Mac OS X Server Open Directory Administration

For Version 10.3 or Later

034-2352/09-20-03

Contents

About This Guide

This guide describes the directory services and authentication services that Mac OS X Server can provide to Mac OS X client computers.

Here is a summary of each chapter's contents:

- Chapter 1, "Directory Service Concepts," explains what directory domains are, how they are used, and how they are organized. It also discusses how the discovery of network services is integrated with directory services.
- Chapter 2, "Open Directory Search Policies," describes search policies with one or more directory domains, and describes automatic, custom, and local-only search policies.
- Chapter 3, "User Authentication With Open Directory," describes Open Directory authentication, shadow and crypt passwords, Kerberos, LDAP bind, single signon, and cached authentication for mobile accounts.
- Chapter 4, "Open Directory Planning," helps you assess your directory domain needs, estimate directory and authentication requirements, identify servers of hosting shared domains, improve performance and redundancy, deal with replication in a multi-building campus, and make your Open Directory services secure. This chapter also introduces the tools you use to manage Open Directory services.
- Chapter 5, "Setting Up Open Directory Services," tells you how to set the Open Directory role of Mac OS X Server: standalone server, connected to a directory system, Open Directory master, or Open Directory replica. This chapter also tells you how to set some options of the LDAP service of an Open Directory master or replica and explains how to migrate a directory domain from NetInfo to LDAP. This chapter also tells you how to set up single signon and Kerberos authentication on an Open Directory master.
- Chapter 6, "Managing User Authentication," describes how to set password policies, change a user's password type, assign administrator rights for Open Directory authentication, reset passwords of imported user accounts, and migrate passwords to Open Directory authentication.

- Chapter 7, "Managing Directory Access," explains how to use the Directory Access application. This chapter tells you how to set up services and authentication and contacts search policies. This chapter also explains how to configure access to different directory domains: LDAP, Active Directory, NIS, BSD configuration files, and NetInfo.
- Chapter 8, "Maintenance and Problem Solving," tells you how to monitor Open Directory services, directly view and edit directory data with the Inspector, and back up Open Directory files. This chapter also describes solutions to some problems you may encounter.
- Appendix A, "Mac OS X Directory Data," lists the Open Directory extensions to the LDAP schema and specifies the standard record types and attributes of Mac OS X.
- Appendix B, "Open Directory Password Server Authentication Methods," describes the authentication methods that Open Directory supports.
- Appendix C, "Authentication Manager," tells you about the Authentication Manager technology that provides compatibility with user accounts created in Mac OS X Server version 10.0–10.2.
- The Glossary defines terms you'll encounter as you read this guide.

Using This Guide

The chapters in this guide are arranged in the order that you're likely to need them when setting up and managing Open Directory on your server.

- Review Chapter 1 through Chapter 3 to acquaint yourself with Open Directory concepts: directory services, search policies, and authentication.
- Read Chapter 4 when you're ready to plan directory services and password authentication for your network.
- After you finish planning, use the instructions in Chapter 5 to set up Open Directory services.
- Whenever you need to set password policies or change password settings in a user account, look for instructions in Chapter 6.
- If you need to set up or change how a Mac OS X or Mac OS X Server computer accesses directory domains, follow the instructions in Chapter 7.
- For ongoing maintenance of directory and authentication services, use Chapter 8.

Getting Additional Information

Mac OS X Server comes with a suite of guides that explain other services and provide instructions for configuring, managing, and troubleshooting those services. Most of these documents are on the server discs in the form of PDF files. All of them are available in PDF format from www.apple.com/server/documentation.

This guide	Tells you how to
Mac OS X Server Getting Started For Version 10.3 or Later	Understand the new features of Mac OS X Server version 10.3 and prepare your server.
Mac OS X Server Migration To Version 10.3 or Later	Reuse data and service settings on Mac OS X Server version 10.3 that are currently being used on earlier versions of the server.
Mac OS X Server User Management For Version 10.3 or Later	Create and manage user, group, and computer accounts. Set up managed preferences for Mac OS 9 and Mac OS X clients.
Mac OS X Server File Services Administration For Version 10.3 or Later	Share selected server volumes or folders among server clients using these protocols: AFP, NFS, FTP, and SMB.
Mac OS X Server Print Service Administration For Version 10.3 or Later	Host shared printers and manage their associated queues and print jobs.
Mac OS X Server System Image Administration For Version 10.3 or Later	Create disk images and set up the server so that other Macintosh computers can start up from those images over the network. This guide covers NetBoot and Network Install.
Mac OS X Server Mail Service Administration For Version 10.3 or Later	Set up, configure, and administer mail services on the server.
Mac OS X Server Web Technologies Administration For Version 10.3 or Later	Set up and manage a web server, including WebDAV, WebMail, and web modules.
Mac OS X Server Network Services Administration For Version 10.3 or Later	Set up, configure, and administer DHCP, DNS, IP firewall, NAT, and VPN services on the server.
Mac OS X Server Windows Services Administration For Version 10.3 or Later	Set up and manage services for Windows users.
Mac OS X Server QuickTime Streaming Server Administration For Version 10.3 or Later	Set up and manage QuickTime streaming services.
Mac OS X Server: Java Application Server Administration	Deploy and manage J2EE applications using a JBoss application server on Mac OS X Server.
Mac OS X Server Command-Line Administration For Version 10.3 or Later	Use commands and configuration files to perform server administration tasks in a UNIX command shell.

For more information, consult these resources:

- Read Me documents contain important updates and special information. Look for them on the server discs.
- Online help, available from the Help menu in all the server applications, provides onscreen instructions for administration tasks as well as late-breaking news and web updates.
- Apple support webpages and the AppleCare Knowledge Base provide answers to common questions and the latest information updates. These are available at www.info.apple.com/
- Apple Training offers courses for technical coordinators and system administrators. For a course catalog, visit the following website:
  train.apple.com/
- Discussion groups and mailing lists put you in touch with other server administrators, who may have already found solutions to problems you encounter. To find discussion groups and mailing lists, visit the following websites:
  discussions.info.apple.com/

  www.lists.apple.com/

Directory Service Concepts

1

A directory service provides a central repository for information about computer users and network resources in an organization.

Storing administrative data in a central repository has many benefits:

- Reduces data entry effort.
- Ensures all network services and clients have consistent information about users and resources.
- Simplifies administration of users and resources.
- Provides identification, authentication, and authorization services for other network services.

In education and enterprise environments, directory services are the ideal way to manage users and computing resources. Organizations with as few as 10 people can benefit by deploying 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 all 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. Having information about users and resources centralized can reduce the system administrator's user management burden. In addition, users can log in to any authorized computer on the network. Anywhere a user logs in, the user can get the same home directory, and with it the user's personal desktop appears, customized for the user's individual preferences. The user always has access to personal files and can easily locate and use authorized 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:

- LDAP service on a Mac OS X Server system
- NetInfo service on a computer with Mac OS X or Mac OS X Server
- Active Directory service on a Microsoft Windows server
- OpenLDAP or other LDAP service on a third-party server such as Sun One or Novell eDirectory
- NIS on a UNIX server
- BSD configuration files stored locally (not retrieved from a server)

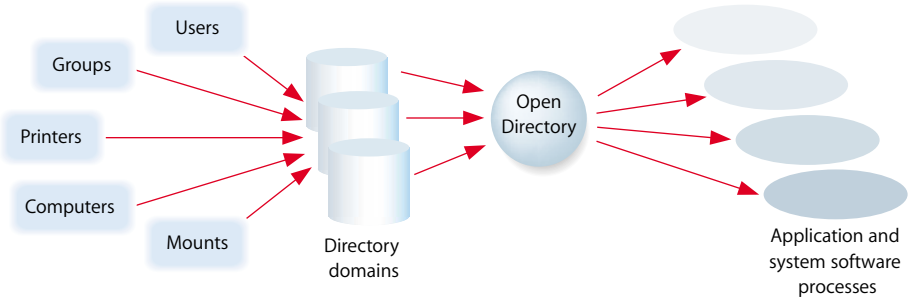
Mac OS 9 and Mac OS 8 managed clients also use Open Directory to retrieve some user information. For more information, see the Macintosh Manager chapter in the user management guide (available at www.apple.com/server/documentation/).

In addition, Mac OS X and Mac OS X Server can use Open Directory to discover network services, such as file servers, that make themselves known with the Rendezvous, AppleTalk, SLP, or SMB service discovery protocols.

The Open Directory architecture also includes authentication service. Open Directory can securely store and validate the passwords of users who want to log in to client computers on your network or use other network resources that require authentication. Open Directory can also enforce such policies as password expiration and minimum length. Open Directory can also authenticate Windows computer users for domain login, file service, print service, and other Windows services provided by Mac OS X Server.

Directory Services and Directory Domains

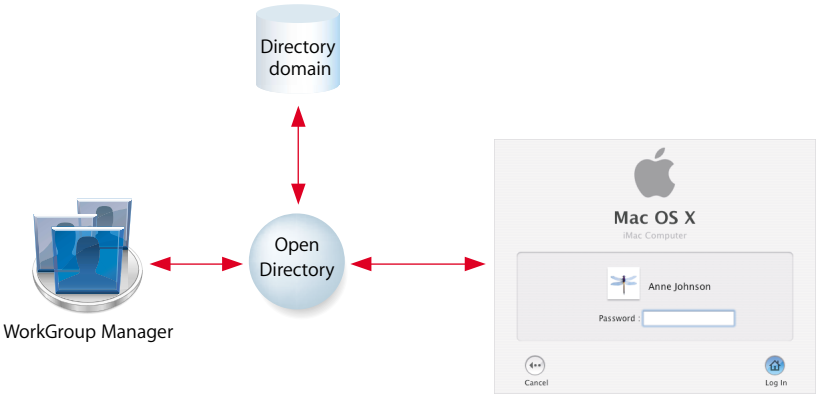
A directory service acts as an intermediary between application and system software processes, which need information about users and resources, and the *directory domains* that store the information. In Mac OS X and Mac OS X Server, Open Directory provides directory services. Open Directory can access information in one directory domain or several directory domains.



A directory domain stores information in a specialized database that is optimized to handle a great many requests for information and to find and retrieve information quickly.

Processes running on Mac OS X computers can use the Open Directory services to save information in directory domains. For example, when you create a user account with Workgroup Manager, it has Open Directory store user name and other account information in a directory domain. Of course you can then review user account information with Workgroup Manager, and it has Open Directory retrieve the user information from a directory domain.

Other application and system software processes can also use the user account information stored in directory domains. When someone attempts to log in to a Mac OS X computer, the login process uses Open Directory services to validate the user name and password.

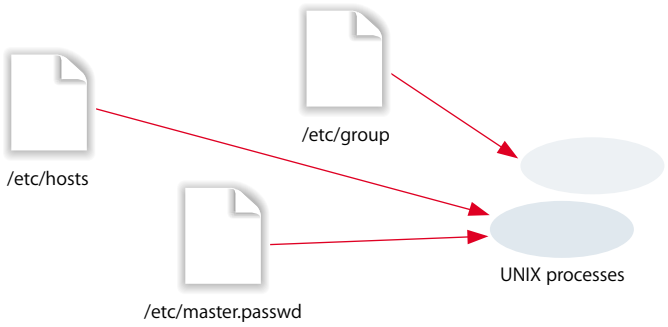


A Historical Perspective

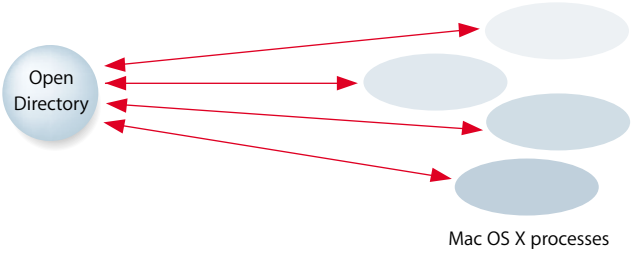
Like Mac OS X, Open Directory has a UNIX heritage. Open Directory provides access to administrative data that UNIX systems have generally kept in configuration files, which require much painstaking work to maintain. (Some UNIX systems still rely on configuration files.) Open Directory consolidates the data and distributes it for ease of access and maintenance.

Data Consolidation

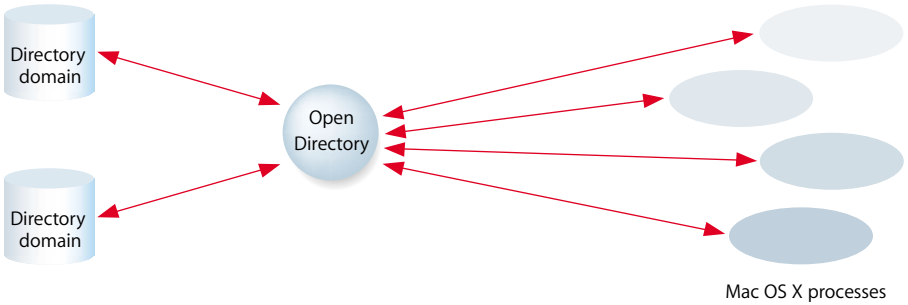
For years, UNIX systems have stored administrative information in a collection of files located in the /etc directory. This scheme requires each UNIX computer to have its own set of files, and processes that are running on a UNIX computer read its files when they need administrative information. If you're experienced with UNIX, you probably know about the files in the /etc directory—group, hosts, hosts.eq, master.passwd, and so forth. For example, a UNIX process that needs a user's password consults the /etc/master.passwd file. The /etc/master.passwd file contains a record for each user account. A UNIX process that needs group information consults the /etc/group file.



Open Directory consolidates administrative information, simplifying the interactions between processes and the administrative data they create and use.



Processes no longer need to know how and where administrative data is stored. Open Directory gets the data for them. If a process needs the location of a user's home directory, the process simply has Open Directory retrieve the information. Open Directory finds the requested information and then returns it, insulating the process from the details of how the information is stored. If you set up Open Directory to access administrative data in several directory domains, Open Directory automatically consults them as needed.



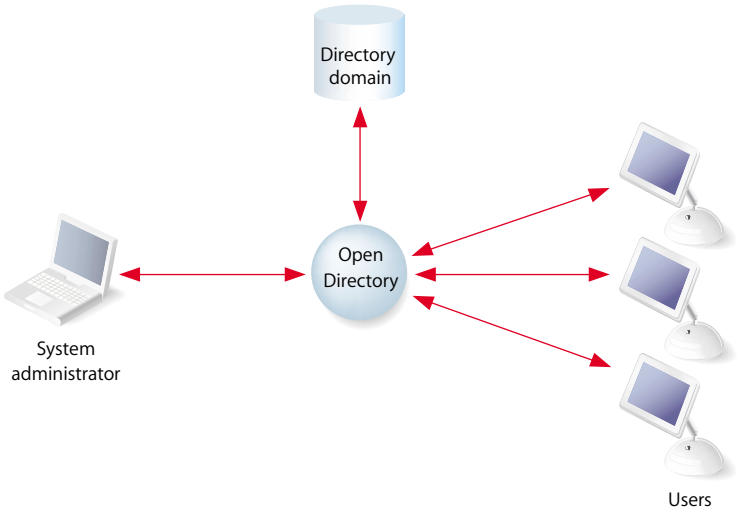
Some of the data stored in a directory domain is identical to data stored in UNIX configuration files. For example, the crypt password, home directory location, real name, user ID, and group ID—all stored in the user records of a directory domain—have corresponding entries in the standard /etc/passwd file. However, a directory domain stores much additional data to support functions that are unique to Mac OS X, such as support for managing Mac OS X client computers.

Data Distribution

Another characteristic of UNIX configuration files is that the administrative data they contain is available only to the computer on which they are stored. Each computer has its own UNIX configuration files. With UNIX configuration files, each computer that someone wants to use must have that person's user account settings stored on it, and each computer must store the account settings for every person who can use the computer. To set up a computer's network settings, the administrator needs to go to the computer and directly enter the IP address and other information that identifies the computer on the network.

Similarly, when user or network information needs to be changed in UNIX configuration files, the administrator must make the changes on the computer where the files reside. Some changes, such as network settings, require the administrator to make the same changes on multiple computers. This approach becomes unwieldy as networks grow in size and complexity.

Open Directory solves this problem by letting you store administrative data in a directory domain that can be managed by a network administrator from one location. Open Directory lets you distribute the information so that it is visible on a network to the computers that need it and the administrator who manages it.



Uses of Directory Data

Open Directory makes it possible to consolidate and maintain network information easily in a directory domain, but this information has value only if application and system software processes running on network computers actually access the information.

Here are some of the ways in which Mac OS X system and application software use directory data:

- **Login:** As mentioned already, Workgroup Manager can create user records in a directory domain, and these records can be used to authenticate users who log in to Mac OS X computers and Windows computers. When a user specifies a name and a password in the Mac OS X login window, the login process asks Open Directory to authenticate the name and password. Open Directory uses the name to find the user's account record in a directory domain and uses additional data in the user record to validate the password.

- **Folder and file access:** After logging in successfully, a user can access files and folders. Mac OS X uses another data item from the user record—the user ID (UID)—to determine the user's access privileges for a file or folder that the user wants to access. When a user accesses a folder or file, the file system compares this user's UID to the UID assigned to the folder or file. If the UIDs are the same, the file system grants owner privileges (usually read and write privileges) to the user. If the UIDs are different, the user doesn't get owner privileges.
- **Home directories:** Each user record in a directory domain stores the location of the user's home directory, which is also known as the user's home folder. This is where the user keeps personal files, folders, and preferences. A user's home directory can be located on a particular computer that the user always uses or on a network file server.
- **Automount share points:** Share points can be configured to automount (appear automatically) in the /Network folder (the Network globe) in the Finder windows of client computers. Information about these automount share points is stored in a directory domain. *Share points* are folders, disks, or disk partitions that you have made accessible over the network.
- **Mail account settings:** Each user's record in a directory domain specifies whether the user has mail service, which mail protocols to use, how to present incoming mail, whether to alert the user when mail arrives, and more.
- **Resource usage:** Disk, print, and mail quotas can be stored in each user record of a directory domain.
- **Managed client information:** The administrator can manage the Mac OS X environment of users whose account records are stored in a directory domain. The administrator makes mandatory preference settings that are stored in the directory domain and override users' personal preferences.
- **Group management:** In addition to user records, a directory domain also stores group records. Each group record affects all users who are in the group. Information in group records specifies preferences settings for group members. Group records also determine access to files, folders, and computers.

Inside a Directory Domain

Information in a directory domain is organized into *record types*, which are specific categories of records, such as users, computers, and mounts. For each record type, a directory domain may contain any number of records. Each record is a collection of attributes, and each attribute has one or more values. If you think of each record type as a spreadsheet that contains a category of information, then records are like the rows of the spreadsheet, attributes are like spreadsheet columns, and each spreadsheet cell contains one or more values.

For example, when you define a user by using Workgroup Manager, you are creating a user record (a record of the user's record type). The settings that you configure for the user—short name, full name, home directory location, and so on—become values of attributes in the user record. The user record and the values of its attributes reside in a directory domain.

In some directory services, such as LDAP and Active Directory, record types are called *object classes* or simply classes, and records are called objects. Each class (record type) is a set of rules that define similar objects (records) by specifying certain attributes that each object must have and certain other attributes that each object may have. For example, the inetOrgPerson class defines objects that contain user attributes. The inetOrgPerson class is a standard LDAP class defined by RFC 2798. Other standard LDAP classes and attributes are defined by RFC 2307.

A collection of attributes and record types or object classes provides a blueprint for the information in a directory domain. This blueprint is called the *schema* of the directory domain.

Local and Shared Directory Domains

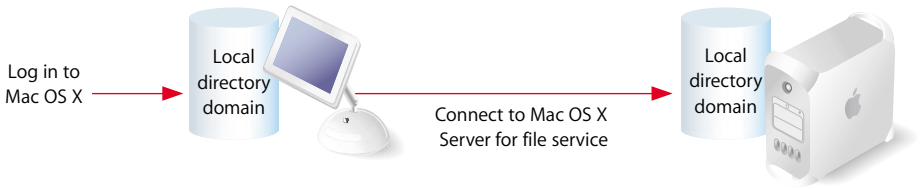
Where you store your server's user information and other administrative data is determined by whether the data needs to be shared. This information may be stored in the server's local directory domain or in a shared directory domain.

About the Local Directory Domain

Every Mac OS X computer has a local directory domain. A local domain's administrative data is visible *only* to applications and system software running on the computer where the domain resides. It is the first domain consulted when a user logs in or performs some other operation that requires data stored in a directory domain.

When the user logs in to a Mac OS X computer, Open Directory searches the computer's local directory domain for the user's record. If the local directory domain contains the user's record (and the user typed the correct password), the login process proceeds and the user gets access to the computer.

After login, the user could choose "Connect to Server" from the Go menu and connect to Mac OS X Server for file service. In this case, Open Directory on the server searches for the user's record in the server's local directory domain. If the server's local directory domain has a record for the user (and the user types the correct password), the server grants the user access to the file services.

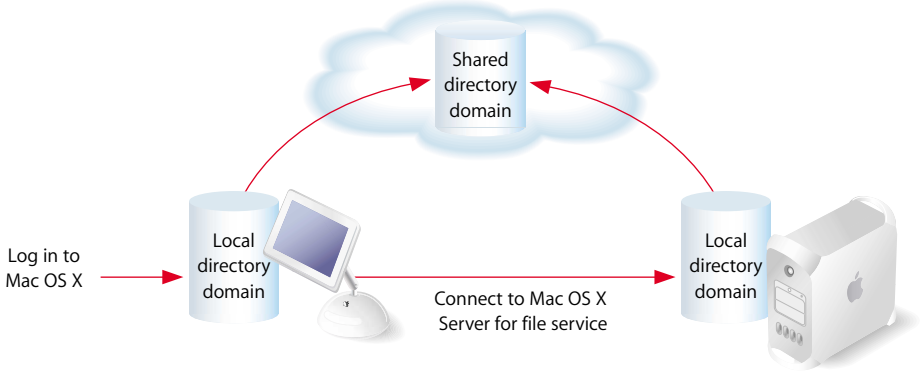


When you first set up a Mac OS X computer, its local directory domain is automatically created and populated with records. For example, a user record is created for the user who performed the installation. It contains the user name and password entered during setup, as well as other information, such as a unique ID for the user and the location of the user's home directory.

About Shared Directory Domains

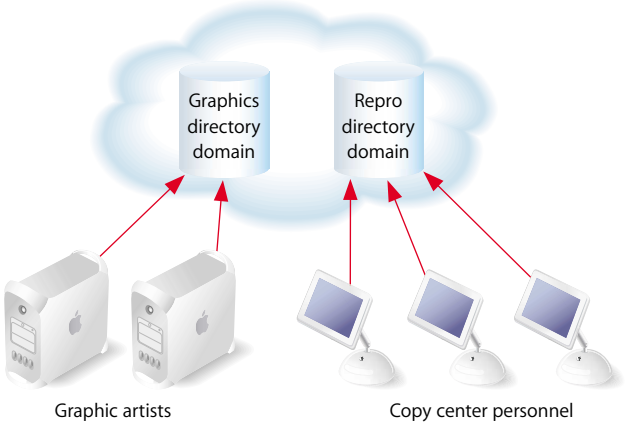
While Open Directory on any Mac OS X computer can store administrative data in the computer's local directory domain, the real power of Open Directory is that it lets multiple Mac OS X computers share administrative data by storing the data in shared directory domains. When a computer is configured to use a shared domain, any administrative data in the shared domain is also visible to applications and system software running on that computer.

If Open Directory does not find a user's record in the local domain of a Mac OS X computer, Open Directory can search for the user's record in any shared domains to which the computer has access. In the following example, the user can access both computers because the shared domain accessible from both computers contains a record for the user.



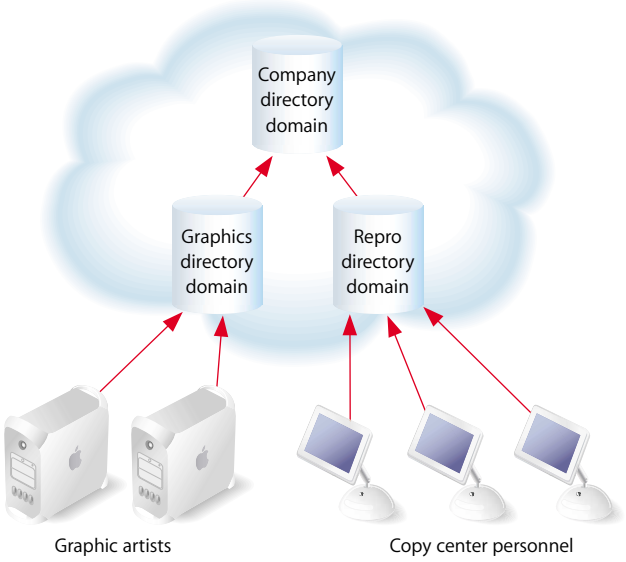
Shared domains generally reside on servers because directory domains store extremely important data, such as the data for authenticating users. Access to servers is usually tightly restricted to protect the data on them. In addition, directory data must always be available. Servers often have extra hardware features that enhance their reliability, and servers can be connected to uninterruptible power sources.

Shared directories can also be used to isolate network resources from some users. For example, graphic artists in a company might need to work on high-performance computers, while copy center personnel can work on computers with standard equipment. You could restrict access to the two kinds of computers by setting up user records in two shared directory domains. One shared domain would contain records for users who work on the high-performance computers. The other shared domain would contain records for users who work on the standard computers. Rather than configuring user access on each computer individually, you would use Workgroup Manager to create all the user records in the two shared domains: Graphics and Repro.



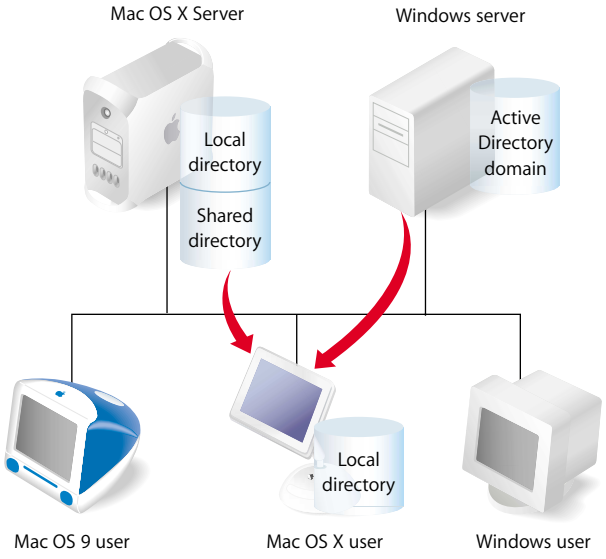
When users log in to high-performance computers, user records in the Graphics directory domain will be used to authenticate them. Likewise when users log in to standard computers, user records in the Repro directory domain will be used to authenticate them. A user whose record is in the Repro directory can't log in to a high-performance computer. A user whose record in the Graphics directory domain can't log in to a standard computer.

If you wanted some users to be able to log in to any computer, you could create their user records in another shared directory domain that all computers can access.



Shared Data in Existing Directory Domains

Some organizations—such as universities and worldwide corporations—maintain user information and other administrative data in directory domains on UNIX or Windows servers. Open Directory can be configured to search these non-Apple domains as well as shared Open Directory domains of Mac OS X Server systems.



The order in which Mac OS X searches directory domains is configurable. A search policy determines the order in which Mac OS X searches directory domains. The next chapter discusses search policies.

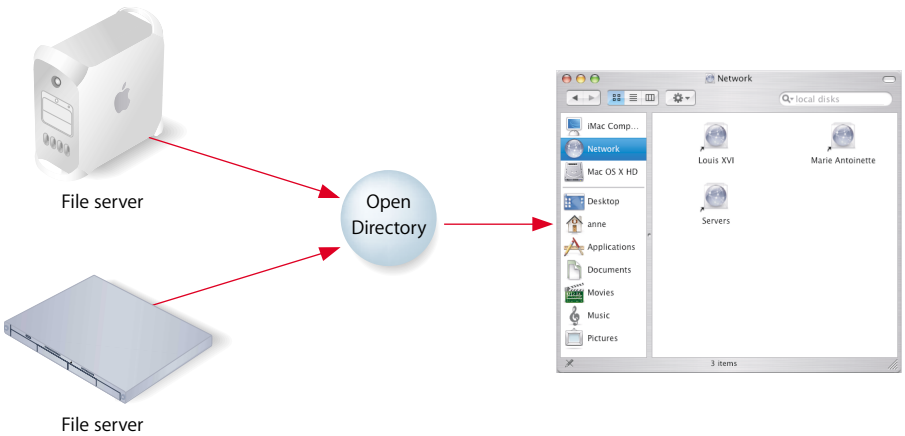
Access to Directory Services

Open Directory can access directory domains in the following kinds of directory services:

- Lightweight Directory Access Protocol (LDAP), an open standard commonly used in mixed environments and the native directory service for shared directories in Mac OS X Server version 10.3
- NetInfo, the legacy directory service of Mac OS X and Mac OS X Server
- Active Directory, the directory service of Microsoft Windows 2000 and 2003 servers
- Network Information System (NIS), a directory service of many UNIX servers
- BSD flat files, the legacy directory service of UNIX systems

Discovery of Network Services

Open Directory can provide more than administrative data from directories. Open Directory can also provide information about services that are available on the network. For example, Open Directory can provide information about file servers that are currently available.



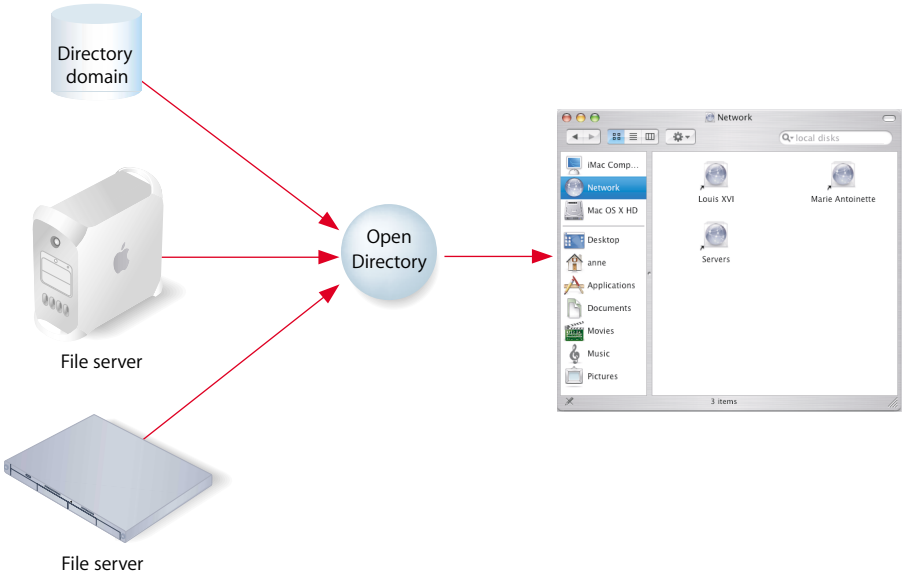
Open Directory can discover network services that make their existence and whereabouts known. Services make themselves known by means of standard protocols. Open Directory supports the following service discovery protocols:

- Rendezvous, the Apple protocol that uses multicast DNS for discovering file, print, chat, music sharing, and other network services
- AppleTalk, the legacy protocol for discovering file, print, and other network services

- Service Location Protocol (SLP), an open standard for discovering file and print services
- Server Message Block (SMB), the protocol used by Microsoft Windows for file, print, and other services

In fact, Open Directory can provide information about network services both from service discovery protocols and from directory domains. To accomplish this, Open Directory simply asks all its sources of information for the type of information requested by a Mac OS X process. The sources that have the requested type of information provide it to Open Directory, which collects all the provided information and hands it over to the Mac OS X process that requested it.

For example, if Open Directory requests information about file servers, the file servers on the network respond via service discovery protocols with their information. A directory domain that contains relatively static information about some file servers also responds to the request. Open Directory collects the information from the service discovery protocols and the directory domains.



When Open Directory requests information about a user, service discovery protocols don't respond because they don't have user information. (Theoretically, AppleTalk, Rendezvous, SMB, and SLP could provide user information, but in practice they don't have any user information to provide.) The user information that Open Directory collects comes from whatever sources have it—from directory domains.

Open Directory Search Policies

2

Each computer has a search policy that specifies one or more directory domains and the sequence in which Open Directory searches them.

Every Mac OS X computer has a local directory domain and can also access shared directory domains. Nothing prevents the directory domains from having interchangeable records. For example, two directory domains could have user records with the same name but other differences. Therefore, each Mac OS X computer needs a policy for resolving potential conflicts between equivalent but not identical records.

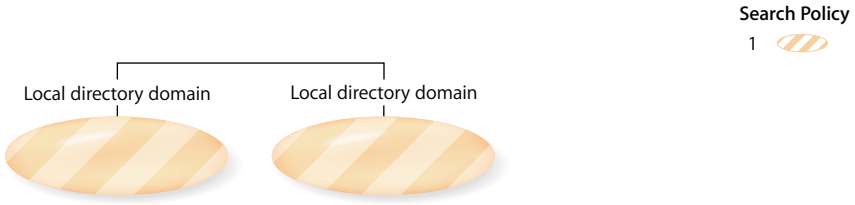
Each Mac OS X computer has a *search policy* that specifies which directory domains Open Directory can access, such as the computer's local directory and a particular shared directory. The search policy also specifies the order in which Open Directory accesses directory domains. Open Directory searches each directory domain in turn and stops searching when it finds a match. For example, Open Directory stops searching for a user record when it finds a record whose user name matches the name it's looking for.

Search Policy Levels

A search policy can include the local directory alone, the local directory and a shared directory, or the local directory and multiple shared directories. On a network with a shared directory, several computers generally access the shared directory. This arrangement can be depicted as a tree-like structure with the shared directory at the top and local directories at the bottom.

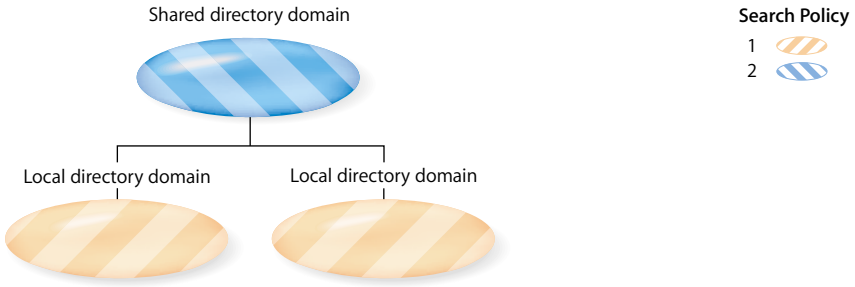
Local Directory Search Policy

The simplest search policy consists only of a computer's local directory. In this case, Open Directory looks for user information and other administrative data only in the local directory domain of each computer. If a server on the network hosts a shared directory, Open Directory does not look there for user information or administrative data because the shared directory is not part of the computer's search policy.

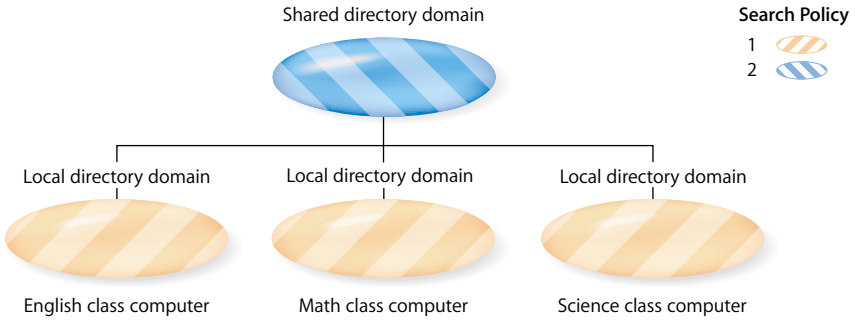


Two-Level Search Policies

If one of the servers on the network hosts a shared directory, all the computers on the network can include the shared directory in their search policies. In this case, Open Directory looks for user information and other administrative data first in the local directory. If Open Directory doesn't find the information it needs in the local directory, it looks in the shared directory.

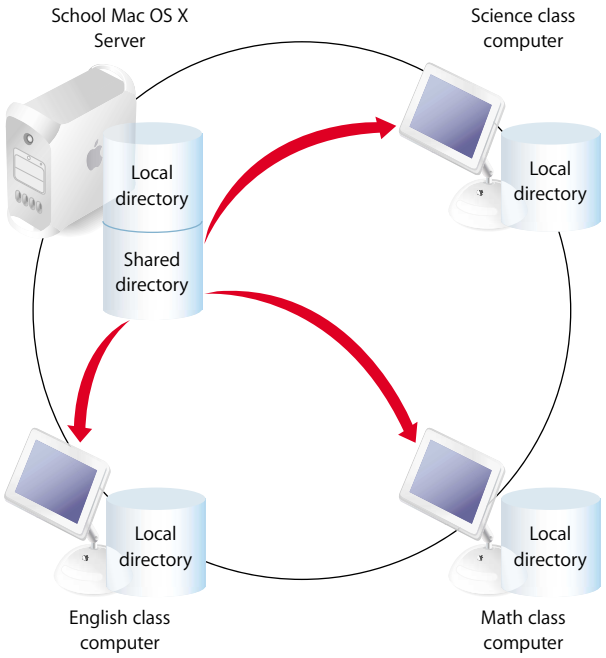


Here's a scenario in which a two-level search policy might be used:



Each class (English, math, science) has its own computer. The students in each class are defined as users in the local domain of that class's computer. All three of these local domains have the same shared domain, in which all the instructors are defined. Instructors, as members of the shared domain, can log in to all the class computers. The students in each local domain can log in to only the computer where their local account resides.

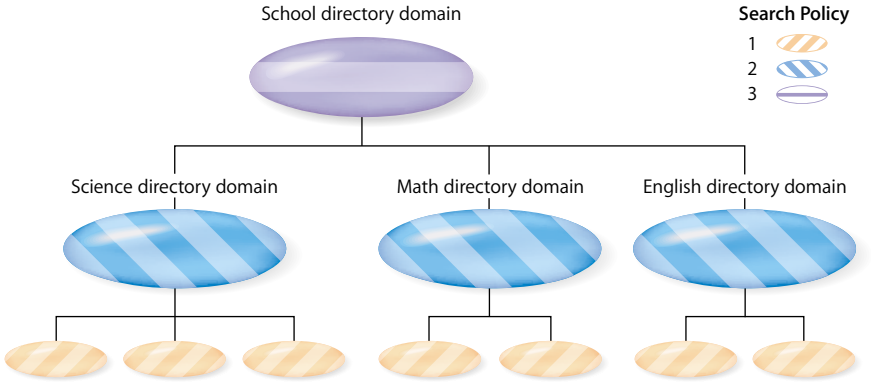
While local domains reside on their respective computers, a shared domain resides on a server accessible from the local domain's computer. When an instructor logs in to any of the three class computers and cannot be found in the local domain, Open Directory searches the shared domain. In this example, there is only one shared domain, but in more complex networks, there may be more shared domains.



Multilevel Search Policies

If more than one server on the network hosts a shared directory, the computers on the network can include two or more shared directories in their search policies. As with simpler search policies, Open Directory always looks for user information and other administrative data first in the local directory. If Open Directory does not find the information it needs in the local directory, it searches each shared directory in the sequence specified by the search policy.

Here's a scenario in which more than one shared directory might be used:



Each class (English, math, science) has a server that hosts a shared directory domain. Each classroom computer's search policy specifies the computer's local domain, the class's shared domain, and the school's shared domain. The students in each class are defined as users in the shared domain of that class's server, allowing them to log in to any computer in the class. The instructors are defined in the shared domain of the school server, allowing them to log in to any classroom computer.

You can affect an entire network or just a group of computers by choosing the domain in which to define administrative data. The higher the administrative data resides in a search policy, the fewer places it needs to be changed as users and system resources change. Probably the most important aspect of directory services for administrators is planning directory domains and search policies. These should reflect the resources you want to share, the users you want to share them among, and even the way you want to manage your directory data.

Automatic Search Policies

Initially, Mac OS X computers are configured to set their search policies automatically. An automatic search policy consists of three parts, two of which are optional:

- Local directory domain
- Shared NetInfo domains (optional)
- Shared LDAP directory (optional)

A computer's automatic search policy always begins with the computer's local directory domain. If a Mac OS X computer is not connected to a network, the computer searches only its local directory domain for user accounts and other administrative data.

Next the automatic search policy looks at the binding of shared NetInfo domains. The computer's local domain can be bound to a shared NetInfo domain, which can in turn be bound to another shared NetInfo domain, and so on. The NetInfo binding, if any, constitutes the second part of the automatic search policy. See "About NetInfo Binding" on page 110 for additional information.

Finally, a computer with an automatic search policy can bind to a shared LDAP directory. When the computer starts up, it can get the address of an LDAP directory server from DHCP service. The DHCP service of Mac OS X Server can supply an LDAP server address just as it supplies the addresses of DNS servers and a router. (A non-Apple DHCP service may also be able to supply an LDAP server address; this feature is known as DHCP option 95.)

If you want the DHCP service of Mac OS X Server to supply its clients with a particular LDAP server's address for their automatic search policies, you need to configure the LDAP options of DHCP service. For instructions, see the DHCP chapter of the network services administration guide.

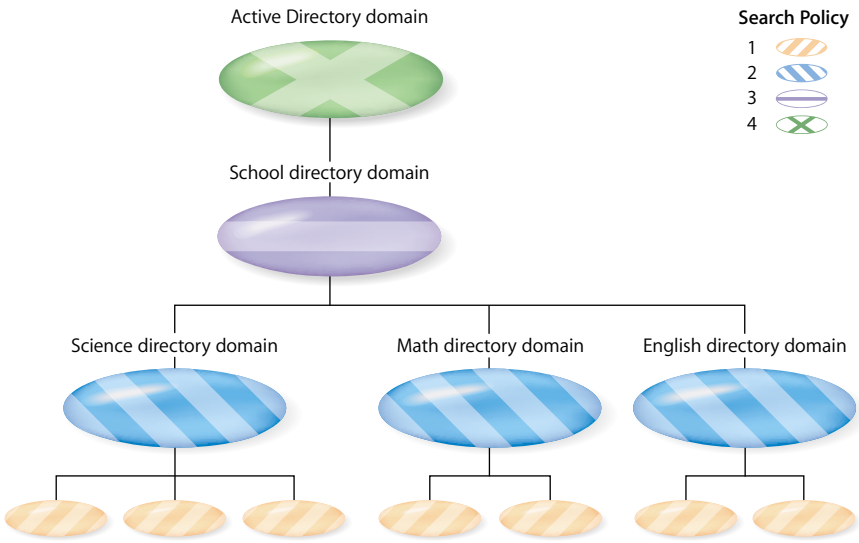
If you want a Mac OS X computer to get the address of an LDAP server from DHCP service:

- The computer must be configured to use an automatic search policy. Mac OS X version 10.2 and later is initially configured to use an automatic search policy. See "Setting Up the Authentication and Contacts Search Policies" on page 87 for more information.
- The computer's Network preferences must be configured to use DHCP or DHCP with manual IP address. Mac OS X is initially configured to use DHCP. For information on setting Network preferences, search Mac Help.

An automatic search policy offers convenience and flexibility and is the recommended option, especially for mobile computers. If a computer with an automatic search policy is disconnected from the network, connected to a different network, or moved to a different subnet, the automatic search policy can change. If the computer is disconnected from the network, it uses its local directory domain. If the computer is connected to a different network or subnet, it can automatically change its NetInfo binding and can get an LDAP server address from the DHCP service on the current subnet. With an automatic search policy, a computer doesn't have to be reconfigured to get directory and authentication services in its new location.

Custom Search Policies

If you don't want a Mac OS X computer to use the automatic search policy supplied by DHCP, you can define a custom search policy for the computer. For example, a custom search policy could specify that an Active Directory domain be consulted when a user record or other administrative data cannot be found in other directory domains.



A custom search policy may not work in multiple network locations because it relies on the availability of specific directory domains. Therefore a custom search policy is usually not the best choice for a mobile computer. An automatic or local-only search policy is generally more reliable for a mobile computer.

Search Policies for Authentication and Contacts

A Mac OS X computer actually has more than one search policy. It has a search policy for finding authentication information, and it has a separate search policy for finding contact information. Open Directory uses the authentication search policy to locate and retrieve user authentication information and other administrative data from directory domains. Open Directory uses the contacts search policy to locate and retrieve name, address, and other contact information from directory domains. Mac OS X Address Book uses this contact information, and other applications can be programmed to use it as well.

Each search policy can be automatic, custom, or local directory only.

User Authentication With Open Directory

3

Open Directory offers a variety of options for authenticating users whose accounts are stored in directory domains on Mac OS X Server, including Kerberos and the many authentication methods that network services require.

Open Directory can authenticate users by:

- Using single signon with the Kerberos KDC built into Mac OS X Server
- Using a password stored securely in the Open Directory Password Server database
- Using a shadow password stored as several hashes, including NT and LAN Manager, in a file that only the root user can access
- Using a crypt password stored directly in the user's account, for backward compatibility with legacy systems
- Using a non-Apple LDAP server for LDAP bind authentication

In addition, Open Directory lets you set up specific password policies for each user, such as automatic password expiration and minimum password length. (Password policies do not apply to shadow passwords, crypt passwords, or LDAP bind authentication.)

This chapter describes the authentication options available in Mac OS X Server.

Authentication and Authorization

Services such as the login window and Apple file service request user authentication from Open Directory. *Authentication* is part of the process by which a service determines whether it should grant a user access to a resource. Usually this process also requires *authorization*. Authentication proves a user's identity, and authorization determines what the authenticated user is allowed to do. A user typically authenticates by providing a valid name and password. A service can then authorize the authenticated user to access specific resources. For example, file service authorizes full access to folders and files that an authenticated user owns.

You experience authentication and authorization when you use a credit card. The merchant authenticates you by comparing your signature on the sales slip to the signature on your credit card. Then the merchant submits your authorized credit card account number to the bank, which authorizes payment based on your account balance and credit limit.

Open Directory authenticates user accounts, but does not authorize access to any services. After Open Directory authenticates you, the login window can authorize you to log in, file service can authorize access to certain folders and files, mail service can authorize access to your email, and so on.

Determining Which Authentication Option to Use

To authenticate a user, Open Directory first must determine which authentication option to use—Kerberos, Open Directory Password Server, shadow password, crypt password, or LDAP bind. The user's account contains information that specifies which authentication option to use. This information is called the *authentication authority attribute*. Therefore Open Directory uses the name provided by the user to locate the user's account in the directory domain. Then Open Directory consults the authentication authority attribute in the user's account and learns which authentication option to use.

The authentication authority attribute is not limited to specifying a single authentication option. For example, an authentication authority attribute could specify that a user can be authenticated by Kerberos and Open Directory Password Server.

Nor must a user's account contain an authentication authority attribute at all. If a user's account contains no authentication authority attribute, Mac OS X Server assumes a crypt password is stored in the user's account. For example, user accounts created using Mac OS X version 10.1 and earlier contain a crypt password but not an authentication authority attribute.

You can change a user's authentication authority attribute by changing the password type in the Advanced pane of Workgroup Manager. Some password type settings result in the authentication authority attribute specifying more than one authentication option. See Chapter 6, "Managing User Authentication," for instructions on setting the password type.

Open Directory Authentication

When a user's account has a password type of Open Directory, the user can be authenticated by Kerberos or the Open Directory Password Server. Neither Kerberos nor Open Directory Password Server stores the password in the user's account.

Both Kerberos and Open Directory Password Server store passwords outside the directory domain and never allow passwords to be read. Passwords can only be set and verified. Malicious users might attempt to log in over the network hoping to gain access to Kerberos and Open Directory Password Server. The Open Directory logs can alert you to unsuccessful login attempts. (See "Viewing Open Directory Status and Logs" on page 115.)

Password Policies

Both Kerberos and Open Directory Password Server enforce password policies. For example, a user's password policy can specify a password expiration interval. If the user is logging in and Open Directory discovers the user's password has expired, the user must replace the expired password. Then Open Directory can authenticate the user.

Password policies can disable a user account on a certain date, after a number of days, after a period of inactivity, or after a number of failed login attempts. Password policies can also require passwords to be a minimum length, contain at least one letter, contain at least one numeral, differ from the account name, differ from recent passwords, or be changed periodically. Open DIrectory applies the same password policy rules to Open Directory Password Server and Kerberos, except that Kerberos does not support all the rules.

Password policies do not affect administrator accounts. Administrators are exempt from password policies because they can change the policies at will. In addition, enforcing password policies on administrators would subject them to denial-of-service attacks.

Which Users Can Have Open Directory Passwords

All user accounts stored on a server with Mac OS X Server version 10.3 can be configured to have a password type of Open Directory. User accounts in the server's local directory domain can be configured to have a password type of Open Directory. If this server hosts a shared directory domain, user accounts in it can also be configured to have a password type of Open Directory.

Users who need to log in using the login window of Mac OS X version 10.1 or earlier must be configured to use crypt passwords. The password type doesn't matter for other services. For example, a user could authenticate for Apple file service with an Open Directory password.

Open Directory Password Server Authentication Methods

The Open Directory Password Server is based on a standard known as Simple Authentication and Security Layer (SASL). It is an extensible authentication scheme that allows Open Directory Password Server to support a variety of network user authentication methods required by mail service, file services, and other services of Mac OS X Server. Each service negotiates with Open Directory Password Server for an authentication method before exchanging user credentials.

The Open Directory Password Server supports authentication methods that do not require a clear text password be stored in the Open Directory Password Server database. Only encrypted passwords, called hashes, are stored in the database. These methods are CRAM-MD5, DHX, Digest-MD5, MS-CHAPv2, SMB-NT, and SMB-LAN Manager.

No one—including an administrator and the root user—can recover encrypted passwords by reading them from the database. An administrator can use Workgroup Manager to set a user's password, but can't read any user's password.

In addition, Open Directory Password Server supports authentication methods that may require a clear text password be stored in the authentication database. These methods are APOP and WebDAV-Digest.

Note: If you connect Mac OS X Server version 10.3 or later to a directory domain of Mac OS X Server version 10.2 or earlier, be aware that users defined in the older directory domain cannot be authenticated with the MS-CHAPv2 method. This method may be required to securely authenticate users for the VPN service of Mac OS X Server version 10.3 and later. Open Directory Password Server in Mac OS X Server version 10.3 supports MS-CHAPv2 authentication, but Password Server in Mac OS X Server version 10.2 does not support MS-CHAPv2.

Contents of Open Directory Password Server Database

Open Directory Password Server maintains an authentication database separate from the Mac OS X Server directory domain. Open Directory tightly restricts access to the authentication database, whereas anyone on the network can access the directory domain.

Open Directory Password Server maintains a record in its authentication database for each user account that has a password type of Open Directory. An authentication record includes the following:

- The user's password ID is a 128-bit value assigned when the password is created. It is also stored in the user's record in the directory domain and is used as a key for finding a user's record in the Open Directory Password Server database.

- The password is stored in recoverable (clear text) or hashed (encrypted) form. The form depends on the authentication method. A recoverable password is stored for the APOP and WebDAV authentication methods. For all other methods, the record stores a hashed (encrypted) password. If no authentication method requiring a clear text password is enabled, the Open Directory authentication database stores only hashes of passwords.
- The user's short name, for use in log messages viewable in Server Admin.
- Password policy data.

Kerberos Authentication

Kerberos is a network authentication protocol developed at MIT to provide secure authentication and communication over open networks. In addition, Kerberos enables single signon authentication (see "Single Signon" on page 39).

Mac OS X and Mac OS X Server versions 10.2 and later support Kerberos v5. If your network already has a Kerberos Key Distribution Center (KDC), you can set up your Mac OS X computers to use it for authentication.

If a server with Mac OS X Server version 10.3 has a shared LDAP directory, the server also has a Kerberos KDC built in. This KDC can authenticate all users whose accounts are stored in a directory domain on the server and whose password type is Open Directory. The built-in KDC requires minimal setup. Computers with Mac OS X version 10.3 and later require minimal setup to use the Mac OS X Server KDC for authentication of user accounts stored on the server.

Kerberized Services

Kerberos can authenticate users for the following services of Mac OS X Server:

- Login window
- Mail service
- FTP
- AFP service
- SSH

These services have been "Kerberized." Only services that have been Kerberized can use Kerberos to authenticate a user.

Kerberos Principals and Realms

Kerberized services are configured to authenticate principals who are known to a particular Kerberos realm. You can think of a realm as a particular Kerberos database or authentication domain, which contains validation data for users, services, and sometimes servers, which are all known as principals. For example, a realm contains principals' secret keys, which are the result of a one-way function applied to passwords. Service principals are generally based on randomly generated secrets rather than passwords.

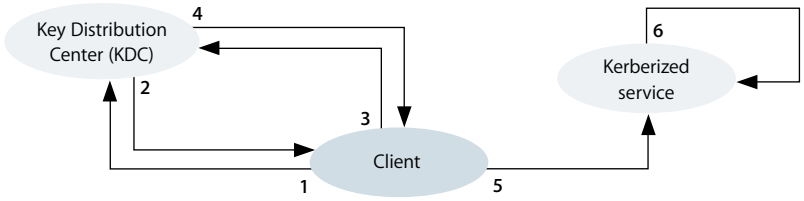
Here are examples of realm and principal names; note that realm names are capitalized by convention to distinguish them from DNS domain names:

- Realm: MYREALM.EXAMPLE.COM
- User principal: smitty@MYREALM.EXAMPLE.COM
- Service principal: afpserver/somehost.example.com@MYREALM.EXAMPLE.COM

Kerberos Authentication Process

There are several phases to Kerberos authentication. In the first phase, the client obtains credentials to be used to request access to Kerberized services. In the second phase, the client requests authentication for a specific service. In the final phase, the client presents those credentials to the service.

The following illustration summarizes these activities. Note that the service and the client in this picture may be the same entity (such as the login window) or two different entities (such as a mail client and the mail server).



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.
2. The KDC issues the client a ticket-granting ticket, the credential needed when the client wants to use Kerberized services. The ticket-granting ticket is good for a configurable period 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 ticket-granting ticket when it wants to use a particular Kerberized service.
4. The KDC issues a ticket for that service.
5. The client presents the ticket to the 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. An AFP server, for example, needs to consult a user's account in a directory domain to obtain the UID.) The service uses information in the ticket if required to retrieve additional information about the user from a directory domain.

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

Time is very important with Kerberos. If the client and the KDC are out of sync by more than a few minutes, the client will fail to achieve authentication with the KDC. The date, time, and time zone information needs to be correct on the KDC server and clients, and they all should use the same network time service to keep their clocks in sync.

For more information on Kerberos, go to the MIT Kerberos website:

web.mit.edu/kerberos/www/index.html

Single Signon

Mac OS X Server uses Kerberos for *single signon* authentication, which relieves users from entering a name and password separately for every Kerberized service. With single signon, a user always enters a name and password in the login window. Thereafter, the user does not have to enter a name and password for Apple file service, mail service, or other services that use Kerberos authentication. To take advantage of the single signon feature, users and services must be configured for Kerberos authentication and use the same Kerberos Key Distribution Center (KDC) server.

User accounts that reside in an LDAP directory of Mac OS X Server version 10.3 and have a password type of Open Directory use the server's built-in KDC. These user accounts are automatically configured for Kerberos and single signon. This server's Kerberized services also use the server's built-in KDC and are automatically configured for single signon. Other servers require some configuration to use the Mac OS X Server KDC. Servers with Mac OS X Server version 10.3 require only minimal configuration to use the built-in KDC of another server with Mac OS X Server version 10.3.

Shadow and Crypt Passwords

Shadow and crypt passwords do not depend on the Kerberos or Open Directory Password Server infrastructure for password validation. Both transmit a scrambled form of a user's password, or hash, when sending the password over the network. Both also store a scrambled form of the password, but they differ in where the password is stored.

A crypt password is stored as a hash in the user account, making the password fairly easy to capture from another computer on the network. This strategy, historically called basic authentication, is most compatible with software that needs to access user records directly. For example, Mac OS X version 10.1 and earlier expects to find a crypt password stored in the user account.

A shadow password is stored as several hashes in a file on the same computer as the directory domain where the user account resides. Because the password is not stored in the user account, the password is not easy to capture over the network. Each user's shadow password is stored in a different file, called a shadow password file, and these files are protected so they can be read only by the root user account. Only user accounts that are stored in a computer's local directory can have a shadow password. User accounts that are stored in a shared directory can't have a shadow password.

A shadow password's primary hash function is SHA-1. In addition, NT and LAN Manager hashes are stored in the shadow password file for backwards compatibility with Windows SMB file and print services. The NT and LAN Manager hashes can be used for Windows personal file sharing from a Mac OS X computer and can also be used to authenticate Windows file and print services provided by Mac OS X Server.

Shadow passwords also provide cached authentication for mobile user accounts.

Shadow and crypt do not support all services. Some services require the authentication methods that Open Directory supports, such as APOP, CRAM-MD5, Digest-MD5, MS-CHAPv2, and WebDAV-Digest. For secure transmission of passwords over a network, crypt supports the DHX authentication method. Shadow additionally supports NT and LAN Manager for network-secure authentication.

Crypt authentication only supports a maximum password length of eight bytes (eight ASCII characters). If a longer password is entered in a user account, only the first eight bytes are used for crypt password validation.

The eight-character limit does not apply to a shadow password. With a shadow password, the first 128 characters are used for NT authentication, and the first 14 characters are used for LAN Manager authentication.

Encrypting Shadow and Crypt Passwords in User Accounts

Shadow and crypt passwords are not stored in clear text; they are concealed and made illegible by encryption.

Shadow and crypt encrypt a password by feeding the clear text password along with a random number to a mathematical function, known as a one-way hash function. A one-way hash function always generates the same encrypted value from particular input, but cannot be used to recreate the original password from the encrypted output it generates.

To validate a password using the encrypted value, Mac OS X applies the function to the password entered by the user and compares it with the value stored in the user account. If the values match, the password is considered valid.

Different hash functions are used to encrypt shadow and crypt passwords. For crypt passwords, the standard UNIX crypt function is used. Shadow passwords are encrypted using several hash functions, including NT and LAN Manager.

Cracking Readable Passwords

Because crypt passwords are stored directly in user accounts, they are potentially subject to cracking. User accounts in a shared directory domain are openly accessible on the network. Anyone on the network who has Workgroup Manager or knows how to use command-line tools can read the contents of user accounts, including the passwords stored in them.

A malicious user, or cracker, could use Workgroup Manager or UNIX commands to copy user records to a file. The cracker can transport this file to a system and use various techniques to figure out which unencrypted passwords generate the encrypted passwords stored in the user records. After identifying a password, the cracker can log in unnoticed with a legitimate user name and password.

This form of attack is known as an offline attack, since it does not require successive login attempts to gain access to a system.

A very effective way to thwart password cracking is to use good passwords. A password should contain letters, numbers, and symbols in combinations that won't be easily guessed by unauthorized users. Passwords should not consist of actual words. Good passwords might include digits and symbols (such as # or $). Or they might consist of the first letter of all the words in a particular phrase. Use both uppercase and lowercase letters.

Note: Shadow and Open Directory passwords are far less susceptible to offline attack because they are not stored in user records. Shadow passwords are stored in separate files that can be read only by someone who knows the password of the root user (also known as the System Administrator). Open Directory passwords are stored securely in the Kerberos KDC and in the Open Directory Password Server database. A user's Open Directory password can't be read by other users, not even by a user with administrator rights for Open Directory authentication. (This administrator can only change Open Directory passwords and password policies.)

LDAP Bind Authentication

For user accounts that reside in an LDAP directory on a non-Apple server, Open Directory attempts to use simple LDAP bind authentication. Open Directory sends the LDAP directory server the name and password supplied by the authenticating user. If the LDAP server finds a matching user record and password, authentication succeeds.

Simple LDAP bind authentication is inherently insecure because it transmits clear text passwords over the network. But you can secure simple LDAP bind authentication by setting up access to the LDAP directory via the Secure Sockets Layer (SSL) protocol. SSL makes access secure by encrypting all communications with the LDAP directory.

Open Directory Planning

4

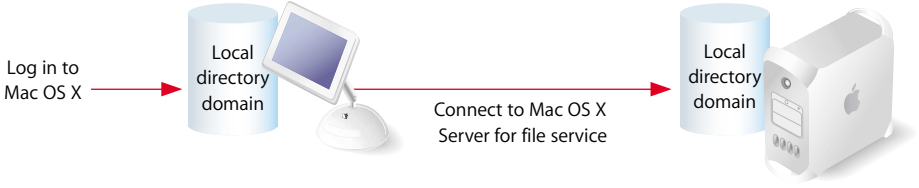
Like the plumbing and wiring in a building, directory services for a network must be planned in advance, not on an ad hoc basis.

Keeping information in shared directory domains gives you more control over your network, allows more users access to the information, and makes maintaining the information easier for you. But the amount of control and convenience depends on the effort you put into planning your shared domains. The goal of directory domain planning is to design the simplest arrangement of shared domains that gives your Mac OS X users easy access to the network resources they need *and* minimizes the time you spend maintaining user records and other administrative data.

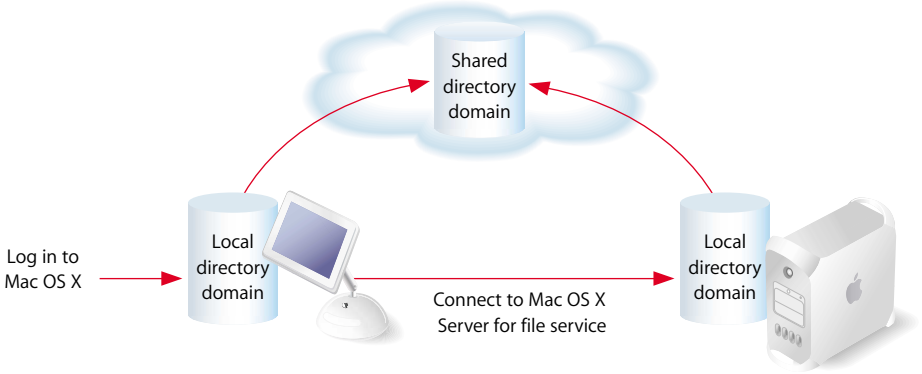
This chapter presents guidelines for planning Open Directory services and describes tools for managing them.

General Planning Guidelines

If you do not need to share user and resource information among multiple Mac OS X computers, there is very little directory domain planning necessary. Everything can be accessed from local directory domains. Just ensure that all individuals who need to use a particular Mac OS X computer are defined as users in the local directory domain on the computer.

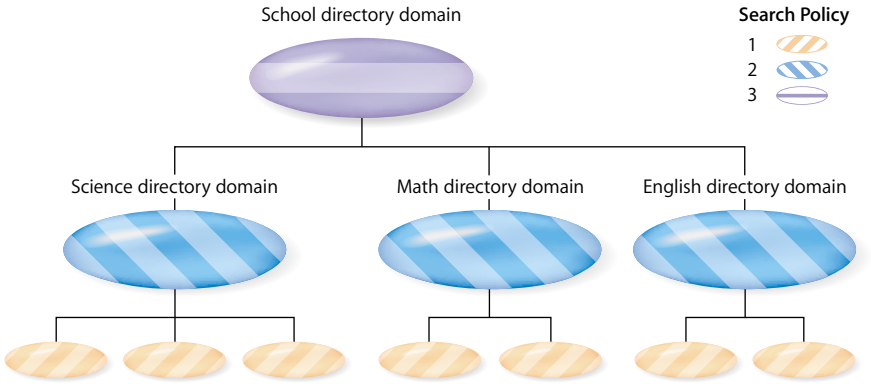


If you want to share information among Mac OS X computers, you need to set up at least one shared directory domain.



A single shared directory domain may be completely adequate if all your network computer users share the same resources, such as printers, share points for home directories, share points for applications, and share points for documents.

Larger, more complex organizations can benefit from additional shared directory domains.



Controlling Data Accessibility

If your network has several shared directory domains, you can make directory information visible only to subsets of a network's computers. In the foregoing example arrangement, the administrator can tailor the users and resources visible to the community of Mac OS X computers by distributing directory information among four shared directory domains.

If you want all computers to have access to certain administrative data, you store the data in a shared directory domain that is in all computers' search policies. To make some data accessible only to a subset of computers, you store it in a shared directory domain that is only in the search policies of those computers.

Simplifying Changes to Data in Directories

If you need more than one shared directory domain, you should organize your search policies to minimize the number of places data has to change over time. You should also devise a plan that addresses how you want to manage such ongoing events as:

- New users joining and leaving your organization
- File servers being added, enhanced, or replaced
- Printers being moved among locations

You'll want to try to make each directory domain applicable to all the computers that use it so you don't have to change or add information in multiple domains. In the foregoing illustration of multilevel shared domains, adding a new student to a class's shared domain enables the student to log in to any of the class's computers. As instructors are hired or retire, the administrator can make adjustments to user information simply by editing the school's shared domain.

If you have a widespread or complex hierarchy of directory domains in a network that is managed by several administrators, you need to devise strategies to minimize conflicts. For example, you can predefine ranges of user IDs (UIDs) to avoid inadvertent file access. (For more information, see the chapter on setting up accounts in the user management guide.)

Estimating Directory and Authentication Requirements

In addition to considering how you want to distribute directory data among multiple domains, you must also consider the capacity of each directory domain. A number of factors affect how large a directory domain can be. One factor is the performance of the database that stores the directory information. The LDAP directory domain of Mac OS X Server version 10.3 and later uses the Berkeley DB database, which will remain efficient with 100,000 records. Of course, a server hosting a directory domain of that size would need sufficient hard disk space to store all the records.

The number of connections that a directory service can handle is harder to measure because directory service connections occur in the context of the connections of all the services that the server provides. With Mac OS X Server version 10.3, a server dedicated to Open Directory has a limit of 250 simultaneous client computer connections.

The Open Directory server may actually be able to provide LDAP and authentication services to more client computers, because all the client computers will not need these services at once. Each client computer connects to the LDAP directory for up to two minutes, and connections to the Open Directory Password Server are even shorter lived. For example, an Open Directory server may be able to support 750 client computers because the odds are that only a fraction of the client computers that could make a connection with Open Directory will actually make connections at the same time. Determining what the fraction is—what percentage of the potential client computers will make connections at the same time—can be difficult. For example, client computers that each have a single user who spends all day working on graphics files will need Open Directory services relatively infrequently. In contrast, computers in a lab will have many users logging in throughout the day, each with a different set of managed client preference settings, and these computers will place a relatively high load on Open Directory services.

In general, you can correlate Open Directory usage with login and logout. These activities will generally dominate directory and authentication services in any system. The more frequently users log in and out, the fewer client computers an Open Directory server (or any directory and authentication server) can support. You need more Open Directory servers if users log in very frequently. You can get by with fewer Open Directory servers if work sessions are long duration and login is infrequent.

Identifying Servers for Hosting Shared Domains

If you need more than one shared domain, you need to identify the servers on which shared domains should reside. Shared domains affect many users, so they should reside on Mac OS X Server computers that have the following characteristics:

- Restricted physical access
- Limited network access
- Equipped with high-availability technologies, such as uninterruptible power supplies

You should select computers that will not be replaced frequently and that have adequate capacity for growing directory domains. While you can move a shared domain after it has been set up, you may need to reconfigure the search policies of computers that bind to the shared domain so that their users can continue to log in.

Replicating Open Directory Services

Mac OS X Server supports replication of the LDAP directory service, the Open Directory Password Server, and the Kerberos KDC.

By replicating your directory and authentication services you can:

- Move directory information closer to a population of users in a geographically distributed network, improving performance of directory and authentication services to these users.
- Achieve redundancy, so that users see little disruption in service if a directory system fails or becomes unreachable.

One server has a primary copy of the shared LDAP directory domain, Open Directory Password Server, and Kerberos Key Distribution Center (KDC). This server is called an Open Directory master. Each Open Directory replica is a separate server with a copy of the master's LDAP directory, Open Directory Password Server, and Kerberos KDC.

Access to the LDAP directory on a replica is read only. All changes to user records and other account information in the LDAP directory can be made only on the Open DIrectory master.

The Open Directory master automatically updates its replicas with changes to the LDAP directory. The master can update the replicas every time a change occurs, or you can set up a schedule so that updates occur only at regular intervals. The fixed schedule option is best if replicas are connected to the master by a slow network link.

Passwords and password policies can be changed on any replica. If a user's password or password policy is changed on more than one replica, the most recent change prevails.

The updating of replicas relies on the clocks of the master and all replicas being in sync. If replicas and the master have a wildly different notion of time, updating could be somewhat arbitrary. The date, time, and time zone information needs to be correct on the master and replicas, and they all should use the same network time service to keep their clocks in sync.

Replication in a Multi-Building Campus

A network that spans multiple buildings may have slower network links between buildings than within each building. The network links between buildings may also be overloaded. These conditions can adversely affect the performance of computers that get Open Directory services from a server in another building. Accordingly, you may want to set up an Open Directory replica in each building. Depending on need, you may even want to set up an Open Directory replica on each floor of a multistory building. Each replica provides efficient directory and authentication services to client computers in its vicinity. The client computers do not have to make connections with an Open Directory server across the slow, crowded network link between buildings.

Having more replicas does have a disadvantage. Replicas communicate with each other and with the master over the network. This network communication overhead increases as you add replicas. Adding too many replicas can actually add more network traffic between buildings in the form of replication updates than it removes in the form of Open Directory client communications.

Therefore in deciding how many replicas to deploy, you must consider how heavily the client computers will use Open Directory services. If the client computers are relatively light users of Open Directory services on average and your buildings are connected by fairly fast network links (such as 100 Mbit/s Ethernet), you probably do not need a replica in each building.

You can reduce the communication overhead between Open Directory replicas and the master by scheduling how often the Open Directory master updates the replicas. You might not need the replicas updated every time a change occurs in the master. Scheduling less frequent updates of replicas will improve performance of the network.

Improving Performance and Redundancy

You can improve the performance of Open Directory services by adding more memory to the server and having it provide fewer services. This strategy applies to every other service of Mac OS X Server as well. The more you can dedicate an individual server to a particular task, the better its performance will be.

Beyond that general strategy, you can also improve Open Directory server performance by directing the LDAP database to its own disk volume and the Open Directory logs to another disk volume.

If your network will include replicas of an Open Directory master, you can improve performance of the network by scheduling less frequent updates of replicas. Updating less frequently means the replicas have less up-to-date directory data. You have to strike a balance between higher network performance and less accuracy in your replicas.

For greater redundancy of Open Directory services, you can set up additional servers as Open Directory replicas. Another strategy for increasing redundancy is to use servers with RAID sets for Open Directory services.

Open Directory Security

With Mac OS X Server version 10.3, a server that has a shared LDAP directory domain also provides Open Directory authentication. The authentication data stored by Open Directory is particularly sensitive. This authentication data includes the Open Directory Password Server database and the Kerberos database, which is extraordinarily sensitive. Therefore you need to make sure that an Open Directory master and all Open Directory replicas are secure:

- Physical security of a server that is an Open Directory master or replica is paramount. It should be behind a locked door. It should always be left logged out.
- Secure the media you use to back up an Open Directory Password Server database and a Kerberos database. Having your Open Directory servers behind locked doors won't protect a backup tape that you leave on your desk every night.
- If possible, do not use a server that is an Open Directory master or replica to provide any other services. If you can't dedicate servers to be Open Directory master and replicas, at least minimize the number of other services they provide. One of the other services could have a security breach that allows someone inadvertent access to the Kerberos or Open Directory Password Server databases. Dedicating servers to providing Open Directory services is an optimal practice but not required.
- Avoid using a RAID volume that's shared with other computers as the startup volume of a server that is an Open Directory master or replica. A security breach on one of the other computers could jeopardize the security of the Open Directory authentication information.
- Set up IP firewall service to block all ports except ports used for directory, authentication, and administration protocols.
  - Open Directory Password Server uses ports 106 and 3659.
  - The Kerberos KDC uses TCP/UDP port 88, and TCP/UDP port 749 is used for Kerberos administration.
  - The shared LDAP directory uses TCP port 389 for an ordinary connection and TCP port 636 for an SSL connection.
  - Workgroup Manager uses TCP port 311 and 625.
  - Server Admin uses TCP port 311.
  - SMB uses TCP/UDP ports 137, 138, 139, and 445.
- Equip the Open Directory master computer with an uninterruptible power supply.

In summary, the most secure and best practice is to dedicate each server that is an Open Directory master or replica to provide only Open Directory services. Set up a firewall on each of these servers to allow only directory access, authentication, and administration protocols: LDAP, Password Server, Kerberos, Workgroup Manager, and Server Manager. Physically secure each Open Directory server and all backup media used with it.

Replication introduces a minimal increase in security risk. The replicated LDAP directory data has no access controls to restrict reading it, so anyone on the network can download the entire directory irrespective of replication. The password data is securely replicated using random keys negotiated during each replication session. The authentication portion of replication traffic—the Open Directory Password Server and the Kerberos KDC—is fully encrypted. For extra security, you could configure network connections between the Open Directory servers to use network switches rather than hubs. This configuration would isolate authentication replication traffic to trusted network segments.

Tools for Managing Open Directory Services

The Server Admin, Directory Access, and Workgroup Manager applications provide a graphical interface for managing Open Directory services in Mac OS X Server. In addition, you can manage Open Directory services from the command line by using Terminal. If your network includes legacy NetInfo domains, you can manage them with the Inspector in Workgroup Manager. (You could also use NetInfo Manager.)

All these applications are included with Mac OS X Server and can be installed on another computer with Mac OS X version 10.3 or later, making that computer an administrator computer. For more information on setting up an administrator computer, see the server administration chapter of the getting started guide.

Server Admin

You use Server Admin to:

- Set up Mac OS X Server as an Open Directory master, an Open Directory replica, a server that's connected to a directory system, or a standalone server with only a local directory. For instructions, see Chapter 5, "Setting Up Open Directory Services."
- Set up additional Mac OS X Server systems to use the Kerberos KDC of an Open Directory master or replica. For instructions, see Chapter 5.
- Migrate an upgraded server's shared directory domain from NetInfo to LDAP. For instructions, see Chapter 5.
- Configure LDAP options on an Open Directory master. For instructions, see Chapter 5.
- Configure DHCP service to supply an LDAP server address to Mac OS X computers with automatic search policies. For instructions, see the DHCP chapter of the network services administration guide.
- Set up password policies that apply to all users who don't have overriding individual password policies. For instructions, see Chapter 6, "Managing User Authentication." (To set up individual password policies, use Workgroup Manager; see Chapter 6.)
- Monitor Open Directory services. For instructions, see Chapter 8, "Maintenance and Problem Solving."

For basic information about using Server Admin, see the chapter on server administration in the getting started guide.

Server Admin is installed in /Applications/Server/.

Directory Access

You use Directory Access to:

- Enable or disable kinds of directory services and kinds of network service discovery on a Mac OS X computer.
- Define authentication and contacts search policies for a Mac OS X computer.
- Configure connections to LDAP directories, an Active Directory domain, an NIS domain, and NetInfo domains.
- Configure data mapping for LDAP directories.

Directory Access can connect to other computers on your network so you can configure them remotely.

For instructions on using Directory Access, see Chapter 7, "Managing Directory Access."

Directory Access is installed on every Mac OS X computer in /Applications/Utilities/.

Workgroup Manager

You use Workgroup Manager to:

- Set up and manage user, group, and computer accounts. For instructions, see the chapters on user, group, and computer accounts in the user management guide and the Windows services administration guide.
- Manage share points for file service and for user home directories and roaming user profiles. For instructions, see the chapter on share points in the file services administration guide and the chapter on managing Windows services in the Windows services administration guide.
- Access the Inspector, which lets you work with all Open Directory records and attributes. For instructions, see Chapter 8, "Maintenance and Problem Solving."

For basic information about using Workgroup Manager, see the chapter on server administration in the getting started guide.

Workgroup Manager is installed in /Applications/Server/.

Command-Line Tools

A full range of command-line tools are available for administrators who prefer to use command-driven server administration. For remote server management, submit commands in a Secure Shell (SSH) session. You can type commands on Mac OS X servers and computers by using the Terminal application, located in /Applications/Utilities/. For instructions, see the command-line administration guide.

NetInfo Manager

You use NetInfo Manger to view and change records, attributes, and values in legacy NetInfo domains on computers that still use or have been upgraded from Mac OS X Server version 10.2 or earlier. You can do these same tasks by using the Inspector in Workgroup Manager. You can also use NetInfo Manager to manage a legacy NetInfo hierarchy and back up and restore a legacy NetInfo domain.

NetInfo Manager is located in /Applications/Utilities/.

Setting Up Open Directory Services

5

You can use Server Admin to set up the Open Directory role of a server, set up single signon and Kerberos authentication services, configure LDAP options, and migrate from NetInfo to LDAP.

Open Directory services—directory services and authentication services—are an essential part of a network's infrastructure. These services have a significant effect on other network services and on users. Therefore Open Directory must be set up correctly from the beginning.

Setup Overview

Here is a summary of the major tasks you perform to set up Open Directory services. See the pages indicated for detailed information about each step.

Step 1: Before you begin, do some planning
See "Before You Begin" on page 54 for a list of items to think about before you configure Open Directory on Mac OS X Server.

Step 2: Set up your Open Directory master
See "Setting Up an Open Directory Master" on page 56 and "Setting LDAP Options" on page 63.

Step 3: Set up replicas of your Open Directory master
See "Setting Up an Open Directory Replica" on page 57 and "Setting LDAP Options" on page 63.

Step 4: Set up servers that connect to other directory systems
See "Setting Up a Connection to a Directory System" on page 60.

Step 5: Set up single signon and Kerberos authentication
See "Setting Up Single Signon and Kerberos" on page 61.

Step 6: Migrate upgraded servers from NetInfo to LDAP
See "Migrating a Directory Domain From Netinfo to LDAP" on page 66 and "Disabling NetInfo After Migrating to LDAP" on page 69.

Step 7: Set up Directory Access on servers and client computers
See Chapter 7, "Managing Directory Access."

Before You Begin

Before setting up Open Directory services for the first time:

- Understand the uses of directory data and assess your directory needs.
  Identify the services that require data from directory domains, and determine which users will need access to those services.

  Users whose information can be managed most easily on a server should be defined in the shared LDAP directory of a Mac OS X Server that is an Open Directory master. Some of these users may instead be defined in directory domains on other servers, such as an Active Directory domain on a Windows server.

  These concepts are discussed in Chapter 1, "Directory Service Concepts."
- Assess whether you need more than one shared domain. If so, decide which users will be defined in each shared domain. See Chapter 2, "Open Directory Search Policies," for more information.
- Determine which authentication options users need. For descriptions of the available options, see Chapter 3, "User Authentication With Open Directory."
- Decide how to organize your directory domains, including replicas of Open Directory masters. Chapter 4, "Open Directory Planning," provides some guidelines.
- Pick server administrators very carefully. Give only trusted people administrator passwords. Have as few administrators as possible. Don't delegate administrator access for minor tasks, such as changing settings in a user record.

Important: Directory information is authoritative. It vitally affects everyone whose computers use it.

Setting Up Open Directory With Server Assistant

The initial setup of Open Directory occurs when you use Server Assistant during installation of Mac OS X Server. For instructions on using Server Assistant, see the getting started guide.

Managing Open Directory on a Remote Server

You can install Server Admin on a computer with Mac OS X version 10.3 or later and use it to manage Open Directory on any server locally or remotely. You can also manage Open Directory remotely by using command-line tools from a Mac OS X computer or a non-Macintosh computer. For more information, see the server administration chapter of the getting started guide.

Setting Up a Standalone Server

Using Server Admin, you can set up Mac OS X Server to use only the server's local directory domain. The server does not provide directory information to other computers or get directory information from an existing system. (The local directory domain cannot be shared.)

Important: If you change Mac OS X Server to get directory information only from its local directory domain, then user records and other information that the server formerly retrieved from a shared directory domain will become unavailable:

- The user records and other information will still exist in the shared directory domain but will become unavailable to the server's users and services.
- Files and folders on the server may become unavailable to users whose accounts are in the shared directory domain.
- If the server was an Open Directory master and other servers were connected to it:
  - Services may be disrupted on the connected servers when the user accounts and other information in the shared directory domain become unavailable.
  - Users whose accounts are in the shared directory domain may no longer be able to access files and folders on the Open Directory master and on other servers that were connected to its shared LDAP directory domain.

You can back up the LDAP directory and Open Directory Password Server database before changing from Open Directory master to standalone server. For instructions, see "Backing Up Open Directory Files" on page 118.

To configure a server to use only its own nonshared local directory domain:

1. Open Server Admin and select Open Directory for a server in the Computers & Services list.
2. Click Settings (near the bottom of the window), then click General (near the top).
3. Choose Standalone Server from the Role pop-up menu.
4. If you are sure that users and services no longer need access to the directory data stored in the shared directory domain that the server has been hosting or connected to, click Save.

Setting Up an Open Directory Master

Using Server Admin, you can set up Mac OS X Server to be an Open Directory master so it can provide directory information and authentication information to other systems. Mac OS X Server provides directory information by hosting a shared LDAP directory domain. In addition, the server authenticates users whose accounts are stored in the shared LDAP directory domain.

Important: If you change a Mac OS X Server computer that was connected to another directory system to be an Open Directory master instead, the server remains connected to the other directory system. The server will search for user records and other information in its shared LDAP directory domain before searching in other directory systems to which it is connected.

To configure a server to host a shared LDAP domain:

1. Open Server Admin and select Open Directory for a server in the Computers & Services list.

   A server must have Mac OS X Server version 10.3 or later to be an Open Directory master.

2. Click Settings (near the bottom of the window), then click General (near the top).
3. Choose Open Directory Master from the Role pop-up menu and enter the requested information.

   Administrator short name: The short name of an administrator account in the server's local directory domain that you want to have copied to the new shared LDAP directory. This account will be an administrator of the LDAP directory domain.

   Administrator password: The password for the administrator account whose short name you entered.

   Kerberos realm name: By convention, the Kerberos realm name is the same as the server's DNS name but in all uppercase letters. For example, a server whose DNS name is example.com would have a Kerberos realm name of EXAMPLE.COM.

   Search base (optional): The search base suffix for the new LDAP directory. Typically, the search base suffix is derived from the server's DNS name. For example, the search base suffix could be "dc=example, dc=com" for a server whose DNS name is server.example.com.

4. Click OK, then click Save.

After setting up a Mac OS X Server computer to be an Open Directory master, you can configure other computers with Mac OS X or Mac OS X Server to access the server's shared LDAP directory domain:

- You can configure DHCP service to supply the Open Directory master as an LDAP server to computers with automatic search policies. Computers with Mac OS X or Mac OS X Server version 10.2 can have automatic search policies. These computers don't have to be configured individually to access the LDAP server. When these computers start up, they try to get the address of an LDAP server from DHCP service.
- You can configure a computer to access the server's LDAP directory and then add the server's LDAP directory to the computer's custom search policy.

For instructions on configuring DHCP to supply an LDAP server's address, see the network services administration guide. For instructions on setting up search policies and configuring access to specific LDAP directory domains, see Chapter 7, "Managing Directory Access."

Setting Up an Open Directory Replica

Using Server Admin, you can set up Mac OS X Server to be a replica of an Open Directory master so it can provide the same directory information and authentication information to other systems as the master. The replica server hosts a read-only copy of the master's LDAP directory domain. The replica server also hosts a read/write copy of the authentication database associated with the master directory domain and the Kerberos Key Distribution Center (KDC).

Open Directory replicas can provide these benefits:

- In a wide area network (WAN) of local area networks (LANs) interconnected by slow links, replicas on the LANs can provide servers and client computers with fast access to user accounts and other directory information.
- A replica provides redundancy. If the Open Directory master fails, computers connected to it automatically switch to a nearby replica. This automatic failover behavior is a feature of version 10.3 and later of Mac OS X and Mac OS X Server.

Important: When you set up an Open Directory replica, all the directory and authentication data must be copied to it from the Open Directory master. Replication may take several seconds or several minutes depending on the size of the directory domain. Replication over a slow network link can take a very long time. During replication, the master cannot provide directory or authentication services. User accounts in the master LDAP directory can't be used to log in or authenticate for services until replication is finished. To minimize the disruption of directory service, set up a replica before the master LDAP directory is fully populated or at a time of day when the directory service is not needed. Having another replica already set up will insulate clients of directory service from the master being unavailable.

Important: If you change a Mac OS X Server computer that was connected to another directory system to be an Open Directory replica instead, the server remains connected to the other directory system. The server will search for user records and other information in its shared LDAP directory domain before searching in other directory systems to which it is connected.

To configure a server to host a replica of an Open Directory master:

1. Open Server Admin and select Open Directory for a server in the Computers & Services list.

   A server must have Mac OS X Server version 10.3 or later to be an Open Directory replica.

2. Click Settings (near the bottom of the window), then click General (near the top).
3. Choose Open Directory Replica from the Role pop-up menu and enter the requested information.

   IP address of LDAP master: Enter the IP address of the server that is the Open Directory master.

   root's password on LDAP master: Enter the password of the Open Directory master system's root user (user name System Administrator).

   Password Server admin's name on replica: Enter the name of an administrator account whose password type is Open Directory.

   Password Server admin's password on replica: Enter the password of the administrator account whose name you entered.

4. Click OK, then click Save.
5. Make sure the date, time, and time zone are correct on the replica and the master.

   The replica and the master should use the same network time service so their clocks remain in sync.

After you set up an Open Directory replica, other computers will connect to it automatically as needed. Computers with version 10.3 and later of Mac OS X and Mac OS X Server maintain a list of all replicas of an Open Directory master to which they are connected. If one of these computers can't contact the Open Directory master for directory and authentication services, the computer automatically connects to the nearest replica of the master.

You can configure Mac OS X computers to connect to an Open Directory replica instead of the Open Directory master for directory and authentication services. On each Mac OS X computer, you can use Directory Access to create an LDAPv3 configuration for accessing the replica's LDAP directory and set up a custom search policy that includes this LDAPv3 configuration. You can also configure a DHCP service to supply the replica's LDAP directory to Mac OS X computers that get the address of an LDAP server from the DHCP service. See "Accessing LDAP Directories" on page 90 and "Defining Automatic Search Policies" on page 88. See the network services administration guide for instructions on setting up DHCP service to supply an LDAP server's address.

The Open Directory master automatically updates the replica. You can configure the master to update its replicas at a specific interval or whenever the master directory changes. For instructions, see "Setting the Replication Frequency of an Open Directory Master" on page 64.

Setting Up Open Directory Failover

If an Open Directory master or any of its replicas become unavailable, its client computers with Mac OS X version 10.3 or Mac OS X Server version 10.3 will automatically find an available replica and connect to it.

Replicas only allow clients to read directory information. Directory information on a replica can't be modified with administration tools such as Workgroup Manager.

Users whose password type is Open Directory can change their passwords on computers that are connected to Open Directory replicas. The replicas automatically synchronize password changes with the master. If the master is unavailable for a while, the replicas synchronize password changes with the master when it becomes available again.

If an Open Directory master or replica becomes unavailable and it has client computers with version 10.2 or earlier of Mac OS X or Mac OS X Server, these client computers must be reconfigured manually to connect to an available replica. You can use Directory Access to create an LDAPv3 configuration that specifies how the computer accesses an available replica. For instructions, see "Accessing LDAP Directories" on page 90.

Setting Up a Connection to a Directory System

Using Server Admin, you can set up Mac OS X Server to get user records and other directory information from another server's shared directory domain. The other server also provides authentication for its directory information. Mac OS X Server will still get directory information from its own local directory domain and will provide authentication for this directory information.

Important: Changing Mac OS X Server to be connected to another directory system instead of being an Open Directory master will deactivate its shared LDAP directory domain, with the following ramifications:

- User records and other directory information will still exist in the deactivated directory domain but will be unavailable to the server's users and services.
- If other servers were connected to the master directory domain, their services may be disrupted when the user accounts and other information in the deactivated directory domain become unavailable.
- Users who had accounts in the deactivated directory domain may no longer be able to access files and folders on the Open Directory master and on other servers that were connected to the master directory domain.

To configure a server to get directory services from an existing system:

1. Open Server Admin and select Open Directory for a server in the Computers & Services list.
2. Click Settings (near the bottom of the window), then click General (near the top).
3. Choose "Connected to a Directory System" from the Role pop-up menu.
4. If the server was an Open Directory master and you are sure that users and services no longer need access to the directory data stored in the shared directory domain that the server has been hosting, click Save.
5. Click the Open Directory Access button to configure access to one or more directory systems.

   For instructions on configuring access to a particular kind of directory service, see Chapter 7, "Managing Directory Access."

   Note: If you connect Mac OS X Server version 10.3 or later to a directory domain of Mac OS X Server version 10.2 or earlier, be aware that users defined in the older directory domain cannot be authenticated with the MS-CHAPv2 method. This method may be required to securely authenticate users for the VPN service of Mac OS X Server version 10.3 and later. Open Directory in Mac OS X Server version 10.3 supports MS-CHAPv2 authentication, but Password Server in Mac OS X Server version 10.2 does not support MS-CHAPv2.

Setting Up Single Signon and Kerberos

Setting up single signon and Kerberos authentication involves these tasks:

- An administrator who has authority to manage directory domains sets up a server as an Open Directory master, which hosts a Kerberos Key Distribution Center (KDC). See "Setting Up an Open Directory Master for Single Signon and Kerberos" on page 61.
- The network administrator delegates to specific server administrators the authority to join their servers to the Open Directory master server for single signon and Kerberos authentication. (If you want to set up a server to join an Open Directory master for single signon and Kerberos, you must delegate authority to yourself.) See "Delegating Authority to Join an Open Directory Master for Single Signon and Kerberos" on page 62.
- Delegated administrators join their servers to the Open Directory master, which then provides single signon and Kerberos authentication for services provided by the servers that have joined. See "Joining a Server to an Open Directory Master for Single Signon and Kerberos" on page 63.
- All computers using single signon and Kerberos should be set to the correct date, time, and time zone. They should all be configured to use the same network time server. Kerberos depends on the clocks of all participating computers being in sync.
- DNS must be available on the network.

The individual services of Mac OS X Server version 10.3 and later do not require any configuration for single signon or Kerberos. The following services are ready for Kerberos and single signon on every server with Mac OS X Server version 10.3 and later that is an Open Directory master or has joined one:

- Login window
- Mail service
- FTP
- AFP service
- SSH

These services are "Kerberized" whether they are running or not.

Setting Up an Open Directory Master for Single Signon and Kerberos

You can provide single signon and Kerberos authentication on your network by setting up an Open Directory master. You can set up an Open Directory master during the initial configuration that follows installation of Mac OS X Server version 10.3 and later. If you have set up Mac OS X Server to have a different Open Directory role, you can change its role to that of Open Directory master by using Server Admin.

A server that is an Open DIrectory master requires no additional configuration to support single signon and Kerberos authentication for all the Kerberized services that the server itself provides. This server can also support single signon and Kerberos authentication for Kerberized services of other servers on the network. The other servers must be set up to join the Open Directory master for single signon and Kerberos.

For instructions, see the getting started guide, "Setting Up an Open Directory Master" on page 56, "Delegating Authority to Join an Open Directory Master for Single Signon and Kerberos" on page 62, and "Joining a Server to an Open Directory Master for Single Signon and Kerberos" on page 63.

Delegating Authority to Join an Open Directory Master for Single Signon and Kerberos

Using Server Admin, you can delegate the authority to join a server to an Open Directory master for single signon and Kerberos authentication. You can delegate authority to one or more user accounts on one server. The user accounts to which you are delegating authority must have a password type of Open Directory and must reside in the LDAP directory of the Open Directory master. The server for which you are delegating authority must have Mac OS X Server version 10.3 or later.

If you want to delegate authority for more than one server, repeat this procedure for each one.

Important: If a delegated administrator's account is deleted and recreated on the target server, the new account will not have authority to join the Kerberos server. As a precaution, you should delegate authority to at least two accounts on the target server. One account can belong to a network administrator (an administrator of the Kerberos domain).

To delegate authority to join an Open Directory master for single signon and Kerberos:

1. Open Workgroup Manager, make sure the target server has been added to a computer account in the LDAP directory domain of the server from which you're delegating authority, and note the name of the target server in the computer account.
2. The name of the target server in the computer account corresponds to the name of the server's computer record in the LDAP directory domain. Adding the server to a computer account creates a computer record for the server. For instructions on adding the server to a computer account, see the computer accounts chapter of the user management guide. Open Server Admin and select Open Directory for the Open Directory master server in the Computers & Services list.
3. Click Settings (near the bottom of the window), then click General (near the top).
4. Confirm that the Role is Open Directory Master, then click Add Kerberos Record and enter the requested information.

Administrator Name: Enter the name of an LDAP directory administrator on the Open Directory master server.

Administrator Password: Enter the password of the administrator account you entered.

Configuration Record Name: Enter the computer record name of the server for which you are delegating authority to join Kerberos. The server's computer record name is the same as the server's name in a computer account.

Delegated Administrators: Enter a short name or a long name for each user account to which you want to delegate authority. Separate multiple names by pressing Return after each name.

Joining a Server to an Open Directory Master for Single Signon and Kerberos

Using Server Admin, a server administrator whose user account has the properly delegated authority can join a server to an Open Directory master for single signon and Kerberos authentication. This authority must be delegated in advance by an administrator of the Open Directory master.

To join a server to an Open Directory master for single signon and Kerberos:

1. Open Server Admin and select Open Directory for the target server in the Computers & Services list.
2. Click Settings (near the bottom of the window), then click General (near the top).
3. Confirm that the Role is Connected to a Directory System, then click Join Kerberos and enter the name and password of a user account that has been delegated authority for the target server.

Setting LDAP Options

You can set several options for LDAP directories of an Open Directory master or replica. See the following:

- "Setting the Replication Frequency of an Open Directory Master" (next)
- "Changing the Location of an LDAP Database" on page 64
- "Limiting Search Results for LDAP Service" on page 65
- "Changing the Search Timeout for LDAP Service" on page 65
- "Setting up SSL for LDAP Service" on page 65

Setting the Replication Frequency of an Open Directory Master

Using Server Admin, you can specify how frequently an Open Directory master will update its replicas with changes to directory and authentication information. The master can update the replicas whenever a change occurs in the master directory domain or on a schedule you specify.

To specify how frequently an Open Directory master updates its replicas:

1. Open Server Admin and select Open Directory for an Open Directory master server in the Computers & Services list.
2. Click Settings (near the bottom of the window), then click General (near the top).
3. Specify a replication frequency.

   "Replicate to clients whenever the directory is modified": Keeps replicas accurate, but increases network load. May impair the performance of the master if a replica is connected via a slow network link.

   "Replicate to clients every __": Allows you to schedule less frequent updates (by specifying a longer interval). Less frequent updates trades less accuracy of replicas for fewer network connections between the master and its replicas. Fewer network connections may be desirable if replicas are not all on the same LAN as the master.
4. Click Save.

Changing the Location of an LDAP Database

Using Server Admin, you can specify the disk location of the database that stores the user records and other information in an LDAP directory domain of an Open Directory master or replica. The LDAP database is usually located on the startup volume, but can be on a different local volume.

Note: For security purposes, databases that store authentication information for Open Directory and Kerberos are always located on the startup volume regardless of the LDAP database location.

To change the location of a shared LDAP database:

1. Open Server Admin and in the Computers & Services list, select Open Directory for a server that is an Open Directory master or an Open Directory replica.
2. Click Settings (near the bottom of the window), then click Protocols (near the top).
3. Choose LDAP Settings from the Configure pop-up menu, then specify the folder path where you want the LDAP database to be located.
4. Click Save.

Limiting Search Results for LDAP Service

Using Server Admin, you can prevent one type of denial-of-service attack on Mac OS X Server by limiting the number of search results returned by the server's shared LDAP directory domain. Limiting the number of search results prevents a malicious user from tying up the server by sending it multiple all-inclusive LDAP search requests.

To set a maximum number of LDAP search results:

1. Open Server Admin and in the Computers & Services list, select Open Directory for a server that is an Open Directory master or an Open Directory replica.
2. Click Settings (near the bottom of the window), then click Protocols (near the top).
3. Choose LDAP Settings from the Configure pop-up menu, then enter the maximum number of search results.
4. Click Save.

Changing the Search Timeout for LDAP Service

Using Server Admin, you can prevent one type of denial-of-service attack on Mac OS X Server by limiting the amount of time the server will spend on one search of its shared LDAP directory domain. Setting a search timeout prevents a malicious user from tying up the server by sending it an exceptionally complex LDAP search request.

To set a timeout interval for LDAP searches:

1. Open Server Admin and in the Computers & Services list, select Open Directory for a server that is an Open Directory master or an Open Directory replica.
2. Click Settings (near the bottom of the window), then click Protocols (near the top).
3. Choose LDAP Settings from the Configure pop-up menu, then specify a search timeout interval.
4. Click Save.

Setting up SSL for LDAP Service

Using Server Admin, you can set up encrypted communications between a shared LDAP directory domain on Mac OS X Server and other servers that connect to the directory domain. You can enable Secure Sockets Layer (SSL) for encrypted LDAP communications and specify the location of the SSL certificate file, key file, and certificate authority (CA) certificate file.

SSL communications for LDAP use port 636. If SSL is disabled for LDAP service, communications are sent as clear text on port 389.

To set up SSL communications for LDAP service:

1. Open Server Admin and in the Computers & Services list, select Open Directory for a server that is an Open Directory master or an Open Directory replica.
2. Click Settings (near the bottom of the window), then click Protocols (near the top).
3. Choose LDAP Settings from the Configure pop-up menu, then select Use SSL.
4. Enter the location and name for the SSL Certificate, SSL Key, and CA Certificate.

   Instead of typing or pasting the location and name of the SSL Certificate, SSL Key, or CA Certificate, you can locate it by clicking the Browse button next to the field.
5. Click Save.

Migrating a Directory Domain From Netinfo to LDAP

You can use Server Admin to migrate a shared NetInfo directory domain to LDAP. The migration process irreversibly replaces the directory domain's NetInfo back-end database with a Berkeley DB back-end database. After migration, client computers that were configured to use NetInfo to access the directory domain will be able to continue accessing it.

After migration, you can configure DHCP service to provide the migrated directory domain as an LDAP server to client computers with Mac OS X or Mac OS X Server version 10.2 and later that have automatic authentication search policies.

You can have client computers with Mac OS X version 10.3 or Mac OS X Server version 10.3 automatically switch to using LDAP to access the migrated directory domain. The migration process can store auto-switch information in the directory domain. When Mac OS X and Mac OS X Server version 10.3 and later use NetInfo to access a directory domain that has been migrated to LDAP, they pick up the auto-switch information from the directory domain and reconfigure themselves to access the directory domain using LDAP henceforth.

When you set up migration, you can specify a date on which NetInfo access to the migrated directory domain will be disabled. Alternatively, you can disable NetInfo access at any time by clicking a button. After NetInfo is disabled, client computers can't switch automatically to LDAP.

The migration process moves all standard record types and data types from the NetInfo database to an LDAP database. If the NetInfo directory domain was modified to contain custom record types or data types, they are not moved to the LDAP database.

Migration to LDAP does not change how user passwords are validated except for passwords validated by Authentication Manager. Passwords that were validated by a Password Server continue to be validated by the same Password Server. If any user accounts in the NetInfo domain used Authentication Manager for password validation, the migration process converts them to have a password type of Open Directory. Of course, an administrator can change the password type of any migrated user account to Open Directory so that the user account can take advantage of single signon and Kerberos authentication.

Important: Do not click the Disable NetInfo button by accident. Clicking Disable NetInfo immediately disables NetInfo access to the directory domain. You can't undo this change. After disabling NetInfo, all computers that need to connect to the directory domain must be configured to do so using LDAP.

To migrate a server's shared directory domain from NetInfo to LDAP:

1. Open Server Admin and select Open Directory for an Open Directory master server in the Computers & Services list.
2. Click Settings (near the bottom of the window), then click Protocols (near the top).
3. Choose NetInfo Migration from the Configure pop-up menu.
4. Click Migrate and set the migration options.

   Administrator short name: The short name of an administrator account in the server's local directory domain that you want to have copied to the migrated LDAP directory. This account will be an administrator of the LDAP directory domain.

   Administrator password: The password for the administrator account whose short name you entered.

   Kerberos realm name: By convention, the Kerberos realm name is the same as the server's DNS name but in all uppercase letters. For example, a server whose DNS name is example.com would have a Kerberos realm name of EXAMPLE.COM.

   Search base (optional): The search base suffix for the migrated LDAP directory. Typically, the search base suffix is derived from the server's DNS name. For example, the search base suffix could be "dc=example, dc=com" for a server whose DNS name is server.example.com.

   Switch existing NetInfo clients to LDAP: Enables client computers with Mac OS X or Mac OS X Server version 10.3 to automatically reconfigure themselves to access the migrated directory domain using LDAP instead of NetInfo.

   Shut down NetInfo Server at 2:00 am on __: Enter a date when you want to end NetInfo access to the migrated directory domain. After NetInfo is disabled, all computers must use LDAP to access the migrated directory domain.
5. Click OK to begin migration.

   The migration process can take a while.

6 After migration finishes, set up DHCP service to provide the LDAP server's address to client computers with automatic search policies.

Computers with Mac OS X or Mac OS X Server version 10.2 can have automatic search policies. These computers don't have to be configured individually to access the LDAP server. When these computers start up, they try to get an LDAP server's address from DHCP service.

For instructions on setting up DHCP service to supply an LDAP server's address, see the network services administration guide.

Switching Directory Access From NetInfo to LDAP

After you migrate a shared directory domain of Mac OS X Server from NetInfo to LDAP, some clients will switch to LDAP automatically, but you may have to configure other clients to use LDAP and you may have to reconfigure DHCP service.

- Computers with an automatic authentication search policy get the address of their directory server from DHCP service. Therefore, you need to change DHCP service to supply the address of the migrated LDAP directory's server.
- Computers with Mac OS X Server version 10.3 that were using NetInfo to access the migrated directory domain can switch to LDAP automatically. Automatic switching must be enabled when the directory domain is migrated from NetInfo to LDAP. Mac OS X can no longer switch automatically to LDAP after you disable NetInfo on the migrated directory domain's server.
- You can manually switch a Mac OS X computer to LDAP by using Directory Access.
  - You can configure the computer to use an automatic authentication search policy. In this case, you also need to configure DHCP service to supply the migrated LDAP directory server's address to its clients.
  - Alternatively, you can set up an LDAPv3 configuration on the computer and add this LDAPv3 configuration to the computer's custom authentication search policy.
- After you disable NetInfo on the server, make sure DHCP is not supplying the server's address for NetInfo binding.

For more information, see "Migrating a Directory Domain From Netinfo to LDAP" on page 66, "Setting Up the Authentication and Contacts Search Policies" on page 87, and "Accessing LDAP Directories" on page 90, and the DHCP chapter in the network services administration guide.

Disabling NetInfo After Migrating to LDAP

If none of the client computers on your network needs NetInfo access to a directory domain that has been migrated to LDAP, you can use Server Admin to disable NetInfo. You can manually disable the NetInfo server even if you scheduled a shutdown of the NetInfo server while setting up the migration to LDAP.

Important: Do not disable NetInfo prematurely. After disabling NetInfo, all computers that need to connect to the directory domain must be configured to do so using LDAP.

To disable NetInfo access to a directory domain that has been migrated to LDAP:

1. Open Server Admin and select Open Directory for an Open Directory master server in the Computers & Services list.
2. Click Settings (near the bottom of the window), then click Protocols (near the top).
3. Choose NetInfo Migration from the Configure pop-up menu.
4. Click Disable NetInfo.

   Clicking Disable NetInfo immediately disables NetInfo access to the directory domain. You can't undo this change.

Managing User Authentication

6

The authentication services included with Mac OS X Server don't require any setup, but you can change how each user is authenticated.

Mac OS X Server can authenticate users by:

- Using single signon with the Kerberos Key Distribution Center (KDC) built into Mac OS X Server
- Using a password stored securely in the Open Directory Password Server database
- Using a shadow password stored as several hashes, including NT and LAN Manager, in a file that only the root user can access
- Using a crypt password stored directly in the user's account
- Using a non-Apple LDAP server for simple LDAP bind authentication

Single signon and Kerberos authentication require minimal setup of Mac OS X Server. The other authentication options require no setup of Mac OS X Server.

You can manage how Mac OS X Server uses the available options to authenticate users. For task descriptions and instructions, see:

- "Composing a Password" on page 72
- "Changing a User's Password" on page 72
- "Resetting the Passwords of Multiple Users" on page 73
- "Changing the Global Password Policy" on page 74
- "Setting Password Policies for Individual Users" on page 75
- "Changing a User's Password Type" on page 76
  This includes changing the password type to Open Directory, shadow password, or crypt password; and enabling single signon, Kerberos, or LDAP bind authentication.
- "Assigning Administrator Rights for Open Directory Authentication" on page 80
- "Exporting and Importing Users Whose Password Type Is Open Directory" on page 81
- "Migrating Passwords to Open Directory Authentication" on page 82

Composing a Password

The password associated with a user's account must be entered by the user when he or she authenticates for login or some other service. The password is case sensitive (except for SMB LAN Manager passwords) and is masked on the screen as it is entered.

Regardless of the password type you choose for any user, here are some guidelines for composing a password for Mac OS X Server users:

- A password should contain letters, numbers, and symbols in combinations that won't be easily guessed by unauthorized users. Passwords should not consist of actual words. Good passwords might include digits and symbols (such as # or $). Or they might consist of the first letter of all the words in a particular phrase. Use both uppercase and lowercase letters.
- Avoid spaces and Option-key combinations.
- Avoid characters that can't be entered on computers the user will be using or that might require knowing a special keystroke combination to enter correctly on different keyboards and platforms.
- Some network protocols do not support passwords that contain leading spaces, embedded spaces, or trailing spaces.
- A zero-length password is not recommended; Open Directory and some systems (such as LDAP bind) do not support a zero-length password.

For maximum compatibility with computers and services your users might use, use only ASCII characters in passwords.

Changing a User's Password

You can use Workgroup Manager to change a user's password.

To change a user's password:

1. In Workgroup Manager, click the Accounts button, then click the User button.
2. Open the directory domain that contains the user account whose password you want to change, and authenticate as an administrator of the domain.

   To open a directory domain, click the small globe icon above the list of users and choose from the pop-up menu.

   If the user's password type is Open Directory, you must authenticate as an administrator whose password type is Open Directory.
3. Select the account whose password needs to be changed.
4. Enter a password on the Basic pane, then click Save.
5. Tell the user the new password so he or she can log in.

   After the user logs in to Mac OS X with the new password, the user can change the password by clicking Accounts in System Preferences.

If you change the password of an account whose password type is Open Directory and the account resides in the LDAP directory of an Open Directory replica or master, the change will eventually be synchronized with the master and all its replicas. Mac OS X Server automatically synchronizes changes to Open Directory passwords among a master and its replicas.

Resetting the Passwords of Multiple Users

You can use Workgroup Manager to simultaneously select multiple user accounts and change them all to have the same password type and the same temporary password.

To change the password type and password of multiple user accounts:

1. In Workgroup Manager, click the Accounts button, then click the User button.
2. Open the directory domain that contains the user account whose password types and passwords you want to reset, and authenticate as an administrator of the domain.

   To open a directory domain, click the small globe icon above the list of users and choose from the pop-up menu.

   If you want to set the password type to be Open Directory, you must authenticate as an administrator whose password type is Open Directory.
3. Command-click or Shift-click user accounts to select all accounts whose password type needs to be changed.
4. Enter a password on the Basic pane, then set the User Password Type option on the Advanced pane.
5. Click Save.
6. Tell the users the temporary password so they can log in.

   After logging in with the temporary password, a user can change the password by clicking Accounts in System Preferences.

   If you change the password of accounts whose password type is Open Directory and the accounts reside in the LDAP directory of an Open Directory replica or master, the change will eventually be synchronized with the master and all its replicas. Mac OS X Server automatically synchronizes changes to Open Directory passwords among a master and its replicas.

Changing the Global Password Policy

Using Server Admin, you can set a global password policy for user accounts in a Mac OS X Server directory domain. The global password policy affects user accounts in the server's local directory domain. If the server is an Open Directory master or replica, the global password policy also affects the server's LDAP directory domain. If you change the global password policy on an Open Directory replica, the policy settings will eventually be synchronized with the master and any other replicas of it.

Both Kerberos and Open Directory Password Server enforce password policies. Some password policy rules apply to Open Directory Password Server and Kerberos, and some apply only to Open Directory Password Server. Mac OS X Server synchronizes the password policy rules that apply to both Kerberos and Open Directory Password Server.

Administrator accounts are always exempt from password policies. Each user can have an individual password policy that overrides some of the global password policy settings. For more information, see "Setting Password Policies for Individual Users" on page 75.

To change the global password policy of all user accounts in the same domain:

1. Open Server Admin and select Open Directory for a server in the Computers & Services list.
2. Click Settings (near the bottom of the window), then click Authentication (near the top).
3. Set the password policy options you want enforced for users who do not have their own individual password policies.

   "Disable login on __": If you select this option, enter a date in mm/dd/yyyy format; for example, 02/22/2005.

   "Password must be changed every __": If you select this option, remember that some service protocols don't allow users to change passwords. For example, users can't change their passwords when authenticating for IMAP mail service, and users can't change passwords when authenticating for Windows file service.
4. Click Save.

Replicas of the Open Directory master automatically inherit its global password policy.

Setting Password Policies for Individual Users

Using Workgroup Manager, you can set password policies for individual user accounts whose password type is Open Directory. The password policy for a user overrides the global password policy defined on the Authentication Settings pane of Open Directory service in Server Admin. Administrator accounts are always exempt from password policies.

Both Kerberos and Open Directory Password Server enforce password policies. Some password policy rules apply to Open Directory Password Server and Kerberos, and some apply only to Open Directory Password Server. Mac OS X Server synchronizes the password policy rules that apply to both Kerberos and Open Directory Password Server.

To set a user account's password policy, you must have administrator rights for Open Directory authentication in the directory domain that contains the user accounts whose password policy you want to change. This means you must authenticate as a directory domain administrator whose password type is Open Directory. For more information, see "Assigning Administrator Rights for Open Directory Authentication" on page 80.

To change the password policy for a user account:

1. In Workgroup Manager, open the account you want to work with if it is not already open.

   To open an account, click the Accounts button, then click the Users button. Click the small globe icon above the list of users and choose from the pop-up menu to open the directory domain where the user's account resides. Click the lock and authenticate as a directory domain administrator whose password type is Open Directory. Then select the user in the list.

2. Click Advanced, then click Options.

   You can click Options only if the password type is Open Directory.

3. Change password policy options, then click OK.

   "Disable login on date __": If you select this option, enter a date in mm/dd/yyyy format; for example, 02/22/2005.

   "Require a change every __ days": If you select this option, remember that some service protocols don't allow users to change passwords. For example, users can't change their passwords when authenticating for IMAP mail service.

   The password ID is a unique 128-bit number assigned when the password is created in the Open Directory Password Server database. It might be helpful in troubleshooting, since it appears in the Password Server log when a problem occurs. View this Open Directory log in Server Admin.

4. Click Save.

Changing a User's Password Type

You can set the password type on the Advanced pane of Workgroup Manager to one of the following:

- Open Directory
- Shadow password
- Crypt password

Setting a user's password type to Open Directory enables multiple legacy authentication methods and also enables single signon and Kerberos if the user's account is in an LDAP directory. You can also enable a user account to use simple LDAP bind authentication. For explanations of the authentication options, see Chapter 3, "User Authentication With Open Directory."

Changing the Password Type to Open Directory

Using Workgroup Manager, you can specify that Open Directory be used for authenticating one or more user accounts stored in the local directory domain or the LDAP directory domain of Mac OS X Server. In addition, you can specify that Open Directory be used for authenticating user accounts in any LDAP or NetInfo directory domain residing on a server with Mac OS X Server version 10.2.

The Open Directory password type supports single signon using Kerberos authentication. It also supports the Simple Authentication and Security Layer (SASL) authentication protocols, which include APOP, CRAM-MD5, DHX, Digest-MD5, MS-CHAPv2, SMB-NT, SMB-LAN Manager, and WebDAV-Digest.

To set a user account's password type to Open Directory, you must have administrator rights for Open Directory authentication in the directory domain that contains the user account. This means you must authenticate as a directory domain administrator whose password type is Open Directory. For more information, see "Assigning Administrator Rights for Open Directory Authentication" on page 80.

To specify that a user account authenticate using Open Directory:

1. Make sure the user's account resides in a directory domain that supports Open Directory authentication.

   Directory domains on Mac OS X Server version 10.3 support Open Directory authentication, as do directory domains on Mac OS X Server version 10.2 that are configured to use a Password Server.

2. In Workgroup Manager, open the account you want to work with if it is not already open.

   To open an account, click the Accounts button, then click the Users button. Click the small globe icon above the list of users and choose from the pop-up menu to open the directory domain where the user's account resides. Click the lock and authenticate as a directory domain administrator whose password type is Open Directory. Then select the user in the list.

3. Click Advanced, then choose Open Directory from the User Password Type pop-up menu.

4. If you changed the user's password type, you will be prompted to enter and verify a new password.

   If you are working with a new user, enter the password on the Basic pane in the Password field, then reenter it in the Verify field.

   The password must contain no more than 512 bytes (up to 512 characters, although the network authentication protocol can impose different limits; for example, 128 characters for SMB-NT and 14 for SMB-LAN Manager. The user management guide provides guidelines for choosing passwords).

5. On the Advanced pane, click Options to set up the user's password policy, and click OK when you have finished specifying options

   If you select the "Disable login as of" option, enter a date in MM/DD/YYYY format; for example, 02/22/2004.

   If you use a policy that requires user password changing, remember that not all protocols support changing passwords. For example, users can't change their passwords when authenticating for IMAP mail service.

   The password ID is a unique 128-bit number assigned when the password is created in the Open Directory Password Server database. It might be helpful in troubleshooting, since it appears in the Password Server log when a problem occurs. View this Open Directory log in Server Admin.

6. Click Save.

Changing the Password Type to Crypt Password

Using Workgroup Manager, you can specify that a crypt password be used for authenticating one or more user accounts stored in an LDAP or NetInfo directory domain. The LDAP directory domain can be on any server, but cannot be a read-only directory. The NetInfo domain can be on any Mac OS X Server.

The crypt password is stored as an encrypted value, or hash, in the user account. Because the crypt password can be recovered easily from the directory domain, it is subject to offline attack and therefore is less secure than other password types.

To specify that a user account authenticate using a crypt password:

1. In Workgroup Manager, open the account you want to work with if it is not already open.

   To open an account, click the Accounts button, then click the Users button. Click the small globe icon above the list of users and choose from the pop-up menu to open the directory domain where the user's account resides. Click the lock and authenticate as a directory domain administrator. Then select the user in the list.

2. Click Advanced, then choose "Crypt password" from the User Password Type pop-up menu.

3. If you changed the user's password type, you will be prompted to enter and verify a new password.

   If you are working with a new user, enter the password on the Basic pane in the Password field, then reenter it in the Verify field.

   A crypt password can be at most eight bytes (eight ASCII characters) long. If you enter a longer password, only the first eight bytes are used.

4. Click Save.

Changing the Password Type to Shadow Password

Using Workgroup Manager, you can specify that a user have a shadow password stored in a secure file apart from the directory domain. Only users whose accounts reside in the local directory domain can have a shadow password.

To specify that a user account authenticate using a shadow password:

1. In Workgroup Manager, open the account you want to work with if it is not already open.

   To open an account, click the Accounts button, then click the Users button. Click the small globe icon above the list of users and choose from the pop-up menu to open the local directory domain where the user's account resides. Click the lock and authenticate as a directory domain administrator. Then select the user in the list.

2. Click Advanced, then choose Shadow Password from the User Password Type pop-up menu.

3. If you changed the user's password type, you will be prompted to enter and verify a new password.

   If you are working with a new user, enter the password on the Basic pane in the Password field, then reenter it in the Verify field.

4. Click Save.

Enabling Single Signon Authentication for a User

You enable single signon authentication for a user account in an LDAP directory Mac OS X Server version 10.3 by using the Advanced pane of Workgroup Manager to set the account's password type to Open Directory. Single signon is a feature of Kerberos authentication. For instructions, see "Changing the Password Type to Open Directory" on page 76.

Enabling Kerberos Authentication for a User

You enable Kerberos authentication for a user account in an LDAP directory of Mac OS X Server version 10.3 by setting the account's password type to Open Directory on the Advanced pane of Workgroup Manager. For instructions, see "Changing the Password Type to Open Directory" on page 76.

Enabling LDAP Bind Authentication for a User

You can use Workgroup Manager to enable the use of LDAP bind authentication for a user account stored in an LDAP directory domain. When you use this password validation technique, you rely on the LDAP server that contains the user account to authenticate the user's password.

To enable LDAP bind user authentication using Workgroup Manager:

1. Make sure the account for a user whose password you want to validate using LDAP bind resides on an LDAP server in the search path of the Mac OS X computer that needs to validate the password.

   See "Accessing LDAP Directories" on page 90 for information about configuring LDAP server connections. Avoid mapping the password attribute when configuring the connection; bind authentication will occur automatically. Also, set up the connection so it uses SSL in order to protect the password, passed in clear text, while it is in transit.

2. In Workgroup Manager, open the account you want to work with if it is not already open.

   To open an account, click the Accounts button, then click the Users button. Click the small globe icon above the list of users and choose from the pop-up menu to open the LDAP directory domain where the user's account resides. Click the lock and authenticate as a directory domain administrator. Then select the user in the user list.

3. On the Advanced pane, choose "Crypt password" from the User Password Type pop-up menu.

4. On the Basic pane, make sure the Password field is empty.

5. Click Save.

Assigning Administrator Rights for Open Directory Authentication

You can work with Open Directory authentication settings in Workgroup Manager only if you authenticate as an administrator of the directory domain that contains the user accounts you want to work with. In addition, the administrator must use Open Directory authentication. These restrictions protect the security of passwords stored in the Kerberos KDC and the Open Directory Password Server database. See "Changing the Password Type to Open Directory" on page 76. For instructions on assigning administrator rights for a directory domain, see the user accounts chapter in the user management guide.

Do not use the Options button on the Advanced pane to set up password policies for directory domain administrators. Password policies are not enforced for administrator accounts. Directory domain administrators need to be able to change password policies of individual user accounts.

Exporting and Importing Users Whose Password Type Is Open Directory

When you export user accounts whose password type is set to Open Directory, passwords are not exported. This protects the security of the Open Directory Password Server database. Before importing, you can use a spreadsheet application to open the file of exported users and preset their passwords, which they can change the next time they log in.

After importing, you have a couple of options for setting the passwords of the imported user accounts:

- You can set all the imported user accounts to use a temporary password, which each user can change the next time he or she logs in. For instructions, see "Resetting the Passwords of Multiple Users" on page 73.
- You can set the password of each imported user account individually on the Basic pane of Workgroup Manager. For instructions, see "Changing a User's Password Type" on page 76.

Exporting and Importing Authentication Manager Users

When you export user accounts that have crypt passwords from a NetInfo domain for which Authentication Manager is enabled, passwords are not exported. After importing to a directory domain of Mac OS X Server version 10.3, you have a couple of options for setting the passwords of the imported user accounts:

- You can set all the imported user accounts to use a temporary password, which each user can change the next time he or she logs in. For instructions, see "Resetting the Passwords of Multiple Users" on page 73.
- You can set the password of each imported user account individually on the Basic pane of Workgroup Manager. For instructions, see "Changing a User's Password Type" on page 76.

Authentication Manager is a legacy technology for securely validating passwords. Authentication Manager only works with user accounts that were created in a NetInfo domain of Mac OS X Server version 10.0–10.2. Authentication Manager must have been enabled for the NetInfo domain. For more information, see Appendix C, "Authentication Manager."

Migrating Passwords to Open Directory Authentication

User accounts can be migrated from earlier versions of Mac OS X Server by importing the account records or upgrading the server where they reside. User accounts created with Mac OS X Server version 10.1 or earlier have no authentication authority attribute but do have crypt passwords. For compatibility with such user accounts, Mac OS X Server version 10.2 and later assumes a user account without an authentication authority attribute has a crypt password.

If you import user accounts from Mac OS X Server version 10.1 or earlier, the user accounts have no authentication authority attribute. Therefore these user accounts are initially configured to have crypt passwords. An appendix in the user management guide describes how to import user accounts.

Likewise, if you upgrade from Mac OS X Server version 10.1 or earlier, user accounts that were created before upgrading have no authentication authority attribute. After upgrading, these user accounts are assumed to have crypt passwords.

While all the existing crypt passwords can continue to be used after importing or upgrading, you can change the user accounts to use Open Directory authentication. You can change individual user accounts or multiple user accounts by using Workgroup Manager. Changing a user account's password type will reset the password. For instructions, see "Changing the Password Type to Open Directory" on page 76.

Some user accounts created with Mac OS X Server version 10.1 or earlier may use Authentication Manager. It is a legacy technology for authenticating users of Windows file service and users of Apple file service whose Mac OS 8 computers have not been upgraded with AFP client software version 3.8.3 or later.

When migrating Authentication Manager users, you have the following options:

- If you upgrade server version first from Mac OS X Server version 10.1 to version 10.2 and then to version 10.3, existing users can continue to use their same passwords.
- You can change some or all upgraded user accounts to use Open Directory authentication, which is the preferred option for authenticating Windows users. Users of both types can coexist in the same directory domain.
- If you import user accounts that use Authentication Manager, they will be converted to Open Directory authentication during importing.

Managing Directory Access

7

You can use Directory Access to set up and manage how a computer with Mac OS X or a server with Mac OS X Server accesses directory services and discovers network services.

For setup and management task descriptions and instructions, see:

Setting Up Services in Directory Access

Directory Access lists the different kinds of services that Mac OS X can access. The list includes directory services, which give Mac OS X access to user information and other administrative data stored in directory domains. The list also includes kinds of network services that Mac OS X can discover on the network.

You can enable or disable access to each kind of service. If you disable a kind of service in Directory Access, Mac OS X no longer accesses services of the disabled kind. However, disabling a kind of service in Directory Access does not affect the ability of Mac OS X to use or provide services of that kind. For example, if you disable Rendezvous, Mac OS X does not use it to discover file services, but you can still share your files and connect to a file server if you know its address.

Enabling or Disabling Active Directory Service

You can use Directory Access to enable or disable the use of Active Directory on a Windows server. Active Directory is the directory service of Windows 2000 and 2003 servers.

To enable or disable access to Active Directory:

1. In Directory Access, click Services.
2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Click the checkbox next to Active Directory and click Apply.

For configuration instructions, see "Accessing LDAP Directories" on page 90.

Enabling or Disabling AppleTalk Service Discovery

You can use Directory Access to enable or disable the discovery of AppleTalk network services. AppleTalk is a legacy Mac OS protocol for network file and print services. Some computers use AppleTalk to share files, and some servers use AppleTalk for file service. In addition, some shared printers use AppleTalk.

To enable or disable AppleTalk service discovery:

1. In Directory Access, click Services.
2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Click the checkbox next to AppleTalk and click Apply.

AppleTalk does not require configuration.

Enabling or Disabling BSD Flat File and NIS Directory Services

You can use Directory Access to enable or disable the use of BSD configuration files and access to Network Information Service (NIS) directory services. BSD configuration files are the original method for accessing administrative data on UNIX computers, and some organizations still use them. Some UNIX servers use NIS to provide directory services.

To enable or disable BSD flat file and NIS directory services:

1. In Directory Access, click Services.
2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Click the checkbox next to "BSD Flat File and NIS" and click Apply.

For configuration instructions, see "Accessing an NIS Domain" on page 107 and "Using BSD Configuration Files" on page 108.

Enabling or Disabling LDAP Directory Services

You can use Directory Access to enable or disable access to directory services that use Lightweight Directory Access Protocol (LDAP) versions 2 and 3. A single Directory Access plug-in named LDAPv3 provides access to both LDAP versions 2 and 3. (The LDAPv2 plug-in of Mac OS X version 10.2 is not needed with Mac OS X version 10.3.)

Mac OS X Server version 10.3 and later provides only LDAPv3 directory service to other computers, including Mac OS X computers. Mac OS X Server version 10.2 can provide LDAPv3 directory service to other computers (and it can provide NetInfo directory service). Many other servers also provide LDAPv3 directory service; LDAPv3 is an open standard common in mixed networks of Macintosh, UNIX, and Windows systems. Some servers also use the older version, LDAPv2, to provide directory service.

To enable or disable LDAP directory services:

1. In Directory Access, click Services.
2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Click the checkbox next to LDAPv3 and click Apply.

For configuration instructions, see "Accessing LDAP Directories" on page 90.

Enabling or Disabling NetInfo Directory Services

You can use Directory Access to enable or disable access to shared NetInfo directory domains. NetInfo is a legacy directory service that is still used for the local directory domain on every Mac OS X computer, including Mac OS X Server. NetInfo can also be used for a shared directory domain of Mac OS X Server version 10.2 and earlier.

Disabling NetInfo in Directory Access does not disable access to the computer's local NetInfo domain. Only access to shared NetInfo domains can be disabled.

To enable or disable NetInfo directory services:

1. In Directory Access, click Services.
2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Click the checkbox next to NetInfo and click Apply.

For configuration instructions, see "Accessing Legacy NetInfo Domains" on page 109.

Enabling or Disabling Rendezvous Service Discovery

You can use Directory Access to enable or disable the discovery of some Rendezvous network services. For example, disabling Rendezvous in Directory Access prevents Rendezvous-enabled file servers from appearing in the Network globe in the Finder. But disabling Rendezvous in Directory Access does not prevent Rendezvous-enabled printers from appearing in the Printer Setup Utility or prevent iTunes from using Rendezvous for music sharing. Rendezvous is an Apple protocol for discovering file, print, and other services on Internet Protocol (IP) networks.

To enable or disable Rendezvous service discovery:

1. In Directory Access, click Services.
2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Click the checkbox next to Rendezvous and click Apply.

   Rendezvous does not require configuration.

Enabling or Disabling SLP Service Discovery

You can use Directory Access to enable or disable the discovery of services that use Service Location Protocol (SLP) to make themselves known on the network. SLP is an open standard for discovering file and print services on Internet Protocol (IP) networks.

To enable or disable SLP service discovery:

1. In Directory Access, click Services.
2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Click the checkbox next to SLP and click Apply.

   SLP does not require configuration.

Enabling or Disabling SMB Service Discovery

You can use Directory Access to enable or disable the discovery of services that use Server Message Block (SMB) to make themselves known on the network. SMB is a protocol used by Microsoft Windows for file and print services.

To enable or disable SMB service discovery:

1. In Directory Access, click Services.
2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Click the checkbox next to SMB and click Apply.

   For configuration instructions, see “Configuring SMB Service Discovery” on page 87.

Configuring SMB Service Discovery

You can configure how Mac OS X uses the Server Message Block (SMB) protocol to discover Windows file servers on the network. You can use the Directory Access application to specify the following:

- The Windows workgroup that the computer is a member of
- A Windows Internet Naming Service (WINS) server on the network

To configure discovery of Windows SMB file servers:

1. In Directory Access, click Services.
2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Select SMB in the list of services, then click Configure.
4. In the Workgroup field, type a workgroup name or select one from the drop-down list.

   The drop-down list includes the names of Windows workgroups that other computers on the network are members of.
5. Enter the DNS name or IP address of a WINS server that provides NetBIOS name resolution for the network, then click OK.

   A WINS Server resolves Windows computer names to IP addresses on a network with routers and multiple subnets.

   If the network does not have a WINS server, leave the WINS Server field blank.

Setting Up the Authentication and Contacts Search Policies

Directory Access defines an authentication search policy and a contacts search policy.

- Mac OS X uses the authentication search policy to locate and retrieve user authentication information and other administrative data from directory domains.
- Mac OS X uses the contacts search policy to locate and retrieve name, address, and other contact information from directory domains. Mac OS X Address Book uses this contact information, and other applications can be programmed to use it as well.

Each search policy consists of a list of directory domains (also known as directory nodes). The order of directory domains in the list defines the search policy. Starting at the top of the list, Mac OS X searches each listed directory domain in turn until it either finds the information it needs or reaches the end of the list without finding the information.

Each search policy, authentication and contacts, can be set to Automatic, Local directory, or Custom path.

- **Automatic** starts with the local directory domain and can include an LDAP directory supplied automatically by DHCP and NetInfo domains to which the computer is bound. An automatic search policy is the default setting for Mac OS X version 10.2 and later and offers the most flexibility for mobile computers.
- **Local directory** includes only the local directory domain.
- **Custom path** starts with the local directory domain and includes your choice of LDAP directories, an Active Directory domain, NetInfo domains, BSD configuration files, and an NIS domain.

Defining Automatic Search Policies

Using Directory Access, you can configure a Mac OS X computer's authentication and contacts search policies to be defined automatically. An automatically defined search policy includes the local directory domain. It can also include an LDAP directory server specified by DHCP service and shared NetInfo domains to which the computer is bound. This is the default configuration for both the authentication and the contacts search policy.

Note: Some applications, such as Mac OS X Mail and Address Book, can access LDAP directories directly, without using Open Directory. To set up one of these applications to access LDAP directories directly, open the application and set the appropriate preference.

To have a search policy defined automatically:

1. In Directory Access, click Authentication or click Contacts.

   Authentication shows the search policy used for authentication and most other administrative data.

   Contacts shows the search policy used for contact information in applications such as Address Book.

2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Choose Automatic from the Search pop-up menu, then click Apply.
4. In System Preferences, make sure the computer's Network preferences are configured to use DHCP or DHCP with manual IP address.
5. If you want the DHCP service of Mac OS X Server to supply its clients with a particular LDAP server's address for their automatic search policies, you need to configure the LDAP options of DHCP service.

   For instructions, see the DHCP chapter of the network services administration guide.

Defining Custom Search Policies

Using Directory Access, you can configure a Mac OS X computer's authentication and contacts search policies to use a custom list of directory domains. A custom list starts with the computer's local directory domain and you can also include Open Directory and other LDAP directory domains, an Active Directory domain, shared NetInfo domains, BSD configuration files, and an NIS domain.

Note: Make sure the computer has been configured to access the LDAP directories, Active Directory domain, NetInfo domains, and NIS domain that you want to add to the search policy. For instructions, see the subsequent sections of this chapter.

To specify a custom list of directory domains for a search policy:

1. In Directory Access, click the Authentication or click Contacts.

   Authentication shows the search policy used for authentication and most other administrative data.

   Contacts shows the search policy used for contact information in applications such as Address Book.

2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Choose "Custom path" from the Search pop-up menu.
4. Add directory domains as needed.

   Add directory domains by clicking Add, selecting one or more directories, and clicking Add again.

5. Change the order of the listed directory domains as needed, and remove listed directory domains that you don't want in the search policy.

   Move a directory domain by dragging it up or down the list.

   Remove a listed directory domain by selecting it and clicking Remove.

6. Click Apply.

Defining Local Directory Search Policies

Using Directory Access, you can configure a Mac OS X computer's authentication and contacts search policies to use only the computer's local directory domain. A search policy that uses only the local directory limits the access that a computer has to authentication information and other administrative data. If you restrict a computer's authentication search policy to use only the local directory, only users with local accounts can log in.

To have a search policy use only the local directory domain:

1 In Directory Access, click the Authentication or click Contacts.

Authentication shows the search policy used for authentication and most other administrative data.

Contacts shows the search policy used for contact information in applications such as Address Book.

2 If the lock icon is locked, click it and type the name and password of an administrator.

3 Choose "Local directory" from the Search pop-up menu, then click Apply.

Accessing LDAP Directories

You can configure a server with Mac OS X Server or a computer with Mac OS X to access specific LDAP directories, including the LDAP directory of a Mac OS X Server Open Directory master. For task descriptions and instructions, see:

- "Enabling or Disabling Use of a DHCP-Supplied LDAP Directory" (next)
- "Showing or Hiding Options for LDAP Directories" on page 91
- "Configuring Access to an LDAP Directory" on page 92
- "Changing a Configuration for Accessing an LDAP Directory" on page 93.
- "Duplicating a Configuration for Accessing an LDAP Directory" on page 93.
- "Deleting a Configuration for Accessing an LDAP Directory" on page 94.
- "Changing the Connection Settings for an LDAP Directory" on page 95.
- "Configuring LDAP Searches and Mappings" on page 96
- "Mapping Config Record Attributes for LDAP Directories" on page 98
- "Editing RFC 2307 Mapping to Enable Creating Users" on page 98.
- "Populating LDAP Directories With Data for Mac OS X" on page 100.

In Mac OS X version 10.3, a single Directory Access plug-in named LDAPv3 provides access to both LDAP versions 2 and 3. The LDAPv2 plug-in of Mac OS X version 10.2 is not needed with Mac OS X version 10.3. Existing LDAPv2 configurations are automatically converted to LDAPv3 when a computer is upgraded to Mac OS X version 10.3.

Note: Mac OS X Mail, Address Book and some similar applications can access LDAP directories directly, without using Open Directory. You can configure these applications to search specific LDAP directories. For instructions, open Mail and choose Help > Mail Help or open Address Book and choose Help > Address Book Help; then search for help on LDAP.

Enabling or Disabling Use of a DHCP-Supplied LDAP Directory

Using Directory Access, you can configure a Mac OS X computer to get the address of an LDAP directory server automatically when it starts up. Mac OS X requests the address of an LDAP directory server from the DHCP service that also supplies the computer's IP address, router address, and DNS server addresses. Mac OS X adds the LDAP server's address supplied by DHCP to the computer's automatic search policy. See "Defining Automatic Search Policies" on page 88 for more information.

To enable or disable automatic access to an LDAP server:

1. In Directory Access, click Services.
2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Select LDAPv3 in the list of services, then click Configure.
4. Click "Used DHCP-supplied LDAP Server."

   If you disable this setting, this computer doesn't use an LDAP directory server supplied by DHCP. However, the computer can automatically access shared NetInfo domains. See "Accessing Legacy NetInfo Domains" on page 109 for more information.

   If you enable this setting, the DHCP service should be configured to supply the address of an LDAP directory server. For instructions, see the DHCP chapter of the network services administration guide.

Showing or Hiding Options for LDAP Directories

You can show or hide a list of available configurations for accessing LDAP directories. Each configuration specifies how Open Directory accesses a particular LDAP directory. When you show the list, you see and can change some settings for each LDAP configuration.

To show or hide the available LDAP directory configurations:

1. In Directory Access, click Services.
2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Select LDAPv3 in the list of services, then click Configure.
4. Click the Show Options control or the Hide Options control, whichever is present.

Configuring Access to an LDAP Directory

You can use Directory Access to create a configuration that specifies how Mac OS X accesses a particular LDAPv3 or LDAPv2 directory.

To create a configuration for accessing an LDAP directory:

1. In Directory Access, click Services.
2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Select LDAPv3 in the list of services, then click Configure.
4. If the list of LDAP directory configurations is hidden, click Show Options.
5. Click New and enter a name for the configuration.
6. Press Tab and enter the DNS name or IP address of the server that hosts the LDAP directory you want to access.
7. Click the pop-up menu next to the DNS name or IP address and choose a mapping template or choose From Server.
8. Enter the search base suffix for the LDAP directory and click OK.

   If you chose a template in step 7, you must enter a search base suffix, or the computer will not be able to find information in the LDAP directory. Typically, the search base suffix is derived from the server's DNS name. For example, the search base suffix could be "dc=example, dc=com" for a server whose DNS name is server.example.com.

   If you chose From Server in step 7, you don't need to enter a search base. In this case, Open Directory assumes the search base is the first level of the LDAP directory.
9. Select the SSL checkbox if you want Open Directory to use Secure Sockets Layer (SSL) for connections with the LDAP directory.

   If you want the computer to access the LDAP directory for which you just created a configuration, you must add the directory to a custom search policy in the Authentication or Contacts pane of Directory Access. You must also make sure LDAPv3 is enabled in the Services pane. For instructions, see "Enabling or Disabling LDAP Directory Services" on page 85 and "Defining Custom Search Policies" on page 89.

   Note: Before you can use Workgroup Manager to create users on a non-Apple LDAP server that uses RFC 2307 (UNIX) mappings, you must edit the mapping of the Users record type. For instructions, see "Editing RFC 2307 Mapping to Enable Creating Users" on page 98.

Changing a Configuration for Accessing an LDAP Directory

You can use Directory Access to change the settings of an LDAP directory configuration. The configuration settings specify how Open Directory accesses a particular LDAPv3 or LDAPv2 directory.

To edit a configuration for accessing an LDAP directory:

1. In Directory Access, click Services.
2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Select LDAPv3 in the list of services, then click Configure.
4. If the list of server configurations is hidden, click Show Options.
5. Change any of the settings displayed in the list of server configurations.

   Enable: Click a checkbox to enable or disable access to an LDAP directory server.

   Configuration Name: Double-click a configuration name to edit it.

   Server Name or IP Address: Double-click a server name or IP address to change it.

   LDAP Mapping: Choose a template from the pop-up menu, then enter the search base for the LDAP directory and click OK.

   If you chose a template, you must enter a search base suffix, or the computer will not be able to find information in the LDAP directory. Typically, the search base suffix is derived from the server's DNS name. For example, the search base suffix could be "dc=example, dc=com" for a server whose DNS name is server.example.com.

   If you chose From Server instead of a template, you don't need to enter a search base. In this case, Open Directory assumes the search base is the first level of the LDAP directory.

   SSL: Click a checkbox to enable or disable Secure Sockets Layer (SSL) connections.

Duplicating a Configuration for Accessing an LDAP Directory

You can use Directory Access to duplicate a configuration that specifies how Mac OS X accesses a particular LDAPv3 or LDAPv2 directory. After duplicating an LDAP directory configuration, you can change its settings to make it different from the original configuration.

To duplicate a configuration for accessing an LDAP directory:

1. In Directory Access, click Services.
2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Select LDAPv3 in the list of services, then click Configure.
4. If the list of server configurations is hidden, click Show Options.
5. Select a server configuration in the list, then click Duplicate.

6 Change any of the duplicate configuration's settings.

Enable: Click a checkbox to enable or disable access to an LDAP directory server.

Configuration Name: Double-click a configuration name to edit it.

Server Name or IP Address: Double-click a server name or IP address to change it.

LDAP Mapping: Choose a template from the pop-up menu, then enter the search base for the LDAP directory and click OK.

If you chose a template, you must enter a search base suffix, or the computer will not be able to find information in the LDAP directory. Typically, the search base suffix is derived from the server's DNS name. For example, the search base suffix could be "dc=example, dc=com" for a server whose DNS name is server.example.com.

If you chose From Server instead of a template, you don't need to enter a search base. In this case, Open Directory assumes the search base is the first level of the LDAP directory.

SSL: Click a checkbox to enable or disable Secure Sockets Layer (SSL) connections.

If you want the computer to access the LDAP directory specified by the duplicate configuration you just created, you must add the directory to a custom search policy in the Authentication or Contacts pane of Directory Access. You must also make sure LDAPv3 is enabled in the Services pane. For instructions, see "Enabling or Disabling LDAP Directory Services" on page 85 and "Defining Custom Search Policies" on page 89.

Deleting a Configuration for Accessing an LDAP Directory

You can use Directory Access to delete a configuration that specifies how the computer accesses a particular LDAPv3 or LDAPv2 directory.

To delete a configuration for accessing an LDAP directory:

1 In Directory Access, click Services.

2 If the lock icon is locked, click it and type the name and password of an administrator.

3 Select LDAPv3 in the list of services, then click Configure.

4 If the list of server configurations is hidden, click Show Options.

5 Select a server configuration in the list, then click Delete.

Changing the Connection Settings for an LDAP Directory

You can use Directory Access to change the connection settings of a configuration that specifies how the computer accesses a particular LDAPv3 or LDAPv2 directory.

To change the connection settings for accessing an LDAP directory:

1. In Directory Access, click Services.
2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Select LDAPv3 in the list of services, then click Configure.
4. If the list of server configurations is hidden, click Show Options.
5. Select a server configuration in the list, then click Edit.
6. Click Connection and change any of the settings.

Configuration Name identifies this configuration in the list of LDAP directory configurations. (You can also change the name directly in the list of LDAP directory configurations.)

Server Name or IP Address specifies the server's DNS name or its IP address. (You can also change this directly in the list of LDAP directory configurations.)

"Open/close times out in" specifies the number of seconds that Open Directory waits before cancelling an attempt to connect to the LDAP server.

"Connection times out in" specifies the number of seconds that Open Directory allows an idle or unresponsive connection to remain open.

"Use authentication when connecting" determines whether Open Directory authenticates itself as a user of the LDAP directory by supplying the Distinguished Name and Password when connecting to the directory.

"Encrypt using SSL" determines whether Open Directory encrypts communications with the LDAP directory by using Secure Sockets Layer (SSL) connection. (You can also change this setting directly in the list of LDAP directory configurations.)

"Use custom port" specifies a port number other than the standard port for LDAP connections (389 without SSL or 636 with SSL).

Configuring LDAP Searches and Mappings

Using Directory Access, you can edit the mappings, search bases, and search scopes that specify how Mac OS X finds specific data items in an LDAP directory. You can edit these settings separately for each LDAP directory configuration listed in Directory Access. Each LDAP directory configuration specifies how Mac OS X accesses data in an LDAPv3 or LDAPv2 directory.

- You can edit the mapping of each Mac OS X record type to one or more LDAP object classes.
- For each record type, you can also edit the mapping of Mac OS X data types, or attributes, to LDAP attributes.
- You can edit the LDAP search base and search scope that determine where Mac OS X looks for a particular Mac OS X record type in an LDAP directory.

Important: When mapping Mac OS X user attributes to a read/write LDAP directory domain (an LDAP domain that is not read-only), the LDAP attribute mapped to RealName must not be the same as the first attribute in a list of LDAP attributes mapped to RecordName. For example, the cn attribute must not be the first attribute mapped to RecordName if cn is also mapped to RealName. If the LDAP attribute mapped to RealName is the same as the first attribute mapped to RecordName, problems will occur when you try to edit the full (long) name or the first short name in Workgroup Manager.

For detailed specifications of Mac OS X record types and attributes, see Appendix A, "Mac OS X Directory Data."

To edit the search bases and mappings for an LDAP server:

1. In Directory Access, click Services.
2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Select LDAPv3 in the list of services, then click Configure.
4. If the list of server configurations is hidden, click Show Options.
5. Select a server configuration in the list, then click Edit.
6. Click Search & Mappings.
7. Select the mappings that you want to use as a starting point, if any.

   Click the "Access this LDAPv3 server using" pop-up menu and choose a mapping template to use its mappings as a starting point, or choose Custom to begin with no predefined mappings.

   Or click "Read from Server" to edit the mappings currently stored in the LDAP directory server whose configuration you are editing.

8 Add record types and change their search bases as needed.

To add record types, click the Add button below the Record Types and Attributes list. In the sheet that appears, select Record Types, select one or more record types from the list, and then click OK.

To change the search base of a record type, select it in the Record Types and Attributes List. Then click the "Search base" field and edit the search base.

To remove a record type, select it in the Record Types and Attributes List and click Delete.

To add a mapping for a record type, select the record type in the Record Types and Attributes List. Then click the Add button below "Map to __ items in list" and enter the name of an object class from the LDAP directory. To add another LDAP object class, you can press Return and enter the name of the object class. Specify whether to use all or any of the listed LDAP object classes by using the pop-up menu above the list.

To change a mapping for a record type, select the record type in the Record Types and Attributes List. Then double-click the LDAP object class that you want to change in the "Map to __ items in list" and edit it. Specify whether to use all or any of the listed LDAP object classes by using the pop-up menu above the list.

To remove a mapping for a record type, select the record type in the Record Types and Attributes List. Then click the LDAP object class that you want to remove from the "Map to __ items in list" and click the Delete button below "Map to __ items in list."

9 Add attributes and change their mappings as needed.

To add attributes to a record type, select the record type in the Record Types and Attributes List. Then click the Add button below the Record Types and Attributes list. In the sheet that appears, select Attribute Types, select one or more attribute types, and then click OK.

To add a mapping for an attribute, select the attribute in the Record Types and Attributes List. Then click the Add button below "Map to __ items in list" and enter the name of an attribute from the LDAP directory. To add another LDAP attribute, you can press Return and enter the name of the attribute.

To change a mapping for an attribute, select the attribute in the Record Types and Attributes List. Then double-click the item that you want to change in the "Map to __ items in list" and edit the item name.

To remove a mapping for an attribute, select the attribute in the Record Types and Attributes List. Then click the item that you want to remove from the "Map to __ items in list" and click the Delete button below "Map to __ items in list."

To change the order of attributes displayed in the list on the right, drag the attributes up or down in the list.

10 Click Write to Server if you want to store the mappings in the LDAP directory so that it can supply them automatically to its clients.

You must enter a search base to store the mappings, a distinguished name of an administrator (for example, cn=admin,dc=example,dc=com), and a password. If you are writing mappings to an Open Directory LDAP server, the correct search base is "cn=config, <suffix>" (where <suffix> is the server's search base suffix, such as "dc=example,dc=com").

The LDAP directory supplies its mappings to clients that are configured to use an automatic search policy. For instructions on configuring the client search policy, see "Setting Up the Authentication and Contacts Search Policies" on page 87.

The LDAP directory also supplies its mappings to clients that have been configured manually to get mappings from the server. For instructions on configuring client access to the server, see "Configuring Access to an LDAP Directory" on page 92 through "Changing the Connection Settings for an LDAP Directory" on page 95.

Mapping Config Record Attributes for LDAP Directories

If you want to store information for managed Mac OS X users in an LDAP directory, make sure you map the following attributes of the Config record type: RealName and DataStamp. If you do not map these attributes, the following error message will be displayed when you use Workgroup Manager to change a user record that resides in the LDAP directory:

The attribute with name "dsRecTypeStandard:Config" is not mapped.

You can ignore this message if you are not using Mac OS X client management, which depends on the Config record type's RealName and DataStamp attributes for a cache.

Editing RFC 2307 Mapping to Enable Creating Users

Before you can use Workgroup Manager to create users on a non-Apple LDAP directory server that uses RFC 2307 (UNIX) mappings, you must edit the mapping of the Users record type. You do this with the Directory Access application.

To enable creating user records in an LDAP directory with RFC 2307 mappings:

1 In Directory Access, click Services.

2 If the lock icon is locked, click it and type the name and password of an administrator.

3 Select LDAPv3 in the list of services, then click Configure.

4 If the list of server configurations is hidden, click Show Options.

5 Select the directory configuration with RFC 2307 mappings, then click Edit.

6 Click Search & Mappings.

7 Select Users in the list on the left.

By default, "Map to __ items in list" is set to Any and the list on the right includes posixAccount, inetOrgPerson, and shadowAccount.

8 Change "Map to __ items in list" to All and change the list on the right to the exact set of LDAP object classes to which you want the Users record type mapped.

For example, you could delete shadowAccount from the list so that Users maps to only posixAccount and inetOrgPerson. Or you could map Users to account, posixAccount, and shadowAccount.

To change an item on the list, double-click it.

To add an item to the list, click Add.

To delete the selected item from the list, click Delete.

To change the order of listed items, drag items up or down in the list.

You can find out the object classes of existing user records in the LDAP directory by using the UNIX tool ldapsearch in a Terminal window. The following example would display the object classes for a user record whose cn attribute is "Leonardo da Vinci:"

```
ldapsearch -x -h ldapserver.example.com -b "dc=example, dc=com"
    'cn=Leonardo da Vinci' objectClass
```

The output displayed for this example command could be something similar to the following:

```
# Leonardo da Vinci, example.com
dn: cn=Leonardo da Vinci, dc=example, dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
```

Preparing a Read-Only LDAP Directory for Mac OS X

If you want a Mac OS X computer to get administrative data from a read-only LDAP directory, the data must exist in the read-only LDAP directory in the format required by Mac OS X. You may need to add, modify, or reorganize data in the read-only LDAP directory. Mac OS X cannot write data to a read-only LDAP directory, so you must make the necessary modifications by using tools on the server that hosts the read-only LDAP directory.

To prepare a read-only LDAP directory for Mac OS X:

1 Go to the server that hosts the read-only LDAP directory and configure it to support LDAP-based authentication and password checking.

2 Modify the LDAP directory's object classes and attributes as necessary to provide the data needed by Mac OS X.

For detailed specifications of the data required by Mac OS X directory services, see Appendix A, "Mac OS X Directory Data."

Populating LDAP Directories With Data for Mac OS X

After configuring access to LDAP directory domains and setting up their data mapping, you can populate them with records and data for Mac OS X. For directory domains that allow remote administration (read/write access), you can use the Workgroup Manager application, which is included with Mac OS X Server, as follows:

- Identify share points and shared domains that you want to mount automatically in a user's /Network directory (the Network globe in Finder windows). Use the Sharing module of Workgroup Manager. For instructions, see the file services administration guide.
- Define user records and group records and configure their settings. Use the Accounts module of Workgroup Manager. For instructions, see the user management guide.
- Define lists of computers that have the same preference settings and are available to the same users and groups. Use the Computers module of Workgroup Manager. For instructions, see the user management guide.

In all cases, click the small globe icon above the list of users and choose from the pop-up menu in Workgroup Manager to open the LDAP directory domain. If the LDAP directory is not listed in the pop-up menu, choose Other from this menu to select the LDAP directory.

Note: To add records and data to a read-only LDAP directory, you must use tools on the server that hosts the LDAP directory.

Accessing an Active Directory Domain

You can configure a server with Mac OS X Server or a computer with Mac OS X to access an Active Directory domain on a Windows 2000 or Windows 2003 server. For task descriptions and instructions, see:

- "Learning About the Active Directory Plug-in" (next)
- "Configuring Access to an Active Directory Domain" on page 102
- "Enabling or Disabling Active Directory Credential Caching" on page 104
- "Mapping the UID to an Active Directory Attribute" on page 105
- "Changing the Active Directory Groups That Can Administer the Computer" on page 105
- "Editing User Accounts and Other Records in Active Directory" on page 106

Alternative methods for accessing an Active Directory domain are appropriate for some networks. The alternatives include the following:

- "Setting Up LDAP Access to Active Directory Domains" on page 106

Learning About the Active Directory Plug-in

You can configure Mac OS X to access basic user account information in an Active Directory domain of a Windows 2000 or Windows 2003 server. What makes this possible is an Active Directory plug-in for Directory Access. This Active Directory plug-in is listed on the Services pane of Directory Access.

You do not need to make any schema modifications to the Active Directory domain to get basic user account information. You may need to change the default Access Control List (ACL) of specific attributes so that computer accounts will have the ability to read the properties. The Active Directory plug-in generates all attributes required for Mac OS X authentication from standard attributes in Active Directory user accounts. The plug-in also supports Active Directory authentication policies, including password changes, expiration, and forced change.

The Active Directory plug-in dynamically generates a unique user ID and a primary group ID based on the user account's Globally Unique ID (GUID) in the Active Directory domain. The generated user ID and primary group ID are always the same for each user account even if the account is used to log in to different Mac OS X computers. Alternatively, you can force the Active Directory plug-in to map the user ID to an Active Directory attribute that you specify.

When someone logs in to Mac OS X with an Active Directory user account, the Active Directory plug-in creates a home directory on the startup volume of the Mac OS X computer. The plug-in also tells Mac OS X to mount the user's Windows home directory (as specified in the Active Directory user account) to mount on the desktop as a share point. Using the Finder, the user can copy files between the Windows home directory in the Network globe and the Mac OS X home directory.

Each time a user logs in to Mac OS X with an Active Directory user name and password, the Active Directory plug-in can cache the authentication credentials on the Mac OS X computer. The user can log in again on the same computer when the computer is not connected to the network. You can enable or disable caching of credentials.

If the Active Directory schema has been extended to include Mac OS X record types (object classes) and attributes, the Active Directory plug-in automatically detects and accesses them. For example, the Active Directory schema could be modified using Windows administration tools to include Mac OS X Server managed client attributes. This schema modification would enable the Active Directory plug-in to support managed client settings made in the Preferences module of Workgroup Manager. Mac OS X clients assume full read access to attributes that are added to the directory. Therefore, it may be necessary to modify the ACL of those attributes to allow Computer accounts to read these added attributes.

The Active Directory plug-in automatically discovers all domains in an Active Directory forest. You can configure the plug-in to allow users from any domain in the forest to authenticate on a Mac OS X computer. The multi-domain authentication can also be disabled to allow only specific domains to be authenticated on the client.

The Active Directory plug-in fully supports Active Directory replication and failover. It discovers multiple domain controllers and determines the closest one. If a domain controller becomes unavailable, the plug-in automatically falls back to another nearby domain controller.

The Active Directory plug-in uses LDAP to access the Active Directory user accounts and Kerberos to authenticate them. The Active Directory plug-in does not use Microsoft's proprietary Active Directory Services Interface (ADSI) to get directory or authentication services.

Configuring Access to an Active Directory Domain

Using the Active Directory plug-in listed in Directory Access, you can configure Mac OS X to access basic user account information in an Active Directory domain on a Windows server. The Active Directory plug-in generates all attributes required for Mac OS X authentication. No changes to the Active Directory schema are required. Yet the Active Directory plug-in detects and accesses standard Mac OS X record types and attributes, such as the attributes required for Mac OS X client management, if the Active Directory schema has been extended to include them.

Important: An advanced option of the Active Directory plug-in allows you to map the Mac OS X unique user ID (UID) attribute to an appropriate attribute that has been added to the Active Directory schema. If you change the setting of this mapping option at a later date, users may lose access to previously created files.

To configure access to an Active Directory domain:

1. In Directory Access, click Services.
2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Select Active Directory in the list of services, then click Configure.
4. Enter the DNS names of the servers that host the Active Directory forest and domain of which the computer you're configuring will be a member.

   The administrator of the Active Directory domain can tell you the names of the forest and domain. If you have a single forest with a single domain, enter the same name for forest and domain.
5. Enter the Computer ID, which is the name that the computer you're configuring has been assigned in the Active Directory domain.

   If you're not sure what name to enter, ask the Active Directory domain administrator.

6 Click Bind, authenticate as a user who has rights to set up a connection to the Active Directory domain, and click OK.

Name and Password: You may be able to authenticate by entering the name and password of your Active Directory user account, or the Active Directory domain administrator may have to provide a name and password.

OU: Enter the organizational unit (OU) for the computer you're configuring.

7 Optionally, set the advanced options.

If the advanced options are hidden, click Show Advanced Options.

"**Cache last user logon for offline operation":** Select this option to enable the use of offline credentials without modifying the Active Directory schema. This is considered the default setting for users logging in to the computer. An equivalent capability is provided by managed client settings in an Open Directory domain and most LDAP directory domains. If a user account has actual managed client settings, then this option is ignored.

"**Authenticate in multiple domains":** Select this option to allow users from any domain within the forest to authenticate on this computer. If this option is unchecked, a list of specific domains within the forest will be presented when you configure a custom Authentication search policy so that you can add domains individually to the search policy.

"**Prefer this domain server":** Select this option to specify the DNS name of the server whose Active Directory domain you want used by default. If the server becomes unavailable in the future, the Active Directory plug-in automatically falls back to another nearby server in the forest. If this option is unselected, the Active Directory plug-in automatically determines the closest Active Directory domain in the forest.

"**Map UID to attribute":** If the Active Directory schema has been extended to store a unique UID (unique user ID) for each user—usually because the Active Directory server has already been configured to support UNIX computers—you can specify the attribute that stores the UID. If this option is unselected, a UID is automatically generated based on Active Directory's standard GUID attribute.

"**Allow administration by":** Select this option to specify a list of groups whose members are allowed to do administrative tasks on this computer (for example, install software). Use commas to separate group names in the list. For security, group names must be qualified by the domain name they are from (for example, ADS\Domain Admins,IL2\Domain Admins). This option is useful if you have desktop administrators who need administrative access but are not domain administrators.

If you want the computer to access the Active Directory domain you just configured, you must make sure Active Directory is enabled in the Services pane.

In addition, you must add the Active Directory domain to a custom search policy in the Authentication or Contacts pane of Directory Access.

- If you selected "Authenticate in multiple domains" in step 7, adding the Active Directory forest to a custom Authentication search policy enables this computer to authenticate users from any domain in the forest.
- If you deselected "Authenticate in multiple domains," you can add domains individually to the search policy.

For instructions, see "Enabling or Disabling Active Directory Service" on page 84 and "Defining Custom Search Policies" on page 89.

Enabling or Disabling Active Directory Credential Caching

Using Directory Access, you can enable or disable the use of offline authentication credentials from an Active Directory domain accessed by the Active Directory plug-in. A user with Active Directory credentials cached on a Mac OS X computer can log in while the computer is disconnected from the network. This credential caching does not require modifying the Active Directory schema. If the Active Directory schema has been extended to include Mac OS X managed client attributes, their mobile account setting will be used instead of the Active Directory plug-in's cached account setting.

To enable or disable caching of authentication credentials from an Active Directory domain:

1. In Directory Access, click Services.
2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Select Active Directory in the list of services, then click Configure.
4. If the advanced options are hidden, click Show Advanced Options.
5. Click "Cache last user logon for offline operation."

Specifying a Preferred Active Directory Server

Using Directory Access, you can specify the DNS name of the server whose Active Directory domain you want the Active Directory plug-in to access by default. If the server becomes unavailable in the future, the Active Directory plug-in automatically falls back to another nearby server in the forest. If this option is unselected, the Active Directory plug-in automatically determines the closest Active Directory domain in the forest.

To specify a server you prefer the Active Directory plug-in to access:

1. In Directory Access, click Services.
2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Select Active Directory in the list of services, then click Configure.
4. If the advanced options are hidden, click Show Advanced Options.
5. Select "Prefer this domain server" and enter the DNS name of the Active Directory server.

Mapping the UID to an Active Directory Attribute

Using Directory Access, you can specify an Active Directory attribute that you want the Active Directory plug-in to map to the unique user ID (UID) attribute of Mac OS X. Usually the Active Directory schema must be extended to include an attribute that's suitable for mapping to the UID.

If UID mapping is disabled, the Active Directory plug-in automatically generates a UID based on Active Directory's standard GUID attribute.

Important: If you change the mapping of the UID at a later date, users may lose access to previously created files.

To map the UID to an attribute in an extended Active Directory schema:

1. In Directory Access, click Services.
2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Select Active Directory in the list of services, then click Configure.
4. If the advanced options are hidden, click Show Advanced Options.
5. Select "Map UID to attribute" and enter the name of the Active Directory attribute you want mapped to the UID.

Changing the Active Directory Groups That Can Administer the Computer

Using Directory Access, you can grant administrator privileges to groups of user accounts accessed by the Active Directory plug-in. These Active Directory user accounts can be used to perform administrative tasks such as installing software on the Mac OS X computer that you are configuring.

To specify which groups of Active Directory user accounts have administrator privileges:

1. In Directory Access, click Services.
2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Select Active Directory in the list of services, then click Configure.
4. If the advanced options are hidden, click Show Advanced Options.
5. Select "Allow administration by" and enter the names of groups.

   Use commas to separate group names. For security, group names must be qualified by the domain name they are from (for example, ADS\Domain Admins,IL2\Domain Admins).

Editing User Accounts and Other Records in Active Directory

You can use Workgroup Manager to make changes to user accounts, group accounts, computer accounts, and other records in an Active Directory domain. You can also use Workgroup manager to delete records in an Active Directory domain. For instructions, see the user management guide.

To create user accounts, group accounts, computer accounts, and other records in an Active Directory domain, use the Microsoft Active Directory administration tools on a Windows server administration computer.

Setting Up LDAP Access to Active Directory Domains

Using Directory Access, you can set up an LDAPv3 configuration to access an Active Directory domain on a Windows server. An LDAPv3 configuration gives you full control over mapping of Mac OS X record types and attributes to Active Directory object classes, search bases, and attributes. Mapping of some important Mac OS X record types and attributes, such as the unique user ID (UID), requires extending the Active Directory schema.

An LDAPv3 configuration does not include many features of the Active Directory plug-in listed in Directory Access. These include dynamic generation of unique user ID and primary group ID; creation of a local Mac OS X home directory; automatic mounting of the Windows home directory; cached authentication credentials; discovery of all domains in an Active Directory forest; and support for Active Directory replication and failover. See "Learning About the Active Directory Plug-in" on page 101 for more information.

You can use Directory Access to create a configuration that specifies how Mac OS X accesses a particular LDAPv3 or LDAPv2 directory.

To create an Active Directory server configuration:

1. In Directory Access, click Services.
2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Select LDAPv3 in the list of services, then click Configure.
4. If the list of server configurations is hidden, click Show Options.
5. Click New and enter a name for the configuration.
6. Press Tab and enter the Active Directory server's DNS name or IP address.
7. Click the pop-up menu next to the DNS name or IP address and choose Active Directory.
8. Enter the search base for the Active Directory domain, then click OK.
9. Select the SSL checkbox if you want Open Directory to use Secure Sockets Layer (SSL) for connections with the Active Directory server.

The Active Directory mapping template for an LDAPv3 configuration maps some Mac OS X record types and attributes to object classes and attributes that are not part of a standard Active Directory schema. You can change the mappings defined by the template or extend the Active Directory schema. (Alternatively, you may be able to access your Active Directory domain via the Active Directory plug-in instead of LDAPv3.)

If you want the computer to access the Active Directory domain for which you just created an LDAPv3 configuration, you must add the directory to a custom search policy in the Authentication or Contacts pane of Directory Access. You must also make sure LDAPv3 is enabled in the Services pane. For instructions, see "Enabling or Disabling LDAP Directory Services" on page 85 and "Defining Custom Search Policies" on page 89.

Accessing an NIS Domain

Using Directory Access, you create a configuration that specifies how Mac OS X accesses an NIS domain.

To create a configuration for accessing an NIS domain:

1. In Directory Access, click Services.
2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Select "BSD Flat File and NIS" in the list of services, then click Configure.
4. Enter the NIS domain name.

Include the NIS server's hostname or IP address if it is required for security or the server is not on the same subnet as the computer you're configuring.

5 Optionally, enter the DNS name or the IP address of the server or servers where the NIS domain resides.

If you don't specify any servers, NIS uses a broadcast protocol to discover an NIS server on the subnet.

6 Create a custom search policy that includes the NIS domain.

In a custom search policy, the NIS domain is listed as /BSD/*domain*, where *domain* is what you entered in step 4.

For instructions, see "Defining Custom Search Policies" on page 89.

Using BSD Configuration Files

Historically, UNIX computers have stored administrative data in configuration files such as /etc/master.passwd, /etc/group, and /etc/hosts. Mac OS X is based on a BSD version of UNIX, but normally gets administrative data from directory domains for the reasons discussed at the beginning of this chapter.

In Mac OS X version 10.2 and later (including Mac OS X Server version 10.2 and later), Open Directory can retrieve administrative data from BSD configuration files. This capability enables organizations that already have BSD configuration files to use copies of the existing files on Mac OS X computers. BSD configuration files can be used alone or in conjunction with other directory domains.

To use BSD configuration files:

1 Set up each BSD configuration file with the data required by Mac OS X directory services.

See "Setting Up Data in BSD Configuration Files" on page 109 for instructions.

2 Create a custom search policy that includes the BSD configuration files domain.

In a custom search policy, the BSD configuration files domain is listed as /BSD/local.

For instructions, see "Defining Custom Search Policies" on page 89.

Mac OS X Server version 10.3 supports a fixed set of BSD configuration files. You can't specify which configuration files to use, nor can you map their contents to Mac OS X record types and attributes.

Setting Up Data in BSD Configuration Files

If you want a Mac OS X computer to get administrative data from BSD configuration files, the data must exist in the files and must be in the format required by Mac OS X. You may need to add, modify, or reorganize data in the files. Workgroup Manager cannot make changes to data in BSD configuration files, so you must make the necessary modifications by using a text editor or other tools.

For detailed specifications of the data required by Mac OS X directory services, see Appendix A, "Mac OS X Directory Data."

Accessing Legacy NetInfo Domains

Shared directory domains that were created with Mac OS X Server versions earlier than 10.3 used the NetInfo protocol (and optionally the LDAPv3 protocol). NetInfo can still be used to access these legacy NetInfo domains. This means:

- Any Mac OS X Server or other Mac OS X computer can access a shared NetInfo domain hosted by a server that has been upgraded to Mac OS X Server version 10.3.
- Mac OS X Server version 10.3 and Mac OS X version 10.3 can access any existing shared NetInfo domain.

Note: You cannot create a new shared NetInfo domain with Mac OS X Server version 10.3 except by using command-line utilities. If you use Server Assistant or Server Admin to set up Mac OS X Server version 10.3 to be an Open Directory master (that is, to host a shared LDAP directory), other computers can access this new shared directory only by using LDAP.

For instructions on setting up a Mac OS X Server or other Mac OS X computer to access a shared NetInfo domain, see "About NetInfo Binding" and "Configuring NetInfo Binding," following this topic.

Expert system administrators can manage NetInfo domains as follows:

- Create machine records for broadcast binding to an existing shared NetInfo domain. For instructions, see "Adding a Machine Record to a Parent NetInfo Domain" on page 111.
- Configure a shared NetInfo domain to use a particular port number instead of a dynamically assigned port number. For instructions, see "Configuring Static Ports for Shared NetInfo Domains" on page 112.

About NetInfo Binding

When a Mac OS X computer starts up, it can bind its local directory domain to a shared NetInfo domain. The shared NetInfo domain can bind to another shared NetInfo domain. The binding process creates a hierarchy of NetInfo domains.

A NetInfo hierarchy has a branched structure. Local domains at the bottom of the hierarchy bind to shared domains, which can in turn bind to other shared domains, and so on. Each domain binds to only one shared domain, but a shared domain can have any number of domains bind to it. A shared domain is called a *parent* domain, and each domain that binds to it is a *child* domain. At the top of the hierarchy is one shared domain that doesn't bind to another domain; this is the root domain.

A Mac OS X computer can bind to a shared NetInfo domain by using any combination of three protocols: static, broadcast, or DHCP.

- With static binding, you specify the address and NetInfo tag of the shared NetInfo domain. This is most commonly used when the shared domain's computer is not on the same IP subnet as the computer that needs to access it.
- With DHCP binding, a DHCP server automatically supplies the address and NetInfo tag of the shared NetInfo domain. To use DHCP binding, the DHCP server must be configured to supply a NetInfo parent's address and tag.
- With broadcast binding, the computer locates a shared NetInfo domain by sending out an IP broadcast request. The computer hosting the shared domain responds with its address and tag.

  For broadcast binding, both computers must be on the same IP subnet or on a network that is configured for IP broadcast forwarding.

  The parent domain must have the NetInfo tag "network."

  The parent domain must have a machine record for each computer that can bind to it with broadcast binding.

If you configure a computer to use multiple binding protocols and a parent is not located with one protocol, another one is used. The protocols are used in this order: static, DHCP, broadcast.

Configuring NetInfo Binding

Using Directory Access, you can configure Mac OS X or Mac OS X Server to bind to a parent NetInfo domain by using the static, broadcast, or DHCP protocols in any combination. The computer attempts to bind to a parent NetInfo domain when the computer starts up.

Note: If your network has no shared NetInfo domain, setting a computer to bind to a parent NetInfo domain will cause delays when the computer starts up.

To bind a Mac OS X computer to a shared NetInfo domain:

1. In Directory Access, click Services.
2. If the lock icon is locked, click it and type the name and password of an administrator.
3. Select NetInfo in the list of services, then click Configure.
4. Select the binding protocols that you want the computer to use.

   For broadcast binding, select "Attempt to connect using Broadcast protocol."

   For DHCP binding, select "Attempt to connect using DHCP protocol."

   For static binding, select "Attempt to connect to a specific NetInfo server." Then enter the IP address of the parent domain's computer in the Server Address field and the parent domain's NetInfo tag in the Server Tag field.
5. Click OK, then click Apply.
6. Restart the computer.

Adding a Machine Record to a Parent NetInfo Domain

Mac OS X computers can bind their directory domains to a parent NetInfo domain by using broadcast binding. The parent NetInfo domain must have a machine record for each Mac OS X computer that can bind to it with broadcast binding. You can create a machine record with the NetInfo Manager application.

To add a machine record to a parent NetInfo domain:

1. Open NetInfo Manager on the computer where the parent domain resides, then open the domain.
2. Click the lock and authenticate using the name and password of an administrator for the directory domain.
3. Select the "machines" directory in the Directory Browser list.
4. Choose New Subdirectory from the Directory menu.
5. Double-click new_directory in the lower list and enter the DNS name of the child computer.
6. Choose New Property from the Directory menu.
7. In the lower list, change new_property to ip_address and change new_value to the IP address of the child computer.

8 Choose New Property from the Directory menu.

9 Change new_property to "serves" and then change new_value to the name and NetInfo tag of the child's local domain, using a "/" to separate the name and the tag.

For example, you would change new_value to marketing.demo/local for the local domain of the computer named marketing.demo.

10 Choose Save Changes from the Domain menu, then click Update This Copy.

Configuring Static Ports for Shared NetInfo Domains

By default, Mac OS X dynamically selects a port in the range 600 through 1023 when it accesses a shared NetInfo domain. You can configure a shared domain for NetInfo access over specific ports. Use the NetInfo Manager application to do this.

To configure specific ports for NetInfo access to shared domains:

1 Open NetInfo Manager on the computer where the shared domain resides, then open the domain.

2 Click the lock and authenticate using the name and password of an administrator for the directory domain.

3 Select the "/" directory in the Directory Browser list.

4 To change the value of an existing port property, double-click the value in the Value(s) column and make the change.

5 To delete a port property, select it and choose Delete from the Edit menu.

6 To add a property, choose New Property from the Directory menu and proceed as follows.

If you want to use one port for both TCP and UDP packets, double-click new_property and change it to "port." Then change new_value to the port number you want to use.

If you want separate TCP and UDP ports, double-click new_property and change it to tcp_port. Then change new_value to the TCP port number you want to use. Next double-click new_property and change it to udp_port. This time, change new_value to the UDP port number you want to use.

Setting Up Directory Access on a Remote Server

You can use the Directory Access application on your computer to set up and manage how a server with Mac OS X Server accesses directory services and discovers network services. Your computer must have version 10.2 or later of Mac OS X or Mac OS X Server, and the remote server must have Mac OS X Server version 10.2.

To configure directory access on a remote server:

1. In Directory Access on your computer, choose Connect from the Server menu.
2. Enter the connection and authentication information for the server that you want to configure, then click Connect.

   Address: enter the DNS name or IP address of the server that you want to configure.

   User Name: enter the user name of an administrator on the server.

   Password: enter the password for the user name you entered.
3. Click the Services, Authentication, and Contacts tabs and change settings as needed.

   All the changes you make affect the remote server to which you connected in the foregoing steps.
4. When you finish configuring the remote server, choose Disconnect from the Server menu on your computer.

Maintenance and Problem Solving

8

You can monitor Open Directory services, view and edit raw data from Open Directory domains, and back up Open Directory files. You can also solve some common Open Directory problems.

For descriptions and instructions of Open Directory maintenance tasks, see:

- "Monitoring Open Directory" (next)
- "Directly Viewing and Editing Directory Data" on page 116
- "Backing Up Open Directory Files" on page 118
- "Restoring Open Directory Files" on page 120

For descriptions of some problems that may be related to Open Directory services, and for fixes for the problems, see:

- "Solving Directory Access Problems" on page 121
- "Solving Authentication Problems" on page 122

Monitoring Open Directory

You can view Open Directory status and logs, and you can inspect Open Directory authentication logs for suspicious activities.

Viewing Open Directory Status and Logs

You can use the Server Admin application to view status information and logs for Open Directory services. The following logs are available:

- Directory services server log
- Directory services error log
- Lookup log
- NetInfo log
- LDAP log
- Password service server log
- Password service error log

To see directory services status or logs:

1. Open Server Admin and select Open Directory for a server in the Computers & Services list.
2. Click Overview to see status information.
3. Click Logs and use the Show pop-up menu to choose the log you want to see.

Monitoring Open Directory Authentication

You can use the password service logs, visible using Server Admin, to monitor failed login attempts for suspicious activity.

Open Directory logs all failed authentication attempts, including IP addresses that generate them. Periodically review the logs to determine whether there are a large number of failed trials for the same password ID, indicating that somebody might be generating login guesses.

To see Open Directory authentication logs:

1. Open Server Admin and select Open Directory for a server in the Computers & Services list.
2. Click Logs and choose a password service log from the Show pop-up menu.

Directly Viewing and Editing Directory Data

You can view or edit raw directory data by using the Inspector in Workgroup Manager. The Inspector allows you to see directory data not otherwise visible in Workgroup Manager or any other application. Furthermore, the Inspector allows you to edit directory data that you cannot otherwise change in Workgroup Manager or other applications. For example, you can use the Inspector to change a user's short name.

Showing the Directory Inspector

You can make the Inspector visible in Workgroup Manager by selecting an option in Workgroup Manager Preferences. Then you can use the Inspector to view or edit raw directory data.

Important: Changing raw data in a directory can have unexpected and undesirable consequences. You could inadvertently incapacitate users or a computers, or you could unintentionally authorize users to access more resources.

To make the Inspector visible:

1. Open Workgroup Manager and choose Workgroup Manager > Preferences.
2. Select "Show 'All Records' tab and inspector" and click OK.
3. Click the Users button, Group button, or Computers button (on the left), then click Inspector (on the right).

You can also click the All Records button, which is next to the Computers button, and choose a record type from the pop-up menu at the top of the list. The pop-up menu lists all standard record types that exist in the directory domain. You can also choose Native from the pop-up menu and type the name of a native record type into the box that appears below the pop-up menu. The list displays all records, including predefined records, of the currently chosen record type.

Hiding the Directory Inspector

If the Inspector is visible in Workgroup Manager, you can hide it by changing an option in Workgroup Manager Preferences.

To hide the Inspector:

1. Open Workgroup Manager and choose Workgroup Manager > Preferences.
2. Deselect "Show 'All Records' tab and inspector" and click OK.

Changing a User's Short Name

You can use the Inspector in Workgroup Manager to change a user's short name or short names, including a user's first short name.

Important: Changing a user's first short name can have unexpected and undesirable consequences. Other services use each user's first short name as a unique and persistent identifier. For example, changing a user's first short name does not rename the user's home directory. The user has the same home directory (even though its name doesn't match the user's new first short name) unless the user happens to access his or her home directory through a group membership. Changing a user's first short name effectively revokes all group memberships, because each group's membership is stored as a list of members' first short names.

To change the short name of a user account:

1. Open Workgroup Manager and make the Inspector visible if it is hidden.
2. Click the Accounts button, then click the Users button.
3. Open the directory domain that contains the user account whose short name you want to change, and authenticate as an administrator of the domain.

   To open a directory domain, click the small globe icon above the list of users and choose from the pop-up menu.
4. Select the account whose short name you want to change, then click Inspector (on the right).

5 Locate RecordName in the list of attributes, and if a triangle appears next to RecordName, click the triangle to see all RecordName values.

The RecordName attribute stores the user's short name or names.

6 Double-click the RecordName value that is the short name you want to change, then type another short name and press Return.

You can also click a RecordName value, and then click Edit to change the value in an editing sheet.

7 Click Save.

Backing Up Open Directory Files

To back up an Open Directory master, you need to back up its shared LDAP directory domain, its configuration files, and its Open Directory Password Server database. You can back up the server's local directory, which is a NetInfo domain, as well.

You can do a hot backup of an Open Directory master. That is, you can back up an Open Directory master while it is in service.

Although you can back up an Open Directory replica, there is no real need to do so. In fact, restoring a replica can be dangerous because it puts an outdated copy of the master on the network. Since a replica is a copy of the master, the master effectively backs up the replica. If a replica develops a problem, you can just change its role to standalone server. Then set up the server as though it were a brand new server, with a new host name, and set it up as a replica of the same master as before. Therefore, if you have a reliable backup of the master, you effectively have a backup of all replicas of the master.

Important: Be sure to perform the backup procedure without much delay between steps. By minimizing delays between steps, you ensure that the backed up LDAP directory, configuration files, and password storage database are as in sync as possible.

Important: Carefully safeguard the backup media that contains a copy of the Open Directory Password database. The backup contains passwords of all users who have an Open Directory password, both in the shared LDAP directory domain and in the local NetInfo directory domain. Your security precautions for the backup media should be just as stringent as for the Open Directory master server.

To do a hot backup of an Open Directory master:

1 Open a Terminal session as root.

2 Type the following command and press Return.

```
slapcat -l backup.ldif
```

This use of `slapcat` saves the complete contents of the LDAP directory as a raw LDIF dump in a text file named backup.ldif. You can specify a different filename and a pathname. The file you specify contains all user records, group records, computer records, and so on. (The file does not contain passwords for user records whose password type is Open Directory. These passwords are not stored in the LDAP directory database. They are stored separately in the Open Directory Password Server database.)

3 Make a copy of the /etc/openldap folder.

This folder contains files that determine the setup of the LDAP directory domain, including schema files.

4 If your LDAP server uses SSL, make a copy of the server certificate file, LDAP server's private key file, and the certificate authority (CA) certificate file.

5 Type the following commands, pressing Return after each.

```
mkdir -p backup_folder_pathname
mkpassdb -backupdb backup_folder_pathname
```

The Open Directory Password backup folder, located at `backup_folder_pathname`, will contain backup copies of all Open Directory Password Server files, including the database.

Carefully safeguard the Open Directory Password Server backup folder! It contains the passwords of all users who have an Open Directory password, both in the shared LDAP directory domain and in the local NetInfo directory domain. Keep the backup media as secure as the Open Directory master server.

6 Optionally, make a copy of the Library/Preferences/DirectoryService folder.

Files in this folder specify the server's search policies and specify how the server accesses its LDAP directory. The folder may contain additional files that specify how the server access other directory domains.

7 Optionally, make a copy of the /etc/hostconfig file.

8 If you want to back up the local NetInfo directory domain, type the following command and press Return:

```
nidump -r / . > local.dump
```

This use of `nidump` saves the entire contents of the local NetInfo domain as a raw text file named local.dump. You can specify a different filename and a pathname.

Note: If all user accounts have an Open Directory password and the Open Directory Password Server stops working, you can log in as root. The root user account in the local directory domain has a shadow password, which is not stored in the Open Directory Password Server database.

Restoring Open Directory Files

To restore an Open Directory master from backup files, you need to restore its shared LDAP directory domain, its configuration files, and its Open Directory Password Server database. You might want to restore the server's local directory, which is a NetInfo domain, as well.

To restore an Open Directory master from backup files:

1. If you have to recover from a catastrophic failure by reinstalling the Mac OS X Server software, set the directory usage to standalone server when Server Assistant takes you through the initial configuration of the server.

   If Mac OS X Server was reinstalled but the directory usage was initially configured to something other than standalone server in Server Assistant, open Server Admin, select Open Directory, click Settings, click General, and change Role to Standalone Server.

2. Open a Terminal session as root.

3. Restore the /etc/openldap folder from a backup copy.

   This folder contains files that determine the setup of the LDAP directory domain, including schema files.

4. Restore the /etc/hostconfig file from a backup copy.

   If you don't have a backup copy of the /etc/hostconfig file, you can edit the existing file. The file needs to contain the following line:

   ```
   LDAPSERVER=-YES-
   ```

   If the restored LDAP server will use SSL, the /etc/hostconfig file must also contain the following line:

   ```
   LDAPSSL=-YES-
   ```

5. If the restored LDAP server will use SSL, restore a copy of the server certificate file, LDAP server's private key file, and the certificate authority (CA) certificate file.

   Restore these files to their previous locations. The correct pathnames are specified in /etc/openldap/slapd_macosxserver.conf or /etc/openldap/slapd.conf.

6. Make sure the folder that will contain the LDAP database exists and is empty.

   This folder's pathname, which by default is /var/db/openldap/openldap-data/, is specified in /etc/openldap/slapd_macosxserver.conf or /etc/openldap/slapd.conf.

7. Type the following command and press Return.

   ```
   slapadd -c -l backup.ldif
   ```

   This use of `slapadd` adds user records, group records, computer records, and so on to the LDAP directory from the raw LDIF text file named backup.dif. You can specify a different filename and a pathname. (Adding LDAP records from the LDIF text file does not restore passwords for user records whose password type is Open Directory. These passwords are not stored in the LDAP directory database. They are stored separately in the Open Directory Password Server database.)

8 Type the following command and press Return.

```
mkpassdb -mergedb backup_folder_pathname
```

This use of `mkpassdb` adds all of the passwords from the Open Directory Password Server backup folder, located at `backup_folder_pathname`, into the server's existing Open Directory Password Server database. (The server has an existing Open Directory Password Server for its local directory domain.)

9 Type the following command and press Return:

```
/System/Library/StartupItems/LDAP/LDAP start
```

This command starts the LDAP server.

10 If you want to restore the local NetInfo directory domain, type the following command and press Return:

```
niload -r / . < local.dump
```

This use of `niload` loads the entire contents of the local NetInfo domain from a raw text file named local.dump.

11 Restore the Library/Preferences/DirectoryService folder from a backup copy and restart the server, or use Directory Access to configure access to the restored LDAP directory and add it to a custom search policy.

Files in the Library/Preferences/DirectoryService folder specify the server's search policies and specify how the server accesses its LDAP directory. The folder may contain additional files that specify how the server access other directory domains. If you restore this folder from a backup copy, you need to restart the server so that Open Directory recognizes the restored search policies and server access configurations.

If you don't have a backup copy of this folder, use Directory Access to create an LDAPv3 configuration for the loopback IP address (127.0.0.1), and add this to a custom authentication search policy. For instructions, see "Configuring Access to an LDAP Directory" on page 92 and "Defining Custom Search Policies" on page 89.

Solving Directory Access Problems

Problems accessing directory services during startup can have several causes.

A Delay Occurs During Startup

If Mac OS X or Mac OS X Server experiences a startup delay while a message about NetInfo, LDAP, or directory services appears above the progress bar, the computer could be trying to access a NetInfo domain or LDAP directory that is not available on your network.

- Use Directory Access to make sure the NetInfo and LDAP configurations are correct.
- Use the Network pane of System Preferences to make sure the computer's network location and other network settings are correct.
- Inspect the physical network connection for faults.

Solving Authentication Problems

You can solve some common problems with authentication services.

A User's Password Can't Be Modified

Before you can modify the password of a user whose password is authenticated by Open Directory, you must be an administrator of the directory domain in which the user's record resides. In addition, your user account must be configured for Open Directory authentication.

A User Can't Authenticate for VPN Service

Users whose accounts are stored on a server with Mac OS X Server version 10.2 can't authenticate for VPN service provided by Mac OS X Server version 10.3 or later. VPN service requires the MS-CHAPv2 authentication method, which isn't supported in Mac OS X Server version 10.2. To enable the affected users to log in, you can move their user accounts to a server with Mac OS X Server version 10.3 or later. Alternatively, you can upgrade the older server to Mac OS X Server version 10.3 or later.

A User's Password Type Can't Be Changed to Open Directory

Before you can modify a user account to use Open Directory authentication, you must be an administrator of the directory domain in which the user's record resides. In addition, your user account must be configured for Open Directory authentication.

Kerberos Users Can't Authenticate

When a user or service that uses Kerberos experiences authentication failures, try these techniques:

- Kerberos behavior is based on encrypted time stamps. If there's more than a five-minute difference between the KDC, client, and service computers, authentication may fail. Make sure that the clocks for all computers are synchronized using a network time server.
- If Kerberos is being used, make sure that Kerberos authentication is enabled for the service in question.
- If a Kerberos server used for password validation is not available, reset the user's password to use a server that is available.
- Make sure that the server providing the Kerberized service has access to directory domains containing accounts for users who are authenticated using Kerberos. One way to do this is to use a shared directory domain on the KDC server that hosts user records that correspond to all the user principals.
- Refer to the KDC log (kdc.log) for information that can help you solve problems. Incorrect setup information such as wrong configuration file names can be detected using the logs.
- Make sure all your configuration files are complete and correct. For example, make sure the keytab file on your server has the principals of interest in it.

Resetting an Administrator Password

Using the Mac OS X Server installation disc, you can change the password of a user account that has administrator privileges, including the System Administrator (root or superuser) account.

Important: Because a user with the installation disc can gain unrestricted access to your server, you should restrict physical access to the server hardware.

To change the password of an administrator account:

1. Start up from the Mac OS X Server "Install Disc 1."
2. When the Installer appears, choose Installer > Reset Password.
3. Select the hard disk volume that contains the administrator account whose password you want to reset.
4. Choose the administrator account from the pop-up menu, enter a new password, and click Save.

   System Administrator is the root user (superuser) account. Don't confuse this account with a normal administrator account.

   Avoid changing the password of any predefined user account. For more information on predefined user accounts, see the user management guide.

   Note: This procedure changes the password of the administrator account stored in the server's local directory domain. It does not change the password of an administrator account stored in the server's shared directory domain, if the server has one.

   If you know the password of any administrator account that's stored in the local domain, you can change the password of any other administrator account in the local directory domain by using Workgroup Manager instead of this procedure. For instructions, see the user management guide.

Mac OS X Directory Data

A Appendix

Knowing the Open Directory LDAP schema and the record types and attributes in Mac OS X directory domains can help you map to other directory domains and import or export user and group accounts.

For specifications of Open Directory extensions to LDAP schema, mappings of Open Directory attributes to LDAP and Active Directory attributes, and the standard attributes in various types of records, see:

Use these specifications for reference when you:

- Map object classes and attributes of non-Apple LDAP directories or Active Directory domains to Open Directory record types and attributes, as described in Chapter 7, "Managing Directory Access."
- Import or export user or group accounts to an Open Directory domain, as described in the user management guide.

Open Directory Extensions to LDAP Schema

The schema for the Open Directory LDAP directories is based on the de facto standard attributes and object classes defined in the following Request for Comments documents of the Internet Engineering Task Force (RFCs of the IETF):

- RFC 2307 "An Approach for Using LDAP as a Network Information Service"
- RFC 2798 "Definition of the inetOrgPerson LDAP Object Class"

These RFCs are available at the IETF website: www.ietf.org/rfc.html.

The attributes and object classes defined in these RFCs form the basis of the Open Directory LDAP schema.

The extended schema for Open Directory LDAP directories includes the attributes and object classes defined in this section.

Note: Apple may extend the Open Directory LDAP schema in the future, for example, to support new versions of Mac OS X and Mac OS X Server. The latest schema is available in text files on a computer with Mac OS X Server installed. The schema files are in the /etc/openldap/schema/ directory. In particular, the apple.schema file contains the latest schema extensions for Open Directory LDAP directories.

Object Classes in Open Directory LDAP Schema

This section defines the Open Directory LDAP object classes that extend the standard LDAP schema.

Container Structural Object Class

Container is a structural object class which is used for the top level record containers such as cn=users, cn=groups, and cn=mounts. There is no Directory Services analog to this object class, but the container name is part of the search base for each record type.

```
objectclass (
  1.2.840.113556.1.3.23
  NAME 'container'
  SUP top
  STRUCTURAL
  MUST ( cn ) )
```

User Object Class

The apple-user object class is an auxiliary class used to store Mac OS X specific attributes which are not part of inetOrgPerson or posixAccount. This object class is used with kDSStdRecordTypeUsers records.

```
objectclass (
  1.3.6.1.4.1.63.1000.1.1.2.1
  NAME 'apple-user'
  SUP top
  AUXILIARY
  DESC 'apple user account'
  MAY (  apple-user-homeurl $ apple-user-class $
         apple-user-homequota $ apple-user-mailattribute $
         apple-user-printattribute $ apple-mcxflags $
         apple-mcxsettings $ apple-user-adminlimits $
         apple-user-picture $ apple-user-authenticationhint $
         apple-user-homesoftquota $ apple-user-passwordpolicy $
         apple-keyword $ apple-generateduid $ authAuthority $
         acctFlags $ pwdLastSet $ logonTime $ logoffTime $
         kickoffTime $ homeDrive $ scriptPath $ profilePath $
         userWorkstations $ smbHome $ rid $ primaryGroupID ) )
```

Group Auxiliary Object Class

The apple-group object class is an auxiliary class used to store Mac OS X specific attributes which are not part of posixGroup. This object class is used with kDSStdRecordTypeGroups records.

```
objectclass (
  1.3.6.1.4.1.63.1000.1.1.2.14
  NAME 'apple-group'
  SUP top
  AUXILIARY
  DESC 'group account'
  MAY (  apple-group-homeurl $
         apple-group-homeowner $
         apple-mcxflags $
         apple-mcxsettings $
         apple-group-realname $
         apple-user-picture $
         apple-keyword $
         apple-generateduid ) )
```

Machine Auxiliary Object Class

```
objectclass (
  1.3.6.1.4.1.63.1000.1.1.2.3
  NAME 'apple-machine'
  SUP top
  AUXILIARY
  MAY (  apple-machine-software $
         apple-machine-hardware $
         apple-machine-serves $
         apple-machine-suffix ) )
```

Mount Object Class

```
objectclass (
  1.3.6.1.4.1.63.1000.1.1.2.8
  NAME 'mount'
  SUP top STRUCTURAL
  MUST ( cn )
  MAY (  mountDirectory $
         mountType $
         mountOption $
         mountDumpFrequency $
         mountPassNo ) )
```

Printer Object Class

```
objectclass (
  1.3.6.1.4.1.63.1000.1.1.2.9
  NAME 'apple-printer'
  SUP top STRUCTURAL
  MUST ( cn )
  MAY (  apple-printer-attributes $
         apple-printer-lprhost $
         apple-printer-lprqueue $
         apple-printer-type $
         apple-printer-note ) )
```

Computer Object Class

```
objectclass (
  1.3.6.1.4.1.63.1000.1.1.2.10
  NAME 'apple-computer'
  DESC 'computer'
  SUP top STRUCTURAL
  MUST ( cn )
  MAY (  apple-realname $
         description $
```

```
        macAddress $
        apple-computer-list-groups $
        apple-mcxflags $
        apple-mcxsettings $
        apple-xmlplist $
        authAuthority $
        uidNumber $ gidNumber $ apple-generateduid $
        acctFlags $ pwdLastSet $ logonTime $
        logoffTime $ kickoffTime $ rid $ primaryGroupID ) )
```

ComputerList Object Class

```
objectclass (
  1.3.6.1.4.1.63.1000.1.1.2.11
  NAME 'apple-computer-list'
  DESC 'computer list'
  SUP top STRUCTURAL
  MUST ( cn )
  MAY (  apple-mcxflags $
         apple-mcxsettings $
         apple-computer-list-groups $
         apple-computers $
         apple-keyword ) )
```

Configuration Object Class

```
objectclass (
  1.3.6.1.4.1.63.1000.1.1.2.12
  NAME 'apple-configuration'
  DESC 'configuration'
  SUP top STRUCTURAL
  MAY (  cn $ apple-config-realname $
         apple-data-stamp $ apple-password-server-location $
         apple-password-server-list $ apple-ldap-replica $
         apple-ldap-writable-replica $ apple-keyword $
         apple-kdc-authkey $ apple-kdc-configdata $ apple-xmlplist )
    )
```

Preset Computer List Object Class

```
objectclass (
  1.3.6.1.4.1.63.1000.1.1.2.13
  NAME 'apple-preset-computer-list'
  DESC 'preset computer list'
  SUP top STRUCTURAL
  MUST ( cn )
  MAY (  apple-mcxflags $
         apple-mcxsettings $
         apple-keyword ) )
```

Preset Group Object Class

```
objectclass (
  1.3.6.1.4.1.63.1000.1.1.3.14
  NAME 'apple-preset-group'
  DESC 'preset group'
  SUP top STRUCTURAL
  MUST ( cn )
  MAY (  memberUid $
         gidNumber $
         apple-group-homeurl $
         apple-group-homeowner $
         apple-mcxflags $
         apple-mcxsettings $
         apple-group-realname $
         apple-keyword ) )
```

Preset User Object Class

```
objectclass (
  1.3.6.1.4.1.63.1000.1.1.2.15
  NAME 'apple-preset-user'
  DESC 'preset user'
  SUP top STRUCTURAL
  MUST ( cn )
  MAY (  uid $
         memberUid $
         gidNumber $
         homeDirectory $
         apple-user-homeurl $
         apple-user-homequota $
         apple-user-homesoftquota $
         apple-user-mailattribute $
         apple-user-printattribute $
```

```
        apple-mcxflags $
        apple-mcxsettings $
        apple-user-adminlimits $
        apple-user-passwordpolicy $
        userPassword $
        apple-user-picture $
        apple-keyword $
        loginShell $
        shadowLastChange $
        shadowExpire $
        authAuthority $
        apple-preset-user-is-admin ) )
```

Authentication Authority Object Class

```
objectclass (
  1.3.6.1.4.1.63.1000.1.1.2.16
  NAME 'authAuthorityObject'
  SUP top AUXILIARY
  MAY (  authAuthority ) )
```

Server Assistant Configuration Object Class

```
objectclass (
  1.3.6.1.4.1.63.1000.1.1.2.17
  NAME 'apple-serverassistant-config'
  SUP top AUXILIARY
  MUST ( cn )
  MAY (  apple-xmlplist ) )
```

Location Object Class

```
objectclass (
  1.3.6.1.4.1.63.1000.1.1.2.18
  NAME 'apple-location'
  SUP top AUXILIARY
  MUST ( cn )
  MAY (  apple-dns-domain $ apple-dns-nameserver ) )
```

Attributes in Open Directory LDAP Schema

This section defines the Open Directory LDAP attributes that extend the standard LDAP schema.

User Attributes

apple-user-homeurl

Used to store home directory information in the form of a URL and path. This maps to the kDS1AttrHomeDirectory attribute type in Directory Services.

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.1.6
  NAME 'apple-user-homeurl'
  DESC 'home directory URL'
  EQUALITY caseExactIA5Match
  SUBSTR caseExactIA5SubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
```

apple-user-class

Unused.

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.1.7
  NAME 'apple-user-class'
  DESC 'user class'
  EQUALITY caseExactIA5Match
  SUBSTR caseExactIA5SubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
```

apple-user-homequota

Used to specify the home directory quota in kilobytes.

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.1.8
  NAME 'apple-user-homequota'
  DESC 'home directory quota'
  EQUALITY caseExactIA5Match
  SUBSTR caseExactIA5SubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
```

apple-user-mailattribute
Stores mail-related settings as XML.

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.1.9
  NAME 'apple-user-mailattribute'
  DESC 'mail attribute'
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
```

apple-mcxflags
Used to store managed client information. This attribute can be found in user, group, computer, and computer list records.

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.1.10
  NAME 'apple-mcxflags'
  DESC 'mcx flags'
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
```

apple-mcxsettings
Used to store managed client information. This attribute can be found in user, group, computer, and computer list records.

```
#attributetype (
# 1.3.6.1.4.1.63.1000.1.1.1.1.11
# NAME 'apple-mcxsettings'
# DESC 'mcx settings'
# EQUALITY caseExactMatch
# SUBSTR caseExactSubstringsMatch
# SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.1.16
  NAME ( 'apple-mcxsettings' 'apple-mcxsettings2' )
  DESC 'mcx settings'
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

apple-user-picture
Stores a file system path to the picture to use for this user record when displayed in login window. This is used when the network user shows in the login window scrolling list (in managed networks).

Users can modify their own pictures by default.

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.1.12
  NAME 'apple-user-picture'
  DESC 'picture'
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
```

apple-user-printattribute
Stores print quota settings as an XML plist.

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.1.13
  NAME 'apple-user-printattribute'
  DESC 'print attribute'
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
```

apple-user-adminlimits
This attribute is used by Workgroup Manager to store an XML plist describing the abilities of an administrator. These settings are respected and updated by Workgroup Manager but do not affect other parts of the system.

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.1.14
  NAME 'apple-user-adminlimits'
  DESC 'admin limits'
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
```

apple-user-authenticationhint

The apple-user-authenticationhint is used by login window to provide a hint if the user logs in incorrectly three times.

By default each user can update their own authentication hint.

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.1.15
  NAME 'apple-user-authenticationhint'
  DESC 'password hint'
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
```

apple-user-homesoftquota

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.1.17
  NAME 'apple-user-homesoftquota'
  DESC 'home directory soft quota'
  EQUALITY caseExactIA5Match
  SUBSTR caseExactIA5SubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
```

apple-user-passwordpolicy

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.1.18
  NAME 'apple-user-passwordpolicy'
  DESC 'password policy options'
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
```

apple-keyword

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.1.19
  NAME ( 'apple-keyword' )
  DESC 'keywords'
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

apple-generateduid

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.1.20
  NAME ( 'apple-generateduid' )
  DESC 'generated unique ID'
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
```

apple-user-homeDirectory

This is not used by the Open Directory Server, but provided as an example OID and attribute to use as an alternative to the homeDirectory attribute from RFC 2307. This is primarily of interest to Active Directory since it uses a different homeDirectory attribute than RFC 2307.

```
# Alternative to using homeDirectory from RFC 2307.

#attributetype (
# 1.3.6.1.4.1.63.1000.1.1.1.1.100
# NAME 'apple-user-homeDirectory'
# DESC 'The absolute path to the home directory'
# EQUALITY caseExactIA5Match
# SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
```

Group Attributes

apple-group-homeurl

Specifies the home directory associated with a managed client workgroup. This is mounted on login of any of the users in this workgroup.

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.14.1
  NAME 'apple-group-homeurl'
  DESC 'group home url'
  EQUALITY caseExactIA5Match
  SUBSTR caseExactIA5SubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
```

apple-group-homeowner

The apple-group-homeowner attribute determines the owner of the workgroup home directory when created in the file system. The group of the directory is the workgroup it is associated with.

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.14.2
  NAME 'apple-group-homeowner'
  DESC 'group home owner settings'
  EQUALITY caseExactIA5Match
  SUBSTR caseExactIA5SubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
```

apple-group-realname

Used to associate a longer, more user friendly name with groups. This name appears in Workgroup Manager and can contain non-ASCII characters.

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.14.5
  NAME 'apple-group-realname'
  DESC 'group real name'
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
```

apple-group-memberUid

Not used by Open Directory Server, but defined as an example attribute and OID that could be added to another LDAP server to support Mac OS X clients.

```
# Alternative to using memberUid from RFC 2307.

#attributetype (
# 1.3.6.1.4.1.63.1000.1.1.1.14.1000
# NAME 'apple-group-memberUid'
# DESC 'group member list'
# EQUALITY caseExactIA5Match
# SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
# can also use OID 1.3.6.1.4.1.63.1000.1.1.2.1000
```

Machine Attributes

apple-machine-software

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.3.8
  NAME 'apple-machine-software'
  DESC 'installed system software'
  EQUALITY caseIgnoreIA5Match
  SUBSTR caseIgnoreIA5SubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
```

apple-machine-hardware

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.3.9
  NAME 'apple-machine-hardware'
  DESC 'system hardware description'
  EQUALITY caseIgnoreIA5Match
  SUBSTR caseIgnoreIA5SubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
```

apple-machine-serves

```
attributeType (
  1.3.6.1.4.1.63.1000.1.1.1.3.10
  NAME 'apple-machine-serves'
  DESC 'NetInfo Domain Server Binding'
  EQUALITY caseExactIA5Match
  SUBSTR caseExactIA5SubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
```

apple-machine-suffix

```
attributeType (
  1.3.6.1.4.1.63.1000.1.1.1.3.11
  NAME 'apple-machine-suffix'
  DESC 'DIT suffix'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

Mount attributes

mountDirectory

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.8.1
  NAME 'mountDirectory'
  DESC 'mount path'
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
```

mountType

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.8.2
  NAME 'mountType'
  DESC 'mount VFS type'
  EQUALITY caseIgnoreIA5Match
  SUBSTR caseIgnoreIA5SubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
```

mountOption

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.8.3
  NAME 'mountOption'
  DESC 'mount options'
  EQUALITY caseIgnoreIA5Match
  SUBSTR caseIgnoreIA5SubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
```

mountDumpFrequency

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.8.4
  NAME 'mountDumpFrequency'
  DESC 'mount dump frequency'
  EQUALITY caseIgnoreIA5Match
  SUBSTR caseIgnoreIA5SubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
```

mountPassNo

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.8.5
  NAME 'mountPassNo'
  DESC 'mount passno'
  EQUALITY caseIgnoreIA5Match
  SUBSTR caseIgnoreIA5SubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
```

Printer Attributes

apple-printer-attributes

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.9.1
  NAME 'apple-printer-attributes'
  DESC 'printer attributes in /etc/printcap format'
  EQUALITY caseIgnoreIA5Match
  SUBSTR caseIgnoreIA5SubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
```

apple-printer-lprhost

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.9.2
  NAME 'apple-printer-lprhost'
  DESC 'printer LPR host name'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

apple-printer-lprqueue

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.9.3
  NAME 'apple-printer-lprqueue'
  DESC 'printer LPR queue'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

apple-printer-type

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.9.4
  NAME 'apple-printer-type'
  DESC 'printer type'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

apple-printer-note

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.9.5
  NAME 'apple-printer-note'
  DESC 'printer note'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

Computer Attributes

apple-realname

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.10.2
  NAME 'apple-realname'
  DESC 'real name'
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

ComputerList Attributes

apple-computers

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.11.3
  NAME 'apple-computers'
  DESC 'computers'
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

apple-computer-list-groups

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.11.4
  NAME 'apple-computer-list-groups'
  DESC 'groups'
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

apple-xmlplist

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.17.1
  NAME 'apple-xmlplist'
  DESC 'XML plist data'
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
```

Configuration Attributes

apple-password-server-location

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.12.1
  NAME 'apple-password-server-location'
  DESC 'password server location'
  EQUALITY caseExactIA5Match
  SUBSTR caseExactIA5SubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
```

apple-data-stamp

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.12.2
  NAME 'apple-data-stamp'
  DESC 'data stamp'
  EQUALITY caseExactIA5Match
  SUBSTR caseExactIA5SubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
```

apple-config-realname

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.12.3
  NAME 'apple-config-realname'
  DESC 'config real name'
  EQUALITY caseExactIA5Match
  SUBSTR caseExactIA5SubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
```

apple-password-server-list

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.12.4
  NAME 'apple-password-server-list'
  DESC 'password server replication plist'
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
```

apple-ldap-replica

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.12.5
  NAME 'apple-ldap-replica'
  DESC 'LDAP replication list'
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

apple-ldap-writable-replica

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.12.6
  NAME 'apple-ldap-writable-replica'
  DESC 'LDAP writable replication list'
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

apple-kdc-authkey

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.12.7
  NAME 'apple-kdc-authkey'
  DESC 'KDC master key RSA encrypted with realm public key'
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

apple-kdc-configdata

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.12.8
  NAME 'apple-kdc-configdata'
  DESC 'Contents of the kdc.conf file'
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
```

PresetUser Attribute

apple-preset-user-is-admin

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.15.1
  NAME 'apple-preset-user-is-admin'
  DESC 'flag indicating whether the preset user is an administrator'
  EQUALITY caseExactIA5Match
  SUBSTR caseExactIA5SubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
```

Authentication Authority Attribute

authAuthority

```
#attributetype (
# 1.3.6.1.4.1.63.1000.1.1.2.16.1
# NAME 'authAuthority'
# DESC 'password server authentication authority'
# EQUALITY caseExactIA5Match
# SUBSTR caseExactIA5SubstringsMatch
# SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
```

Location Attributes

apple-dns-domain

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.18.1
  NAME 'apple-dns-domain'
  DESC 'DNS domain'
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

apple-dns-nameserver

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.18.2
  NAME 'apple-dns-nameserver'
  DESC 'DNS name server list'
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

Mapping Standard Attributes to LDAP and Active Directory

The tables in this section specify how the LDAPv3 plug-in in Directory Access maps the Open Directory record types and attributes to LDAP object classes and attributes. The tables also specify how the Active Directory plug-in in Directory Access maps and generates Active Directory object categories and attributes from Open Directory record types and attributes.

Mappings for Users

The following tables specify the LDAPv3 plug-in in Directory Access maps the Open Directory Users record type and attributes to LDAP object classes and attributes. The tables also specify how the Active Directory plug-in in Directory Access maps and generates Active Directory object categories and attributes from Open Directory record types and attributes.

Record Type Mappings for Users

Open Directory name, RFC/class	LDAP object class name OID	Active Directory plug-in
Users, RFC 2798	inetOrgPerson 2.16.840.1.113730.3.2.2	ObjectCategory = Person
Users, RFC 2307	posixAccount 1.3.6.1.1.1.2.0	
Users, RFC 2307	shadowAccount 1.3.6.1.1.1.2.1	
Users, Apple registered	apple-user 1.3.6.1.4.1.63.1000.1.1.2.1	Apple extended schema

Attribute Mappings for Users

Open Directory name, RFC/class, special purpose	LDAP attribute name OID	Active Directory plug-in
HomeDirectory, Apple registered	apple-user-homeurl 1.3.6.1.4.1.63.1000.1.1.1.6	Generated from homeDirectory
HomeDirectoryQuota, Apple registered	apple-user-homequota 1.3.6.1.4.1.63.1000.1.1.1.8	Apple extended schema
HomeDirectorySoftQuota, Apple registered	apple-user-homesoftquota 1.3.6.1.4.1.63.1000.1.1.1.17	Apple extended schema
MailAttribute, Apple registered	apple-user-mailattribute 1.3.6.1.4.1.63.1000.1.1.1.9	Apple extended schema
PrintServiceUserData, Apple registered	apple-user-printattribute 1.3.6.1.4.1.63.1000.1.1.1.13	Apple extended schema
MCXFlags, Apple registered	apple-mcxflags 1.3.6.1.4.1.63.1000.1.1.1.10	Apple extended schema
MCXSettings, Apple registered	apple-mcxsettings 1.3.6.1.4.1.63.1000.1.1.1.16	Apple extended schema
AdminLimits, Apple registered	apple-user-adminlimits 1.3.6.1.4.1.63.1000.1.1.1.14	Apple extended schema
AuthenticationAuthority, Apple registered	authAuthority 1.3.6.1.4.1.63.1000.1.1.2.16.1	Generated as a Kerberos authority
AuthenticationHint, Apple registered	apple-user-authenticationhint 1.3.6.1.4.1.63.1000.1.1.1.15	Apple extended schema
PasswordPolicyOptions, Apple registered	apple-user-passwordpolicy 1.3.6.1.4.1.63.1000.1.1.1.18	Apple extended schema
Keywords, Apple registered	apple-keyword 1.3.6.1.4.1.63.1000.1.1.1.19	Apple extended schema
Picture, Apple registered	apple-user-picture 1.3.6.1.4.1.63.1000.1.1.1.12	Apple extended schema

Open Directory name, RFC/class, special purpose	LDAP attribute name OID	Active Directory plug-in
GeneratedUID, Apple registered	apple-generateduid 1.3.6.1.4.1.63.1000.1.1.1.1.20	From GUID—formatted
RecordName, RFC 2256	cn 2.5.4.3	Generated from cn, userPrincipal, mail, sAMAccoutName
RecordName, RFC 1274	uid 0.9.2342.19200300.100.1.1	N/A
EMailAddress, RFC 1274	mail 0.9.2342.19200300.100.1.3	RFC standard
RealName, RFC 2256	cn 2.5.4.3	1.2.840.113556.1.2.13 (Microsoft)
Password, RFC 2256	userPassword 2.5.4.35	No mapping
Comment, RFC 2256	description 2.5.4.13	RFC standard
LastName, RFC 2256	sn 2.5.4.4	RFC standard
FirstName, RFC 2256	givenName 2.5.4.42	RFC standard
PhoneNumber, RFC 2256	telephoneNumber 2.5.4.20	RFC standard
AddressLIne1, RFC 2256	street 2.5.4.9	RFC standard
PostalAddress, RFC 2256	postalAddress 2.5.4.16	RFC standard
PostalCode, RFC 2256	postalCode 2.5.4.17	RFC standard
OrganizationName, RFC 2256	o 2.5.4.10	1.2.840.113556.1.2.146 (Microsoft)
UserShell, RFC 2307	loginShell 1.3.6.1.1.1.1.4	Extended using RFC
Change, RFC 2307	shadowLastChange 1.3.6.1.1.1.1.5	No mapping
Expire, RFC 2307	shadowExpire 1.3.6.1.1.1.1.10	No mapping
UniqueID, RFC 2307	uidNumber 1.3.6.1.1.1.1.0	Generated from GUID
NFSHomeDirectory, RFC 2307	homeDirectory 1.3.6.1.1.1.1.3	Generated from homeDirectory
PrimaryGroupID, RFC 2307	gidNumber 1.3.6.1.1.1.1.1	Extended using RFC or generated from GUID

Open Directory name, RFC/class, special purpose	LDAP attribute name OID	Active Directory plug-in
SMBAccountFlags, Samba registered, Apple PDC	acctFlags 1.3.6.1.4.1.7165.2.1.4	1.2.840.113556.1.4.302 (Microsoft)
SMBPasswordLastSet, Samba registered, Apple PDC	pwdLastSet 1.3.6.1.4.1.7165.2.1.3	1.2.840.113556.1.4.96 (Microsoft)
SMBLogonTime, Samba registered, Apple PDC	logonTime 1.3.6.1.4.1.7165.2.1.5	1.2.840.113556.1.4.52 (Microsoft)
SMBLogoffTime, Samba registered, Apple PDC	logoffTime 1.3.6.1.4.1.7165.2.1.6	1.2.840.113556.1.4.51 (Microsoft)
SMBKickoffTime, Samba registered, Apple PDC	kickoffTime 1.3.6.1.4.1.7165.2.1.7	No mapping
SMBHomeDrive, Samba registered, Apple PDC	homeDrive 1.3.6.1.4.1.7165.2.1.10	1.2.840.113556.1.4.45 (Microsoft)
SMBScriptPath, Samba registered, Apple PDC	scriptPath 1.3.6.1.4.1.7165.2.1.11	1.2.840.113556.1.4.62 (Microsoft)
SMBProfilePath, Samba registered, Apple PDC	profilePath 1.3.6.1.4.1.7165.2.1.12	1.2.840.113556.1.4.139 (Microsoft)
SMBUserWorkstations, Samba registered, Apple PDC	userWorkstations 1.3.6.1.4.1.7165.2.1.13	1.2.840.113556.1.4.86 (Microsoft)
SMBHome, Samba registered, Apple PDC	smbHome 1.3.6.1.4.1.7165.2.1.17	1.2.840.113556.1.4.44 (Microsoft)
SMBRID, Samba registered, Apple PDC	rid 1.3.6.1.4.1.7165.2.1.14	1.2.840.113556.1.4.153 (Microsoft)
SMBGroupRID, Samba registered, Apple PDC	primaryGroupID 1.3.6.1.4.1.7165.2.1.15	1.2.840.113556.1.4.98 (Microsoft)
FaxNumber, RFC 2256	fax 2.5.4.23	RFC standard
MobileNumber, RFC 1274	mobile 0.9.2342.19200300.100.1.41	RFC standard
PagerNumber, RFC 1274	pager 0.9.2342.19200300.100.1.42	RFC standard

Open Directory name, RFC/class, special purpose	LDAP attribute name OID	Active Directory plug-in
Department, RFC 2798,	departmentNumber 2.16.840.1.113730.3.1.2	1.2.840.113556.1.2.141 (Microsoft)
NickName, Microsoft Attribute		1.2.840.113556.1.2.447 (Microsoft)
JobTitle, RFC 2256	title 2.5.4.12	RFC standard
Building, RFC 2256	buildingName 2.5.4.19	RFC standard
Country, RFC 2256	c 2.5.4.6	RFC standard
Street, RFC 2256	street 2.5.4.9	1.2.840.113556.1.2.256 (Microsoft)
City, RFC 2256	locality 2.5.4.7	RFC standard
State, RFC 2256	st 2.5.4.8	RFC standard

Mappings for Groups

The following tables specify how the LDAPv3 plug-in in Directory Access maps the Open Directory Groups record type and attributes to LDAP object classes. The tables also specify how the Active Directory plug-in in Directory Access maps and generates Active Directory object categories and attributes from Open Directory record types and attributes.

Record Type Mappings for Groups

Open Directory name, RFC/class	LDAP object class name OID	Active Directory plug-in
Groups, RFC 2307	posixGroup 1.3.6.1.1.1.2.2	objectCategory = Group
Groups, Apple registered	apple-group 1.3.6.1.4.1.63.1000.1.1.2.14	Apple extended schema

Attribute Mappings for Groups

Open Directory name, RFC/class	LDAP attribute name OID	Active Directory plug-in
RecordName, RFC 2256	cn 2.5.4.3	RFC standard
HomeDirectory, Apple registered	apple-group-homeurl 1.3.6.1.4.1.63.1000.1.1.1.14.1	Apple extended schema
HomeLocOwner, Apple registered	apple-group-homeowner 1.3.6.1.4.1.63.1000.1.1.1.14.2	Apple extended schema
MCXFlags, Apple registered	apple-mcxflags 1.3.6.1.4.1.63.1000.1.1.1.1.10	Apple extended schema
MCXSettings, Apple registered	apple-mcxsettings 1.3.6.1.4.1.63.1000.1.1.1.1.16	Apple extended schema
RealName, Apple registered	apple-group-realname 1.3.6.1.4.1.63.1000.1.1.1.14.5	1.2.840.113556.1.2.13 (Microsoft)
Picture, Apple registered	apple-user-picture 1.3.6.1.4.1.63.1000.1.1.1.1.12	Apple extended schema
Keywords, Apple registered	apple-keyword 1.3.6.1.4.1.63.1000.1.1.1.1.19	Apple extended schema
GeneratedUID, Apple registered	apple-generateduid 1.3.6.1.4.1.63.1000.1.1.1.1.20	From GUID—formatted
GroupMembership, RFC 2307	memberUid 1.3.6.1.1.1.1.12	Generated from member
Member, RFC 2307	memberUid 1.3.6.1.1.1.1.12	Same as GroupMembership
PrimaryGroupID, RFC 2307	gidNumber 1.3.6.1.1.1.1.1	Extended using RFC or generated from GUID

Mappings for Mounts

The following tables specify how the LDAPv3 plug-in in Directory Access maps the Open Directory Mounts record type and attributes to LDAP object classes and attributes. The tables also specify how the Active Directory plug-in in Directory Access maps and generates Active Directory object categories and attributes from Open Directory record types and attributes.

Record Type Mappings for Mounts

Open Directory name, RFC/class	LDAP object class name OID	Active Directory plug-in
Mounts, Apple registered	mount 1.3.6.1.4.1.63.1000.1.1.2.8	Apple extended schema

Attribute Mappings for Mounts

Open Directory name, RFC/class	LDAP attribute name OID	Active Directory plug-in
RecordName, RFC 2256	cn 2.5.4.3	RFC standard
VFSLinkDir, Apple registered	mountDirectory 1.3.6.1.4.1.63.1000.1.1.1.8.1	Apple extended schema
VFSOpts, Apple registered	mountOption 1.3.6.1.4.1.63.1000.1.1.1.8.3	Apple extended schema
VFSType, Apple registered	mountType 1.3.6.1.4.1.63.1000.1.1.1.8.2	Apple extended schema
VFSDumpFreq, Apple registered	mountDumpFrequency 1.3.6.1.4.1.63.1000.1.1.1.8.4	Apple extended schema
VFSPassNo, Apple registered	mountPassNo 1.3.6.1.4.1.63.1000.1.1.1.8.5	Apple extended schema

Mappings for Computers

The following tables specify how the LDAPv3 plug-in in Directory Access maps the Open Directory Computers record type and attributes to LDAP object classes. The tables also specify how the Active Directory plug-in in Directory Access maps and generates Active Directory object categories and attributes from Open Directory record types and attributes.

Record Type Mappings for Computers

Open Directory name, RFC/class	LDAP object class name OID	Active Directory plug-in
Computers, Apple registered	apple-computer 1.3.6.1.4.1.63.1000.1.1.2.10	objectCategory = Computer

Attribute Mappings for Computers

Open Directory name, RFC/class, special purpose	LDAP attribute name OID	Active Directory plug-in
RecordName, RFC 2256	cn 2.5.4.3	RFC standard
RealName, Apple registered	apple-realname 1.3.6.1.4.1.63.1000.1.1.1.10.2	1.2.840.113556.1.2.13 (Microsoft)
MCXFlags, Apple registered	apple-mcxflags 1.3.6.1.4.1.63.1000.1.1.1.1.10	Apple extended schema
MCXSettings, Apple registered	apple-mcxsettings 1.3.6.1.4.1.63.1000.1.1.1.1.16	Apple extended schema
Group, Apple registered	apple-computer-list-groups 1.3.6.1.4.1.63.1000.1.1.1.11.4	Apple extended schema

Open Directory name, RFC/class, special purpose	LDAP attribute name OID	Active Directory plug-in
AuthenticationAuthority, Apple registered	authAuthority 1.3.6.1.4.1.63.1000.1.1.2.16.1	Apple extended schema
GeneratedUID, Apple registered	apple-generateduid 1.3.6.1.4.1.63.1000.1.1.1.1.20	From GUID—formatted
XMLPlist, Apple registered	apple-xmlplist 1.3.6.1.4.1.63.1000.1.1.1.17.1	Apple extended schema
Comment, RFC 2256	description 2.5.4.13	RFC standard
ENetAddress, RFC 2307	macAddress 1.3.6.1.1.1.1.22	Extended using RFC
UniqueID, RFC 2307	uidNumber 1.3.6.1.1.1.1.0	Generated from GUID
PrimaryGroupID, RFC 2307	gidNumber 1.3.6.1.1.1.1.1	Extended using RFC or generated
SMBAccountFlags, Samba registered, Apple PDC	acctFlags 1.3.6.1.4.1.7165.2.1.4	1.2.840.113556.1.4.302 (Microsoft)
SMBPasswordLastSet, Samba registered, Apple PDC	pwdLastSet 1.3.6.1.4.1.7165.2.1.3	1.2.840.113556.1.4.96 (Microsoft)
SMBLogonTime, Samba registered, Apple PDC	logonTime 1.3.6.1.4.1.7165.2.1.5	1.2.840.113556.1.4.52 (Microsoft)
SMBLogoffTime, Samba registered, Apple PDC	logoffTime 1.3.6.1.4.1.7165.2.1.6	1.2.840.113556.1.4.51 (Microsoft)
SMBKickoffTime, Samba registered, Apple PDC	kickoffTime 1.3.6.1.4.1.7165.2.1.7	No mapping
SMBRID, Samba registered, Apple PDC	rid 1.3.6.1.4.1.7165.2.1.14	1.2.840.113556.1.4.153 (Microsoft)
SMBGroupID, Samba registered, Apple PDC	primaryGroupID 1.3.6.1.4.1.7165.2.1.15	1.2.840.113556.1.4.98 (Microsoft)

Mappings for ComputerLists

The following tables specify how the LDAPv3 plug-in in Directory Access maps the Open Directory ComputerLists record type and attributes to LDAP object classes. The tables also specify how the Active Directory plug-in in Directory Access maps and generates Active Directory object categories and attributes from Open Directory record types and attributes.

Record Type Mappings for ComputerLists

Open Directory name, RFC/class	LDAP object class name OID	Active Directory plug-in
ComputerLists, Apple registered	apple-computer-list 1.3.6.1.4.1.63.1000.1.1.2.11	Apple extended schema

Attribute Mappings for ComputerLists

Open Directory name, RFC/class,	LDAP attribute name OID	Active Directory plug-in
RecordName, RFC 2256	cn 2.5.4.3	RFC standard
MCXFlags, Apple registered	apple-mcxflags 1.3.6.1.4.1.63.1000.1.1.1.10	Apple extended schema
MCXSettings, Apple registered	apple-mcxsettings 1.3.6.1.4.1.63.1000.1.1.1.16	Apple extended schema
Computers, Apple registered	apple-computers 1.3.6.1.4.1.63.1000.1.1.1.11.3	Apple extended schema
Group, Apple registered	apple-computer-list-groups 1.3.6.1.4.1.63.1000.1.1.1.11.4	Apple extended schema
Keywords, Apple registered	apple-keyword 1.3.6.1.4.1.63.1000.1.1.1.1.19	Apple extended schema

Mappings for Config

The following tables specify how the LDAPv3 plug-in in Directory Access maps the Open Directory Config record type and attributes to LDAP object classes. The tables also specify how the Active Directory plug-in in Directory Access maps and generates Active Directory object categories and attributes from Open Directory record types and attributes.

Record Type Mappings for Config

Open Directory name, RFC/class	LDAP object class name OID	Active Directory plug-in
Config, Apple registered	apple-configuration 1.3.6.1.4.1.63.1000.1.1.2.12	Apple extended schema

Attribute Mappings for Config

Open Directory name, RFC/class, special purpose	LDAP attribute name OID	Active Directory plug-in
RecordName, RFC 2256	cn 2.5.4.3	RFC standard
RealName, Apple registered	apple-config-realname 1.3.6.1.4.1.63.1000.1.1.1.12.3	1.2.840.113556.1.2.13 (Microsoft)
DataStamp, Apple registered	apple-data-stamp 1.3.6.1.4.1.63.1000.1.1.1.12.2	Apple extended schema
KDCAuthKey, Apple registered, Apple KDC	apple-kdc-authkey 1.3.6.1.4.1.63.1000.1.1.1.12.7	No mapping
KDCConfigData, Apple registered, Apple KDC	apple-kdc-configdata 1.3.6.1.4.1.63.1000.1.1.1.12.8	No mapping
Keywords, Apple registered	apple-keyword 1.3.6.1.4.1.63.1000.1.1.1.19	Apple extended schema
LDAPReadReplicas, Apple registered, Apple LDAP Server	apple-ldap-replica 1.3.6.1.4.1.63.1000.1.1.1.12.5	No mapping
LDAPWriteReplicas, Apple registered, Apple LDAP Server	apple-ldap-writable-replica 1.3.6.1.4.1.63.1000.1.1.1.12.6	No mapping
PasswordServerList, Apple registered, Password Server	apple-password-server-list 1.3.6.1.4.1.63.1000.1.1.1.12.4	No mapping
PasswordServerLocation, Apple registered, Password Server	apple-password-server-location 1.3.6.1.4.1.63.1000.1.1.1.12.1	No mapping
XMLPlist, Apple registered	apple-xmlplist 1.3.6.1.4.1.63.1000.1.1.1.17.1	Apple extended schema

Mappings for People

The following tables specify how the LDAPv3 plug-in in Directory Access maps the Open Directory People record type and attributes to LDAP object classes. The tables also specify how the Active Directory plug-in in Directory Access maps and generates Active Directory object categories and attributes from Open Directory record types and attributes.

Record Type Mappings for People

Open Directory name, RFC/class	LDAP object class name OID	Active Directory plug-in
People, RFC 2798	inetOrgPerson 2.16.840.1.113730.3.2.2	RFC standard

Attribute Mappings for People

Open Directory name, RFC/class	LDAP attribute name OID	Active Directory plug-in
RecordName, RFC 2256	cn 2.5.4.3	RFC standard
EMailAddress, RFC 1274	mail 0.9.2342.19200300.100.1.3	RFC standard
RealName, RFC 2256	cn 1.2.840.113556.1.3.23	RFC standard
LastName, RFC 2256	sn 2.5.4.4	RFC standard
FirstName, RFC 2256	givenName 2.5.4.42	RFC standard
FaxNumber, RFC 2256	fax 2.5.4.23	RFC standard
MobileNumber, RFC 1274	mobile 0.9.2342.19200300.100.1.41	RFC standard
PagerNumber, RFC 1274	pager 0.9.2342.19200300.100.1.42	RFC standard
Department, RFC 2798,	departmentNumber 2.16.840.1.113730.3.1.2	1.2.840.113556.1.2.141 (Microsoft)
JobTitle, RFC 2256	title 2.5.4.12	RFC standard
PhoneNumber, RFC 2256	telephoneNumber 2.5.4.20	RFC standard
AddressLine1, RFC 2256	street 2.5.4.9	RFC standard
Street, RFC 2256	street 2.5.4.9	RFC standard
PostalAddress, RFC 2256	postalAddress 2.5.4.16	RFC standard
City, RFC 2256	locality 2.5.4.7	RFC standard
State, RFC 2256	st 2.5.4.8	RFC standard
Country, RFC 2256	c 2.5.4.6	RFC standard
PostalCode, RFC 2256	postalCode 2.5.4.17	RFC standard
OrganizationName, RFC 2256	o 2.5.4.10	1.2.840.113556.1.2.146 (Microsoft)

Mappings for PresetComputerLists

The following tables specify how the LDAPv3 plug-in in Directory Access maps the Open Directory PresetComputerLists record type and attributes to LDAP object classes. The tables also specify how the Active Directory plug-in in Directory Access maps and generates Active Directory object categories and attributes from Open Directory record types and attributes.

Record Type Mappings for PresetComputerLists

Open Directory name, RFC/class	LDAP object class name OID	Active Directory plug-in
PresetComputerLists, Apple registered	apple-preset-computer-list 1.3.6.1.4.1.63.1000.1.1.2.13	Apple extended schema

Attribute Mappings for PresetComputerLists

Open Directory name, RFC/class	LDAP attribute name OID	Active Directory plug-in
RecordName, RFC 2256	cn 2.5.4.3	RFC standard
MCXFlags, Apple registered	apple-mcxflags 1.3.6.1.4.1.63.1000.1.1.1.10	Apple extended schema
MCXSettings, Apple registered	apple-mcxsettings 1.3.6.1.4.1.63.1000.1.1.1.16	Apple extended schema
Keywords, Apple registered	apple-keyword 1.3.6.1.4.1.63.1000.1.1.1.19	Apple extended schema

Mappings for PresetGroups

The following tables specify how the LDAPv3 plug-in in Directory Access maps the Open Directory PresetGroups record type and attributes to LDAP object classes. The tables also specify how the Active Directory plug-in in Directory Access maps and generates Active Directory object categories and attributes from Open Directory record types and attributes.

Record Type Mappings for PresetGroups

Open Directory name, RFC/class	LDAP object class name OID	Active Directory plug-in
PresetGroups, Apple registered	apple-preset-group 1.3.6.1.4.1.63.1000.1.1.3.14	Apple extended schema

Attribute Mappings for PresetGroups

Open Directory name, RFC/class	LDAP attribute name OID	Active Directory plug-in
HomeDirectory, Apple registered	apple-group-homeurl 1.3.6.1.4.1.63.1000.1.1.1.6	Apple extended schema
HomeLocOwner, Apple registered	apple-group-homeowner 1.3.6.1.4.1.63.1000.1.1.14.2	Apple extended schema
MCXFlags, Apple registered	apple-mcxflags 1.3.6.1.4.1.63.1000.1.1.1.10	Apple extended schema
MCXSettings, Apple registered	apple-mcxsettings 1.3.6.1.4.1.63.1000.1.1.1.16	Apple extended schema
RealName, Apple registered	apple-group-realname 1.3.6.1.4.1.63.1000.1.1.14.5	Apple extended schema
Keywords, Apple registered	apple-keyword 1.3.6.1.4.1.63.1000.1.1.1.19	Apple extended schema
RecordName, RFC 2256	cn 2.5.4.3	RFC standard
GroupMembership, RFC 2307	memberUid 1.3.6.1.1.1.1.12	Extended using RFC
PrimaryGroupID, RFC 2307	gidNumber 1.3.6.1.1.1.1.1	Extended using RFC

Mappings for PresetUsers

The following tables specify how the LDAPv3 plug-in in Directory Access maps the Open Directory PresetUsers record type and attributes to LDAP object classes. The tables also specify how the Active Directory plug-in in Directory Access maps and generates Active Directory object categories and attributes from Open Directory record types and attributes.

Record Type Mappings for PresetUsers

Open Directory name, RFC/class	LDAP object class name OID	Active Directory plug-in
PresetUsers, Apple registered	apple-preset-user 1.3.6.1.4.1.63.1000.1.1.2.15	ObjectCategory = Person

Attribute Mappings for PresetUsers

Open Directory name, RFC/class	LDAP attribute name OID	Active Directory plug-in
HomeDirectory, Apple registered	apple-user-homeurl 1.3.6.1.4.1.63.1000.1.1.1.6	N/A
HomeDirectoryQuota, Apple registered	apple-user-homequota 1.3.6.1.4.1.63.1000.1.1.1.8	Apple extended schema
HomeDirectorySoftQuota, Apple registered	apple-user-homesoftquota 1.3.6.1.4.1.63.1000.1.1.1.17	Apple extended schema

Open Directory name, RFC/class	LDAP attribute name OID	Active Directory plug-in
MailAttribute, Apple registered	apple-user-mailattribute 1.3.6.1.4.1.63.1000.1.1.1.9	Apple extended schema
PrintServiceUserData, Apple registered	apple-user-printattribute 1.3.6.1.4.1.63.1000.1.1.1.13	Apple extended schema
MCXFlags, Apple registered	apple-mcxflags 1.3.6.1.4.1.63.1000.1.1.1.10	Apple extended schema
MCXSettings, Apple registered	apple-mcxsettings 1.3.6.1.4.1.63.1000.1.1.1.16	Apple extended schema
AdminLimits, Apple registered	apple-user-adminlimits 1.3.6.1.4.1.63.1000.1.1.1.14	Apple extended schema
Picture, Apple registered	apple-user-picture 1.3.6.1.4.1.63.1000.1.1.1.12	Apple extended schema
AuthenticationAuthority, Apple registered	authAuthority 1.3.6.1.4.1.63.1000.1.1.2.16.1	Apple extended schema
PasswordPolicyOptions, Apple registered	apple-user-passwordpolicy 1.3.6.1.4.1.63.1000.1.1.1.18	Apple extended schema
PresetUserIsAdmin, Apple registered	apple-preset-user-is-admin 1.3.6.1.4.1.63.1000.1.1.15.1	Apple extended schema
Keywords, Apple registered	apple-keyword 1.3.6.1.4.1.63.1000.1.1.1.19	Apple extended schema
RecordName, RFC 1274	cn 2.5.4.3	RFC standard
RealName, RFC 2256	cn 2.5.4.3	RFC standard
Password, RFC 2256	userPassword 2.5.4.35	N/A
GroupMembership, RFC 2307	memberUid 1.3.6.1.1.1.1.12	Extended using RFC
PrimaryGroupID, RFC 2307	gidNumber 1.3.6.1.1.1.1.1	Extended using RFC
NFSHomeDirectory, RFC 2307	homeDirectory 1.3.6.1.1.1.1.3	N/A
UserShell, RFC 2307	loginShell 1.3.6.1.1.1.1.4	Extended using RFC
Change, RFC 2307	shadowLastChange 1.3.6.1.1.1.1.5	N/A
Expire, RFC 2307	shadowExpire 1.3.6.1.1.1.1.10	N/A

Mappings for Printers

The following tables specify how the LDAPv3 plug-in in Directory Access maps the Open Directory Printers record type and attributes to LDAP object classes. The tables also specify how the Active Directory plug-in in Directory Access maps and generates Active Directory object categories and attributes from Open Directory record types and attributes.

Record Type Mappings for Printers

Open Directory name, RFC/class	LDAP object class name OID	Active Directory plug-in
Printers, Apple registered	apple-printer 1.3.6.1.4.1.63.1000.1.1.2.9	ObjectCategory = Print-Queue
Printers, IETF-Draft-IPP-LDAP	printerIPP 1.3.18.0.2.6.256	

Attribute Mappings for Printers

Open Directory name, RFC/class, special purpose	LDAP attribute name OID	Active Directory plug-in
RecordName, RFC 2256	cn 2.5.4.3	RFC standard
RealName, RFC 2256	cn 2.5.4.3	1.2.840.113556.1.4.300 (Microsoft)
PrinterLPRHost, Apple registered, legacy support	apple-printer-lprhost 1.3.6.1.4.1.63.1000.1.1.1.9.2	N/A
PrinterLPRQueue, Apple registered, legacy support	apple-printer-lprqueue 1.3.6.1.4.1.63.1000.1.1.1.9.3	N/A
PrinterType, Apple registered, legacy support	apple-printer-type 1.3.6.1.4.1.63.1000.1.1.1.9.4	N/A
PrinterNote, Apple registered, legacy support	apple-printer-note 1.3.6.1.4.1.63.1000.1.1.1.9.5	N/A
Location, IETF-Draft-IPP-LDAP	printer-location 1.3.18.0.2.4.1136	1.2.840.113556.1.4.222 (Microsoft)
Comment, RFC 2256	description 2.5.4.13	RFC standard
PrinterMakeAndModel, IETF-Draft-IPP-LDAP	printer-make-and-model 1.3.18.0.2.4.1138	1.2.840.113556.1.4.229 (Microsoft)
PrinterURI, IETF-Draft-IPP-LDAP	printer-uri 1.3.18.0.2.4.1140	Generated from uNCName

Open Directory name, RFC/class, special purpose	LDAP attribute name OID	Active Directory plug-in
PrinterXRISupported, IETF-Draft-IPP-LDAP	printer-xri-supported 1.3.18.0.2.4.1107	Generated from portName/ uNCName
Printer1284DeviceID, Apple registered	printer-1284-device-id 1.3.6.1.4.1.63.1000.1.1.1.9.6	Apple extended schema

Mappings for AutoServerSetup

The following tables specify how the LDAPv3 plug-in in Directory Access maps the Open Directory AutoServerSetup record type and attributes to LDAP object classes. The tables also specify how the Active Directory plug-in in Directory Access maps and generates Active Directory object categories and attributes from Open Directory record types and attributes.

Record Type Mappings for AutoServerSetup

Open Directory name, RFC/class	LDAP object class name OID	Active Directory plug-in
AutoServerSetup, Apple registered	apple-serverassistant-config 1.3.6.1.4.1.63.1000.1.1.2.17	Apple extended schema

Attribute Mappings for AutoServerSetup

Open Directory name, RFC/class	LDAP attribute name OID	Active Directory plug-in
RecordName, RFC 2256	cn 2.5.4.3	RFC standard
XMLPlist, Apple registered	apple-xmlplist 1.3.6.1.4.1.63.1000.1.1.1.17.1	Apple extended schema

Mappings for Locations

The following tables specify how the LDAPv3 plug-in in Directory Access maps the Open Directory Locations record type and attributes to LDAP object classes. The tables also specify how the Active Directory plug-in in Directory Access maps and generates Active Directory object categories and attributes from Open Directory record types and attributes.

Record Type Mappings for Locations

Open Directory name, RFC/class	LDAP object class name OID	Active Directory plug-in
Locations, Apple registered	apple-locations 1.3.6.1.4.1.63.1000.1.1.2.18	Apple extended schema

Attribute Mappings for Locations

Open Directory name, RFC/class	LDAP attribute name OID	Active Directory plug-in
RecordName, RFC 2256	cn 2.5.4.3	RFC standard
DNSDomain, Apple registered	apple-dns-domain 1.3.6.1.4.1.63.1000.1.1.1.18.1	Apple extended schema
DNSNameServer, Apple registered	apple-dns-nameserver 1.3.6.1.4.1.63.1000.1.1.1.18.2	Apple extended schema

Standard Attributes in User Records

The following table specifies facts about the standard attributes, or data types, found in user records of Mac OS X data services. Use these facts when mapping LDAP or Active Directory domains to Mac OS X directory services.

Important: When mapping Mac OS X user attributes to a read/write LDAP directory domain (an LDAP domain that is not read-only), do not map the RealName and the first RecordName attributes to the same LDAP attribute. For example, do not map both RealName and RecordName to the cn attribute. If RealName and RecordName are mapped to the same LDAP attribute, problems will occur when you try to edit the full (long) name or the first short name in Workgroup Manager.

Mac OS X user attribute	Format	Sample values
RecordName: A list of names associated with a user; the first is the user's short name, which is also the name of the user's home directory Important: All attributes used for authentication must map to RecordName.	First value: ASCII characters A–Z, a–z, 0–9, _,- Second value: UTF-8 Roman text	Dave David Mac DMacSmith Non-zero length, 1 to 16 values. Maximum 255 bytes (85 triple-byte to 255 single-byte characters) per instance. First value must be 1 to 30 bytes for clients using Macintosh Manager, or 1 to 8 bytes for clients using Mac OS X version 10.1 and earlier.
RealName: A single name, usually the user's full name; not used for authentication	UTF-8 text	David L. MacSmith, Jr. Non-zero length, maximum 255 bytes (85 triple-byte to 255 single-byte characters).

Mac OS X user attribute	Format	Sample values
UniqueID: A unique user identifier, used for access privilege management	SIgned 32-bit ASCII string of digits 0–9	Values below 100 are typically used for system accounts. Zero is reserved for use by the system. Normally unique among entire population of users, but sometimes can be duplicated. Warning: A non-integer value is interpreted as 0, which is the UniqueID of the root user.
PrimaryGroupID: A user's primary group association	Signed 32-bit ASCII string of digits 0–9	Range is 1 to 2,147,483,648. Normally unique among entire population of group records. If blank, 20 is assumed.
NFSHomeDirectory: Local file system path to the user's home directory	UTF-8 text	/Network/Servers/example/Users/K-M/Tom King Non-zero length. Maximum 255 bytes.
HomeDirectory: The location of an AFP-based home directory	UTF-8 XML text	<home_dir> <url>afp://*server/sharept*</url> <path>*usershomedir*</path> </home_dir> In the following example, Tom King's home directory is K-M/Tom King, which resides beneath the share point directory, Users: <home_dir> <url>afp://example.com/Users</url> <path>K-M/Tom King</path> </home_dir>
HomeDirectoryQuota: The disk quota for the user's home directory	Text for the number of bytes allowed	If the quota is 10MB, the value will be the text string "1048576".
MailAttribute: A user's mail service configuration	UTF-8 XML text	
PrintServiceUserData: A user's print quota statistics	UTF-8 XML plist, single value	.
MCXFlags: If present, MCXSettings is loaded; if absent, MCXSettings isn't loaded; required for a managed user.	UTF-8 XML plist, single value	

Mac OS X user attribute	Format	Sample values
MCXSettings: A user's managed preferences	UTF-8 XML plist, multivalued	
AdminLimits: The privileges allowed by Workgroup Manager to a user that can administer the directory domain	UTF-8 XML plist, single value	
Password: The user's password	UNIX crypt	
Picture: File path to a recognized graphic file to be used as a display picture for the user	UTF-8 text	Maximum 255 bytes.
Comment: Any documentation you like	UTF-8 text	John is in charge of product marketing. Maximum 32,676 bytes.
UserShell: The location of the default shell for command-line interactions with the server	Path name	/bin/tcsh /bin/sh None (this value prevents users with accounts in the directory domain from accessing the server remotely via a command line) Non-zero length.
Change: Not used by Mac OS X, but corresponds to part of standard LDAP schema	Number	
Expire: Not used by Mac OS X, but corresponds to part of standard LDAP schema	Number	

Mac OS X user attribute	Format	Sample values
AuthenticationAuthority: Describes the user's authentication methods, such as Open Directory or crypt password; not required for a user with only a crypt password; absence of this attribute signifies legacy authentication (crypt with Authentication Manager, if it is available).	ASCII text	Values describe the user's authentication methods. Can be multivalued (for example, basic and ShadowHash). Each value has the format *vers*; *tag*; *data* (where *vers* and *data* may be blank). Crypt password: ;basic; Open Directory authentication: ;ApplePasswordServer; *HexID, server's public key IPaddress*:*port* Shadow password (local directory domain only): ;ShadowHash;
AuthenticationHint: Text set by the user to be displayed as a password reminder	UTF-8 text	Your guess is as good as mine. Maximum 255 bytes.
FirstName: Used by Address Book and other applications that use the contacts search policy		
LastName: Used by Address Book and other applications that use the contacts search policy		
EMailAddress: An email address to which mail should be automatically forwarded when a user has no MailAttribute defined; used by Address Book, Mail, and other applications that use the contacts search policy	Any legal RFC 822 email address or a valid "mailto:" URL	user@example.com mailto:user@example.com
PhoneNumber: Used by Address Book and other applications that use the contacts search policy		
AddressLine1: Used by Address Book and other applications that use the contacts search policy		

Mac OS X user attribute	Format	Sample values
PostalAddress: Used by Address Book and other applications that use the contacts search policy		
PostalCode: Used by Address Book and other applications that use the contacts search policy		
OrganizationName: Used by Address Book and other applications that use the contacts search policy		

User Data That Mac OS X Server Uses

The following table describes how your Mac OS X Server uses data from user records in directory domains. Consult this table to determine the attributes, or data types, that your server's various services expect to find in user records of directory domains. Note that "All services" in the far-left column include AFP, SMB, FTP, HTTP, NFS, WebDAV, POP, IMAP, Workgroup Manager, Server Admin, the Mac OS X login window, and Macintosh Manager.

Server component	Mac OS X user attribute	Dependency
All services	RecordName	Required for authentication
All services	RealName	Required for authentication
All services	AuthenticationAuthority	Used for Kerberos, Password Server, and shadow password authentication
All services	Password	Used for basic (crypt password) or LDAP bind authentication
All services	UniqueID	Required for authorization (for example, file permissions and mail accounts)
All services	PrimaryGroupID	Required for authorization (for example, file permissions and mail accounts)

Server component	Mac OS X user attribute	Dependency
FTP service Web service Apple file service NFS service Macintosh Manager Mac OS X login window Application and system preferences	HomeDirectory NFSHomeDirectory	Optional
Mail service	MailAttribute	Required for login to mail service on your server
Mail service	EMailAddress	Optional

Standard Attributes in Group Records

The following table specifies facts about the standard attributes, or data types, found in group records of Mac OS X data services. Use these facts when mapping LDAP or Active Directory domains to Mac OS X directory services.

Mac OS X group attribute	Format	Sample values
RecordName: Name associated with a group	ASCII characters A–Z, a–z, 0–9, _	Science Science_Dept Science.Teachers Non-zero length, maximum 255 bytes (85 triple-byte to 255 single-byte characters).
RealName: Usually the group's full name	UTF-8 text	Science Department Teachers Non-zero length, maximum 255 bytes (85 triple-byte to 255 single-byte characters).
PrimaryGroupID: A unique identifier for the group	Signed 32-bit ASCII string of digits 0–9	Normally unique among entire population of group records.
GroupMembership: A list of short names of user records that are considered part of the group	ASCII characters A–Z, a–z, 0–9, _,-	bsmith, jdoe Can be an empty list (normally for users' primary group).

Mac OS X group attribute	Format	Sample values
HomeDirectory: The location of an AFP-based home directory for the group	Structured UTF-8 text	<home_dir> <url>afp://*server*/*sharept*</url> <path>*grouphomedir*</path> </home_dir> In the following example, the Science group's home directory is K-M/Science, which resides beneath the share point directory, Groups: <home_dir> <url>afp://example.com/Groups</url> <path>K-M/Science</path> </home_dir>
Member: Same data as GroupMembership but each is used by different services of Mac OS X Server	ASCII characters A–Z, a–z, 0–9, _,-	bsmith, jdoe Can be an empty list (normally for users' primary group).
HomeLocOwner: The short name of the user that owns the group's home directory	ASCII characters A–Z, a–z, 0–9, _,-	
MCXFlags: If present, MCXSettings is loaded; if absent, MCXSettings isn't loaded; required for a managed user	UTF-8 XML plist, single value	
MCXSettings: The preferences for a workgroup (a managed group)	UTF-8 XML plist, multivalued	

Standard Attributes in Computer Records

The following table specifies facts about the standard attributes, or data types, found in computer records of Mac OS X data services. Computer records associate the hardware address of a computer's Ethernet interface with a name for the computer. The name is part of a computer list record (much as a user is in a group). Use these facts when mapping LDAP or Active Directory domains to Mac OS X directory services.

Mac OS X computer attribute	Format	Sample values
RecordName: Name associated with a computer	UTF-8 text	iMac 1
Comment: Any documentation you like	UTF-8 text	
EnetAddress: The MAC address of the computer's Ethernet interface	Colon-separated hex notation; leading zeroes may be omitted	00:05:02:b7:b5:88
MCXFlags: Used only in the "guest" computer record; if present, MCXSettings is loaded; if absent, MCXSettings isn't loaded; required for a managed computer	UTF-8 XML plist, single value	
MCXSettings: Used only in the "guest" computer record; a managed computer's preferences	UTF-8 XML plist, multivalued	

Standard Attributes in Computer List Records

The following table specifies facts about the standard attributes, or data types, found in computer list records of Mac OS X data services. A computer list record identifies a group of computers (much as a group record identifies a collection of users). Use these facts when mapping LDAP or Active Directory domains to Mac OS X directory services.

Mac OS X computer list attribute	Format	Sample values
RecordName: Name associated with a computer list	UTF-8 text	Lab Computers Non-zero length, maximum 255 bytes (85 triple-byte to 255 single-byte characters).
MCXFlags	UTF-8 XML plist, single value	
MCXSettings: Stores preferences for a managed computer	UTF-8 XML plist, multivalued	
Computers	Multivalued list of computer record names	iMac 1, iMac 2
Group A list of groups whose members may log in on the computers in this computer list	Multivalued list of short names of groups	herbivores,omnivores

Standard Attributes in Mount Records

The following table specifies facts about the standard attributes, or data types, found in mount records of Mac OS X data services. Use these facts when mapping LDAP or Active Directory domains to Mac OS X directory services.

Mac OS X mount attributes	Format	Sample values
RecordName: Host and path of the sharepoint	UTF-8 text	*hostname:/path on server* indigo:/Volumes/home2
VFSLinkDir Path for the mount on a client	UTF-8 text	/Network/Servers
VFSType	ASCII text	For AFP: url For NFS: nfs
VFSOpts	UTF-8 text	For AFP (two values): net url==afp:// ;AUTH=NO%20USER%20 AUTHENT@*server/sharepoint/* For NFS: net
VFSDumpFreq		
VFSPassNo		

Standard Attributes in Config Records

The following table specifies facts about the standard attributes, or data types, found in config records of Mac OS X data services.

Mac OS X Server version 10.2 and later uses the following two types of config records:

- The mcx_cache record always has the RecordName of mcx_cache. It also uses RealName and DataStamp to determine whether the cache should be updated or the server settings ignored. If you want managed clients, you must have an mcx_cache config record.
- The passwordserver record has the additional attribute PasswordServerLocation.

Use these facts when mapping LDAP or Active Directory domains to Mac OS X directory services.

Mac OS X config attributes	Format	Sample values
RecordName: Name associated with a config	ASCII characters A–Z, a–z, 0–9, _,-,.	mcx_cache passwordserver Non-zero length, maximum 255 bytes (85 triple-byte to 255 single-byte characters).
PasswordServerLocation: Identifies the host of the Password Server that's associated with the directory domain	IP address or host name	192.168.1.90
RealName		For the mcx_cache config record, RealName is a GUID
DataStamp		For the mcx_cache config record, DataStamp is a GUID

Open Directory Password Server Authentication Methods

Open Directory Password Server is based on the SASL standard for supporting multiple methods of authenticating user passwords.

The authentication methods supported by Open Directory Password Server include APOP, CRAM-MD5, DHX, Digest-MD5, MS-CHAPv2, SMB-NT, SMB-LAN Manager, and WebDAV-Digest. Open Directory Password Server can support a wide range of authentication methods because it is based on the Simple Authentication and Security Layer (SASL) standard.

Open Directory needs to support many different authentication method because each service that requires authentication uses some methods but not others. File service uses one set of authentication methods, Web service uses another set of methods, mail service uses another set, and so on.

Some authentication methods are more secure than others. The more secure methods use tougher algorithms to encode the information that they transmit between client and server. The more secure authentication methods also store passwords in a form that can't be recovered from the server.

Enabling or Disabling Authentication Methods

All password authentication methods supported by Open Directory Password Server are initially enabled. You can disable and enable Open Directory Password Server authentication methods by using the `NeST` command in Terminal. For information, see the command-line administration guide.

When deciding whether to disable or enable authentication methods, your goal should be to provide maximum convenience to legitimate users while keeping other users from gaining access to the server. Consider the following:

- Which types of password validation are needed by the services that my server or servers provide?
- What balance do I want between ease of access and security?
- What types of hardware and software will the server's clients use?
- Is my server in a physically secure location?

Note: Disabling or enabling an authentication method may necessitate resetting passwords in user accounts. If a user can't use additional methods after you enable them, the user or a directory domain administrator needs to reset the user's password.

Basic information about Open Directory Password Server's authentication methods is provided on the following pages. This information is not a substitute for a thorough knowledge of authentication methods and how they affect security and ease of access.

APOP Password Validation

APOP can be used for POP mail service by Mac OS X Server and users' mail client software. It encodes passwords when they are sent over the network, but stores them in a recoverable form on the server. It offers good security during network transmission. A malicious user might be able to obtain passwords by gaining access to the server and reading the password file, although doing this would be difficult. If APOP is disabled, some email programs will transmit passwords over the network in clear text format, which is a significant security risk. If you use your server for POP email, you should probably keep APOP enabled.

CRAM-MD5 Password Validation

CRAM-MD5 can be used for IMAP mail service by Mac OS X Server and users' mail client software. CRAM-MD5 is also used by some LDAP software. This authentication method encodes passwords when they are sent over the network, and stores them in a scrambled form on the server. It offers good security during network transmission. A malicious user might be able to obtain passwords by gaining access to the server and decoding the password file, although doing this would be very difficult. If CRAM-MD5 is disabled, some email programs will transmit passwords over the network in clear text format, which is a significant security risk. If you use your server for SMTP or IMAP email, you should probably keep CRAM-MD5 enabled.

DHX Password Validation

Diffie-Hellman Exchange (DHX) password validation is used by the Apple file service of Mac OS X Server and some other Apple Filing Protocol (AFP) file servers. DHX is required for Open Directory administration and password changes. A malicious user might be able to obtain passwords by gaining access to the server and decoding the password file, although doing this would be very difficult. DHX strongly encodes passwords when they are sent over the network. DHX cannot be disabled.

Mac OS 8.1–8.6 computers must have their AppleShare Client software upgraded to use DHX.

- Mac OS 8.6 computers should use AppleShare Client version 3.8.8.
- Mac OS 8.1–8.5 clients should use AppleShare Client version 3.8.6.
- Mac OS 8.1–8.6 client computers that have file server volumes mount automatically during startup should use AppleShare Client version 3.8.3 with the DHX UAM (User Authentication Module) installed. The DHX UAM is included with the AppleShare Client 3.8.3 installation software.

Digest-MD5 Password Validation

Digest-MD5 is used by the Mac OS X login window, many email programs, and some LDAP software. This authentication method encodes passwords when they are sent over the network, and stores them in a scrambled form on the server. It offers good security during network transmission. A malicious user might be able to obtain passwords by gaining access to the server and decoding the password file, although doing this would be very difficult. Digest-MD5 cannot be disabled.

MS-CHAPv2 Password Validation

MS-CHAPv2 is used by the VPN service of Mac OS X Server. This authentication method encodes passwords when they are sent over the network, and stores them in a scrambled form on the server. It offers good security during network transmission. A malicious user might be able to obtain passwords by gaining access to the server and decoding the password file, although doing this would be very difficult.

SMB-NT Password Validation

SMB-NT password validation is required by default for some Microsoft Windows computers to connect to the Mac OS X Server for Windows services. It is sometimes called Windows Secure Password Exchange (NT). It encodes passwords when they are sent over the network, and stores them in a scrambled form on the server. A malicious user might be able to obtain passwords by gaining access to the server and decoding the password file, although doing this would be very difficult. If SMB-NT password validation is disabled, each individual Windows client system must be configured to work with the server. If you want Windows users to be able to easily share files on your system, you should keep SMB-NT enabled.

SMB-LAN Manager Password Validation

SMB-LAN Manager password validation is required by default for some Microsoft Windows systems to connect to the Mac OS X SMB Server. It is sometimes called Windows Secure Password Exchange (LAN Manager). It encodes passwords when they are sent over the network, and stores them in a scrambled form on the server. A malicious user might be able to obtain passwords by gaining access to the server and decoding the password file, although doing this would be very difficult. If SMB-LAN Manager password validation is disabled, each individual Windows client system must be configured to work with the server. If you want Windows users to be able to easily share files on your system, you should keep SMB-LAN Manager enabled.

WebDAV-Digest Password Validation

WebDAV-Digest handles Digest-MD5 password validation for the WebDAV protocol, which is used to authenticate access to an iDisk. You should keep WebDAV-Digest enabled so that users can mount iDisks and other WebDAV servers in the Finder. WebDAV-Digest encodes passwords when they are sent over the network, and stores them in a scrambled form on the server. It offers good security during network transmission. A malicious user might be able to obtain passwords by gaining access to the server and decoding the password file, although doing this would be very difficult.

Authentication Manager

Mac OS X Server supports users that were configured to use the legacy Authentication Manager technology in Mac OS X Server version 10.0–10.2.

Authentication Manager is a legacy technology for securely validating passwords of the following users:

- Users of Windows services (including support for SMB-NT, SMB-LM, and CRAM-MD5)
- Users of Apple file service whose Mac OS 8 computers have not been upgraded with AFP client software version 3.8.3 or later
- Users who need to authenticate for mail service by using APOP or CRAM-MD5

Authentication Manager only works with user accounts that were created in a NetInfo domain of Mac OS X Server version 10.0–10.2. Authentication Manager must have been enabled for the NetInfo domain.

When you upgrade a server to Mac OS X Server version 10.3 from an earlier version that has Authentication Manager enabled, it remains enabled. Existing users can continue to use their same passwords. An existing user account uses Authentication Manager if the account is in a NetInfo domain for which Authentication Manager has been enabled and the account is set to use a crypt password.

After upgrading a server to Mac OS X Server version 10.3, you can change existing user accounts to authenticate using Open Directory. Open Directory authentication is the preferred authentication option for users of Windows services and is required for domain login from a Windows workstation to a Mac OS X Server primary domain controller. New user accounts created in Mac OS X Server version 10.3 are set to use Open Directory authentication.

Glossary

Active Directory The directory service of Microsoft Windows 2000 and 2003 servers.

administrator A user with server or *directory domain* administration privileges. Administrators are always members of the predefined "admin" group.

administrator computer A Mac OS X computer onto which you have installed the server administration applications from the Mac OS X Server Admin CD.

AFP (Apple Filing Protocol) A client/server protocol used by Apple file service on Macintosh-compatible computers to share files and network services. AFP uses TCP/IP and other protocols to communicate between computers on a network.

authentication The process of proving a user's identity, typically by validating a user name and password. Usually authentication occurs before an *authorization* process determines the user's level of access to a resource. For example, file service authorizes full access to folders and files that an authenticated user owns.

authentication authority attribute A value that identifies the password validation scheme specified for a user and provides additional information as required.

authorization The process by which a service determines whether it should grant a user access to a resource and how much access the service should allow the user to have. Usually authorization occurs after an *authentication* process proves the user's identity. For example, file service authorizes full access to folders and files that an authenticated user owns.

BSD (Berkeley System Distribution) A version of UNIX on which Mac OS X software is based.

child A computer that gets configuration information from the shared directory domain of a *parent*.

class See *object class*.

computer account A list of computers that have the same preference settings and are available to the same users and groups.

DHCP (Dynamic Host Configuration Protocol) A protocol used to distribute IP addresses to client computers. Each time a client computer starts up, the protocol looks for a DHCP server and then requests an IP address from the DHCP server it finds. The DHCP server checks for an available IP address and sends it to the client computer along with a *lease period*—the length of time the client computer may use the address.

directory domain A specialized database that stores authoritative information about users and network resources; the information is needed by system software and applications. The database is optimized to handle many requests for information and to find and retrieve information quickly. Also called a directory node or simply a directory.

directory domain hierarchy A way of organizing local and shared directory domains. A hierarchy has an inverted tree structure, with a root domain at the top and local domains at the bottom.

directory node See *directory domain*.

directory services Services that provide system software and applications with uniform access to directory domains and other sources of information about users and resources.

FTP (File Transfer Protocol) A protocol that allows computers to transfer files over a network. FTP clients using any operating system that supports FTP can connect to a file server and download files, depending on their access privileges. Most Internet browsers and a number of freeware applications can be used to access an FTP server.

group A collection of users who have similar needs. Groups simplify the administration of shared resources.

group directory A directory that organizes documents and applications of special interest to group members and allows group members to pass information back and forth among them.

guest user A user who can log in to your server without a user name or password.

hash An encrypted form of a password or other text.

home directory A folder for a user's personal use. Mac OS X also uses the home directory, for example, to store system preferences and managed user settings for Mac OS X users.

IP (Internet Protocol) Also known as IPv4. A method used with Transmission Control Protocol (TCP) to send data between computers over a local network or the Internet. IP delivers packets of data, while TCP keeps track of data packets.

IP address A unique numeric address that identifies a computer on the Internet.

Kerberos A secure network authentication system. Kerberos uses tickets, which are issued for a specific user, service, and period of time. Once a user is authenticated, it is possible to access additional services without retyping a password (this is called single-signon) for services that have been configured to take Kerberos tickets. Mac OS X Server uses Kerberos v5.

LDAP (Lightweight Directory Access Protocol) A standard client-server protocol for accessing a directory domain.

local domain A directory domain that can be accessed only by the computer on which it resides.

long name See *user name*.

Mac OS X The latest version of the Apple operating system. Mac OS X combines the reliability of UNIX with the ease of use of Macintosh.

Mac OS X Server An industrial-strength server platform that supports Mac, Windows, UNIX, and Linux clients out of the box and provides a suite of scalable workgroup and network services plus advanced remote management tools.

managed client A user, group, or computer whose access privileges and/or preferences are under administrative control.

managed preferences System or application preferences that are under administrative control. Workgroup Manager allows administrators to control settings for certain system preferences for Mac OS X managed clients. Macintosh Manager allows administrators to control both system preferences and application preferences for Mac OS 9 and Mac OS 8 managed clients.

NetInfo One of the Apple protocols for accessing a *directory domain*.

object class A set of rules that define similar objects in a directory domain by specifying attributes that each object must have and other attributes that each object may have.

Open Directory The Apple directory services architecture, which can access authoritative information about users and network resources from directory domains that use *LDAP, NetInfo,* or *Active Directory* protocols; *BSD* configuration files; and network services.

open source A term for the cooperative development of software by the Internet community. The basic principle is to involve as many people as possible in writing and debugging code by publishing the source code and encouraging the formation of a large community of developers who will submit modifications and enhancements.

owner The person who created a file or folder and who therefore has the ability to assign access privileges for other users. The owner of an item automatically has read/write privileges for that item. An owner can also transfer ownership of an item to another user.

parent A computer whose shared directory domain provides configuration information to another computer.

primary group A user's default group. The file system uses the ID of the primary group when a user accesses a file he or she doesn't own.

primary group ID A unique number that identifies a primary group.

protocol A set of rules that determines how data is sent back and forth between two applications.

Rendezvous A protocol developed by Apple for automatic discovery of computers, devices, and services on IP networks. This proposed Internet standard protocol is sometimes referred to as "ZeroConf" or "multicast DNS." For more information, visit www.apple.com or www.zeroconf.org.

schema The collection of attributes and record types or classes that provide a blueprint for the information in a *directory domain*.

search path See *search policy*.

search policy A list of directory domains searched by a Mac OS X computer when it needs configuration information; also the order in which domains are searched. Sometimes called a search path.

share point A folder, hard disk (or hard disk partition), or CD that is accessible over the network. A share point is the point of access at the top level of a group of shared items. Share points can be shared using *AFP,* Windows *SMB, NFS* (an "export"), or *FTP* protocols.

short name An abbreviated name for a user. The short name is used by Mac OS X for home directories, authentication, and email addresses.

single signon An authentication strategy that relieves users from entering a name and password separately for every network service. Mac OS X Server uses Kerberos to enable single signon.

SLP (Service Location Protocol) DA (Directory Agent) A protocol that registers services available on a network and gives users easy access to them. When a service is added to the network, the service uses SLP to register itself on the network. SLP/DA uses a centralized repository for registered network services.

SMB (Server Message Block) A protocol that allows client computers to access files and network services. It can be used over TCP/IP, the Internet, and other network protocols. Windows services use SMB to provide access to servers, printers, and other network resources.

SSL (Secure Sockets Layer) An Internet protocol that allows you to send encrypted, authenticated information across the Internet.

user name The long name for a user, sometimes referred to as the user's "real" name. See also *short name*.

WebDAV (Web-based Distributed Authoring and Versioning) A live authoring environment that allows client users to check out webpages, make changes, and then check the pages back in while a site is running.

WebDAV realm A region of a website, usually a folder or directory, that is defined to provide access for WebDAV users and groups.

Index

D

E

F

G

H

I

O

P

Q

R

S

T

U

V

W