Spring 5-1-2006

Using Hypertext Preprocessor Lightweight Directory Access Protocol (PHPLDAP) to Manage Infrastructure Services for an Educational Institution

John B. Lukach
Dakota State University

Using Hypertext Preprocessor Lightweight Directory Access Protocol (PHPLDAP) to manage Infrastructure services for an educational institution

A graduate project submitted to Dakota State University in partial fulfillment of the requirements for the degree of

Master of Science

in

Information Systems

May 2006

By

John B. Lukach

Project Committee:

Mark Moran

Stephen Krebsbach

Michael Moriarty

We certify that we have read this project and that, in our opinion, it is satisfactory in scope and quality as a project for the degree of Master of Science in Information Systems.

Project Committee

Faculty supervisor: Mark Moran Date: 5/5/06

Committee member: Date: 5/10/06

Committee member: Michael J Moriarty Date: 5-3-06

ABSTRACT

Independent School District #299 needed a web-based application to manage user accounts and groups of the heterogeneous computer system. PHPLDAP was developed internally using PHP programming language since a satisfactory solution was not currently available. The computer system design was critical to the achievement of single sign-on through a centralized Sun Java System Directory Server. The project takes a close look at integration of the schools computer system specifically the development of: Sun Java System (LDAP) Directory server setup, Samba file server setup, pGina Microsoft Windows client authentication setup, Apple Computer client authentication setup, and Smart Filter web filtering setup. PHPLDAP has been very successful at reducing support time, license fees, and management requirements of Caledonia Area Public School's computer system.

DECLARATION

I hereby certify that this project constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions or writings of another.

I declare that the project describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

John B. Lukach

John B. Lukach

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

STATEMENT OF THE PROBLEM

Caledonia Area Public Schools needed free web-based account management software to manage its heterogeneous computer system. Previously information needed to be generated, stored, changed, and deleted in the centralized Lightweight Directory Access Protocol (LDAP) Directory server for the Email server, Calendar server, Sun Rays thin clients, Microsoft Windows computers, Cisco networking equipment, Apple computers, Samba file server, Smart Filter web filtering, and Roaring Penguin Can IT Pro spam/anti-virus server required seven programs to create one user account not including groups and other special settings. Lightweight Directory Access Protocol (LDAP) is a network protocol for querying and modifying a tree of entries that consists of attributes and values stored in a centralized database of account information (1). Sun Ray thin clients are display devices that have users applications and session running else where independently on a server (2). Samba file server is a free software package that duplicates a Microsoft Windows Server networking environment (3).

Commercially Sun Microsystems Java Identity Management Suite made up of Sun Java System Identity Manager and Sun Java Access Manager is available to manage the Sun Java System (LDAP) Directory Server. The software packages make it possible to manage passwords, profiles, identity synchronization, access control, and single sign-on (4). This option is financially impossible for Caledonia Area Public Schools due to the yearly $25,000 education software license fee for 1,000 users.

Two free Open Source software packages found on Source Forge, a hosting site for developers to share their work are available (5). “phpLDAPadmin” is a web-based application that provides template-based entry editing for creating custom accounts and groups but does not provide Samba support for Microsoft Windows clients (6). “LDAP Account Manager (LAM)” is a web-based application that does not provide template-based entry or generate the necessary NT/LM password hashing necessary for Microsoft Windows file sharing support through Samba (7).

System management at Caledonia Area Public Schools was labor and time intensive to manage user accounts and groups for the lone technology staff member. Resources needed to be allocated by the District Office and Student Services staff through a web-based application when end-users' needs change. Since no commercial or Open Source software package was available a homegrown web application was developed to meet the district's needs.

BACKGROUND OF THE PROBLEM

Independent School District #299 had a yearly software license bill three times the annual budget in the 2003 school year. The major reason for the expense was multiple duplicate directory servers were being maintained. Two separate Sun Java System (LDAP) Directory Servers were being used to authenticate Email server, Calendar server, and Sun Rays thin clients. One Microsoft Active Directory Server was used for Microsoft Windows computers and Cisco networking equipment authentication (8). One Open LDAP Directory was used for the Apple computers authentication (9). All directories were consolidated into a single Sun Java System (LDAP) Directory Server. Sun Java Communications Suite solution was picked over the Microsoft Windows Exchange Server solution because there are no yearly fees for email, calendar, and directory services (10). Windows Exchange Server is Microsoft's solution for email, calendar, and directory services at a yearly cost (11).

Sun Java Communications Suite includes Java Identity Management Suite, which was built in during Quarter Two of Sun's 2003 fiscal year. Sun then completely dropped Java Identity Management Suite from the Java Communications Suite in Quarter Two of Sun's 2004 fiscal year leaving major account and group creation problems. Sun brought Java Identity Management Suite back into the Java Communications Suite in Quarter Two in their 2005 fiscal year on a fee basis only.

OBJECTIVES OF THE PROJECT

The PHPLDAP project will describe configuration of the Sun Java System (LDAP) Directory server to support the Email server, Calendar server, Sun Rays thin clients, Microsoft Windows computers, Cisco networking equipment, Apple computers, Samba file server, Smart Filter web filtering, and Roaring Penguin Can IT Pro spam/anti-virus. Attributes and entries for software development are listed for dependencies of software development. Creates the foundation the entire heterogeneous computer system functions and depends on.

The Samba file server provides the necessary services for Microsoft Windows client file access. A conflict with the directory server design was identified requiring in-depth setup directions. Samba file server allows Microsoft operating systems access to the district's Network File System (NFS) file server.

pGina Microsoft Windows client is for authentication against the Sun Java System (LDAP) Directory server. pGina is an Open Source software that allows you to bypass the single authentication method Microsoft provides for its Windows operating systems (12). This makes crypt password hashing acceptable for user login.

Apple Computer client authentication allows computers to work against the Sun Java System (LDAP) Directory server. Directory Access tool uses RFC 2307 (Unix) settings through a simple graphical user interface (GUI) to make the directory server accessible. This is necessary for Apple Macintosh computers running operating system (OS) 7.2 and 8.6 with only Apple Talk network protocol.

Smart Filter web filtering setup allows for the use of centralized Sun Java System (LDAP) Directory server. Configuring the Internet filter against the directory allows reports to show user account information. This provides meaningful information for the Children's Internet Protection Act (CIPA) tracking.

Each PHPLDAP configuration step will provide a detailed description of Caledonia Area Public Schools heterogeneous computer system setup. PHP is a scripted programming language that stands for Hypertext Preprocessor used to create dynamic web applications and was the basis to build PHPLDAP (13). The district needed this solution to allow existing and future hardware to remain in production without creating financial hardships.

LITERATURE REVIEW

Entegrity Solutions Assure Access was a client-based tool that worked with Netscape LDAP. Over the years Netscape LDAP has become iPlanet and Sun ONE, which is now called Sun Java Enterprise System (JES). Entegrity Solutions says Assure Access was not designed for user administration, because most LDAP servers bundle sophisticated user management GUI's (14). Java Enterprise System (JES) is an end-to-end software system that supports all infrastructure service needs (15). Many open source account and group management clients exist on Source Forge. However Caledonia Area Public Schools requirement of all web-based software limited the options to creating a homegrown web application, phpLDAPadmin software, LDAP Account Manager (LAM) software, and Java Identity Management Suite formally called Wave Set (16).

Java Identity Management Suite is commercially supported software that is available to manage the Sun Java System (LDAP) Directory Server. Financially this option is impossible for Caledonia Area Public Schools due to the yearly $25,000 education software license fee for 1,000 users. Sun's product does not support Samba file server software, Network File System (NFS) server management, or Sun Java Communications Suite directory clean up. Network File System (NFS) is a protocol that allows a distributed file system to be accessed over the network (17). Java Identity Management Suite cannot generate the necessary attributes or NT/LM password hash necessary for Samba to operate properly. Also the software does not create or delete the necessary Sun Rays thin clients, Microsoft Windows computers, Cisco networking equipment, and Apple computers profiles necessary for each individual user account to function. In addition, when user accounts are deleted the Sun Java Communications Suite email, calendar, and address book data is not removed.

phpLDAPadmin and LDAP Account Manager (LAM) are two free Open Source software packages written in PHP for the management of LDAP directory server (18). They both have the same inherent problems as Java Identity Management Suite except

LAM has included the necessary Samba directory server attributes. The phpLDAPadmin advantage is its ability to create template based user account and groups where as LAM can only manage Unix operating system user and group posix accounts. Unix is a computer operating system that was designed to be portable, multi-tasking, and multi-user in a time-sharing configuration (34). Posix is the collective name of a family of related standards to define the application-programming interface (API) for software compatible with variants of the Unix operating system (19). Neither can manage Microsoft Windows client authentication nor create the personal address book (PAB) for the Sun Java Communications Suite.

The Open Source solutions are still miles behind the commercial product for system management. Sun Microsystems has started releasing their software to Open Source free of charge for custom modification by the development community (20). However the ones Caledonia Area Public Schools need are not available yet. Until this happens, Caledonia Area Public Schools had no choice but to develop a home grown web-based application to save on yearly license fees and support costs. Ideally if the Sun Java Identity Management Suite was not cost prohibitive it would be used in conjunction with a home grown web-based application for the missing features.

SYSTEM DESIGN

SUN JAVA SYSTEM (LDAP) DIRECTORY SETUP

Since Independent School District #299 decided to create a home grown web-based application it was necessary to determine the exact schema of attributes and entries for PHPLDAP development (21). The Sun Java System (LDAP) Directory Server is configured by the Sun Java Communications Suite but does not create the attributes or entries needed for the Sun Rays thin clients, Microsoft Windows computers, Cisco networking equipment, Apple computers, Samba file server, Smart Filter web filtering, and Roaring Penguin Can IT Pro spam/anti-virus server. To create the necessary directory schema the /usr/lib/ldap/idsconfig script needs to be run (22). The directory also needs to be configured to store the password in crypt hashing. Examples of a user accounts and groups are provided in Table 1 through Table 3 for reference of software development.

TABLE 1: USER ACCOUNT LDAP ATTRIBUTES

dn: uid=jblukach, ou=people, o=cps.k12.mn.us,o=isp
profilePath:
sunUCDateFormat: M/D/Y
homeDrive: /export/home
scriptPath:
icsExtendedUserPrefs: ceDefaultView=monthview
icsExtendedUserPrefs: ceInterval=PT0H30M
icsExtendedUserPrefs: ceExcludeSatSun=0
icsExtendedUserPrefs: ceGroupInviteAll=1
icsExtendedUserPrefs: ceSingleCalendarTZID=0

TABLE 1 CONTINUED: USER ACCOUNT LDAP ATTRIBUTES

icsExtendedUserPrefs: ceAllCalendarTZIDs=1
icsExtendedUserPrefs: ceNotifyEnable=0
icsExtendedUserPrefs: ceNotifyEmail=jblukach@cps.k12.mn.us
icsExtendedUserPrefs: ceDefaultAlarmStart=
icsExtendedUserPrefs: ceDefaultAlarmEmail=jblukach@cps.k12.mn.us
icsExtendedUserPrefs: ceShowCompletedTasks=false
icsExtendedUserPrefs: ceDefaultCategory=Business
icsExtendedUserPrefs: ceDayHead=9
icsExtendedUserPrefs: ceDayTail=18
icsExtendedUserPrefs: ceWeekEndDays=1,7
icsExtendedUserPrefs: ceIncludeWeekendInViews=true
icsExtendedUserPrefs: sunCalEventfilter=accepted,tentative,declined,needs-action
icsExtendedUserPrefs: sunCalInitialized=true
pwdMustChange: 2147483647
sunUCDefaultApplication: mail
mail: jblukach@cps.k12.mn.us
uid: jblukach
icsFirstDay: 1
pwdCanChange: 1123519039
sunUCTimeFormat: 12
icsCalendar: jblukach
mailMsgQuota: -1
objectClass: top
objectClass: person
objectClass: inetorgperson
objectClass: iplanet-am-managed-person
objectClass: organizationalPerson

TABLE 1 CONTINUED: USER ACCOUNT LDAP ATTRIBUTES

objectClass: inetUser
objectClass: ipUser
objectClass: userPresenceProfile
objectClass: inetMailUser
objectClass: inetLocalMailRecipient
objectClass: icsCalendarUser
objectClass: iplanet-am-user-service
objectClass: inetAdmin
objectClass: iPlanetPreferences
objectClass: posixAccount
objectClass: shadowAccount
objectClass: account
objectClass: sunUCPreferences
objectClass: sambaAccount
sunUCDefaultEmailHandler: uc
uidNumber: 2071
sn: Lukach
domain:
mailUserStatus: active
sunUCTheme: uwc
gidNumber: 20
sunUCTimeZone: America/Chicago
icsStatus: active
mailDeliveryOption: mailbox
mailQuota: -1
primaryGroupID: 1041
userPassword:: e0NSWVBUfWN4bW5vTUxkN2FRMzY=

TABLE 1 CONTINUED: USER ACCOUNT LDAP ATTRIBUTES

nswmExtendedUserPrefs: meDraftFolder=Drafts
nswmExtendedUserPrefs: meSentFolder=Sent
nswmExtendedUserPrefs: meTrashFolder=Trash
nswmExtendedUserPrefs: meInitialized=true
nswmExtendedUserPrefs: mepabmigration=1
icsSubscribed: jblukach$John Lukach
lmPassword: 67DCFF7D6A96AB301104594F8C2EF12B
mailHost: webmail.cps.k12.mn.us
rid: 5142
sunUCColorScheme: 2
cn: John Lukach
acctFlags: [U]
logoffTime:
icsCalendarOwned: jblukach$John Lukach
pabURI:ldap://webmail.cps.k12.mn.us:389/ou=jblukach,ou=people, o=cps.k12.mn.us,o=isp,o=pab
iplanet-am-user-login-status: Active
sunUCDateDelimiter: /
givenName: John
homeDirectory: /export/home/jblukach
pwdLastSet: 1123519039
sunUCExtendedUserPrefs: sunUCInitialized=true
sunAbExtendedUserPrefs: abName=Personal Address Book
sunAbExtendedUserPrefs: abDescription=This is the personal address book
sunAbExtendedUserPrefs: abEntriesPerPage=25
sunAbExtendedUserPrefs: abSearchDisplayColumn1=displayname
sunAbExtendedUserPrefs: abSearchDisplayColumn2=primaryemail

TABLE 1 CONTINUED: USER ACCOUNT LDAP ATTRIBUTES

sunAbExtendedUserPrefs: abSearchDisplayColumn3=primaryphone
sunAbExtendedUserPrefs: sunAbInitialized=true
inetUserStatus: active
ntPassword: 19FC595DDA06D16F05D6DA22739AC71A
kickoffTime:
userWorkstations:
smbHome: \\172.16.56.78\homes
loginShell: /bin/sh
logonTime:
preferredLanguage: en

TABLE 2: EMAIL GROUP LDAP ATTRIBUTES

dn: cn=schoolboard, ou=Groups, o=cps.k12.mn.us,o=isp
uniqueMember: uid=mike_moriarty, ou=people o=cps.k12.mn.us,o=isp
mgmanMemberVisibility: none
owner: uid=ServiceAdmin, ou=People, o=cps.k12.mn.us,o=isp
mgmanJoinability: none
nsNumUsers: 0
mail: schoolboard@cps.k12.mn.us
objectClass: top
objectClass: groupOfUniqueNames
objectClass: inetMailGroup
objectClass: inetMailGroupManagement
objectClass: inetLocalMailRecipient
objectClass: nsManagedMailList

TABLE 2 CONTINUED: EMAIL GROUP LDAP ATTRIBUTES

mgrpAllowedDomain: cps.k12.mn.us
mailHost: webmail.cps.k12.mn.us
mailDeliveryOption: members
inetMailGroupStatus: active
cn: schoolboard
preferredLanguage: en
mgmanHidden: false
mgrpRFC822MailMember: mjfrank@acegroup.cc
mgrpRFC822MailMember: hurley@acegroup.cc
mgrpRFC822MailMember: ctschult@acegroup.cc
mgrpRFC822MailMember: wray@acegroup.cc
mgrpRFC822MailMember: jeanm@acegroup.cc
mgrpRFC822MailMember: vnfruechte@sgwb.coop
mgrpRFC822MailMember: knutsonmj@acegroup.cc
nsMaxUsers: 1000

TABLE 3: POSIX GROUP LDAP ATTRIBUTES

dn: gidnumber=1029, ou=group, o=cps.k12.mn.us,o=isp
gidNumber: 1029
memberUid: mike_moriarty
memberUid: karen_schiltz
memberUid: amy_schmidt
memberUid: barb_meyer
objectClass: top
objectClass: posixgroup
cn: district

SAMBA FILE SERVER SETUP

Samba File Server software needs the following attributes added to the 99user.ldif schema file to store information in the Sun Java System (LDAP) Directory Server: sambaAccount objectClass, acctFlags, domain, homeDrive, kickoffTime, lmPassword, logoffTime, logonTime, ntPassword, primaryGroupID, profilePath, pwdCanChange, pwdLastSet, pwdMustChange, rid, scriptPath, smbHome, and userWorkstations. Custom attributes needed to be added because Sun used the Object Identifiers (OIDs) that are reserved for Samba directory configuration (23). An Object Identifier (OID) is a sequence of integers assigned to all attributes and object classes to conform to LDAP standards distributed by Internet Assigned Number Authority (IANA) (24). To compile Samba 3 the following additional packages must be installed: Make, GCC, LibGCC, LibIConv, NCurses, PopT, ReadLine, and OpenLDAP. Syncing the directory server password and creating the NT/LM password hashes is accomplished using the smbpasswd command. Taking the UID times two plus one thousand configures the RID attribute. Taking the GID times two plus one thousand one configures the primaryGroupID attribute. Samba functionality is controlled through the SMB.CONF file (25). The build of Samba against Sun Java System (LDAP) Directory Server required an advanced configuration file (SMB.CONF) as displayed in Table 4 through Table 6.

TABLE 4: SMB.CONF GLOBAL SETTINGS

[global]
ldap ssl = off
ldap port = 389
ldap server = 172.16.56.95
ldap suffix = o=cps.k12.mn.us,o=isp
ldap admin dn = "cn=Directory Manager"
workgroup = Caledonia
netbios name = Samba
os level = 255

TABLE 4 CONTINUED: SMB.CONF GLOBAL SETTINGS

server string = Caledonia File Server
wins support = yes
security = user
encrypt passwords = yes
log file = /usr/local/samba/log/log.%m
log level = 1
socket options = TCP_NODELAY
dns proxy = No
max log size = 1000
socket options = TCP_NODELAY SO_SNDBUF=8192 SO_RCVBUF=8192
level2 oplocks = true
read raw = no
max xmit = 2048
follow symlinks = yes

TABLE 5: SMB.CONF HOME DIRECTORY SETTINGS

[homes]
comment = Home Directories
delete readonly = yes
browseable = yes
writeable = yes
create mask = 0664
directory mask = 0755

TABLE 6: SMB.CONF POSIX GROUP SETTINGS

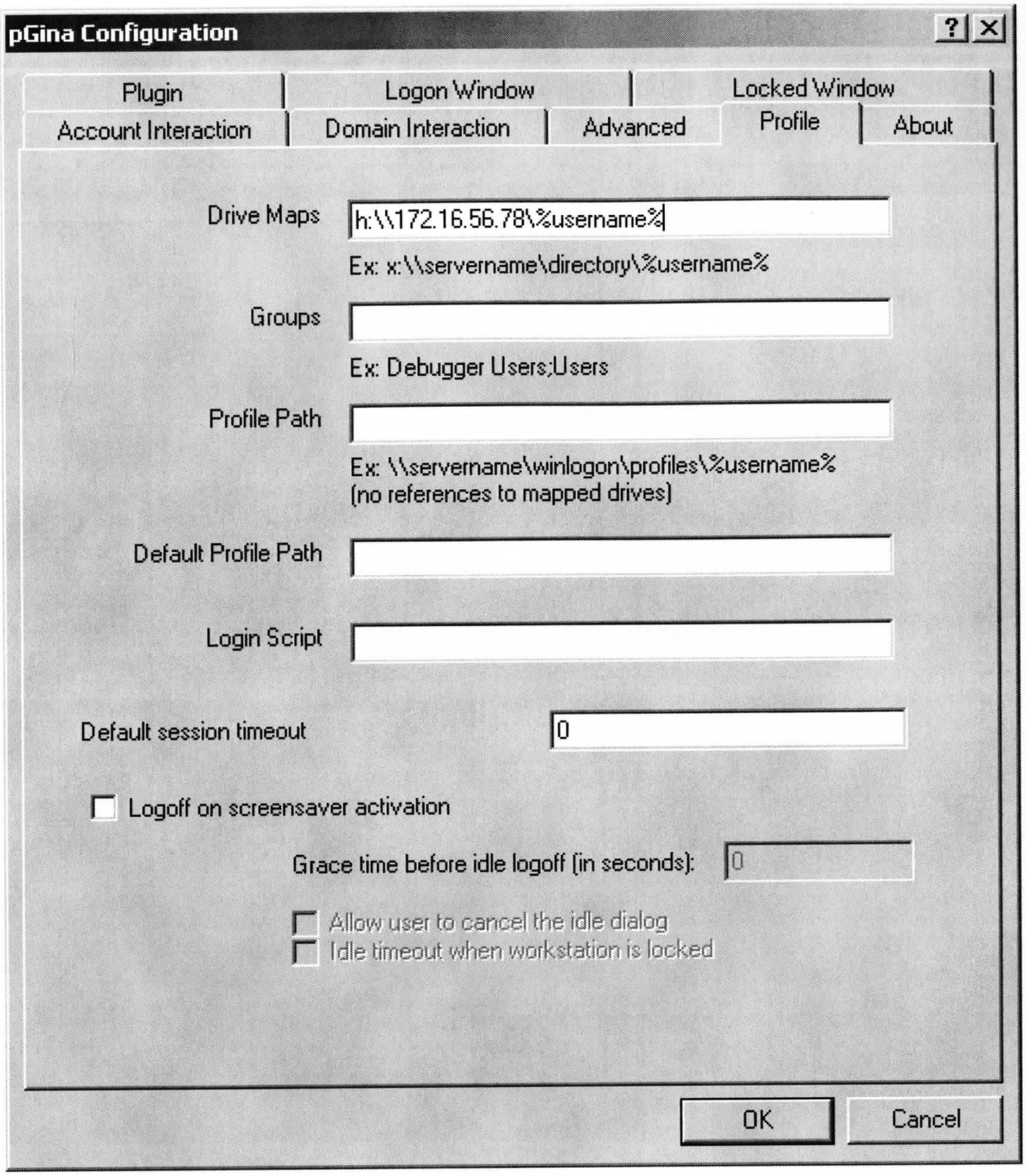
[district]
comment = district
valid users = @district

TABLE 6 CONTINUED: SMB.CONF POSIX GROUP SETTINGS

admin users = @district
path = /export/shared/district
read only = no

PGINA MICROSOFT WINDOWS CLIENT AUTHENTICATION SETUP

Microsoft allows only a single method of authentication through Active Directory. pGina software customized the Graphical Identification and Authentication (GINA) dynamic link library allowing plug-ins to be loaded for other forms of authentication. Using LDAP Auth module allows Microsoft Windows clients to use crypt passwords stored in the Sun Java System (LDAP) Directory Server. pGina allows for both local and Terminal Services Windows authentication while still mapping the Samba home and shared directories (12). Caledonia Area Public Schools uses Open Source pGina software for user authentication on all Microsoft Windows operating systems throughout the district. Figures One through Three displays the necessary settings for pGina to function in the district's heterogeneous network after client installation.



Drive Maps h:\\172.16.56.78\%username%

Ex: x:\\servername\directory\%username%

Groups

Ex: Debugger Users;Users

Profile Path

Ex: \\servername\winlogon\profiles\%username%
(no references to mapped drives)

Default Profile Path

Login Script

Default session timeout 0

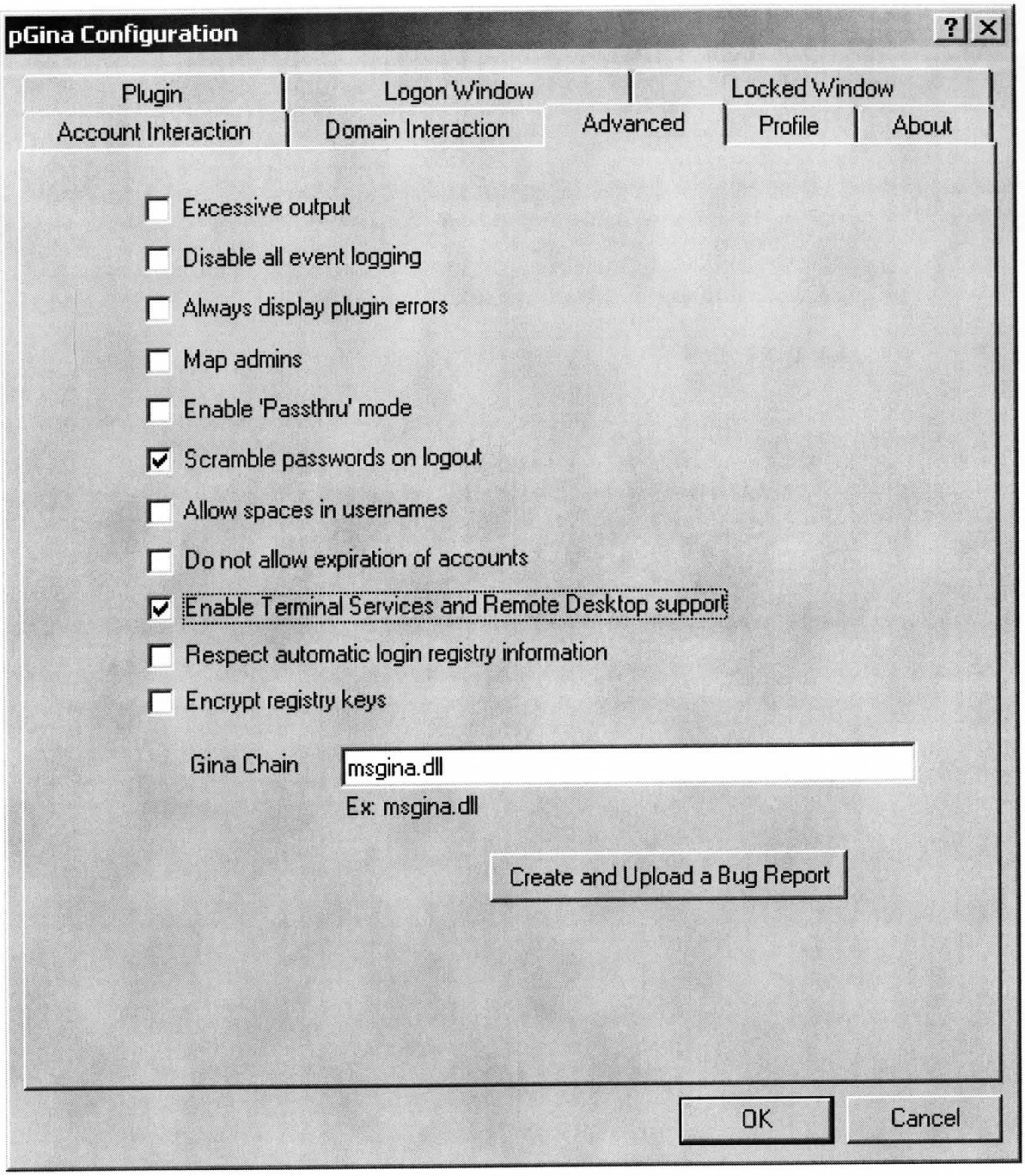
□ Logoff on screensaver activation

Grace time before idle logoff (in seconds): 0

□ Allow user to cancel the idle dialog

□ Idle timeout when workstation is locked

FIGURE 1: PGINA PROFILE CONFIGURATION TAB



□ Excessive output

□ Disable all event logging

□ Always display plugin errors

□ Map admins

□ Enable 'Passthru' mode

☑ Scramble passwords on logout

□ Allow spaces in usernames

□ Do not allow expiration of accounts

☑ Enable Terminal Services and Remote Desktop support

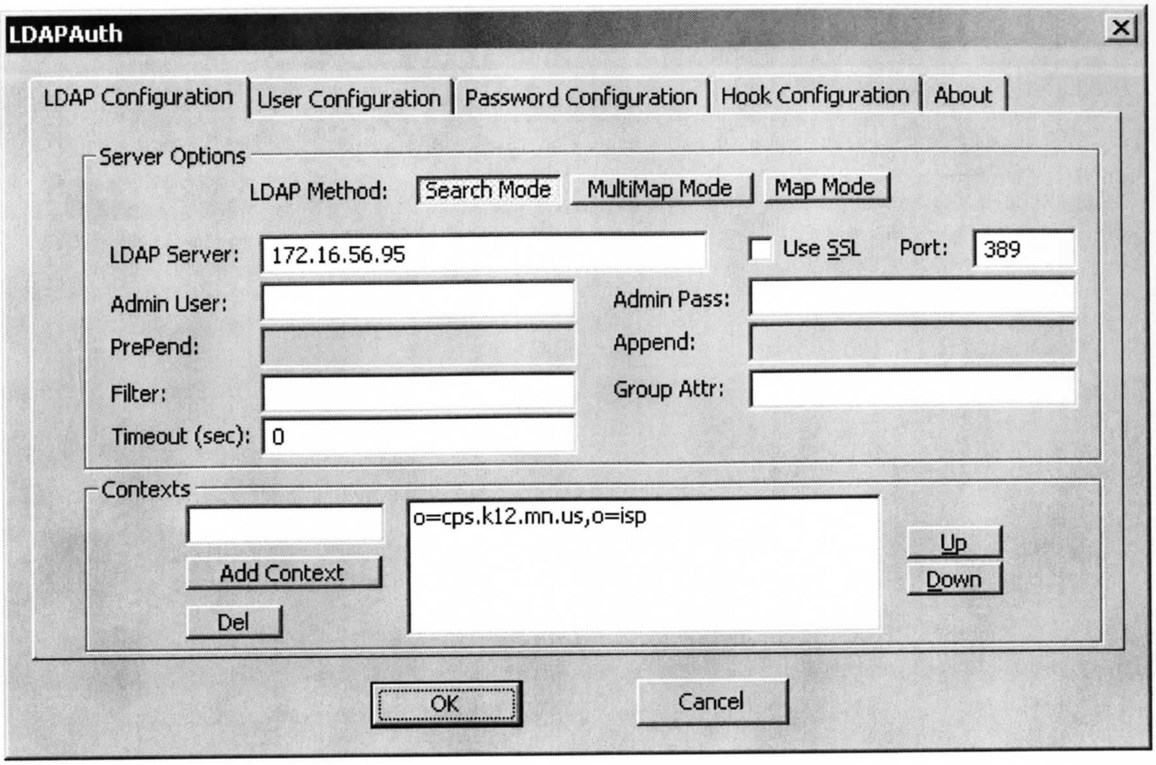
□ Respect automatic login registry information

□ Encrypt registry keys

Gina Chain: msgina.dll

Ex: msgina.dll

FIGURE 2: PGINA ADVANCED CONFIGURATION TAB



LDAP Method: Search Mode

LDAP Server: 172.16.56.95

□ Use SSL

Port: 389

Admin User:

Admin Pass:

PrePend:

Append:

Filter:

Group Attr:

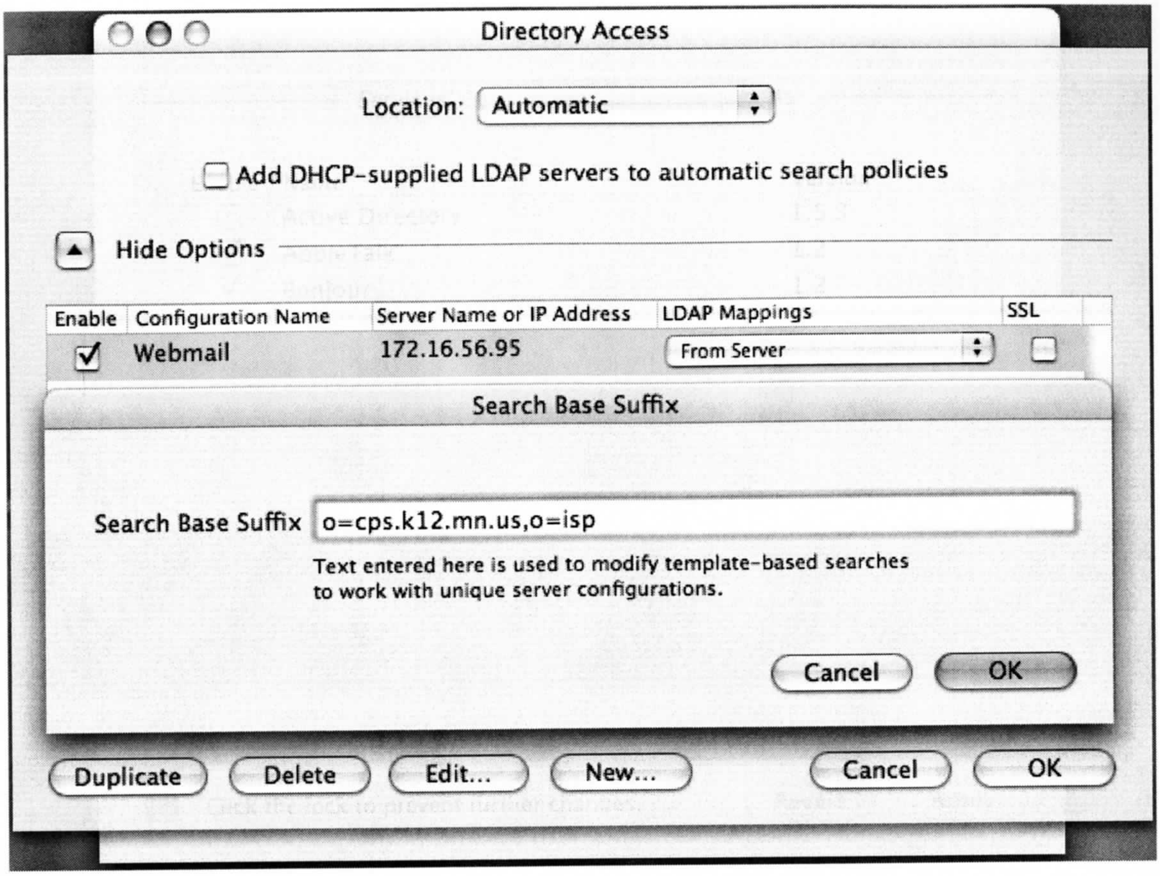
Timeout (sec): 0

o=cps.k12.mn.us,o=isp

FIGURE 3: LDAPAUTH LDAP CONFIGURATION TAB

APPLE COMPUTER CLIENT AUTHENTICATION SETUP

Apple Computer built a Directory Access tool into their operating system. Directory access allows the computer to read Sun Java System (LDAP) Directory Server by using RFC 2307 (Unix) settings through a simple graphical user interface. Starting in version 10.2 of Mac OS X Server the necessary configuration is displayed in Figure Four. Independent School District 299 had to maintain a lab of old Apple Macintosh computers running operating system (OS) 7.2 and 8.6. These computers can only run Apple Talk protocol to access the network and Internet through a Mac OS X Server. Apple Talk is a suite of network protocols for computer access developed by Apple Computer that is now depreciated (26).



Directory Access

Location: Automatic

□ Add DHCP-supplied LDAP servers to automatic search policies

Hide Options

Enable	Configuration Name	Server Name or IP Address	LDAP Mappings	SSL
☑	Webmail	172.16.56.95	From Server	□

Search Base Suffix

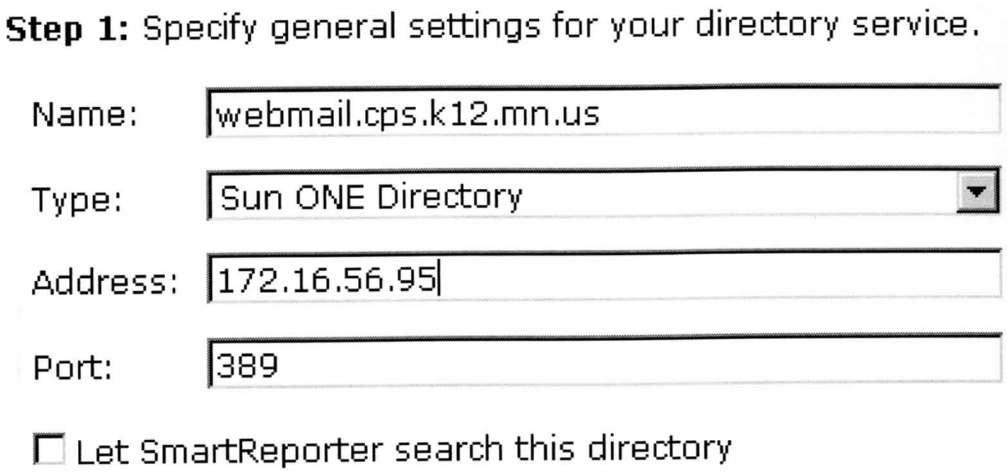
Search Base Suffix: o=cps.k12.mn.us,o=isp

Text entered here is used to modify template-based searches to work with unique server configurations.

FIGURE 4: APPLE DIRECTORY ACCESS SETUP

SMART FILTER WEB FILTERING SETUP

Smart Filter web filtering is a software product from Secure Computer that filters, blocks, manages, and reports on Internet usage of each end-user (27). The Children's Internet Protection Act (CIPA) for public schools requires Internet filtering (28). Smart Filter has a web-based application for setting up directory access. By entering the following information: Name, Type, Address, Port, Base DN, User RDN, User Key, Admin DN, and Password then the Sun Java System (LDAP) Directory Server is accessible. Figure Five through Figure Seven displays the system configuration that is not documented by the product help files. The benefit is the usernames are listed instead of computer Internet Protocol (IP) address for every web page visited. Internet Protocol (IP) address is a unique number that devices use in order to identify and communicate with other computers on the network (29).



Step 1: Specify general settings for your directory service.

Name: webmail.cps.k12.mn.us

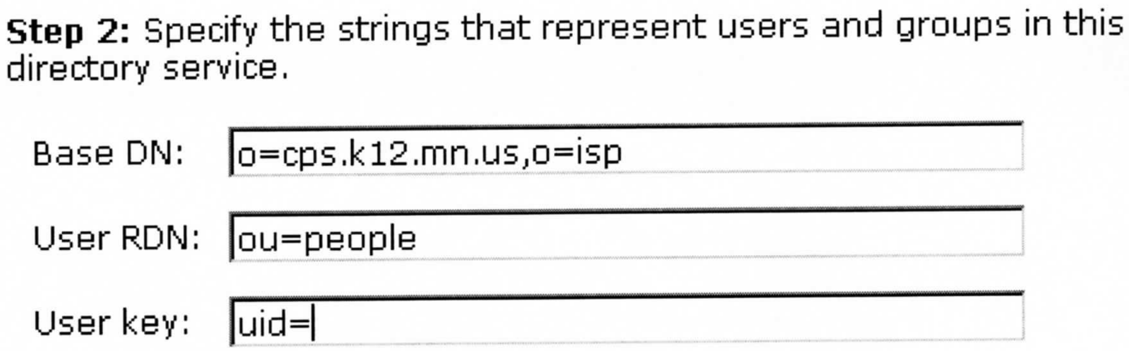
Type: Sun ONE Directory

Address: 172.16.56.95

Port: 389

□ Let SmartReporter search this directory

FIGURE 5: SMART FILTER DIRECTORY SETUP STEP ONE



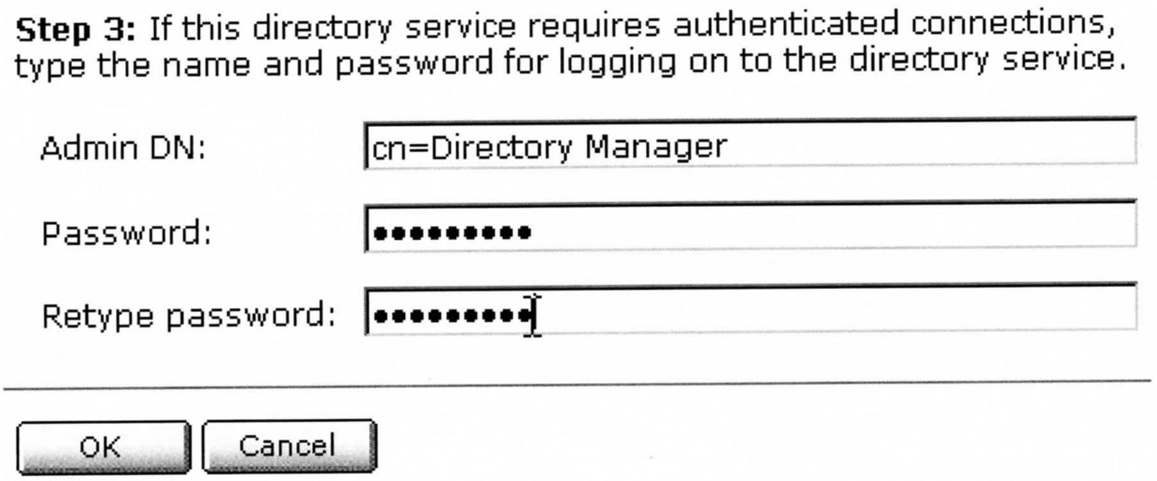
Step 2: Specify the strings that represent users and groups in this directory service.

Base DN: o=cps.k12.mn.us,o=isp

User RDN: ou=people

User key: uid=

FIGURE 6: SMART FILTER DIRECTORY SETUP STEP TWO



Step 3: If this directory service requires authenticated connections, type the name and password for logging on to the directory service.

Admin DN: cn=Directory Manager

Password: ●●●●●●●●●●

Retype password: ●●●●●●●●●●

FIGURE 7: SMART FILTER DIRECTORY SETUP STEP THREE

CONCLUSIONS

Caledonia Area Public Schools needed free web-based account management software to manage its heterogeneous computer system. Information needed to be generated, stored, changed, and deleted in the centralized Lightweight Directory Access Protocol (LDAP) Directory server for the Email server, Calendar server, Sun Rays thin clients, Microsoft Windows computers, Cisco networking equipment, Apple computers, Samba file server, Smart Filter web filtering, and Roaring Penguin Can IT Pro spam/anti-virus server. Independent School District 299 evaluated Sun Java Identity Management Suite commercially available software that had missing features plus was cost prohibitive. The district looked at using Open Source software from Source Forge called phpLDAPadmin and LAM. Neither software packages were close to having the necessary functionality. System management at Caledonia Area Public Schools was labor and time intensive to manage user accounts and groups for the lone technology staff member thus a homegrown web application called PHPLDAP was developed to meet the district's needs.

The major challenge of the PHPLDAP project was to create a computer system that allowed existing and future equipment to cost effectively work together. The five parts of the project focused on creating a single sign-on integration of the schools computer system: Sun Java System (LDAP) Directory server setup, Samba file server setup, pGina Microsoft Windows client authentication setup, Apple Computer client authentication setup, and Smart Filter web filtering setup. System design described the necessary setup to repeat the deployment.

PHPLDAP development took a close look at the exact schema of attributes and entries necessary for the Sun Java System (LDAP) Directory Server to work with all software and hardware. Samba File Server software was setup to work against the directory server to provide free file services to Sun Ray thin clients, Microsoft Windows computers, and Apple computers. Microsoft Windows was inflexible to work with others authentication methods requiring the use of pGina Client software for authentication. Apple Computer and Smart Filter made great efforts to be compatible with the Sun directory software but never documented the solution.

PHPLDAP was successful at reducing support time, license fees, and management requirements of Caledonia Area Public School's computer system. PHPLDAP's next version will happen mid-July 2006 with the renewal of Roaring Penguin Can-IT Pro Spam/Anti-Virus software license. The school district's Intranet web server needs to be reconfigured to support Secure Shell (SSH) functionality and Secure Sockets Layer (SSL) security. Secure Shell (SSH) is a network protocol that allows a secure channel between a local computer and a remote server (30). Secure Sockets Layer (SSL) is a cryptographic protocol that provides secure communications on the Internet (31). PHPLDAP already outputs the Samba, Webmail, and NFS server commands necessary. PHPLDAP will consist of adding the PHP SSH code instead of printing the commands to the screen as shown in Table 7.

TABLE 7: PHP SSH CODE

//Open SSH Tunel
$con=ssh2_connect('172.16.56.95',22);
ssh2_auth_password($con,"user","password");
$shell=ssh2_shell($con,'xterm');
//Executes SSH Commands
fwrite($shell,"pwd\n");
fwrite($shell,"ls\n");
//Close SSH Tunel
fclose($shell);

PHPLDAP will be released as Open Source software. The basic idea behind open source is very simple: Open Source software is when programmers can read, redistribute, and modify the source code for a piece of software, the software evolves (32). Caledonia Area Public Schools development work will be donated to the phpLDAPadmim project, LDAP Account Manager project, and any other interested parties to help promote advancements in LDAP account and group management.

Caledonia Area Public School's cost effective and fully integrated technology deployment has become a showcase for other schools and businesses to copy. Most organizations unfortunately do not have the technology skills or resources necessary to reach a single sign-on environment. To make PHPLDAP marketable the district needs to add installation and configuration scripts to the existing management software. A Small School-Business Appliance (SSBA) server could include the following services:

- Directory Services
- Email Services
- Calendar Services
- Unified Web Client (UWC) Services
- Instant Messaging Services
- Can-IT Spam/Anti-Virus Services
- PHPLDAP Management Services
- Samba File Services
- DNS Services

A Solaris 10 image will be used to create a base install image of the server operating system and software. Solaris is a Unix server operating system released by Sun Microsystems (33). PHPLDAP installation will be run to setup and configure the appliance server and all services. The Small School-Business Appliance (SSBA) server will only use free software supporting up to fifty users. PHPLDAP allows organizations a simple cost effective plug and play solution to their high-end technology needs with options for future growth.

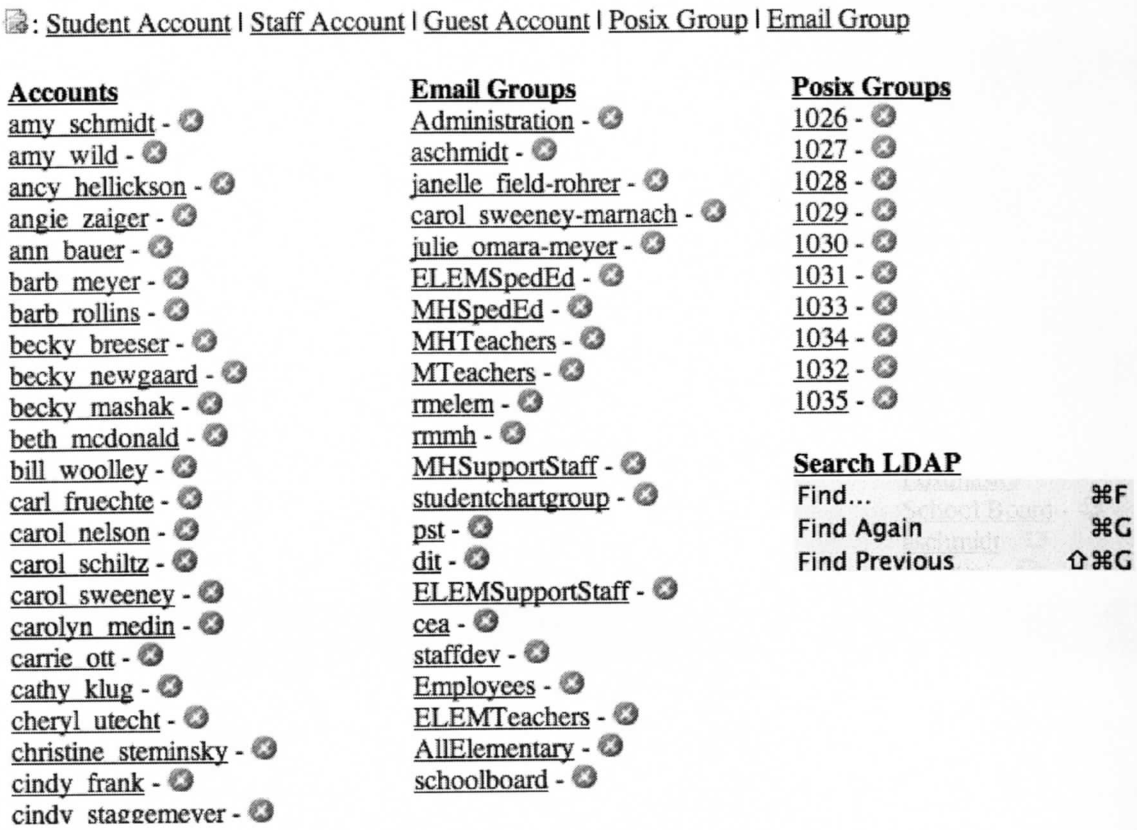
REFERENCES

1. Lightweight Directory Access Protocol (LDAP) definition. http://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol
2. Sun Ray Thin Client definition. http://en.wikipedia.org/wiki/Sun_Ray.
3. Samba File Server definition. http://en.wikipedia.org/wiki/Samba_software.
4. Sun Java Identity Management Suite. http://www.sun.com/software/javaenterprisesystem/identity_mgmt_suite/.
5. SourceForge.net. http://sourceforge.net/docs/about
6. phpLDAPadmin. http://phpldapadmin.sourceforge.net/.
7. LDAP Account Manager (LAM). http://lam.sourceforge.net/.
8. Microsoft Windows 2003 Server System. http://www.microsoft.com/windowsserver2003/evaluation/overview.
9. Apple – Mac OS X Server v10.4. http://www.apple.com/server/macosx/.
10. Sun Java Communications Suite. http://www.sun.com/software/javaenterprisesystem/communications_suite.
11. Microsoft Windows Exchange Server. http://www.microsoft.com/exchange.
12. pGina Making the big boys play nice. http://pgina.xpasystems.com.
13. PHP: Hypertext Preprocessor definition. http://en.wikipedia.org/wiki/Php.
14. Ross, Michael and Rubin, Jeff. "Authentication Gets Tough." Network Computing; May 28, 2001; 12,11; ProQuest Education Journals. pg. 97.
15. Java Enterprise System (JES). http://www.sun.com/software/javaenterprisesystem.
16. Sun Welcomes Waveset Customers and Partners. http://www.sun.com/software/waveset/.
17. Network File System definition. http://en.wikipedia.org/wiki/Network_File_System.
18. PHP: Hypertext Preprocessor. http://php.net.
19. Posix definition. http://en.wikipedia.org/wiki/Posix.
20. Solaris Enterprise System. http://www.sun.com/software/solaris/.

21. Sun Java System Directory Server 5 2005Q4. http://docs.sun.com/app/docs/coll/1316.1.
22. Chapter 11 Setting Up Sun Java System Directory Server With LDAP Client. http://docs.sun.com/app/docs/doc/816-4556/6maort2sc?q=idsconfig&a=view.
23. Sun ONE Directory Server 5.2 Reference Manual: Chapter 9 About Schema. http://docs.sun.com/source/816-6699-10/schemaov.html.
24. Internet Assigned Numbers Authority (IANA). http://www.iana.org/.
25. The Official Samba-3 HOWTO and Reference Guide. http://us2.samba.org/samba/docs/man/Samba-HOWTO-Collection.
26. Apple Talk definition. http://en.wikipedia.org/wiki/AppleTalk.
27. Secure Computing: Smart Filter web URL filtering and reporting. http://securecomputing.com/index.cfm?skey=85.
28. Children's Internet Protection Act (CIPA). http://www.fcc.gov/cgb/consumerfacts/cipa.html.
29. Internet Protocol (IP) Address definition. http://en.wikipedia.org/wiki/IP_Address.
30. Secure Shell (SSH) definition. http://en.wikipedia.org/wiki/SSH.
31. Secure Sockets Layer (SSL) definition. http://en.wikipedia.org/wiki/Secure_Sockets_Layer.
32. Open Source Initiative (OSI). http://www.opensource.org/.
33. Solaris Operating Environment definition. http://en.wikipedia.org/wiki/Solaris_Operating_Environment.
34. Unix definition. http://en.wikipedia.org/wiki/Unix.

APPENDIX A: USERS' MANUAL

The index page (Figure 8) contains links for adding, editing, and deleting objects in the directory server. Clicking on Student Account, Staff Account, Guest Account, Posix Group, or Email Group at the top of the page allows you to create a new directory object. Accounts category is a list of student, staff, and guest objects under ou=people,o=cps.k12.mn.us,o=isp with the exception of admin, msg-admin-webmail.cps.k12.mn.us-20050112224950Z, and calmaster that are server required accounts. Email Groups is a list of objects under Groups,o=cps.k12.mn.us,o=isp with the exception of Domain Administrators and Postmaster that are server required groups. Posix Groups is a list of objects under group,o=cps.k12.mn.us,o=isp that are used for file server permissions. Clicking on the object name takes you to the acceptable editing attributes options where the red x will allow you to delete. Sophisticated search tools were not necessary as Caledonia Area Public Schools is a small organization. Built in browser functionality is effective for the districts searching needs.



: Student Account | Staff Account | Guest Account | Posix Group | Email Group

Accounts
amy_schmidt -
amy_wild -
ancy_hellickson -
angie_zaiger -
ann_bauer -
barb_meyer -
barb_rollins -
becky_breeser -
becky_newgaard -
becky_mashak -
beth_mcdonald -
bill_woolley -
carl_fruechte -
carol_nelson -
carol_schiltz -
carol_sweeney -
carolyn_medin -
carrie_ott -
cathy_klug -
cheryl_utecht -
christine_steminsky -
cindy_frank -
cindy_staggemeyer -

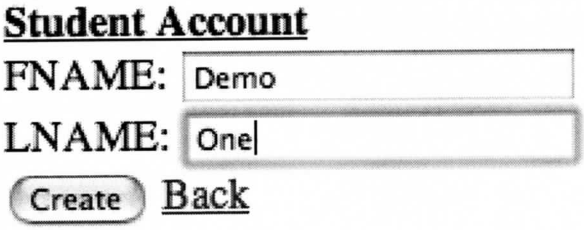
Email Groups
Administration -
aschmidt -
janelle_field-rohrer -
carol_sweeney-marnach -
julie_omara-meyer -
ELEMSpedEd -
MHSpedEd -
MHTeachers -
MTeachers -
rmelem -
rmmh -
MHSupportStaff -
studentchartgroup -
pst -
dit -
ELEMSupportStaff -
cea -
staffdev -
Employees -
ELEMTeachers -
AllElementary -
schoolboard -

Posix Groups
1026 -
1027 -
1028 -
1029 -
1030 -
1031 -
1033 -
1034 -
1032 -
1035 -

Search LDAP

FIGURE 8: PHPLDAP INDEX.PHP

To create a new user account, click Student Account to bring up Figure 9. Enter the first and last name, as you want them to appear with proper capitalization. Click Create or Back to return to the main index page. The account creation process is the same for Student, Staff, and Guest accounts. Staff accounts get 100MB of email storage where as student accounts get only 10 MB is the only difference. Guest accounts have no email or calendar service access just computer access with all other district services.



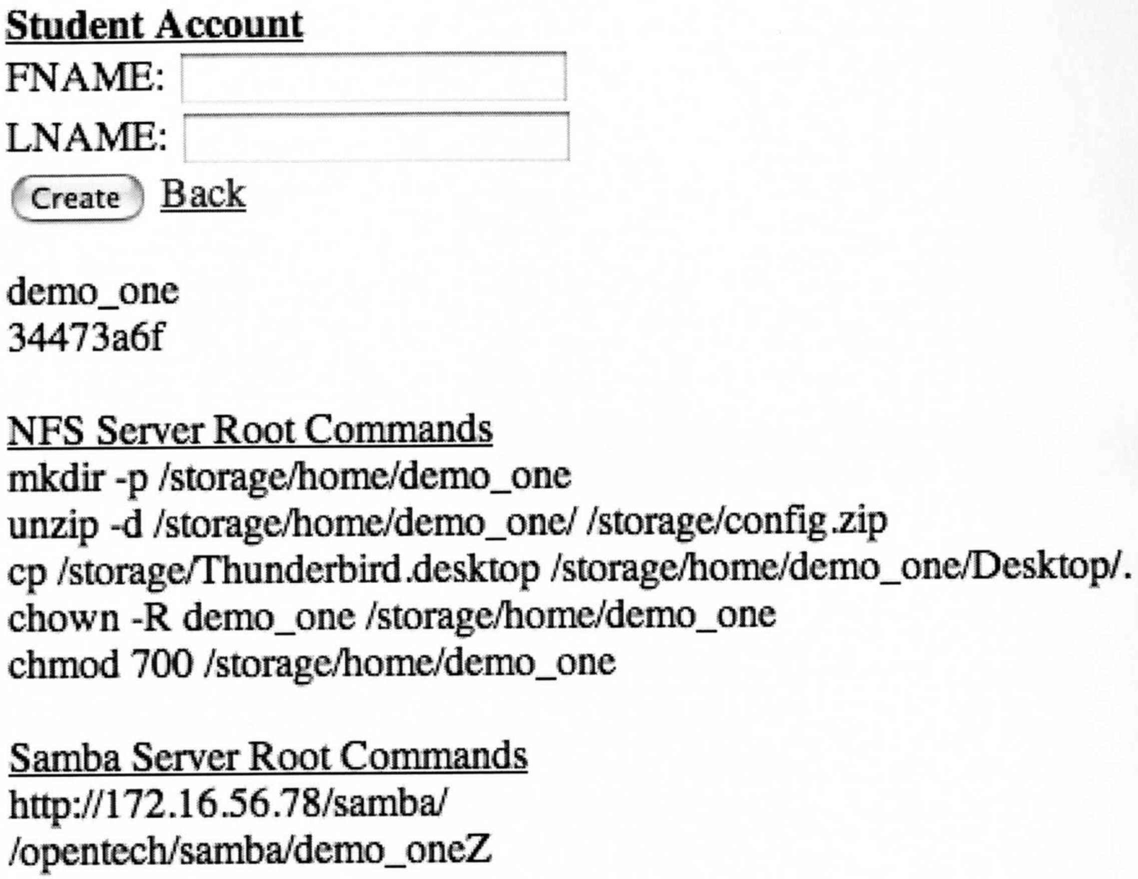
Student Account

FNAME: Demo

LNAME: One

FIGURE 9: PHPLDAP STUDENT.PHP

After clicking Create button the program returns to previous location so you can create another student account. By default the program will display the username and auto generated password to print for the student's records. Notice that there are still additional account creations commands that still need to be run until next software upgrade (Figure 10).



demo_one
34473a6f

NFS Server Root Commands

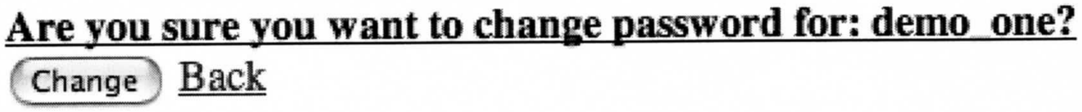
```
mkdir -p /storage/home/demo_one
unzip -d /storage/home/demo_one/ /storage/config.zip
cp /storage/Thunderbird.desktop /storage/home/demo_one/Desktop/.
chown -R demo_one /storage/home/demo_one
chmod 700 /storage/home/demo_one
```

Samba Server Root Commands

```
http://172.16.56.78/samba/
/opentech/samba/demo_oneZ
```

FIGURE 10: PHPLDAP STUDENT.PHP

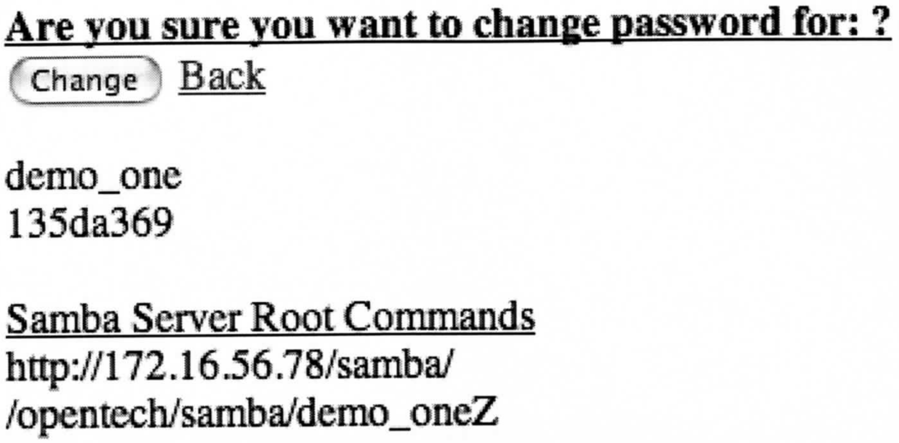
Once an account is created the only thing you are allowed to change through the program is the user's password. Click on the username to edit the account bringing up Figure 11. Click the Change button or Back to cancel out of the password change function.



Are you sure you want to change password for: demo_one?

FIGURE 11: PHPLDAP EDITACCT.PHP

The program will display the user's new account password after clicking the Create button. This password is used for everything internally on the Independent School District 299's network since we have reached a single sign-on system (Figure 12).



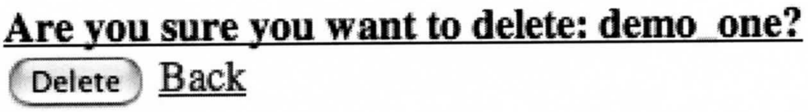
Are you sure you want to change password for: ?

demo_one
135da369

Samba Server Root Commands
http://172.16.56.78/samba/
/opentech/samba/demo_oneZ

FIGURE 12: PHPLDAP EDITACCT.PHP

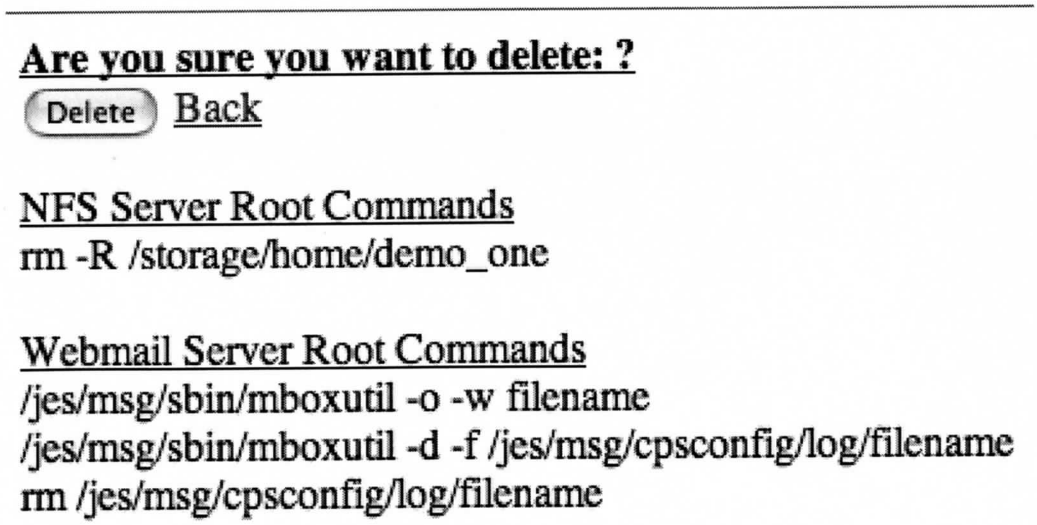
To delete an account you click the red x that appears next to the name. This will load the page in Figure 13. This is for error checking to make sure you really want to delete the account. Once you delete an account there is no reversal of this operation.



Are you sure you want to delete: demo_one?

FIGURE 13: PHPLDAP DELACCT.PHP

The new version of PHPLDAP will no longer have Figure 14.



Are you sure you want to delete: ?

NFS Server Root Commands

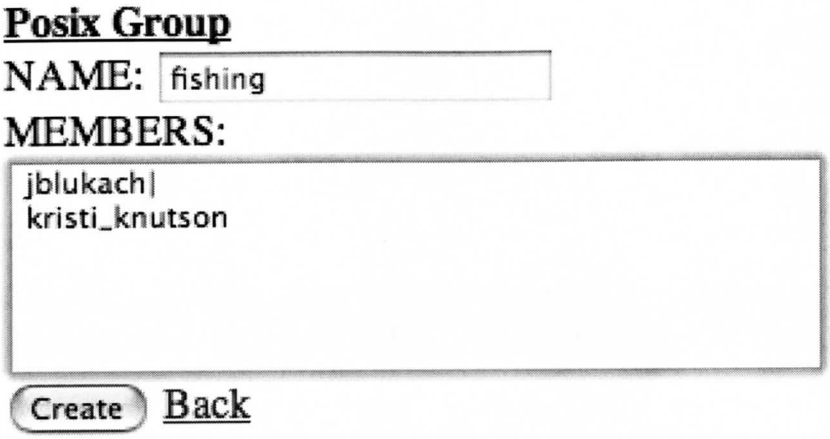
```
rm -R /storage/home/demo_one
```

Webmail Server Root Commands

```
/jes/msg/sbin/mboxutil -o -w filename
/jes/msg/sbin/mboxutil -d -f /jes/msg/cpsconfig/log/filename
rm /jes/msg/cpsconfig/log/filename
```

FIGURE 14: PHPLDAP DELACCT.PHP

To create a posix group, enter the name of the group. Then enter the members of the group separated by the "|" pipe symbol. You must have at least one member and the last one can't have a pipe following it. Click Create button when finished entering information (Figure 15).



Posix Group

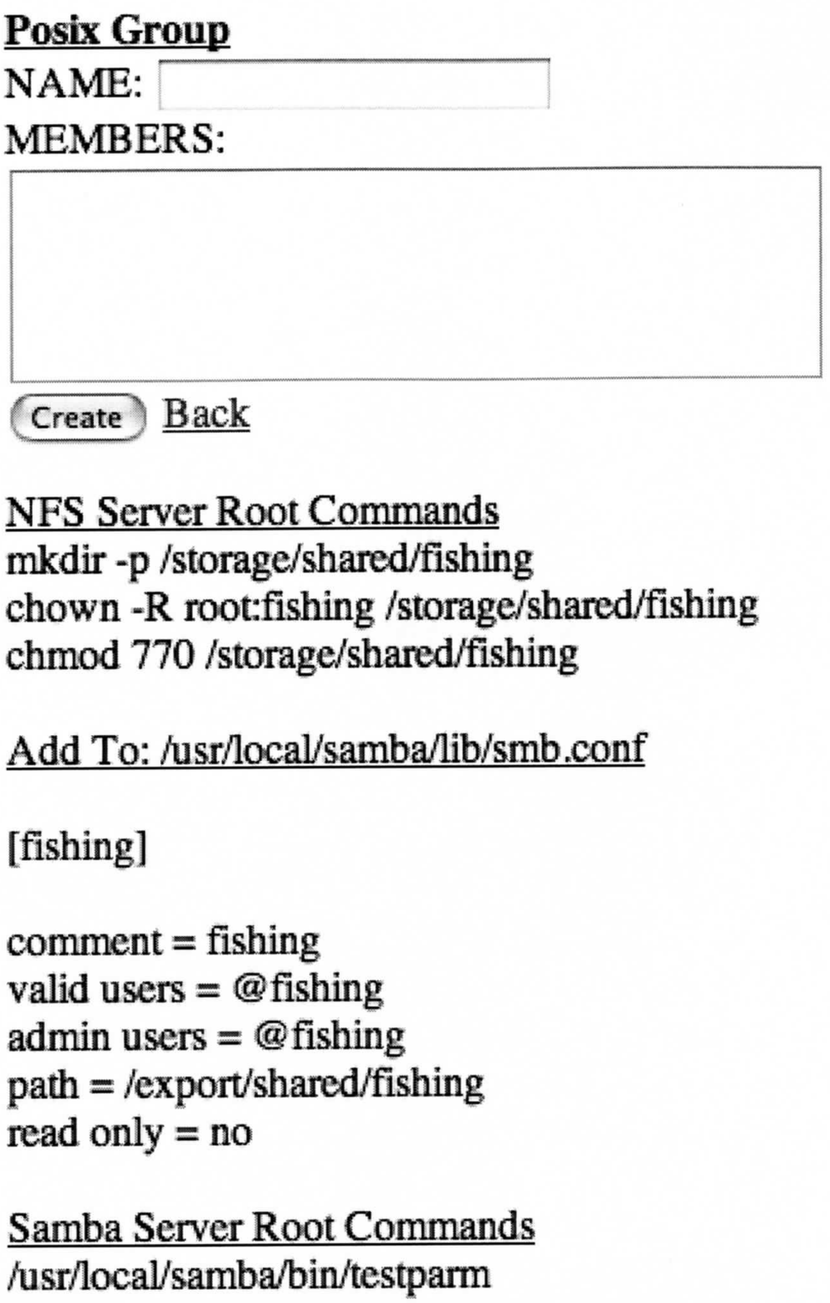
NAME: fishing

MEMBERS:

jblukach|
kristi_knutson

FIGURE 15: PHPLDAP POSIX.PHP

After clicking Create button the program returns to previous location so you can create another posix group. This part of the program will create a Samba share that is accessible to all Apple, Microsoft, and Sun Ray clients (Figure 16).



Posix Group

NAME:

MEMBERS:

NFS Server Root Commands

```
mkdir -p /storage/shared/fishing
chown -R root:fishing /storage/shared/fishing
chmod 770 /storage/shared/fishing
```

Add To: /usr/local/samba/lib/smb.conf

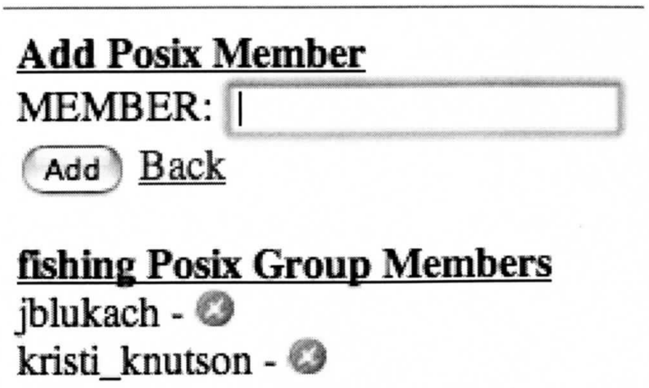
```
[fishing]

comment = fishing
valid users = @fishing
admin users = @fishing
path = /export/shared/fishing
read only = no
```

Samba Server Root Commands

```
/usr/local/samba/bin/testparm
```

FIGURE 16: PHPLDAP POSIX.PHP

Once you create a posix group the only thing you are allowed to change is the group membership. Enter the username in the member field and click the Add button or Back to cancel and return to the index page. To remove a group member click on the red x but be careful since there is no error checking on group membership removal (Figure 17).



Add Posix Member

MEMBER: |

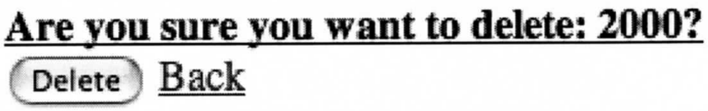
fishing Posix Group Members

jblukach -

kristi_knutson -

FIGURE 17: PHPLDAP EDITPOSIX.PHP

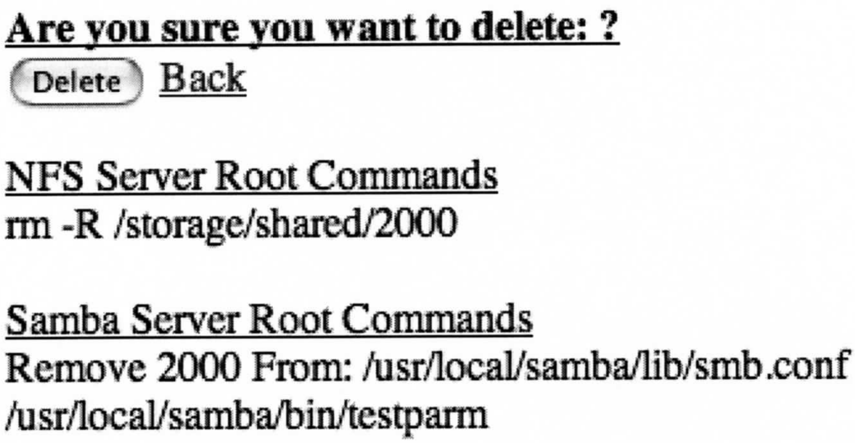
To delete a posix group click the red x next to the name. This will load the page in Figure 18. This is for error checking to make sure you really want to delete the posix group. Once a posix group is deleted there is no reversal of this operation.



Are you sure you want to delete: 2000?

FIGURE 18: PHPLDAP DELPOSIX.PHP

The new version of PHPLDAP will no longer have Figure 19.



Are you sure you want to delete: ?

NFS Server Root Commands

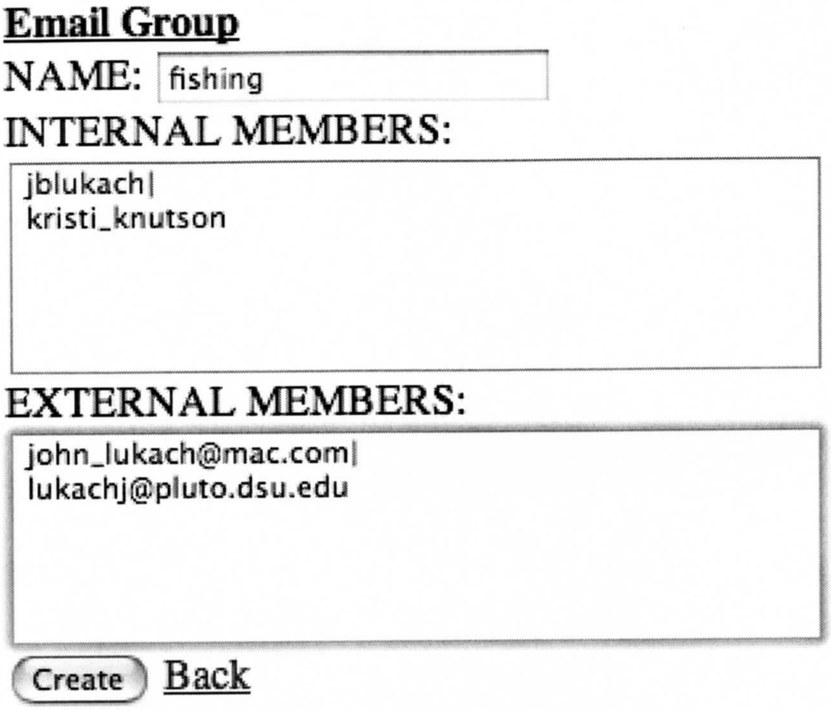
```
rm -R /storage/shared/2000
```

Samba Server Root Commands

```
Remove 2000 From: /usr/local/samba/lib/smb.conf
/usr/local/samba/bin/testparm
```

FIGURE 19: PHPLDAP DELPOSIX.PHP

To create an Email Group enter the name of the group. An email group can have both internal and external member. Then enter the members of the group separated by the "|" pipe symbol. You must have at least one member and the last one can't have a pipe following it. Click Create button when finished entering information. For internal members enter the users network username where as the external members must have the entire email address including the fully qualified domain as shown in the example (Figure 20).



Email Group

NAME: fishing

INTERNAL MEMBERS:

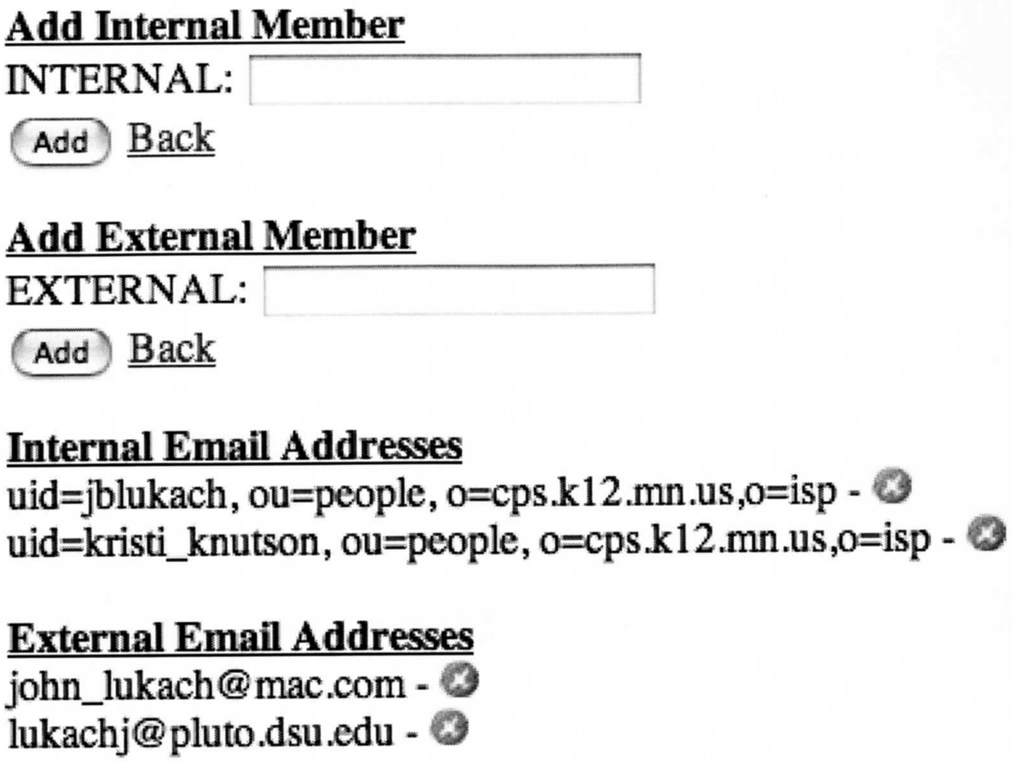
jblukach|
kristi_knutson

EXTERNAL MEMBERS:

john_lukach@mac.com|
lukachj@pluto.dsu.edu

FIGURE 20: PHPLDAP EMAIL.PHP

Once you create an email group the only thing that can be changed is both the internal and external group membership. Enter their username in either the internal or external member field and click the Add button or Back to cancel and return to the index page. To remove a group member click on the red x but be careful since there is no error checking on group membership removal (Figure 21).



Add Internal Member

INTERNAL:

Add External Member

EXTERNAL:

Internal Email Addresses

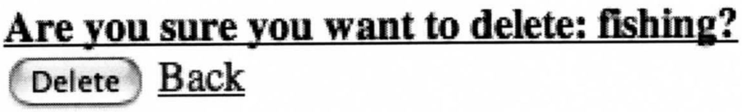
uid=jblukach, ou=people, o=cps.k12.mn.us,o=isp - ⊗
uid=kristi_knutson, ou=people, o=cps.k12.mn.us,o=isp - ⊗

External Email Addresses

john_lukach@mac.com - ⊗
lukachj@pluto.dsu.edu - ⊗

FIGURE 21: PHPLDAP EDITEMAIL.PHP

To delete an email group click the red x next to the name. This will load the page in Figure 22. This is for error checking to make sure you really want to delete the email group. Once you delete an email group there is no retrieving the group.



Are you sure you want to delete: fishing?

FIGURE 22: PHPLDAP DELEMAIL.PHP

APPENDIX B: PROGRAMMING CODE

PHPLDAP: INDEX.PHP

```
<?php
//phpLDAP ALPHA for ISD 299
//by John B. Lukach

//LDAP Connection Information
$ldaphost = "172.16.56.95";
$ldapport = "389";
$ldapuser = "cn=Directory Manager";
$ldappass = "";

//LDAP Bind Functions
$ldapcon = ldap_connect($ldaphost,$ldapport);
$ldapbind = ldap_bind($ldapcon, $ldapuser, $ldappass);

//Add LDAP Entry Links
echo "<img src=add.gif>: <A HREF=student.php>Student Account</A>
| <A HREF=staff.php>Staff Account</A>
| <A HREF=guest.php>Guest Account</A>
| <A HREF=posix.php>Posix Group</A>
| <A HREF=email.php>Email Group</A><BR><BR>";

?>
<TABLE WIDTH=100%>
        <TR>
                <TD>

                        <!-- ACCOUNTS -->
                        <?

                        //LDAP List Information
                        $basedn = "ou=people,o=cps.k12.mn.us,o=isp";
                        $justthese = array("uid");

                        //LDAP List Functions
                        $sr=ldap_list($ldapcon, $basedn, "uid=*", $justthese);
                        $info = ldap_get_entries($ldapcon, $sr);

                        //Display LDAP List
                        echo "<B><U>Accounts</U></B><BR>";
                        for ($i=0; $i<$info["count"]; $i++) {
```

```
                    //Sort Out Required Accounts
                    if(trim($info[$i]["uid"][0]) != "admin" and
trim($info[$i]["uid"][0]) != "msg-admin-webmail.cps.k12.mn.us-20050112224950Z" and
trim($info[$i]["uid"][0]) != "calmaster" and trim($info[$i]["uid"][0]) != "jblukach"){
                    echo "<A
HREF=editacct.php?id=".trim($info[$i]["uid"][0]).">".trim($info[$i]["uid"][0])."</A> -
<A HREF=delacct.php?id=".trim($info[$i]["uid"][0]).">"<img src=delete.gif></A><br>";
                    }
                }

                ?>

            </TD>
            <TD VALIGN="TOP">

                <!-- EMAIL -->
                <?

                //LDAP List Information
                $basedn = "ou=Groups,o=cps.k12.mn.us,o=isp";
                $justthese = array("cn");

                //LDAP List Functions
                $sr=ldap_list($ldapcon, $basedn, "cn=*", $justthese);
                $info = ldap_get_entries($ldapcon, $sr);

                //Display LDAP List
                echo "<B><U>Email Groups</U></B><BR>";
                for ($i=0; $i<$info["count"]; $i++) {
                    //Sort Out Required Accounts
                    if(trim($info[$i]["cn"][0]) != "Domain
Administrators" and trim($info[$i]["cn"][0]) != "Postmaster"){
                    echo "<A
HREF=editemail.php?id=".trim($info[$i]["cn"][0]).">".trim($info[$i]["cn"][0])."</A> -
<A HREF=delemail.php?id=".trim($info[$i]["cn"][0]).">"<img
src=delete.gif></A><br>";
                    }
                }

                ?>

            </TD>
            <TD VALIGN="TOP">

                <!-- POSIX -->
                <?
                //LDAP List Information
                $basedn = "ou=group,o=cps.k12.mn.us,o=isp";
                $justthese = array("gidnumber");

                //LDAP List Functions
                $sr=ldap_list($ldapcon, $basedn, "gidnumber=*",
$justthese);
                $info = ldap_get_entries($ldapcon, $sr);

                //Display LDAP List
                echo "<B><U>Posix Groups</U></B><BR>";
                for ($i=0; $i<$info["count"]; $i++) {
                echo "<A
HREF=editposix.php?id=".trim($info[$i]["gidnumber"][0]).">".trim($info[$i]["gidnumb
er"][0])."</A> - <A HREF=delposix.php?id=".trim($info[$i]["gidnumber"][0])."><img
src=delete.gif></A><br>";
                }

                //LDAP Search Commands
                echo "<BR><B><U>Search LDAP</U></B><BR>";
                echo "<img src=search.gif>";
                ?>

            </TD>
        </TR>
    </TABLE>

    <?

    //Close LDAP Connection
    ldap_close($ldapbind);
    ?>
```

PHPLDAP: STUDENT.PHP

```
    <?php
    //phpLDAP ALPHA for ISD 299
    //by John B. Lukach

    //LDAP Connection Information
    $ldaphost = "172.16.56.95";
    $ldapport = "389";
    $ldapuser = "cn=Directory Manager";
    $ldappass = "";
```

```
//LDAP Bind Functions
$ldapcon = ldap_connect($ldaphost,$ldapport);
$ldapbind = ldap_bind($ldapcon, $ldapuser, $ldappass);

?>
<!-- Create LDIF Information-->
<FORM ACTION=student.php METHOD=post>
<U><B>Student Account</B></U><BR>
FNAME: <INPUT TYPE=text NAME=fname><BR>
LNAME: <INPUT TYPE=text NAME=lname><BR>
<INPUT TYPE=submit VALUE=Create> <A HREF=index.php>Back</A>
</FORM>
<?

if($_POST["fname"] != "" and $_POST["lname"] != ""){

        //Reads uidnumber.txt File
        $fp = fopen("/Users/jblukach/Sites/uidnumber.txt",'r');
        while(!feof($fp)){
                $uidnumber = trim(fgets($fp, 999));
        }
        fclose($fp);

        //Data Clean Up For LDIF Creation
        $fname = trim($_POST["fname"]);
        $lname = trim($_POST["lname"]);
        $lowerfn = strtolower($fname);
        $lowerln = strtolower($lname);
        $email = $lowerfn."_".$lowerln."@cps.k12.mn.us";
        echo $uname = $lowerfn."_".$lowerln;
        echo "<BR>";

        //Generate Password
        $pwd = md5(crypt("4student"));
        echo $passwd = substr($pwd, 0, 8);
        echo "<BR><BR>";

        //Create Samba RID Nubmer
        $rid = $uidnumber * 2 + 1000;

        //Creates LDIF Array
        $info["objectClass"][0] = "top";
        $info["objectClass"][1] = "person";
        $info["objectClass"][2] = "inetorgperson";
        $info["objectClass"][3] = "iplanet-am-managed-person";
        $info["objectClass"][4] = "organizationalPerson";
```

```
$info["objectClass"][5] = "inetUser";
$info["objectClass"][6] = "ipUser";
$info["objectClass"][7] = "userPresenceProfile";
$info["objectClass"][8] = "inetMailUser";
$info["objectClass"][9] = "inetLocalMailRecipient";
$info["objectClass"][10] = "icsCalendarUser";
$info["objectClass"][11] = "iplanet-am-user-service";
$info["objectClass"][12] = "inetAdmin";
$info["objectClass"][13] = "iPlanetPreferences";
$info["objectClass"][14] = "posixAccount";
$info["objectClass"][15] = "shadowAccount";
$info["objectClass"][16] = "account";
$info["objectClass"][17] = "sambaAccount";
$info["acctFlags"] = "[U          ]";
$info["domain"] = "";
$info["homeDrive"] = "/export/home";
$info["kickoffTime"] = "0";
$info["lmPassword"] = "96B61488C946472009752A3293831D17";
$info["logoffTime"] = "0";
$info["logonTime"] = "0";
$info["ntPassword"] = "A7FB8A589D7F463136D1B332F2AFB0F6";
$info["primaryGroupID"] = "1041";
$info["profilePath"] = "";
$info["pwdCanChange"] = "0";
$info["pwdLastSet"] = "1114176758";
$info["pwdMustChange"] = "0";
$info["rid"] = $rid;
$info["scriptPath"] = "";
$info["smbHome"] = "\\\\172.16.56.78\homes";
$info["userWorkstations"] = "";
$info["cn"] = $fname." ".$lname;
$info["sn"] = $lname;
$info["uid"] = $uname;
$info["givenName"] = $fname;
$info["uidNumber"] = $uidnumber;
$info["gidNumber"] = "20";
$info["homeDirectory"] = "/export/home/".$uname;
$info["loginShell"] = "/bin/sh";
$info["mail"] = $email;
$info["mailUserStatus"] = "active";
$info["mailQuota"] = "10485760";
$info["mailMsgQuota"] = "-1";
$info["mailHost"] = "webmail.cps.k12.mn.us";
$info["mailDeliveryOption"] = "mailbox";
$info["userPassword"] = $passwd;
$info["icsCalendar"] = $uname;
```

```
                $info["icsSubscribed"] = $uname."$".$fname." ".$lname;
                $info["icsExtendedUserPrefs"][0] = "ceDefaultView=monthview";
                $info["icsExtendedUserPrefs"][1] = "ceInterval=PT0H30M";
                $info["icsExtendedUserPrefs"][2] = "ceExcludeSatSun=0";
                $info["icsExtendedUserPrefs"][3] = "ceGroupInviteAll=1";
                $info["icsExtendedUserPrefs"][4] = "ceSingleCalendarTZID=0";
                $info["icsExtendedUserPrefs"][5] = "ceAllCalendarTZIDs=1";
                $info["icsExtendedUserPrefs"][6] = "ceNotifyEnable=0";
                $info["icsExtendedUserPrefs"][7] = "ceNotifyEmail=".$email;
                $info["icsExtendedUserPrefs"][8] = "ceDefaultAlarmStart=";
                $info["icsExtendedUserPrefs"][9] = "ceDefaultAlarmEmail=".$email;
                $info["icsCalendarOwned"] = $uname."$".$fname." ".$lname;
                $info["inetUserStatus"] = "active";
                $info["icsStatus"] = "active";
                $info["nswmExtendedUserPrefs"][0] = "meDraftFolder=Drafts";
                $info["nswmExtendedUserPrefs"][1] = "meSentFolder=Sent";
                $info["nswmExtendedUserPrefs"][2] = "meTrashFolder=Trash";
                $info["nswmExtendedUserPrefs"][3] = "meInitialized=true";
                $info["pabURI"] =
"ldap://webmail.cps.k12.mn.us:389/ou=".$uname.",ou=people,o=cps.k12.mn.us,o=isp,o=
pab";
                $info["preferredLanguage"] = "en";

                //Creates LDAP Entry
                $dnentry = "uid=".$uname.", ou=people, o=cps.k12.mn.us,o=isp";
                $r=ldap_add($ldapcon, $dnentry, $info);

                //Commands To Run On NFS Server
                echo "<U>NFS Server Root Commands</U><BR>";
                echo "mkdir -p /storage/home/".$uname."<BR>";
                echo "unzip -d /storage/home/".$uname."/ /storage/config.zip<BR>";
                echo "cp /storage/Thunderbird.desktop
/storage/home/".$uname."/Desktop/.<BR>";
                echo "chown -R ".$uname." /storage/home/".$uname."<BR>";
                echo "chmod 700 /storage/home/".$uname."<BR><BR>";

                //Commands To Run On Samba Server
                echo "<U>Samba Server Root Commands</U><BR>";
                echo "http://172.16.56.78/samba/<BR>";
                echo "/opentech/samba/".$uname."Z<BR><BR>";

                //Sets Next Available uidnumber.txt
                ++$uidnumber;
                $fp = fopen("/Users/jblukach/Sites/uidnumber.txt",'w');
                fwrite($fp, $uidnumber, strlen($uidnumber));
                fclose($fp);
```

```
}

//Close LDAP Connection
ldap_close($ldapbind);
?>
```

PHPLDAP: STAFF.PHP

```
<?php
//phpLDAP ALPHA for ISD 299
//by John B. Lukach

//LDAP Connection Information
$ldaphost = "172.16.56.95";
$ldapport = "389";
$ldapuser = "cn=Directory Manager";
$ldappass = "";

//LDAP Bind Functions
$ldapcon = ldap_connect($ldaphost,$ldapport);
$ldapbind = ldap_bind($ldapcon, $ldapuser, $ldappass);

?>
<!-- Create LDIF Information-->
<FORM ACTION=staff.php METHOD=post>
<U><B>Staff Account</B></U><BR>
FNAME: <INPUT TYPE=text NAME=fname><BR>
LNAME: <INPUT TYPE=text NAME=lname><BR>
<INPUT TYPE=submit VALUE=Create> <A HREF=index.php>Back</A>
</FORM>
<?

if($_POST["fname"] != "" and $_POST["lname"] != ""){

    //Reads uidnumber.txt File
    $fp = fopen("/Users/jblukach/Sites/uidnumber.txt",'r');
    while(!feof($fp)){
        $uidnumber = trim(fgets($fp, 999));
    }
    fclose($fp);

    //Data Clean Up For LDIF Creation
    $fname = trim($_POST["fname"]);
    $lname = trim($_POST["lname"]);
    $lowerfn = strtolower($fname);
```

```
$lowerln = strtolower($lname);
$email = $lowerfn."_".$lowerln."@cps.k12.mn.us";
echo $uname = $lowerfn."_".$lowerln;
echo "<BR>";

//Generate Password
$pwd = md5(crypt("4student"));
echo $passwd = substr($pwd, 0, 8);
echo "<BR><BR>";

//Create Samba RID Nubmer
$rid = $uidnumber * 2 + 1000;

//Creates LDIF Array
$info["objectClass"][0] = "top";
$info["objectClass"][1] = "person";
$info["objectClass"][2] = "inetorgperson";
$info["objectClass"][3] = "iplanet-am-managed-person";
$info["objectClass"][4] = "organizationalPerson";
$info["objectClass"][5] = "inetUser";
$info["objectClass"][6] = "ipUser";
$info["objectClass"][7] = "userPresenceProfile";
$info["objectClass"][8] = "inetMailUser";
$info["objectClass"][9] = "inetLocalMailRecipient";
$info["objectClass"][10] = "icsCalendarUser";
$info["objectClass"][11] = "iplanet-am-user-service";
$info["objectClass"][12] = "inetAdmin";
$info["objectClass"][13] = "iPlanetPreferences";
$info["objectClass"][14] = "posixAccount";
$info["objectClass"][15] = "shadowAccount";
$info["objectClass"][16] = "account";
$info["objectClass"][17] = "sambaAccount";
$info["acctFlags"] = "[U          ]";
$info["domain"] = "";
$info["homeDrive"] = "/export/home";
$info["kickoffTime"] = "0";
$info["lmPassword"] = "96B61488C946472009752A3293831D17";
$info["logoffTime"] = "0";
$info["logonTime"] = "0";
$info["ntPassword"] = "A7FB8A589D7F463136D1B332F2AFB0F6";
$info["primaryGroupID"] = "1041";
$info["profilePath"] = "";
$info["pwdCanChange"] = "0";
$info["pwdLastSet"] = "1114176758";
$info["pwdMustChange"] = "0";
$info["rid"] = $rid;
```

```
          $info["scriptPath"] = "";
          $info["smbHome"] = "\\\\172.16.56.78\homes";
          $info["userWorkstations"] = "";
          $info["cn"] = $fname." ".$lname;
          $info["sn"] = $lname;
          $info["uid"] = $uname;
          $info["givenName"] = $fname;
          $info["uidNumber"] = $uidnumber;
          $info["gidNumber"] = "20";
          $info["homeDirectory"] = "/export/home/".$uname;
          $info["loginShell"] = "/bin/sh";
          $info["mail"] = $email;
          $info["mailUserStatus"] = "active";
          $info["mailQuota"] = "104857600";
          $info["mailMsgQuota"] = "-1";
          $info["mailHost"] = "webmail.cps.k12.mn.us";
          $info["mailDeliveryOption"] = "mailbox";
          $info["userPassword"] = $passwd;
          $info["icsCalendar"] = $uname;
          $info["icsSubscribed"] = $uname."$".$fname." ".$lname;
          $info["icsExtendedUserPrefs"][0] = "ceDefaultView=monthview";
          $info["icsExtendedUserPrefs"][1] = "ceInterval=PT0H30M";
          $info["icsExtendedUserPrefs"][2] = "ceExcludeSatSun=0";
          $info["icsExtendedUserPrefs"][3] = "ceGroupInviteAll=1";
          $info["icsExtendedUserPrefs"][4] = "ceSingleCalendarTZID=0";
          $info["icsExtendedUserPrefs"][5] = "ceAllCalendarTZIDs=1";
          $info["icsExtendedUserPrefs"][6] = "ceNotifyEnable=0";
          $info["icsExtendedUserPrefs"][7] = "ceNotifyEmail=".$email;
          $info["icsExtendedUserPrefs"][8] = "ceDefaultAlarmStart=";
          $info["icsExtendedUserPrefs"][9] = "ceDefaultAlarmEmail=".$email;
          $info["icsCalendarOwned"] = $uname."$".$fname." ".$lname;
          $info["inetUserStatus"] = "active";
          $info["icsStatus"] = "active";
          $info["nswmExtendedUserPrefs"] = "meDraftFolder=Drafts";
          $info["nswmExtendedUserPrefs"] = "meSentFolder=Sent";
          $info["nswmExtendedUserPrefs"] = "meTrashFolder=Trash";
          $info["nswmExtendedUserPrefs"] = "meInitialized=true";
          $info["pabURI"] =
"ldap://webmail.cps.k12.mn.us:389/ou=".$uname.",ou=people,o=cps.k12.mn.us,o=isp,o=
pab";
          $info["preferredLanguage"] = "en";

          //Creates LDAP Entry
          $dnentry = "uid=".$uname.", ou=people, o=cps.k12.mn.us,o=isp";
          $r=ldap_add($ldapcon, $dnentry, $info);
```

```
                //Commands To Run On NFS Server
                echo "<U>NFS Server Root Commands</U><BR>";
                echo "mkdir -p /storage/home/".$uname."<BR>";
                echo "unzip -d /storage/home/".$uname."/ /storage/config.zip<BR>";
                echo "cp /storage/Thunderbird.desktop
/storage/home/".$uname."/Desktop/.<BR>";
                echo "chown -R ".$uname." /storage/home/".$uname."<BR>";
                echo "chmod 700 /storage/home/".$uname."<BR><BR>";

                //Commands To Run On Samba Server
                echo "<U>Samba Server Root Commands</U><BR>";
                echo "http://172.16.56.78/samba/<BR>";
                echo "/opentech/samba/".$uname."Z<BR><BR>";

                //Sets Next Available uidnumber.txt
                ++$uidnumber;
                $fp = fopen("/Users/jblukach/Sites/uidnumber.txt",'w');
                fwrite($fp, $uidnumber, strlen($uidnumber));
                fclose($fp);

        }

        //Close LDAP Connection
        ldap_close($ldapbind);
        ?>
```

PHPLDAP: GUEST.PHP

```
        <?php
        //phpLDAP ALPHA for ISD 299
        //by John B. Lukach

        //LDAP Connection Information
        $ldaphost = "172.16.56.95";
        $ldapport = "389";
        $ldapuser = "cn=Directory Manager";
        $ldappass = "";

        //LDAP Bind Functions
        $ldapcon = ldap_connect($ldaphost,$ldapport);
        $ldapbind = ldap_bind($ldapcon, $ldapuser, $ldappass);

        ?>
        <!-- Create LDIF Information-->
        <FORM ACTION=guest.php METHOD=post>
        <U><B>Guest Account</B></U><BR>
```

```
FNAME: <INPUT TYPE=text NAME=fname><BR>
LNAME: <INPUT TYPE=text NAME=lname><BR>
<INPUT TYPE=submit VALUE=Create> <A HREF=index.php>Back</A>
</FORM>
<?

if($_POST["fname"] != "" and $_POST["lname"] != ""){

        //Reads uidnumber.txt File
        $fp = fopen("/Users/jblukach/Sites/uidnumber.txt",'r');
        while(!feof($fp)){
                $uidnumber = trim(fgets($fp, 999));
        }
        fclose($fp);

        //Data Clean Up For LDIF Creation
        $fname = trim($_POST["fname"]);
        $lname = trim($_POST["lname"]);
        $lowerfn = strtolower($fname);
        $lowerln = strtolower($lname);
        $email = $lowerfn."_".$lowerln."@cps.k12.mn.us";
        echo $uname = $lowerfn."_".$lowerln;
        echo "<BR>";

        //Generate Password
        $pwd = md5(crypt("4student"));
        echo $passwd = substr($pwd, 0, 8);
        echo "<BR><BR>";

        //Create Samba RID Nubmer
        $rid = $uidnumber * 2 + 1000;

        //Creates LDIF Array
        $info["objectClass"][0] = "top";
        $info["objectClass"][1] = "person";
        $info["objectClass"][2] = "inetorgperson";
        $info["objectClass"][3] = "iplanet-am-managed-person";
        $info["objectClass"][4] = "organizationalPerson";
        $info["objectClass"][5] = "inetUser";
        $info["objectClass"][6] = "ipUser";
        $info["objectClass"][7] = "userPresenceProfile";
        $info["objectClass"][8] = "inetMailUser";
        $info["objectClass"][9] = "inetLocalMailRecipient";
        $info["objectClass"][10] = "icsCalendarUser";
        $info["objectClass"][11] = "iplanet-am-user-service";
        $info["objectClass"][12] = "inetAdmin";
```

```
$info["objectClass"][13] = "iPlanetPreferences";
$info["objectClass"][14] = "posixAccount";
$info["objectClass"][15] = "shadowAccount";
$info["objectClass"][16] = "account";
$info["objectClass"][17] = "sambaAccount";
$info["acctFlags"] = "[U          ]";
$info["domain"] = "";
$info["homeDrive"] = "/export/home";
$info["kickoffTime"] = "0";
$info["lmPassword"] = "96B61488C946472009752A3293831D17";
$info["logoffTime"] = "0";
$info["logonTime"] = "0";
$info["ntPassword"] = "A7FB8A589D7F463136D1B332F2AFB0F6";
$info["primaryGroupID"] = "1041";
$info["profilePath"] = "";
$info["pwdCanChange"] = "0";
$info["pwdLastSet"] = "1114176758";
$info["pwdMustChange"] = "0";
$info["rid"] = $rid;
$info["scriptPath"] = "";
$info["smbHome"] = "\\\\172.16.56.78\homes";
$info["userWorkstations"] = "";
$info["cn"] = $fname." ".$lname;
$info["sn"] = $lname;
$info["uid"] = $uname;
$info["givenName"] = $fname;
$info["uidNumber"] = $uidnumber;
$info["gidNumber"] = "20";
$info["homeDirectory"] = "/export/home/".$uname;
$info["loginShell"] = "/bin/sh";
$info["mail"] = $email;
$info["mailUserStatus"] = "inactive";
$info["mailQuota"] = "10485760";
$info["mailMsgQuota"] = "-1";
$info["mailHost"] = "webmail.cps.k12.mn.us";
$info["mailDeliveryOption"] = "mailbox";
$info["userPassword"] = $passwd;
$info["icsCalendar"] = $uname;
$info["icsSubscribed"] = $uname."$".$fname." ".$lname;
$info["icsExtendedUserPrefs"][0] = "ceDefaultView=monthview";
$info["icsExtendedUserPrefs"][1] = "ceInterval=PT0H30M";
$info["icsExtendedUserPrefs"][2] = "ceExcludeSatSun=0";
$info["icsExtendedUserPrefs"][3] = "ceGroupInviteAll=1";
$info["icsExtendedUserPrefs"][4] = "ceSingleCalendarTZID=0";
$info["icsExtendedUserPrefs"][5] = "ceAllCalendarTZIDs=1";
$info["icsExtendedUserPrefs"][6] = "ceNotifyEnable=0";
```

```
                $info["icsExtendedUserPrefs"][7] = "ceNotifyEmail=".$email;
                $info["icsExtendedUserPrefs"][8] = "ceDefaultAlarmStart=";
                $info["icsExtendedUserPrefs"][9] = "ceDefaultAlarmEmail=".$email;
                $info["icsCalendarOwned"] = $uname."$".$fname." ".$lname;
                $info["inetUserStatus"] = "active";
                $info["icsStatus"] = "inactive";
                $info["nswmExtendedUserPrefs"][0] = "meDraftFolder=Drafts";
                $info["nswmExtendedUserPrefs"][1] = "meSentFolder=Sent";
                $info["nswmExtendedUserPrefs"][2] = "meTrashFolder=Trash";
                $info["nswmExtendedUserPrefs"][3] = "meInitialized=true";
                $info["pabURI"] =
"ldap://webmail.cps.k12.mn.us:389/ou=".$uname.",ou=people,o=cps.k12.mn.us,o=isp,o=
pab";
                $info["preferredLanguage"] = "en";

                //Creates LDAP Entry
                $dnentry = "uid=".$uname.", ou=people, o=cps.k12.mn.us,o=isp";
                $r=ldap_add($ldapcon, $dnentry, $info);

                //Commands To Run On NFS Server
                echo "<U>NFS Server Root Commands</U><BR>";
                echo "mkdir -p /storage/home/".$uname."<BR>";
                echo "unzip -d /storage/home/".$uname."/ /storage/config.zip<BR>";
                echo "cp /storage/Thunderbird.desktop
/storage/home/".$uname."/Desktop/.<BR>";
                echo "chown -R ".$uname." /storage/home/".$uname."<BR>";
                echo "chmod 700 /storage/home/".$uname."<BR><BR>";

                //Commands To Run On Samba Server
                echo "<U>Samba Server Root Commands</U><BR>";
                echo "http://172.16.56.78/samba/<BR>";
                echo "/opentech/samba/".$uname."Z<BR><BR>";

                //Sets Next Available uidnumber.txt
                ++$uidnumber;
                $fp = fopen("/Users/jblukach/Sites/uidnumber.txt",'w');
                fwrite($fp, $uidnumber, strlen($uidnumber));
                fclose($fp);

        }

        //Close LDAP Connection
        ldap_close($ldapbind);
        ?>
```

PHPLDAP: EMAIL.PHP

```
          <?php
          //phpLDAP ALPHA for ISD 299
          //by John B. Lukach

          //LDAP Connection Information
          $ldaphost = "172.16.56.95";
          $ldapport = "389";
          $ldapuser = "cn=Directory Manager";
          $ldappass = "";

          //LDAP Bind Functions
          $ldapcon = ldap_connect($ldaphost,$ldapport);
          $ldapbind = ldap_bind($ldapcon, $ldapuser, $ldappass);

          ?>
          <!-- Create LDIF Information-->
          <FORM ACTION=email.php METHOD=post>
          <U><B>Email Group</B></U><BR>
          NAME: <INPUT TYPE=text NAME=name><BR>
          INTERNAL MEMBERS:<BR>
          <TEXTAREA NAME=inside COLS=40 ROWS=6></TEXTAREA><BR>
          EXTERNAL MEMBERS:<BR>
          <TEXTAREA NAME=outside COLS=40 ROWS=6></TEXTAREA><BR>
          <INPUT TYPE=submit VALUE=Create> <A HREF=index.php>Back</A>
          </FORM>
          <?

          if($_POST["name"] != "" and $_POST["inside"] != "" and $_POST["outside"] !=
"")}{

                    //Data Clean Up For LDIF Creation
                    $name = strtolower(trim($_POST["name"]));
                    $inside = strtolower(trim($_POST["inside"]));
                    $outside = strtolower(trim($_POST["outside"]));

                    //Creates LDIF Array
                    $info["mgmanMemberVisibility"] = "none";
                    $info["owner"] = "uid=ServiceAdmin, ou=People,
o=cps.k12.mn.us,o=isp";
                    $info["mgmanJoinability"] = "none";
                    $info["nsNumUsers"] = "0";
                    $info["mail"] = $name."@cps.k12.mn.us";
                    $info["objectClass"][0] = "top";
                    $info["objectClass"][1] = "groupOfUniqueNames";
                    $info["objectClass"][2] = "inetMailGroup";
```

```
        $info["objectClass"][3] = "inetMailGroupManagement";
        $info["objectClass"][4] = "inetLocalMailRecipient";
        $info["objectClass"][5] = "nsManagedMailList";
        $info["mgrpAllowedDomain"] = "cps.k12.mn.us";
        $info["mailHost"] = "webmail.cps.k12.mn.us";
        $info["inetMailGroupStatus"] = "active";
        $info["mailDeliveryOption"] = "members";
        $info["mgmanHidden"] = "false";
        $info["preferredLanguage"] = "en";
        $info["cn"] = $name;
        $info["nsMaxUsers"] = "1000";

        //Split INSIDE Members String
        $eachmember = split("\|", $inside);

        //Count Number Of Members
        $result = count($eachmember);

        //Looping LDIF Output
        for($i=0 ; $i < $result ; $i++){
            $info["uniqueMember"][$i] = "uid=".trim($eachmember[$i]).",
ou=people, o=cps.k12.mn.us,o=isp";
        }

        //Split OUTSIDE Members String
        $eachmember = split("\|", $outside);

        //Count Number Of Members
        $result = count($eachmember);

        //Looping LDIF Output
        for($i=0 ; $i < $result ; $i++){
            $info["mgrpRFC822MailMember"][$i] = trim($eachmember[$i]);
        }

        //Creates LDAP Entry
        $dnentry = "cn=".$name.", ou=Groups, o=cps.k12.mn.us,o=isp";
        $r=ldap_add($ldapcon, $dnentry, $info);

        echo "DONE!";

    }

    //Close LDAP Connection
    ldap_close($ldapbind);
    ?>
```

PHPLDAP: POSIX.PHP

```
<?php
//phpLDAP ALPHA for ISD 299
//by John B. Lukach

//LDAP Connection Information
$ldaphost = "172.16.56.95";
$ldapport = "389";
$ldapuser = "cn=Directory Manager";
$ldappass = "";

//LDAP Bind Functions
$ldapcon = ldap_connect($ldaphost,$ldapport);
$ldapbind = ldap_bind($ldapcon, $ldapuser, $ldappass);

?>
<!-- Create LDIF Information-->
<FORM ACTION=posix.php METHOD=post>
<U><B>Posix Group</B></U><BR>
NAME: <INPUT TYPE=text NAME=name><BR>
MEMBERS:<BR>
<TEXTAREA NAME=members COLS=40 ROWS=6></TEXTAREA><BR>
<INPUT TYPE=submit VALUE=Create> <A HREF=index.php>Back</A>
</FORM>
<?

if($_POST["name"] != "" and $_POST["members"] != ""){

        //Reads gidnumber.txt File
        $fp = fopen("/Users/jblukach/Sites/gidnumber.txt",'r');
        while(!feof($fp)){
                $gidnumber = trim(fgets($fp, 999));
        }
        fclose($fp);

        //Data Clean Up For LDIF Creation
        $name = strtolower(trim($_POST["name"]));
        $members = strtolower(trim($_POST["members"]));

        //Creates LDIF Array
        $info["gidNumber"] = $gidnumber;
        $info["objectClass"][0] = "top";
        $info["objectClass"][1] = "posixgroup";
        $info["cn"] = $name;

        //Split Members String
```

```
                $eachmember = split("\|", $members);

                //Count Number Of Members
                $result = count($eachmember);

                //Looping LDIF Output
                for($i=0 ; $i < $result ; $i++){
                        $info["memberUid"][$i] = trim($eachmember[$i]);
                }

                //Creates LDAP Entry
                $dnentry = "gidnumber=".$gidnumber.", ou=group,
o=cps.k12.mn.us,o=isp";
                $r=ldap_add($ldapcon, $dnentry, $info);

                //Commands To Run On NFS Server
                echo "<U>NFS Server Root Commands</U><BR>";
                echo "mkdir -p /storage/shared/".$name."<BR>";
                echo "chown -R root:".$name." /storage/shared/".$name."<BR>";
                echo "chmod 770 /storage/shared/".$name."<BR><BR>";

                //Add Lines To Samba Configuration File
                echo "<U>Add To: /usr/local/samba/lib/smb.conf</U><BR>";
                echo "<BR>";
                echo "[".$name."]<BR>";
                echo "<BR>";
                echo "comment = ".$name."<BR>";
                echo "valid users = @".$name."<BR>";
                echo "admin users = @".$name."<BR>";
                echo "path = /export/shared/".$name."<BR>";
                echo "read only = no<BR><BR>";

                //Commands To Run On Samba Server
                echo "<U>Samba Server Root Commands</U><BR>";
                echo "/usr/local/samba/bin/testparm<BR><BR>";

                //Sets Next Available uidnumber.txt
                ++$gidnumber;
                $fp = fopen("/Users/jblukach/Sites/gidnumber.txt",'w');
                fwrite($fp, $gidnumber, strlen($gidnumber));
                fclose($fp);

        }

        //Close LDAP Connection
        ldap_close($ldapbind);
        ?>
```

PHPLDAP: EDITACCT.PHP

```
        <?php
        //phpLDAP ALPHA for ISD 299
        //by John B. Lukach

        //LDAP Connection Information
        $ldaphost = "172.16.56.95";
        $ldapport = "389";
        $ldapuser = "cn=Directory Manager";
        $ldappass = "";

        //LDAP Bind Functions
        $ldapcon = ldap_connect($ldaphost,$ldapport);
        $ldapbind = ldap_bind($ldapcon, $ldapuser, $ldappass);

        //Pulls DN Information
        $name = trim($_GET["id"]);

        ?>
        <!-- Delete LDIF Information-->
        <FORM ACTION=editacct.php METHOD=post>
        <U><B>Are you sure you want to change password for:
<?=$name?>?</B></U><BR>
        <INPUT TYPE=hidden NAME=name VALUE=<?=$name?>>
        <INPUT TYPE=submit VALUE=Change> <A HREF=index.php>Back</A>
        </FORM>
        <?

        //Tests to make sure you want to change password
        if($_POST["name"] != ""){

                //Sets Name Variable
                $name = trim($_POST["name"]);

                //Prints Username
                echo $name."<BR>";
                //Generate Password
                $pwd = md5(crypt("4student"));
                echo $passwd = substr($pwd, 0, 8);
                echo "<BR><BR>";

                //Change LDAP Password
                $info["userPassword"] = $passwd;
```

```
		$dnentry = "uid=".$name.", ou=people, o=cps.k12.mn.us,o=isp";
		ldap_modify($ldapcon,$dnentry,$info);

		//Commands To Run On Samba Server
		echo "<U>Samba Server Root Commands</U><BR>";
		echo "http://172.16.56.78/samba/<BR>";
		echo "/opentech/samba/".$name."Z<BR><BR>";

	}

	//Close LDAP Connection
	ldap_close($ldapbind);
	?>
```

PHPLDAP: EDITEMAIL.PHP

```
	<?php
	//phpLDAP ALPHA for ISD 299
	//by John B. Lukach

	//LDAP Connection Information
	$ldaphost = "172.16.56.95";
	$ldapport = "389";
	$ldapuser = "cn=Directory Manager";
	$ldappass = "";

	//LDAP Bind Functions
	$ldapcon = ldap_connect($ldaphost,$ldapport);
	$ldapbind = ldap_bind($ldapcon, $ldapuser, $ldappass);

	//Check To See If Adding New
	if($_POST["secret"] != ""){

		//Clean Up So Program Operates
		$_GET["id"] = $_POST["secret"];
		$name = trim($_GET["id"]);

		//Tests If Internal Email Add
		if($_POST["internal"] != ""){

			//Adds New Member
			$dnentry = "cn=".$name.", ou=Groups, o=cps.k12.mn.us,o=isp";
			$entry["uniquemember"][] = "uid=".trim($_POST["internal"].",
ou=people, o=cps.k12.mn.us,o=isp");
			ldap_mod_add($ldapcon, $dnentry, $entry);
```

```
                }

                //Tests If External Email Add
                if($_POST["external"] != ""){

                        //Adds New Member
                        $dnentry = "cn=".$name.", ou=Groups, o=cps.k12.mn.us,o=isp";
                        $entry["mgrprfc822mailmember"][] = trim($_POST["external"]);
                        ldap_mod_add($ldapcon, $dnentry, $entry);

                }

        }

        //Pulls DN Information
        $name = trim($_GET["id"]);
        $delete = trim($_GET["delete"]);

        //Tests To See If Deleting Internal
        if($name != "" and $delete == "intdelete"){

                //Cleans Data For CORRECT DN
                $info = split(",", $_GET["user"]);
                $info = split("=", $info[0]);

                //Checks Which Delete Entry
                if($info[1] != ""){
                        $entry["uniquemember"][] = trim("uid=".$info[1].", ou=people,
o=cps.k12.mn.us,o=isp");
                } else {
                        $entry["uniquemember"][] = trim($_GET["user"]);
                }

                //Deletes Member
                $dnentry = "cn=".$name.", ou=Groups, o=cps.k12.mn.us,o=isp";
                ldap_mod_del($ldapcon, $dnentry, $entry);

        }

        //Tests To See If Deleting External
        if($name != "" and $delete == "extdelete"){

                //Deletes Member
                $dnentry = "cn=".$name.", ou=Groups, o=cps.k12.mn.us,o=isp";
                $entry["mgrprfc822mailmember"][] = trim($_GET["user"]);
                ldap_mod_del($ldapcon, $dnentry, $entry);
        }

        ?>
        <!-- Create LDIF Information-->
        <FORM ACTION=editemail.php METHOD=post NAME=internal>
        <U><B>Add Internal Member</B></U><BR>
        INTERNAL: <INPUT TYPE=text NAME=internal><INPUT TYPE=hidden
NAME=secret VALUE=<?=$name?>><BR>
        <INPUT TYPE=submit VALUE=Add> <A HREF=index.php>Back</A>
        </FORM>
        <FORM ACTION=editemail.php METHOD=post NAME=external>
        <U><B>Add External Member</B></U><BR>
        EXTERNAL: <INPUT TYPE=text NAME=external><INPUT TYPE=hidden
NAME=secret VALUE=<?=$name?>><BR>
        <INPUT TYPE=submit VALUE=Add> <A HREF=index.php>Back</A>
        </FORM>
        <?

        //Gets Group Membership Lists
        $dnentry = "cn=".$name.", ou=Groups, o=cps.k12.mn.us,o=isp";
        $sr = ldap_search($ldapcon, $dnentry, "uniqueMember=*");
        $info = ldap_get_entries($ldapcon, $sr);

        //Group Membership Counts
        $intcount = count($info[0]["uniquemember"]);
        $extcount = count($info[0]["mgrprfc822mailmember"]);

        //Display Internal Email Addresses
        echo "<B><U>Internal Email Addresses</U></B><BR>";
        for ($i=0; $i<$intcount; $i++) {
                if($info[0]["uniquemember"][$i] != "" and $info[0]["uniquemember"][$i]
!= "mgmanMemberVisibility: none"){
                        echo $info[0]["uniquemember"][$i]." - <a
href=editemail.php?id=".$name."&delete=intdelete&user=".$info[0]["uniquemember"][$
i]."><img src=delete.gif></a><BR>";
                }
        }

        //Adds Extra Break For Clean Display
        echo "<BR>";

        //Display External Email Addresses
        echo "<B><U>External Email Addresses</U></B><BR>";
        for ($i=0; $i<$extcount; $i++) {
                if($info[0]["mgrprfc822mailmember"][$i] != ""){
                    echo $info[0]["mgrprfc822mailmember"][$i]." - <a
href=editemail.php?id=".$name."&delete=extdelete&user=".$info[0]["mgrprfc822mailm
ember"][$i]."><img src=delete.gif></a><BR>";
                }
        }

        //Close LDAP Connection
        ldap_close($ldapbind);
        ?>
```

PHPLDAP: EDITPOSIX.PHP

```
        <?php
        //phpLDAP ALPHA for ISD 299
        //by John B. Lukach

        //LDAP Connection Information
        $ldaphost = "172.16.56.95";
        $ldapport = "389";
        $ldapuser = "cn=Directory Manager";
        $ldappass = "";

        //LDAP Bind Functions
        $ldapcon = ldap_connect($ldaphost,$ldapport);
        $ldapbind = ldap_bind($ldapcon, $ldapuser, $ldappass);

        //Check To See If Adding New
        if($_POST["secret"] != ""){

                //Clean Up So Program Operates
                $_GET["id"] = $_POST["secret"];
                $name = trim($_GET["id"]);

                //Adds New Member
                $dnentry = "gidnumber=".$name.", ou=group, o=cps.k12.mn.us,o=isp";
                $entry["memberuid"][] = trim($_POST["name"]);
                ldap_mod_add($ldapcon, $dnentry, $entry);

        }

        //Pulls DN Information
        $name = trim($_GET["id"]);
        $delete = trim($_GET["delete"]);

        //Tests To See If Deleting
        if($name != "" and $delete == "delete"){
                //Deletes Member
                $dnentry = "gidnumber=".$name.", ou=group, o=cps.k12.mn.us,o=isp";
                $entry["memberuid"][] = trim($_GET["user"]);
                ldap_mod_del($ldapcon, $dnentry, $entry);

        }

        ?>
        <!-- Create LDIF Information-->
        <FORM ACTION=editposix.php METHOD=post>
        <U><B>Add Posix Member</B></U><BR>
        MEMBER: <INPUT TYPE=text NAME=name><INPUT TYPE=hidden
NAME=secret VALUE=<?=$name?>><BR>
        <INPUT TYPE=submit VALUE=Add> <A HREF=index.php>Back</A>
        </FORM>
        <?

        //Gets Group Membership Lists
        $dnentry = "gidnumber=".$name.", ou=group, o=cps.k12.mn.us,o=isp";
        $sr = ldap_search($ldapcon, $dnentry, "memberUid=*");
        $info = ldap_get_entries($ldapcon, $sr);

        //Group Membership Counts
        $intcount = count($info[0]["memberuid"]);

        //Display Internal Email Addresses
        echo "<B><U>".$info[0]["cn"][0]." Posix Group Members</U></B><BR>";
        for ($i=0; $i<$intcount; $i++) {
                if($info[0]["memberuid"][$i] != ""){
                        echo $info[0]["memberuid"][$i]." - <a
href=editposix.php?id=".$name."&delete=delete&user=".$info[0]["memberuid"][$i]."><i
mg src=delete.gif></a><BR>";
                }
        }

        //Close LDAP Connection
        ldap_close($ldapbind);
        ?>
```

PHPLDAP: DELACCT.PHP

```
        <?php
        //phpLDAP ALPHA for ISD 299
        //by John B. Lukach
```

```
//LDAP Connection Information
$ldaphost = "172.16.56.95";
$ldapport = "389";
$ldapuser = "cn=Directory Manager";
$ldappass = "";

//LDAP Bind Functions
$ldapcon = ldap_connect($ldaphost,$ldapport);
$ldapbind = ldap_bind($ldapcon, $ldapuser, $ldappass);

//Pulls DN Information
$name = trim($_GET["id"]);

?>
<!-- Delete LDIF Information-->
<FORM ACTION=delacct.php METHOD=post>
<U><B>Are you sure you want to delete: <?=$name?>?</B></U><BR>
<INPUT TYPE=hidden NAME=name VALUE=<?=$name?>>
<INPUT TYPE=submit VALUE=Delete> <A HREF=index.php>Back</A>
</FORM>
<?

//Tests to make sure you want to delete
if($_POST["name"] != ""){

        //Sets Name Variable
        $name = trim($_POST["name"]);

        //Deletes LDAP Entry
        $dnentry = "uid=".$uname.", ou=people, o=cps.k12.mn.us,o=isp";
        ldap_delete($ldapcon,$dnentry);

        //Commands To Run On NFS Server
        echo "<U>NFS Server Root Commands</U><BR>";
        echo "rm -R /storage/home/".$name."<BR><BR>";

        //Commands To Run On Webmail Server
        echo "<U>Webmail Server Root Commands</U><BR>";
        echo "/jes/msg/sbin/mboxutil -o -w filename<BR>";
        echo "/jes/msg/sbin/mboxutil -d -f /jes/msg/cpsconfig/log/filename<BR>";
        echo "rm /jes/msg/cpsconfig/log/filename<BR><BR>";

}

//Close LDAP Connection
ldap_close($ldapbind);
```

```
?>
```

PHPLDAP: DELEMAIL.PHP

```
<?php
//phpLDAP ALPHA for ISD 299
//by John B. Lukach

//LDAP Connection Information
$ldaphost = "172.16.56.95";
$ldapport = "389";
$ldapuser = "cn=Directory Manager";
$ldappass = "";

//LDAP Bind Functions
$ldapcon = ldap_connect($ldaphost,$ldapport);
$ldapbind = ldap_bind($ldapcon, $ldapuser, $ldappass);

//Pulls DN Information
$name = trim($_GET["id"]);

?>
<!-- Delete LDIF Information-->
<FORM ACTION=delemail.php METHOD=post>
<U><B>Are you sure you want to delete: <?=$name?>?</B></U><BR>
<INPUT TYPE=hidden NAME=name VALUE=<?=$name?>>
<INPUT TYPE=submit VALUE=Delete> <A HREF=index.php>Back</A>
</FORM>
<?

//Tests to make sure you want to delete
if($_POST["name"] != ""){

        //Sets Name Variable
        $name = trim($_POST["name"]);

        //Deletes LDAP Entry
        $dnentry = "cn=".$name.", ou=Groups, o=cps.k12.mn.us,o=isp";
        ldap_delete($ldapcon,$dnentry);

        echo "DONE!";

}

//Close LDAP Connection
ldap_close($ldapbind);
```

```
        ?>
```

PHPLDAP: DELPOSIX.PHP

```
        <?php
        //phpLDAP ALPHA for ISD 299
        //by John B. Lukach

        //LDAP Connection Information
        $ldaphost = "172.16.56.95";
        $ldapport = "389";
        $ldapuser = "cn=Directory Manager";
        $ldappass = "";

        //LDAP Bind Functions
        $ldapcon = ldap_connect($ldaphost,$ldapport);
        $ldapbind = ldap_bind($ldapcon, $ldapuser, $ldappass);

        //Pulls DN Information
        $name = trim($_GET["id"]);

        ?>
        <!-- Delete LDIF Information-->
        <FORM ACTION=delposix.php METHOD=post>
        <U><B>Are you sure you want to delete: <?=$name?>?</B></U><BR>
        <INPUT TYPE=hidden NAME=name VALUE=<?=$name?>>
        <INPUT TYPE=submit VALUE=Delete> <A HREF=index.php>Back</A>
        </FORM>
        <?

        //Tests to make sure you want to delete
        if($_POST["name"] != ""){

                //Sets Name Variable
                $name = trim($_POST["name"]);

                //Deletes LDAP Entry
                $dnentry = "gidnumber=".$gidnumber.", ou=group,
o=cps.k12.mn.us,o=isp";
                ldap_delete($ldapcon,$dnentry);

                //Commands To Run On NFS Server
                echo "<U>NFS Server Root Commands</U><BR>";
                echo "rm -R /storage/shared/".$name."<BR><BR>";

                //Commands To Run On Samba Server
        echo "<U>Samba Server Root Commands</U><BR>";
        echo "Remove ".$name." From: /usr/local/samba/lib/smb.conf<BR>";
        echo "/usr/local/samba/bin/testparm<BR><BR>";

}

//Close LDAP Connection
ldap_close($ldapbind);
?>
```

APPENDIX C: PROJECT PLAN

788 Project Planning

by John Lukach

"phpLDAP" will create account management software for Caledonia Area Public School's heterogeneous computer system. Allowing resources to be allocated by district office and student services through a web-based application as end-users' needs change without the involvement of the lone technology staff member. Independent School District 299 through our partnership with Sun Microsystems has reached single sign-on into a Java Enterprise System (JES) Directory Server (LDAP) over the past two years. Currently information needs to be generated, stored, changed, and deleted in LDAP for email, calendar, sunrays, windows, cisco, apple, samba, n2h2, and roaring penguin requiring seven programs to create one user account not including groups and other special settings.

Lightweight directory access protocol (LDAP) is a technology that provides directory services to applications ranging from email systems to distributed system management tools. LDAP directory databases consist of entries composed of one or more attributes. Creating LDAP schema that is a collection of attribute definitions, object classes, and other information for returning queries (1). The Internet Engineering Task Force (IETF) has worked to create an industry accepted standard LDAPv3 used by Microsoft, IBM, Novell, Sun, and others to create their own products. Entegrity Solutions Assure Access was the only tool that worked with Netscape LDAP. Over the years Netscape LDAP has become iPlanet and Sun ONE that is now called JES. Entegrity says Assure Access was not designed for user administration, since most LDAP servers bundle sophisticated user management GUI's (2). Another option was Wave Set that is now owned by Sun to create their JES Identity Manager product. Identity Manger does not support sunrays, local windows, Cisco, apple, samba, n2h2, or roaring penguin software and hardware that Caledonia Area Public Schools uses (3).

I plan to create a web-based application written in PHP for LDAP user account management. This program will allow non-technical staff to manage Independent School District 299 single sign-on technology solution. After the completion it will be released on Source Forge as an open source project for all future upkeep. The phpLDAP software capabilities will include:

- Create, manage, and delete user accounts
- Create, manage, and delete posix groups
- Create, manage, and delete email groups
- Synchronize passwords for all software and hardware

I will use my test environment to write my PHP software against. The biggest challenge will be replicating or consolidating the seven programs necessary to create a single user account into my web-based software. This will allow six additional products not commercially support by Sun to be used internally.

phpLDAP project timeline will be:

January 2nd to January 22nd

Development and testing for user account module

January 23rd to February 12th

Development and testing for posix group module

February 13th to March 5th

Development and testing for email group module

March 6th to March 26th

Write phpLDAP project report

March 27th to April 2nd

Create phpLDAP training material

April 3rd to April 9th

Submit project to committee supervisor

April 10th to April 16th

Submit project to full committee

May 1st

phpLDAP project presentation

References:

1. Shi, Stokes, Byrne, Corn, Bachmann, Jones. "An enterprise directory solution with DB2." IBM Systems Journal; 2000; 39,2; Research Library. pg. 360.
2. Ross, Michael and Rubin, Jeff. "Authentication Gets Tough." Network Computing; May 28, 2001; 12,11; ProQuest Education Journals. pg. 97.
3. JES Identity Management. http://www.sun.com/identity_mgmt.