Comparison of LDAP Clients under Practical Conditions

Admin's Little Helpers

Without a well planned management concept and suitable administration tools any LDAP directory is surely heading for chaos. This article investigates the options for optimizing administrative approaches using only freeware tools. VOLKER SCHWABEROW

After setting up an LDAP directory in a productive environment one of the first questions that arises will probably concern options for effective management. As a system administrator you may soon find yourself demoted to the personnel department's gopher when user phone extensions or addresses need to be modified. That is definitely not what you had in mind when you applied for the admin job – and probably not what your employer had in mind either.

You will often find admins purchasing expensive toolkits in order to cope with the administration of a directory service, but why buy tools if you are working with an Open Source directory service such as OpenLDAP [1]? This is the question we will attempting to answer in this article using several Open Source solutions as examples.

Toolkits must be able to provide several fundamental capabilities:

- Delegation: The tools will need the ability to delegate management of the directory service, and any tasks this involves to administrative roles.
- Usability: The toolkit must be useable by inexperienced users. You should not need to be an expert to change a record.
- Management: The solution must be easily maintainable for the sysad and provide understandable core functions.

You will also need to decide who will manage what records in the LDAP directory. One possibility would be to allow the users to keep their own records up to date. However, the administrator could just as easily assign this task to trained staff, possibly from the personnel department. Whatever approach you take, you will need to ensure that the procedures you implement allow you to maintain consistency for new and existing data.

A toolkit is no improvement if it allows users to let the LDAP directory to go haywire, preventing even the administrator from keeping track of the status of the records. And this is why the introduction of an LDAP toolkit should be modelled on your administrative processes. If not, the introduction of an interface for data maintenance is doomed to failure. It should be obvious that models of this kind do not lend themselves to quick and dirty implementations.

Web Interfaces

The Web interface is a typical method for administration, the advantage being that users can access their own records, no matter what platform they use, Web interfaces provide a similar level of functionality to native applications, since the functionality of a fat client can be implemented almost entirely using Web programming languages.

Gonicus – a company that rose like a tiny phoenix from the ashes of ID-Pro – recently placed a tool called Gosa on their website [3]. Gosa is a Web application based on the PHP [2] programming language.

Although you can transfer the administration of multiple network services to a directory service, you may not be able to use a common interface. This is the gap that Gosa attempts to fill.

Gosa was developed as an add-on to Gonicus' own thin client project, Goto. Gosa's strength is user and group management for Posix accounts, Samba, Squid and Qmail. If you are not using Samba or Qmail in a given network, it does not make much sense to install and run Gosa. Additionally, an *nss_ldap* server link should be in place.

Installation and Configuration of Gosa

After downloading the current Gosa package from the FTP server, you will probably want to use the */opt* directory to expand the package. You will find a tar-gzip file containing schema for Gonicus' own applications in */opt/gosa/contrib*. You will also need to expand the schema and copy the *Gonicus* directory that this action creates to */etc/openldap/schema*.

The *qmail .schema* file contains the schema for the Qmail-LDAP interface that also needs to be moved to the */etc/openldap/schema* directory.

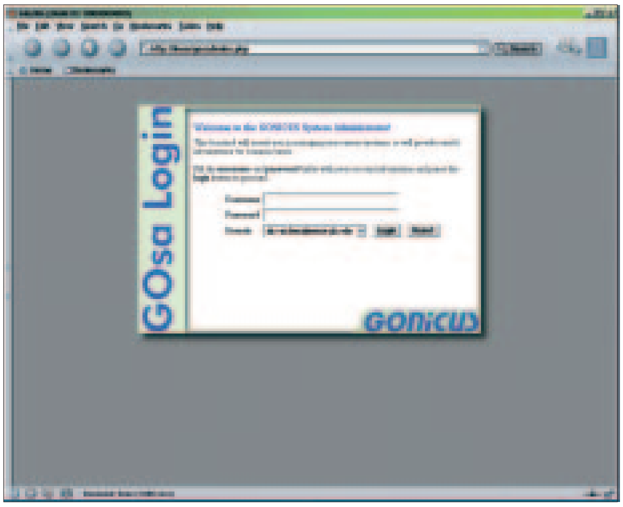


Figure 1: Gonicus' freeware tool, Gosa, immediately following installation

At this point the **.schema* files need to be imported into your OpenLDAP server's schema. To do so, use an *include* statement in the *slapd .conf* file. Listing 1 shows you the order in which to add the schemas. If you are adding additional schemas to your LDAP server, you will probably want to change this order.

The next step is to define the Access Control Lists recommended for Gosa (see Listing 2) in the *slapd.conf* file. Pay attention to the comments in the lines starting with hash signs. The ACLs must reference the distinguishing name of the LDAP admin account.

Finally, you will need to set up a Posix account for the administrator of your directory. If you have already set up some Posix accounts on your directory server, you can simply point to the distinguishing name of an existing account. If not, use a short LDIF file to set up the Posix account on your directory server, as follows:

```
dn: uid=myadmin,dc=myname,dc=com
objectClass: top
objectClass: posixAccount
homeDirectory: /root
userPassword: secret
loginShell: /bin/false
uid: admin
cn: admin
uidNumber: 501
gidNumber: 501
```

Use *ldapadd -x -D "cn=Manager,dc*=domain,*dc*=com" *-W -f* filename.*ldif* to add this to the directory and complete the configuration steps for your directory server.

Our last step is to install the PHP scripts that will install Gosa on the web

Listing 1: Schema File Order

```
include     /etc/openldap/schema/core.schema
include     /etc/openldap/schema/cosine.schema
include
/etc/openldap/schema/inetorgperson.schema
include     /etc/openldap/schema/nis.schema
include     /etc/openldap/schema/misc.schema
include     /etc/openldap/schema/qmail.schema
include
/etc/openldap/schema/gonicus/gohard.schema
include
/etc/openldap/schema/gonicus/goto.schema
include
/etc/openldap/schema/gonicus/goaccount.schema
include
/etc/openldap/schema/gonicus/gofirewall.schema
include
/etc/openldap/schema/gonicus/gofax.schema
```

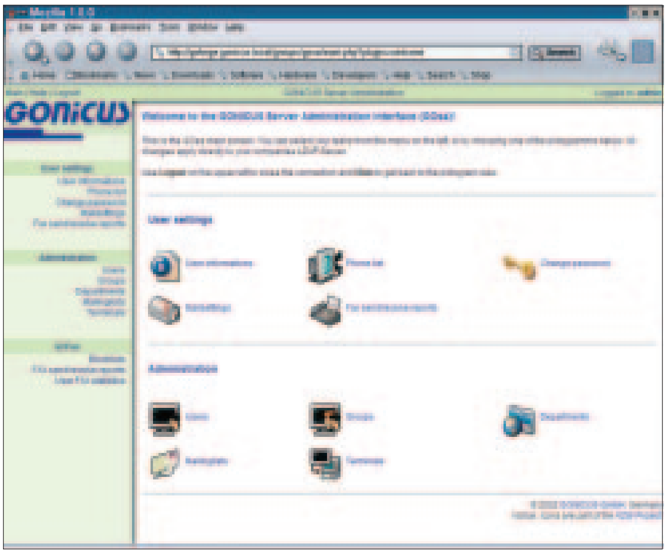


Figure 2: The Gosa user interface from the viewpoint of the administrator

server. You can either create a relative link in the Apache configuration file, *httpd.conf*, or copy your Gosa root to the root directory of your web server. Now, Gosa should really be ready to go at this point – if it wasn't for those pesky bugs and issues.

Listing 2: ACLs for Gosa

```
# DN must reference the DN of the Directory Management
# Account.
access to attribute=deliveryMode
       by dn="cn=Manager,dc=myname,dc=com" write
       by self write
       by * read

# DN must reference the DN of the Directory Management
# Account.
access to attribute=mailForwardingAddress
       by dn="cn=Manager,dc= myname,dc=com" write
       by self write
       by * read

# DN must reference the DN of the Directory Management
# Account
access to attribute=mailReplyText
       by dn="cn=Manager,dc= myname,dc=com" write
       by self write
       by * read

# The DN can point to an existing
# POSIX object in this case, Admin for example.
# This DN is used to manage the Gosa solution itself.
access to *
       by dn="uid=myadmin,dc= myname,dc=com" write
       by * read
```

First Impressions Spoilt by Weaknesses

For example, when you modify a user, the corresponding LDAP record is first deleted, and then reinstated. This is by no means a perfect solution, because an error could mean losing the account entirely. The PHP programming language actually includes a statement for just such a task, *ldap_modify*, but Gosa does not use it.

The Gosa developers' solution for checking privileges is also slightly cumbersome. To check the role assigned to a user, the program attempts to add a user account called *admincheck* to the directory when the user logs on. If this works, the user is an administrator from Gosa's point of view. If it does not work, possibly because the account already exists, you may find your admin account being degraded to a normal user – this is an unnecessarily complicated and dangerous system. Conclusion: Gosa is headed in the right direction, but the project itself is tied to other projects under development by Gonicus.

Users who prefer modular software may be disappointed by this product, as it can hardly be classified as a stand-alone toolkit. There are several issues involving the Gosa installation procedure which we were only able to resolve by modifying the PHP scripts.

Work is still in progress on improving the universal appeal of the project. According to Linux Magazine sources, a new version is due to be released shortly and the folks at Gonicus claim that it will be easier to adapt to third-party tasks than the current version.

Using Webmin Plugins to organize LDAP

The well-known Webmin [4] tool offers a variety of administrative functions for Linux servers. Several third-party modules are available to enhance the functionality of Webmin and one of them is the LDAP Users Admin Module [5]. The *ldap-users* module is easy to install

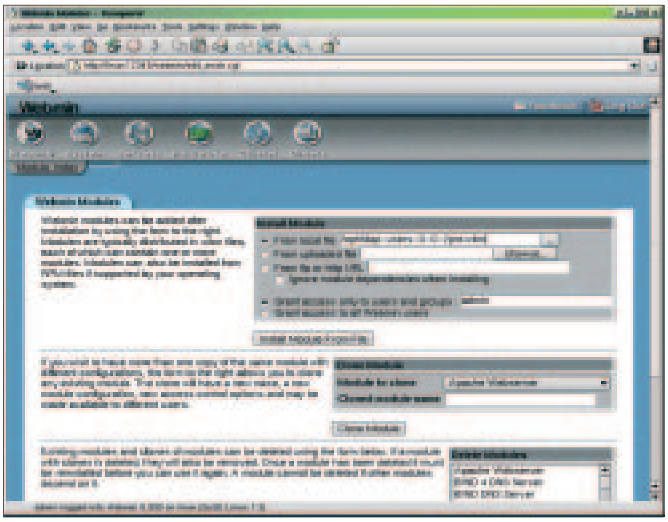


Figure 3: It is easy to install an additional Webmin module

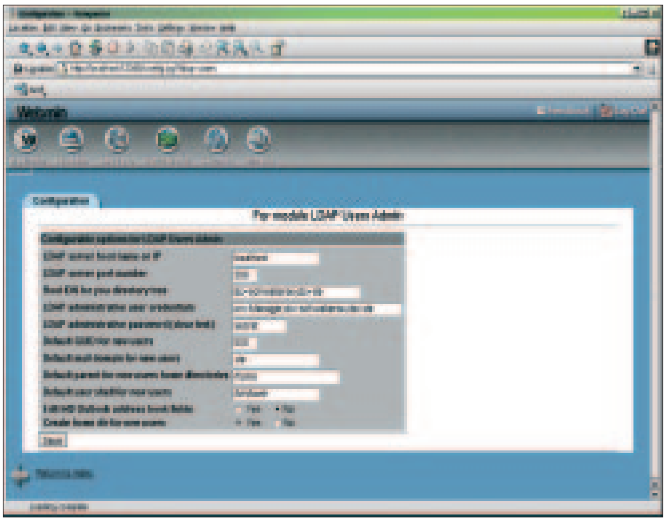


Figure 4: A sample *ldap-users* configuration

via the active Webmin interface. To do so, just run *Webmin Configuration* in the main menu and then call *Webmin Modules* (see Figure 3). After selecting *Install Module from File* you can then configure the module – configuration normally takes place immediately after selecting the new module.

You can start using the user interface immediately after these steps (see Figure 4). The interface provides quick and easy access to the attributes of any Posix object, and also allows you to create new users, although these features are unfortunately not available for groups.

In addition to the LDAP user administration you can also use Webmin to administer the OpenLDAP server. The plug-in is called OpenLDAP, for want of a better name, and is available from [6]. The module is also available for older OpenLDAP versions. The module for version 2 is called *openldap2*-X_X.*wbm*. The installation procedure is similar to the one used for the LDAP Users Admin Module, and can be accessed via Webmin's *Servers* menu. In addition to configuring Access Control Lists you will hopefully be able to modify and create object classes in future versions. An option for maintaining server attributes is now available.

Conclusion: It is easy to configure an OpenLDAP server using Webmin and LDAP modules. You can also delegate daily administrative tasks. If you already use Webmin, you will immediately feel at ease with the LDAP modules, as the look and feel of other modules is apparent.

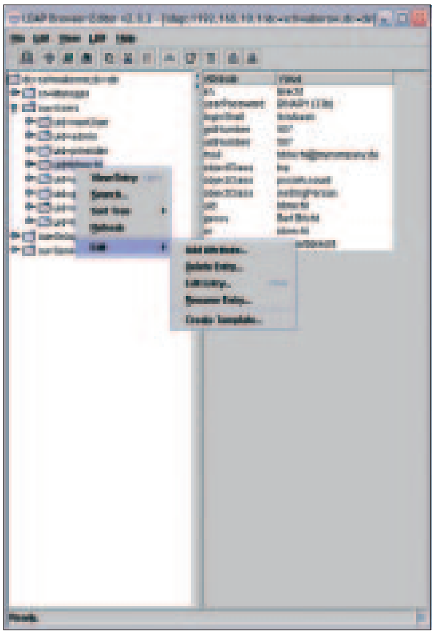


Figure 6: The LDAP Browser/Editor by Jarek Gawor provides a range of features comparable to commercial LDAP-Clients

Traditional: Native LDAP Client Programs

Besides the Web browser based administration you can also opt for the traditional method and try a native graphic interface. Native tools may be quicker in comparison to generic solutions, but you will have the disadvantage of having to select a single operating system. (Neither of these statements applies to Java programs, of course.) There are several Linux applications of this type that permit more or less professional administration of your LDAP directories.

One important pre-condition for all the programs described in this article is the ability to delegate administrative functions via Access Control Lists on the directory server itself. In the case of OpenLDAP the *access* rule is recommended for this purpose, as it can be defined for any user attribute. The following listing provides an example based on OpenLDAP:

```
# All users are allowed to
# maintain their own records
# Other users have read-only
# access.
access to *
  by self write
  by * read
```

LDAP Browsers/Editors

If users in large companies are to be allowed to maintain their own data, a user client platform dependent interface can cause you headaches. Java based GUIs are one solution. Although the language is not exactly famous for its graphic output speed, it will at least run on most platforms. The LDAP Browser/Editor [8] by Jarek Gawor, who works for Chicago University, is just one example of a Java program. The current version of the tool, 2.8.1, requires the Java Runtime Environment 1.2.2 or newer, and is fairly stable on systems with at least 128 mbytes of RAM. The flexibility of the Java GUI is comparable to that of commercial LDAP clients – which makes this tool a must for people who normally use native LDAP client software (see Figure 6).

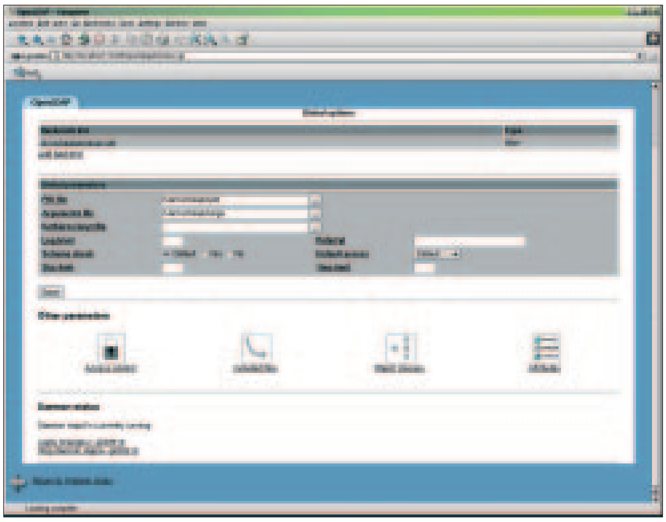
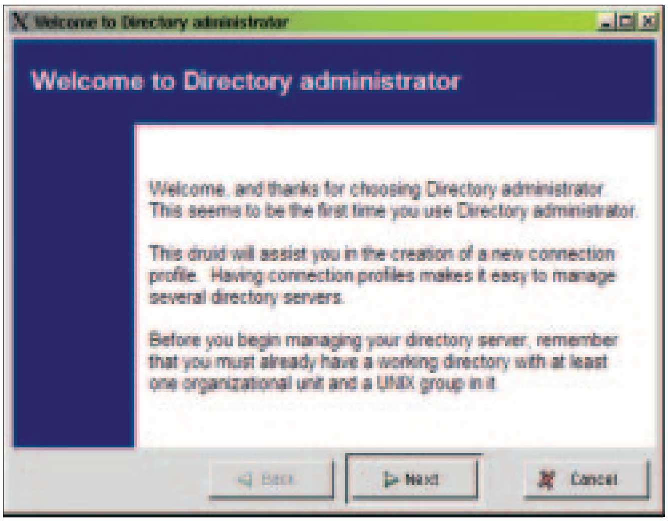


Figure 5: OpenLDAP Webmin modules for administering OpenLDAP servers



Welcome to Directory administrator

Welcome, and thanks for choosing Directory administrator. This seems to be the first time you use Directory administrator.

This druid will assist you in the creation of a new connection profile. Having connection profiles makes it easy to manage several directory servers.

Before you begin managing your directory server, remember that you must already have a working directory with at least one organizational unit and a UNIX group in it.

Figure 7: The Gnome Directory Administrator Tool includes wizards

The interface allows easy moving, manipulation and copying of directory objects. In addition to standard functionality, the LDAP Browser/Editor can also export data to the LDAP Data Interchange Format, LDIF. This allows you to export a complete LDAP tree in a matter of seconds.

This approach is useful for creating backups and equally so for migration tasks. The interface uses templates to create new entries. The administrator can use a simple template to define the required attributes, and that can be a big help if you are defining custom objects.

Conclusion: A universal and OS independent interface, such as the browser programmed by Jarek Gawor is a good thing to have around for those daily maintenance tasks, although performance could be better. Most administrators will appreciate a quality tool such as this – especially as it comes for free.

Directory Administrator

A number of LDAP GUIs are available for the Gnome desktop. One of them is the Directory Administrator, which is mainly suitable for user administration. The website [9] offers RPM binary archives for Mandrake 8.2 and Red Hat Linux 7.3, so you will need the source archives for any other OS. After expanding the archive you can follow standard procedure to compile and install the sources: *./configure; make; make install.*

A wizard is available to the admin user on initially launching the program (Figure 7) and can be used to create a connection profile. If everything works out okay, you will be able to view the users and groups stored in your directory (Figure 7). The interface can also perform tasks such as assigning users to groups, and it is extremely flexible with respect to storing user accounts and groups in OU hierarchies.

Conclusion: If you are managing the users and groups of a department in an organizational structure, the Directory Administrator is a good choice. The tool will perform tasks such as creating users, or defining a user's Samba shares, quickly and easily.

GQ and KdirAdm

There are numerous alternatives to the tools already discussed. GQ [9] for example, an LDAP client for the Gtk environment which is comparable to the LDAP browsers already discussed in most respects as it allows you to manage the objects in your directory. Another lookalike is KdirAdm [10].

Conclusion: useful for simple cases

Depending on their quality and the individual application you have in mind, the Open Source directory management tools that we have introduced in this article may (or may not) be useful to an administrator. Many of the tools tend to introduce too many levels to what are in effect simple administrative concepts.

THE AUTHOR

Volker Schwaberow is a technology consultant for RAG INFORMATIK GmbH in Gelsenkirchen, Germany, and started looking into Linux and associated topics in 1995. The author's hobbies are reading, listening to music and programming in C/C++, Java, Perl, and PHP.

It is also vital that you restrict access to the directory via ACLs. As an administrator you can base your choice of tool on the task in hand, although this may make life difficult for you when you first attempt to draw up an implementation plan.

Statistical evaluation is probably the best way to handle this. Statistics will help you determine whether you can safely delegate a wide range of administrative tasks.

As in many other cases the approach, and the results, will only be as good as your advanced planning. Your only option at this stage will normally be to create a list of mandatory administrative tasks and check whether one of the tools we discussed is suited to them.

A combination of tools may even prove to be your best option for simplifying your daily workload. You might consider using a Web interface that allows your users to modify their own personal data and passwords, but provide a native graphic frontend for the administrator. ■

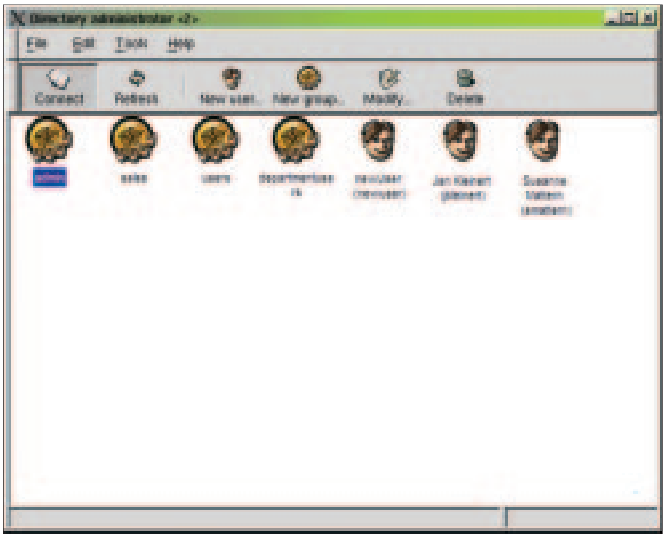


Figure 8: Making the Administrator's life simple: All the groups and users are visible at a single glance with Directory Administrator

INFO

[1] OpenLDAP: *http://www.openldap.org*

[2] PHP: *http://www.php.net*

[3] Gonicus: *http://www.gonicus.de*

[4] Webmin: *http://www.webmin.com*

[5] LDAP Users Admin: *http://ldap-users.sourceforge.net*

[6] OpenLDAP Webmin Module: *http://gaia.anet.fr/webmin/openldap*

[7] Directory Administrator: *http://diradmin.open-it.org/index.php*

[8] LDAP Browser/Editor: *http://www.iit.edu/~gawojar/ldap*

[9] GQ for Gtk: *http://biot.com/gq/*

[10] KDE Directory Administrator: >*http://www.carillonis.com/kdiradm*

[11] RFC1779 – A String Representation of Distinguished Names: *ftp://ftp.isi.edu/in-notes/rfc1779.txt*

[12] RFC1778 – The String Representation of Standard Attribute Syntaxes: *ftp://ftp.isi.edu/in-notes/rfc1778.txt*

[13] RFC1777 – Lightweight Directory Access Protocol: *ftp://ftp.isi.edu/in-notes/rfc1777.txt*