

LDAPifying Applications

Brad Marshall

`brad.marshall@member.sage-au.org.au`

Contents

- LDAP Servers
- OpenLDAP
- Linux Authentication
- PAM and Name Service Switch (NSS)
- System Authentication
- Sendmail and LDAP
- Apache and LDAP
- Squid and LDAP
- Netscape Addressbook and LDAP
- Active Directory and LDAP
- LDAP GUIs
- Perl and LDAP

LDAP Servers

- Slapd
 - University of Michigan
 - Openldap
- iPlanet/SunONE Directory Server
- Microsoft Active Directory (AD)
- Novell eDirectory
- Oracle Internet Directory
- IBM SecureWay Directory
- Critical Path InJoin Directory Server
- Data Connection Directory
- OctetString Virtual Directory Engine

Openldap

- Based on UMich Idap server
- Available from <http://www.openldap.org/>
- Versions:
 - Historic: 1.2.13 - implements LDAPv2
 - Stable: 2.0.27 - implements LDAPv3
 - Release: 2.1.21 - 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 am I? Extended Operation
 - LDAP no-op Control
 - Matched Values Control
 - Misc LDAP Feature Extensions
 - DNS-based service location
- 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 backend databases - see next slide
 - Multiple database instances
 - Access control - via ACLs and tcp wrappers
 - Threaded
 - Replication
 - Security - privacy via TLS, authentication via SASL
 - Internationalization

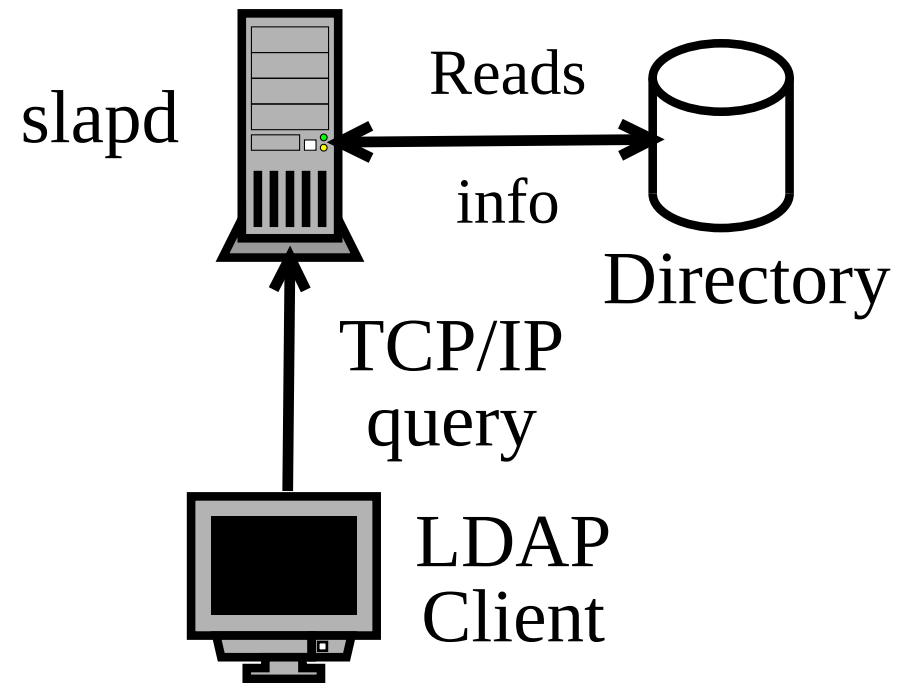
slapd backend databases

- BDB - Sleepycat Berkeley DB backend - standard in OpenLDAP 2.1
- DNSSRV - dns based backend to serve referrals from SRV records
- LDAP - ldap proxy backend
- LDBM - high performance disk based db - uses BerkeleyDB, GNU DBM, MDBM or NDBM
- META - ldap proxy backend for multiple servers and naming context masq - similar to LDAP
- NULL - null backend db, similar to /dev/null

slapd backend databases cont

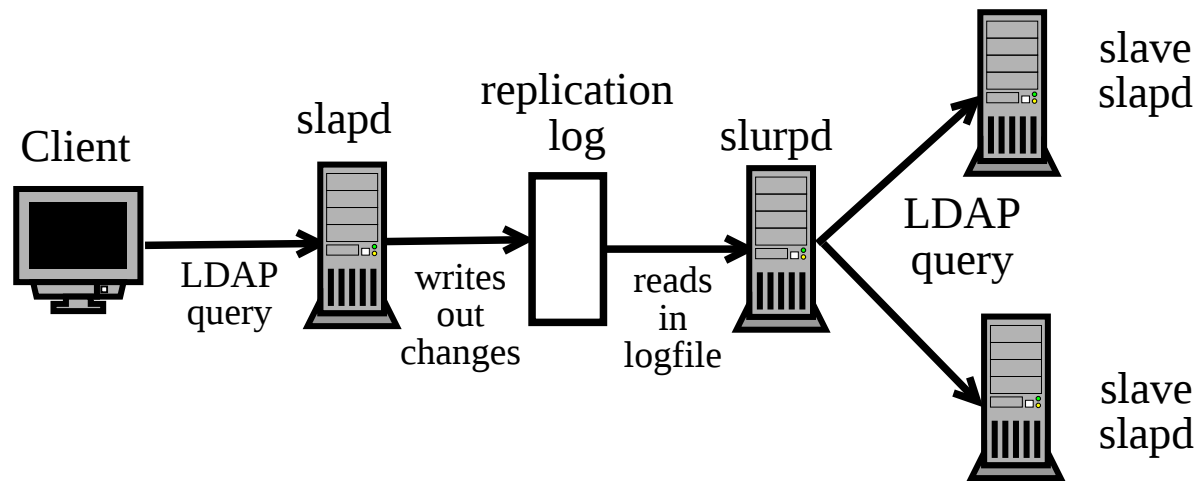
- SHELL - shell interpreter embedded backend
- PERL - perl interpreter embedded backend
- TCL - tcl interpreter embedded backend
- PASSWORD - simple password file db - serves up user account info from /etc/passwd style files
- SQL - mapping sql to ldap to present information from legacy RDBMS (in OpenLDAP 2.x)

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
-
```

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/
```

ACL for who

Can restrict by:

- Distinguished Name
- Filter that matches some attributes
- Attributes

ACL for what

Can restrict with:

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

ACL permissions

Permissions are:

- none
- auth
- compare
- search
- read
- write

ACL Priority

Access control priority:

- Local database
- Global rules
- Runs thru in order the rules appear in the config file
- First checks what is being requested, then who
- First matching rule is used
- This means ordering is important

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
```

OpenLDAP and SASL

- SASL - Simple Authentication and Security Layer (RFC2222)
- Offers several industry standard authentication mechanisms
 - PLAIN, LOGIN
 - DIGEST-MD5
 - KERBEROS_V4
 - GSSAPI
 - EXTERNAL

SASL Authentication

- Basic steps:
 - Configure slapd to communicate with client program (service key, public key, shared secret)
 - Map authentications identities to LDAP DN
- Authentication ID
 - If realm is the default, can leave that section out completely

```
uid=<username>,cn=<realm>,  
cn=<mechanism>,cn=auth
```

Mapping Auth Id to LDAP Entries

- Not intended that cn=auth exists, use mapping to existing users
- Use sasl-regexp directives to define maps
- sasl-regexp <search pattern> <replacement pattern>
- Search pattern uses regex as per regex(7)
 - . = any char
 - * = zero or more of previous char
 - + = one or more of previous char
 - ? = zero or one of previous char
 - () = store match in \$n, where n is the n'th paren set
- Replacement pattern is users DN, or LDAP URL

sasl-regex examples

```
sasl-regex uid=(.*),cn=digest-md5,cn=auth  
uid=$1,ou=People,dc=pisoftware,dc=com
```

```
sasl-regex uid=(.*),cn=pisoftware.com,  
cn=kerberos_v4,cn=auth  
uid=$1,ou=People,dc=pisoftware,dc=com
```

```
sasl-regex uid=(.*),cn=digest-md5,cn=auth  
ldap:///ou=People,dc=pisoftware,dc=com  
??sub?(uid=$1)
```

sasl-regex Recommendations

- Don't set search pattern too leniently - easy to allow access when shouldn't
- Allow for realm being omitted, as well as explicit realm entry
- List explicit realm entry first
- If users are spread over multiple ou's, use a LDAP URL
- If LDAP URL returns more than one or zero entries, authentication fails

SASL DIGEST-MD5

- Client and server share a secret
- Server generates challenge, client response proving it knows the secret
- Stores secrets either in directory (Cyrus SASL 2.1) or separate database (sasldb)
- Obviously important to protect passwords - either ACLs or file permissions
- Shared secrets needs access to plain text password

DIGEST-MD5 Passwords

- Secrets stored in sasldb (Cyrus SASL 2.1)

```
$ saslpasswd2 -c <username>
```

- Secrets stored in LDAP directory
 - Password stored in userPassword in clear text
 - slapd.conf needs:

```
password-hash {CLEARTEXT}
```

- Authentication id form:

```
uid=<username>,cn=<realm>,  
cn=digest-md5,cn=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.pem  
TLSCertificateKeyFile   /usr/lib/ssl/misc/server.pem  
TLSCACertificateFile    /usr/lib/ssl/misc/server.pem  
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 - Subordinate

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/
```

Referral - Superior

To specify another ldap server to go to if the current request is outside the servers naming context, use the referral directive.

```
referral          ldap://root.openldap.org:389/
```

Referral - ManageDsaIT

- Managing referral objects is done using a tool which supports the ManageDsaIT control
- Tells the server that you want to manage the referral object as an entry
- Stops server from sending a referral result
- Use the -M option to ldapmodify or ldapsearch

OpenLDAP Schemas

Schema	Use
core	OpenLDAP core
cosine	Cosine and Internet X.500 (RFC 1274)
inetorgperson	InetOrgPerson
misc	Assorted
nis	Network Information Services (RFC 2307)
openldap	OpenLDAP Project
java	Java Object (RFC 2714)
corba	Corba Object References (RFC 2714)
krb5-kdc	Kerberos KDC
netscape-profile	Netscape Roaming Profiles
sendmail	Sendmail LDAP Routing

RootDSE

To discover what the server supports, use something like:

```
$ ldapsearch -s base -b "" +
```

```
dn:
```

```
namingContexts: dc=pisoftware,dc=com
```

```
supportedControl: 2.16.840.1.113730.3.4.2
```

```
supportedExtension: 1.3.6.1.4.1.4203.1.11.1
```

```
supportedExtension: 1.3.6.1.4.1.1466.20037
```

```
supportedFeatures: 1.3.6.1.4.1.4203.1.5.1
```

```
supportedLDAPVersion: 2
```

```
supportedLDAPVersion: 3
```

```
supportedSASLMechanisms: CRAM-MD5
```

```
supportedSASLMechanisms: DIGEST-MD5
```

```
subschemaSubentry: cn=Subschema
```

Schema Discovery

To discover what schemas etc the server supports, use something like:

```
$ ldapsearch -s base -b "cn=Subschema" +
```

It will return:

- ldapSyntaxes
- matchingRules
- attributeTypes
- objectClasses

Server Monitoring

- Compile slapd with `--enable-monitor`
- Added the following to `slapd.conf`:

```
modulepath          /usr/lib/ldap
moduleload           back_monitor
# The backend type
database             monitor
# Access controls
access to *
        by dn="cn=admin,dc=gumby" write
        by * read
```

Server Monitoring

- To search do the following:

```
$ ldapsearch -x -b 'cn=Monitor'
```

- Top level output:

```
dn: cn=Monitor
objectClass: top
objectClass: monitor
objectClass: extensibleObject
cn: Monitor
description: @(#) $OpenLDAP: slapd 2.1.17
             (May 17 2003 22:02:20) $
```

SunONE Directory Server

- Originally based on U.Mich LDAP server
- Was Netscape Directory Server, then Iplanet, then SunONE
- Available from <http://www.sun.com/>
- Current version is 5.2
- Platforms supported:
 - Solaris
 - Linux
 - Windows 2000
 - HP-UX
 - AIX

SunONE Directory Companion Products

- Directory Proxy Server
 - Provides a firewall for the directory - can route requests
- Identity Server
 - Help manage secure access to web-based resources
- Identity Synchronization for Windows
 - Helps synchronize authentication data between Windows NT, Active Directory and SunONE
- Metadirectory
 - Consolidates information from disparate sources, eg directorys and databases

SunONE Directory Server Components

- Directory server
- Admin server
- Server console for remote management
- Command line tools
- SNMP agent
- Migration tools for previous versions
- Client tools

SunONE Directory Server Architecture

- Core server to process requests
- Directory server console for managing server
- Frontends for LDAP, DSML and SNMP
- Plugins for access control, replication etc
- Initial directory tree, for server config etc

SunONE Directory Server features

- LDAPv3 - RFC2251
 - Search filters - RFC2254
 - Search references (smart referrals)
 - LDAP URL - RFC2255
 - LDIF - RFC2849
- DSMLv2
 - HTTP and SOAP transports
 - Native DSML support, not gateway
 - Allows non-LDAP clients access to data
 - Allows interfacing using XML
 - DSML front end is restricted HTTP server
- All access controls apply to both

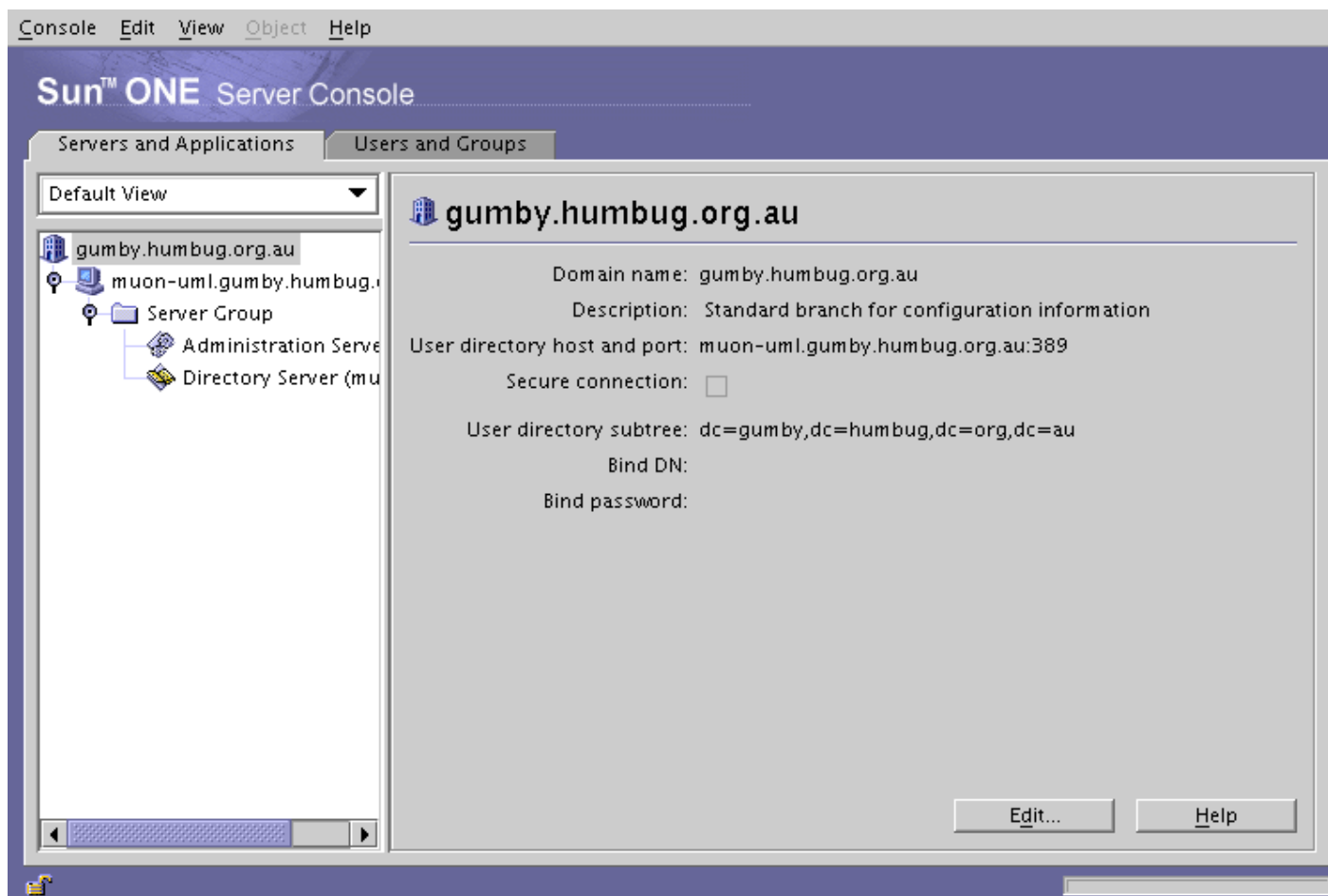
SunONE Directory Server features cont

- Multiplatform - including 64 bit systems
- Multidatabase design
- Large cache support - can support > 4GB caches
- Improved update performance
 - Group flush
 - Index compression
 - Replication compression
 - Improved checkpointing
- Improved searching
 - 64 bit server process
 - Improved algorithms for reading caches

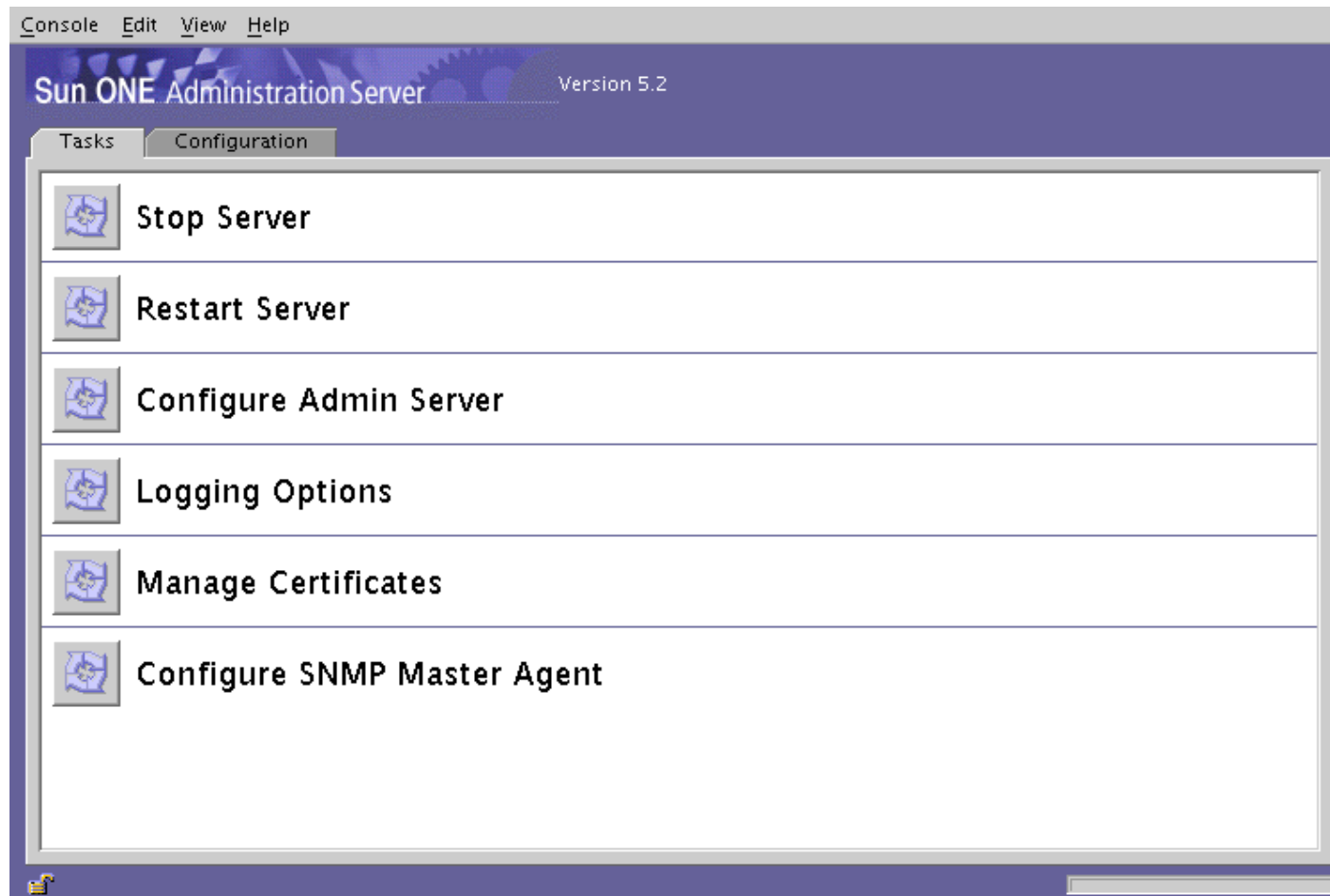
SunONE Directory Server features cont

- Supports Sun Cluster
- Advanced replication
 - Simple replication
 - Cascading replication
 - Multi-master replication
 - Fractional replication
- Indexes
- SSL, TLS and SASL encryption and authentication
- Dynamic groups
- Schema and ACL replication

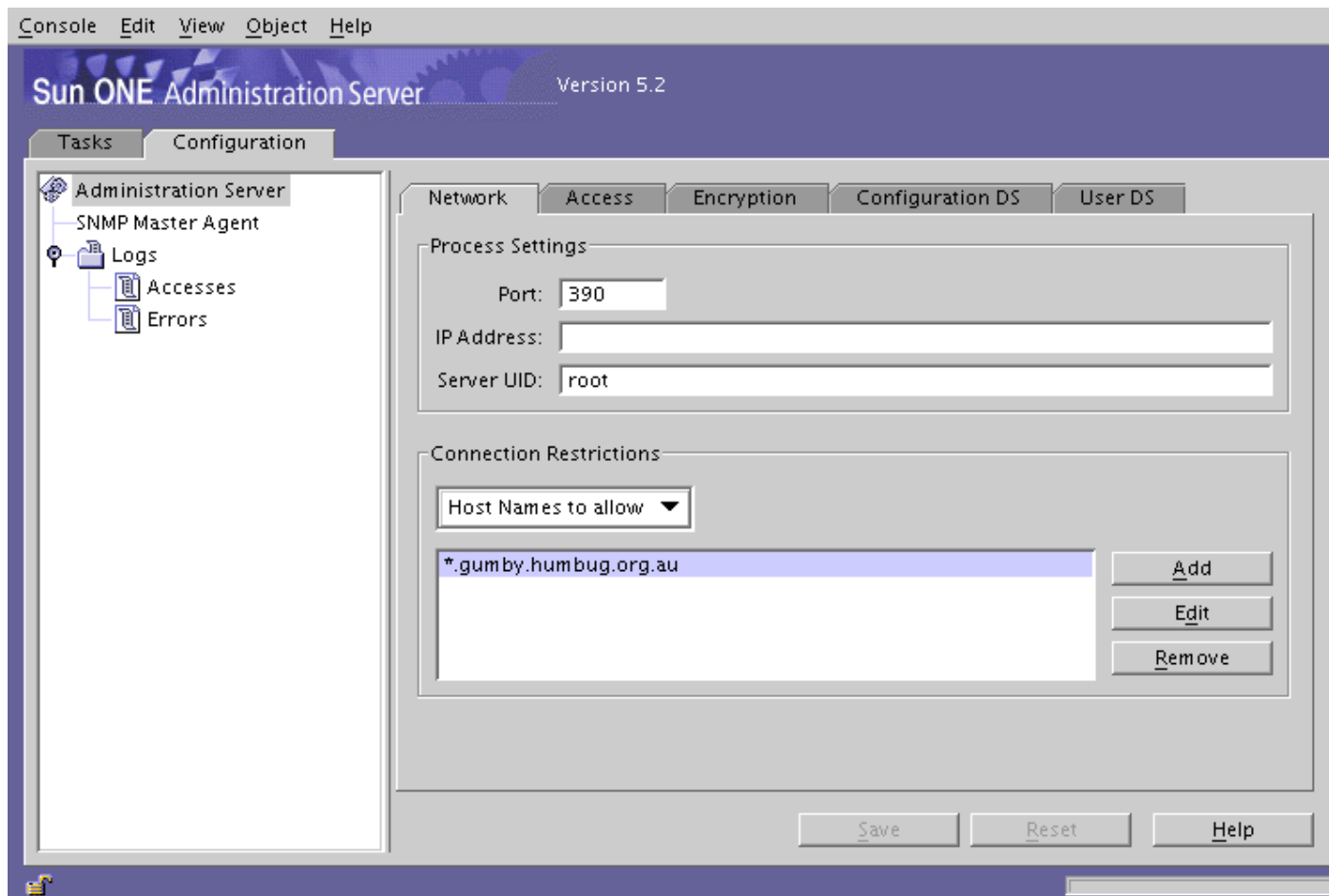
SunONE Server Console



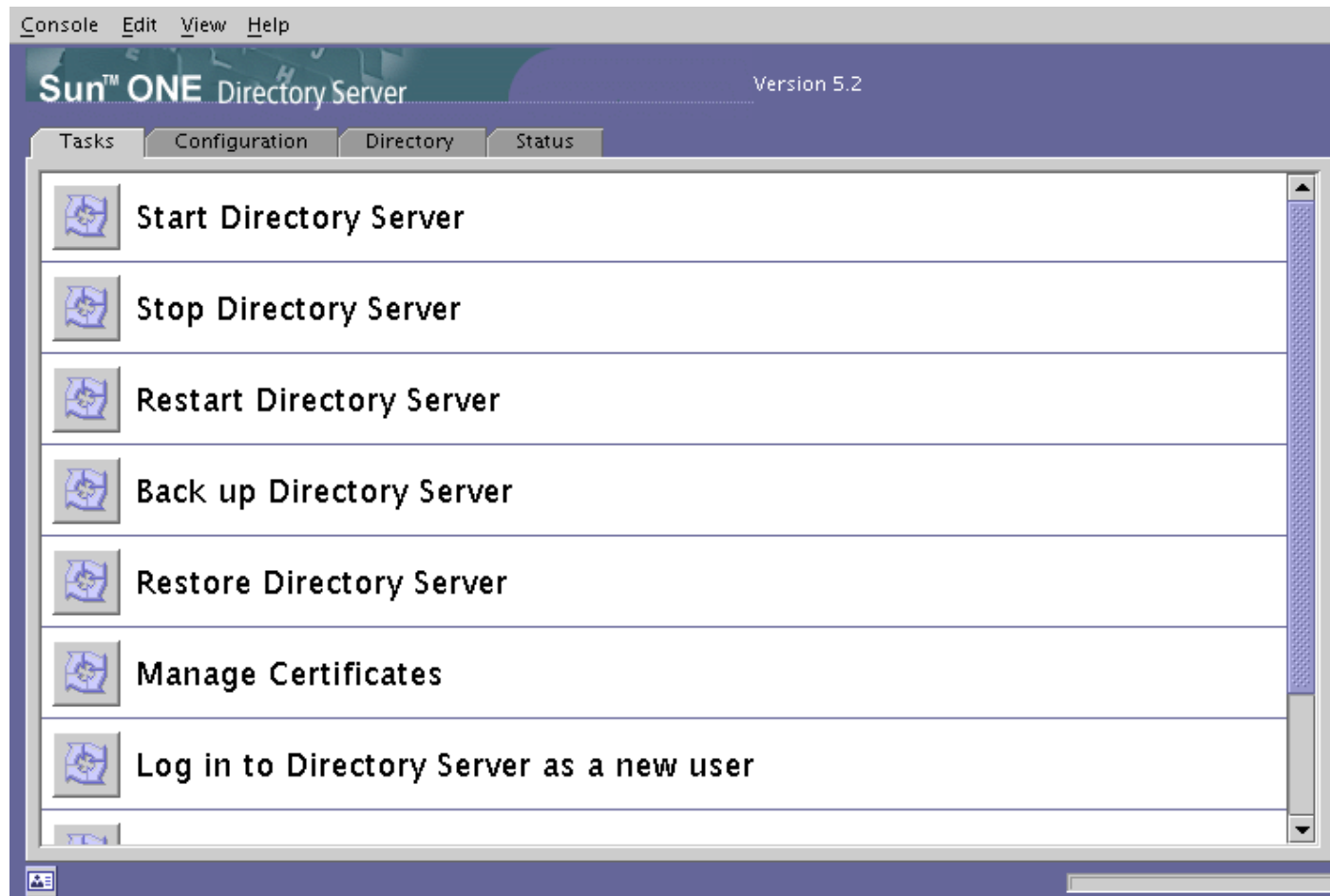
SunONE Admin Tasks



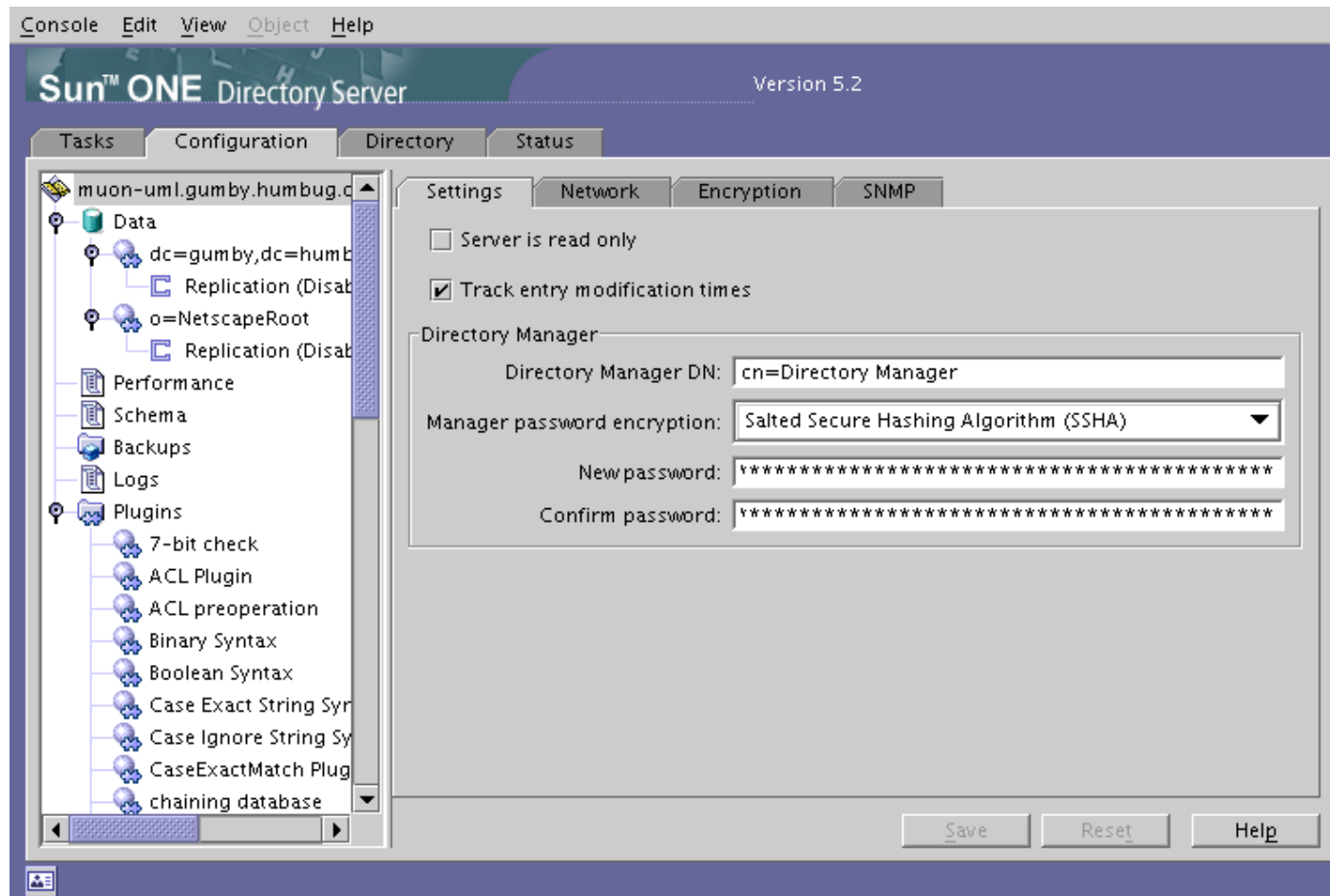
SunONE Admin Config



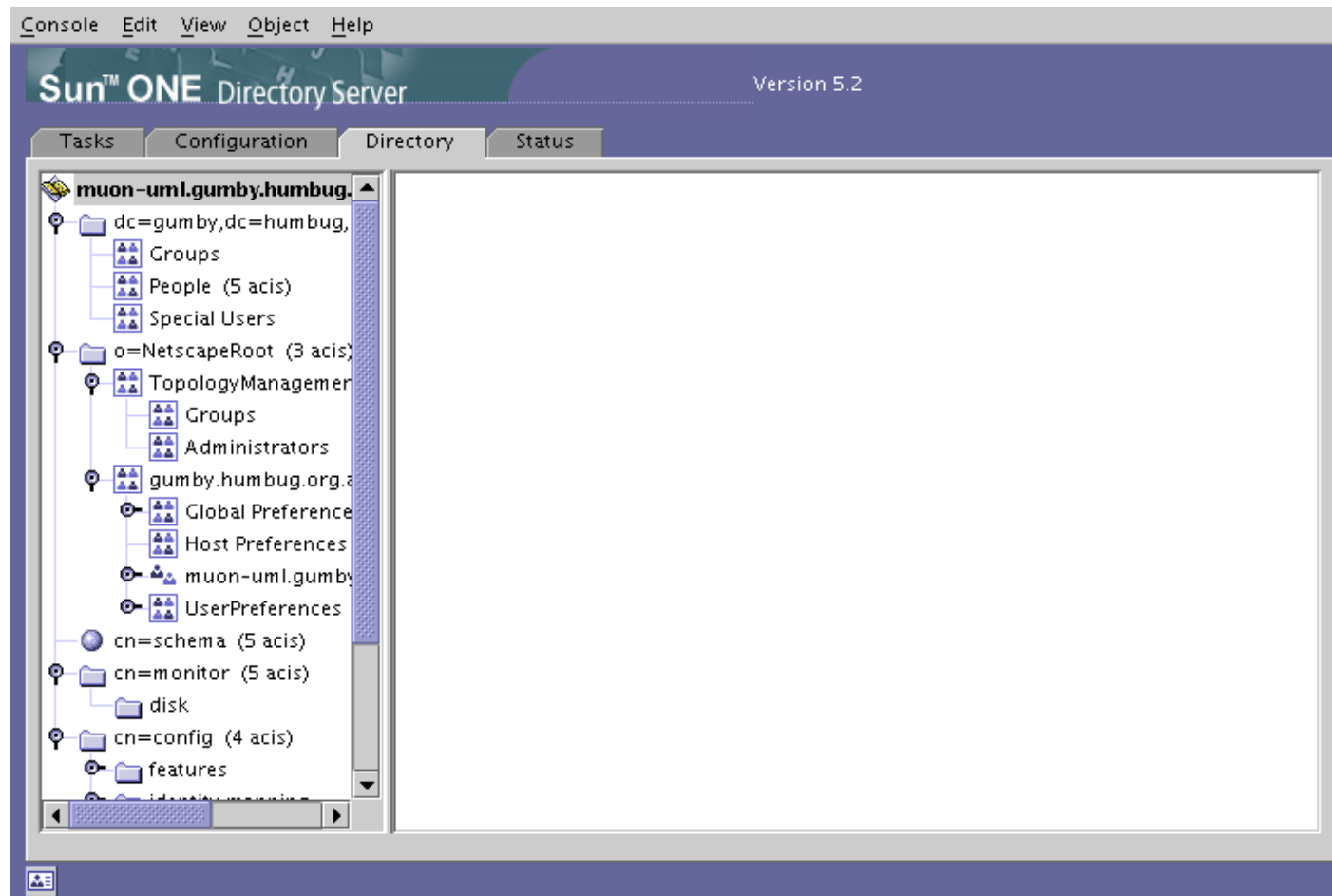
SunONE Directory Server Tasks



SunONE Directory Server Config



SunONE Directory Server Directory



SunONE Directory Server Status

Console Edit View Object Help

Sun™ ONE Directory Server Version 5.2

Tasks Configuration Directory Status

muon-uml.gumby.humbug.org

- Suffixes
- Chained suffixes
- Logs
- Replication

Refresh ☐ Continuous refresh

Startup time on server: 26/07/03 18:24
Current time on server: 26/07/03 20:20

Resource Summary:

Resource	Usage Since Startup	Average Per Minute
Connections	59	0.5
Operations Initiated	699	6.1
Operations Completed	698	6.1
Entries Sent To Clients	1265	11.0
Kilobytes Sent To Clients	1275.2	11.1

Current Resource Usage:

Active Threads	30
Open Connections	11
Remaining Available Connections	949
Threads Waiting To Read From Client	0
Databases In Use	2

Open Connections:

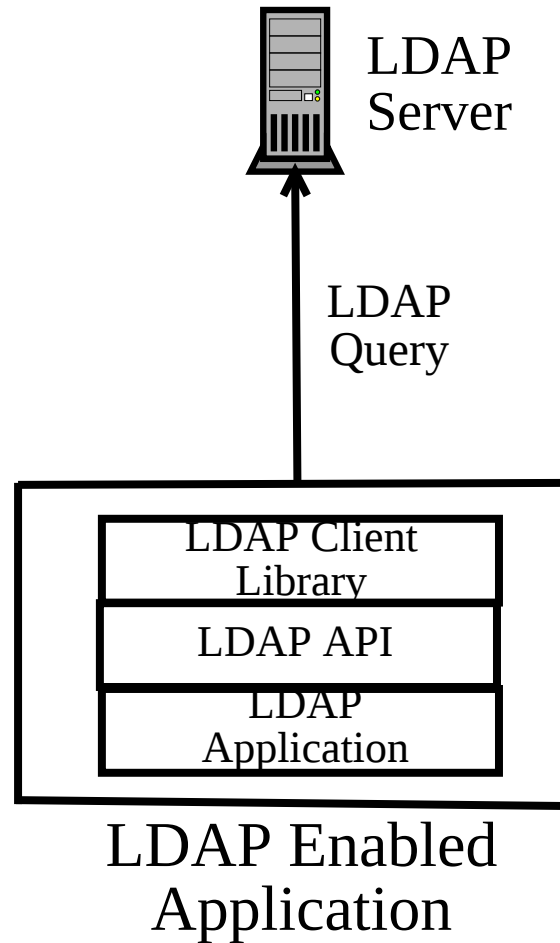
Time Opened	Initiated	Completed	Bound As	State	Type
Sat Jul 26 20:...	313	312	uid=admin,...	Not blocked	LDAP
Sat Jul 26 18:...	1	1	cn=admin-...	Not blocked	LDAP

Help

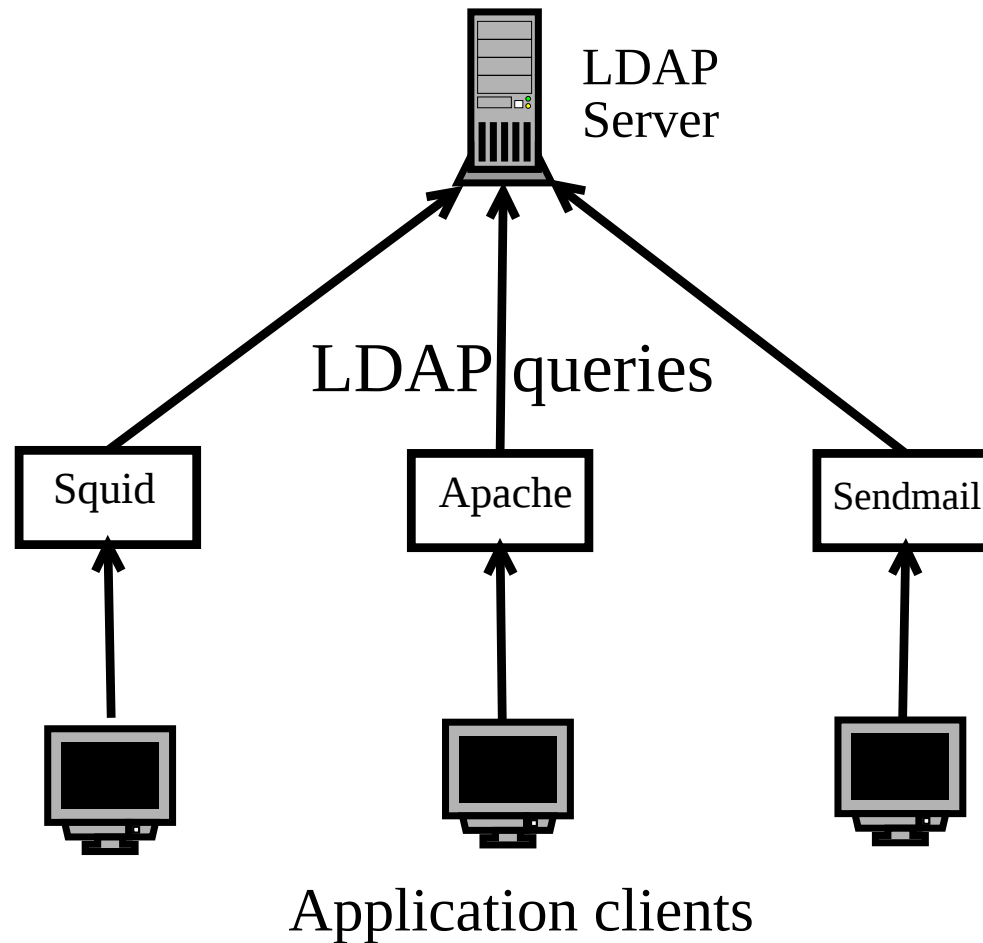
Security Considerations

- Slapd defaults to binding to all IPv4 and IPv6 interfaces, consider binding to only the required ones - eg, listen just on localhost
- Firewall the port to restrict access
- Use tcp wrappers to restrict at application level
- Use TLS or SSL if possible
- Consider VPN / other encryption techniques

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 separate 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 authentication 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 access to user information after authentication
- 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`

Name Service Caching Daemon - NSCD

- Caches name service lookups
- Part of glibc
- Config file is /etc/nscd.conf
- Useful for not requiring an ldap lookup for everything

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.

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
```

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  
      relogfile /var/lib/openssldap/replication.log  
# cleartext passwords, especially for the  
# rootdn, should be avoid.  See slapd.conf(5)  
# for details.  
directory      /var/lib/openssldap/
```

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 # [1]
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
```

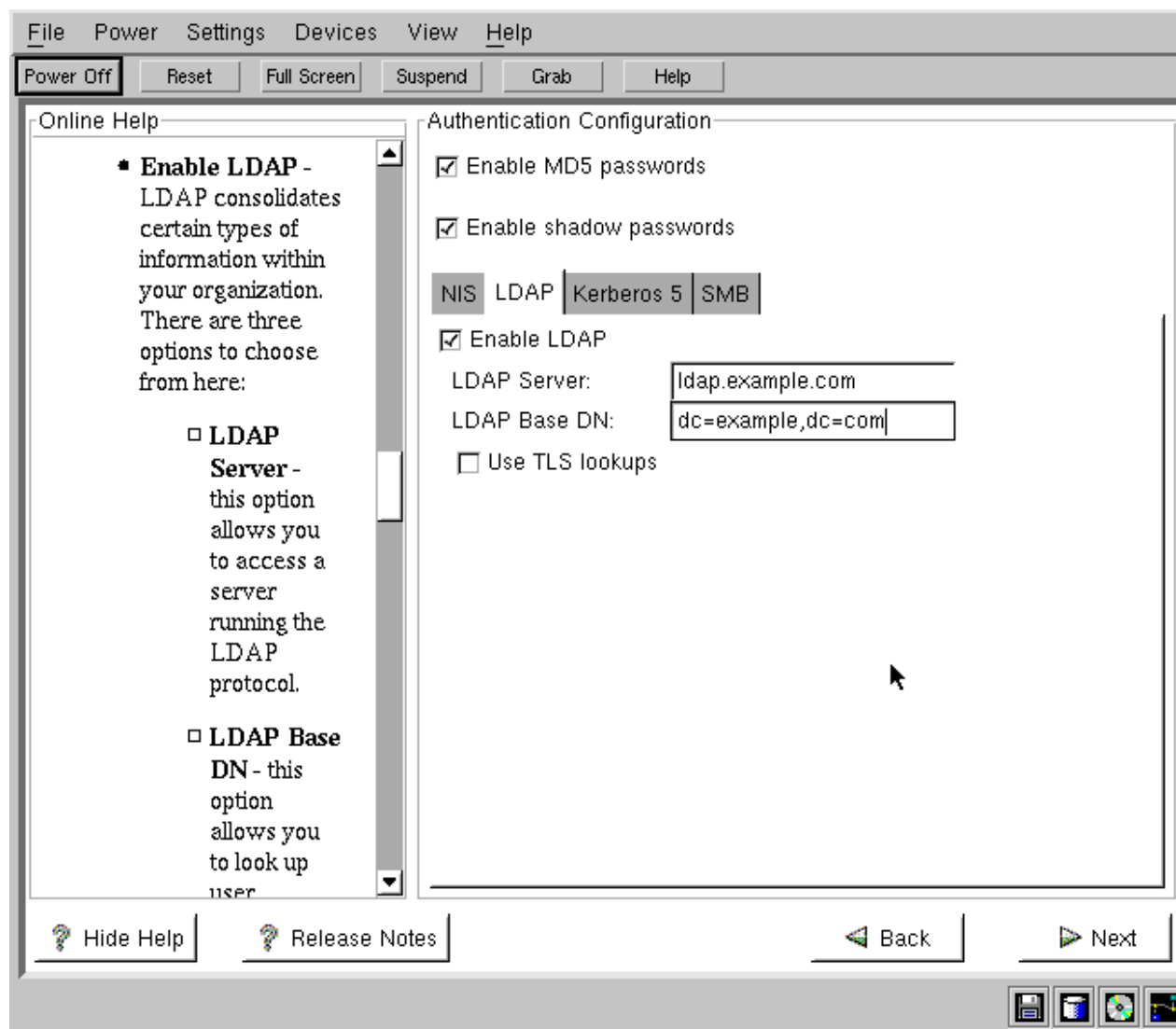
NSS configuration - nsswitch.conf

```
/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.

Redhat 7.3 Install Config



RH7.3 Authconfig - Text

authconfig 4.2.8 - (c) 1999-2001 Red Hat, Inc.

User Information Configuration	
☐ Cache Information	
☐ Use NIS	Domain: _____ Server: _____
☒ Use LDAP	☐ Use TLS Server: 127.0.0.1 Base DN: dc=example,dc=com
☐ Use Hesiod	LHS: _____ RHS: _____
Next Cancel	

<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen

RH7.3 Authconfig - GTK User

User Information | **Authentication**

☐ Cache User Information

NIS
NIS is the Name Information Service. It is commonly used on small to medium networks.

☐ Enable NIS Support [Configure NIS...](#)

LDAP
The Lightweight Directory Access Protocol is a standard way of searching a directory, which can hold arbitrary data in a structured hierarchy. LDAP is increasingly being used in small to large networks.

☐ Enable LDAP Support [Configure LDAP...](#)

Hesiod
Hesiod allows a system administrator to publish user and group information in DNS. It is sometimes used in very large networks.

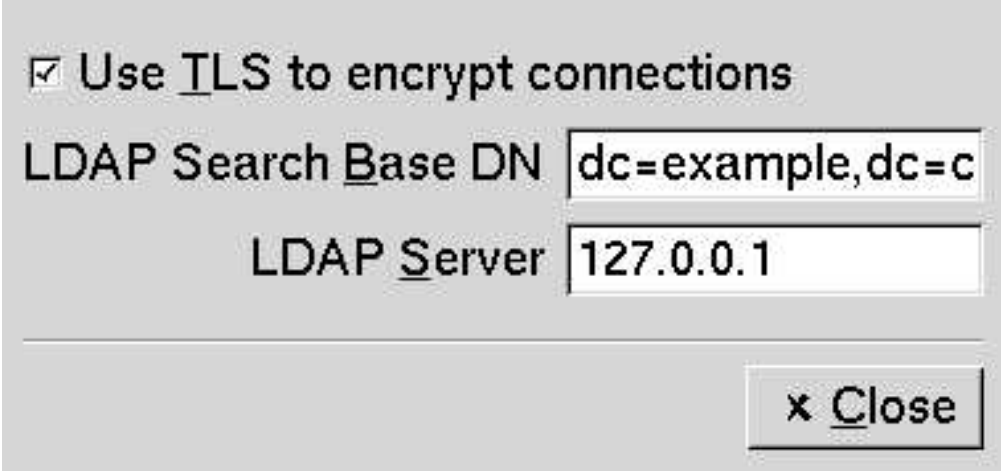
☐ Enable Hesiod Support [Configure Hesiod...](#)

OK **Cancel**

RH7.3 Authconfig - GTK Authentication



RH7.3 Authconfig - GTK LDAP



A screenshot of a GTK dialog box titled "RH7.3 Authconfig - GTK LDAP". The dialog has a light gray background. At the top, there is a checked checkbox labeled "Use TLS to encrypt connections". Below this, there are two text input fields. The first is labeled "LDAP Search Base DN" and contains the text "dc=example,dc=c". The second is labeled "LDAP Server" and contains the text "127.0.0.1". At the bottom right of the dialog, there is a button labeled "x Close".

☒ Use TLS to encrypt connections

LDAP Search Base DN

LDAP Server

Windows LDAP Auth - pGina

- Replacement for domain auth in Windows
- GINA (Graphical Identification and Authentication) module
- Inserts itself between Winlogon and MS's GINA module
- Handles certain operations, passes rest on transparently
- Winlogon loads pGina which then loads plugin
- If plugin allows user to login, will
 - Create account for user
 - Add to specified groups
 - Map drives
 - Other config options

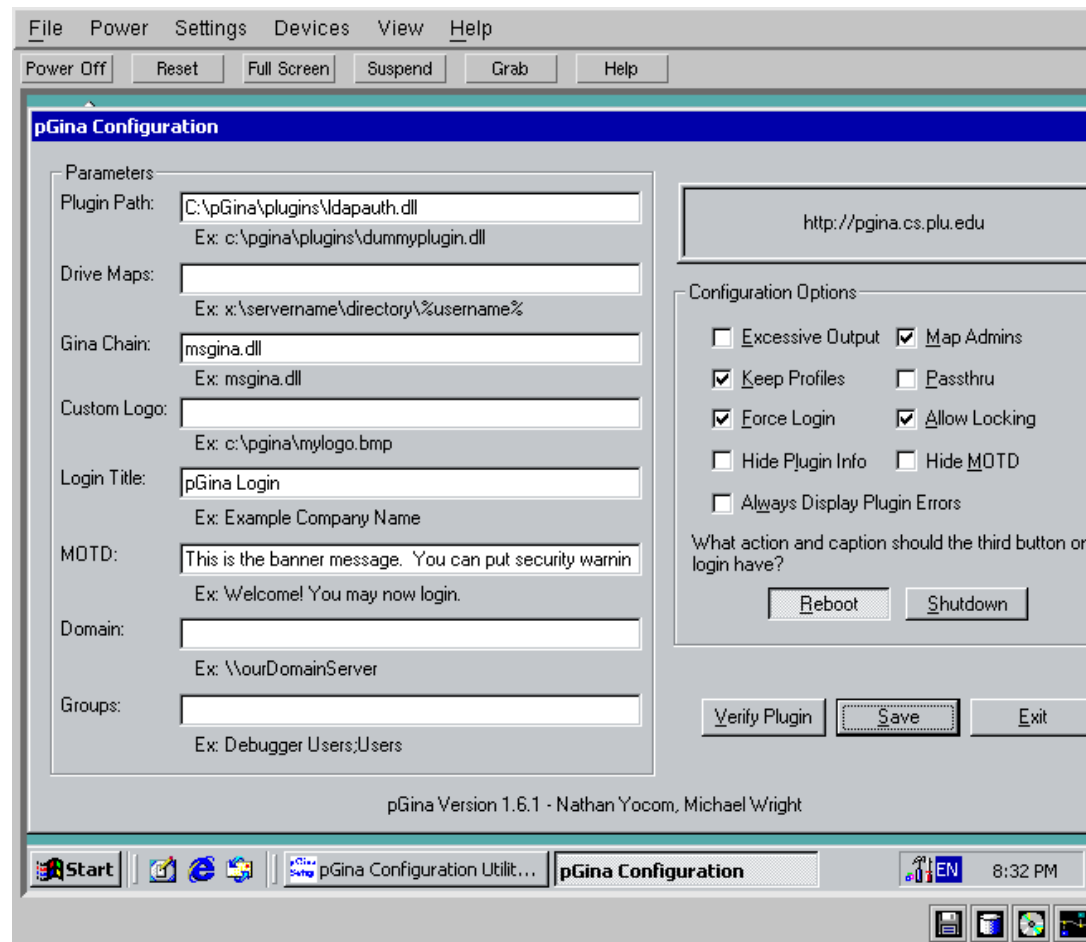
pGina Config

- Download and install pGina from <http://pgina.sf.net/>
- Install ldapauth.dll into c:
pgina
plugins
- Run regedit and create a new key called ldapauth in
HKey_Local_Machine
Software
pGina
 - ldapServer ldap.example.com
 - ldapPrepend uid=
 - ldapMethod 0
 - ldapContext0 ou=People,dc=example,dc=com

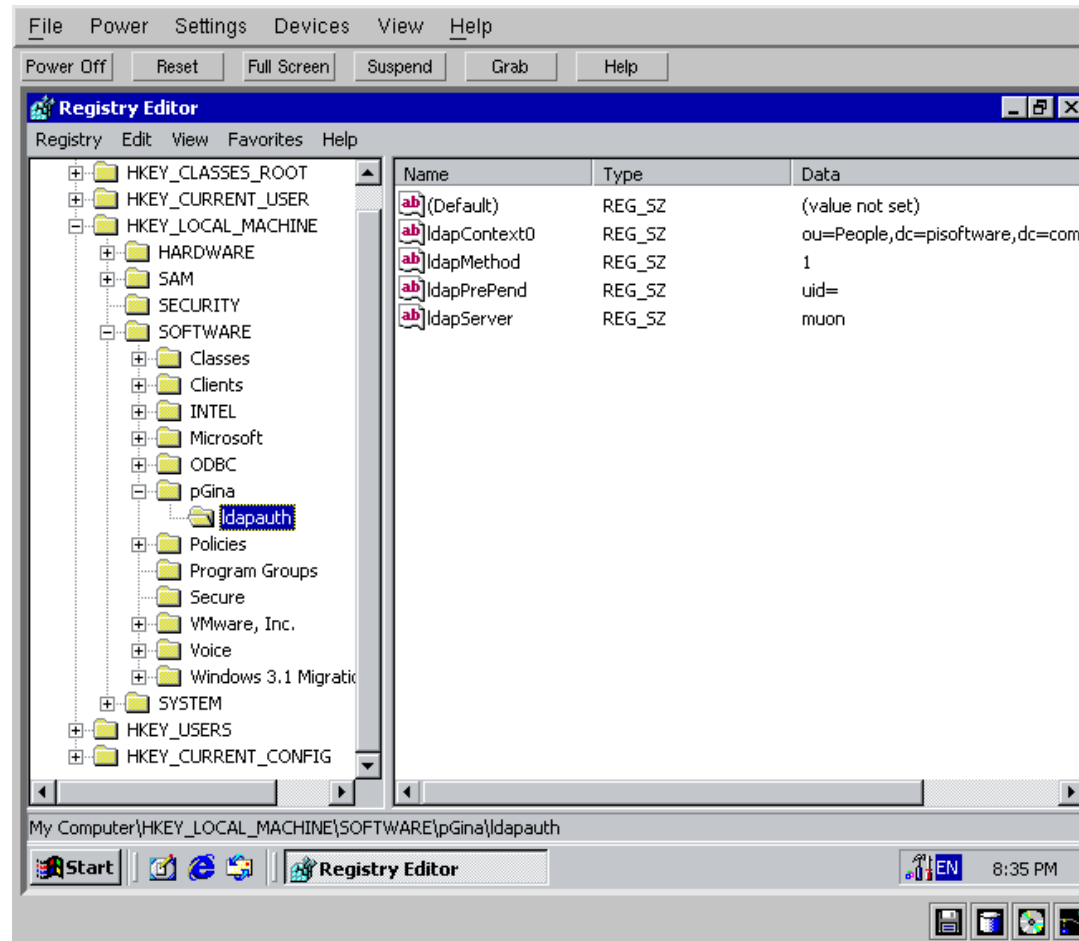
pGina Registry Entries

Key	Value
IdapMethod	1 = Multimap, 2 = search, 3 = map
useSSL	Use SSL
IdapPrePend	For map and multimap what it puts before the username
IdapAppend	For map, what goes after the username
IdapContext0-255	For multimap, different contexts to try
IdapAdminUsername	User to bind as
IdapAdminPassword	Password for IdapAdminUsername
userOK0-255	LDAP Group(s) user must be in
adminOK0-255	LDAP Group(s) user must be a member to be in Admin group

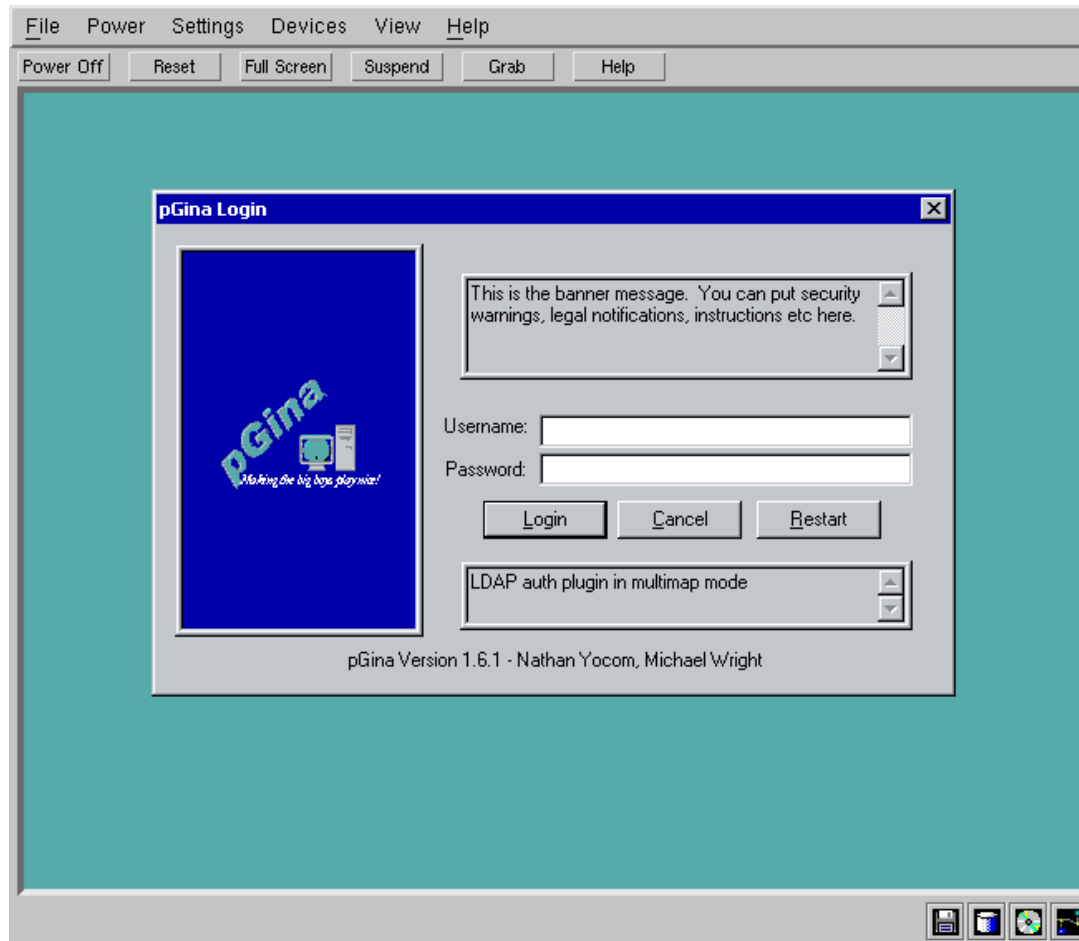
pGina Config



pGina ldapauth Regedit



pGina Login



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')dn1  
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	sendmailMTAClassName
CANONIFY_DOMAIN_FILE()	Canonify
EXPOSED_USER_FILE()	E
GENERIC_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
```

Exim

```
system_aliases:  
  driver = aliasfile  
  search_type = ldap  
  hide query = \  
    user = "cn=admin,dc=example,dc=com" \  
    pass = mypasswd \  
    ldap:/// \  
    cn=${quote_ldap:$local_part},dc=example,\  
    dc=com?mailbox?base?
```

Use ldapm for search_type to return multiple entries

Bind and LDAP

- Uses a sdb ldap backend
- Available from <http://www.venaas.no/ldap/bind-sdb/>
- Uses schema called dNSZone
- Build bind9 with the sdb backend, see the instructions included
- Add the following to named.conf:

```
zone "example.com" {  
    type master;  
    database "ldap ldap://ldap.example.com/ \\  
        dc=example,dc=com,o=DNS,dc=example,dc=com 172  
};
```

Bind and LDAP LDIF

```
dn: relativeDomainName=@, dc=example, dc=com, \
    o=DNS, dc=example, dc=com
objectClass: dNSZone
relativeDomainName: @
zoneName: example.com
dNSTTL: 3600
dNSClass: IN
sOARecord: ns.example.com. hostmaster.example.com.
    2002052201 3600 1800 604800 86400
nSRecord: ns.example.com.
nSRecord: ns.other-domain.com.
mXRecord: 10 mail.example.com.
mXRecord: 20 mail.other-domain.com.
```

Bind and LDAP LDIF cont

Equivalent to:

```
@      3600    IN      SOA      ns.example.com. hostmaster.example.com.
                               2002052201 3600 1800 604800 86400
                               NS       ns.example.com.
                               NS       ns.other-domain.com.
                               MX       10 mail.example.com.
                               MX       20 mail.other-domain.com.
```

Bind and LDAP LDIF cont

```
dn: relativeDomainName=my-hosta, dc=example,  
    dc=com, o=DNS, dc=example, dc=com  
objectClass: dNSZone  
relativeDomainName: my-hosta  
zoneName: example.com  
dNSTTL: 86400  
dNSClass: IN  
aRecord: 10.10.10.10  
mXRecord: 10 mail.example.com.  
mXRecord: 20 mail.other-domain.com.
```

Bind and LDAP LDIF

Equivalent to:

my-hosta

```
A      10.10.10.10
MX     10 mail.example.com.
MX     20 mail.other-domain.com.
```

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/ldap_auth \
    -b dc=yourdomain,dc=com ldap.domain.com
acl ldapauth proxy_auth REQUIRED
#acl ldapauth proxy_auth bmarshal pag
```
- Restart squid

Samba and winbind

- Install winbind from Samba
- Add the following to /etc/samba/smb.conf

```
security = domain
workgroup = DOMAIN
winbind separator = +
winbind cache time = 10
template shell = /bin/bash
template homedir = /home/%D/%U
winbind uid = 10000-20000
winbind gid = 10000-20000
password server = ip.ad.dr.es
wins server = ip.ad.dr.es
```

Samba and winbind cont

- /etc/nsswitch.conf (under debian)

```
passwd:          compat winbind
group:           compat winbind
shadow:          compat winbind
```

- Addition to /etc/pam.d/login

```
auth            sufficient pam_winbind.so
account         sufficient pam_winbind.so
session         sufficient pam_winbind.so
```

Samba and winbind cont

- Create a machine account for the workstation in Active Directory in Programs | Administrative Tools | Active Directory Users and Computers
- Join the domain by the following

```
$ sudo smbpasswd -j <domainname> \  
    -r <domainservername> -U Administrator
```

- Restart samba and winbind
- Login as DOMAIN+username

Samba and LDAP

- Install OpenLDAP 2.0.x
- Compile samba 2.2.3 or later with `--with-ldapsam`
- Download and install smbldap-tools from www.idealx.org
- Copy samba.schema into OpenLDAP schema dir
- Configure slapd.conf as below
- Import base.ldif
- Configure smb.conf as below
- As root, run:

```
# smbpasswd -w secret
# smbldap-useradd.pl -a -m \
                    -g 200 administrator
```

`\item Get the local system authenticating off LDAP`

Samba and LDAP - slapd.conf

```
# Schema and objectClass definitions
include      /etc/ldap/schema/core.schema
include      /etc/ldap/schema/cosine.schema
include      /etc/ldap/schema/nis.schema
include      /etc/ldap/schema/inetorgperson.schema
include      /etc/ldap/schema/misc.schema
include      /etc/ldap/schema/samba.schema
```

Samba and LDAP - slapd.conf cont

```
database            ldbm
# The base of your directory
suffix              "dc=gumby"
# Where the database file are physically stored
directory           "/var/lib/ldap"
# Root user
rootdn              "cn=Manager,dc=gumby"
rootpw              secret
# Indexing options
index objectClass,rid,uid, \
uidNumber,gidNumber,memberUID eq
index cn,mail,surname, \
givenname eq,subinitial
```

Samba and LDAP - smb.conf

```
[global]
  workgroup = GROUP
  security = user
  wins support = yes
  os level = 80
  domain master = true
  domain logons = yes
  local master = yes
  preferred master = true
  passwd program = /usr/local/sbin/ \
    smbldap-passwd.pl -o %u
```

Samba and LDAP - smb.conf cont

```
ldap suffix = dc=gumby
ldap admin dn = cn=Manager,dc=gumby
ldap port = 389
ldap server = 127.0.0.1
ldap ssl = No
add user script = /usr/local/sbin/ \
    smbldap-useradd.pl -w %u
domain admin group = @"Domain Admins"
logon path = \\%N\profiles\%u
logon drive = H:
logon home = \\homesrv\%u
logon script = logon.cmd
```

Samba and LDAP - smb.conf cont

```
[netlogon]
    comment = Network Logon Service
    path = /data/samba/netlogon
    guest ok = yes
    writable = no
    share modes = no
```

```
; share for storing user profiles
```

```
[profiles]
    path = /data/samba/profiles
    read only = no
    create mask = 0600
    directory mask = 0700
```

Samba and LDAP - Example Idif

```
dn: uid=administrator,ou=Users,dc=gumby
cn: administrator
sn: administrator
uid: administrator
gidNumber: 200
homeDirectory: /home/administrator
loginShell: /bin/bash
gecos: System User
description: System User
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: sambaAccount
pwdLastSet: 0
logonTime: 0
logoffTime: 2147483647
```

Samba and LDAP - Example Idif cont

```
kickoffTime: 2147483647
pwdCanChange: 0
pwdMustChange: 2147483647
displayName: System User
acctFlags: [UX]
primaryGroupID: 1401
homeDrive: H:
smbHome: \\muon\homes
profilePath: \\muon\profiles\administrator
scriptPath: administrator.cmd
lmPassword: 81CBCEA8A9AF93BBAAD3B435B51404EE
ntPassword: 561CBDAE13ED5ABD30AA94DDEB3CF52D
uidNumber: 0
rid: 1000
```

Samba and LDAP - Joining Domains

● WinNT

- Go to Control Panel | Network | Identification
- Click on Change, then choose Member Of Domain, and enter the domain
- Click on Create Computer Account in the Domain, then enter a domain admin username and password
- Reboot

Samba and LDAP - Joining Domains cont

● Win2k

- Right click on My Computers | Properties
- Go to Network Identification | Properties
- Click on Member Of Domain, and input the domain you want to join
- Enter a username / password combination for a domain administrator
- Reboot

Samba and LDAP - Joining Domains cont

● Win95

- Go to Control Panel | Network | Configuration
- Click on Client for Microsoft Network | Properties
- In the General tab, tick the box in Logon Validation for Logon to Windows NT Domain and put the domain in the Windows NT Domain textbox
- Go to Control Panel | Passwords | User Profiles
- Select the setting that says users can customize their own profiles
- Reboot

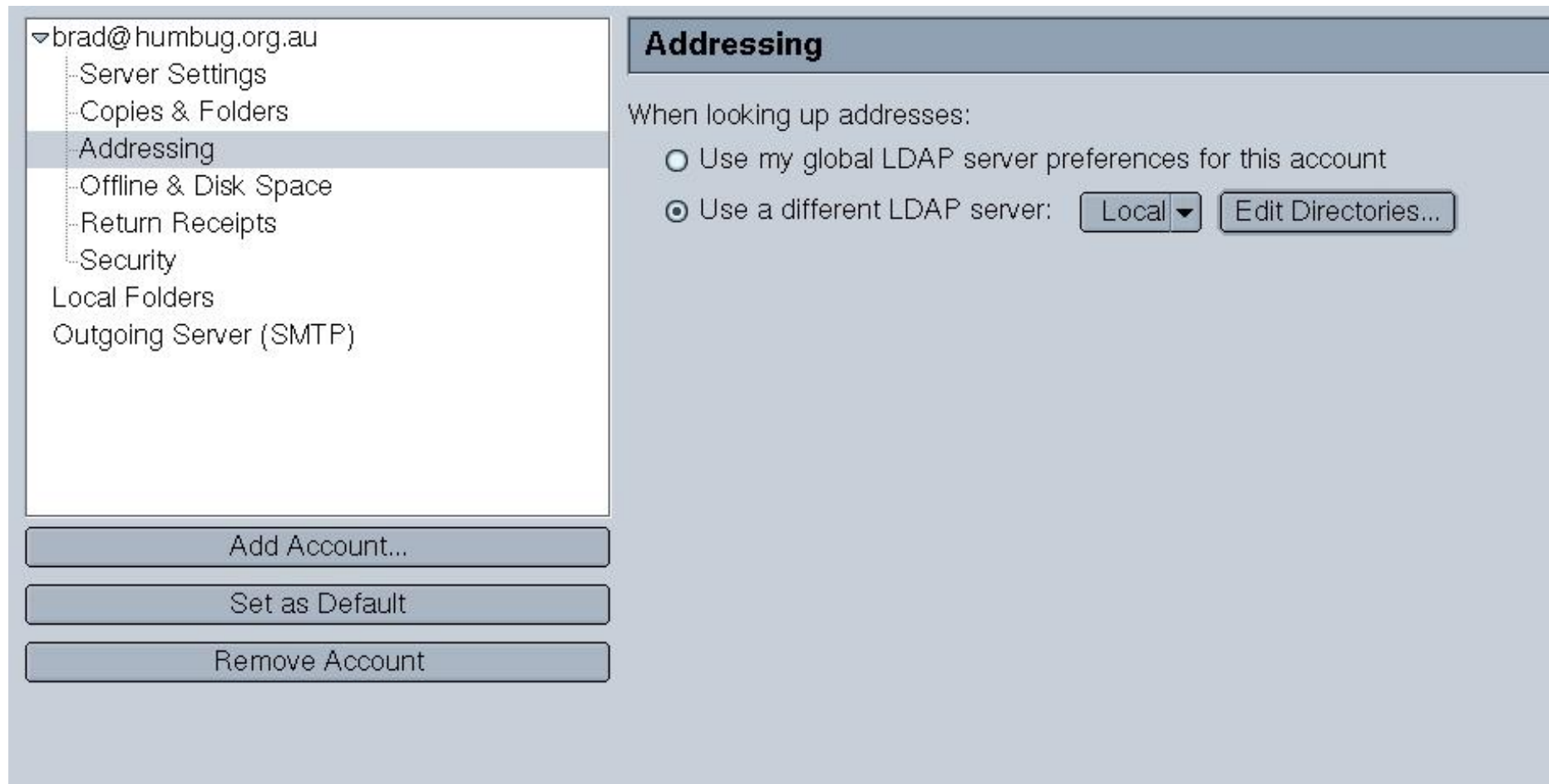
Netscape Addressbook and LDAP

Go to:

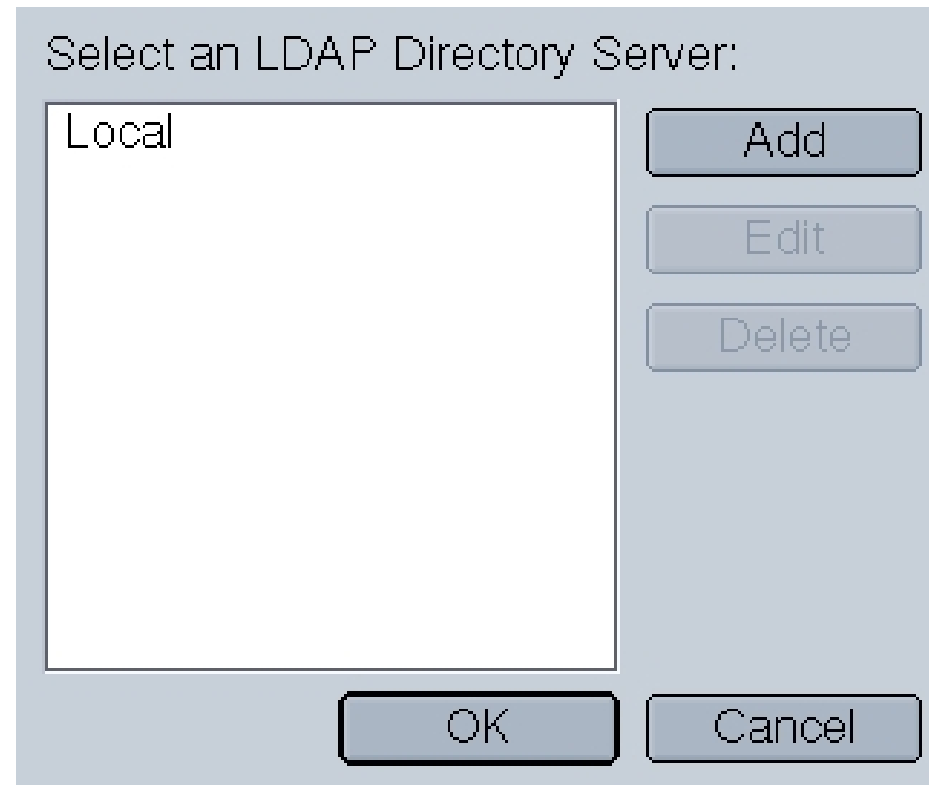
- Edit | Mail & Newsgroup Account Setup | Addressing
- Click on Edit Directories | Add
- Fill out hostname, base DN etc

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

Netscape Addressbook Adding



Netscape Addressbook Editing



Netscape Addressbook Editing cont

The screenshot shows the 'General' tab of the Netscape Addressbook Editing dialog box. The dialog has three tabs: 'General', 'Offline', and 'Advanced'. The 'General' tab is selected. It contains several text input fields and a checkbox. The fields are labeled 'Name:', 'Hostname:', 'Base DN:', 'Port number:', and 'Bind DN:'. The 'Name' field contains 'Local', 'Hostname' contains 'localhost', 'Base DN' contains 'dc=pisoftware,dc=com', and 'Port number' contains '389'. The 'Bind DN' field is empty. To the right of the 'Base DN' field is a 'Find' button. At the bottom left is a checkbox labeled 'Use secure connection (SSL)' which is unchecked. At the bottom right are three buttons: 'OK', 'Cancel', and 'Help'.

General Offline Advanced

Name: Local

Hostname: localhost

Base DN: dc=pisoftware,dc=com Find

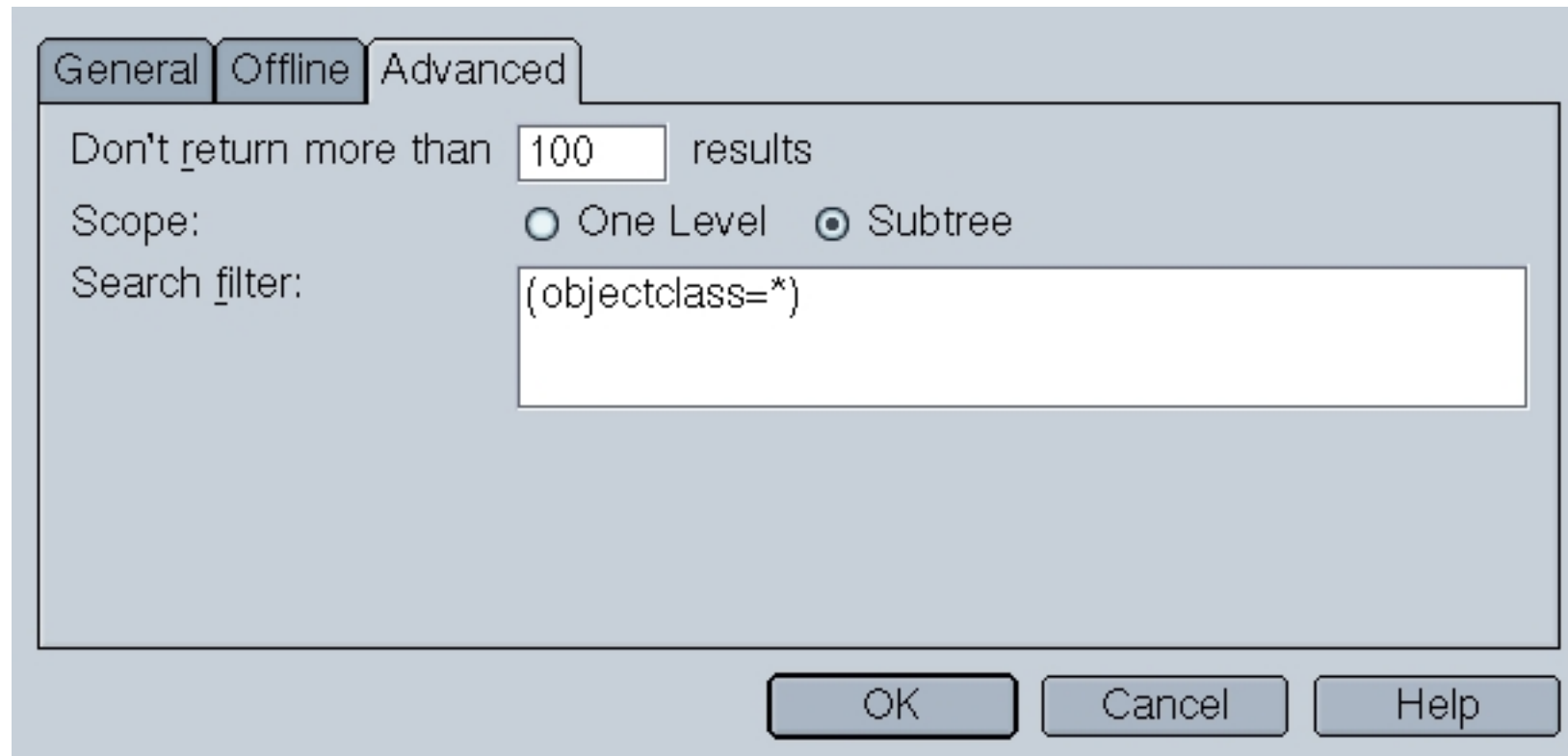
Port number: 389

Bind DN:

☐ Use secure connection (SSL)

OK Cancel Help

Netscape Addressbook Editing cont



The screenshot shows the 'Advanced' tab of the Netscape Addressbook editing window. It features three tabs at the top: 'General', 'Offline', and 'Advanced'. The 'Advanced' tab is active. Below the tabs, there are three settings: 'Don't return more than' followed by a text box containing '100' and the word 'results'; 'Scope:' followed by two radio buttons, 'One Level' (unselected) and 'Subtree' (selected); and 'Search filter:' followed by a large text box containing '(objectclass=*)'. At the bottom right, there are three buttons: 'OK', 'Cancel', and 'Help'.

General Offline Advanced

Don't return more than results

Scope: ☐ One Level ☒ Subtree

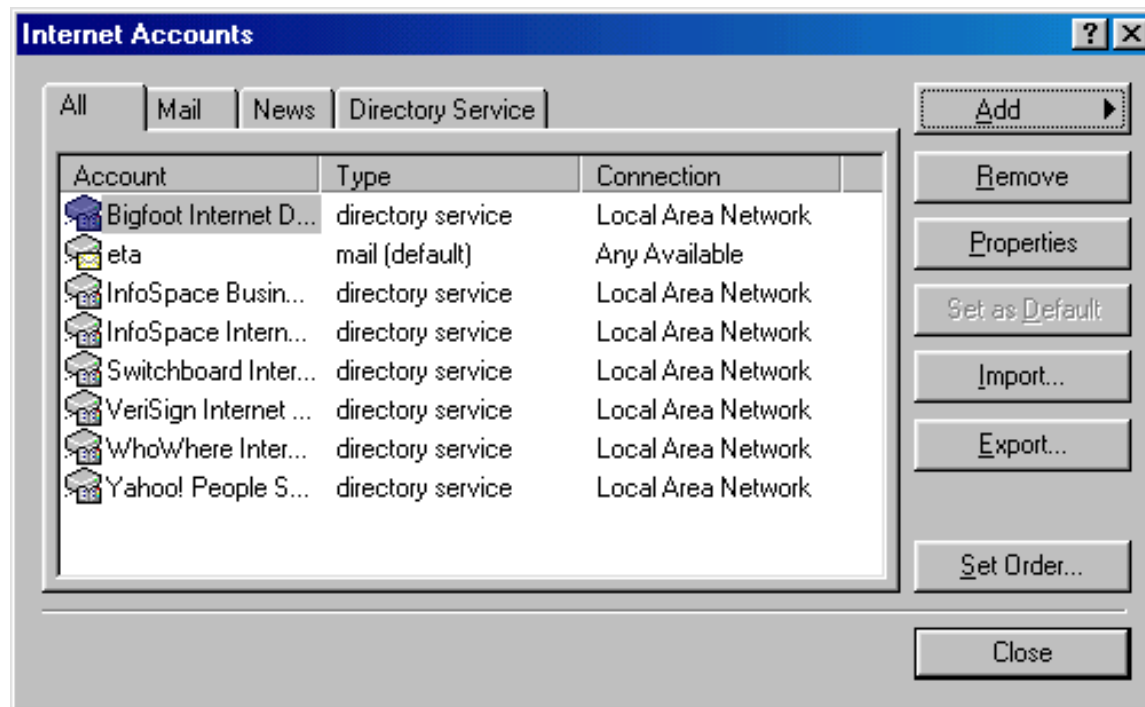
Search filter:

OK Cancel Help

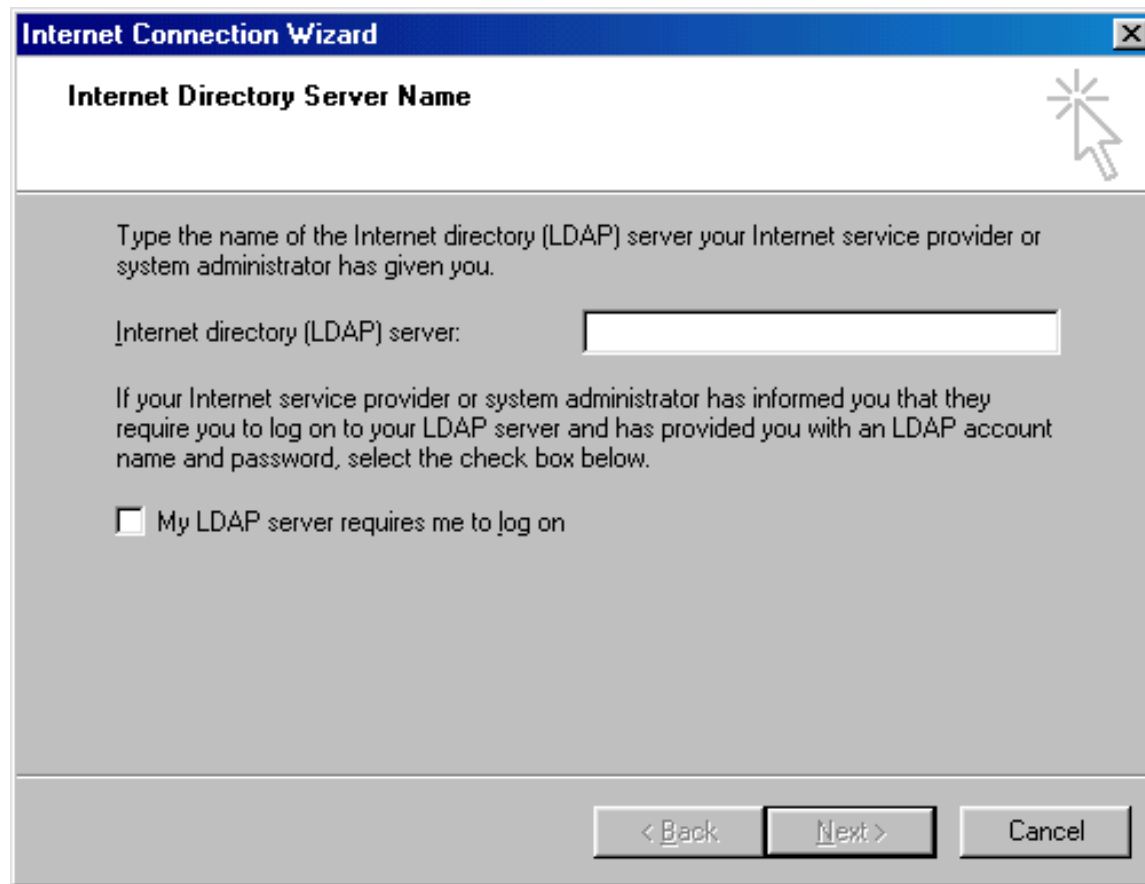
Outlook Express Addressbook

- Go to Tools | Accounts
- Click on Add | Directory Service
- Enter the hostname in the Internet Directory Server field, click on Next
- Click yes to using the directory to check addresses, then Next, then Finish
- Select the Account you just created, click on Properties
- Click on Advanced, then enter the search base

Outlook Express Directory



Outlook Express Directory



The screenshot shows a Windows-style dialog box titled "Internet Connection Wizard" with a standard close button (X) in the top right corner. The main heading inside the box is "Internet Directory Server Name". Below this heading, there is instructional text: "Type the name of the Internet directory (LDAP) server your Internet service provider or system administrator has given you." This is followed by a text input field with the label "Internet directory (LDAP) server:". Below the input field, there is more text: "If your Internet service provider or system administrator has informed you that they require you to log on to your LDAP server and has provided you with an LDAP account name and password, select the check box below." Underneath this text is a checkbox that is currently unchecked, with the label "My LDAP server requires me to log on". At the bottom of the dialog box, there are three buttons: "< Back", "Next >", and "Cancel". A mouse cursor is visible in the top right corner of the dialog box, pointing towards the close button.

Internet Connection Wizard

Internet Directory Server Name

Type the name of the Internet directory (LDAP) server your Internet service provider or system administrator has given you.

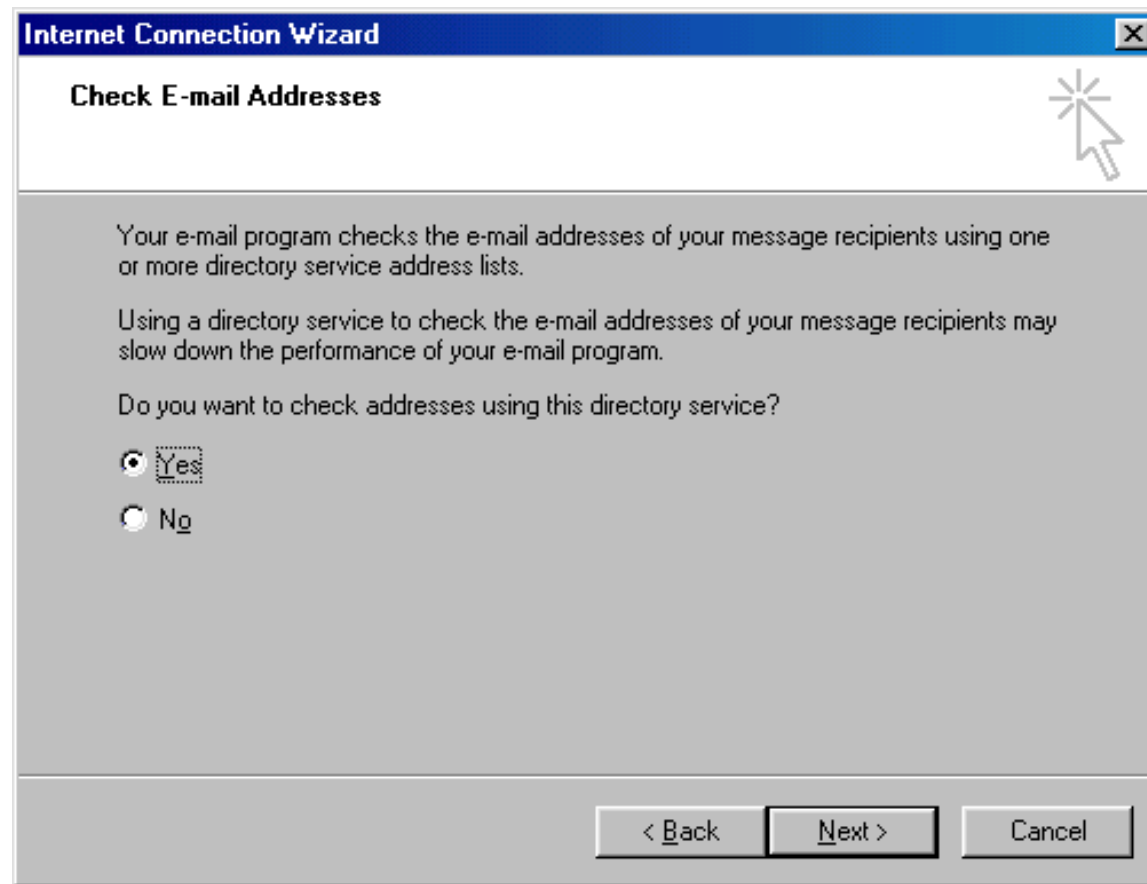
Internet directory (LDAP) server:

If your Internet service provider or system administrator has informed you that they require you to log on to your LDAP server and has provided you with an LDAP account name and password, select the check box below.

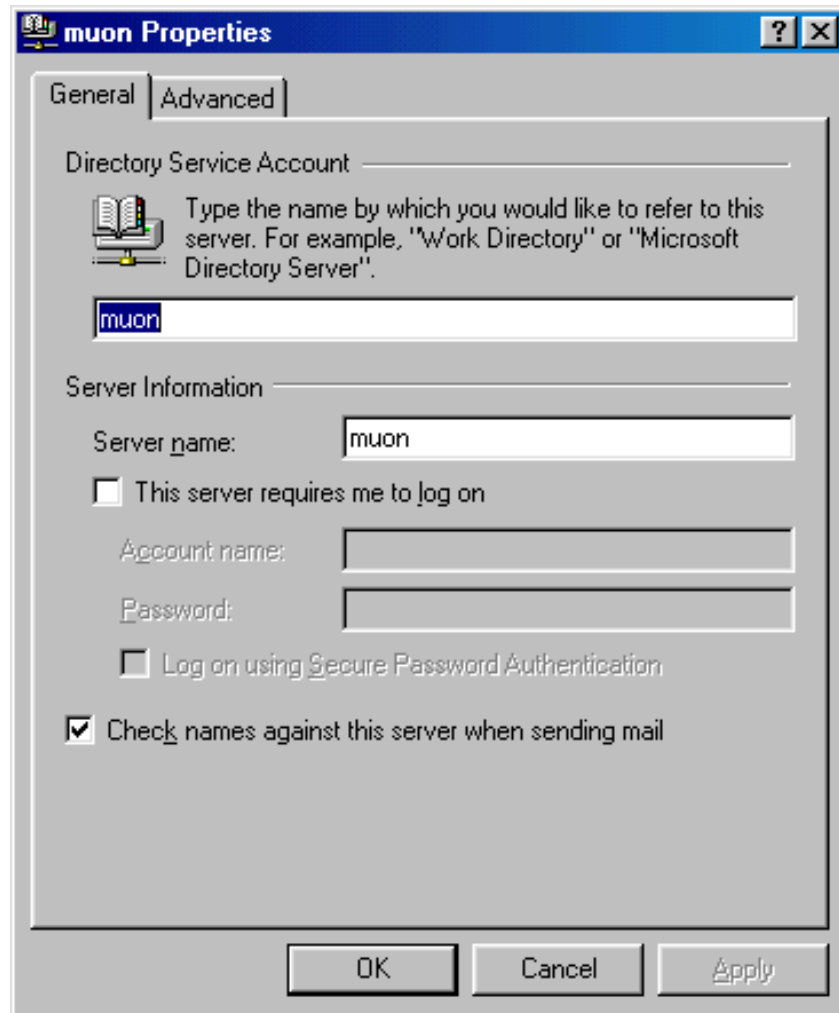
☐ My LDAP server requires me to log on

< Back Next > Cancel

Outlook Express Directory



Outlook Express Directory



The screenshot shows the 'muon Properties' dialog box with the 'General' tab selected. The 'Directory Service Account' section has a text box containing 'muon'. The 'Server Information' section includes a 'Server name' text box with 'muon', a checkbox for 'This server requires me to log on' which is unchecked, and fields for 'Account name' and 'Password'. There is also a checkbox for 'Log on using Secure Password Authentication' which is unchecked, and a checked checkbox for 'Check names against this server when sending mail'. At the bottom are 'OK', 'Cancel', and 'Apply' buttons.

muon Properties

General | Advanced

Directory Service Account

Type the name by which you would like to refer to this server. For example, "Work Directory" or "Microsoft Directory Server".

muon

Server Information

Server name: muon

☐ This server requires me to log on

Account name:

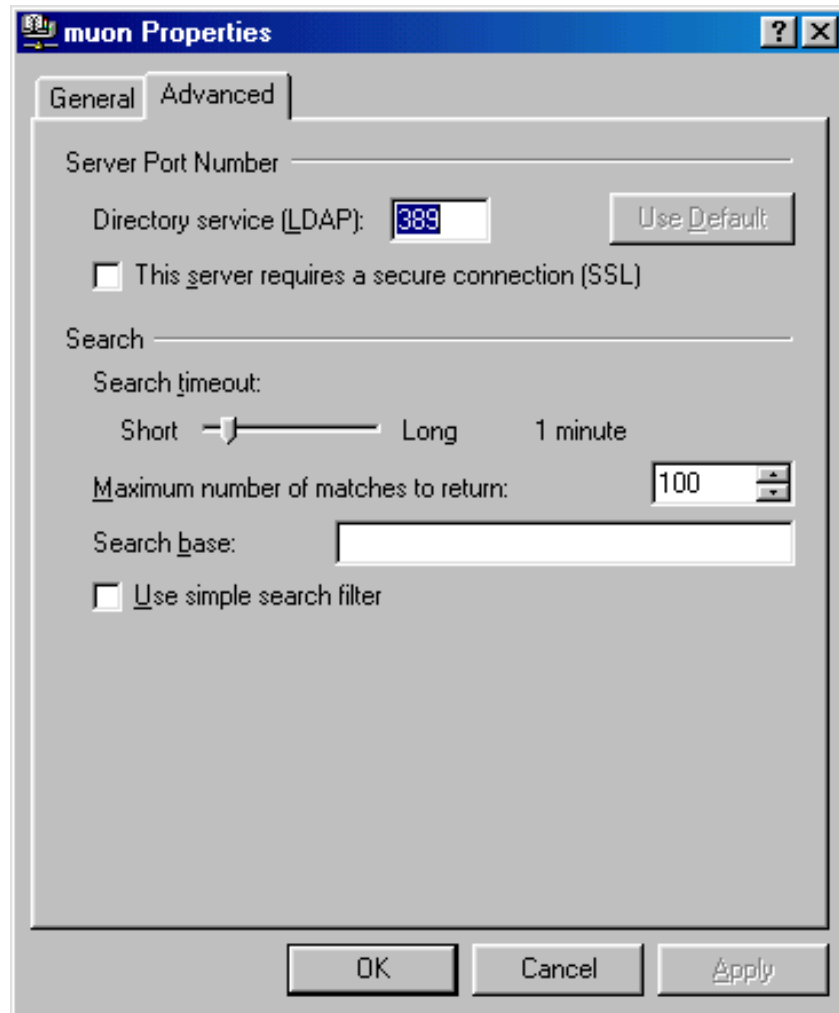
Password:

☐ Log on using Secure Password Authentication

☒ Check names against this server when sending mail

OK Cancel Apply

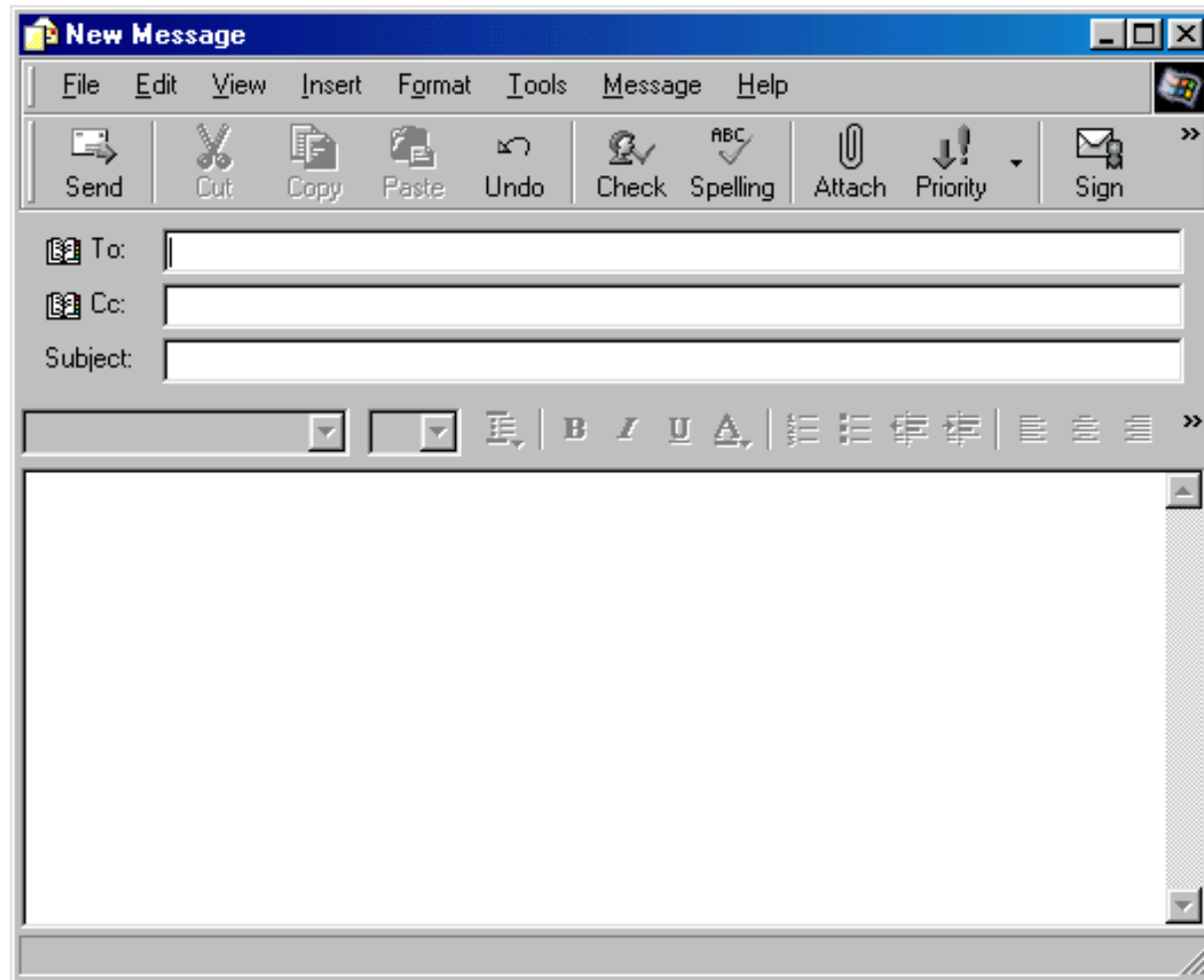
Outlook Express Directory



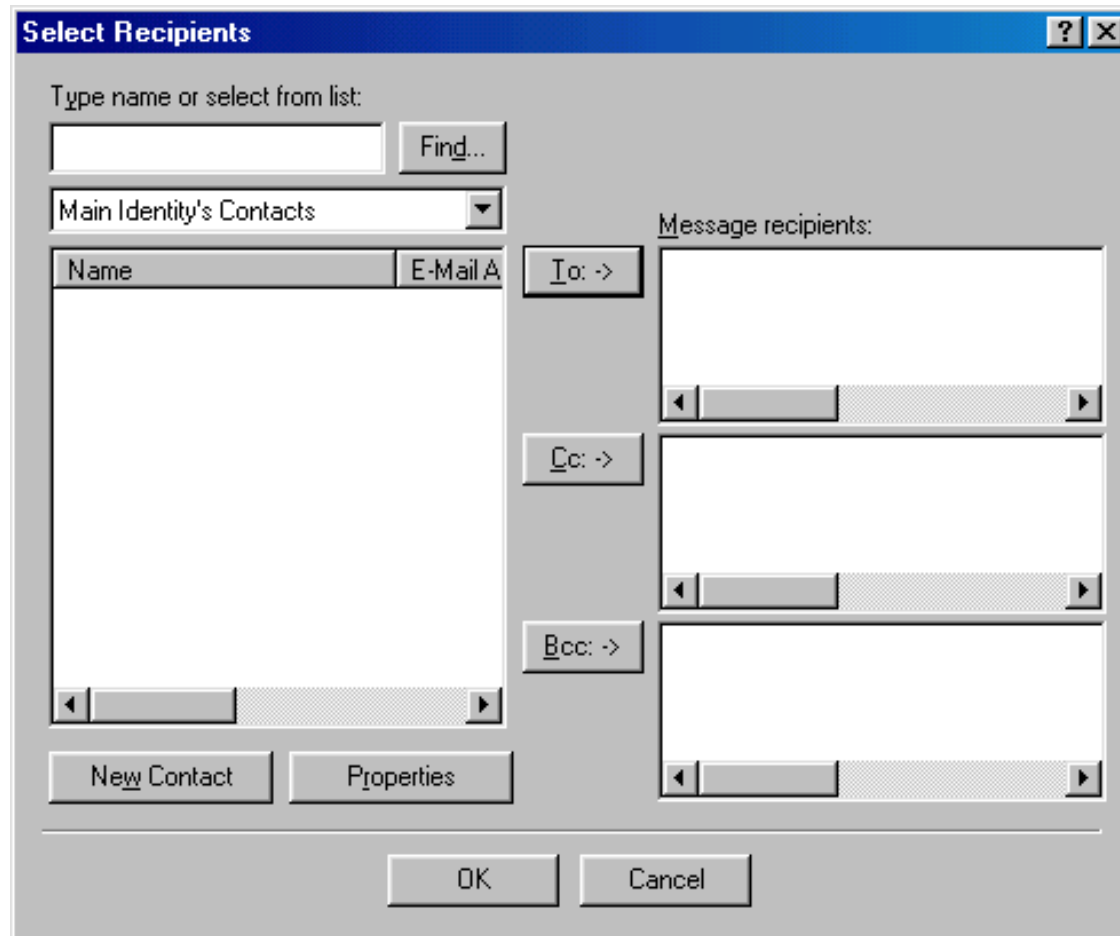
Outlook Express Addressbook - Composing

- Click on New Mail, then click on To | Find
- Pull down the Look in menu and select your directory
- Type in what who you're looking for in the Name field, then hit Find Now

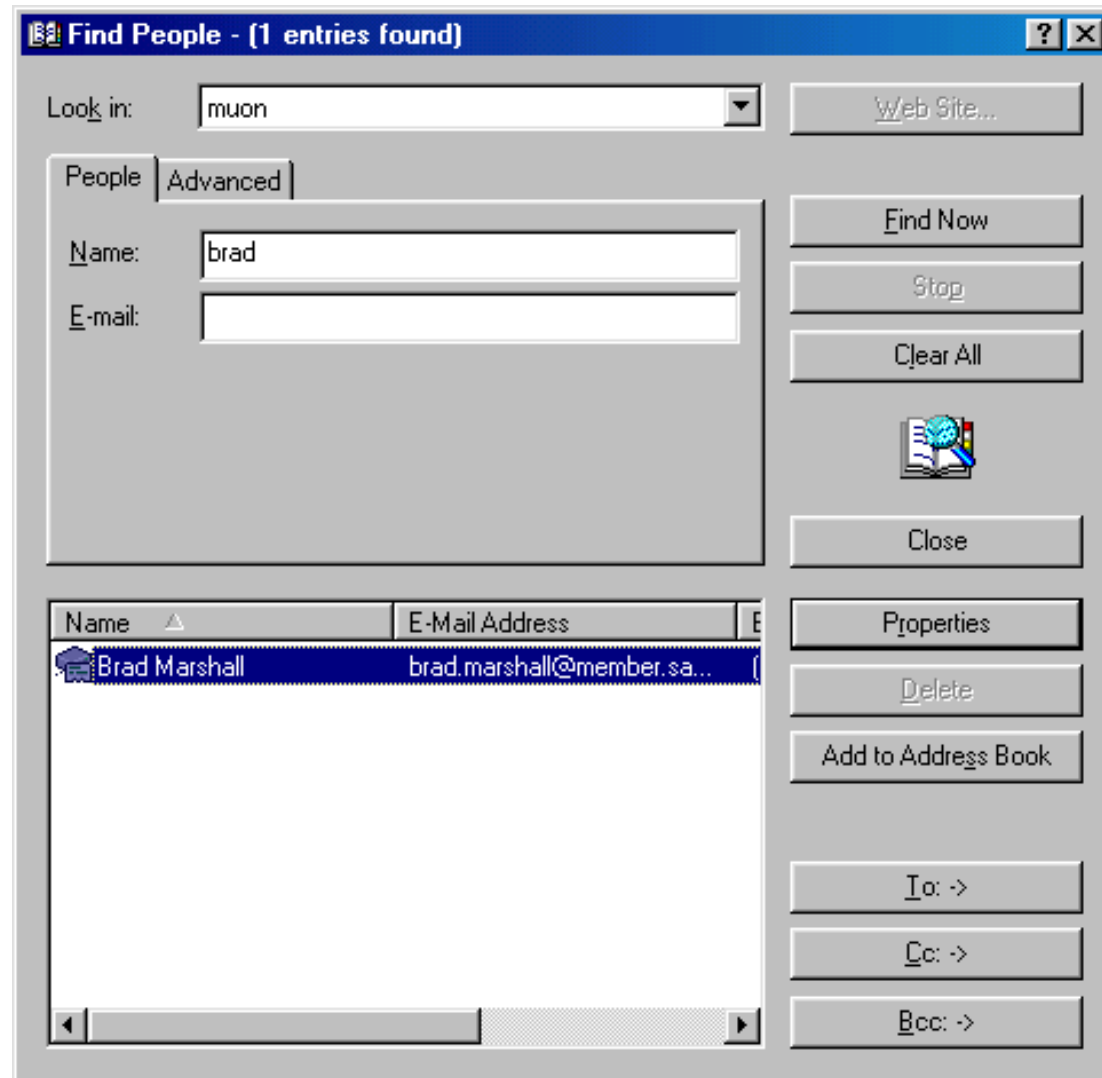
Outlook Express Addressbook - Composing



Outlook Express Addressbook - Composing



Outlook Express Addressbook - Composing



Address Book LDIF

```
dn: cn=Brad Marshall, ou=addressbook, dc=gumby
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: Brad Marshall
givenName: Brad
sn: Marshall
mail: brad.marshall@member.sage-au.org.au
```

Address Book LDIF cont

```
physicalDeliveryOfficeName: Plugged In Software  
postalAddress: PO BOX 1818  
l: Milton  
ou: addressbook  
st: Qld  
postalCode: 4064  
telephoneNumber: (07) 38762188  
facsimileTelephoneNumber: (07) 38764899  
pager: 1800-PAGER  
mobile: 1800-MOBILE  
homePhone: 1800-HOME
```

Active Directory and LDAP

Provides a directory for a Microsoft network:

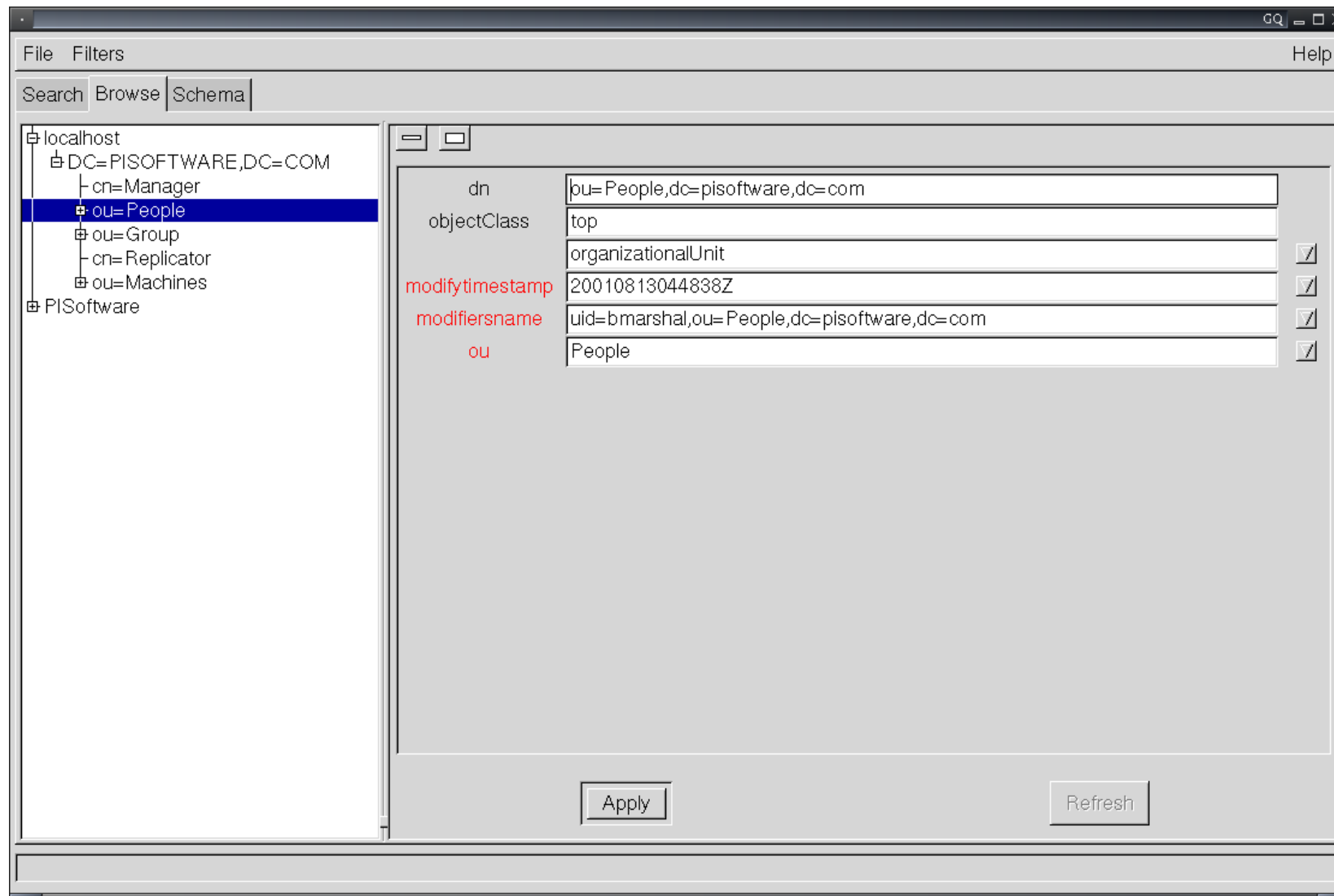
- Centrally manage
- Central security
- Central user administration
- Integrates with DNS
- Information replication
- Provides all the services a domain controller did

LDAP GUIs

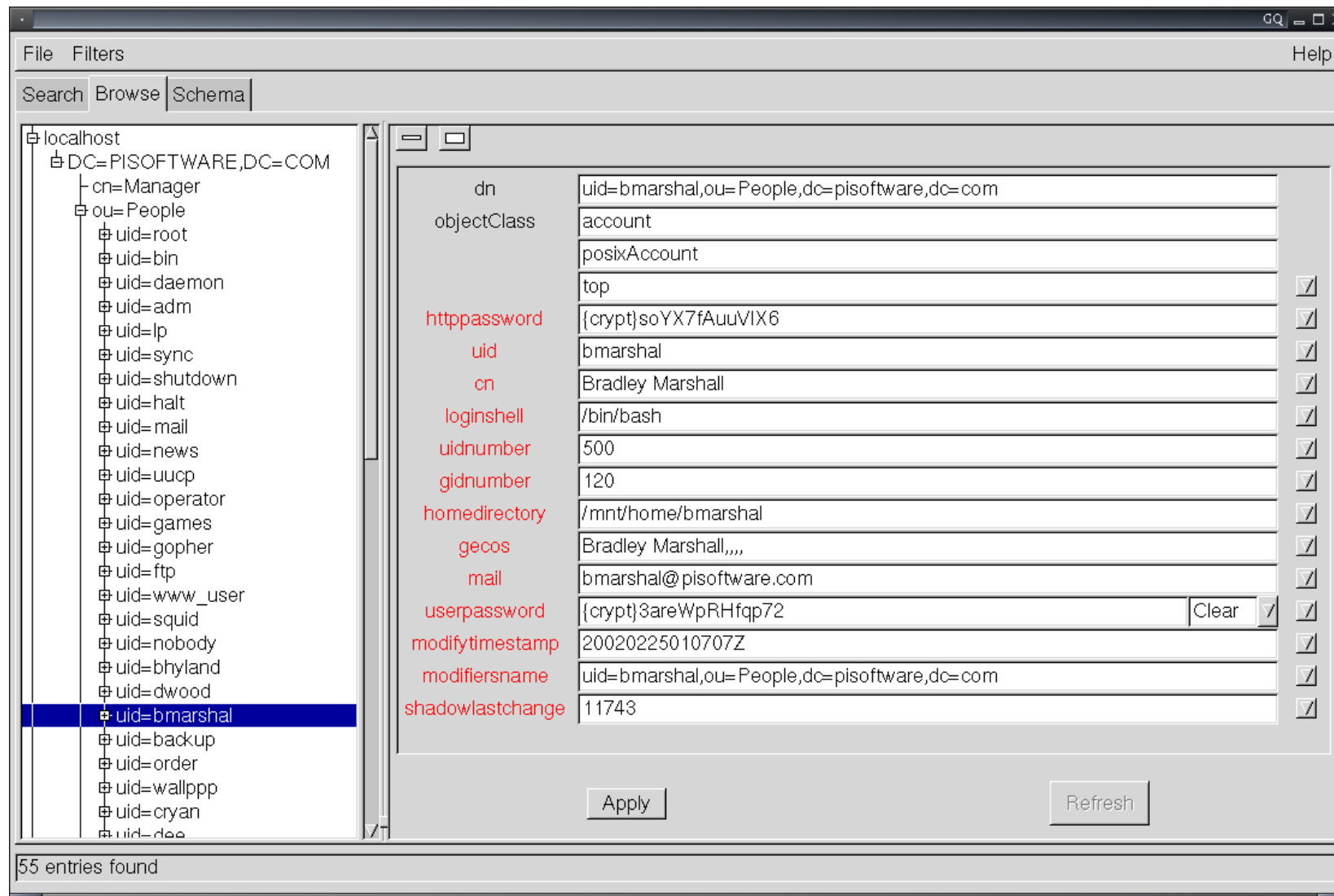
There are many LDAP administration GUIs, such as:

- 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, bddb etc

LDAP GUIs - GQ View People



LDAP GUIs - GQ View User



LDAP GUIs - GQ Search

File Filters Help

Search Browse Schema

filter localhost Find

DN	objectClass	modifytimestamp	modifiersname	uid	cn	loginshell
uid=squid,ou=People,dc=piSoftware,dc=com	account posixAccount	20010813044847	uid=bmarshal,ou=	squid	squid	/usr/local/bin/ppp
uid=nobody,ou=People,dc=piSoftware,dc=com	account posixAccount	20010813044847	uid=bmarshal,ou=	nobody	Nobody	/bin/bash
uid=bhyland,ou=People,dc=piSoftware,dc=com	account posixAccount	20010813044847	uid=bmarshal,ou=	bhyland	Bernadette Hylar	/bin/bash
uid=dwood,ou=People,dc=piSoftware,dc=com	account posixAccount	20011030013001	uid=bmarshal,ou=	dwood	David Wood	/bin/bash
uid=bmarshal,ou=People,dc=piSoftware,dc=com	account posixAccount	20020225010701	uid=bmarshal,ou=	bmarshal	Bradley Marshall	/bin/bash
uid=backup,ou=People,dc=piSoftware,dc=com	account posixAccount	20010813044849	uid=bmarshal,ou=	backup	Backup User	/bin/bash
uid=order,ou=People,dc=piSoftware,dc=com	account posixAccount	20010813044849	uid=bmarshal,ou=	order	Product Ordering	/bin/bash
uid=wallppp,ou=People,dc=piSoftware,dc=com	account posixAccount	20010813044850	uid=bmarshal,ou=	wallppp	Wall PPP	/etc/ppp/ppplogir
uid=cryan,ou=People,dc=piSoftware,dc=com	account posixAccount	20010813044850	uid=bmarshal,ou=	cryan	Chris Ryan	/bin/bash
uid=dee,ou=People,dc=piSoftware,dc=com	account posixAccount	20011220004714	uid=bmarshal,ou=	dee	Dee McGrath	/bin/bash
uid=javadoc,ou=People,dc=piSoftware,dc=com	account posixAccount	20010813044851	uid=bmarshal,ou=	javadoc	Javadoc	/bin/bash
uid=pppuser,ou=People,dc=piSoftware,dc=com	account posixAccount	20010813044851	uid=bmarshal,ou=	pppuser	PPP dialup user	/usr/local/bin/ppp
uid=helpdesk,ou=People,dc=piSoftware,dc=com	account posixAccount	20010813044851	uid=bmarshal,ou=	helpdesk	Helpdesk	/bin/bash
uid=jparker,ou=People,dc=piSoftware,dc=com	account posixAccount	20020405050947	uid=jparker,ou=F	jparker	Jason Parker	/bin/bash
uid=shill,ou=People,dc=piSoftware,dc=com	account posixAccount	20010813044851	uid=bmarshal,ou=	shill	Stefanie Hill	/bin/bash
uid=postgres,ou=People,dc=piSoftware,dc=com	account posixAccount	20010813044851	uid=dwood,ou=F	postgres	PostgreSQL Ser	/bin/bash
cn=test,ou=People,dc=piSoftware,dc=com	account posixAccount	20020124071321	uid=bmarshal,ou=	test	Plugged In Linux	/bin/false
cn=gdm,ou=People,dc=piSoftware,dc=com	account posixAccount	20011106003431	uid=jparker,ou=F	gdm	Gnome Display	/bin/false
cn=pag,ou=People,dc=piSoftware,dc=com	account posixAccount	20010813044910	uid=bmarshal,ou=	pag	Paul Gearon	/bin/bash
cn=ben,ou=People,dc=piSoftware,dc=com	account posixAccount	20020426005721	cn=ben,ou=Peop	ben	Ben Warren	/bin/bash
cn=archive,ou=People,dc=piSoftware,dc=com	account posixAccount	20010813044911	uid=bmarshal,ou=	archive	Plugged In Linux	/bin/bash
cn=amanda,ou=People,dc=piSoftware,dc=com	account posixAccount	20010813044911	uid=bmarshal,ou=	amanda	Plugged In Linux	/bin/bash
cn=nocol,ou=People,dc=piSoftware,dc=com	account posixAccount	20010813044911	uid=bmarshal,ou=	nocol	Plugged In Linux	/bin/bash
cn=netsaint,ou=People,dc=piSoftware,dc=com	account posixAccount	20010813044914	uid=bmarshal,ou=	netsaint	Netsaint User	/bin/bash
cn=david,ou=People,dc=piSoftware,dc=com	account posixAccount	20010813044914	uid=bmarshal,ou=	david	David Makepeac	/bin/bash
cn=keith,ou=People,dc=piSoftware,dc=com	account posixAccount	20010923001311	uid=jparker,ou=F	keith	Keith Ahern	/bin/bash
cn=tom,ou=People,dc=piSoftware,dc=com	account posixAccount	20010813044914	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

Add user...

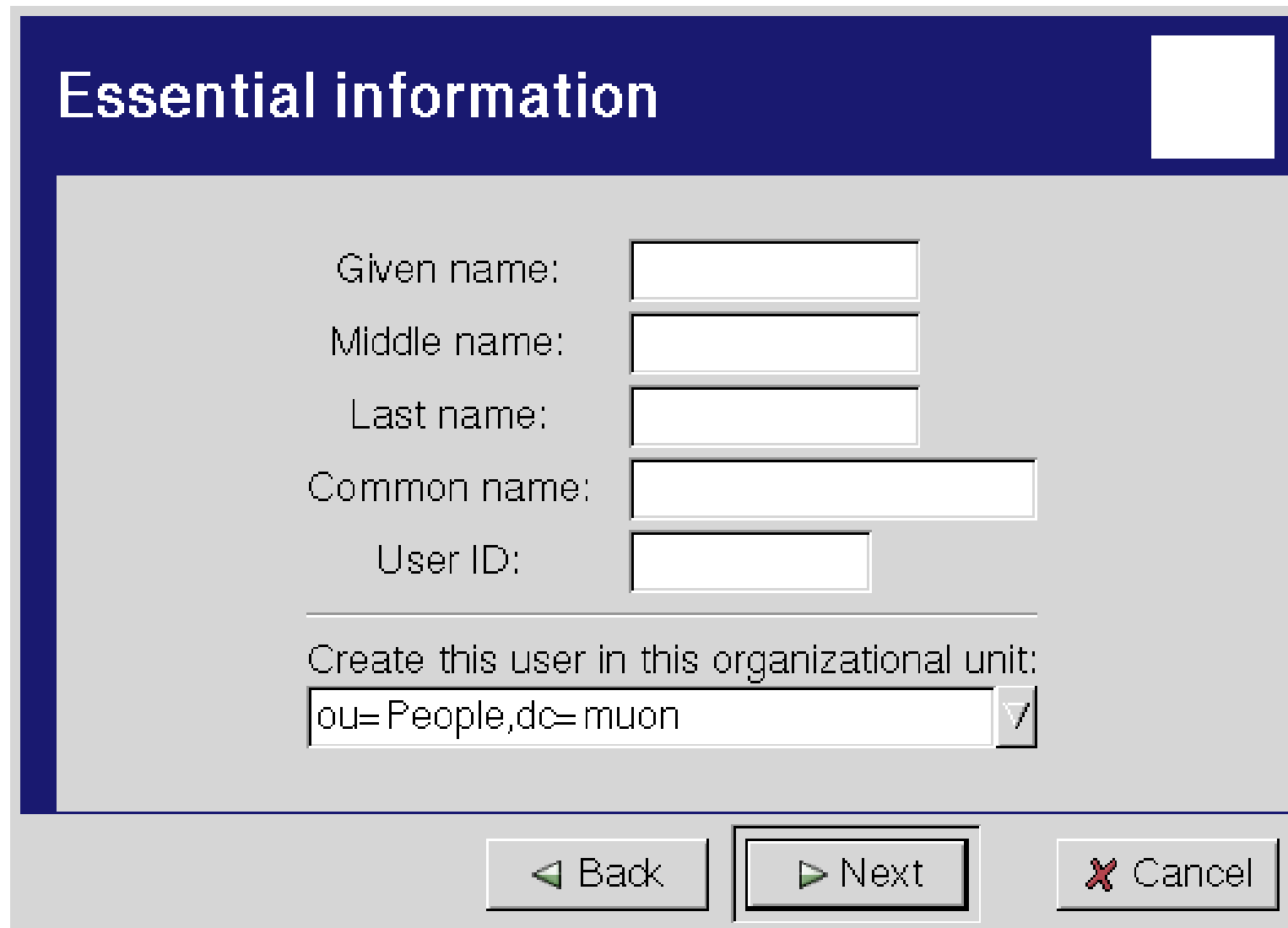
Remove user

◀ Back

▶ Next

✖ Cancel

LDAP GUIs - Directory Admin New User



The image shows a screenshot of a graphical user interface (GUI) window titled "Essential information". The window has a dark blue header bar with the title in white text. Below the header, the main area is light gray and contains several input fields for user information. The fields are labeled "Given name:", "Middle name:", "Last name:", "Common name:", and "User ID:". Each label is followed by a white rectangular input box. Below these fields, there is a label "Create this user in this organizational unit:" followed by a text box containing the value "ou= People,dc= muon". To the right of this text box is a small downward-pointing arrow icon. At the bottom of the window, there are three buttons: "Back" with a left-pointing arrow, "Next" with a right-pointing arrow, and "Cancel" with a red X icon. The "Next" button is highlighted with a black border.

Essential information

Given name:

Middle name:

Last name:

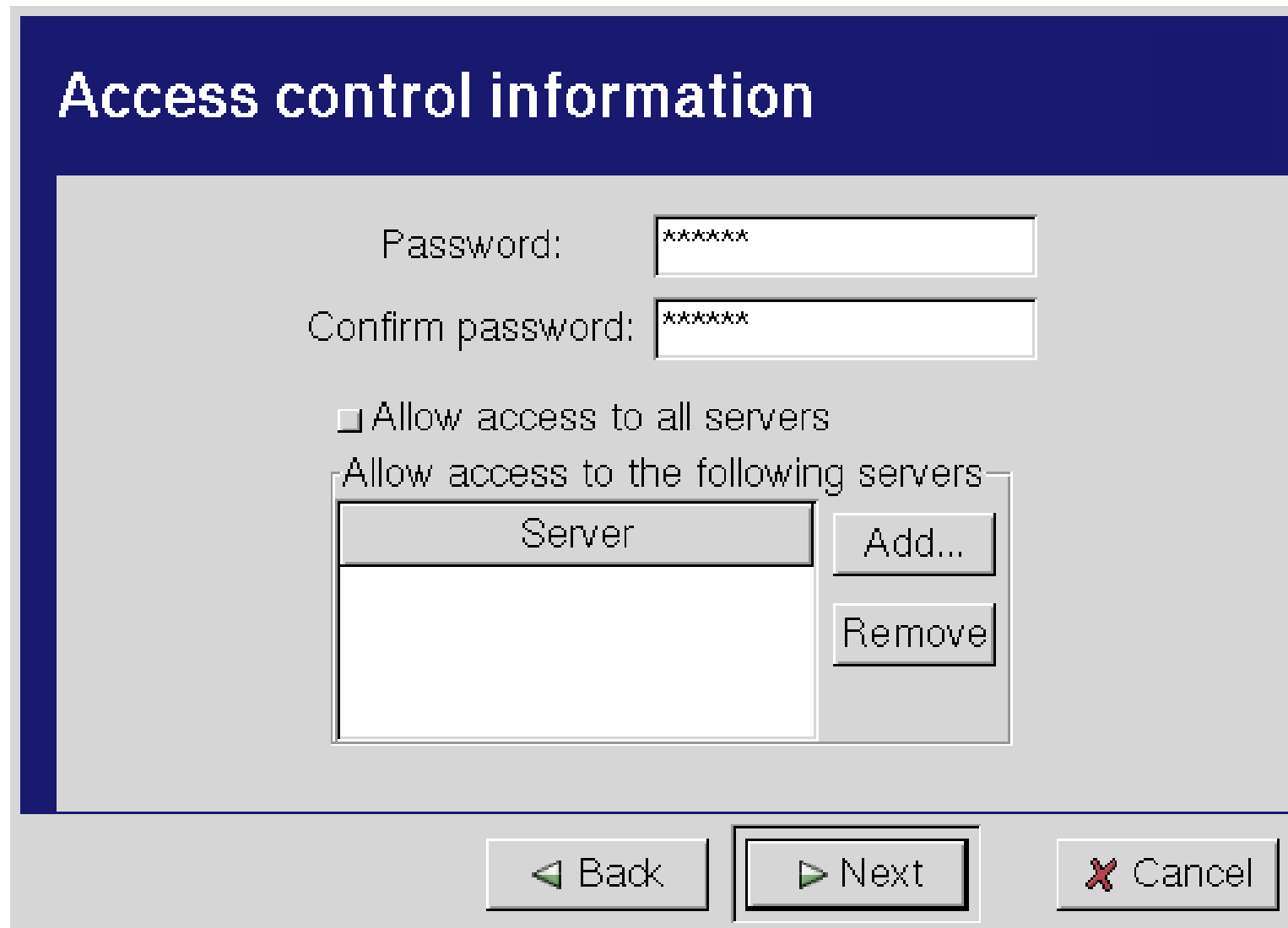
Common name:

User ID:

Create this user in this organizational unit:

LDAP GUIs - Directory Admin New User



The image shows a screenshot of a web-based LDAP GUI for creating a new user. The window has a dark blue header with the text "Access control information" in white. Below the header, the background is light gray. There are two password input fields, one for "Password:" and one for "Confirm password:", both containing masked text (asterisks). Below these fields is a checkbox labeled "Allow access to all servers". Underneath the checkbox is a section titled "Allow access to the following servers" which contains a table with a single header "Server" and an empty body. To the right of the table are two buttons: "Add..." and "Remove". At the bottom of the window, there are three buttons: "Back" with a left-pointing arrow, "Next" with a right-pointing arrow, and "Cancel" with a red X icon.

Access control information

Password:

Confirm password:

☐ Allow access to all servers

Allow access to the following servers

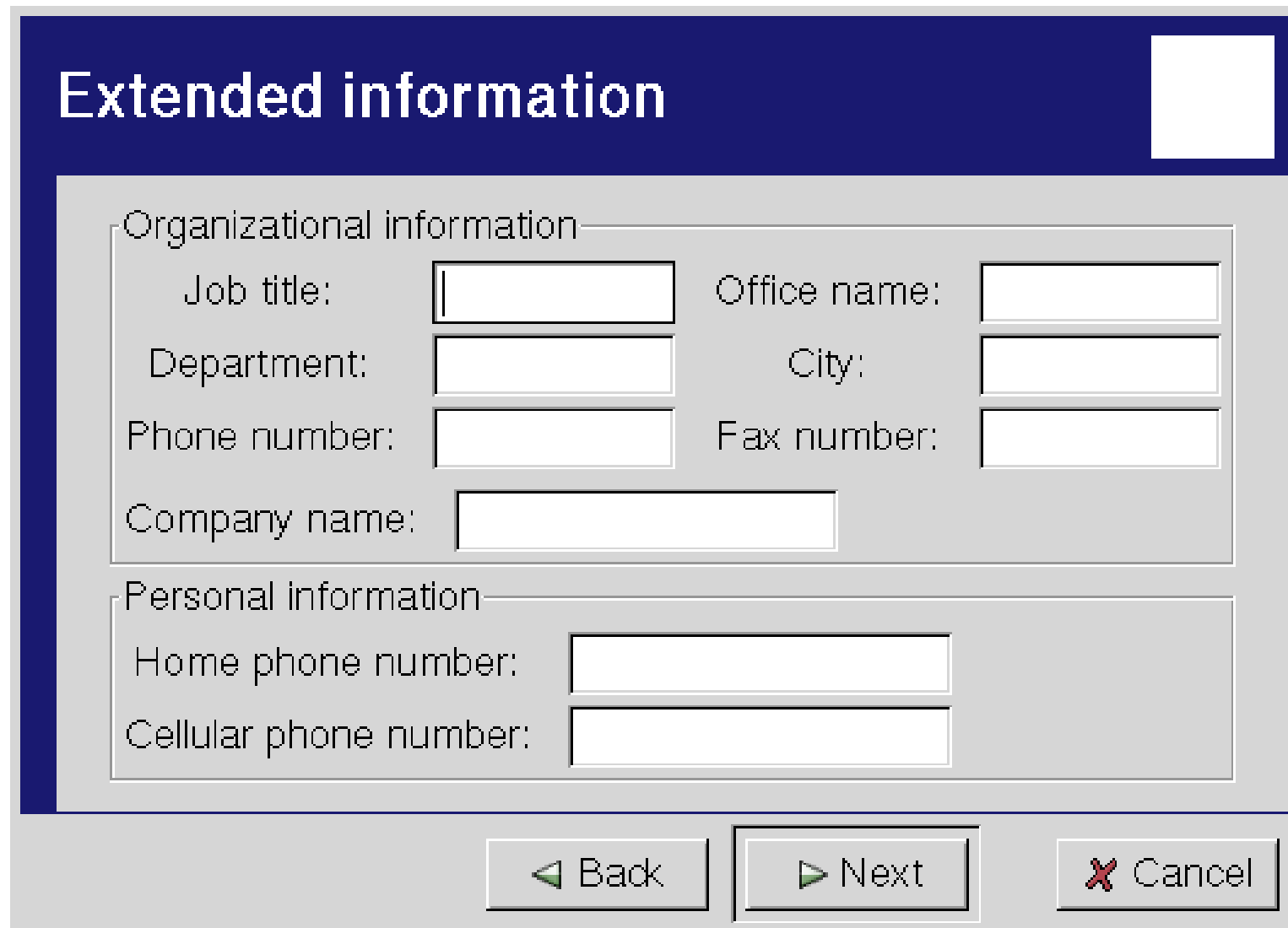
Server

Add...

Remove

◀ Back ▶ Next ✕ Cancel

LDAP GUIs - Directory Admin New User



The image shows a screenshot of a web-based LDAP GUI for creating a new user. The window has a dark blue header bar with the title "Extended information" in white text. To the right of the title is a small white square icon. Below the header, the form is divided into two main sections: "Organizational information" and "Personal information". The "Organizational information" section contains six input fields: "Job title:", "Office name:", "Department:", "City:", "Phone number:", and "Fax number:", each followed by a text box. Below these is a single wide input field for "Company name:". The "Personal information" section contains two input fields: "Home phone number:" and "Cellular phone number:", each followed by a text box. At the bottom of the form are three buttons: "Back" with a left-pointing arrow, "Next" with a right-pointing arrow, and "Cancel" with a red 'X' icon.

Extended information

Organizational information

Job title: Office name:

Department: City:

Phone number: Fax number:

Company name:

Personal information

Home phone number:

Cellular phone number:

 Back  Next  Cancel

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:

Back Next Cancel

LDAP GUIs - Directory Admin New User

UNIX account information

UNIX UID number: ☐ Automatic

Primary group:

Home directory:

Login shell:

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

Back Next Cancel

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] ▼

◀ Back ▶ Next ✕ Cancel

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;
```

PHP and LDAP - Binding

```
$ds=ldap_connect($hostname);  
if ($ds) {  
    ldap_set_option($ds, LDAP_OPT_PROTOCOL_VERSION, 3);  
  
    $r=ldap_bind($ds, $ldaprtn, $ldappass);  
  
    ldap_close($ds);  
}
```

PHP and LDAP - Searching

```
$sr=ldap_search($ds,"dc=example,  
    dc=com", "objectclass=*");  
$info = ldap_get_entries($ds, $sr);  
for ($i=0; $i<$info["count"]; $i++) {  
    echo "dn is: ". $info[$i]["dn"] . "<br>";  
    echo "first objectclass entry is: ".  
        $info[$i]["objectclass"][0] . "<br>";  
}
```

See <http://www.php.net/manual/en/ref.ldap.php>

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



<http://samba.idealx.org/dist/samba-ldap-howto.pdf>