ST LDAP Manager: Zope verwaltet ein LDAP-Directory

Zentral managen

Wenn der Applikationsserver Zope im Zentrum der IT-Landschaft steht und seine Benutzerverwaltung zudem per LDAP macht, keimt beim zuständigen Admin bald ein Wunsch auf: Praktisch wäre es, das ganze LDAP-Verzeichnis in Zope statt mit Spezialtools zu verwalten. Gottfried J. M. Grosshans

ST LDAP Manager

Wer Zope, den LDAP User Folder und (Open-) LDAP bisher im Einsatz hat, kann nach dem Abarbeiten der Anleitung in diesem Artikel in Zope:

- LDAP-Schemata ändern,
- eine dedizierte Rollenstruktur für LDAP-Admins festlegen,
- Usern die Änderung ihrer eigenen LDAP-Daten erlauben,
- Usern die Suche in LDAP-Verzeichnissen nach vorgegebenen Mustern je nach Rechten einer Usergruppe ermöglichen,
- Formularmasken flexibel anpassen,
- LDAP-Objektklassen managen,
- fertige Konfigurationsdateien für LDAP-Server erzeugen.

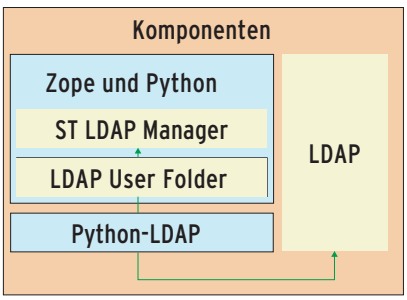

Abbildung 1: Die grüne Verbindung stellt die Kommunikation zwischen den Komponenten dar.

Abteilung	Mitarbeiter
Management	G. Grosshans
Marketing	Thomas Grundmann
Consulting	Volker Packe Katharina Weiler
Developer	Irina Bosley Bernd Pallaske
System Administrators	Michael Lips

Abbildung 2: Die neue interne Struktur der aus [7] bekannten Beispielfirma Cytux.

Zope besitzt eine eigene und damit proprietäre Userverwaltung [1]. Um nicht mehrere Benutzerdatenbanken führen zu müssen, erlaubt es der LDAP User Folder [2], die Userverwaltung einem (Open-)LDAP-Server ([3] bis [6]) zu überlassen. Der Linux-Magazin-Artikel [7] beschreibt das Kopplungsmanöver per LDAP User Folder.

Admins, die diese Softwarekombination am Laufen haben, möchten vielleicht gern ihr LDAP-Directory ganz und gar über Zope verwalten (lassen). Der LDAP User Folder als reiner Zope/LDAP-Koppler setzt diesem Ansinnen aber Grenzen. Beispielsweise das Delegieren einzelner Aufgaben an User funktioniert so nicht. Hier kommt der ST LDAP Manager, der auch als LDAP Directory Manager firmiert, ins Spiel. Dieser Beitrag beschreibt Installation, Setup und Funktion anhand eines Beispiels.

Den ST LDAP Manager [9] installieren zu können, setzt eine Verbindung zwischen Zope, Library Python-LDAP, dem LDAP User Folder und dem LDAP-Server voraus, wie sie [7] beschreibt. Das Zusammenspiel der einzelnen Komponenten zeigt Abbildung 1. [7] benutzte einen Workaround, um Python-LDAP zu installieren, der nicht gerade durch Eleganz in Erinnerung bleibt. Zwischenzeitlich ist der Autor auf die Zope-Version 2.6.2 (als Tar-Datei) umgestiegen und hat Python-LDAP aus den Quellen installiert. Der Kasten „Installationsarbeiten“ beschreibt die Schritte.

User Folder und ST LDAP Manager konfigurieren

Es ist ratsam, zunächst ein Testszenario anzulegen, um sich nicht selbst aus Zope auszusperren. Das gelingt beispielsweise

durch das Anlegen eines Ordnerobjekts »test«. In diesem Ordner legt der Zope-Admin ein LDAP-User-Folder-Objekt an, um die Rollenverteilung der User in diesem Ordnerobjekt an LDAP zu binden. Das setzt einen funktionierenden und administrierten LDAP-Server voraus.

Als Beispiel dient wie in [7] die Firma Cytux, diesmal allerdings um ein paar Mitarbeiter erweitert (Abbildung 2). Abbildung 3 zeigt die Konfigurationseinträge für den LDAP User Folder. Jetzt sollte der Admin die Schemata im LDAP User Folder anpassen, damit die Einträge der LDAP-Verzeichnisse über den LDAP User Folder einstellbar sind.

Wer das Beispiel eins zu eins durchspielen will, muss Gruppen und Mitarbeiter aus Abbildung 2 in das LDAP-Verzeichnis übernehmen – andernfalls natürlich die der eigenen Firma. Die Pflege kann

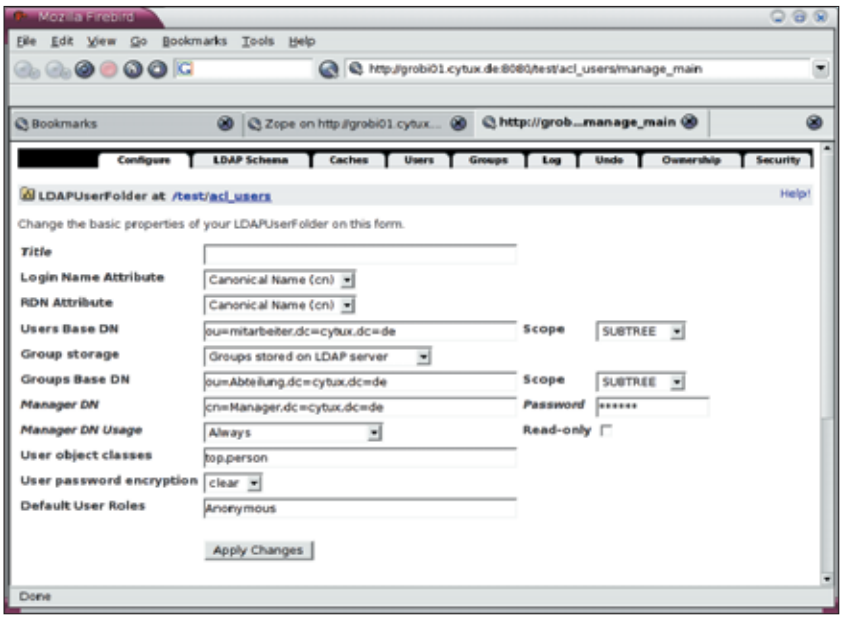

Abbildung 3: Die Konfigurationseinträge für den LDAP User Folder. Die Sternchen verbergen »secret«.

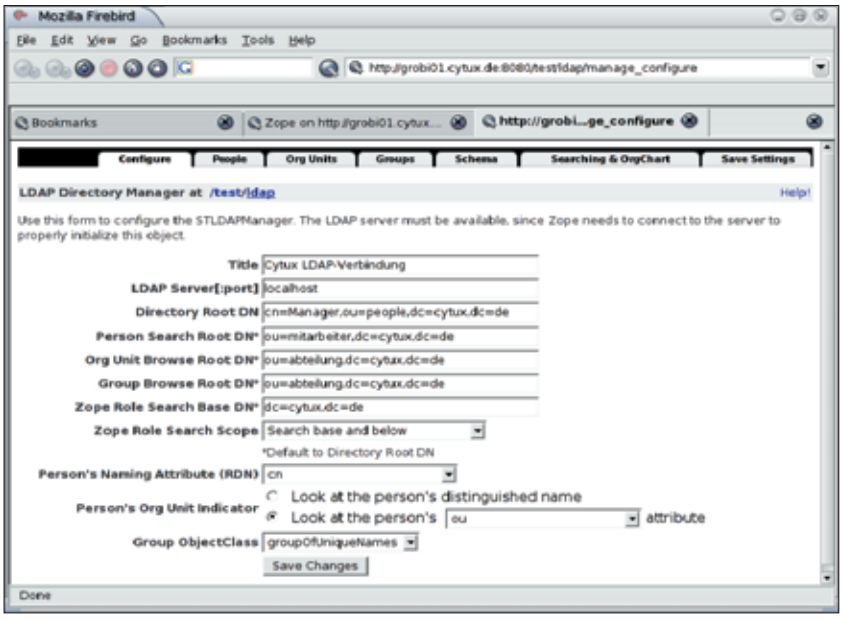

Abbildung 4: Das Formular für die Grundkonfiguration von ST LDAP Manager.

Installationsarbeiten

1. Die Datei »python-ldap-2.0.0pre19.tar.gz« (oder eventuell eine neue Version) von der Projektseite [8] ist mit »tar xvfz *Datei*« zu entpacken.

2. Nach dem Wechseln ins Entpack-Verzeichnis ändert man die Datei »setup.py«: In die Zeilen 105 und 113 gehört vor die Module »ldap« und »ldap.schema« ein Hash (»#«). Das Auskommentieren ist nötig, da beide Module nicht mit Python 2.1 klarkommen. Wer Python ab Version 2.3 für Zope benutzt, darf sich diesen Schritt sparen.

3. Dann ist die Datei »setup.cfg« im gleichen Verzeichnis dran, sodass in ihr folgende Library- und Include-Pfade angegeben sind:

```
library_dirs = /usr/local/?
lib Zope-Pfad/lib /usr/lib
include_dirs = /usr/local/?
include Zope-Pfad/include /usr/?
include /usr/include/sasl
```

Im absolut anzugebenen »*Zope-Pfad*« ist der Zope-Server installiert.

4. Der Aufruf »*Zope-Pfad*/bin/python setup.py build« erzeugt die Python-LDAP-Module und »*Zope-Pfad*/bin/python setup.py install« installiert sie.

5. Anschließend installiert man den LDAP User Folder wie in [7] beschrieben durch Entpacken der Datei im Zope-Produktverzeichnis »*Zope-Pfad*/lib/python/Products«. Wie gehabt sollten keine Rindviecher im Produktpfad in Zopes Webinterface erscheinen, was darauf hindeuten würde, dass die Installation von Python-LDAP nicht erfolgreich war.

6. Um den ST LDAP Manager zu installieren, ist dessen Tar-Datei von [9] ebenfalls im Zope-Produktverzeichnis zu entpacken. Nach einem Neustart erscheint auch der ST-DAP-Manager-Eintrag im Produktpfad des Webinterface. Damit ist die Installation abgeschlossen.

entweder über den LDAP User Folder oder ein anderes LDAP-Admin-Tool außerhalb von Zope geschehen. Anschließend ordnet man die Mitarbeiter ihren Gruppen wie in Abbildung 2 zu.

Nun legt der Admin in der Testumgebung ein LDAP-Directory-Manager-Objekt an. Nach dem Auswählen des Objekttyps zur Neuanlage fordert Zope dazu auf, Daten der Struktur des LDAP-Servers in ein Formular einzutragen. Für das Beispiel lauten die Angaben wie in Abbildung 4. Das Speichern mit »Save Changes« beendet die Vorkonfiguration.

Arbeiten mit dem ST LDAP Manager

Jetzt steht im Testordner das neue Objekt »LDAP«. Ein Klick darauf führt wieder zur Konfigurationsseite von ST LDAP Manager. Die Lasche »People« am oberen Fensterrand präsentiert eine Suchmaske, die per Namen nach Mitarbeitern fahndet. Die Daten der gefundenen Person sind per Klick auch änderbar – die nötigen Rechte vorausgesetzt.

Da aber im Moment der Anmeldung noch der Rollenkontext von Zope gültig ist, hat man keinerlei Rechte LDAP-Daten zu ändern. Darum muss der Admin seinen Rollenkontext durch Neu-Anmelden ändern. Hierzu klickt er auf »Login« und bekommt eine Anmelde-Aufforderung. Im Beispiel heißt der Systemadministrator Michael Lips – beim Anmelden erhält er jetzt die Rechte als Administrator. Als Standard definiert der ST LDAP Manager eine Gruppe »System Administrators« als LDAP-Administratorgruppe. Damit besitzt die Rolle jetzt die Schreibrechte auf LDAP-Daten.

Wenn der frisch gebackene Admin jetzt über die »People«-Lasche wieder einen User sucht und findet, wird er einige Links zusätzlich sehen. Abbildung 5 zeigt das Formular für das Ändern der Benutzerdaten. Das System führt automatisch nur zu den Formularen, für die der Bediener Rechte besitzt. Durch einen Klick auf andere Bereiche erscheinen weitere Formularfelder.

Der ST LDAP Manager ist so voreingestellt, dass Angehörige der LDAP-Gruppe »System Administrators« andere User, Gruppen und Schemata modifizieren dürfen. Mitarbeiter, die dieser Gruppe nicht angehören, dürfen nur ihre eigenen Daten samt Kennwort ändern.

Über die Lasche »ORG Units« lassen sich ORG-Units nach dem gleichen Schema suchen und bearbeiten. Die recht simple LDAP-Struktur von Cytux nutzt aber praktisch keine komplexe ORG-Units-Struktur. Analog zu den Mitarbeitern und ORG-Units funktioniert es mit Gruppen: »Groups | Browse Groups Tree« fördert beispielsweise die Firmenstruktur nach Gruppen sortiert zu Tage. Per Klick auf die entsprechende Gruppe gelangt man wieder zu den Änderungsmasken – soweit es die Rolle erlaubt.

Hier lassen sich mit dem Button »Select Owner« Administratoren für diese eine Gruppe einrichten, die Gruppendaten bearbeiten dürfen (siehe Abbildung 6). Wenn es darum geht, der Gruppe Mitar-

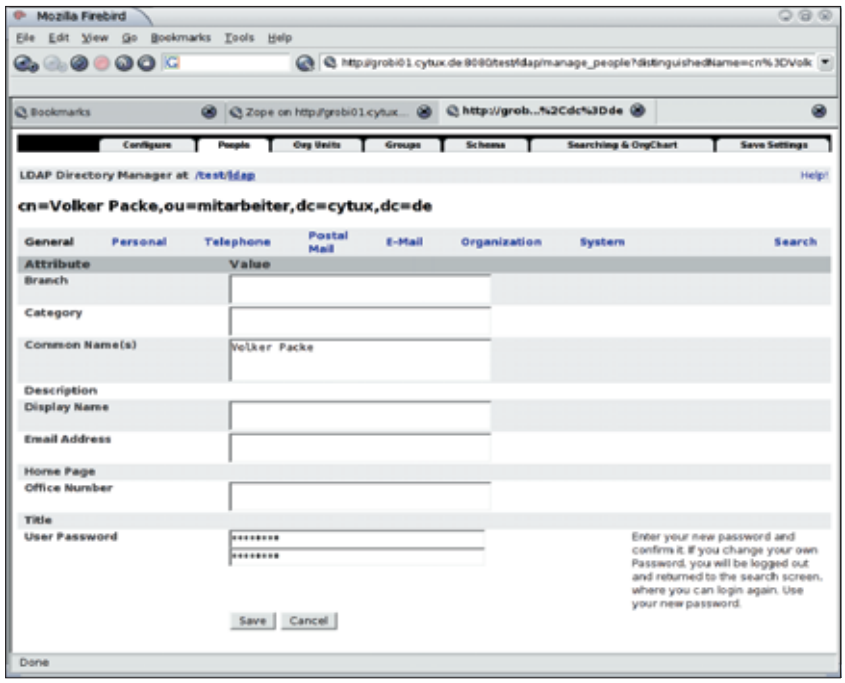

Abbildung 5: Die Änderung der Daten eines LDAP-Users ist jetzt möglich.

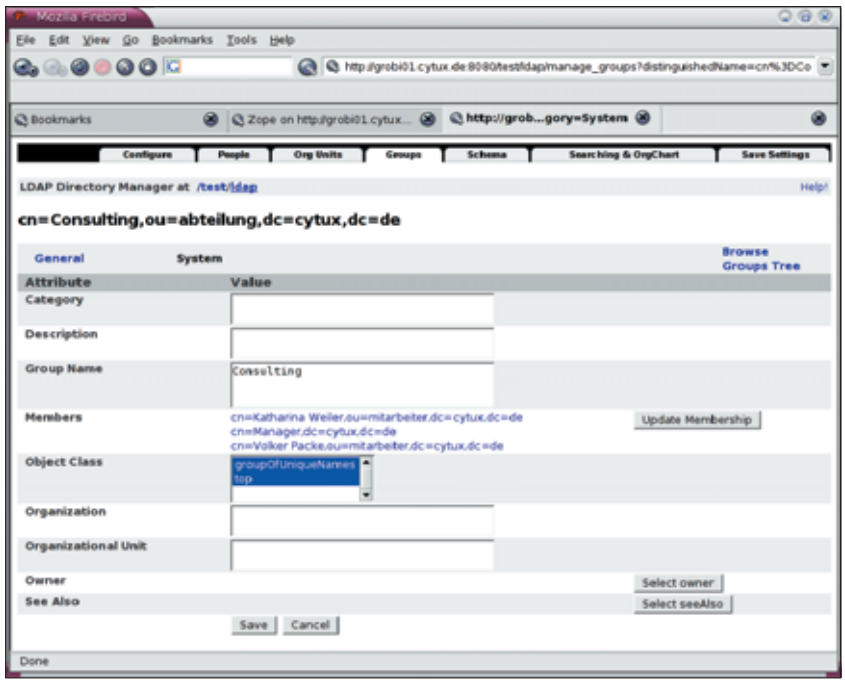

Abbildung 6: Über »Select Owner« lässt sich ein Gruppenadministrator bestimmen, »Update Membership« ordnet der Gruppe die Mitarbeiter zu.

beiter zuzuordnen, ist der Button »Update Membership« gefragt. Das Gleiche bewirkt der entsprechende Button in der Startseite der Lasche »Groups«.

Viele Möglichkeiten

Wie gezeigt ist das System in der Standardkonfiguration schon sehr flexibel gestaltet. Doch die Gestaltungsmöglichkeiten sind damit nicht erschöpft. Beispielsweise versetzt die Lasche »Searching & Orgchart« den Admin in die Lage, die Suchmasken frei anzupassen und zu erweitern. Auch das Anzeigeverhalten der Gruppenstrukturen kann er dort einstellen. Wem das nicht reicht, der passt alle Schemata über die Lasche »Schema« an. Hier wählt er auch das Wunschobjekt: »People«, »ORG-Units« oder »Groups« und modifiziert es anschließend nach Belieben.

In gleicher Weise lassen sich über den Link »Select Attributes« (siehe **Abbildung 8**) beziehungsweise »Manage Attributes« die Attribute für die einzelnen Änderungsmasken und über »Display Categories« einzelne Unterkategorien der Masken zuerst auswählen und dann verwalten. Die Lasche »Roles« öffnet den

Listing 1: Konfiguration von OpenLDAP

```
# Start of LDAP Manager generated configuration

access to dn="cn=subschema" by * read

access to dn.subtree="ou=mitarbeiter,dc=cytux,dc=de"
   filter=(objectclass=person)
   attrs=children,entry
   by * read

access to dn.subtree="ou=mitarbeiter,dc=cytux,dc=de"
   filter=(objectclass=person)
   attrs=homePhone,homePostalAddress,mobile,pager
   by self write
   by dnattr=manager read
   by users auth
   by anonymous auth

access to dn.subtree="ou=mitarbeiter,dc=cytux,dc=de"
   filter=(objectclass=person)
   attrs=mail,mailAlternateAddress,objectClass,uid
   by self read
   by dnattr=manager read
   by users read
   by anonymous read

access to dn.subtree="ou=mitarbeiter,dc=cytux,dc=de"
   filter=(objectclass=person)
   attrs=o
   by self read
   by dnattr=manager read
   by users auth
   by anonymous auth

access to dn.subtree="ou=mitarbeiter,dc=cytux,dc=de"
   filter=(objectclass=person)
   attrs=physicalDeliveryOfficeName,st,street,title
   by self read
   by dnattr=manager read
   by users read
   by anonymous read
...
access to dn.subtree="ou=abteilung,dc=cytux,dc=de"
   filter=(objectclass=organizationalunit)
   attrs=children,entry
   by group="cn=System Administrators,ou=Abteilung,dc=cytux,dc=de" write
   by * read

access to dn.subtree="ou=abteilung,dc=cytux,dc=de"
   filter=(objectclass=organizationalunit)
   attrs=businessCategory,description,facsimileTelephoneNumber,l,ou,physicalDeliveryOfficeName,postOfficeBox,postalAddress,postalCode,searchGuide,seeAlso,st,street,telephoneNumber
   by self auth
   by dnattr=manager read
   by users read
   by anonymous read

access to dn.subtree="ou=abteilung,dc=cytux,dc=de"
   filter=(objectclass=organizationalunit)
   attrs=objectClass
   by self auth
   by users read
   by anonymous read

access to dn.subtree="ou=abteilung,dc=cytux,dc=de"
   filter=(|(objectclass=groupofnames)(objectclass=groupofuniquenames))
   attrs=children,entry
   by group="cn=System Administrators,ou=Abteilung,dc=cytux,dc=de" write
   by * read

access to dn.subtree="ou=abteilung,dc=cytux,dc=de"
   filter=(|(objectclass=groupofnames)(objectclass=groupofuniquenames))
   attrs=o
   by self read
   by users read
   by anonymous auth

access to dn.subtree="ou=abteilung,dc=cytux,dc=de"
   filter=(|(objectclass=groupofnames)(objectclass=groupofuniquenames))
   attrs=cn,objectClass,ou
   by self read
   by users read
   by anonymous read

access to dn.subtree="ou=abteilung,dc=cytux,dc=de"
   filter=(|(objectclass=groupofnames)(objectclass=groupofuniquenames))
   attrs=businessCategory,description,owner
   by self write
   by users read
   by anonymous auth

access to dn.subtree="ou=abteilung,dc=cytux,dc=de"
   filter=(|(objectclass=groupofnames)(objectclass=groupofuniquenames))
   attrs=seeAlso,uniqueMember
   by self write
   by users read
   by anonymous read

# End of LDAP Manager generated configuration
```

Abbildung 7: Die Lasche »Roles« führt zum feingranularen Rechtemanagement.

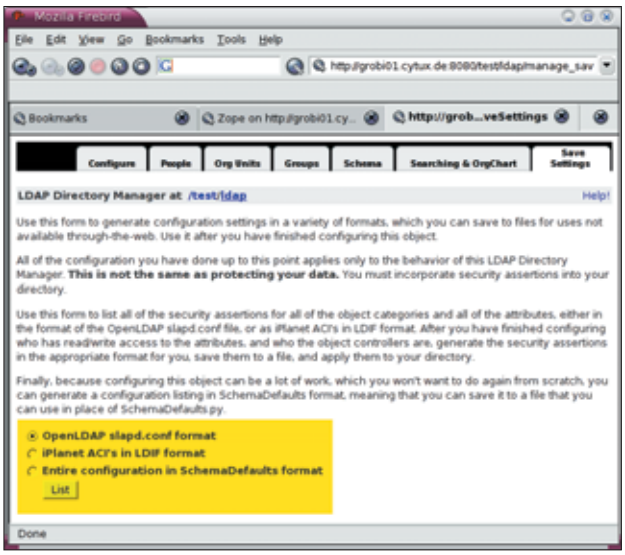

Abbildung 9: Sicherung der Daten in Formaten für LDAP-Server.

Abbildung 8: Einstellen der Attribute für die Änderungsmasken.

Zugang zu dem feingranular aufgebauten Rechtemanagement, wie es Abbildung 7 zeigt. Hier ist zu sehen, dass die Rolle »Systems Administrators«, wie nicht anders zu erwarten, schon definiert ist. Ein Klick auf diese Rolle befördert die Maske für die dedizierte Einstellung aller Rechte der zugehörigen Gruppe auf den Schirm.

Über den Link »Object Controllers« wird eingestellt, wer bestimmte Objekte ändern darf. Außerdem besteht noch die Option, per »Save Settings« die eingestellten Schemata in auch für LDAP-Server lesbare Konfigurationsdateien zu sichern – das spart viel Tipparbeit. In Listing 1 ist der Ausschnitt einer so erzeugten Ausgabedatei zu sehen, die dem Setup aus Abbildung 9 entspricht.

Zope als Admin-Zentrale

Der ST LDAP Manager erweitert Zope und LDAP User Folder so, dass ein Admin eine LDAP-Datenbank von hier aus flexibel managen kann. Der Beitrag skizziert die Basisfunktionalität des Softwaretrios. Es verwaltet bei Bedarf auch deutlich komplexere LDAP-Strukturen. Nähere Erläuterungen zu den einzelnen Formularen von ST LDAP Manager geben ausführliche Hilfeseiten. *(jk)*

Infos

[1] Zope: [http://www.zope.org], [http://www.zope.com], [http://www.dzug.org]
[2] LDAP User Folder: [http://www.dataflake.org/software/ldapuserfolder]
[3] OpenLDAP: [http://www.openldap.org]
[4] V. Schwaberow, „OpenLDAP-Praxis“: Linux-Magazin 5/2001, S. 84
[5] Th. King, „Workshop: LDAP, Teil1“: Linux-Magazin 6/01, S. 106
[6] Th. King, „Workshop: LDAP, Teil2“: Linux-Magazin 8/01, S. 119
[7] G. Grosshans, „Verbindliche Auskünfte“: Linux-Magazin 11/03, S. 106
[8] Python-LDAP: [http://python-ldap.sf.net]
[9] ST LDAP Manager (LDAP Directory Manager): [http://zope.org/Members/stevray/STLDAPManager]