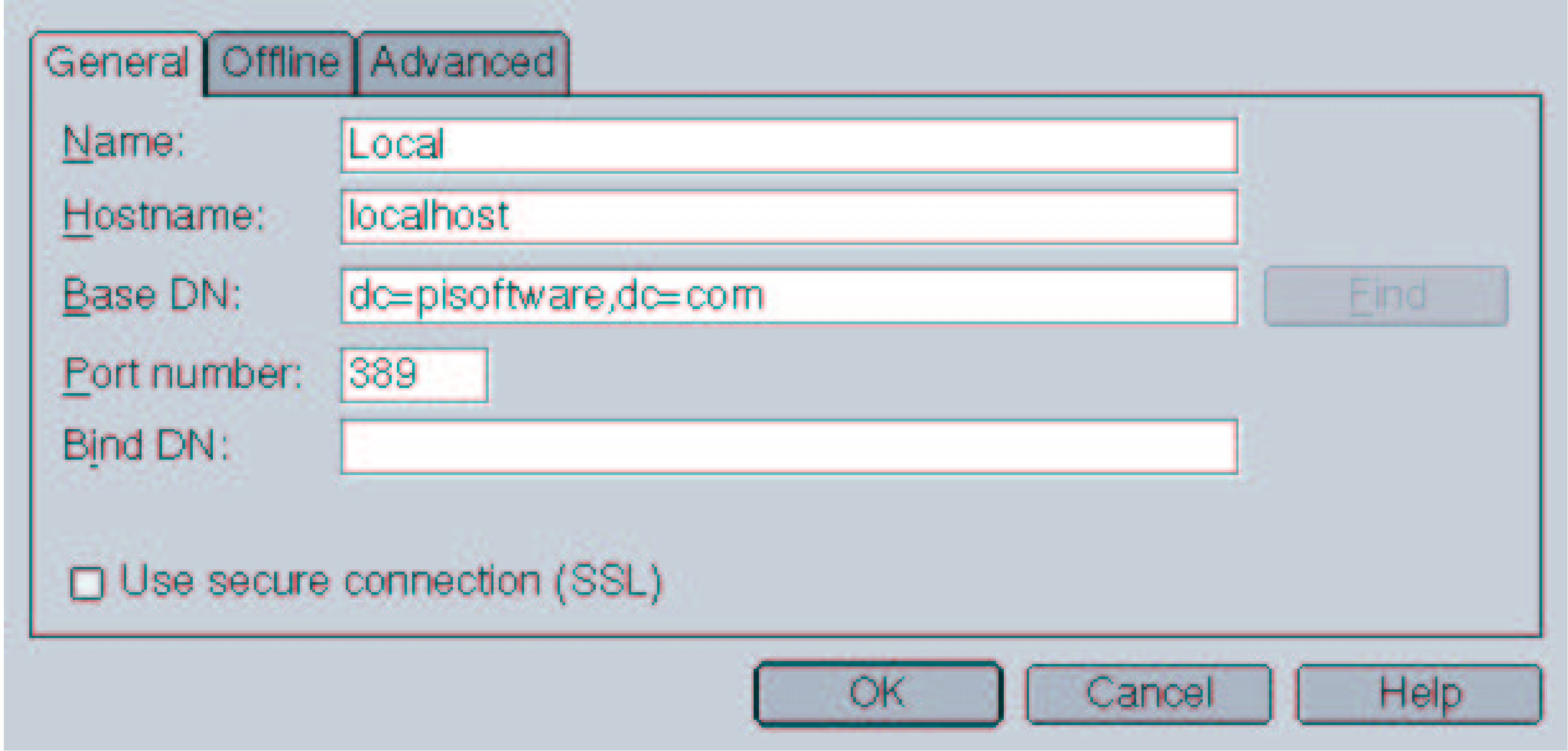
Netscape Addressbook Editing



Select an LDAP Directory Server:

Local

Netscape Addressbook Editing cont



Name: Local

Hostname: localhost

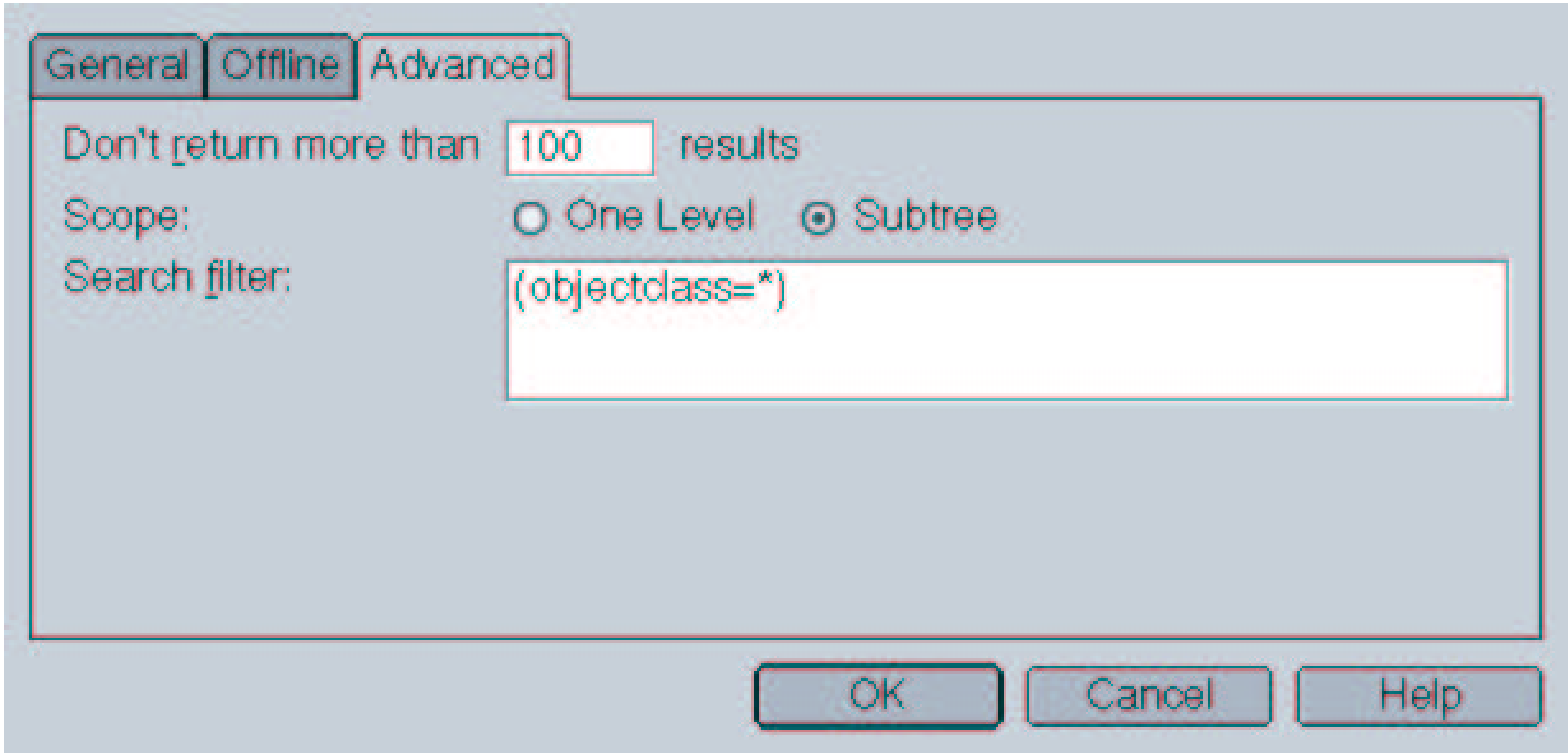
Base DN: dc=pisoftware,dc=com

Port number: 389

Bind DN:

□ Use secure connection (SSL)

Netscape Addressbook Editing cont



Don't return more than 100 results

Scope: ○ One Level ● Subtree

Search filter: (objectclass=*)

Active Directory and LDAP

Provides a directory for a Microsoft network:

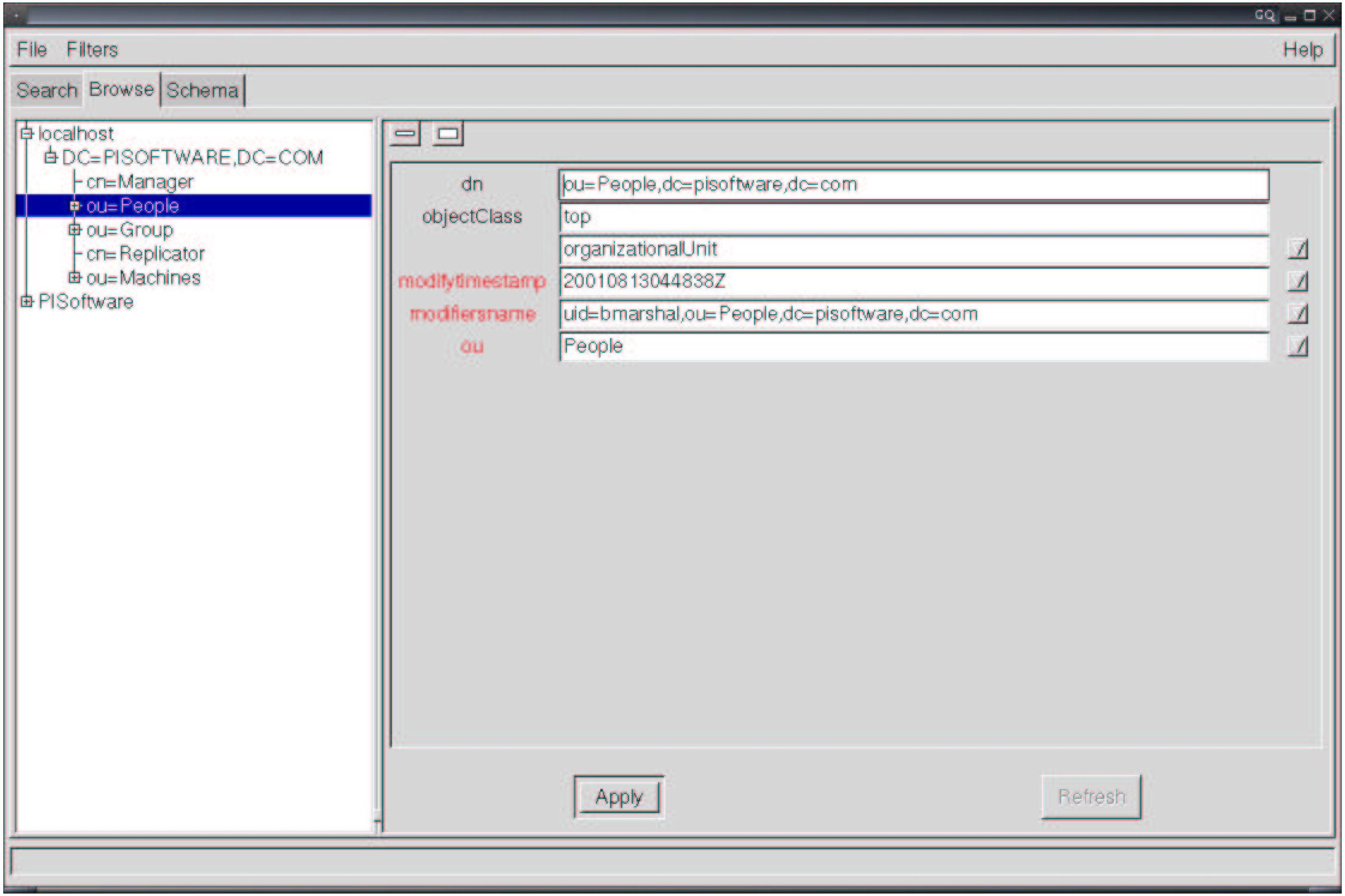
- Centrally manage
- Central security
- Central user administration
- Integrates with DNS
- Information replication
- Provides all the services a domain controller did

LDAP GUIs

There are many LDAP administration GUIs, such as:

- directory administrator: Manages users and groups
- gq: Browse and search LDAP schemas and data
- ldapexplorer: PHP based administration tools
- vlad: LDAP visualisation tools (browse and edit attributes)
- eudc: Emacs Unified Directory Client - common interface to LDAP, bbdb etc

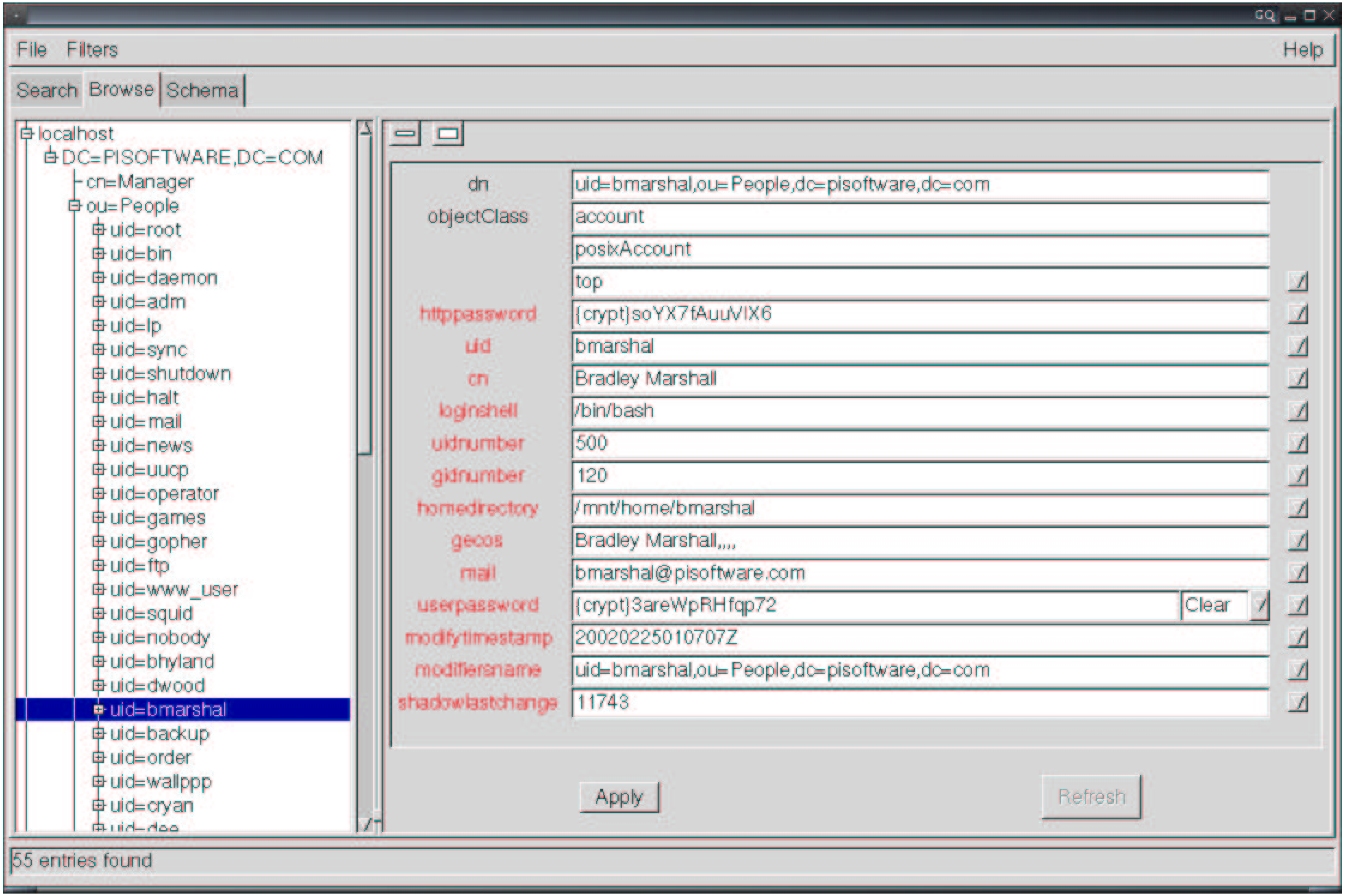
LDAP GUIs - GQ View People



- localhost
  - DC=PISOFTWARE,DC=COM
    - cn=Manager
    - ou=People
    - ou=Group
    - cn=Replicator
    - ou=Machines
- PISoftware

dn	ou=People,dc=pisoftware,dc=com
objectClass	top
	organizationalUnit
modifytimestamp	20010813044838Z
modifiersname	uid=bmarshal,ou=People,dc=pisoftware,dc=com
ou	People

LDAP GUIs - GQ View User

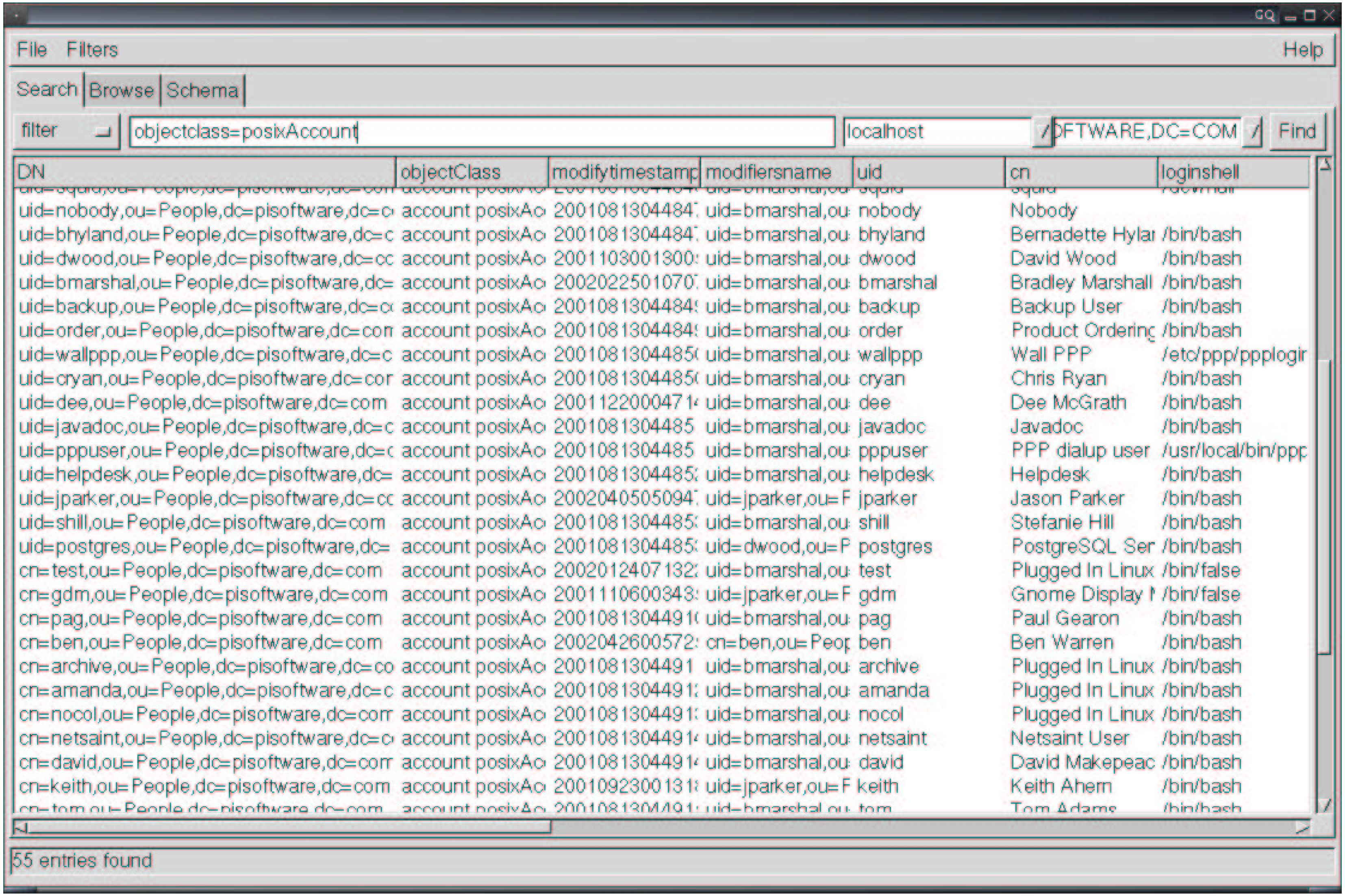


localhost
DC=PISOFTWARE,DC=COM
cn=Manager
ou=People
uid=root
uid=bin
uid=daemon
uid=adm
uid=lp
uid=sync
uid=shutdown
uid=halt
uid=mail
uid=news
uid=uucp
uid=operator
uid=games
uid=gopher
uid=ftp
uid=www_user
uid=squid
uid=nobody
uid=bhyland
uid=dwood
uid=bmarshal
uid=backup
uid=order
uid=wallppp
uid=cryan
uid=dee

dn	uid=bmarshal,ou=People,dc=pisoftware,dc=com
objectClass	account
	posixAccount
	top
httppassword	{crypt}soYX7fAuuVIX6
uid	bmarshal
cn	Bradley Marshall
loginshell	/bin/bash
uidnumber	500
gidnumber	120
homedirectory	/mnt/home/bmarshal
gecos	Bradley Marshall,,,,
mail	bmarshal@pisoftware.com
userpassword	{crypt}3areWpRHfqp72 Clear
modifytimestamp	20020225010707Z
modifiersname	uid=bmarshal,ou=People,dc=pisoftware,dc=com
shadowlastchange	11743

55 entries found

LDAP GUIs - GQ Search



objectclass=posixAccount

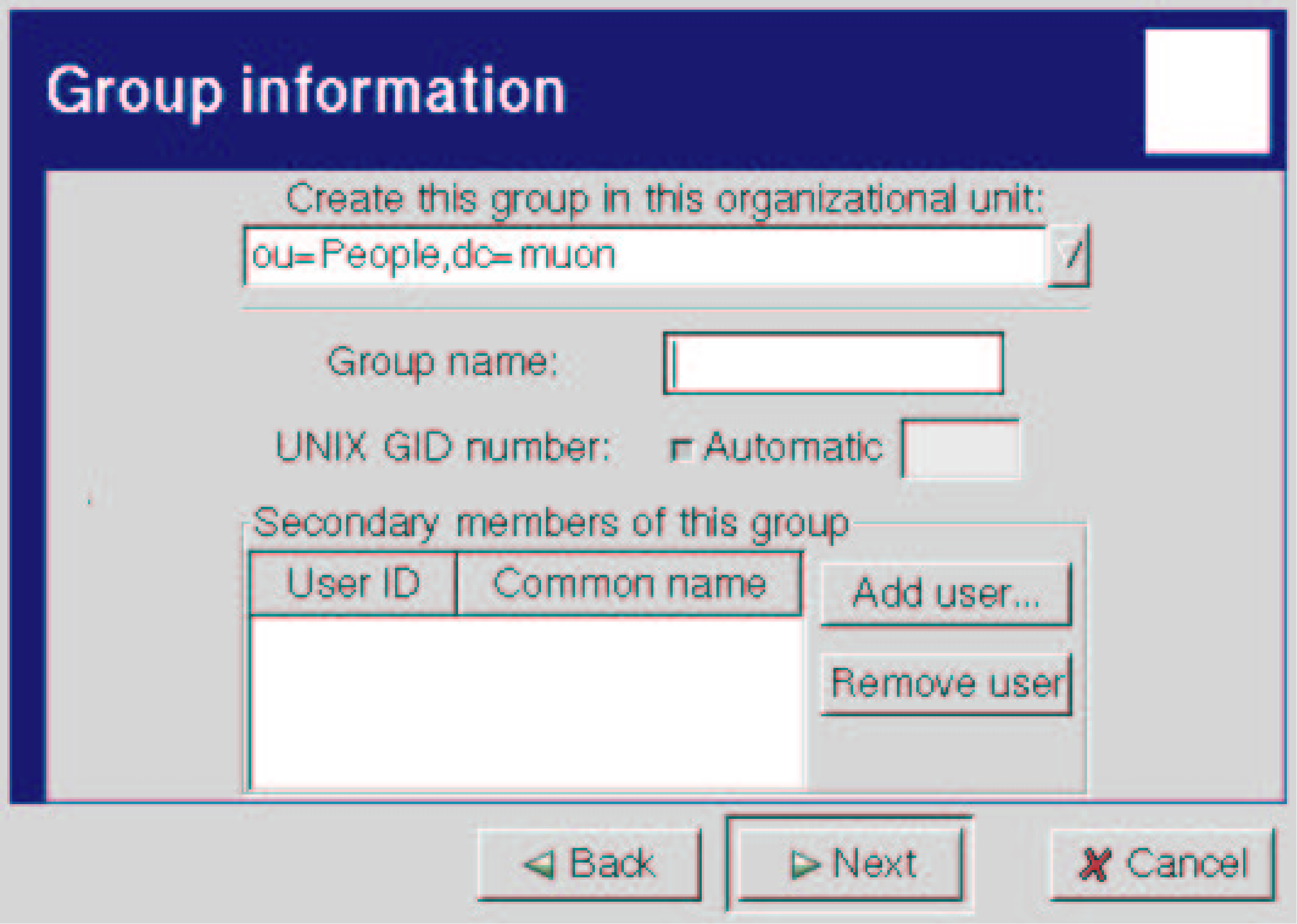
localhost

DFTWARE,DC=COM

DN	objectClass	modifytimestamp	modifiersname	uid	cn	loginshell
uid=squid,ou=People,dc=pisoftware,dc=com	account posixAc	2001081304484	uid=bmarshal,ou	squid	squid	/dev/null
uid=nobody,ou=People,dc=pisoftware,dc=c	account posixAc	2001081304484	uid=bmarshal,ou	nobody	Nobody	
uid=bhyland,ou=People,dc=pisoftware,dc=c	account posixAc	2001081304484	uid=bmarshal,ou	bhyland	Bernadette Hylar	/bin/bash
uid=dwood,ou=People,dc=pisoftware,dc=cc	account posixAc	200110300130	uid=bmarshal,ou	dwood	David Wood	/bin/bash
uid=bmarshal,ou=People,dc=pisoftware,dc=	account posixAc	2002022501070	uid=bmarshal,ou	bmarshal	Bradley Marshall	/bin/bash
uid=backup,ou=People,dc=pisoftware,dc=c	account posixAc	2001081304484	uid=bmarshal,ou	backup	Backup User	/bin/bash
uid=order,ou=People,dc=pisoftware,dc=con	account posixAc	2001081304484	uid=bmarshal,ou	order	Product Ordering	/bin/bash
uid=wallppp,ou=People,dc=pisoftware,dc=c	account posixAc	2001081304485	uid=bmarshal,ou	wallppp	Wall PPP	/etc/ppp/ppplogir
uid=cryan,ou=People,dc=pisoftware,dc=cor	account posixAc	2001081304485	uid=bmarshal,ou	cryan	Chris Ryan	/bin/bash
uid=dee,ou=People,dc=pisoftware,dc=com	account posixAc	200112200047 1	uid=bmarshal,ou	dee	Dee McGrath	/bin/bash
uid=javadoc,ou=People,dc=pisoftware,dc=c	account posixAc	2001081304485	uid=bmarshal,ou	javadoc	Javadoc	/bin/bash
uid=pppuser,ou=People,dc=pisoftware,dc=	account posixAc	2001081304485	uid=bmarshal,ou	pppuser	PPP dialup user	/usr/local/bin/ppp
uid=helpdesk,ou=People,dc=pisoftware,dc=	account posixAc	2001081304485	uid=bmarshal,ou	helpdesk	Helpdesk	/bin/bash
uid=jparker,ou=People,dc=pisoftware,dc=cc	account posixAc	2002040505094	uid=jparker,ou=F	jparker	Jason Parker	/bin/bash
uid=shill,ou=People,dc=pisoftware,dc=com	account posixAc	2001081304485	uid=bmarshal,ou	shill	Stefanie Hill	/bin/bash
uid=postgres,ou=People,dc=pisoftware,dc=	account posixAc	2001081304485	uid=dwood,ou=P	postgres	PostgreSQL Ser	/bin/bash
cn=test,ou=People,dc=pisoftware,dc=com	account posixAc	200201240713 2	uid=bmarshal,ou	test	Plugged In Linux	/bin/false
cn=gdm,ou=People,dc=pisoftware,dc=com	account posixAc	200111060034 3	uid=jparker,ou=F	gdm	Gnome Display I	/bin/false
cn=pag,ou=People,dc=pisoftware,dc=com	account posixAc	200108130449 1	uid=bmarshal,ou	pag	Paul Gearon	/bin/bash
cn=ben,ou=People,dc=pisoftware,dc=com	account posixAc	2002042600572	cn=ben,ou=Peop	ben	Ben Warren	/bin/bash
cn=archive,ou=People,dc=pisoftware,dc=co	account posixAc	200108130449 1	uid=bmarshal,ou	archive	Plugged In Linux	/bin/bash
cn=amanda,ou=People,dc=pisoftware,dc=c	account posixAc	200108130449 1	uid=bmarshal,ou	amanda	Plugged In Linux	/bin/bash
cn=nocol,ou=People,dc=pisoftware,dc=com	account posixAc	200108130449 1	uid=bmarshal,ou	nocol	Plugged In Linux	/bin/bash
cn=netsaint,ou=People,dc=pisoftware,dc=c	account posixAc	200108130449 1	uid=bmarshal,ou	netsaint	Netsaint User	/bin/bash
cn=david,ou=People,dc=pisoftware,dc=com	account posixAc	200108130449 1	uid=bmarshal,ou	david	David Makepeac	/bin/bash
cn=keith,ou=People,dc=pisoftware,dc=com	account posixAc	200109230013 1	uid=jparker,ou=F	keith	Keith Ahern	/bin/bash
cn=tom,ou=People,dc=pisoftware,dc=com	account posixAc	200108130449 1	uid=bmarshal,ou	tom	Tom Adams	/bin/bash

55 entries found

LDAP GUIs - Directory Admin Group



Group information

Create this group in this organizational unit: ou=People,dc=muon

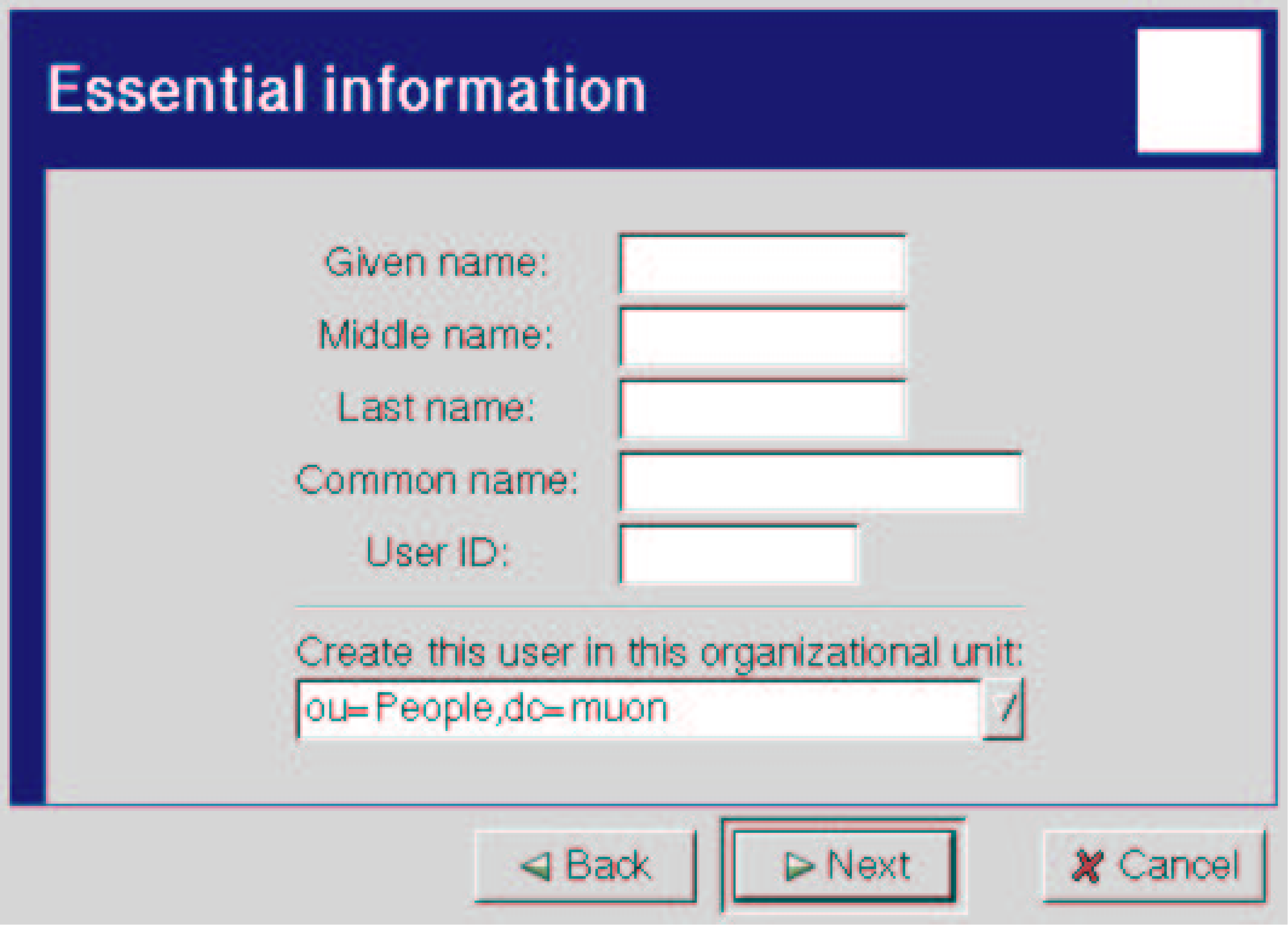
Group name:

UNIX GID number: ☐ Automatic

Secondary members of this group

User ID	Common name

LDAP GUIs - Directory Admin New User



Essential information

Given name:

Middle name:

Last name:

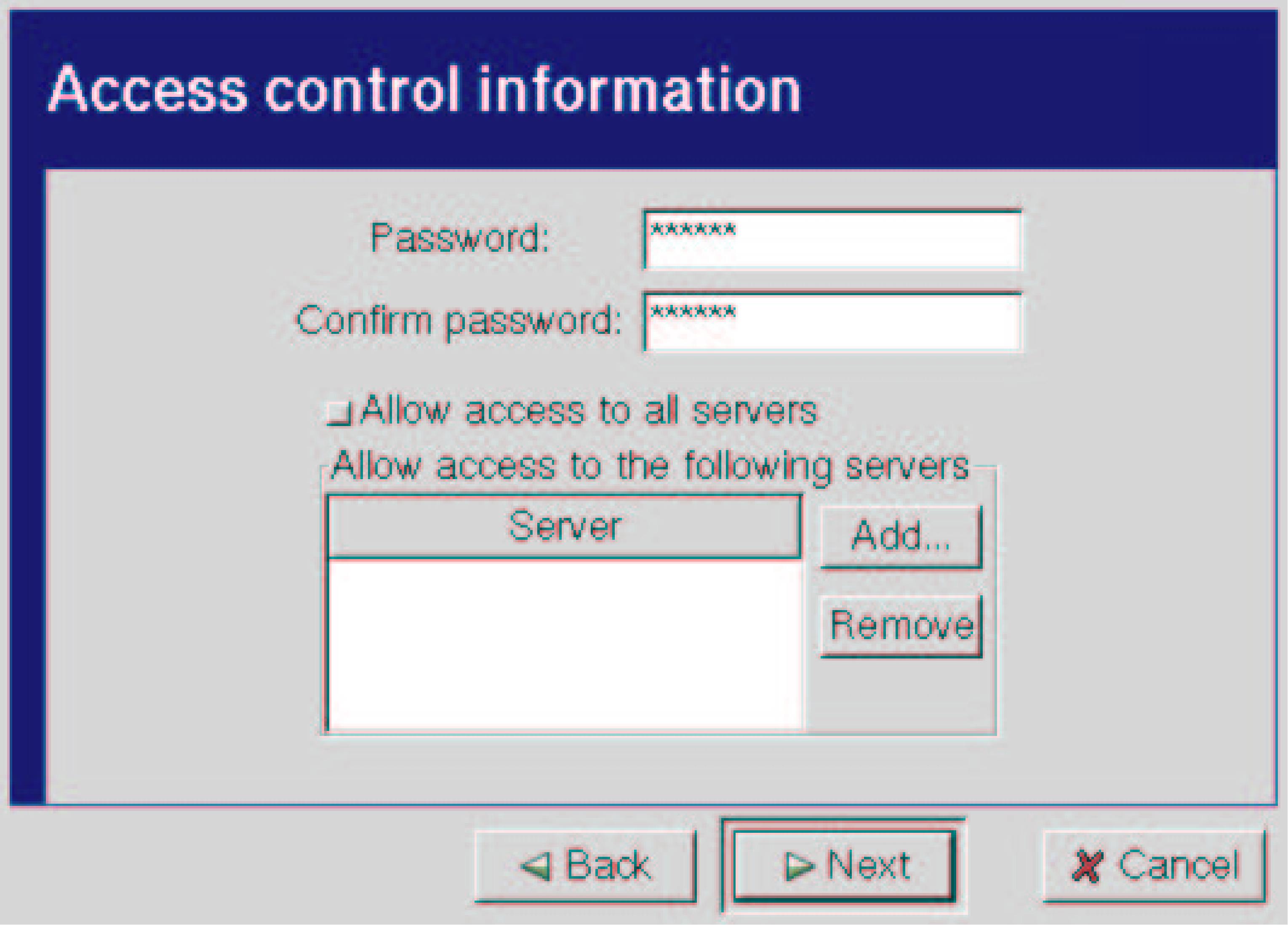
Common name:

User ID:

Create this user in this organizational unit:

ou=People,dc=muon

LDAP GUIs - Directory Admin New User



Access control information

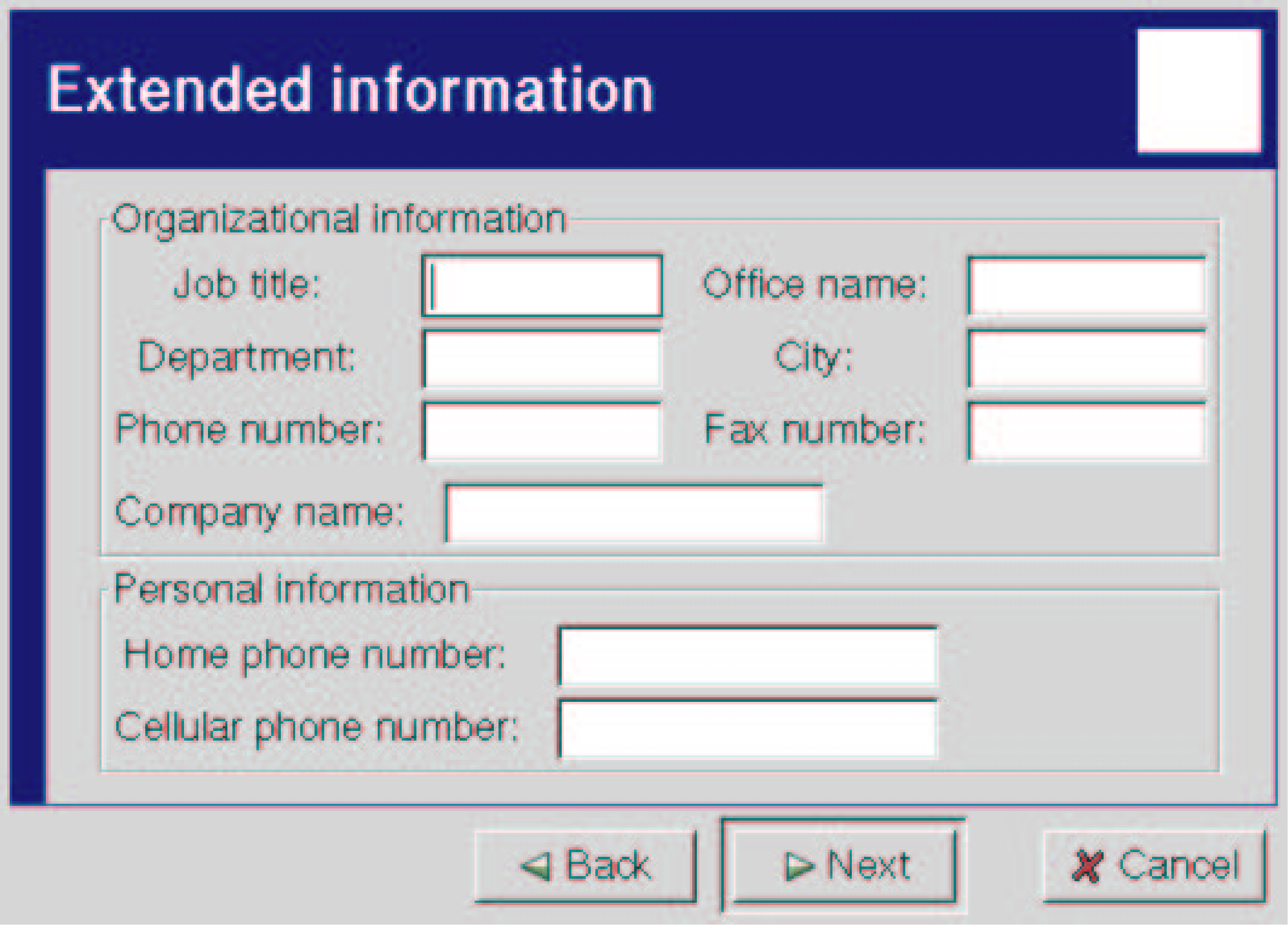
Password: *******

Confirm password: *******

□ Allow access to all servers

Allow access to the following servers

LDAP GUIs - Directory Admin New User



Extended information

Organizational information

Job title:

Office name:

Department:

City:

Phone number:

Fax number:

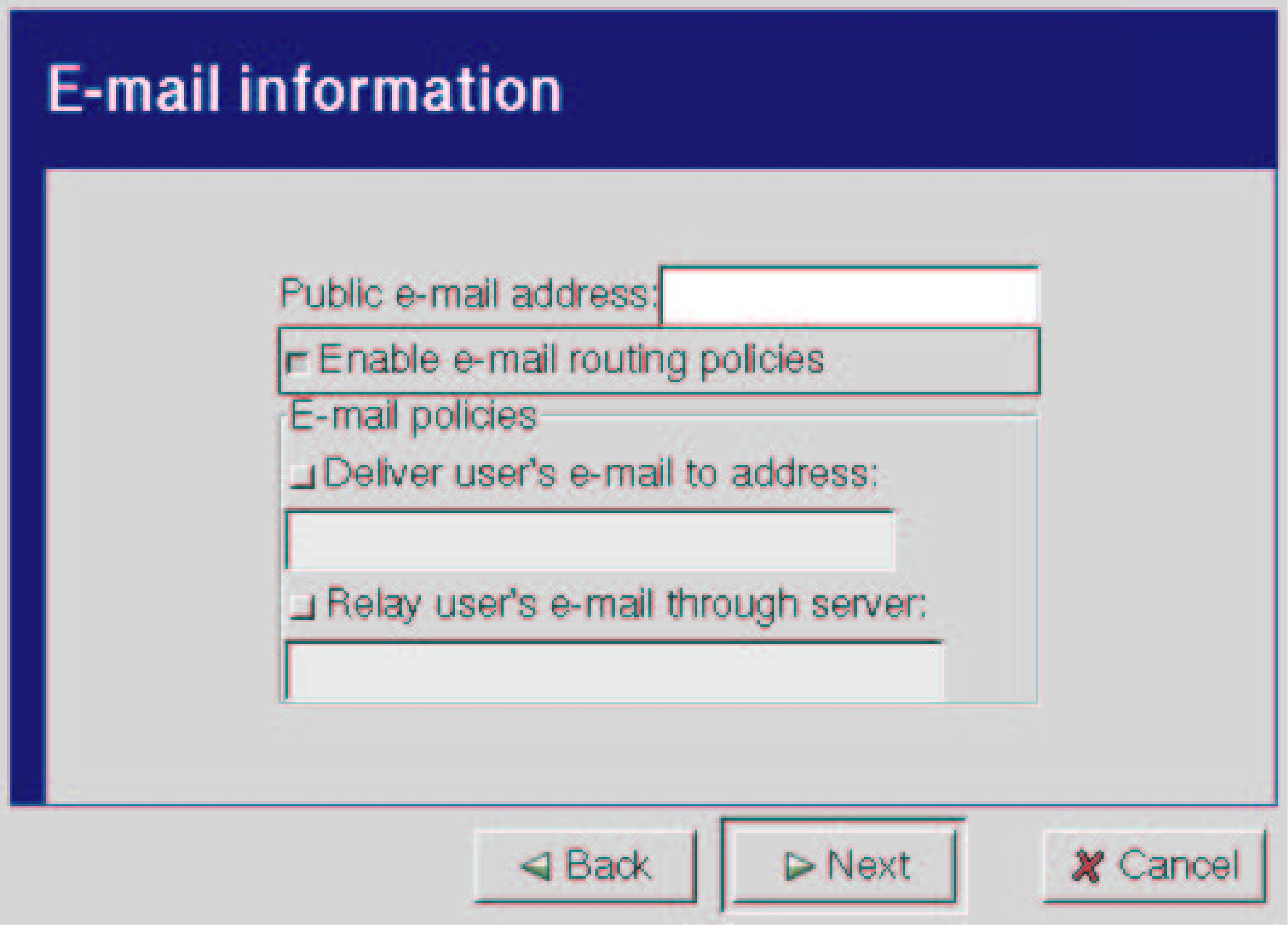
Company name:

Personal information

Home phone number:

Cellular phone number:

LDAP GUIs - Directory Admin New User



E-mail information

Public e-mail address:

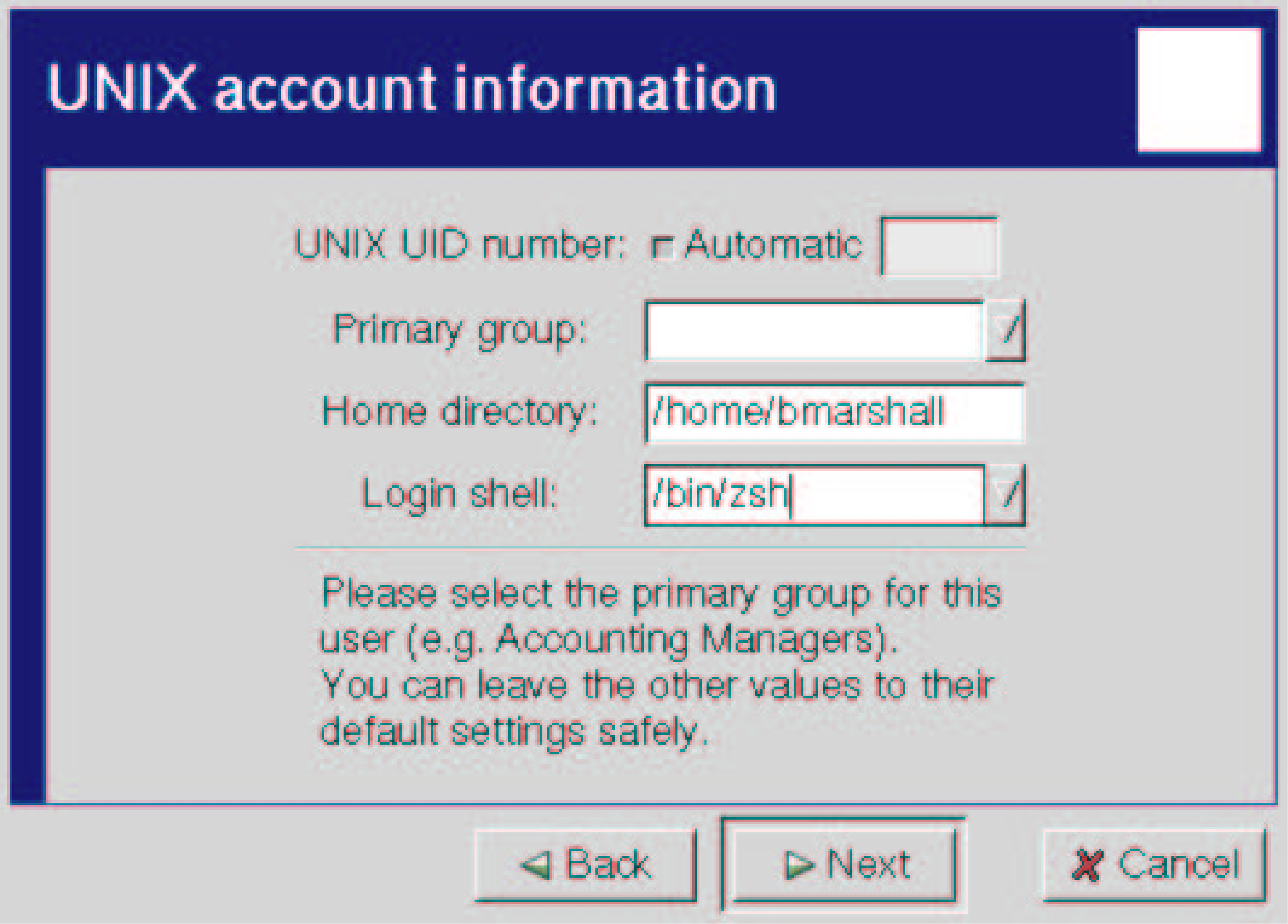
☑ Enable e-mail routing policies

E-mail policies

□ Deliver user's e-mail to address:

□ Relay user's e-mail through server:

LDAP GUIs - Directory Admin New User



UNIX account information

UNIX UID number: □ Automatic

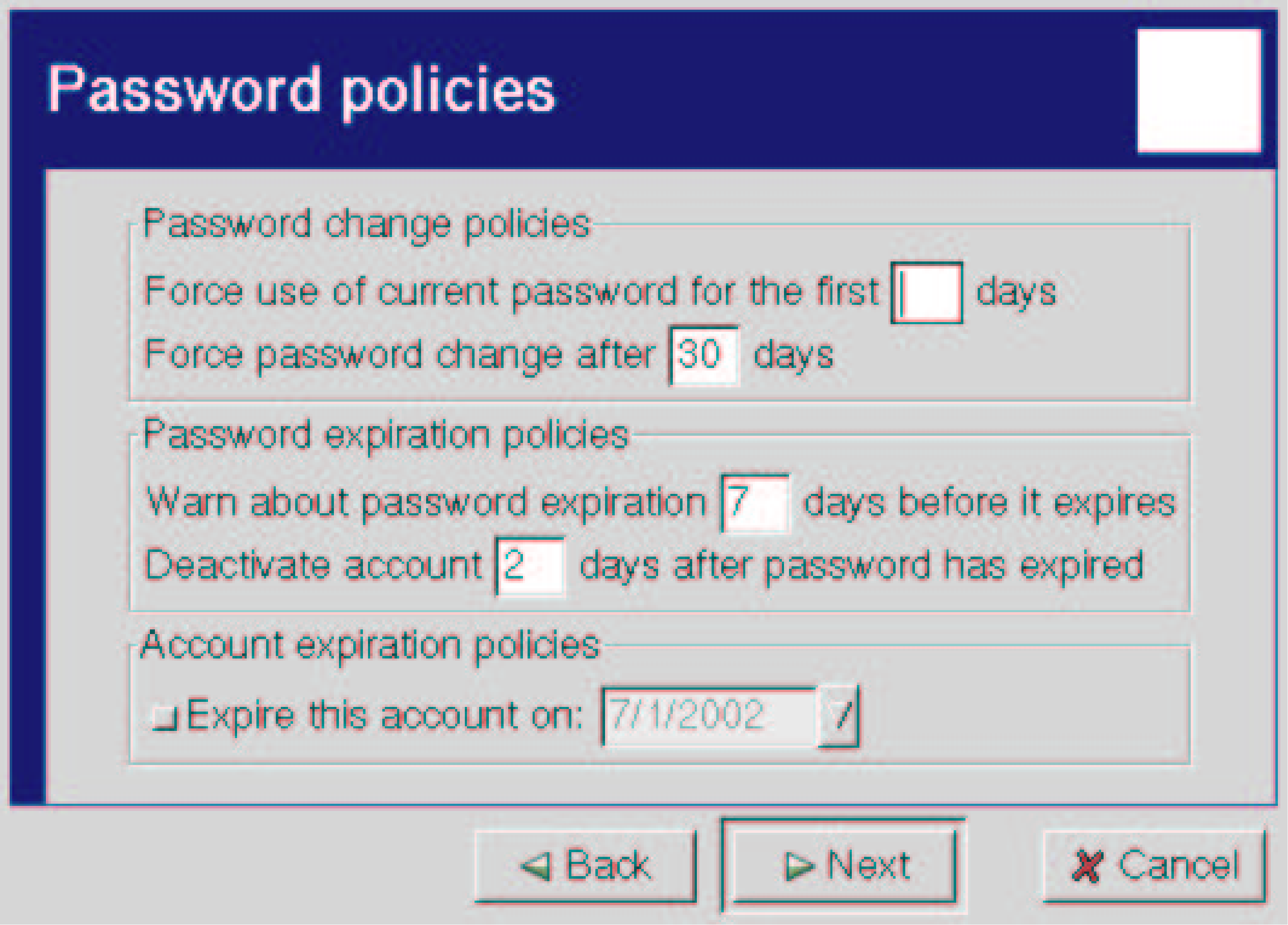
Primary group:

Home directory: /home/bmarshall

Login shell: /bin/zsh

Please select the primary group for this user (e.g. Accounting Managers).
You can leave the other values to their default settings safely.

LDAP GUIs - Directory Admin New User



Password policies

Password change policies

Force use of current password for the first [ ] days

Force password change after [30] days

Password expiration policies

Warn about password expiration [7] days before it expires

Deactivate account [2] days after password has expired

Account expiration policies

□ Expire this account on: [7/1/2002]

Perl and LDAP - Basic Query

```
use Net::LDAP;
my($ldap) = Net::LDAP->new(’ldap.example.com’)
   or die "Can’t bind to ldap: $!\n";
$ldap->bind;
my($mesg) = $ldap->search(
base => "dc=pisoftware,dc=com",
            filter => ’(objectclass=*)’);
$mesg->code && die $mesg->error;
map { $_->dump } $mesg->all_entries;
# OR
foreach $entry ($mesg->all_entries)
       { $entry->dump; }
$ldap->unbind;
```

Perl and LDAP - Adding

```
$ldap->bind(
            dn       => $manager,
            password => $password,
        );

$result = $ldap->add( dn => $groupdn,
             attr => [ ’cn’ => ’Test User’,
                       ’sn’ => ’User’,
                       ’uid’ => ’test’,
                     ];
$ldap->unbind;
```

Perl and LDAP - Deleting

```
$ldap->bind(
            dn       => $manager,
            password => $password,
        );

$ldap->delete( $groupdn );
$ldap->unbind;
```

Perl and LDAP - Modifying

```
$ldap->modify( $dn,
       changes => [
               # Add sn=User
           add     => [ sn => ’User’ ],
               # Delete all fax numbers
           delete  => [ faxNumber => []],
               # Delete phone number 911
           delete  => [ telephoneNumber =>
                    [’911’]],
               # Change email address
           replace => [ email =>
                    ’test@pisoftware.com’]
   ]
);
$ldap->unbind;
```

Questions?

Any Questions ?

References

- Understanding and Deploying LDAP Directory Services
  Timothy A. Howes, Mark C. Smith and Gordon S. Good
  Macmillan Network Architecture and Development Series

- Implementing LDAP
  Mark Wilcox
  Wrox Press Ltd

- Perl for System Administration
  David N. Blank-Edelman
  O'Reilly