

Leveraging Active Directory on Mac OS X

Leopard Edition, Revision 3

Contents

Overview	4
Configure a Mac OS X Server Open Directory Master	6
A. DNS setup	6
B. Configure the Open Directory Master	7
C. Set up a sharepoint for home directories	10
D. Create Open Directory users	11
Configure a client to authenticate to the OD Master	12
A. Add an Open Directory Master to the directory service search path	12
B. Verify directory connectivity and login as a network user	13
C. Troubleshooting Client --> Directory Service connectivity	14
Kerberos	16
A. Kerberos authentication process	16
B. Examine the use of Kerberos SSO for client authentication	17
Securing the LDAP service	19
A. Create a server security certificate	19
B. Implement SSL on the Open Directory service	19
Active Directory integration	21
A. "Feeling out" the Active Directory	21
B. Configure the AD plugin	22
C. Verify directory connectivity	25
D. Home directories and the AD plugin	27
E. Automating AD binding	28
F. AD plugin Troubleshooting	29
Group and Computer Management using OD + AD	31
A. Mac OS X Server configuration	32
B. Configure Directory Utility at the client	34
C. Configure Open Directory Groups	34
D. Configure Open Directory Computer Lists	35
E. Login at the client to test your dual-directory setup	36

Manage Groups and Computers: Workgroup Manager	37
A. Install the Server Administration tools	37
B. Explore Managed settings in Workgroup Manager (WGM)	37
C. Extending management beyond WGM: Preference manifests	40
Augmented Records	43
A. Configure augments in the ODM Configs container	43
B. Augmenting user records	46
C. Augmenting user records <i>en masse</i>	49
Leveraging AD for other Mac OS X Server services	51
A. Bind to AD, join Kerberos	51
B. File Services: AFP	51
C. File Services: SMB	53
D. File Services: FTP	53
E. File Services: WebDAV	54
F. Secure Shell	56
G. Mail services	58
H. VPN	59
I. Xgrid	61
J. Restricting access to services	64
Additional Resources	66
Resources online	66
White papers	66
Scripts and utilities	67
Appendix A: AD settings for your organization	68
Document History	69

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

Configure a Mac OS X Server Open Directory Master

2

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:

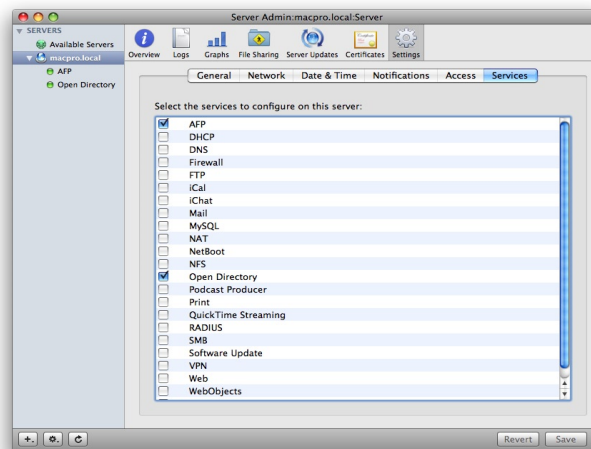
```
% scutil --get HostName
HostName: not set
% sudo scutil --set HostName "xserve.apple.edu"
```

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.

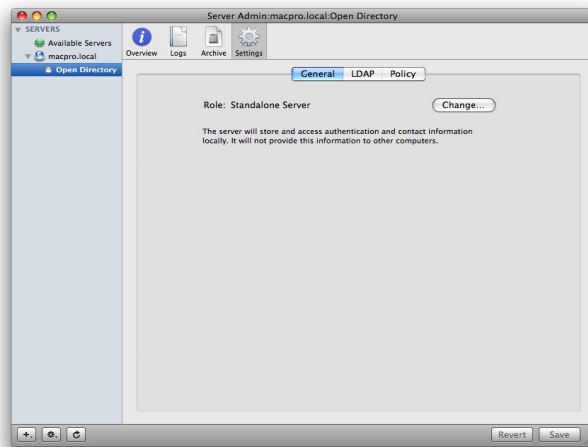
For the remainder of this chapter, use the hostname of your server where “hostname.apple.edu” is indicated.

B. Configure the Open Directory Master

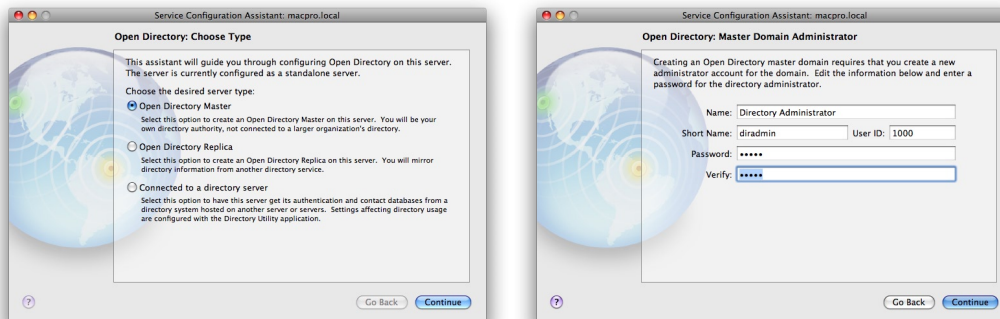
1. Open the Console application and click on the disclosure triangle next to “/Library/Logs”. Click on the “slapconfig.log” file to view its contents. On a fresh installation of Mac OS X Server, the last line in this file should contain a reference to setting standalone mode.
2. Open the Server Admin application and connect to your server (if necessary). Click on your server in the table on the left, then click on the “Services” tab to the far right. Check the box next to “Open Directory” to make that service available for configuration, then click the “Save” button in the lower right corner.



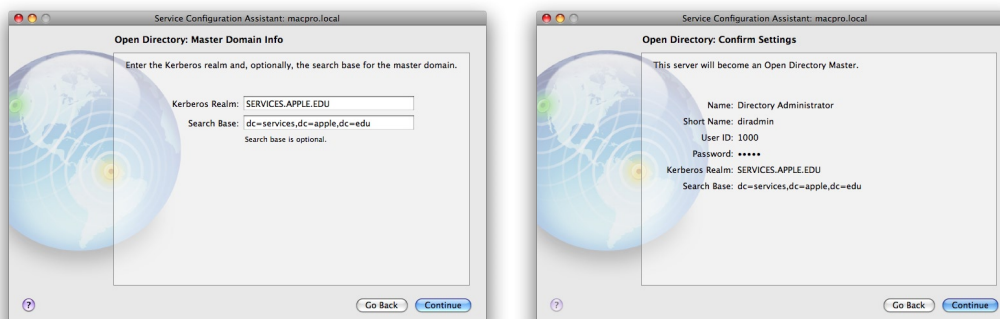
3. Click on your server's disclosure triangle in the table on the left and click on the Open Directory service to view its status. Click on the Settings button in the toolbar.



4. Click on the "Change" button to open the Service Configuration Assistant.



5. Choose to create an "Open Directory Master" and provide a username and password to administer the new domain.



6. In the third pane (Master Domain Info), you will be prompted for a Kerberos Realm name and search base. 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". Note that if you are not prompted for Master Domain Info, then it is likely that you have misconfigured DNS or you have an improper hostname.

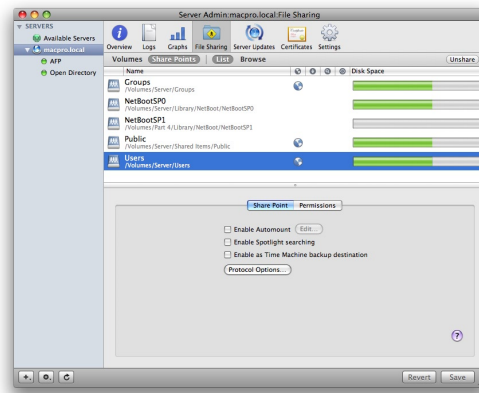
Be sure to use the information that is auto-populated in the form (not the info in the screenshot).

7. Click on the "Continue" button, then 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.
8. When the server has completed configuring itself, return to Server Admin and click on the "Overview" tab in the Open Directory service. Click the Refresh button at the bottom of the window and confirm that LDAP, Password Server and Kerberos are now running.

C. Set up a sharepoint for home directories

Open Directory users may access home directories via AFP, SMB, or NFS. For simplicity in this exercise, we will configure a sharepoint on the Mac OS X Server for Open Directory users. In a later exercise we will experiment with home directories on other servers.

1. In Server Admin, click on your server in the table on the left. Click on the “File Sharing” button in the toolbar, then click on the “Share Points” selection. Confirm that a sharepoint exists at /Volumes/<boot volume>/Users and that it is shared via AFP. If this sharepoint does not exist, create it. Refer to the Mac OS X Server “File Server Administration Guide” for details on creating the sharepoint.



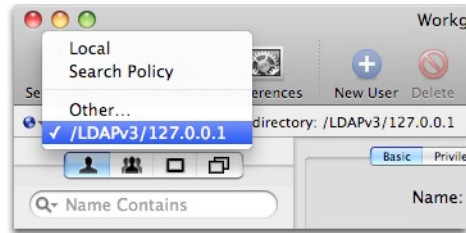
2. Select the “Users” sharepoint and click on the “Share Point” tab in the bottom frame. Check the box to “Enable Automount” -- a panel will appear with client automount options:



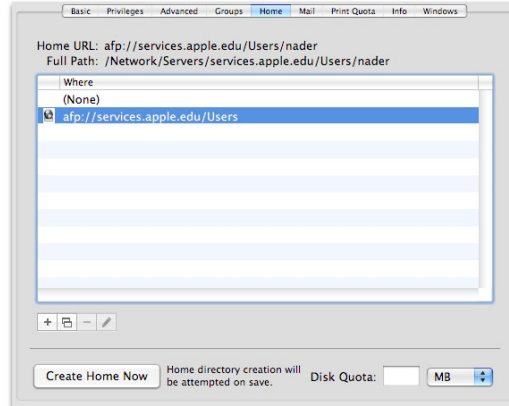
3. The defaults are fine so click OK. Authenticate using the “diradmin” account credentials when prompted. Click the “Save” button when finished editing sharepoints.
4. Start the AFP service.

D. Create Open Directory users

1. Launch Workgroup Manager and connect to your server. Authenticate using the directory administrator account you created when promoting the server to OD Master. Once you have authenticated, verify that you are viewing the contents of the LDAP-shared directory service. Click on the tiny blue globe beneath the Server Admin button in the toolbar and choose /LDAPv3/127.0.0.1 if it is not already selected.



2. 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.



3. Save the changes and close Workgroup Manager.

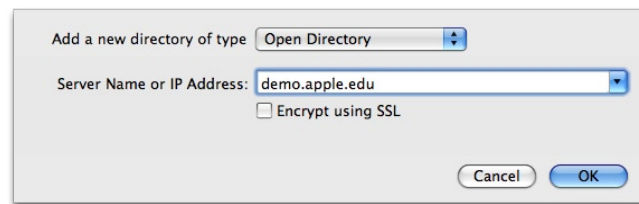
Configure a client to authenticate to the OD Master

3

Mac OS X client directory services are configured with the Directory Utility application located in the Utilities folder.

A. Add an Open Directory Master to the directory service search path

1. Launch Directory Utility and authenticate. Click on the “+” button to add a directory server to the local directory service configuration.

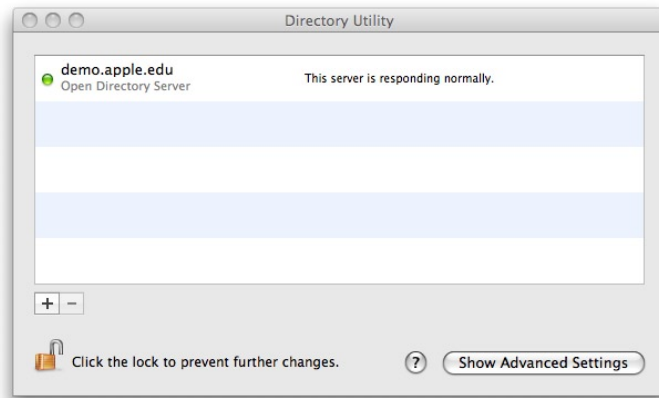


Use your server’s hostname. You can use the hostname or IP address, but hostname is recommended. Click the “OK” button.

2. Directory Utility will contact your server to acquire configuration information, then present to you the opportunity to do an “authenticated bind”. An authenticated bind is useful when you intend to manage the user experience by computer list because it 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 “OK”.



The system is now configured to authenticate against the Open Directory Master.



B. Verify directory connectivity and login as a network user

1. 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
demo.apple.edu
/LDAPv3 > cd demo.apple.edu/
/LDAPv3/demo.apple.edu > read Users/diradmin
apple-generateduid: A036593C-65E7-11D9-BB90-000A95C4219C
apple-mcxflags: <?xml version="1.0" encoding="UTF-8"?>
...
```

2. If you can read the contents of a user's record, then you have successfully configured the client to authenticate against the OD master.
3. Log out at the client machine then log in as one of the users that you created in Workgroup Manager.

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. View the DirectoryService man page to learn what DirectoryService error numbers mean.

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

4. Packet traces can be 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 |

5. 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.

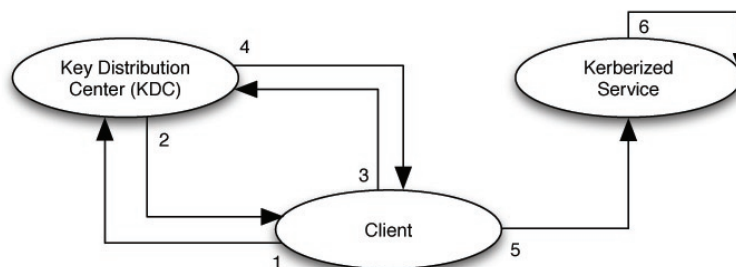
Kerberos is one of the most respected and secure methods of authentication available today. Not only does Kerberos enable single signon (SSO) authentication, it provides authentication without ever passing a password across the network. Kerberos is implemented by Microsoft in Active Directory and by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

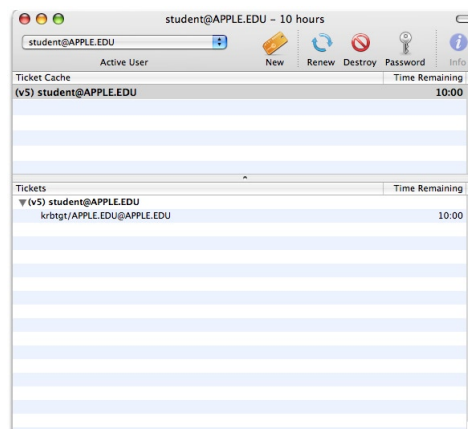
2. The KDC issues the client a ticket-granting-ticket (TGT), the credential needed when the client wants to use the Kerberized service. The TGT is good for a configurable amount of time, but can be revoked before expiration. It is cached on the client until it expires.
3. The client contacts the KDC 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.

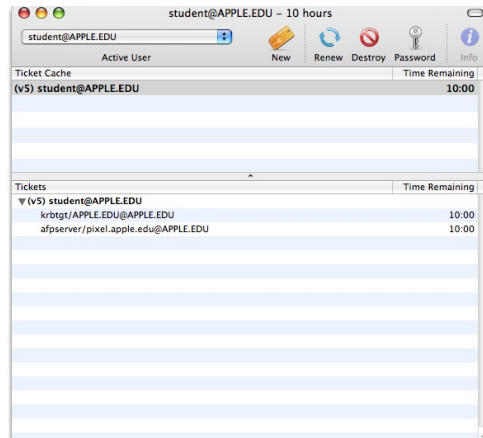
B. Examine the use of Kerberos SSO for client authentication

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 page 102 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 bind to an Open Directory Master in Directory Utility. Read the “kerberosautoconfig” man page for more information on how this file is generated.

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 use a self-signed certificate.

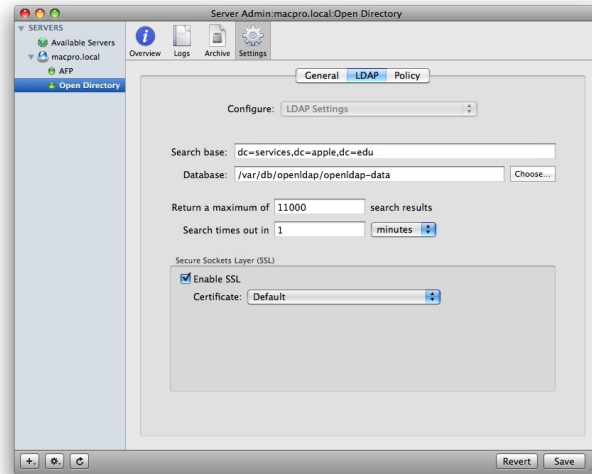
Mac OS X Server is pre-configured with a default self-signed security certificate. To view this certificate, click on your server in Server Admin, then click on the "Certificates" button in the toolbar. Click on the "Default" entry in the table to view the characteristics of the default security certificate.

For more information on implementing security certificates on Mac OS X Server, consult Chapter 4 of the Server Administration manual.

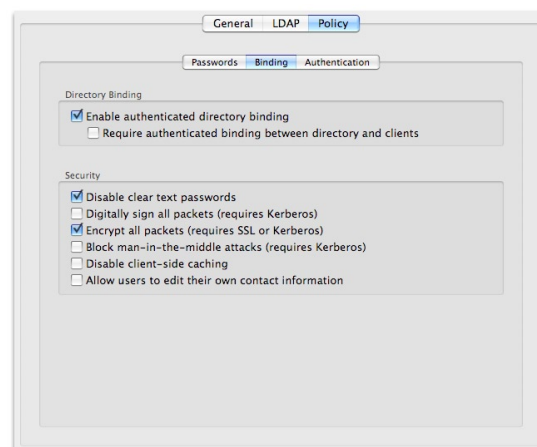
B. Implement SSL on the Open Directory service

Perform the following steps on your Open Directory Master to secure the LDAP service.

1. In Server Admin > [your server] > Open Directory > Settings > LDAP, check the box to "Enable SSL". Choose your "Default" 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 Utility.

4. Unbind your client from Open Directory and demote your Open Directory Master to Standalone before proceeding.

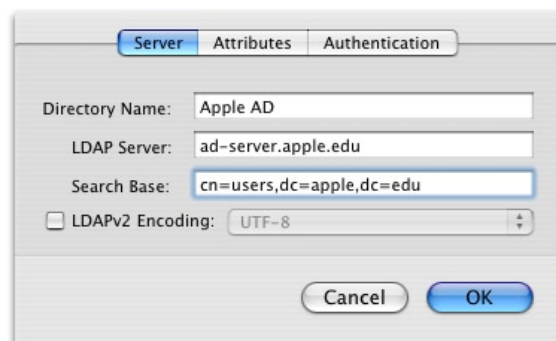
Active Directory is essentially a customized LDAP directory service, and Mac OS X can use AD for authentication using either the LDAPv3 plugin or the Active Directory plugin. Unlike an Open Directory Master, Mac OS X clients do not have prior knowledge of the schema implemented by each and every Active 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.

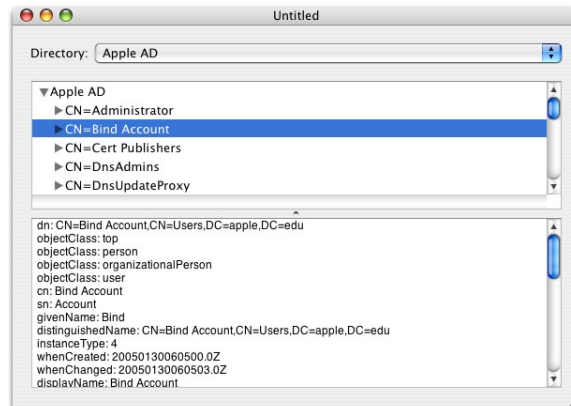
1. Launch the LDapper application (you can find this application using <http://versiontracker.com>).
2. Select “Preferences” from the LDapper menu and click on the “+” icon to add a new directory service. Configure the Directory with settings appropriate for your AD environment.



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”. You may have to use the distinguished name of an AD user account, such as:

`cn=Bind Account,cn=Users,dc=apple,dc=edu`

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, but is also useful when trying to determine the distinguished name of the container you will use for computer account binding.

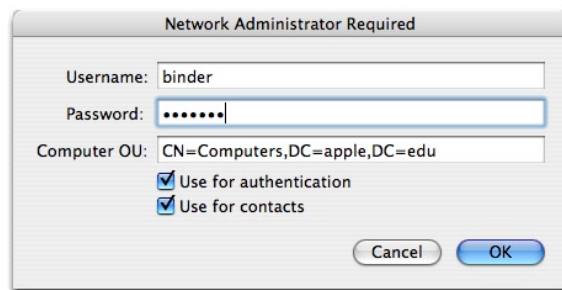
6. Edit the directory configuration in LDapper’s preferences to use your Computer OU as the search base instead. Open a new browse window and browse through the computer records.

B. Configure the AD plugin

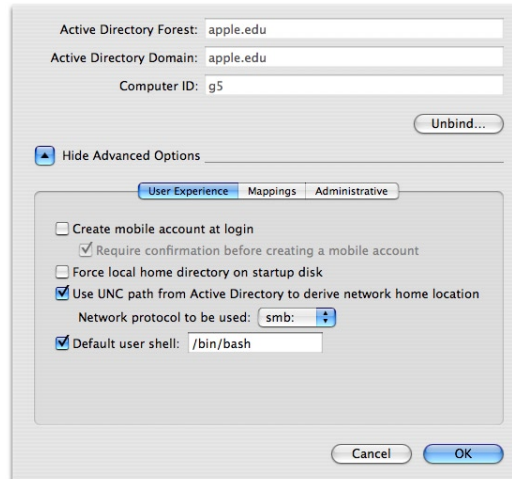
Because the AD plugin uses DNS to locate AD resources on the network, there is very little configuration necessary for the AD plugin. Use the settings provided by your AD administrator or those provided on the last page of this document.

1. Launch the Directory Utility application.

2. To expose more configuration options of the AD plugin, we will configure it using the advanced settings. Click on the "Show Advanced Settings" button., then click on the "Services" button in the toolbar.
3. Authenticate if necessary. Enable the Active Directory service then click on the pencil icon at the bottom of the window.
4. 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. If you are configuring a dual-boot Mac, remember that the computer ID must be unique between Mac OS X and Windows.

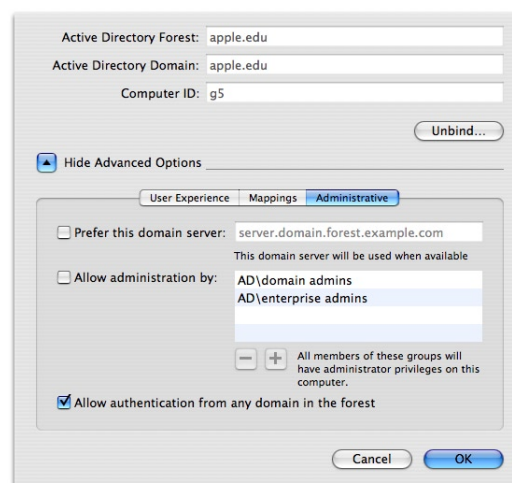


5. Click on the "Show Advanced Options" button. Consider the options in the User Experience tab:
 - "Create mobile account": causes the client to cache the account credentials of the last user to use the machine. This can be handy if your users take their machines home.
 - "Force local home": This should 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.

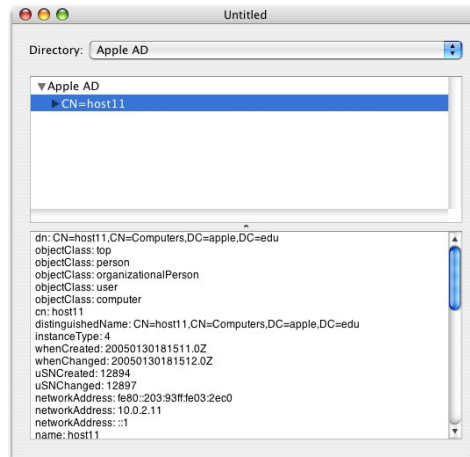


6. Consider the options in the Administrative tab:

- “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.



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



8. Return to Directory Utility and click on the "OK" button to close the AD plugin window.
9. Click on the "Directory Servers" button in the toolbar. If your machine is configured with an Open Directory server from a previous exercise, remove that server from the list, click apply, then close Directory Utility.

C. Verify directory connectivity

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 > 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 this behavior of the AD plug-in. 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. Many applications create cache files on launch. If a lab full of students is instructed to log in and launch several applications that behave this way, your file server and network 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.

Portable Home Directories (PHD) are a hybrid between network and local home directories. A PHD will create a copy of the network home directory locally and sync changes upon login and logout. When user accounts are stored in Active Directory, PHDs can be set up via user management if the AD schema is extended, or by group and computer management in a “Golden Triangle” configuration. Refer to chapter 7 for general information about group and computer management. Refer to Chapter 8 of the “User Management” guide at <http://www.apple.com/server/macosx/resources/> for more information on configuring and managing mobile accounts on Mac OS X.

Another consideration of home directories in regards to the AD plugin is that the attribute is stored within a user’s Active Directory user record, and typically not modifiable by individual departments. On large University campuses, this often makes it difficult for system administrators to provide department-specific storage for their users. To learn one method of working around this limitation, refer to the “Augmented Records” section in Chapter 9.

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 'APPLE\Lab Administrators'
```

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 synchronized 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.

Group and Computer Management using OD + AD



It is fairly common for Active Directory administrators to be unwilling 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 establish cross-domain authorization between an Open Directory Master and Active Directory to facilitate the management of your Macs by group and computer lists.

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: the local directory store, then any configured network directories. For example, when you log in, the system first looks to the local 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 as a subordinate to Active Directory and how to configure a Mac OS X client to use a dual-directory infrastructure.

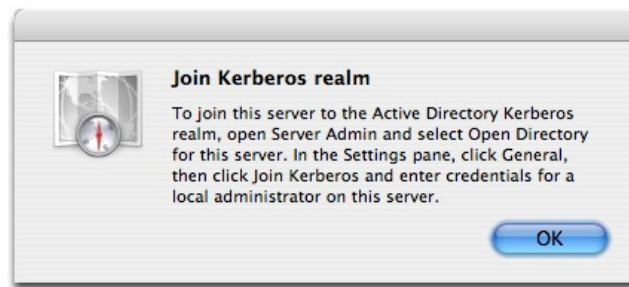
* Leopard Server introduced augmented records, a feature that allows the Mac OS X Server administrator to import user account records from AD into OD and "augment" them with non-AD-native attributes such as MCX settings. If a Mac OS X client was bound (only) to a Mac OS X Leopard Server with augmented AD records, then the client would essentially be drawing information about a single object from two separate directory services. Using augmented records to override AD-specified home directories is covered in Chapter 8. See the Reference section at the end of this document for other references to augmented records documentation.

A. Mac OS X Server configuration

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.

The order of the steps below is very important. To become a subordinate for a directory system you must first bind to Active Directory. Then, using Server Admin, you will promote your Open Directory server to an Open Directory master. The subordinate server automatically determines that it is subordinate to an Active Directory server and configures itself accordingly. See the "Mixing Active Directory and Open Directory Master and Replica Services" section of the Open Directory Administration manual for more information on this subject.

1. Bind your Mac OS X Server to Active Directory using instructions from chapter 6. When you perform the bind on Mac OS X Server, you will get an additional dialog:



2. Ignore the instructions in the dialog, as the appearance of the “Join Kerberos” button in Server Admin can be inconsistent. Run this command in the Terminal to configure the kerberized services on Mac OS X Server with the AD service principals:

```
sudo dsconfigad -enableSSO
```

3. 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).

4. Also verify that your services are configured to use the AD Kerberos realm, not your own. The AFP service is easy to check, run the following command in the Terminal:

```
defaults read /Library/Preferences/com.apple.AppleFileServer  
kerberosPrincipal
```

You should see your AD server’s realm listed. If you do not, unbind from the AD domain, then repeat the steps above.

5. Promote your server to an Open Directory Master using the steps outlined in chapter 2.
6. If user home directories are hosted on another Mac OS X Server machine, bind that machine to Active Directory at this time, and join it to the AD Kerberos realm. In the Advanced options section of the AD plugin, verify that the server

is configured to **not** force local home directories and that it **is** configured to use the UNC path to derive the network home location.

7. Pre-create network home directories with the “createhomedir” utility:

```
sudo createhomedir -s
```

B. Configure Directory Utility at the client

The client-side configuration is simply a combination of binding to both the Open Directory Master and Active Directory. Note that order is important. Bind to Open Directory first, or verify that the Open Directory node is listed first in the Search Policies tab of Directory Utility. If the Open Directory node is not listed first, managed settings may not be applied, and augmented records will be ignored.

1. Bind the client (anonymously) to your OD Master (Review chapter 3).
2. Bind the client to Active Directory using the AD Plugin (Review chapter 6).

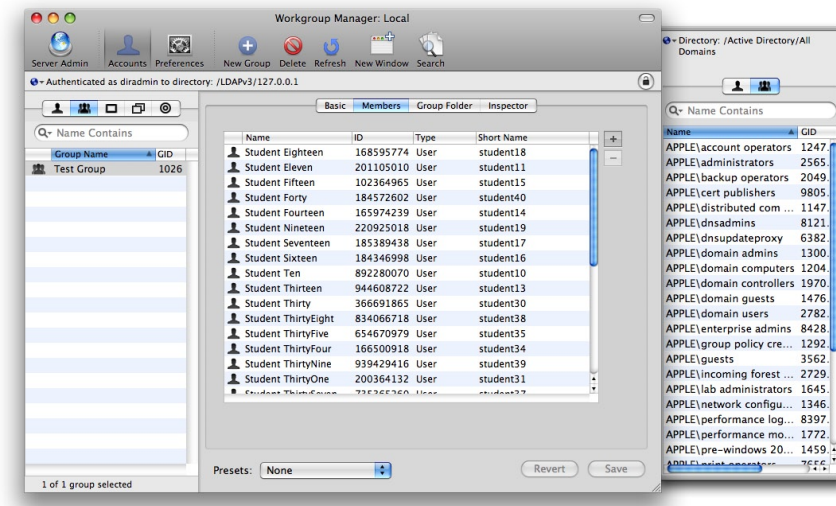
C. Configure Open Directory Groups

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 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. Launch Workgroup Manager and connect to your Open Directory Master authenticating as the diradmin user. Confirm that you are working in the LDAPv3 node.
2. Click on the Groups tab and create a new group.
3. Click on the “Members” tab, then click on the “+” button 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. Save the changes to the group.



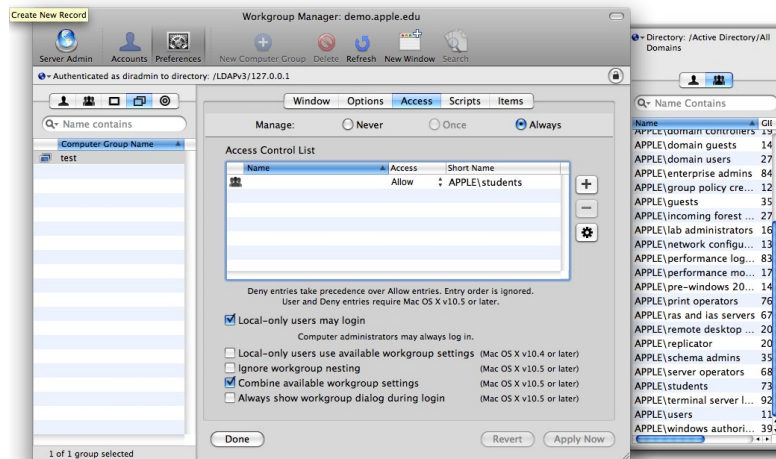
4. Management of preferences could fill another book and will not be covered in detail here, however you should modify at least one preference to verify that management by group is working. Be sure your group is selected, then click on the Preferences button in the toolbar. Click on the "Dock" button, 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 Computer Groups tab (the two-squares icon).
2. Create a new computer list. Name it "Test Lab", click on the "Members" tab, 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 Preferences button in the toolbar. A common administrative task is to limit what users and groups are allowed to use a specific group of computers. You can impose this managed preference on a computer list. Click on the "Login" button, then click on the "Access" tab.
4. Click on the "Always" radio button, then click on the "+" button to add users/groups to the access control list. After you've added a few user or group records, click the "Apply Now" button to save the changes.



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 Utility). 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 chapter 4(B) about Kerberos, then examine your Kerberos tickets while logged in with your AD account.
5. Enable the AFP and SMB 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 Chapter 10.

Manage Groups and Computers: Workgroup Manager

8

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. This chapter serves only to give you an idea of what management features are available. John DeTroye has published a very thorough and extensive document on this topic, covering:

- Defining client management for Mac OS X
- Setting up the server to provide management
- Setting up the client for network based management
- Basic Managed Client information
- Details in MCX that enhance management
- Explaining user accounts, mobility, and PHDs
- Thoughts on additional ways to promote workflow management

The document can be downloaded from his web site at <http://homepage.mac.com/johnd> (full link is in the Resources section at the end of this document).

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.

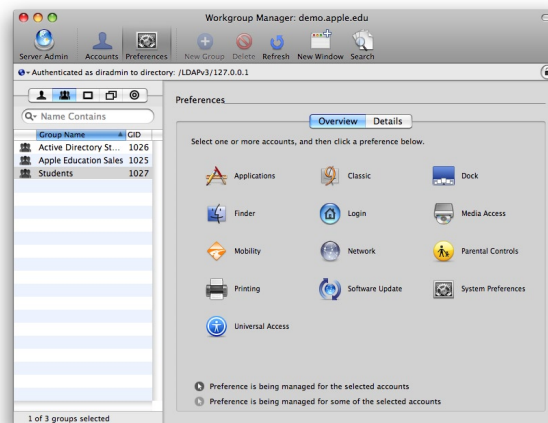
1. Insert the Server Administration Tools CD (from the Mac OS X Server Installation disc set) into your client machine.
2. Double-click on the Server Administration Software package to install the Server Admin tools. The tools will be installed into /Applications/Server.

B. Explore Managed settings in Workgroup Manager (WGM)

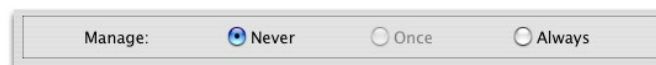
A directory service is really nothing more than a big database filled with information about users, groups, computers, and other network services. Group and computer management settings are also stored in this database. LDAP, or Lightweight Directory Access Protocol, is the protocol used to communicate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

Management 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 be 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 chapter.
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 restrict which applications are

allowed to launch either explicitly (e.g. add specific applications to the list), or you can indicate that applications in certain folders are allowed or denied.

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

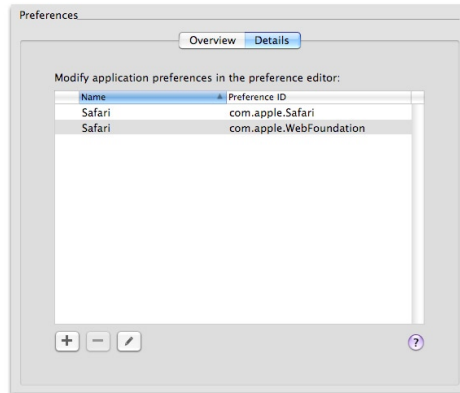
As you've seen, Workgroup Manager gives you a great amount of control over the end-user experience. You'll notice, however, that WGM does not list specific application preferences. What if you want to have much more granular control over how users use specific applications? That's where preference manifests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

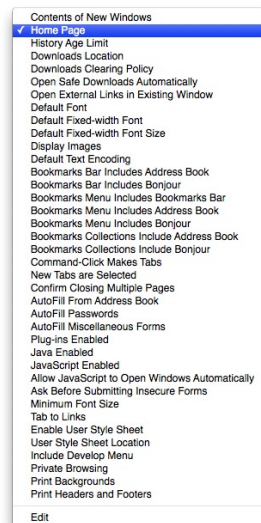
Instead of simply choosing an application, you should import a preferences file that you've manually stripped to contain only the preference settings you want to manage.

For an example manifest file, we'll take a look at Safari's preference manifest.

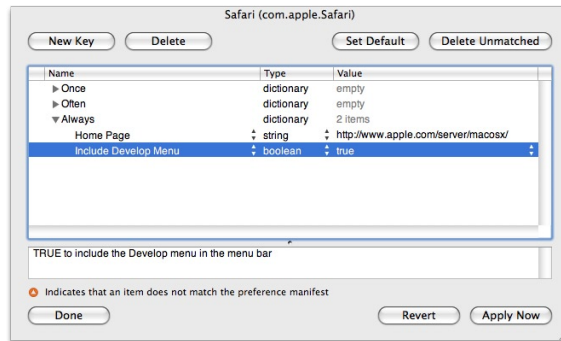
1. Launch the Workgroup Manager application in /Applications/Server and connect to your server, authenticating as a directory administrator.
2. Click on the Groups tab and select a group (created in a previous chapter).
3. Click on the "Preferences" button in the toolbar, then click on the "Details" tab.
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 my preferences for this application" 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 then click on its disclosure triangle to enable the "New Key" 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 "Home Page". The type should be automatically set for you, and the value will be set to the default. Change the default value to your favorite home page.



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.
12. Open the System Profiler application (/Applications/Utilities) and click on the "Managed Client" item in the "Software" section on the left. On the right you will see a list of all managed settings, their values, and where they're coming from. This report is very helpful in tracking down conflicts.

Augmented records were introduced in Open Directory on Leopard Server. As the name implies, augmented records are records that augment content imported from another record. Augmented records allow you to implement attributes in a user record that are not defined within the primary directory service. This is very useful if you do not have the authority to extend the schema of the primary directory service, and you desire to manage objects defined in the primary directory service.

To implement augmented records, you would bind your Open Directory Master to your primary directory service, import objects from the primary directory service, and augment the imported records with Mac OS X-specific attributes, such as `MCXSettings`, `IMHandle`, and `WeblogURI`. Your Mac OS X clients would then be bound to both the Open Directory Master and the other directory service. When Mac OS X performs directory lookups, it will first find augmented records and overlay them onto genuine records from the primary directory service.

In most cases, attributes in augmented records are only honored by the client if the attribute is not defined within the primary directory service. `givenName`, `gidNumber`, and `UniqueID` are examples of common attributes that you would not be able to override with augmented records. `homeDirectory` is also usually defined within most directory services, but it is a special case with the Active Directory plugin (refer to Chapter 6, section C, page 26). The Active Directory plugin generates some Mac OS X-specific attributes dynamically based on other information in the user record. For example, the “`NFSHomeDirectory`” attribute value is derived from the `homeDirectory` attribute in Active Directory. The `homeDirectory` attribute value is also reformatted from the value contained within the AD record, but only when the “Use UNC path from Active Directory to derive network home location” is enabled.

Given this behavior of the AD plugin, it is possible to override the `homeDirectory` attribute provided by AD by implementing augmented records and by configuring the AD plugin to *not* use the UNC path from Active Directory to derive the network home location. The following instructions will describe how to implement augmented records, using the `homeDirectory` attribute as an example augmented attribute.

A. Configure augments in the ODM Configs container

The configuration for augmented records is stored in the “`augmentconfiguration`” record in the Config container in Open Directory. This contains information about the Augment Node, (i.e. the node storing the augment info), the Augmented Node (i.e. the “foreign node”) and which attributes can be augmented for different

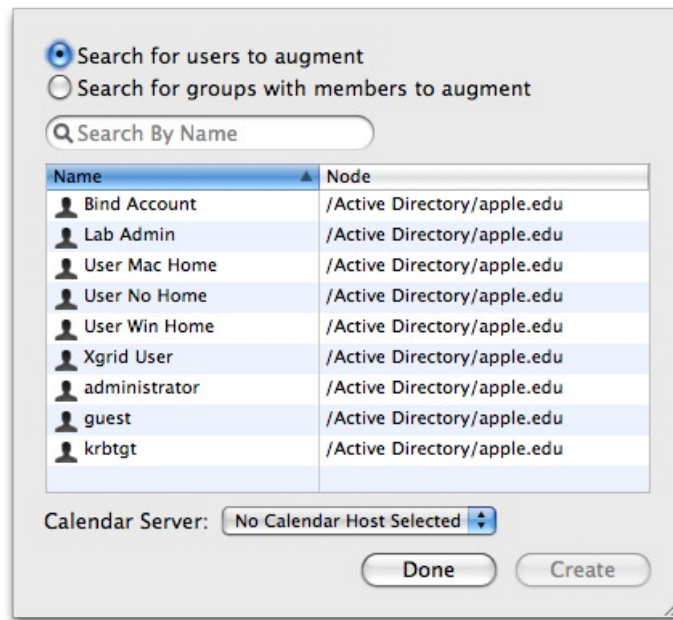
record types.

A typical augment record configuration looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://
  www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>Augment Attribute List</key>
  <dict>
    <key>dsRecTypeStandard:Users</key>
    <array>
      <string>dsAttrTypeStandard:ServicesLocator</string>
    </array>
  </dict>
  <key>Augment Directory Node Name</key>
  <string>/LDAPv3/master.apple.edu</string>
  <key>Augment Search</key>
  <true/>
  <key>Augmented Directory Node Name</key>
  <string>/Active Directory/All Domains</string>
</dict>
</plist>
```

This configuration indicates that augmented records on this Open Directory Master may define the ServicesLocator attribute. The “augmentconfig” config record is created automatically when you import objects from another directory service.

1. Launch Workgroup Manager and connect to your Open Directory Master using the diradmin account.
2. In Workgroup Manager preferences, check the box to ‘Show “All Records” tab and inspector.’ Click OK.
3. Click on the Users tab, then choose “New Augmented User Records” from the Server menu in the menu bar. A panel will appear allowing you to search for user records to import. You can either search for specific users, or search for groups within Active Directory to import all users belonging to a specific group.



4. Select at least one record to import, then click on the “Create” button, then click the “Done” button.
5. Click on the Inspector tab (the tab on the right that has the bullseye icon), then choose “Config” from the popup menu of containers.
6. The Config container should have a record named “augmentconfiguration.” Click on that group, then click on the “XMLPlist” attribute in the inspector on the right. Click the “Edit” button to view the contents of that record, it should be similar to the augment configuration record printed on the previous page.
7. For this exercise, we will add three attributes to the “Augment Attribute List” of the augmentconfiguration record (below the ServicesLocator entry):


```

      <string>dsAttrTypeStandard:HomeDirectory</string>
      <string>dsAttrTypeStandard:NFSHomeDirectory</string>
      <string>dsAttrTypeStandard:Keywords</string>
      
```
8. Save the changes.

These steps only need to be performed once (e.g. not every time you want to import or augment user records).

B. Augmenting user records

1. Click on the Users tab and select one of the augment records. Augment records have a “blue gel” badge.

You’ll notice that, on the right, there are now only three tabs in the record view: Basic, Advanced, and Inspector. The contents of the Basic tab are not editable, and the contents of the Advanced tab are only relevant to per-user iCal Server settings. The Inspector tab allows us to add whatever attributes we want, so that’s what we will use to add the home directory attributes.

Note: Do not confuse the Native and Standard attributes. For example, “homeDirectory” is a Native attribute, whereas “HomeDirectory” is a standard attribute. They each map to different attributes of the other type:

```
dsAttrTypeStandard:NFSHomeDirectory <--> dsAttrTypeNative:homeDirectory
dsAttrTypeStandard:HomeDirectory <--> dsAttrTypeNative:apple-user-homeurl
```

In general, it is best to hide either the Native or the Standard attributes because they map to each other, potentially causing some confusion if you’re not familiar with the mappings. To hide the Standard attributes, click on the “Options...” button below the inspector pane and uncheck the box to “Show Standard attributes.”

2. Click on the homeDirectory attribute, then click on the “Edit” button. In the edit panel, type the *local* path to the user’s home directory. For example:

```
/Network/Servers/yourserver.domain.com/Users/username
```

Remember that if the volume containing the home directories is not the boot volume, you need to include “/Volumes” in the path, e.g.:

```
/Network/Servers/yourserver.domain.com/Volumes/Homes/username
```

3. If an attribute named “apple-user-homeurl” does not exist in the augment record, click on the “New attribute...” button. Set the name of the attribute to “apple-user-homeurl”. The content of apple-user-homeurl should be of the format:

```
<home_dir><url>afp://yourserver.domain.com/Users</url><path>username</path></home_dir>
```

4. Click the Save button to save changes.

Repeat these steps for each user record you would like to augment. If you have

more than a handful of user records to augment, you will want to familiarize yourself with the `dscl` utility so that you can automate this process.

Complete the following steps on the server hosting home directories *and* client machines:

5. Verify that the machine is bound first to Open Directory, then to Active Directory, and verify that you have disabled the “Use UNC path from Active Directory to derive network home location” option in the AD plugin configuration in Directory Utility.

6. Restart the `DirectoryService` daemon to force the system to recognize that augment records are now available in the Open Directory node:

```
sudo killall DirectoryService
```

7. The process of record augmentation occurs at the client (and home directory server) as a function of the `/Search` node. Run the following commands to see how information from the Open Directory augment records is collated with information from the original Active Directory records:

```
host10:~ apple$ dscl localhost
> read /LDAPv3/master.apple.edu/Augments/Users:labadmin/
AppleMetaNodeLocation: /LDAPv3/master.apple.edu
GeneratedUID: 0D3870FD-BFEF-4E8E-8216-5BBFBB3AB5A1
HomeDirectory: <home_dir><url>afp://services.apple.edu/
Users</url><path>labadmin</path></home_dir>
NFSHomeDirectory: /Network/Servers/services.apple.edu/Users/
labadmin
PasswordPlus: *****
PrimaryGroupID: 20
RealName: Lab Admin
RecordName: Users:labadmin
RecordType: dsRecTypeStandard:Augments
ServicesLocator: (null):(null):calendar
UniqueID: 221802749
UserShell: /usr/bin/false

> read /Active\ Directory\ All\ Domains/Users/labadmin/
AppleMetaNodeLocation:
/Active Directory/apple.edu
AuthenticationAuthority: 1.0;Kerberosv5;FD3AA261-AE21-4A60-
9D86-9DA5B7234850;labadmin@APPLE.EDU;APPLE.EDU;
FirstName: Lab
GeneratedUID: FD3AA261-AE21-4A60-9D86-9DA5B7234850
LastName: Admin
Password: *****
```

```

PasswordPlus: *****
PrimaryGroupID: 278245951
RealName: Lab Admin
RecordName: labadmin
    lab admin
    labadmin@apple.edu
    APPLE\labadmin
    APPLE\lab admin
    Lab Admin
RecordType: dsRecTypeStandard:Users
SMBAccountFlags: 805306368
SMBGroupRID: 513
SMBHome: \\master\Users\labadmin
SMBHomeDrive: H:
SMBLogoffTime: 0
SMBLogonTime: 128614125322148848
SMBPasswordLastSet: 128566355521191920
SMBPrimaryGroupSID: S-1-5-21-2554006548-4090241852-
    3873005326-513
SMBSID: S-1-5-21-2554006548-4090241852-3873005326-1201
UniqueID: 221802749
UserShell: /bin/bash

    > read /Search/Users/labadmin/
dsAttrTypeNative:distinguishedName:
    CN=Lab Admin,CN=Users,DC=apple,DC=edu
...
FirstName: Lab
GeneratedUID: 0D3870FD-BFEF-4E8E-8216-5BBFBB3AB5A1
HomeDirectory: <home_dir><url>afp://services.apple.edu/
    Users</url><path>labadmin</path></home_dir>
LastName: Admin
NFSHomeDirectory: /Network/Servers/services.apple.edu/Users/
    labadmin
Password: *****
PasswordPlus: *****
PrimaryGroupID: 278245951
RecordType: dsRecTypeStandard:Users
ServicesLocator: (null):(null):calendar
SMBAccountFlags: 805306368
SMBGroupRID: 513
SMBLogoffTime: 0
SMBLogonTime: 128582793083305216
SMBPasswordLastSet: 128582365602518032
SMBPrimaryGroupSID: S-1-5-21-2554006548-4090241852-
    3873005326-513
SMBSID: S-1-5-21-2554006548-4090241852-3873005326-1136
UniqueID: 221802749
UserShell: /bin/bash

```

C. Augmenting user records *en masse*

Managing augmented records for a handful of users is practical, but managing augments for several hundred users would be nearly impossible, especially if you were augmenting home directory attributes. The following code demonstrates the dscl commands that will create an augment record and populate it with custom home directory attributes. This script also tags the augment record with a keyword, allowing you to perform keyword searches in WGM based on these keywords.

```
host10:~ apple$ dscl /LDAPv3/master.apple.edu
> auth diradmin apple
> create /Augments/Users:labadmin RealName "Lab Admin"
> create /Augments/Users:labadmin GeneratedUID E426858E-DA85-
4085-BB28-4DEE094357A1
> create /Augments/Users:labadmin HomeDirectory
<home_dir><url>afp://services.apple.edu/Users</
url><path>labadmin</path></home_dir>
> create /Augments/Users:labadmin NFSHomeDirectory /Network/
Servers/services.apple.edu/Users/labadmin
> create /Augments/Users:labadmin UniqueID 221802749
> create /Augments/Users:labadmin PrimaryGroupID 20
> create /Augments/Users:labadmin Keywords "ad_group"
```

After creating the augmented records, run the following command on your file server (e.g. services.apple.edu in this example) to create home directories for all the imported users:

```
services:~ admin$ sudo createhomedir -s
```

Refer to the resources section of this document for a complete script that loops through all members of a group to add augment records to an Open Directory master.

There are a few caveats to be aware of when using augmented records as described above:

- If a network home directory is not defined for a user in AD, the HomeDirectory attribute will be overridden, however the NFSHomeDirectory attribute will not. As a result, the network home sharepoint will be mounted at /Users rather than at /Network/Server/server.apple.edu/sharepoint. This will make the local /Users directory unavailable as long as the network user is logged in. When the network user logs out, the /Users directory will be available for local users again.

- Augmented user records cannot be added to Open Directory groups, therefore they cannot be managed like other users.
- You can augment records with the apple-keywords attribute, but any keywords added to a record won't show up in WGM. They will be searchable in WGM, however they won't be searchable, for example, with "dscl /Search search".

Leveraging AD for other Mac OS X Server services | 10

In the previous chapters 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 several 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. 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 chapter 7 (A, steps 1-4) at your 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?

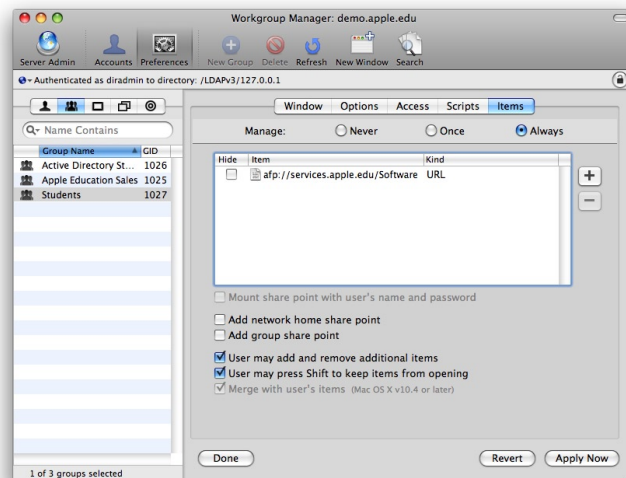
The AppleFileServer does not require any additional configuration to take advantage of AD authentication and single-sign-on. Simply bind your AFP server to the AD and join it to the Kerberos realm.

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 chapter 4, section 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
```

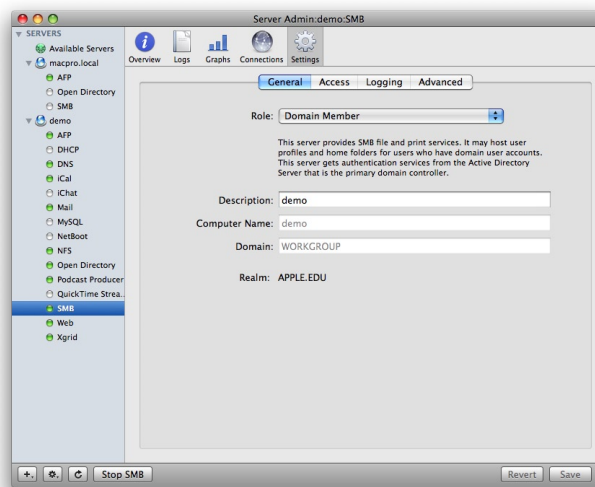
4. Make the script run at login (all one line):

```
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 > SMB. 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.

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 Chapter 8, section 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.

1. In Server Admin, navigate to Web > Sites and select the default web site entry.

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

Security:

- Enable SSL
- Certificate: Choose the default self-signed certificate

3. Save your changes, then start the Web service.

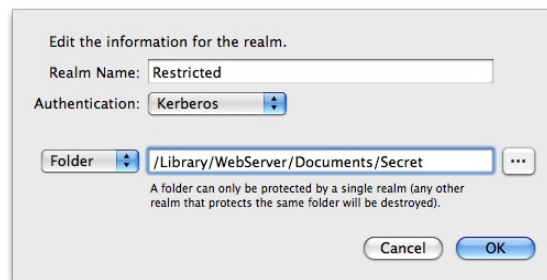
4. Log in to your client machine as an AD user and verify that you initiated your Kerberos tickets.

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 > 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 or group to the Users table, giving it Browse and Read/Write privileges. 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.

By default, Kerberos authentication is disabled for the SSH client on Mac OS X because it imposes (albeit short) login delays for users without a Kerberos TGT. To enable Kerberos authentication for the SSH client on your *client* system(s):

1. Edit the /etc/ssh_config file:

```
sudo nano /etc/ssh_config
```

2. Uncomment the following line and change the value to "yes":

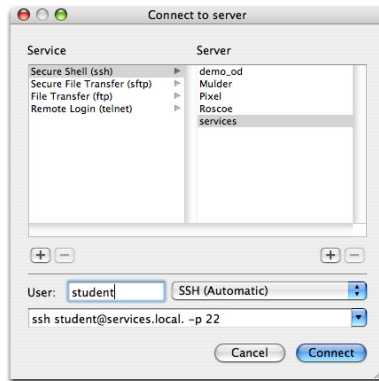
```
GSSAPIAuthentication yes
```

3. Exit the text editor by pressing Control+X, then type "y" to save the changes.

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.

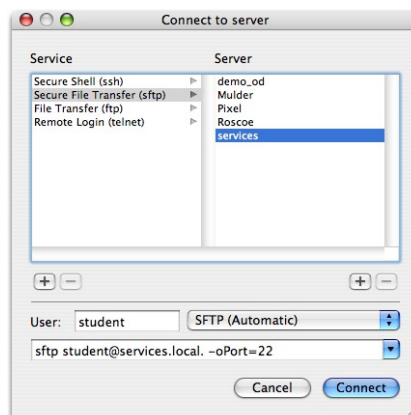
1. Log in as an AD user and verify that you initiated your Kerberos tickets (review Chapter 4, section B)
2. In the Terminal application, choose "Connect to Server..." from the File menu.
3. In the Service column, select "Secure Shell (ssh)"
4. In the Server column, locate your Secure Shell server
5. 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 and verify that you initiated your Kerberos tickets (review Chapter 4, section B)
2. In the Terminal application, choose "Connect to Server..." from the File menu.
3. In the Service column, select "Secure File Transfer (sftp)"
4. In the Server column, locate your Secure Shell server
5. 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, choose the "For selected services below" radio button.
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.

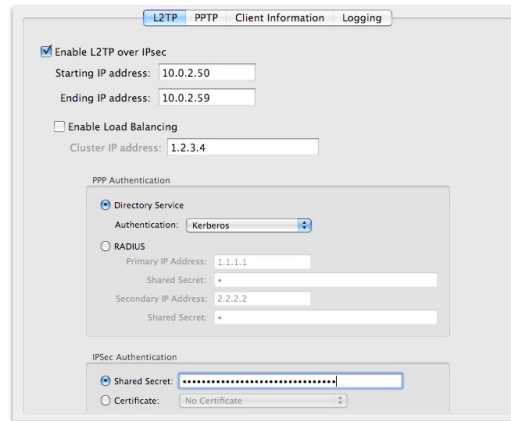
Tip: See the resources section at the end of this document for an article that describes how to create Mail Account bundles. Mail Account bundles make it easy for you to deploy custom account templates to your users.

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 > [your server] > VPN > Settings > L2TP. For example:
 - Enable L2TP over IPsec
 - Starting IP address: 10.0.4.50
 - Ending IP address: 10.0.4.59
 - PPP Authentication: Kerberos
 - IPsec Authentication: Shared Secret (make something up)



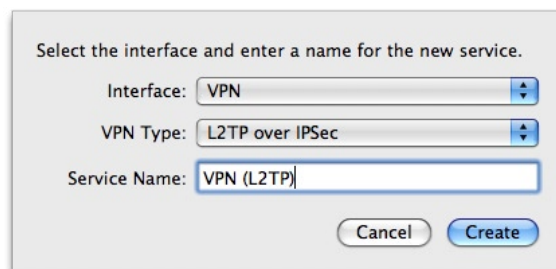
2. In the “Client Information” tab, use settings such as:

- DNS servers: 10.0.4.1
- Search domains: apple.edu
- Network Routing Definition:
 - Address: 10.0.4.0
 - Mask: 255.255.255.0
 - Type: Private

3. Save the settings and start the service

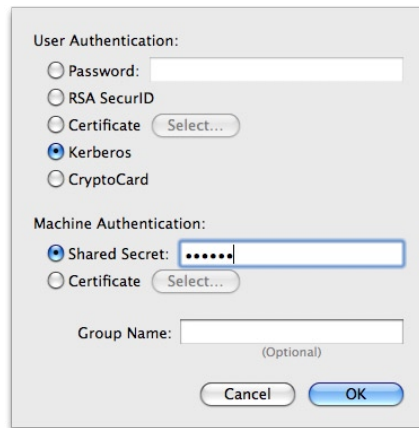
4. At the client, log in as an AD user and open the Network Preference Pane in the System Preferences application.

5. Click on the “+” button to add a network interface. Choose “VPN” from the Interface popup button. Choose “L2TP over IPsec” as the VPN Type. Click the “Create” button.



6. Provide the server address (you can leave the Account Name field blank). Click on the “Authentication Settings” button and configure as follows:

- User Authentication: Kerberos
- Machine Authentication: Shared Secret (type in the shared secret)



7. Click OK to close the Authentication Settings panel, click "Apply" 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.

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.

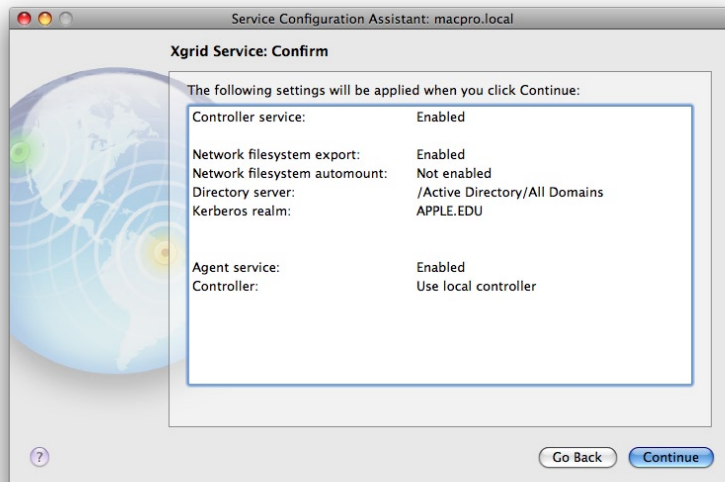
Within a grid, authentication takes place several times: between the agent and controller when an agent joins a grid, between the client and controller when the client submits a job, and between the controller and agents when the controller splits jobs out to the agents. Kerberos authentication makes 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 3.0 or higher installed, with the Developer Examples.

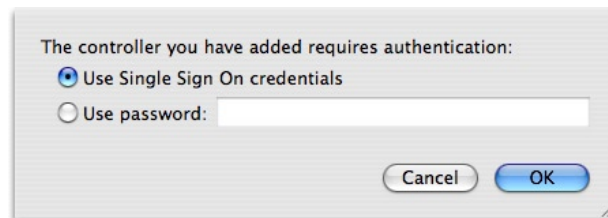
Follow these steps to configure the Xgrid service on Mac OS X Server to enable SSO support from your AD server.

1. Launch Server Admin and enable the Xgrid service for your server. Navigate to the Xgrid service and click on the "General" button in the toolbar.

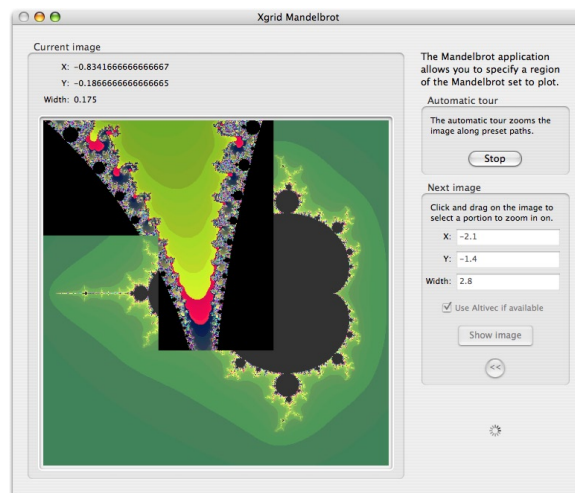
2. Click on the “Configure Xgrid Service...” button in the lower right corner of the window. Follow the instructions in the Server Configuration Assistant to “Host a grid”. When prompted to configure a network filesystem, leave the directory administrator username and password fields blank.



3. Click the Continue button to complete service setup. The Xgrid service (agent and controller) will be enabled.
4. At a client machine bound to AD, login as an AD user.
5. In the Sharing Preference Pane, check the box next to the Xgrid service, then click on the Configure button. Choose your server as the specific controller and choose to accept tasks always. Click OK. Select “Single Sign On” as the authentication method. Close System Preferences.
6. 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.



7. 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.
8. 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.
9. 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.



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



11. 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.

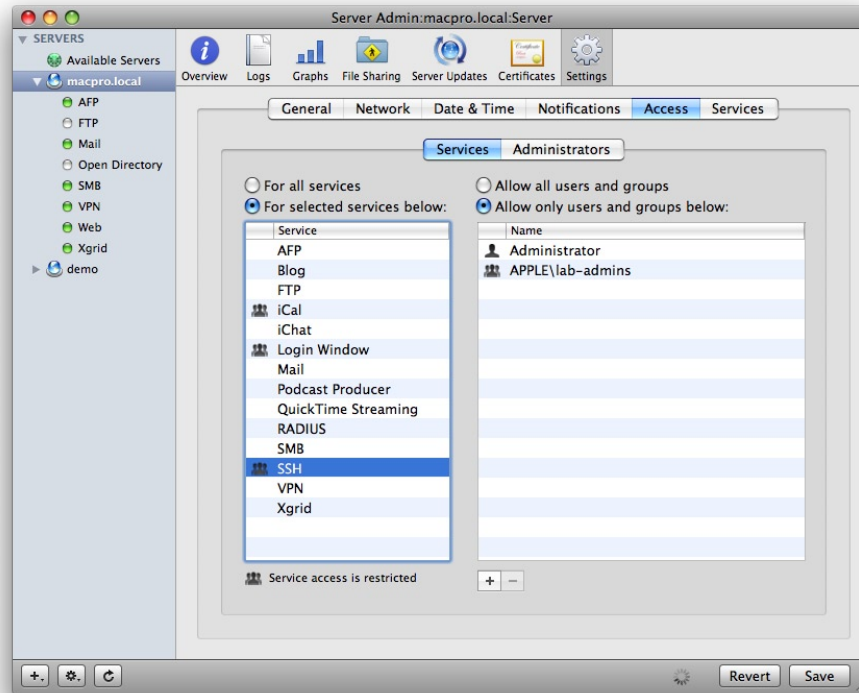
J. Restricting access to services

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 directory store. 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] > Access > Services, select the radio button “For the selected services below”.
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.

Additional Resources

By now you have learned 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.

Mac OS X Directory Services v10.5
<http://training.apple.com/itpro/leopard301>

Additionally, “Mac OS X Server Open Directory Administration” and “User Management” documents are available from Apple at <http://www.apple.com/server/macosx/resources>.

Resources online

<http://www.apple.com/server/macosx/resources>
<http://macenterprise.org>
<http://www.afp548.com>
<http://pgina.xpasystems.com>

White papers

Augmented Records:

<http://www.afp548.com/article.php?story=20071210105328355>

Novell eDirectory integration:

<http://macenterprise.org/content/view/80/77/>

Leveraging eDirectory with Apple Workgroup Manager:

<http://www.novell.com/coolsolutions/tip/5955.html>

Creating Mail Account Bundles:

http://developer.apple.com/documentation/MacOSXServer/Conceptual/XServer_ProgrammingGuide

Leopard Directory Services Release Notes:

<http://developer.apple.com/releasenotes/MacOSXServer/RN-DirectoryServices/index.html>

John DeTroye's "Tips and Tricks for Macintosh Management":

http://homepage.mac.com/johnd/.cv/johnd/Sites/.Public/Latest_Tips/Leopard_Tips/tandtleo.pdf-zip.zip

Scripts and utilities

Example AD bind script:

<http://www.bombich.com/mactips/scripts.html#ad-bind>

Mac OS X Client SACLs:

<http://www.bombich.com/mactips/scripts.html#saclutil>

Creating augmented records en masse

<http://www.bombich.com/mactips/scripts.html#augments>

Appendix A: AD settings for your organization

Active Directory Domain:	_____
AD DNS Server:	_____
Bind Account*:	_____
Bind Account password:	[Just make sure you know it!]
Computer OU:	_____
Computer ID:	_____
AD Search base:	_____
Your department OU:	_____
Users OU:	_____

* Within Active Directory you can delegate the privilege to “Join a computer to the domain”. Creating an account that has only this privilege is preferred to using a fully privileged network administrator.

Document History

June 18, 2008

- Initial release, updated for Leopard

August 22, 2008

- Page 33: Updated instructions for kerberizing services on Mac OS X Server
- Page 34: Added a note about the search policy order when implementing the Golden Triangle management approach on Mac OS X clients
- Page 37: Added note about John DeTroye's Macintosh Management document
- Added a chapter on Augmented Records

October 3, 2008

- Page 47: Made some adjustments to indicate that the home directory server's Directory Services must be configured like the client's.