Leveraging Active Directory on Mac OS X

Mike Bombich
Apple Systems Engineer
bombich@apple.com
Winter 2006, Tiger Edition

I. Overview

A directory service provides a central repository for information about people and resources in an organization. In education and enterprise environments, directory services are the ideal way to manage users and computing resources. Organizations with as few as ten people can benefit from a directory service.

Directory services can be doubly beneficial. They centralize system and network administration, and they simplify a user's experience on the network. With directory services, information about the users -- such as their names, passwords, and locations of network home directories -- can be maintained centrally rather than on each computer individually. Directory services can also maintain centralized information about printers, computers, and other network resources.

Apple has built an open, extensible directory services architecture, called Open Directory, into Mac OS X and Mac OS X Server. A Mac OS X client or Mac OS X Server computer can use Open Directory to retrieve authoritative information about users and network resources from a variety of directory services such as:

- LDAP service on a Mac OS X Server system
- Active Directory service on Windows 2000 or 2003 server
- OpenLDAP or other LDAP service on a third-party server such as Sun One or Novell eDirectory
- NIS on a Unix server

The goal of this document is to provide an understanding of how directory services work in general, how directory services are implemented on Mac OS X and Mac OS X Server, and how to configure a Mac OS X client to authenticate against various directory services.

Parts of this document are "exercise" oriented to get you familiar with directory services on Mac OS X, while other parts will provide documentation specific to integrating your server and clients into your own organization's infrastructure. You are encouraged to appropriate two non-production machines for these exercises. On one machine, install Mac OS X and all currently available updates. On the other machine, install Mac OS X Server and all available updates. You can get an evaluation copy of Mac OS X Server from your Apple Systems Engineer. You are also encouraged to follow each section of this document to gain familiarity with Open Directory on Mac OS X, rather than skipping to sections that seem most appropriate for your environment.

In the sections below, you will:

- Learn how to configure a Mac OS X Server to provide a directory service to Mac OS X clients
- Learn how to configure a Mac OS X client to authenticate against a Mac OS X Open Directory server
- Learn about Kerberos on Mac OS X
- Learn how to configure a Mac OS X client to authenticate against an Active Directory Server
- Learn how to leverage the full management capabilities of Mac OS X by using both an Active Directory Server and a Mac OS X Open Directory server
- Learn how to leverage AD for authentication and Single-Sign-On with Mac OS X Server services
- Learn how to restrict access to Mac OS X Server services

II. Configure a Mac OS X Server Open Directory Master

You will use the Server Admin application in /Applications/Server to configure Mac OS X Server's shared directory service. Before jumping into setup, however, there are a few steps you should take to confirm that your server is prepared to host these services successfully.

A. DNS setup

One of the primary goals of setting up a directory service is to enable the authentication/identification of users. Users, however, also benefit from trusting the identity of the hosts and servers that they connect to. Part of authentication is determining that you are who claim to be. Users provide a username when they authenticate, then provide a password to prove that they are that user.

Computers are represented by hostnames. When you login to a server, the client that you use to login will likely (hopefully) do at least some rudimentary verification on that hostname. The most basic identity check is to verify that the hostname provided by the server matches the hostname/IP address indicated in a DNS server. This is known as forward and reverse DNS verification.

To confirm that your server's hostname is configured properly on the host and in DNS, follow these steps at your server (substituting your own server's values). Do not type the prompt character ("%").

```
% hostname
xserve.apple.edu

% host xserve.apple.edu
xserve.apple.edu has address 10.0.1.8

% host 10.0.1.8
8.1.0.10.in-addr.arpa domain name pointer xserve.apple.edu.
```

If you get any conflicting results, or "host xyz not found: 3(NXDOMAIN)" messages, your hostname is improperly set or not configured correctly in DNS. The hostname is set using the "scutil" command:

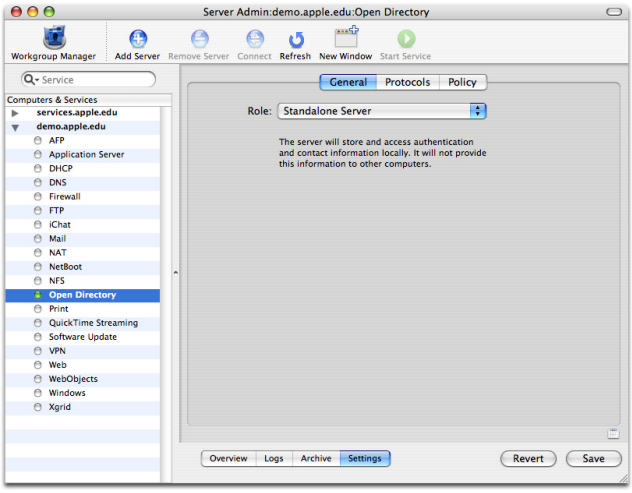
```
% scutil --get HostName
HostName: not set
% sudo scutil --set HostName "xserve.apple.edu"
```

If DNS hostname resolution is not working properly, Single Sign On will not work. You are strongly advised to get DNS hostname resolution working properly before configuring any services on Mac OS X Server.

For the remainder of this chapter, use the hostname of your server where "hostname.apple.edu" is indicated.

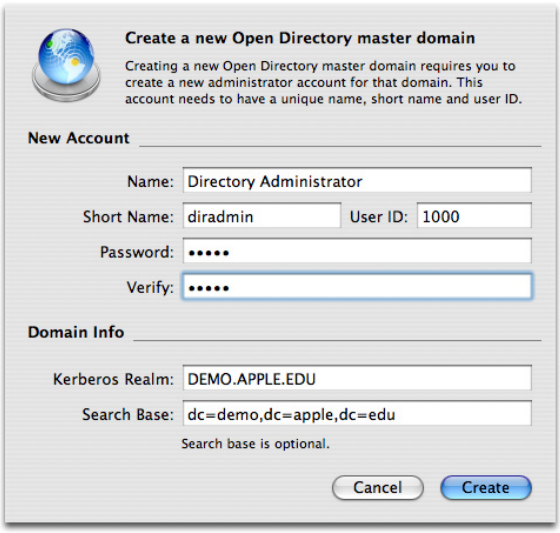
B. Configure the Open Directory Master

1. Open the Console application and click on the disclosure triangle next to "/Library/Logs". Click on the "slapconfig.log" file to view its contents. There should only be one line in this file referencing the initial setup of standalone mode.
2. Open the Server Admin application and authenticate. Click on your server's disclosure triangle and click on the Open Directory service to view its status.

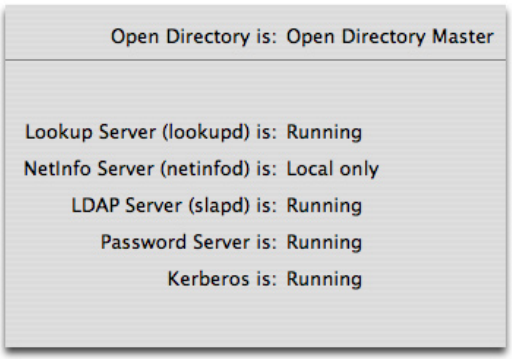


3. Click on the "Settings" tab and select "Open Directory Master" from the Role popup menu.
4. A sheet will prompt you to create a new user. The Kerberos realm name is arbitrary, but is generally of the form SERVER.DEPT_NAME.ORGANIZATION.EDU to avoid conflicts with other Kerberos realms on the network, and it is always uppercase. The search base is a product of the realm name, for example "dc=server,dc=dept_name,dc=organization,dc=edu". If this server will host an authoritative Kerberos realm, set your realm name to "ORGANIZATION.EDU".

For this exercise, use the information that is auto-populated in the form (not the info in the screenshot)



5. Click “Create”, then click the Save button.
6. Return to the Console application and watch the slapconfig.log as the server configures itself. This is the most exciting part of OD setup because in a matter of seconds, you have a fully configured, fully functional directory service and Kerberos Key Distribution Center. Warnings about no policy for various services may be ignored.
7. When the server has completed configuring itself, return to Server Admin and click on the “General” tab in the Open Directory service. Click the Refresh button at the top of the window and confirm that LDAP, Password Server and Kerberos are now running.

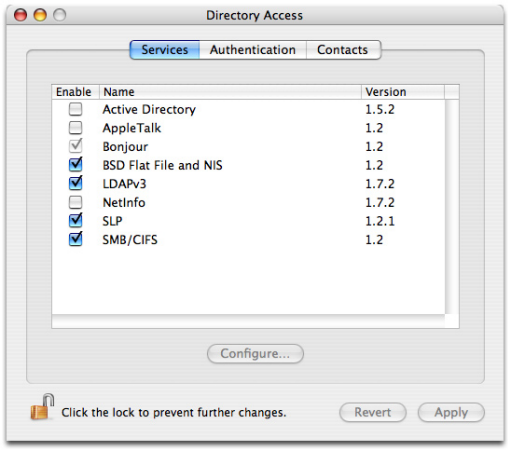


III. Configure a client to authenticate to the OD Master

Configuration of Mac OS X client directory services is done with the Directory Access application located in the Utilities folder. Each item listed in Directory Access represents a method with which Mac OS X uses to gather information about users, groups, computers, and network services.

A. Configure the LDAPv3 plugin and Authentication search path

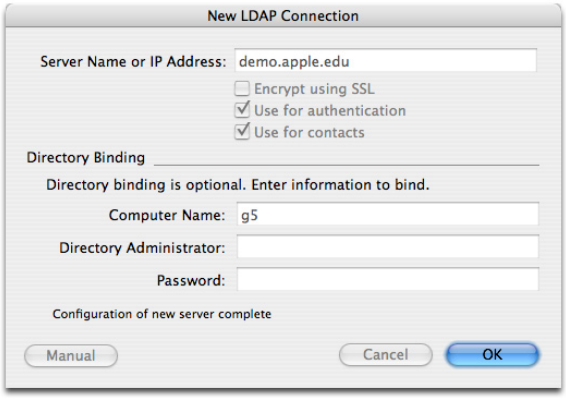
1. Launch Directory Access and authenticate.



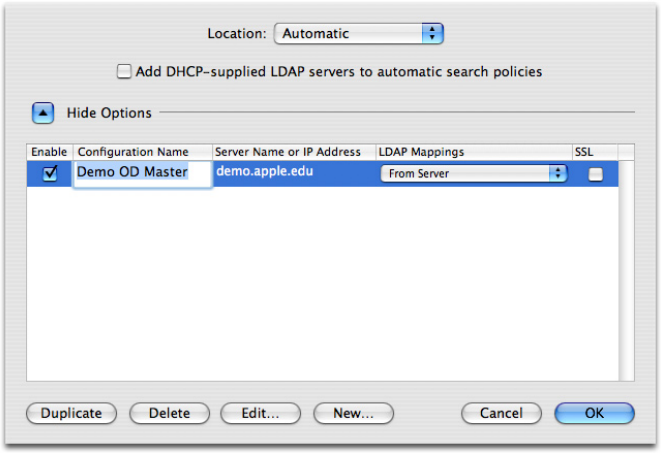
2. To configure Mac OS X to authenticate against an Open Directory Master, we will use the LDAPv3 plugin. Double-click on the LDAPv3 item.
3. Uncheck the box to "Add DHCP-supplied LDAP servers..." (if it is not already unchecked).
4. Click on the disclosure triangle to show additional options.
5. Click on the "New" button to add a new configuration with your OD Master's settings

Use your server's hostname. You can use the hostname or IP address, but hostname is recommended.

6. Directory Access will contact your server to acquire configuration information, then present you the opportunity to do an "authenticated bind". If Mac OS X Server was the only directory service you intended to use, doing an authenticated bind is a good idea, and automatically creates computer records within the directory service for you. Authenticated binds also allow you to manage login and logout scripts from the directory service. For the purpose of this exercise, do not perform an authenticated bind, simply click "Continue".



7. Click "OK", then assign a name to your new LDAPv3 configuration.

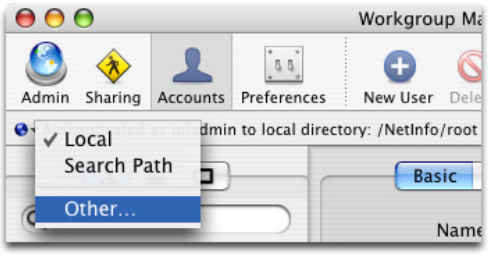


8. Click "OK" to save, then click on the Apply button if it is highlighted. The system is now configured to authenticate against the Open Directory Master.

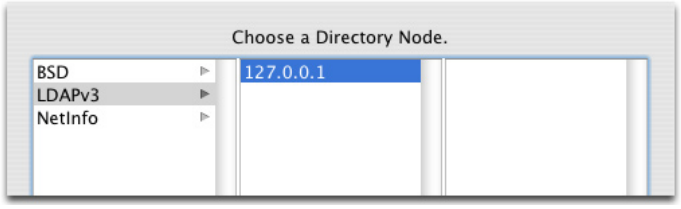
B. Create Open Directory users and verify directory connectivity

Before jumping into logging in as a directory user, you obviously need to create users in the shared directory (other than the directory administrator). It would also be wise to verify that you can look up information in the directory service, otherwise you may end up staring at a never-ending progress indicator in the loginwindow.

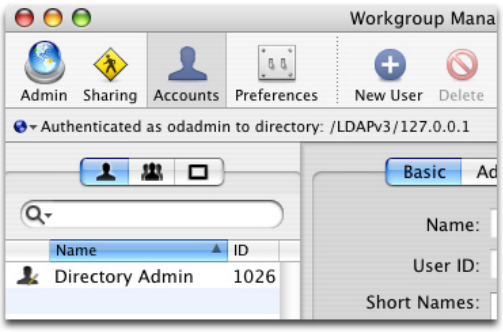
1. Log in to Workgroup Manager using the directory administrator account you used when promoting the server to OD Master. Be sure to use the hostname, not the IP address of your server when you log in. Once you have authenticated, click on the tiny blue globe next to the text "Authenticated as admin to local directory /NetInfo/root" and select "Other...".



2. In the panel that appears, navigate to /LDAPv3/127.0.0.1 and click OK.

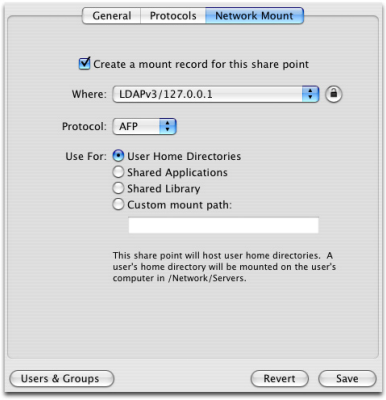


3. Verify that you are working in the LDAP node. In general, future use of WGM should take you directly to the LDAP node.

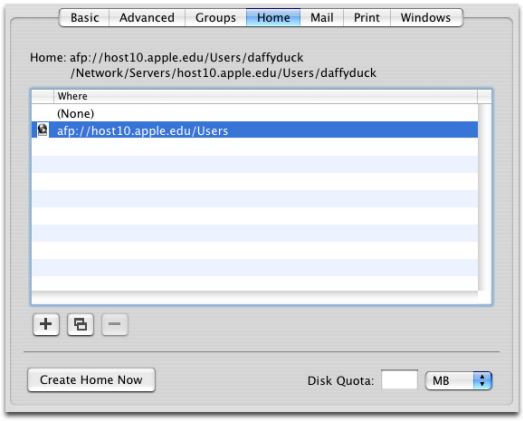


4. Click on the Sharing button. Select the "Users" sharepoint and click on the "Network Mount" tab. Click on the lock to authenticate to the directory (use the directory administrator you created when you promoted your server to OD Master).

5. Check the box “Create a mount record for this sharepoint”. Indicate that the sharepoint is to be used for “User Home Directories”, then click on the Save button.



6. In Server Admin, start the AFP service.
7. Click on the Accounts button and create a few users manually. In the “Home” tab for each user, choose the AFP sharepoint that you set up in a previous step and click on the “Create Home Now” button.



8. At the client, open the Terminal and run the following commands. “dscl” is the “Directory Service Command Line” utility. It is essentially an Open Directory Browser. Don’t type the prompts (“demo:~ admin$” and text before the “>”).

```
demo:~ admin$ dscl localhost
/ > cd LDAPv3/
/LDAPv3 > ls
host10.apple.edu
/LDAPv3 > cd host10.apple.edu/
/LDAPv3/host10.apple.edu > ls
Config
Groups
Users
/LDAPv3/host10.apple.edu > read Users/odadmin
apple-generateduid: A036593C-65E7-11D9-BB90-000A95C4219C
apple-mcxflags: <?xml version="1.0" encoding="UTF-8"?>
```

9. If you can read the contents of a user's record, then you have successfully configured the client to authenticate against the OD master.
10. Restart the client machine (not the server!) to initiate the automounted sharepoint.
11. Log in as one of the users that you created in WGM.

C. Troubleshooting Client --> Directory Service connectivity

Directory services schema are complex and unforgiving. As such, it is easy to make a configuration mistake that is difficult to track down. There are a few tools that you can use to help isolate problems.

1. There are several processes associated with LDAP service on Mac OS X Server:
   - PasswordService
   - kadmind
   - krb5kdc
   - slapd
   - DirectoryService

2. Use the following command to confirm that these required services are running:

```
ps auxw | grep PasswordService
```

The DirectoryService process is the daemon that applications consult to retrieve information from Open Directory. It provides an "abstraction layer" to the directory services; that is, applications do not need to know how to talk to LDAP, AD, NIS, and other various directory services, they only need to know how to talk to the DirectoryService daemon. As everything directory-service related flows through this process, you can learn a lot about your DS configuration by enabling debugging of DirectoryService and watching its debug log.

3. Run these commands in the Terminal at the server to place DirectoryService in debugging mode and to monitor its debug log, then login at the client.

```
sudo killall -USR1 DirectoryService
tail -f /Library/Logs/DirectoryService/DirectoryService.debug.
  log
```

   - type "man DirectoryService" to learn what DirectoryService error numbers mean

DirectoryService provides directory service abstraction to GUI applications through a high-level API (called the DirectoryServices framework). Legacy command-line applications are typically not written to use high-level APIs to access directory information, rather they use the "lookupd" service that is common to Unix platforms. Services like the "login" command and "ssh" are applications that use lookupd to retrieve user information. To maintain consistency between legacy command-line applications and newer applications that use the DS framework, Apple implemented a plugin to lookupd that allows it to query DirectoryService as a last resort. This plugin is called the "DSAgent", and coexists with the other agents that lookupd uses, such as "NIAgent", "DNS", and "Flat Files".

4. Use lookupd in debug mode to browse the directories available to the client.

```
% lookupd -d
> userWithName: odadmin

Dictionary: "NI: user odadmin"
_lookup_agent: NIAgent
...
name: odadmin
passwd: ********
realname: Directory Admin
shell: /bin/bash
uid: 1025

> userWithName: apple

Dictionary: "DS: user apple"
_lookup_agent: DSAgent
name: apple Apple Customer
passwd: ******** ********
realname: Apple Customer
shell: /bin/bash
uid: 1027
```

5. Note that, despite that the "odadmin" user is in both the local NetInfo directory and the LDAP directory, lookupd returns the NetInfo record. Use this command to determine why:

```
> configuration
[hint: you're looking for "LookupOrder"]
[type "?" to see other commands lookupd accepts]
```

6. Packet traces can be extremely helpful in determining if your client and server are speaking to each other. The following commands, executed at the server, will display LDAP-related traffic.

```
tcpdump -i en0 -s 0 -vv port 389
tcpdump -i en0 -s 0 -vv port 389 host 10.0.1.9
tcpdump -i en0 -s 0 -vv port 389 or port 106
```

The following services and ports are used when a Mac OS X client authenticates:

- LDAP 389
- PasswordServer 106
- Secure LDAP 636
- Kerberos 88

7. Verify that the client and server are synchronizing their clocks against the same network time server. Kerberos authentication has strict timing requirements, the clocks on all machines must be within 5 minutes of each other.

IV. Kerberos

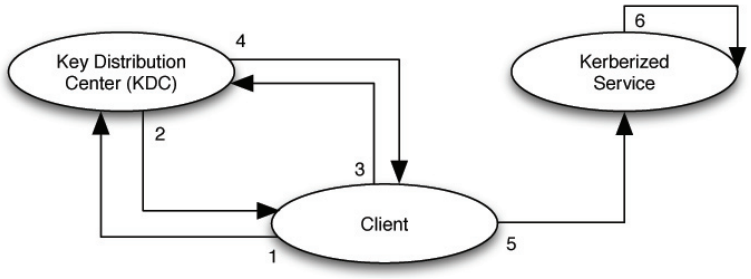
Kerberos is one of the most respected and secure methods of authentication available today. Not only does Kerberos enable single signon (SSO) authentication, it provides authentication without ever passing a password across the network. Kerberos is implemented by Microsoft in Active Directory and by Apple in Open Directory.

Mac OS X Server has a Kerberos Key Distribution Center (KDC) built in. The KDC can authenticate all users whose accounts are stored in a directory domain on the server and whose account password type is Open Directory. Kerberos can authenticate users for the following services of Mac OS X Server:

- Login Window
- Mail services (POP, IMAP, SMTP)
- AFP, SMB, FTP and WebDAV file services
- SSH, SFTP
- Xgrid
- VPN

A. Kerberos authentication process

There are several phases to Kerberos authentication.



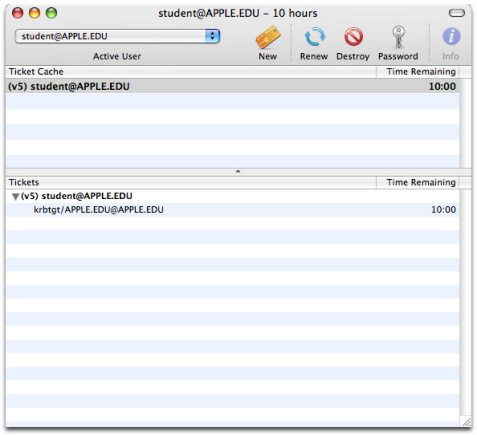
1. The client authenticates to a Kerberos KDC, which interacts with realms to access authentication data. This is the only step in which passwords and associated password policy information needs to be checked. To authenticate, the client sends a login request and is sent back a packet encrypted using the password the server has on record for the user. This packet can only be decrypted if the password that the user provides at the client is the same. The password is never exchanged across the network at this step.
2. The KDC issues the client a ticket-granting-ticket (TGT), the credential needed when the client wants to use the Kerberized service. The TGT is good for a configurable amount of time, but can be revoked before expiration. It is cached on the client until it expires.
3. The client contacts the KDC with the TGT when it wants to use a particular Kerberized service.
4. The KDC issues a ticket for that service.
5. The client presents the ticket for that service.
6. The service verifies that the ticket is valid. If the ticket is valid, use of the service is granted to the client if the client is authorized to use the service. (Kerberos only authenticates clients, it does not authorize them to use services).

Note that the service does not need to know any password or password policy information. Once a TGT has been obtained, no password information needs to be provided.

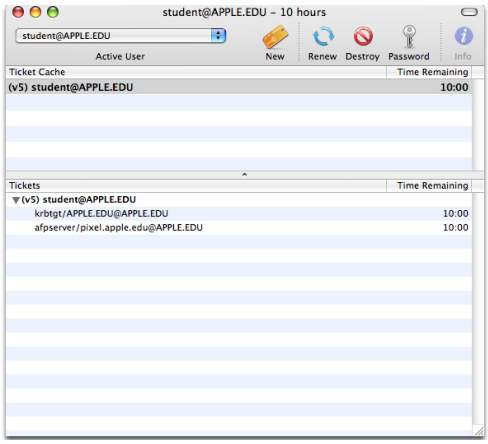
B. Examine the use of Kerberos SSO for client authentication

In this section we will investigate Mac OS X's use of Kerberos to permit single signon authentication.

1. Login to the client as an Open Directory user.
2. In the Finder, navigate to /System/Library/CoreServices and launch the "Kerberos" application. You should see that the user obtained a ticket granting ticket from the KDC in the realm you created.



3. Enable the AFP service on the OD Master (simply "Start" it in Server Admin). Note: If you would like to use a server other than the OD Master for file services (which is strongly recommended), and you would like single signon support, you must add that server to your Kerberos realm to establish trust between the KDC and that server. See pages 83-85 of the Open Directory Administration guide for step-by-step instructions on how to do this.
4. At the client, use the Finder to connect to the AFP server. You will notice that you are not prompted to provide a password. You will also notice the addition of a service ticket for the AFP service of the OD Master.



5. Unmount any server volumes. Using the Kerberos application, destroy the tickets, and try to log in to the AFP server again. This time you should be prompted to authenticate.
6. Examine the /Library/Preferences/edu.mit.Kerberos file. This is the Kerberos configuration file created automatically when you configure the LDAPv3 plugin in Directory Access. Read the "kerberosautoconfig" man page for more information on how this file is generated.

V. Securing the LDAP service

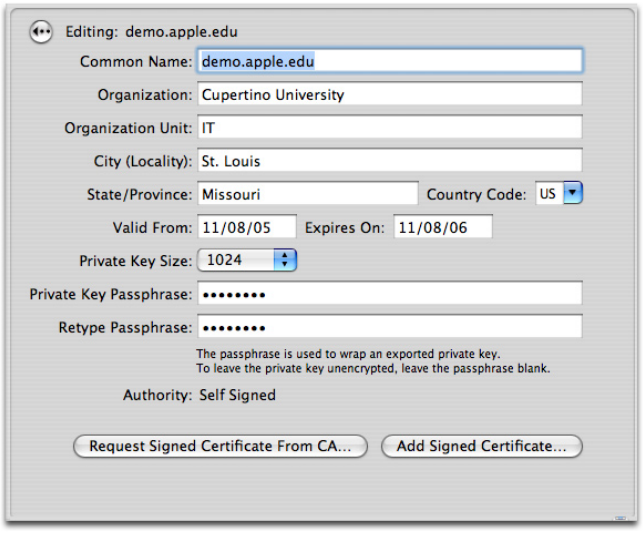
When you communicate with a directory service, information about users is transferred in the clear unless additional steps are taken to encrypt the traffic. In some organizations, such as hospitals or research institutions, this is not only unacceptable, its also illegal. In any case where personal information is transferred across a public network, care should be taken to protect sensitive information.

In particular, there are two kinds of information dealt with in an LDAP transaction: password data and record data. These data are typically kept in separate files on the server, and are (ideally) transacted using different protocols because LDAP is inherently insecure. In the case of Open Directory and Active Directory, password transactions are handled by Kerberos and record transactions are handled by LDAP. Because Kerberos is a pretty secure protocol, the password data is pretty safe. To secure the transaction of record data, administrators typically encrypt the LDAP traffic with SSL.

A. Create a server security certificate

Before you can encrypt traffic with SSL, you need to obtain a security certificate. You must either create a self-signed certificate, or purchase a security certificate signed by a root certificate authority (VeriSign, Thawte, etc.). For the best end-user experience, you should use a certificate signed by an approved root certificate authority. For the purposes of this exercise, you will create a self-signed certificate.

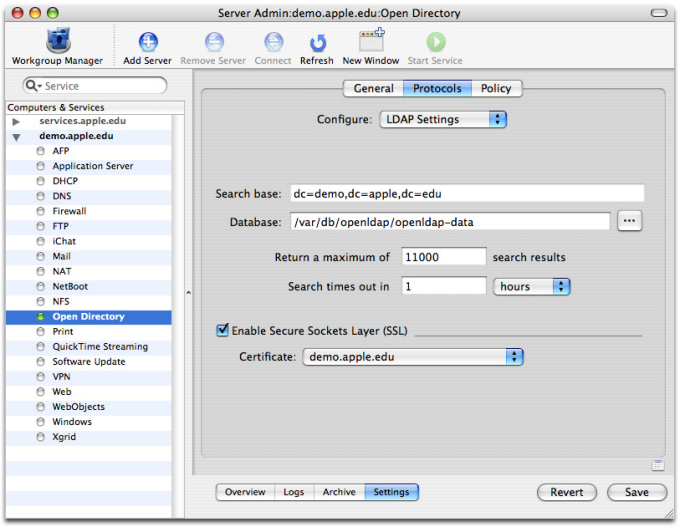
1. In Server Admin, click on the server name in the Computers & Services column.
2. Click on the Settings tab, then on the Certificates tab.
3. Click on the "+" button to create a new self-signed certificate.
4. Fill in the information on the form, then click on the "Save" button. The "Common Name" should be your server's fully qualified domain name (or the CNAME of your web site). If you intended to have the certificate signed, you could click on the "Request Signed Certificate from CA..." button to send the request directly to the CA. Upon receipt of the signed certificate, you would return to this certificate and click on the "Add Signed Certificate..." button.



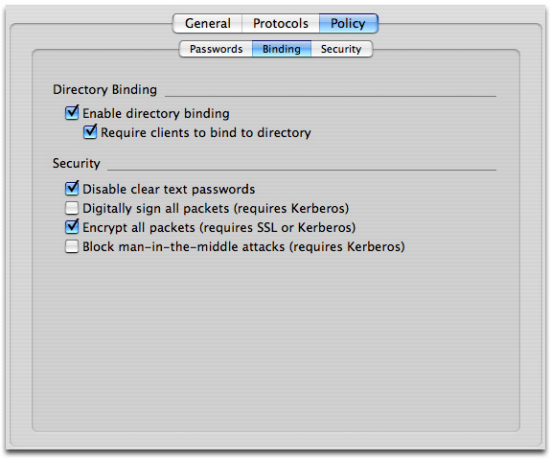
B. Implement SSL on the Open Directory service

Now that you have a security certificate, it is incredibly easy to implement SSL encryption on any Mac OS X Server service that supports it. Perform the following steps on your Open Directory Master to secure the LDAP service.

1. In Server Admin > Open Directory > Settings > Protocols, check the box to “Enable Secure Sockets Layer (SSL)”. Choose your self-signed server certificate in the Certificate popup menu.



2. Click on the Policy tab, then on the Binding tab. By default, directory binding is enabled, but not required. By requiring directory binding, you prevent anonymous access to your directory records. If you require this level of security, check the box to “Require clients to bind to directory”.
3. In the Security section of the Binding tab, there are additional options to increase the security level of the LDAP service. While each of these options improves security, they may also prevent legitimate users from accessing the service. If you have legacy operating systems in your environment, configure your legacy systems to connect to the directory service, then enable these options individually testing that your legacy systems still have access.



Note: You must configure your client machines to use SSL in Directory Access.

VI. Active Directory integration

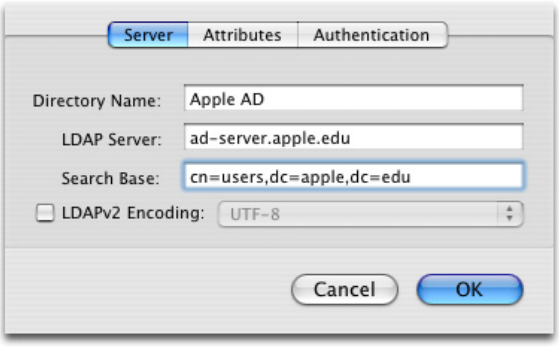
Active Directory is essentially a customized LDAP directory service, and Mac OS X can use AD for authentication using either the LDAPv3 plugin or the Active Directory plugin. Unlike an Open Directory Master, Mac OS X clients do not have prior knowledge of the schema implemented by each and every Active Directory server. As such, use of the LDAPv3 plugin requires considerably more configuration, as well as knowledge of the attribute mappings used by AD. The Active Directory plugin, on the other hand, behaves similar to Windows clients and automatically discovers Domain Controllers in the specified forest and parses the attributes present on the AD for information applicable to Mac OS X.

This section will focus on configuring the Active Directory plugin to authenticate against an Active Directory server. The last page of this document is a worksheet that you can use to keep track of AD settings.

A. "Feeling out" the Active Directory

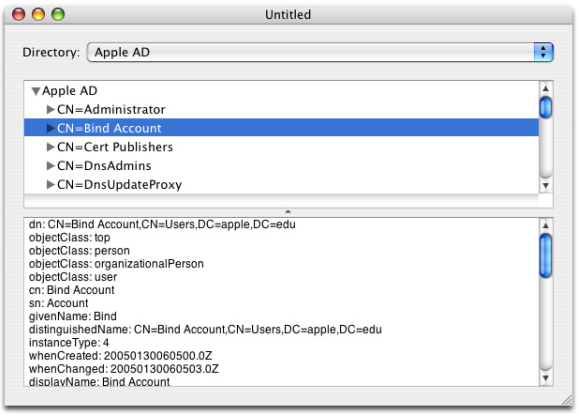
Before simply diving into AD binding, it is often beneficial to verify that you can connect to and search the directory service. We'll use the application "LDapper" to browse the directory service.

1. Launch the LDapper application (you can find this application using http://versiontracker.com).
2. Select "Preferences" from the LDapper menu and click on the "+" icon to add a new directory service. Configure the Directory with settings appropriate for your AD environment.



3. Active Directory does not allow anonymous binding by default. Click on the Authentication tab and provide the AD account and password of a user that has the privilege to "Join a computer to the domain".
4. Click "OK", then, in the "Default Search Options" of LDapper's preferences, set the Fetch popup menu to "All Attributes". Deselect the "Discard responses without email" and "Search for people only" checkboxes". Click OK to close the preferences window.

5. Select “New Browse Window” from the File menu and click the disclosure triangle next to your AD to view the user records in the AD. Click on a record to view its contents.



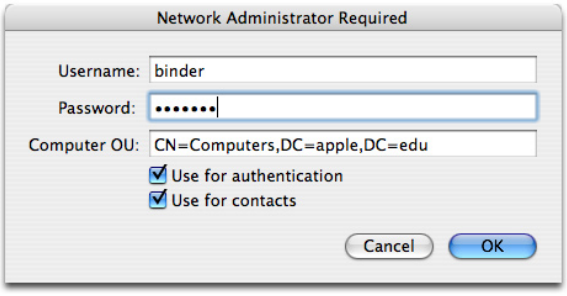
The user record indicates what data exists in each attribute present on the AD server for user records. This is particularly handy when doing a custom mapping for the LDAPv3 plugin.

6. Edit the directory configuration in LDapper’s preferences to use your Computer OU as the search base instead. Open a new browse window and browse through the computer records.

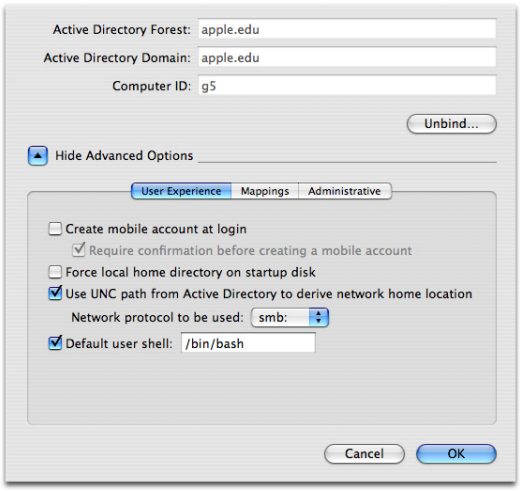
B. Configure the AD plugin

Because the AD plugin uses DNS to locate AD resources on the network, there is very little configuration necessary for the AD plugin. Use the settings provided by your AD administrator or those provided on the last page of this document.

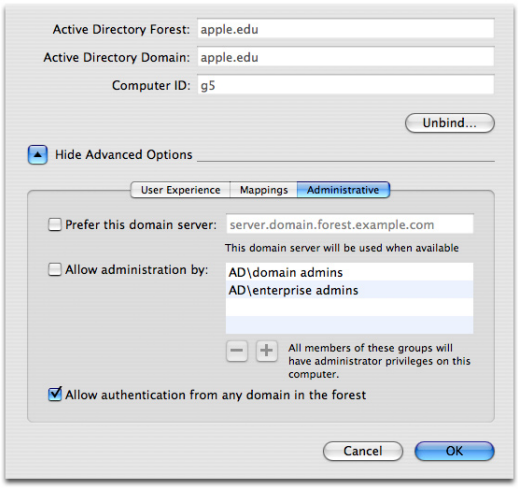
1. Launch the Directory Access application, authenticate, and click on the Active Directory plugin checkbox. Click on the “Configure...” button.
2. Provide the directory domain and a computer ID, then click on the “Bind” button and provide your AD credentials (or bind credentials provided to you by your AD administrator). Consider the Computer OU carefully. The default Computer OU may not exist, may not be appropriate, or may not be accessible with your account privileges. The bind will fail if your account does not have write access to the Computer OU, so ask your AD administrator what the appropriate Computer OU is for your computer. Also, the computer ID should not be longer than 19 characters. In general, it is best practice to use the first part of the DNS hostname.



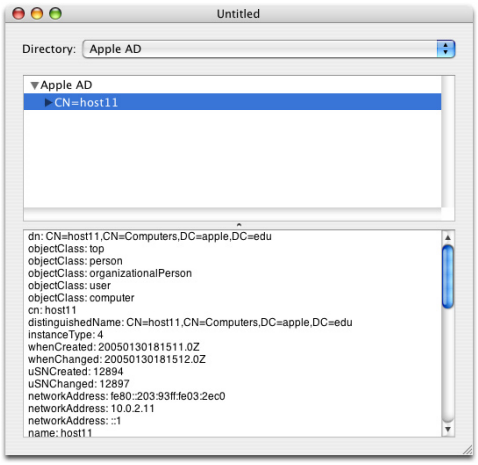
3. Click on the “Show Advanced Options” button. Consider the options in the User Experience tab:
   - “Create mobile account”: causes the client to cache the account credentials of the last user to use the machine. This can be handy if your users take their machines home.
   - “Force local home”: This should be checked if your AD does not specify the location of user home directories, or if you do not want users to have network-based home directories.
   - “Use UNC path from Active Directory to derive home location“: If your AD user accounts indicate the path to a home directory, this option allows the AD plugin to convert the value to a URL that can be used to mount the sharepoint upon login. If you do not specify the correct network protocol, an error will occur when users try to login.



4. Consider the options in the Administrative tab:
   - “Prefer this domain server”: If you have a preferred domain server, indicate it here. If the server becomes unavailable, the AD plug-in automatically falls back to another nearby server in the forest. By default, the AD plug-in automatically determines the closest AD domain in the forest.
   - “Allow administration by”: This allows you to specify AD groups whose members should have administrative privileges on the machine.
   - “Allow authentication from any domain within the forest“: If your forest has multiple domains, this essentially expands the search base that the AD plugin uses to find user records so users from other domains can log in to the machine.



6. When the bind has completed, return to the LDapper application and find your computer's record in the Computer container.



7. Return to Directory Access and click on the "OK" button to close the AD plugin window.
8. Click on the "Authentication" tab. If your machine is configured with an LDAP node from a previous exercise, remove that node from the list, click apply, then close Directory Access.

C. Verify directory connectivity

Again, we don't want to immediately try logging in as an AD user without verifying that the Directory bind is working as we expect.

1. In the Terminal, use the dscl command to navigate to the Active Directory node and read a user record. Don't type the prompts ("client:~ admin$" and text before the ">").

```
client:~ admin# dscl localhost
/ > cd Active\ Directory/All\ Domains/
/Active Directory/All Domains > ls
Computers
Config
Users
/Active Directory/All Domains > cd Users/
/Active Directory/All Domains/Users > ls
administrator
binder
guest
krbtgt
labadmin
student
/Active Directory/All Domains/Users > read student
...
```

2. Alternatively, you can specify arguments to the dscl command to read the student's record:

```
% dscl /Active\ Directory/All\ Domains -read /Users/student

ADDomain: apple.edu
cn: Student Account
displayName: Student Account
distinguishedName: CN=Student Account,CN=Users,DC=apple,DC=edu
givenName: Student
homeDirectory:
homeDrive:
name: Student Account
primaryGroupID: 513
sAMAccountName: student
sAMAccountType: 805306368
sn: Account
userPrincipalName: student@apple.edu
AppleMetaNodeLocation: /Active Directory/apple.edu
AuthenticationAuthority: 1.0;Kerberosv5;86C47B88-6506-4F64-
  8E1D-73A74071A391;student@APPLE.EDU;APPLE.EDU;
FirstName: Student
GeneratedUID: 86C47B88-6506-4F64-8E1D-73A74071A391
NFSHomeDirectory: /Users/student
LastName: Account
PasswordPlus: ********
PrimaryGroupID: 20
RealName: Student Account
RecordName: student student@apple.edu APPLE\student
SMBAccountFlags: 805306368
SMBGroupRID: 513
SMBHome:
SMBHomeDrive:
SMBLogoffTime: 0
SMBLogonTime: 127515826632546368
SMBPasswordLastSet: 127515390413065200
UniqueID: 113539976
```

3. Use the dscl command as indicated above to read the AD-plugin generated record for one of your AD users and compare it to the record displayed in LDapper. The Active Directory plugin generates some of the attributes dynamically based on other information in the user record. For example, the "NFSHomeDirectory", "UniqueID" and "PrimaryGroupID" appear in the output, but do not necessarily exist in AD.

D. Home directories and the AD plugin

By default the AD plugin will create a local directory in /Users as a home directory and mount a Windows network home as an SMB share on the desktop. You can configure this behavior in the "Advanced options > User Experience" section of the Active Directory plugin. You can also use the dsconfigad command-line tool to configure the AD plug-in to configure this behavior. Additional information about the use of dsconfigad to modify these settings is located in this Kbase article: http://docs.info.apple.com/article.html?artnum=107943

At this point we should take a moment to understand the difference between "Using a Network Home Directory" vs. "Mounting network home at login". They sound very similar, the critical difference is that the former actually mounts the network share at the user's home directory location (e.g., /Users/username) whereas the latter uses a local home directory and mounts the network share on the Desktop. With a network home directory, all of the user's preferences and documents are saved directly to the server. With a mounted network home, preferences are saved on the local machine and the user must make the effort to save documents to the network home (or risk losing them).

Network home directories can be very beneficial for students, but they can also be a management headache. Internet Explorer, for example, creates a 10MB cache file as soon as you launch the application. If a lab full of students is instructed to log in and launch IE, your file server will take a hit. If your lab machines are wireless, if your network has low bandwidth, or if your users intend to use multimedia applications such as the iLife suite, you should strongly consider avoiding the use of network home directories and mount the network home on the Desktop instead.

You should also carefully consider which protocol to use for hosting your network home directories. While both AFP and SMB are supported, SMB has many shortcomings that make it less-than-ideal for hosting a home directory. Among other things, SMB supports only 31 characters in a file name, which means some applications will not be able to write out their preferences files (~/Library/Preferences/ByHost/com.apple.Classic.000393a6d5e4.plist). Apple recommends using AFP home directories hosted on Mac OS X Server.

E. Automating AD binding

When you bind a computer to Active Directory, a computer account is created and named with the Computer ID you provide to the AD plugin. As this computer account establishes a level of trust and computer authentication to the domain, every computer bound to the AD domain must have a unique account. This can pose a challenge to administrators in charge of large computer labs or hundreds of computers. To ease this burden, Mac OS X includes a utility that can automate the process of binding. The "dsconfigad" tool can bind your computer to Active Directory and configure all behaviors of the AD plugin.

One word of caution: automating the bind process requires that you store the password of the bind account in a shell script. This has obvious security implications. To mitigate risks, 1) use a restricted account created for the sole purpose of binding machines to the domain, 2) change passwords frequently, 3) restrict access to the shell script, 4) destroy your shell's history files. The following instructions assume that you will either precede the commands with "sudo" or include the commands in a script that is run with root privileges.

1. Read the man page for "dsconfigad" to learn how to use it and what settings it can modify.

2. Use the following command to see the current configuration:

```
dsconfigad -show
```

3. If you are already bound to AD, destroy the bind with this syntax:

```
dsconfigad -r -u binder -p 'password'
```

4. Bind to AD using the following syntax:

```
dsconfigad -f -a "computerid" -domain "apple.edu" -u "binder" -p
  'password' -ou "CN=computers,DC=apple,DC=edu"
```

5. Configure advanced AD plugin options with this syntax:

```
dsconfigad -alldomains enable -localhome enable \
    -protocol afp -mobile disable -mobileconfirm disable \
    -useuncpath enable -shell "/bin/bash" -nopreferred \
    -groups "AD\Mac Admins"
```

6. Add the AD node to the search path using these dscl commands:

```
dscl /Search -create / SearchPolicy CSPSearchPath
dscl /Search -append / CSPSearchPath "/Active Directory/All
  Domains"
dscl /Search/Contacts -create / SearchPolicy CSPSearchPath
dscl /Search/Contacts -append / CSPSearchPath "/Active
  Directory/All Domains"
```

Directory Binding must occur while booted from the production OS. That is, a directory bind cannot be performed as a post-action to an image deployment process -- the bind process makes changes to items in /Library/Preferences/DirectoryService on the boot drive. There are two methods for automating the bind procedure such that it occurs soon after image deployment. One method involves creating a startup item that runs the bind script after an imposed delay (it takes a moment for DirectoryServices to become established, and the loginwindow will not necessarily wait for this to occur). Another method involves running the bind script as a login hook. Each method produces the same result, although the login hook procedure may be more robust because you can force the loginwindow to delay presentation until DirectoryServices is available. An example script that performs the bind, configures the AD plugin, adds the AD node to the search path, disables auto-login and securely removes itself (and the bind password) from the machine is referenced in the References section at the end of this document.

To implement the login-hook method:

1. Install the bind script on your master machine
2. Configure the bind script to run as a login hook:

```
sudo defaults write /var/root/Library/Preferences/com.apple.
  loginwindow LoginHook /path/to/bind_script.sh
```

3. In the Accounts preference Pane, configure the Login Options to automatically login as any local user.

F. AD plugin Troubleshooting

All of the troubleshooting methods for Open Directory discussed previously apply to troubleshooting the AD plugin. There are a few additional things to verify if you're having trouble:

Bind issues:

- Verify that the clocks on the client and server are within five minutes of each other. Kerberos authentication has strict timing requirements, it is recommended that you have all of your machines sysnchronized to the same network time server
- Verify that the Network Administrator account that you are using has write privileges for the Computer OU that you are using
- Verify that the Network Administrator account you are using is correct (use the sAMAccountName and password)
- Verify that your client is using the DNS server that is used (and thus populated) by Active Directory. Your AD administrator should be able to verify this
- Verify that you can communicate to the server's ports: 53, 88, 137, 389, and 445

"You are unable to login to the user account "Student" at this time..."

- Use "dsconfigad -show" to determine how user home directories should be mounted
- Then use the dscl command to read a user's record:

```
dscl /Active\ Directory/All\ Domains -read /Users/student
```

- Determine where the user's home directory is located (the "HomeDirectory" attribute) and verify that you can mount it manually at the client using the protocol indicated
- Use "dsconfigad -mountstyle AFP|SMB" to modify the mount method if necessary

Anything else

- Use the adcheck utility to see if it indicates any errors. Your local Systems Engineer can help you with the acquisition and use of this tool.

VII. Group and Computer Management using OD + AD

It is fairly common for Active Directory administrators to refuse to extend the AD schema to provide support for the Mac OS X native attributes. In Windows 2000 Server, it was understandable because it was impossible to undo schema changes. If you make a mistake, you're stuck with it unless you want to rebuild the entire directory. In Windows 2003 Server however, you can deprecate schema changes. While this makes the situation significantly better, it is still not very flexible and schema changes are not something to be taken lightly.

As such, Mac administrators are typically left with the reality that they can use the AD for basic authentication, but cannot leverage it for greater management functionality such as controlling Finder preferences or access to specific applications. Fortunately, there is a middle ground. This section will describe how to use an Open Directory Master in conjunction with an Active Directory server to manage user, group, and computer attributes.

To understand how dual-directory management works, it is very important to understand LDAP's use of container objects. For the sake of simplicity, I will limit the discussion to the three most common container objects: users, groups, and computers.

When a user logs in, several things occur:

- authentication (username/password verification)
- authorization to use the computer
    - is the user in a group that is authorized to use the computer?
    - are any limits imposed on this particular computer, user, or group?
- authorization to use network services (e.g. to access a home directory server)

The operating system uses Kerberos to handle the authentication, but authorization is a bit more complex. Authorization can be granted to a user, a group, or a particular computer. When the OS looks for authorization privileges, it follows the same path as for authentication: local NetInfo directory, then any configured network directories. For example, when you log in, the system first looks to the local NetInfo directory for an account matching your username. If it doesn't find it there, it looks in Active Directory. If it doesn't find it there, it looks in the next directory service in your authentication path. If it never finds a match, login is refused. After finding your account, it continues looking for objects referenced in your account, such as group membership. Additionally, Mac OS X performs a search for any objects matching your computer's MAC address.

It is important to note that a directory service client (e.g. Mac OS X) cannot pull information about a single object (e.g. a user) from multiple directories. For example, if a user logs in with an AD account, all information about that user must come from Active Directory -- you cannot draw user-specific management information or a home directory location from yet another directory service. When the system looks for groups that contain that user, however, it continues to search all configured directories for matches (because the groups to which a user belong are separate "objects" outside of the user object). If your OD server is configured in your authentication path in addition to an AD server, and the AD user logging in belongs to an OD group you have configured, the system will impose any restrictions on that OD group to the AD user. Computer records can be managed similarly offering an additional layer of management.

Details of what features can be managed of users, groups, and computers are discussed in the next section. This section will explain how to configure a Mac OS X Server Open Directory Master joined to an AD Kerberos realm and how to configure a Mac OS X client to use a dual-directory infrastructure.

A. Mac OS X Server configuration

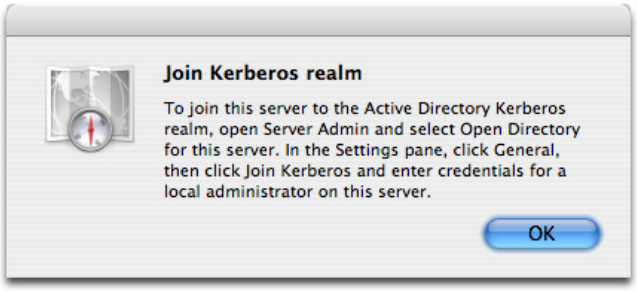
In this scenario, the AD server is your Kerberos server. To have knowledge of AD users, your Open Directory Master must be bound to the Active Directory server. Additionally, if you plan to host "Kerberized" services such as AFP, FTP, or Mail on another Mac OS X Server, you must "establish trust" between those services and the AD Kerberos realm to enable SSO support.

Joining a Mac OS X Server service to a Mac OS X Server Kerberos realm is fairly easy and buttons are provided in Server Admin for this purpose. Joining an Open Directory Master to a Windows AD Kerberos realm requires a couple additional steps.

1. Promote your Mac OS X Server to an Open Directory Master following instructions provided in section II.
2. Because you are using AD for User authentication, the OD Kerberos realm is unnecessary. Additionally, it can cause conflicts with the AD Kerberos realm for for Mac OS X clients bound to AD. Destroy the OD Kerberos realm with the following commands (substituting your OD admin name and password):

```
sudo sso_util remove -k -a diradmin -p password
dscl -u diradmin /LDAPv3/127.0.0.1 -delete /Config/KerberosKDC
dscl -u diradmin /LDAPv3/127.0.0.1 -delete /Config/
  KerberosClient
```

3. Bind your Mac OS X Server to Active Directory using instructions from section VI. When you perform the bind on Mac OS X Server, you will get an additional dialog:



Because your server is also an Open Directory Master, the "Join Kerberos" button will not appear in Server Admin. You can use the command-line tool "dsconfigad" to join the realm instead.

4. Run this command in the Terminal to configure the kerberized services on Mac OS X Server with the AD service principals:

```
sudo dsconfigad -enableSSO
```

5. Verify that your server is bound to AD and that your keytab has entries by running the following command in the terminal:

```
sudo klist -ke
```

You should see three entries per Kerberos realm for each service offered in Mac OS X Server (there should be about a dozen unique services).

6. Also verify that your services are configured to use the AD Kerberos realm, not your own. The AFP and SMB services are easy to check, run the following commands in the Terminal:

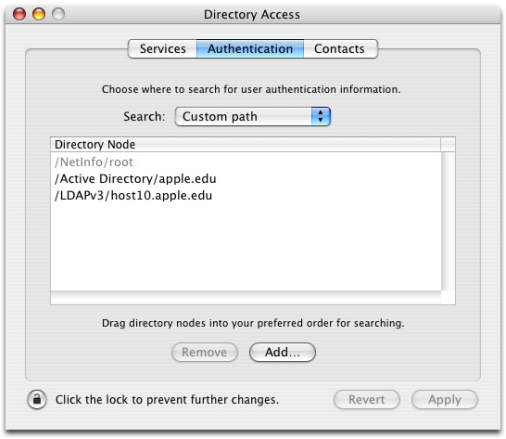
```
defaults read /Library/Preferences/com.apple.AppleFileServer
  kerberosPrincipal
grep “realm” /etc/smb.conf
```

You should see your AD server's realm listed in each instance. If you do not, unbind from, then rebind to the AD domain, then run the “dsconfigad -enableSSO” command again and repeat the verification steps above.

B. Configure Directory Access at the client

The client-side configuration is simply a combination of LDAPv3 configuration/binding to OD and AD plugin configuration/binding.

1. Bind the client to Active Directory using the AD Plugin (Review Section VI).
2. Configure the LDAPv3 plugin to connect to your OD Master (Review Section III).
3. In the Authentication tab, select “Custom path” from the Search popup menu. Add the Active Directory and LDAPv3 nodes to the search path. Order is important. The directory that you use for user objects -- probably Active Directory -- should be first (first after /NetInfo/DefaultLocalNode, which cannot be overridden).



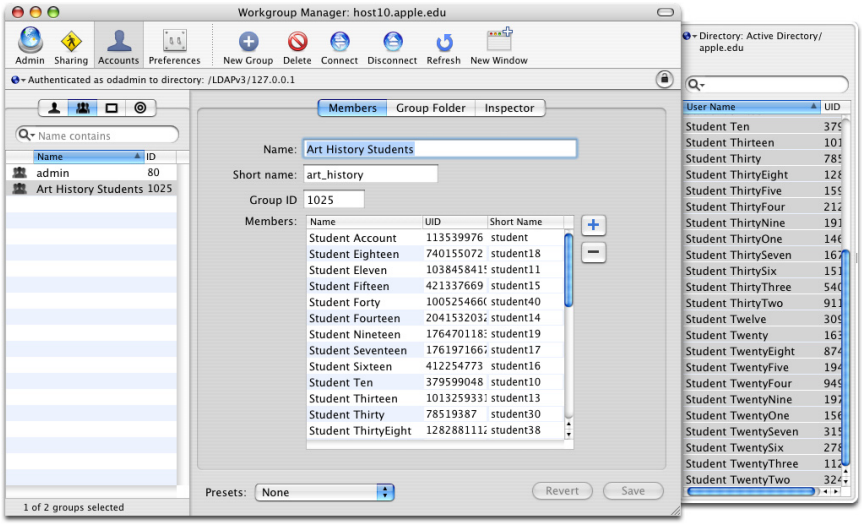
C. Configure Open Directory Groups

If your AD administrator does not extend the AD schema to support the user attributes that Mac OS X uses for user management, then you cannot manage individual users. Likewise, if the group attributes that allow Mac OS X to manage groups are not added to the AD schema, you also cannot manage AD groups using AD alone. If you have a dual-directory service setup, however, you can manage users and groups based on their Open Directory group membership.

Mac OS X Tiger supports nested groups, which means it will recognize groups within groups. If you would like to manage users based on AD group membership, you can create groups in Open Directory and add Active Directory users and groups to them. The actual AD user and AD group records remain within AD, while the OD group contains only references to those users and groups. In addition to

these references, the OD group may contain management information that cannot be stored in AD.

1. Log in to Workgroup Manager at your Open Directory Master and confirm that you are working in the /LDAPv3/127.0.0.1 node.
2. Click on the Groups tab and create a new group.
3. Click on the "+" icon to open the users and groups drawer. At the top, select the Active Directory node -- Active Directory users should appear in the list. Select some user accounts (including your own) and drag them into the Members list, then click on the groups tab on the right and repeat with some group records. You now have an Open Directory group of Active Directory users and groups. Again, this does not copy the users, it merely makes references to the users, therefore you do not have to deal with synchronizing any accounts.



4. Management of preferences could fill another book and will not be covered in detail here, however you should modify at least one preference to verify that management by group is working. Be sure your group is selected, then click on the Preferences button in the toolbar. Click on the "Dock" button, then on the "Dock Display" tab. Set "Manage these settings" to "Always" and "Position on screen" to "Right".

D. Configure Open Directory Computer Lists

This step is optional -- you could simply do management based on the group a user belongs to. However, management by computer lists is handy if there are thousands of users in your AD domain or if management is easier based on computer location.

1. In WGM, Click on the Accounts button in the toolbar, then click on the Computers tab (the square icon).
2. Create a new computer list. Name it "Test Lab", then click on the button with the ellipsis. A network browser will appear allowing you to add machines in your subnet to the list. Find your client and add it to the list.
3. Click on the "Access" tab. Select "Restrict to groups below" to limit what group of users may log in to the system. Click on the "+" icon and drag AD groups or the OD group you created previously into the table. Click Save.
4. As with the Group configuration, click on the Preferences button in the toolbar and establish a

forced preference that will be easy to detect when you login.

E. Login at the client to test your dual-directory setup

1. Login using your AD account. Notice that the dock appears on the right. You are managed!
2. Log out, then log back in as another AD user that is in an administrative group (according to the AD plugin configuration in Directory Access). Because this user is an administrator, it gets an additional dialog upon login to opt-out of Workgroup Management.
3. Log out, then log back in as another AD user that is not in any of the groups in your access list. You will be denied login because that user is not a member of the group that you created.

4. Review section IV(B) about Kerberos, then examine your Kerberos tickets while logged in with your AD account.
5. Enable the AFP and Windows services on your OD Master. At the client, mount a sharepoint from the server using AFP and SMB (for SMB, you must explicitly type “smb://your.server.edu”). Re-examine your Kerberos tickets. This functionality will be explored in more detail in Section IX.

VIII: Manage Groups and Computers: Workgroup Manager

Now that you know how to configure Mac OS X Server to provide group and computer management services, it might be helpful to learn about some of the specific settings that you can manage. There is a document from Apple that covers this topic in full detail. This section serves only to give you an idea of what management features are available.

A. Install the Server Administration tools

Mac OS X Server can be administered from any other Mac -- you do not have to log in to or sit in front of your server to manage groups, computers, or any other services for that matter. In fact, many servers don't even have a video card because its easier to manage Mac OS X Server from the comfort of your own office.

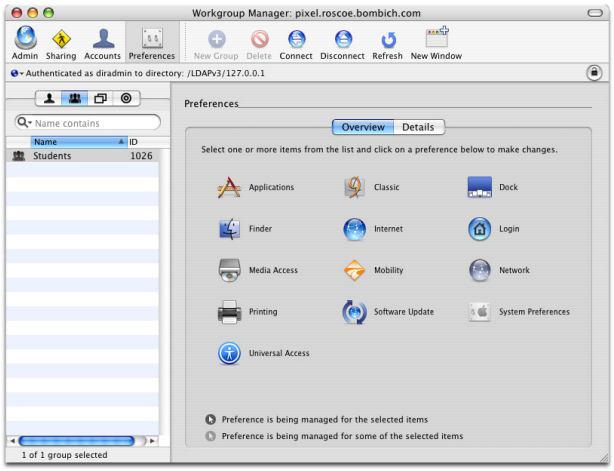
1. Insert the Server Administration Tools CD (from the Mac OS X Server Installation disc set) into your client machine.
2. Double-click on the Server Administration Software package to install the Server Admin tools. The tools will be installed into /Applications/Server.

B. Explore Managed settings in Workgroup Manager (WGM)

A directory service is really nothing more than a big database filled with information about users, groups, computers, and other network services. Group and computer management settings are also stored in this database. LDAP, or Lightweight Directory Access Protocol, is the protocol used to communicate with this data store (add data, change settings, etc). WGM is an application that "speaks LDAP", and has a customized interface for specifically managing management settings.

Managed settings are stored in an xml-formatted property list within user, group or computer records. The attribute is named "MCXSettings" -- MCX stands for "Managed Client for Mac OS X", you'll see many references to this while working with management settings. While these property lists can modified by hand, you typically will not have to do that, you can use the customized WGM GUI for modifying management settings.

1. Launch the Workgroup Manager application in /Applications/Server and log in to your server.
2. Click on the Groups tab and select the group that you created in the previous section.
3. Click on the "Preferences" button in the toolbar.



4. Click on the “Applications” icon. At the top of the content pane there is a box that indicates whether these settings should “Never” be managed, should be managed “Once”, or should “Always” be managed.

“Never” is obvious -- that setting won’t be managed. A setting that is managed “once” means that it is set as a default for the first time that setting is accessed (when the user first logs in, or when the computer is first used). The setting can be changed by the user as he uses the computer. This setting is not applicable if it is a setting that could not be modified by an end-user. A setting that is “always” managed cannot be changed by the end user.

5. Click on “Always”, then examine your options for limiting application usage for members of the selected group. You can limit group members to only applications in the list, or all applications except those in the list. You can also control whether applications can launch non-approved applications (some applications have “helper” applications bundled with them), and whether Unix tools can be run.
6. If you’d like to apply the settings, click on the Apply Now button, otherwise click on Revert, then on the “Done” button to return to the previous view.
7. Click on each management item to see what management capabilities it offers. What you may notice is that many of the settings you previously controlled using shell scripts or default user accounts can be managed right here. Some of the things you can manage are:

   - What items appear in the Dock, and where and how the Dock is displayed
   - What users can do in the Finder (restart, shutdown, eject media, burn CDs/DVDs)
   - How items are displayed in the Finder (window and folder settings, views, etc.)
   - Email and web browsing default settings
   - Mounting network volumes on login, performing other tasks on login
   - Running login and logout scripts
   - Loginwindow preferences, forced logout after inactivity
   - Wake, sleep, display dimming, and other “energy saver” settings
   - Scheduled startup and shutdown
   - Synchronization of user home directories to a network server
   - Proxy settings
   - What printers are available, whether administrator authentication is required for printing
   - What Software Update server to use
   - Access to System Preference Panes
   - Universal Access settings (settings that help those with disabilities)

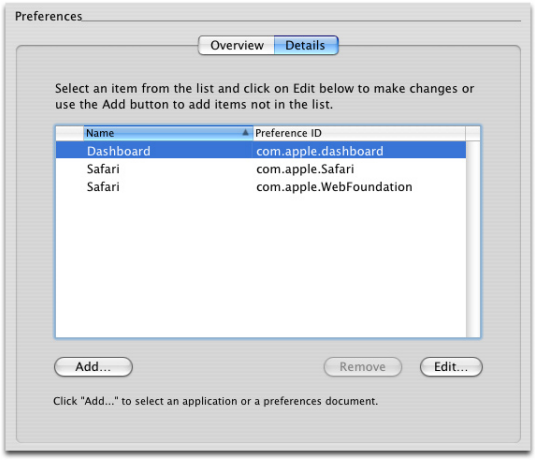
C. Extending management beyond WGM: Preference manifests

As you've seen, Workgroup Manager gives you a great amount of control over the end-user experience. You'll notice, however, that WGM does not list specific application preferences. What if you want to have much more granular control over how users use specific applications? That's where preference manifests come into play.

Preference manifests list some or all of the preference settings available within an application, and indicate how those settings can be managed. Manifest files are located within an application's package, and WGM parses this file when you add the application in WGM's "Details" tab. If an application does not have a manifest file, WGM imports the preferences file from your home directory and uses your settings instead. Note that this imports the current settings (*your* preferences), which may overwhelm you with a number of extraneous and unnecessary options, as well as settings that apply only to your own account and fail to work for other users.

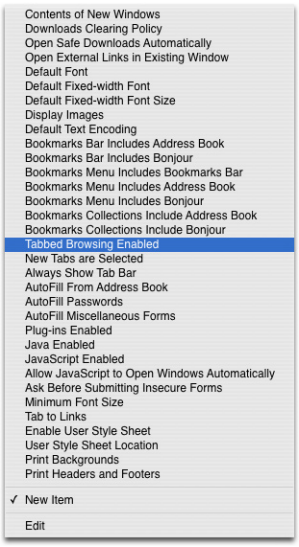
Instead of simply choosing an application, you should import a preferences file that you've manually stripped to contain only the preference settings you want to manage. Ideally, you should create a preference manifest to make the addition of managed settings for your application much easier to manage. Creating and editing preference manifests is beyond the scope of this document, refer to pages 181-183 of the User Management Guide available at http://www.apple.com/server/documentation/. For an example manifest file, we'll take a look at Safari's preference manifest.

1. Launch the Workgroup Manager application in /Applications/Server and log in to your server.
2. Click on the Groups tab and select a group (created in a previous section).
3. Click on the "Preferences" button in the toolbar.
4. Click on the "Add..." button. In this panel, you may choose an application or a preference file. Navigate to /Applications and choose Safari. **Uncheck the "Import application's preferences" box**. Click the Add button.

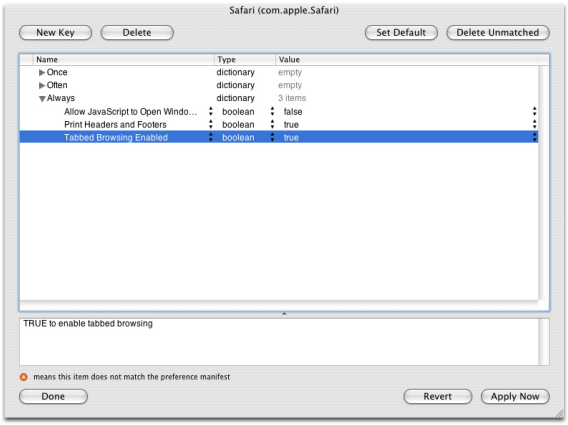


5. You should now have two preference items. Click on the Safari item identified by "com.apple.Safari" and click on the Edit button.
6. Click on the "Always" item to enable the "NewKey" button. Add a new Key.

7. Next to the “New Item” entry is a pop-up button. Click on the arrows to reveal all the preference attributes available within the manifest:



8. Choose “Tabbed Browsing Enabled”. The type should be automatically set for you, and the value will be set to the default (false). Change the value to true to enable tabbed browsing.



9. Repeat for additional preference attributes if desired. When you’re finished, click on the Apply Now button, then click on the Done button.
10. Imported preference manifests are stored in your home directory at ~/Library/Preferences/com.apple.mcx.manifests. Navigate to this directory and command- or right-click on com.apple.Safari.manifest; choose “Show Package Contents”. Navigate to Contents/Resources and open the com.apple.Safari.manifest file in TextEdit. Compare the format and content of this file to your own Safari preferences file at ~/Library/Preferences/com.apple.Safari.plist.
11. Log in to your client machine as a user within your managed group. Verify that the managed settings and managed Safari preferences are applied to that user.

IX: Leveraging AD for other Mac OS X Server services

In the previous sections you learned how to provide directory services on a Mac OS X Server in addition to those provided by Active Directory such that you can have greater management control over groups and computers within your department. In addition to group and computer management, Mac OS X Server can provide several services that leverage the user accounts stored in Active Directory. Not only can Mac OS X Server use AD for authentication, but by joining your server to the AD Kerberos realm, you can extend single-sign-on support for serveral Mac OS X Server services.

In this section you will learn about the services that Mac OS X Server provides that can leverage AD single-sign-on authentication.

A server hosting an Open Directory Master should not provide other server services. If the number of clients is really low, say less than ten, this could be done (that is, there are no technical reasons that it won't work). In general, however, Apple recommends hosting Directory Services alone on one machine, and hosting other services such as file and mail services on other servers. For the following exercises, it is assumed that the server is not an OD Master.

A. Bind to AD, join Kerberos

To take advantage of AD for authentication of users to your server's services, you will need to bind your server to the Active Directory domain and join it to the Kerberos realm. The bind allows the server to escalate authentication requests to AD and joining the Kerberos realm enables single-sign-on for some services.

The steps will not be repeated here, simply follow the instructions in section VII (A) at your server. Because you do not need to (and should not) promote your services server to an OD Master, you should skip steps 1 and 3 in section VII (A). One additional note: In the AD Plugin configuration Advanced Options section, be sure to uncheck the box to "Use UNC Path from Active Directory to derive network home location". If this option is left checked, every time the server reads a user record, it will check that the network home is available. This could slow the server down considerably if home directories are not available at your services server.

B. File Services: AFP

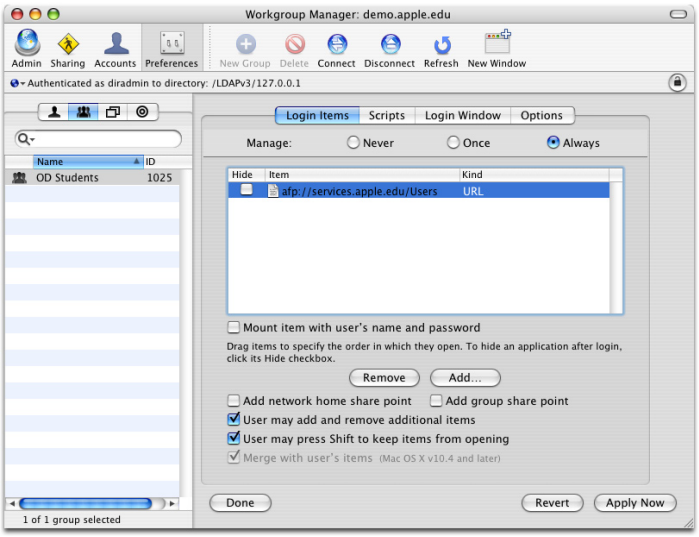
How would you like your Mac users to log into a workstation and automatically have file shares open with their credentials?

The AppleFileServer does not require any additional configuration to take advantage of AD authentication and single-sign-on. Simply bind your AFP server to the AD and join it to the Kerberos realm. Note that in Server Admin > AFP > Settings > Access, you can specify "Standard", "Kerberos", or "Any Method" for authentication. You should leave this setting at "Any Method" (the default) for maximum flexibility. Remember that Kerberos is not required for secure authentication, it is a service of convenience to your end users.

To verify that the AFP service is allowing SSO access, do the following at a client bound to AD.

1. Log in as an AD user
2. Verify that you initiated your Kerberos tickets (review section IV,B)
3. In the Finder, select "Connect to Server..." from the "Go" menu and enter the address of your server. You should bypass the authentication dialog and get a list of shares to mount.
4. Again, examine your kerberos tickets, you should have an additional afpserver service ticket for your server.

There are several different methods for getting sharepoints to mount automatically upon login. If your client machines are bound to an OD master in addition to your AD server, you can add a sharepoint to the Login preferences for your OD group or computer list.



You can also run a login script that initiates the connection:

1. Log in to your client machine as a local administrator
2. In the Terminal, create a shell script at /Library/Management/login.sh containing:

```
#!/bin/sh
open afp://services.apple.edu/Users
```

3. Make the script executable:

```
sudo chmod a+x /Library/Management/login.sh
```

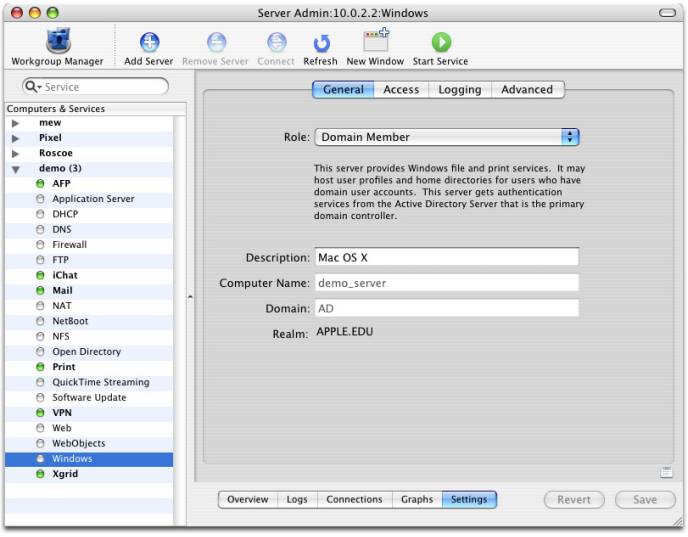
4. Make the script run at login (all one line):

```
sudo defaults write /var/root/Library/Preferences/com.apple.
  loginwindow LoginHook "/Library/Management/login.sh"
```

5. Now log in as an AD user. The sharepoint should mount without any user interaction or password prompts.

C. File Services: SMB

SMB file services are enabled in Server Admin > Windows. Like the Apple File Service, no additional configuration is required to take advantage of AD authentication and single-sign-on. Simply bind your SMB server to the AD and join it to the Kerberos realm, then turn on the Windows service in Server Admin. As long as your Mac or Windows users are logged in to their machine using an AD account, they should have SSO access to your file services.



To verify that the Windows service is allowing SSO access, repeat the procedure from the previous exercise, connecting to your server explicitly via smb://your.server.edu.

D. File Services: FTP

FTP file services are enabled in Server Admin > FTP. Like the AFP and SMB services, there is no additional configuration to take advantage of AD authentication and single-sign-on. Simply bind your FTP server to the AD and join it to the Kerberos realm, then turn on the FTP service in Server Admin.

Note that in Server Admin > FTP > Settings > General, you can specify “Standard”, “Kerberos”, or “Any Method” for authentication. You should leave this setting at “Any Method” (the default) for maximum flexibility.

Mac OS X does not include a Kerberized FTP client. Fetch, from Fetch SoftWorks, does support FTP via GSSAPI. While Fetch works fine for connecting to a Mac OS X Server KDC, there is currently a bug in xftpd that causes an SSO FTP connection to fail if the user obtains a forwardable ticket from an AD server. If you would like to extend SSO support to your FTP users, use the Kerberos application (/System/Library/CoreServices/Kerberos.app) to modify the default Kerberos preferences to not obtain a forwardable ticket. Refer to the Section VIII (C) to learn how to force this preference from your directory service.

Also refer to the “Secure Shell” section below for information on using SFTP, instead. SFTP is a secure alternative to FTP, and offers SSO support via an AD or OD KDC.

E. File Services: WebDAV

With Apache on Mac OS X Server, you can provide file services via WebDAV. By binding to an AD server and joining the AD Kerberos realm, you can extend SSO support to this service as well.

To enable SSO access to your WebDAV service, you need to create an SSL-enabled web site on your server. Follow the steps in Section V to create a server certificate, then follow the steps below.

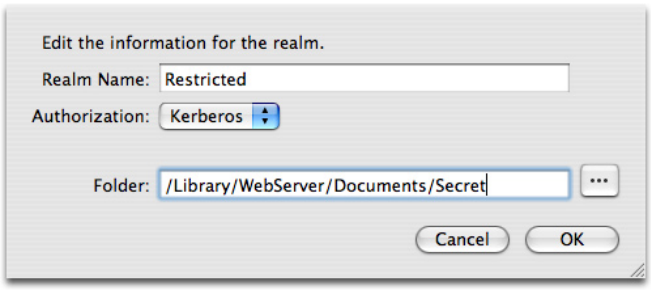
1. In Server Admin, navigate to Web > Settings > Sites.
2. Edit the default site, changing the following settings:
   General:
   - Domain Name: your server's FQDN (or the CNAME of your web site)
   - Port: 443

   Options:
   - Enable WebDAV
   - Disable the performance cache

   Security:
   - Enable SSL
   - Certificate: Choose the self-signed certificate you created in section V.
3. Save your changes, then start the Web service.
4. Log in to your client machine as an AD user and verify that you initiated your Kerberos tickets.
5. In the Finder, select "Connect to Server..." from the "Go" menu and enter the address of your server: "https://your.server.edu". You must use the fully qualified domain name. You will get a warning message about the certificate for the server being invalid (because it is a self-signed certificate). Click "Continue" and the share should mount without any additional authentication.
6. Again, examine your kerberos tickets. You should notice that you do *not* have any additional service tickets for your server. That is because you are currently accessing your site files with browse-only privileges.
7. At your server, create a new folder in /Library/WebServer/Documents named "Secret". Copy a document or create an empty file in this folder (so you have something whose appearance will indicate success).
8. In the Terminal, adjust the ownership of your new directory such that it can be managed by WebDAV:

```
sudo chown www /Library/WebServer/Documents/Secret
```

9. In Server Admin > Web > Settings > Sites > Your Site > Realms, create a new realm by clicking on the "+" button. Name the realm whatever you want, set Authorization to "Kerberos", and specify the full path to the folder. Click OK to complete the entry. Note: Basic authentication also works (albeit without SSO support), but Active Directory does not support Digest authentication.



10. Click the “Users & Groups” button and add an AD user to the Users table, giving it Browse and Author privileges. This does not yet work with AD groups. Save the settings.
11. Return to your client machine. If the WebDAV share is mounted, unmount it, then reconnect. Navigate to the “Secret” folder. The contents of the directory should appear and an HTTP service ticket should appear in the Kerberos application.
12. Destroy your Kerberos tickets, unmount the WebDAV share, then reconnect to the WebDAV share and confirm that you cannot view the contents of the Secret folder (without a Kerberos TGT).

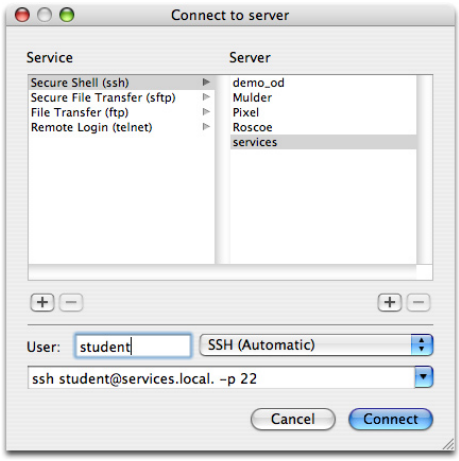
F. Secure Shell

For most servers, you will want to simply limit SSH access altogether, rather than enable SSO access. For the head nodes of clusters, however, there could be hundreds of directory-service based users that need shell access to the server. SSO shell access makes it easy and transparent for users to submit jobs from any Kerberos-enabled platform.

Like file services, SSH does not require any additional configuration to take advantage of single-signon support. Simply bind your server to the AD then join it to the Kerberos realm. SSH is enabled by default on Mac OS X Server. This service can be enabled or disabled in Server Admin > [server name] > Settings > General.

To verify that the SSH service is allowing SSO access, do the following at a client bound to AD.

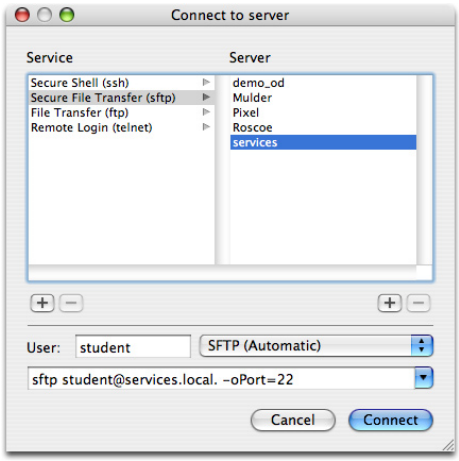
1. Log in as an AD user
2. Verify that you initiated your Kerberos tickets (review section IV,B)
3. In the Terminal application, choose “Connect to Server...” from the File menu.
4. In the Service column, select “Secure Shell (ssh)”
5. In the Server column, locate your Secure Shell server
6. Provide the shortname of the AD user you’re logged in as, then click the Connect button. This should connect you to the server without additional authentication, and a “host” service ticket should appear in the Kerberos application.



The Secure Shell server on Mac OS X Server also provides Secure FTP (sftp) by default. There is no additional configuration required to enable this service, nor is additional configuration required to provide SSO support.

To verify that the SFTP service is allowing SSO access, do the following at a client bound to AD.

1. Log in as an AD user
2. Verify that you initiated your Kerberos tickets (review section IV,B)
3. In the Terminal application, choose "Connect to Server..." from the File menu.
4. In the Service column, select "Secure File Transfer (sftp)"
5. In the Server column, locate your Secure Shell server
6. Provide the shortname of the AD user you're logged in as, verify that "SFTP (Automatic)" is selected as the connection protocol, then click the Connect button. This should connect you to the server with an sftp prompt without additional authentication, and a "host" service ticket should appear in the Kerberos application.



G. Mail services

Email is perhaps the most frequently used server-based service, it is certainly one of the most critical in many organizations. Unfortunately, it is also a service commonly vulnerable to password exploitation via packet sniffing. Securing client-server connections with SSL helps resolve this problem, enforcing strong password policies is also a good approach. By implementing single-sign-on support with a Kerberos KDC, however, you completely prevent the users' passwords from being transmitted across the network. Mac OS X Server provides mail services based on Postfix and Cyrus (SMTP, POP, and IMAP). Additionally, Mac OS X Server provides built-in support for SPAM and virus control.

User mail account information is typically stored in each user's directory service record. Mac OS X Server looks for the "apple-user-mailattribute" for a given user in the directory service to determine if the user has access to a mail account on the server. Because this is an Apple-specific attribute, typically you would need to extend the AD schema in order to leverage AD for authenticating Mac OS X's Mail service. However, if you implement a Service Access Control List at your Mail server, the SACL will override the MailAttribute. Simply add the users and groups you'd like to access your Mail server to the Mail SACL to give them access.

Follow these steps to configure the Mail service on Mac OS X Server to enable SSO support from your AD server.

1. In Server Admin > Mail > Settings, configure the mail service as desired. In particular, enable POP, IMAP, and SMTP.
2. In the Advanced tab, and further under the Security tab, enable Kerberos authentication for SMTP, IMAP, and POP. Save the settings and start the service.
3. In Server Admin > [Server name] > Settings > Access, uncheck the "Use same access for all services" box.
4. Click on the Mail service, then click the radio button to "Allow only users and groups below."
5. Click on the "+" button to reveal the users/groups drawer, then drag the individual users and groups that should have access to the service onto the column on the right ("Name"). Click save when finished.
6. At your client machine, log in as an AD user and launch the Mail application. Create a POP account configuration for your AD user. When given the opportunity to provide a password, leave the password field blank. When Mail attempts to check the configuration, it will automatically fall back on Kerberos authentication. For the Outgoing Server information, check the "Use Authentication" box and, again, provide a username but leave the password field blank.
7. Before dismissing the Account Summary, verify that you have a pop and an smtp service ticket in the Kerberos application.
8. In Mail's preferences > Accounts, create another account that uses IMAP. Verify that you now have an IMAP service ticket.

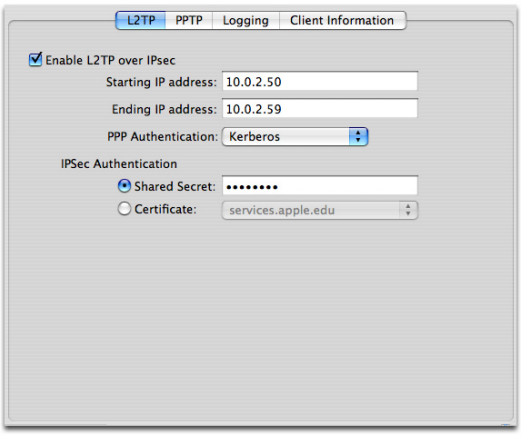
Tip: See the resources section at the end of this document for an article that describes how to create Mail Account bundles. Mail Account bundles make it easy for you to deploy custom account templates to your users.

H. VPN

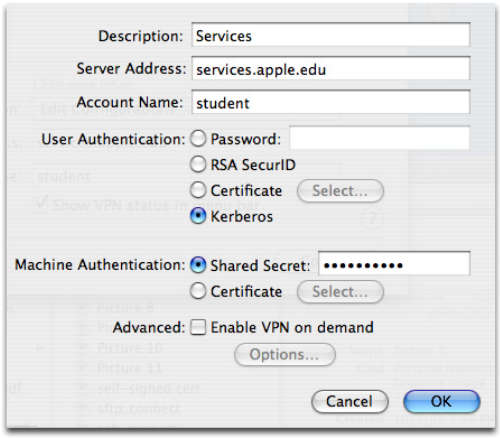
VPNs allow users at home or otherwise away from the LAN to securely connect to the LAN using any network connection, such as the Internet. From the user's perspective, the VPN connection appears as a dedicated private link. VPN technology also allows an organization to connect branch offices over the Internet, while maintaining secure communications. The VPN connection across the Internet acts as a wide area network (WAN) link between the sites.

Mac OS X Server provides VPN, DHCP, and DNS services -- everything you need to provide gateway services to your users. The VPN service on Mac OS X Server also supports Kerberos authentication (L2TP only), so you can extend SSO support to the Mac clients of this service as well. The following steps show an example configuration that supports SSO for the VPN service on a Mac OS X Server bound to AD and joined to the AD Kerberos realm. In a production environment, you would also need to configure and enable the DHCP service.

1. Configure the VPN service in Server Admin > VPN > Settings > L2TP. For example:
   - Enable L2TP over IPsec
   - Starting IP address: 10.0.2.50
   - Ending IP address: 10.0.2.59
   - PPP Authentication: Kerberos
   - IPsec Authentication: Shared Secret (make something up)



2. In the "Client Information" tab, use settings such as:
   DNS servers: 10.0.2.1
   Search domains: apple.edu
   Network Routing Definition:
   - Address: 10.0.2.0
   - Mask: 255.255.255.0
   - Type: Private
3. Save the settings and start the service
4. At the client, log in as an AD user and launch the Internet Connect application.
5. Click on the VPN icon in the toolbar and choose "L2TP over IPsec" when prompted.
6. From the Configuration popup menu, select "Edit configurations...". Create a new configuration with your server's information and the short name of your AD user (this field can be left blank):
   - User Authentication: Kerberos
   - Machine Authentication: Shared Secret



7. Click OK to save the changes, then click on the Connect button. The connection should be made without any additional user authentication. A "vpn" service ticket should appear in the Kerberos application.

Tip: You can export this configuration and provide it to your end users to make setup really simple. The configuration contains your shared secret, though, so handle the file accordingly.

I. Xgrid

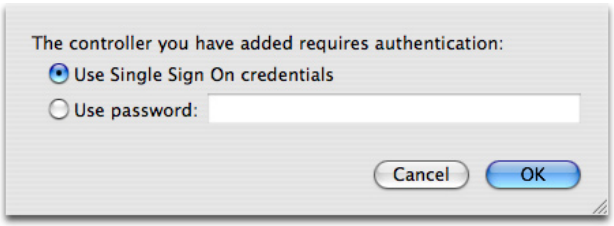
Xgrid, a technology in Mac OS X Server and Mac OS X, simplifies deployment and management of computational grids. Xgrid enables administrators to group computers into grids or clusters, and allows users to easily submit complex computations to groups of computers (local, remote, or both), as either an ad hoc grid or a centrally managed cluster.

Within a grid, authentication takes place several times: between the agent and controller when an agent joins a grid, between the client and controller when the client submits a job, and between the controller and agents when the controller splits jobs out to the agents. Kerberos authentication makes all of these authentication events completely secure and completely transparent to end users. With Mac OS X Server and an array of Mac OS X client machines, its easy to quickly create a powerful computational resource that leverages your existing technology investment and your AD infrastructure.

Note: This exercise requires that you have Xcode 2.1 or higher installed, with the Developer Examples.

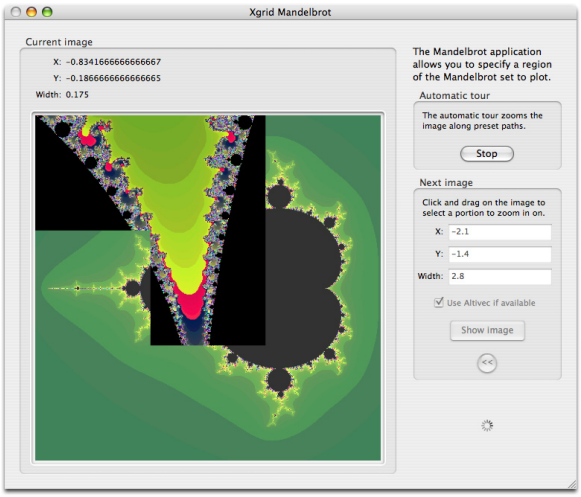
Follow these steps to configure the Xgrid service on Mac OS X Server to enable SSO support from your AD server.

1. Configure the Xgrid agent service on your server at Server Admin > Xgrid > Settings > Agent:
   - Enable agent service
   - Use first available controller
   - Agent accepts tasks always
   - Controller Authentication: Kerberos
2. Configure the Xgrid agent controller on your server at Server Admin > Xgrid > Settings > Controller:
   - Enable controller service
   - Client Authentication: Kerberos
   - Agent Authentication: Kerberos
3. Save the settings and start the service.
4. At a client machine bound to AD, login as an AD user.
5. In the Sharing Preference Pane, click on the Xgrid service, then click on the Configure button. Choose your server as the specific controller, accept tasks always, and select “Single Sign On” as the authentication method. Click OK.
6. Click the Start button to start the Xgrid agent on the client machine.
7. Open the Xgrid Admin application located in /Applications/Server. You should automatically be prompted to connect to your Xgrid Server as your client and server are on the same subnet. Otherwise, click on the Add Controller button and provide your server address. Click Connect, then choose the option to Use Single Sign On Credentials. Click OK.

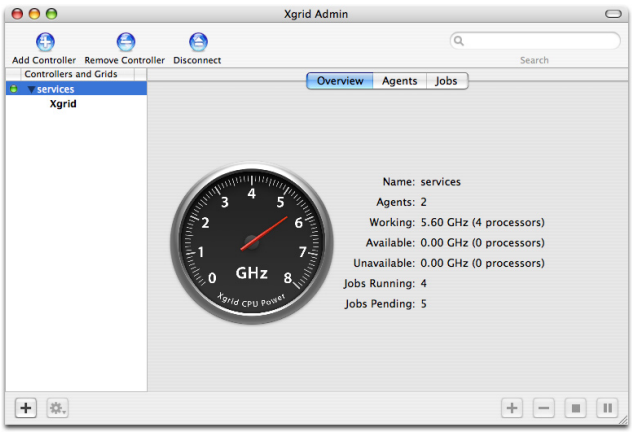


8. In the Overview tab, you will see a sum total of the processing power available to your grid. In the Agents tab, you should see your client and server, each idle. Minimize the Xgrid Admin application for now.

9. At the client machine, navigate to /Developer/Examples/Xgrid/GridMandelbrot and double click on the GridMandelbrot.xcode file to open the project in Xcode. Click on the "Build and Go" icon in the toolbar to build the project and launch the application.
10. You will automatically be prompted to connect to a controller. Choose your Xgrid controller, then choose to use Single Sign On credentials. The simulation will start automatically by submitting jobs to the controller. In turn, the controller submits jobs to each agent, collects the results, then returns them to the client application for display.



11. Return to the Xgrid Admin application to see the overall CPU usage, agent availability, and job status.



12. Examine your Kerberos tickets in the Kerberos application. You should have an "xgrid" service ticket for your Xgrid Controller. Authentication took place several times between your agent and the controller, between the controller and the agents, between your client and the controller, and also between the Xgrid Admin application and the controller. You should not have had to enter a password once, all of these authentications are handled by Kerberos through your AD KDC.

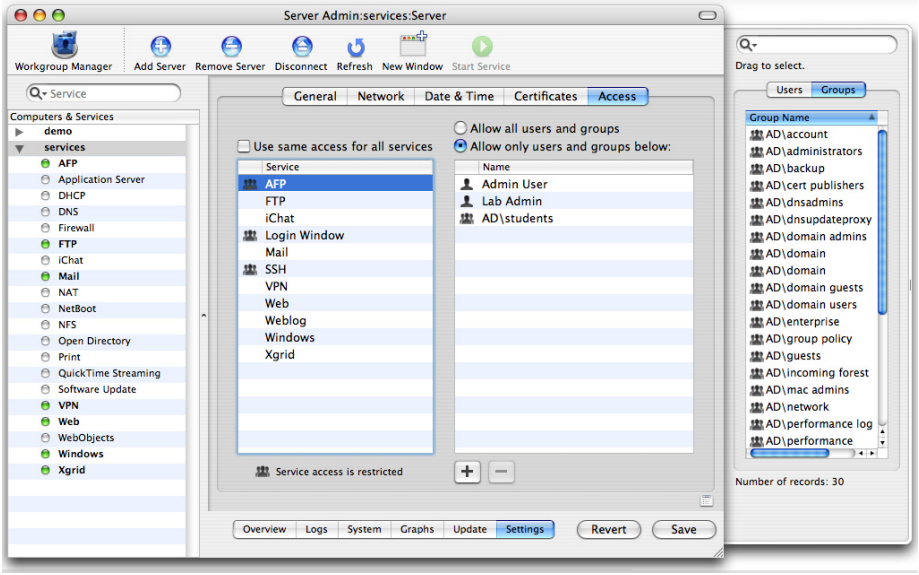
J. Restricting access to services

The one drawback to leveraging your campus-wide directory service is that once you bind your server to it, all of the services that your server provides are instantly accessible to anyone with a DS account. Obviously this is undesirable, and this is where service Access Control Lists, or SACLs, come in handy.

SACLs explicitly state which users and groups are allowed to access what services on your server. SACLs are configured in Server Admin, and consist of groups created in the local NetInfo directory. Each SACL group (named in the format "com.apple.access_service") contains local or DS users and groups that have been permitted to access the service. You can choose to implement the same limits to all services, or you can specify different access privileges for each service.

SACLs can also be implemented on Mac OS X Client to limit which users and groups are allowed to login to the computer. See the Resources section for more information on setting Mac OS X Client SACLs.

1. In Server Admin > [Server Name] > Settings > Access, uncheck the box to "Use same access for all services".
2. Click on the service for which you would like to apply restrictions.
3. Click the radio button to "Allow only users and groups below."
4. Click on the "+" button to reveal the users/groups drawer, then drag the individual users and groups that should have access to the service onto the column on the right ("Name").
5. When you are finished applying restrictions to each service, click on the save button.



6. At another Mac OS X machine, attempt to access a service on your server with a user account that is not permitted to access the service. Depending on the service, you will get an error message or you may simply fail to connect to the service.

X. Additional Resources

Hopefully by now you have gained some basic information on how to configure Mac OS X to access directory information from LDAP and Active Directory. Depending on how many computers you have to manage and how complex the infrastructure is, you may want to consider taking the three-day Apple Directory Services course.

Directory Services Integration and Administration v10.4 (Client/Server)
http://train.apple.com/course/D3459ZA

Additionally, there is a "Mac OS X Server Open Directory Administration" booklet available at http://www.apple.com/server/resources.

Resources online

http://www.apple.com/server/resources
http://macenterprise.org
http://www.afp548.com
http://www.4am-media.com
http://pgina.xpasystems.com

White papers

AD/OD integration: http://www.afp548.com/filemgmt/visit.php?lid=12
Example AD bind script: http://www.bombich.com/mactips/scripts.html#ad-bind
Novell eDirectory integration: http://macenterprise.org/content/view/80/77/
Leveraging eDirectory with Apple Workgroup Manager: http://www.novell.com/coolsolutions/tip/5955.html
Mac OS X Client SACLs: http://www.bombich.com/mactips/scripts.html#saclutil
Creating Mail Account Bundles:
http://developer.apple.com/documentation/MacOSXServer/Conceptual/XServer_ProgrammingGuide