




Mac OS X Server

Open Directory Administration
For Version 10.4 or Later

 Apple Computer, Inc.
© 2005 Apple Computer, Inc. All rights reserved.

The owner or authorized user of a valid copy of Mac OS X Server software may reproduce this publication for the purpose of learning to use such software. No part of this publication may be reproduced or transmitted for commercial purposes, such as selling copies of this publication or for providing paid-for support services.

Every effort has been made to ensure that the information in this manual is accurate. Apple Computer, Inc., is not responsible for printing or clerical errors.

Apple
1 Infinite Loop
Cupertino CA 95014-2084
www.apple.com

The Apple logo is a trademark of Apple Computer, Inc., registered in the U.S. and other countries. Use of the “keyboard” Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

Apple, the Apple logo, AppleTalk, Mac, and Macintosh are trademarks of Apple Computer, Inc., registered in the U.S. and other countries. Finder is a trademark of Apple Computer, Inc.

Adobe and PostScript are trademarks of Adobe Systems Incorporated.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd.

Other company and product names mentioned herein are trademarks of their respective companies. Mention of third-party products is for informational purposes only and constitutes neither an endorsement nor a recommendation. Apple assumes no responsibility with regard to the performance or use of these products.

019-0166/03-24-2005

Contents

Preface	11 About This Guide
	12 What's New in Version 10.4
	13 What's in This Guide
	14 Using This Guide
	14 Using Onscreen Help
	15 The Mac OS X Server Suite
	16 Getting Documentation Updates
	16 Getting Additional Information
Chapter 1	19 Directory Service With Open Directory
	20 Directory Services and Directory Domains
	21 A Historical Perspective
	22 Data Consolidation
	23 Data Distribution
	24 Uses of Directory Data
	25 Access to Directory Services
	26 Discovery of Network Services
	28 Inside a Directory Domain
	29 Structure of LDAP Directory Information
	30 Local and Shared Directory Domains
	30 About the Local Directory Domain
	31 About Shared Directory Domains
	32 Shared Data in Existing Directory Domains
Chapter 2	33 Open Directory Search Policies
	33 Search Policy Levels
	34 Local Directory Search Policy
	34 Two-Level Search Policies
	35 Multilevel Search Policies
	36 Automatic Search Policies
	38 Custom Search Policies
	38 Search Policies for Authentication and Contacts

Chapter 3	39 Open Directory Authentication
	39 Password Types
	40 Authentication and Authorization
	40 Open Directory Passwords
	41 Shadow Passwords
	41 Crypt Passwords
	42 Offline Attacks on Passwords
	43 Determining Which Authentication Option to Use
	44 Password Policies
	44 Single Sign-On Authentication
	45 Kerberos Authentication
	46 Breaking the Barriers to Kerberos Deployment
	46 Single Sign-On Experience
	47 Secure Authentication
	47 Ready to Move Beyond Passwords
	48 Multiplatform Authentication
	48 Centralized Authentication
	48 Kerberized Services
	48 Kerberos Principals and Realms
	49 Kerberos Authentication Process
	50 Open Directory Password Server and Shadow Password Authentication Methods
	51 Disabling Open Directory Authentication Methods
	52 Disabling Shadow Password Authentication Methods
	53 Contents of Open Directory Password Server Database
	54 LDAP Bind Authentication
	54 Authentication Manager
 Chapter 4	 57 Open Directory Planning
	57 General Planning Guidelines
	59 Controlling Data Accessibility
	59 Simplifying Changes to Data in Directories
	60 Estimating Directory and Authentication Requirements
	60 Identifying Servers for Hosting Shared Domains
	61 Replicating Open Directory Services
	62 Load Balancing in Small, Medium, and Large Environments
	62 Replication in a Multibuilding Campus
	63 Using an Open Directory Master or Replica With NAT
	63 Avoiding Kerberos Conflicts With Multiple Directories
	65 Improving Performance and Redundancy
	65 Open Directory Security
	67 Tools for Managing Open Directory Services
	67 Server Admin
	68 Directory Access

- 68 Workgroup Manager
- 69 Command-Line Tools
- 69 NetInfo Manager

Chapter 5

- 71 **Setting Up Open Directory Services**
- 71 Setup Overview
- 72 Before You Begin
- 73 Setting Up Open Directory With Server Assistant
- 73 Managing Open Directory on a Remote Server
- 73 Setting Up a Standalone Server
- 74 Open Directory Master and Replica Compatibility
- 75 Setting Up an Open Directory Master
 - 76 Instructing Users How to Log In
- 77 Setting Up an Open Directory Replica
 - 79 Creating Multiple Replicas of an Open Directory Master
- 79 Setting Up Open Directory Failover
- 80 Setting Up a Connection to a Directory System
- 81 Setting Up Single Sign-On Kerberos Authentication
 - 82 Setting Up an Open Directory Kerberos Realm
 - 82 Starting Kerberos After Setting Up an Open Directory Master
 - 83 Delegating Authority to Join an Open Directory Kerberos Realm
 - 85 Joining a Server to a Kerberos Realm
- 86 Setting Options for an Open Directory Master or Replica
 - 86 Setting a Binding Policy for an Open Directory Master and Replicas
 - 87 Setting a Security Policy for an Open Directory Master and Replicas
 - 88 Changing the Location of an LDAP Database
 - 88 Limiting Search Results for LDAP Service
 - 89 Changing the Search Timeout for LDAP Service
 - 89 Setting up SSL for LDAP Service
- 90 Migrating a Directory Domain From Netinfo to LDAP
- 92 Switching Directory Access From NetInfo to LDAP
- 92 Disabling NetInfo After Migrating to LDAP

Chapter 6

- 93 **Managing User Authentication**
- 93 Composing a Password
- 94 Changing a User's Password
- 95 Resetting the Passwords of Multiple Users
- 96 Changing a User's Password Type
 - 96 Changing the Password Type to Open Directory
 - 97 Changing the Password Type to Crypt Password
 - 98 Changing the Password Type to Shadow Password
- 99 Enabling Single Sign-On Kerberos Authentication for a User
- 99 Changing the Global Password Policy

- 100 Setting Password Policies for Individual Users
- 101 Selecting Authentication Methods for Shadow Password Users
- 102 Selecting Authentication Methods for Open Directory Passwords
- 103 Assigning Administrator Rights for Open Directory Authentication
- 104 Keeping the Primary Administrator's Passwords in Sync
- 104 Enabling LDAP Bind Authentication for a User
- 105 Setting Passwords of Exported or Imported Users
- 105 Migrating Passwords From Mac OS X Server v10.1 or Earlier
- 107 Exporting and Importing Authentication Manager Users

Chapter 7

- 109 **Managing Directory Access**
- 109 Setting Up Directory Access on a Remote Server
- 110 Configuring Access to Services
 - 110 Enabling or Disabling Active Directory Service
 - 111 Enabling or Disabling AppleTalk Service Discovery
 - 111 Enabling or Disabling BSD Flat File and NIS Directory Services
 - 111 Enabling or Disabling LDAP Directory Services
 - 112 Enabling or Disabling NetInfo Directory Services
 - 112 Enabling Bonjour Service Discovery
 - 112 Enabling or Disabling SLP Service Discovery
 - 113 Enabling or Disabling SMB/CIFS Service Discovery
 - 113 Configuring SMB/CIFS Service Discovery
- 113 Setting Up Search Policies
 - 114 Defining Automatic Search Policies
 - 115 Defining Custom Search Policies
 - 116 Defining Local Directory Search Policies
 - 117 Waiting for a Search Policy Change to Take Effect
 - 117 Protecting Computers From a Malicious DHCP Server
- 117 Accessing LDAP Directories
 - 118 Accessing LDAP Directories in Mail and Address Book
 - 118 Enabling or Disabling Use of a DHCP-Supplied LDAP Directory
 - 119 Showing or Hiding Configurations for LDAP Servers
 - 120 Configuring Access to an LDAP Directory
 - 122 Configuring Access to an LDAP Directory Manually
 - 124 Changing a Configuration for Accessing an LDAP Directory
 - 125 Duplicating a Configuration for Accessing an LDAP Directory
 - 126 Deleting a Configuration for Accessing an LDAP Directory
 - 127 Changing the Connection Settings for an LDAP Directory
 - 128 Changing the Security Policy for an LDAP Connection
 - 129 Configuring LDAP Searches and Mappings
 - 132 Setting Up Trusted Binding to an LDAP Directory
 - 133 Stopping Trusted Binding With an LDAP Directory
 - 133 Changing the Open/Close Timeout for an LDAP Connection

134	Changing the Query Timeout for an LDAP Connection
134	Changing the Rebind-Try Delay Time for an LDAP Connection
134	Changing the Idle Timeout for an LDAP Connection
135	Forcing Read-Only LDAPv2 Access
135	Ignoring LDAP Server Referrals
135	Authenticating an LDAP Connection
136	Changing the Password Used for Authenticating an LDAP Connection
136	Mapping Config Record Attributes for LDAP Directories
137	Editing RFC 2307 Mapping to Enable Creating Users
138	Preparing a Read-Only LDAP Directory for Mac OS X
138	Populating LDAP Directories With Data for Mac OS X
139	Accessing an Active Directory Domain
139	About the Active Directory Plug-in
141	Configuring Access to an Active Directory Domain
143	Setting Up Mobile User Accounts in Active Directory
143	Setting Up Home Folders for Active Directory User Accounts
144	Setting a UNIX Shell for Active Directory User Accounts
145	Mapping the UID to an Active Directory Attribute
145	Mapping the Primary Group ID to an Active Directory Attribute
146	Mapping the Group ID in Group Accounts to an Active Directory Attribute
147	Specifying a Preferred Active Directory Server
147	Changing the Active Directory Groups That Can Administer the Computer
148	Controlling Authentication From All Domains in the Active Directory Forest
149	Unbinding From the Active Directory Server
149	Editing User Accounts and Other Records in Active Directory
149	Setting Up LDAP Access to Active Directory Domains
151	Accessing an NIS Domain
151	Using BSD Configuration Files
152	Setting Up Data in BSD Configuration Files
153	Accessing Legacy NetInfo Domains
153	About NetInfo Binding
154	Configuring NetInfo Binding
155	Adding a Machine Record to a Parent NetInfo Domain
156	Configuring Static Ports for Shared NetInfo Domains

Chapter 8

157	Maintenance and Problem Solving
157	Controlling Access to Open Directory Servers
157	Controlling Access to a Server's Login Window
158	Controlling Access to SSH Service
159	Monitoring Open Directory
159	Checking the Status of an Open Directory Master or Replica
160	Monitoring Replicas of an Open Directory Master
160	Viewing Open Directory Status and Logs

160	Monitoring Open Directory Authentication
161	Directly Viewing and Editing Directory Data
161	Showing the Directory Inspector
162	Hiding the Directory Inspector
162	Changing a User's Short Name
163	Setting Directory Access Controls (DACs)
163	Deleting Records
164	Importing Records of Any Type
164	Managing Open Directory Replication
164	Scheduling Replication of an Open Directory Master
165	Synchronizing an Open Directory Replica on Demand
165	Promoting an Open Directory Replica
166	Decommissioning an Open Directory Replica
167	Archiving an Open Directory Master
168	Restoring an Open Directory Master
169	Solving Open Directory Master and Replica Problems
169	Kerberos is Stopped on an Open Directory Master or Replica
170	Can't Create an Open Directory Replica
170	Solving Directory Access Problems
170	A Delay Occurs During Startup
171	Solving Authentication Problems
171	You Can't Modify a User's Open Directory Password
171	A User Can't Access Some Services
171	A User Can't Authenticate for VPN Service
172	You Can't Change a User's Password Type to Open Directory
172	Users Relying on a Password Server Can't Log In
172	Users Can't Log In With Accounts in a Shared Directory Domain
172	Can't Log In as Active Directory User
173	Users Can't Authenticate Using Single Sign-On or Kerberos
174	Users Can't Change Their Passwords
174	Can't Join a Server to an Open Directory Kerberos Realm
175	Resetting an Administrator Password

Appendix

177	Mac OS X Directory Data
178	Open Directory Extensions to LDAP Schema
178	Object Classes in Open Directory LDAP Schema
185	Attributes in Open Directory LDAP Schema
201	Mapping Standard Record Types and Attributes to LDAP and Active Directory
201	Mappings for Users
205	Mappings for Groups
206	Mappings for Mounts
207	Mappings for Computers
208	Mappings for ComputerLists

209	Mappings for Config
210	Mappings for People
211	Mappings for PresetComputerLists
212	Mappings for PresetGroups
213	Mappings for PresetUsers
214	Mappings for Printers
215	Mappings for AutoServerSetup
216	Mappings for Locations
216	Standard Open Directory Record Types and Attributes
217	Standard Attributes in User Records
222	Standard Attributes in Group Records
223	Standard Attributes in Computer Records
224	Standard Attributes in Computer List Records
224	Standard Attributes in Mount Records
225	Standard Attributes in Config Records
Glossary	227
Index	235

This guide describes the directory and authentication services you can set up using Mac OS X Server. It also explains how to configure Mac OS X Server and Mac OS X client computers for directory services and discovery of network services.

Mac OS X Server's Open Directory provides directory and authentication services for mixed networks of Mac OS X, Windows, and UNIX computers. Open Directory uses OpenLDAP, the open source implementation of the Lightweight Directory Access Protocol (LDAP), to provide directory services. It's compatible with other standards-based LDAP servers, and can be integrated with proprietary services such as Microsoft's Active Directory and Novell's eDirectory. For the LDAP database backend, Open Directory uses open source Berkeley DB. It's a highly scalable database for high-performance indexing of hundreds of thousands of user accounts and other records.

Open Directory plug-ins enable a Mac OS X client or Mac OS X Server computer to read and write authoritative information about users and network resources from any LDAP server—even Microsoft's proprietary Active Directory. The server can also access records in legacy directories such as NIS, NetInfo, and local BSD configuration files (/etc).

Open Directory also provides authentication service. It 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, and other Windows services provided by Mac OS X Server.

An MIT Kerberos Key Distribution Center (KDC) is fully integrated with Open Directory and provides strong authentication with support for secure single sign-on. This means users need authenticate only once, with a single user name and password pair, for access to the range of Kerberos-enabled network services. For services that don't accept Kerberos authentication, the integrated Secure Authentication and Service Layer (SASL) service automatically negotiates the strongest possible authentication mechanism.

In addition, directory and authentication replication maximizes availability and scalability. By creating replicas of Open Directory servers, you can easily maintain failover servers as well as remote servers for fast client interaction on distributed networks.

Open Directory also manages discovery of network services. 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 Bonjour, AppleTalk, SLP, or SMB/CIFS service discovery protocols.

What's New in Version 10.4

Mac OS X Server version 10.4 offers the following major enhancements in Open Directory:

- **Simplified configuration of LDAPv3 access:** Directory Access assists you in setting up a connection to an LDAP directory.
- **Trusted LDAPv3 directory binding:** Establishes a mutually authenticated connection between the LDAP directory and its clients. The client proves its identity to the LDAP directory, and the directory proves its authenticity to the client.
- **Improved Active Directory integration:** You can have Mac OS X users' network home directories mounted from the location specified in Active Directory. You can map several Mac OS X attributes—user ID, user primary group ID, and group ID—to existing Active Directory attributes.
- **Improved LDAP server:** Mac OS X Server v10.4 uses OpenLDAP version 2.2.19 and Berkeley DB version 4.2.52.
- **Simplified archive and restore:** Click a button to back up or restore directory and authentication databases.
- **Improved authentication:** You can join a server to an existing Active Directory Kerberos realm or an MIT-based Kerberos realm. Local user accounts can use more authentication methods.
- **Configurable security of password storage:** Authentication methods can be selectively disabled to make password storage on the server more secure.
- **LDAP schema replication:** You can have the LDAP directory store its own custom schema and propagate the schema from the Open Directory master to all its replicas.

What's in This Guide

This guide includes the following chapters:

- Chapter 1, “Directory Service With Open Directory,” 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, “Open Directory Authentication,” describes Open Directory authentication, shadow and crypt passwords, Kerberos, LDAP bind, and single sign-on.
- Chapter 4, “Open Directory Planning,” helps you assess your directory domain needs, estimate directory and authentication requirements, identify servers for hosting shared domains, improve performance and redundancy, deal with replication in a multibuilding 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 sign-on 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 to enable, disable, and configure service discovery protocols. It also explains how to configure authentication and contacts search policies. In addition, this chapter 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, archive an Open Directory master, and perform other directory maintenance. This chapter also describes solutions to some problems you may encounter.
- Appendix, “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.
- The Glossary defines terms you’ll encounter as you read this guide.

Note: Because Apple frequently releases new versions and updates to its software, images shown in this book may be different from what you see on your screen.

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.

Using Onscreen Help

You can view instructions and other useful information from this and other documents in the server suite by using onscreen help.

On a computer running Mac OS X Server, you can access onscreen help after opening Workgroup Manager or Server Admin. From the Help menu, select one of the options:

- *Workgroup Manager Help* or *Server Admin Help* displays information about the application.
- *Mac OS X Server Help* displays the main server help page, from which you can search or browse for server information.
- *Documentation* takes you to www.apple.com/server/documentation, from which you can download server documentation.

You can also access onscreen help from the Finder or other applications on a server or on an administrator computer. (An administrator computer is a Mac OS X computer with server administration software installed on it.) Use the Help menu to open Help Viewer, and then choose Library > Mac OS X Server Help.

To see the latest server help topics, make sure the server or administrator computer is connected to the Internet while you're using Help Viewer. Help Viewer automatically retrieves and caches the latest server help topics from the Internet. When not connected to the Internet, Help Viewer displays cached help topics.

The Mac OS X Server Suite

The Mac OS X Server documentation includes a suite of guides that explain the services and provide instructions for configuring, managing, and troubleshooting the services.

All of the guides are available in PDF format from:

www.apple.com/server/documentation/

This guide ...	tells you how to:
<i>Mac OS X Server Getting Started for Version 10.4 or Later</i>	Install Mac OS X Server and set it up for the first time.
<i>Mac OS X Server Upgrading and Migrating to Version 10.4 or Later</i>	Use data and service settings that are currently being used on earlier versions of the server.
<i>Mac OS X Server User Management for Version 10.4 or Later</i>	Create and manage users, groups, and computer lists. Set up managed preferences for Mac OS X clients.
<i>Mac OS X Server File Services Administration for Version 10.4 or Later</i>	Share selected server volumes or folders among server clients using these protocols: AFP, NFS, FTP, and SMB/CIFS.
<i>Mac OS X Server Print Service Administration for Version 10.4 or Later</i>	Host shared printers and manage their associated queues and print jobs.
<i>Mac OS X Server System Image and Software Update Administration for Version 10.4 or Later</i>	Use NetBoot and Network Install to create disk images from which Macintosh computers can start up over the network. Set up a software update server for updating client computers over the network.
<i>Mac OS X Server Mail Service Administration for Version 10.4 or Later</i>	Set up, configure, and administer mail services on the server.
<i>Mac OS X Server Web Technologies Administration for Version 10.4 or Later</i>	Set up and manage a web server, including WebDAV, WebMail, and web modules.
<i>Mac OS X Server Network Services Administration for Version 10.4 or Later</i>	Set up, configure, and administer DHCP, DNS, VPN, NTP, IP firewall, and NAT services on the server.
<i>Mac OS X Server Open Directory Administration for Version 10.4 or Later</i>	Manage directory and authentication services.
<i>Mac OS X Server QuickTime Streaming Server Administration for Version 10.4 or Later</i>	Set up and manage QuickTime streaming services.
<i>Mac OS X Server Windows Services Administration for Version 10.4 or Later</i>	Set up and manage services including PDC, BDC, file, and print for Windows computer users.
<i>Mac OS X Server Migrating from Windows NT for Version 10.4 or Later</i>	Move accounts, shared folders, and services from Windows NT servers to Mac OS X Server.

This guide ...	tells you how to:
<i>Mac OS X Server Java Application Server Administration For Version 10.4 or Later</i>	Configure and administer a JBoss application server on Mac OS X Server.
<i>Mac OS X Server Command-Line Administration for Version 10.4 or Later</i>	Use commands and configuration files to perform server administration tasks in a UNIX command shell.
<i>Mac OS X Server Collaboration Services Administration for Version 10.4 or Later</i>	Set up and manage weblog, chat, and other services that facilitate interactions among users.
<i>Mac OS X Server High Availability Administration for Version 10.4 or Later</i>	Manage failover, link aggregation, load balancing, and other hardware and software configurations to ensure high availability of Mac OS X Server services.
<i>Mac OS X Server Xgrid Administration for Version 10.4 or Later</i>	Manage computational Xserve clusters using the Xgrid application.
<i>Mac OS X Server Glossary: Includes Terminology for Mac OS X Server, Xserve, Xserve RAID, and Xsan</i>	Interpret terms used for server and storage products.

Getting Documentation Updates

Periodically, Apple posts new onscreen help topics, revised guides, and solution papers. The new help topics include updates to the latest guides.

- To view new onscreen help topics, make sure your server or administrator computer is connected to the Internet and click the Late-Breaking News link on the main Mac OS X Server help page.
- To download the latest guides and solution papers in PDF format, go to the Mac OS X Server documentation webpage: www.apple.com/server/documentation.

Getting Additional Information

For more information, consult these resources:

Read Me documents—important updates and special information. Look for them on the server discs.

Mac OS X Server website (www.apple.com/macosx/server/)—gateway to extensive product and technology information.

AppleCare Service & Support website (www.apple.com/support/)—access to hundreds of articles from Apple's support organization.

Apple customer training (train.apple.com/)—instructor-led and self-paced courses for honing your server administration skills.

Apple discussion groups (discussions.info.apple.com/)—a way to share questions, knowledge, and advice with other administrators.

Apple mailing list directory (www.lists.apple.com/)—subscribe to mailing lists so you can communicate with other administrators using email.

OpenLDAP website (www.openldap.org)—learn about the open source software that Open Directory uses to provide LDAP directory service.

MIT Kerberos website (web.mit.edu/kerberos/www/)—get background information and specifications for the protocol that Open Directory uses to provide robust single sign-on authentication.

Berkeley DB website (www.sleepycat.com/)—investigate feature descriptions and technical documentation for the open source database that Open Directory uses to store LDAP directory data.

RFC3377, "Lightweight Directory Access Protocol (v3): Technical Specification" (www.rfc-editor.org/rfc/rfc3377.txt)—lists a set of eight other Request for Comment (RFC) documents with overview information and detailed specifications for the LDAPv3 protocol.

Directory Service With Open Directory

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:

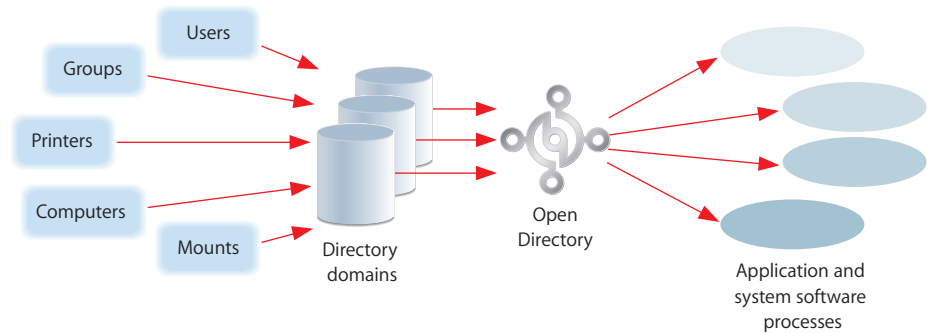
- Reduces data entry effort.
- Ensures all network services and clients have consistent information about users and resources.
- Simplifies administration of users and resources.
- Provides identification, authentication, and authorization information for other network services.

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 simplify system and network administration, and they simplify a user's experience on the network. With directory services, administrators can maintain information about all the users—such as their names, passwords, and locations of network home directories—centrally rather than on each computer. 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 information management burden. And each user has a centralized user account for logging in on any authorized computer on the network. With centralized directory service and file service set up to host network home directories, everywhere a user logs in, the user gets the same home directory, personal desktop, and individual preferences. The user always has access to personal files and can easily locate and use authorized network resources.

Directory Services and Directory Domains

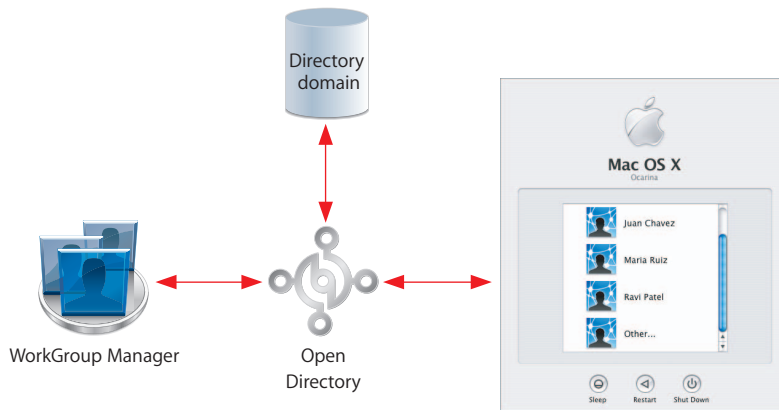
A directory service acts as an intermediary between application and system software processes, which need information about users and resources, and the *directory domains* that store the information. In Mac OS X and Mac OS X Server, Open Directory provides directory services. Open Directory can access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

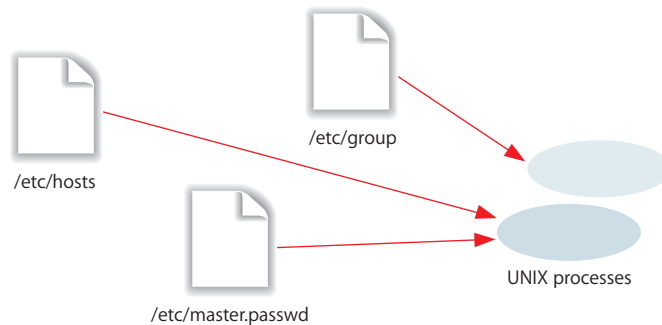


A Historical Perspective

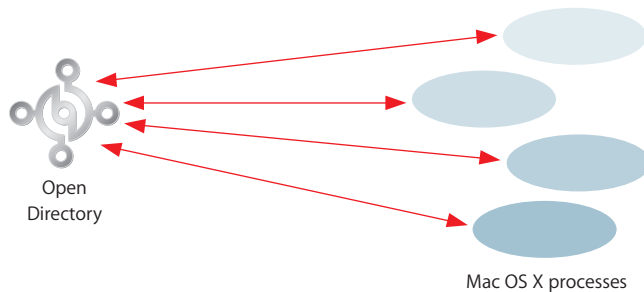
Like Mac OS X, Open Directory has a UNIX heritage. Open Directory provides access to administrative data that UNIX systems have generally kept in configuration files, which require much painstaking work to maintain. (Some UNIX systems still rely on configuration files.) Open Directory consolidates the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

Data Consolidation

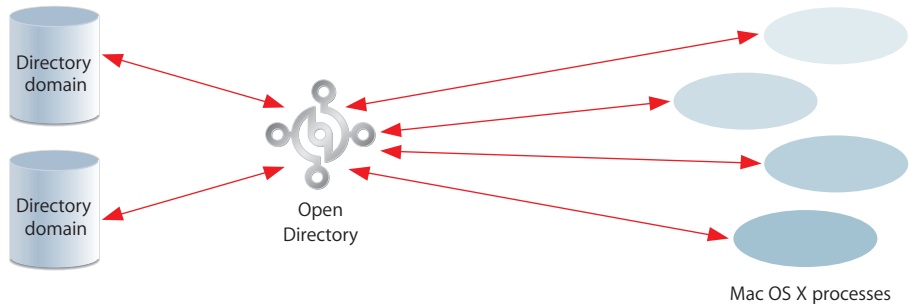
For years, UNIX systems have stored administrative information in a collection of files located in the `/etc` directory. This scheme requires each UNIX computer to have its own set of files, and processes that are running on a UNIX computer read its files when they need administrative information. If you're experienced with UNIX, you probably know about the files in the `/etc` directory—`group`, `hosts`, `hosts.equiv`, `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.



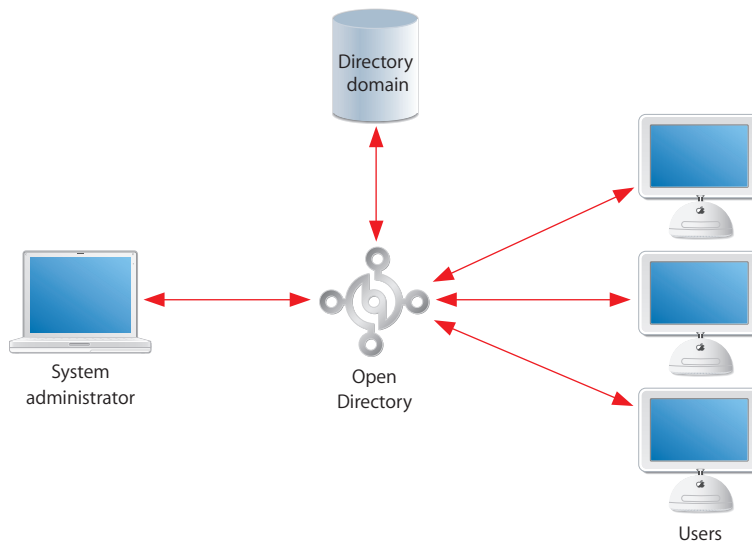
Some of the data stored in a directory domain is identical to data stored in UNIX configuration files. For example, the home directory location, real name, user ID, and group ID are stored in the user records of a directory domain instead of the standard `/etc/passwd` file. However, a directory domain 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 other data from the user record to determine the user's access privileges for each file or folder.

- **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.
- **Managed network views:** The administrator can set up custom views that users see when they select the Network icon in the sidebar of a Finder window. Because these managed network views are stored in a directory domain, they're available automatically when a user logs in.

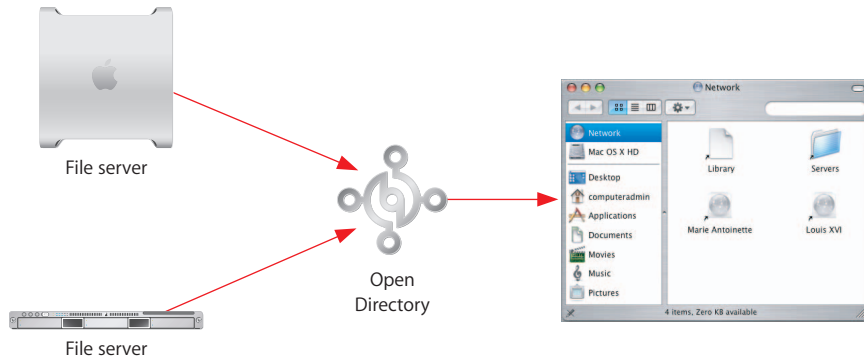
Access to Directory Services

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

- Lightweight Directory Access Protocol (LDAP), an open standard common in mixed environments of Macintosh, UNIX, and Windows systems. LDAP is the native directory service for shared directories in Mac OS X Server.
- NetInfo, the directory service for the local directory domain on every Mac OS X system. It's the legacy directory service of Mac OS X Server.
- Active Directory, the directory service of Microsoft Windows 2000 and 2003 servers
- Network Information System (NIS), the 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.

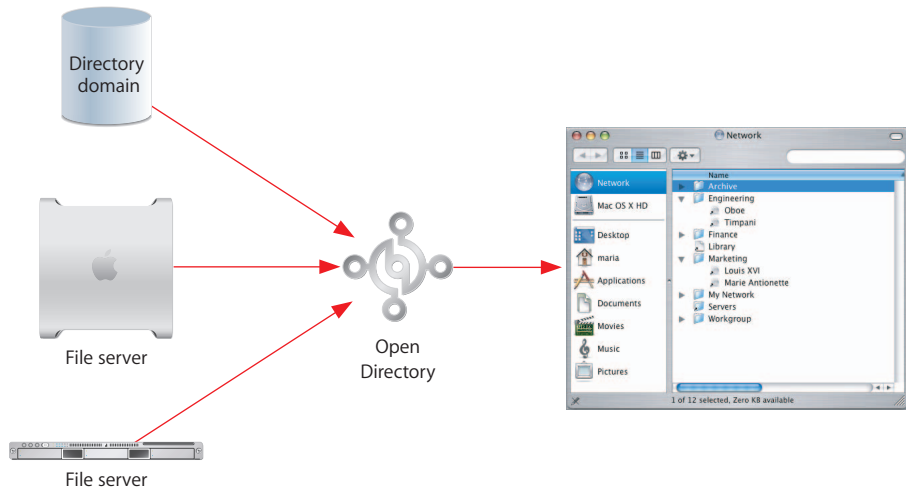


Open Directory can discover network services that make their existence and whereabouts known. Services make themselves known by means of standard protocols. Open Directory supports the following service discovery protocols:

- Bonjour, the Apple protocol that uses multicast DNS for discovering file, print, chat, music sharing, and other services on IP networks
- AppleTalk, the legacy protocol for discovering file, print, and other network services
- Service Location Protocol (SLP), an open standard protocol for discovering file and print services on IP networks
- Server Message Block/Common Internet File System (SMB/CIFS), 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, Bonjour, SMB/CIFS, 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.

Inside a Directory Domain

Information in a directory domain is organized by *record type*. Record types are specific categories of information, such as users, groups, and computers. 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 account by using Workgroup Manager, you are creating a user record (a record of the “user” record type). The settings that you configure for the user account—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.

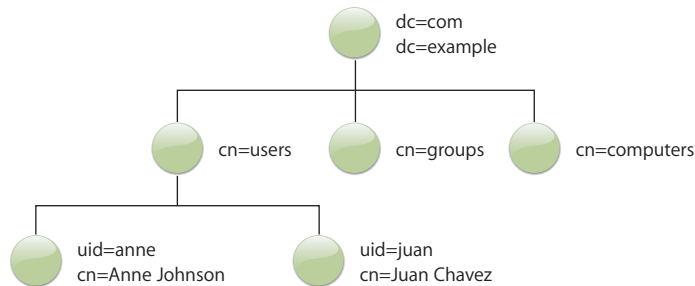
In some directory services, such as LDAP and Active Directory, directory information is organized by *object class*. Like record types, object classes define categories of information. An object class defines similar information objects, called *entries*, by specifying attributes that an entry must or may contain. For a particular object class, a directory domain may contain multiple entries, and each entry may contain multiple attributes. Some attributes have a single value, while others have multiple values. For example, the `inetOrgPerson` object class defines entries that contain user attributes. The `inetOrgPerson` class is a standard LDAP class defined by RFC 2798. Other standard LDAP object classes and attributes are defined by RFC 2307. Open Directory’s default object classes and attributes are based on these RFCs.

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.

Structure of LDAP Directory Information

In an LDAP directory, entries are arranged in a hierarchical tree-like structure. In some LDAP directories, this structure is based on geographic and organizational boundaries. More commonly, the structure is based upon Internet domain names.

In a simple directory organization, entries representing users, groups, computers, and other object classes are immediately below the root level of the hierarchy.



An entry is referenced by its *distinguished name* (DN), which is constructed by taking the name of the entry itself, called the *relative distinguished name* (RDN), and concatenating the names of its ancestor entries. For example, the entry for Anne Johnson could have an RDN of “uid=anne” and a DN of “uid=anne, cn=users, dc=example, dc=com.”

The LDAP service retrieves data by searching the hierarchy of entries. The search can begin at any entry. The entry at which the search begins is called the *search base*. You can specify a search base by giving the distinguished name of an entry in the LDAP directory. For example, the search base “cn=users, dc=example, dc=com” specifies that the LDAP service will begin searching at the entry whose “cn” attribute has a value of “users.”

You can also specify how much of the LDAP hierarchy to search below the search base. The search scope can include all subtrees below the search base or just the first level of entries below the search base. If you use command-line tools to search an LDAP directory, you can also restrict the search scope to just the search base entry.

Local and Shared Directory Domains

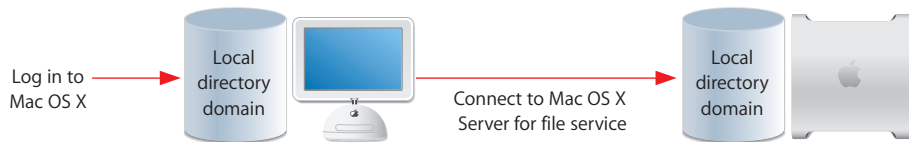
Where you store your server's user information and other administrative data is determined by whether the data needs to be shared. This information may be stored in the server's local directory domain or in a shared directory domain.

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.

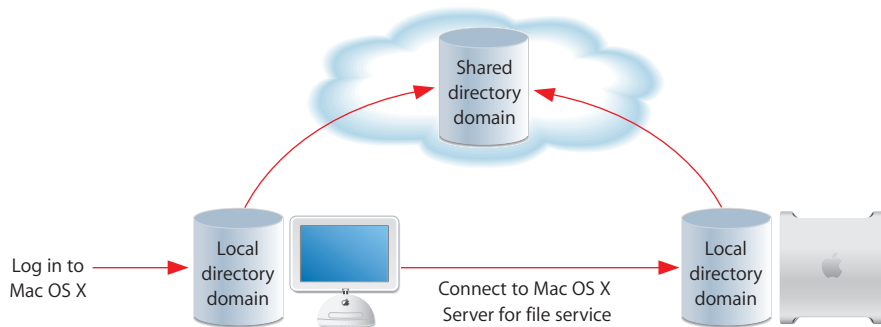


When you first set up a Mac OS X computer, its local directory domain is automatically created and populated with records. For example, a user record is created for the user who performed the installation. It contains the user name and password entered during setup, as well as other information, such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

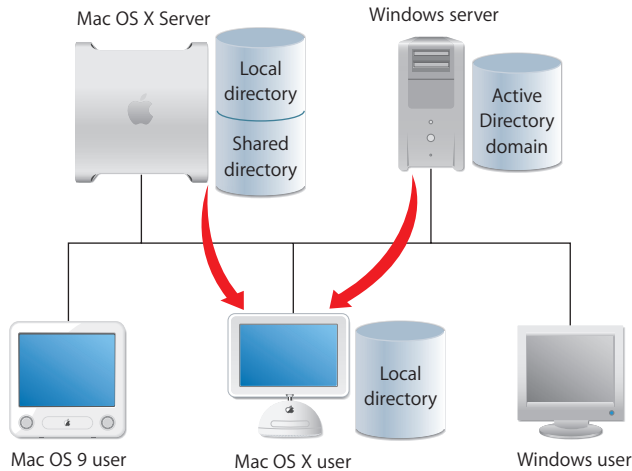
If Open Directory does not find a user's record in the local domain of a Mac OS X computer, Open Directory can search for the user's record in any shared domains to which the computer has access. In the following example, the user can access both computers because the shared domain accessible from both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

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

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.

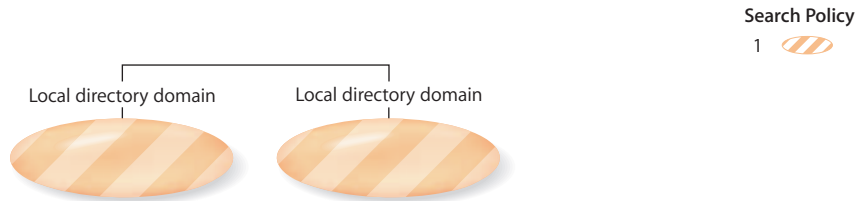
The search policy is also called the *search path*.

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.

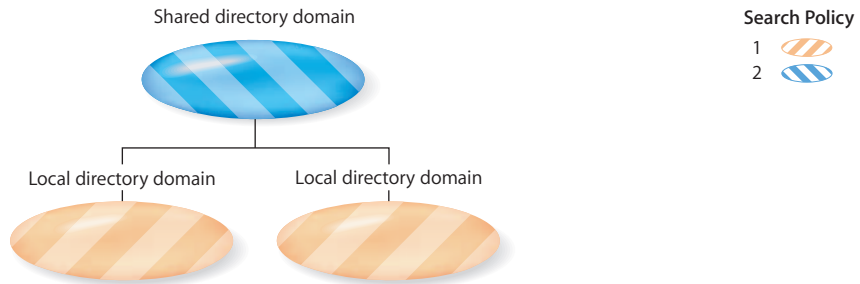
Local Directory Search Policy

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.

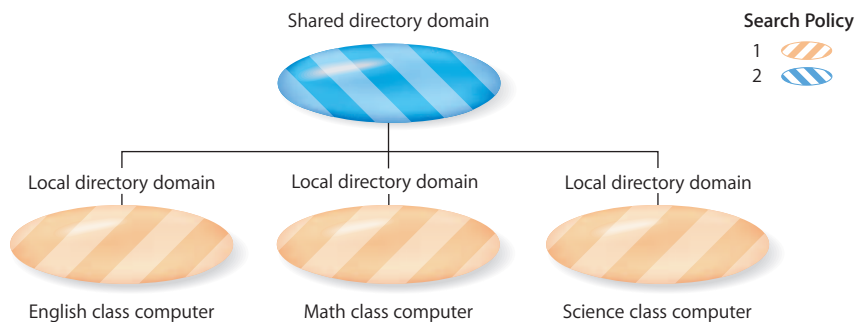


Two-Level Search Policies

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.

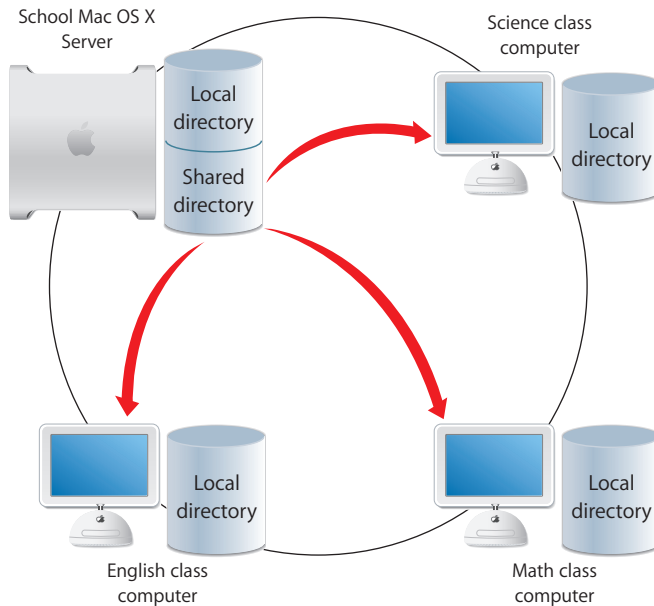


Here's a scenario in which a two-level search policy might be used:



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.

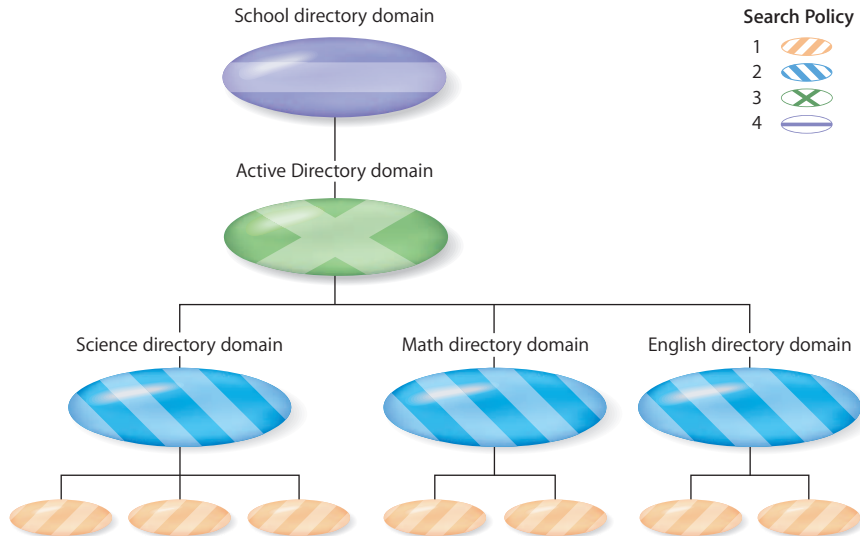
While local domains reside on their respective computers, a shared domain resides on a server accessible from the local domain's computer. When an instructor logs in to any of the three class computers and cannot be found in the local domain, Open Directory searches the shared domain. In this example,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

Mac OS X computers can be 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 determines whether the computer is configured to bind to shared NetInfo domains. The computer 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 153 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. This includes selecting the option to add DHCP-supplied LDAP directories. See "Setting Up Search Policies" on page 113, and "Enabling or Disabling Use of a DHCP-Supplied LDAP Directory" on page 118 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, 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.

Important: If you configure Mac OS X to use an automatic authentication search policy and a DHCP-supplied LDAP server or a DHCP-supplied NetInfo domain, you will increase the risk of an attacker gaining control of your computer. The risk is higher if your computer is configured to connect to a wireless network. See “Protecting Computers From a Malicious DHCP Server” on page 117 for more information.

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 searched before an Open Directory server's shared directory domain. This would allow users to log in using user records from the Active Directory domain and have their preferences managed by group and computer records from the Open Directory domain.

A custom search policy generally will not work in multiple network locations or while not connected to a network because it relies on the availability of specific directory domains on a particular network. If a portable computer is disconnected from its usual network, it will no longer have access to the shared directory domains on its custom search policy. The disconnected computer will still have access to its own local directory domain, since it is the first directory domain on every search policy. The portable computer user will be able to log in using a user record from the local directory domain, which may include mobile user accounts. These mirror user accounts from the shared directory domain that the portable computer accesses when it's connected to its usual network.

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.

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 traditional authentication methods that network services require.

Open Directory can authenticate users by:

- Using Kerberos authentication for single sign-on
- Using traditional authentication methods and a password stored securely in the Open Directory Password Server database
- Using traditional authentication methods and a shadow password stored in a secure shadow password file for each user
- 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 a password policy for all users as well as specific password policies for each user, such as automatic password expiration and minimum password length. (Password policies do not apply to administrators, crypt password authentication, or LDAP bind authentication.)

Password Types

Each user account has a password type that determines how the user account is authenticated. In a local directory domain, the standard password type is shadow password. On a server upgraded from Mac OS X Server version 10.3, user accounts in the local directory domain can also have an Open Directory password type.

For user accounts in the LDAP directory of Mac OS X Server, the standard password type is Open Directory. User accounts in the LDAP directory can also have a password type of crypt password.

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, and service access control lists (SACLs) authorize use of services. If Open Directory authenticates you, the SACL for login window determines whether you can log in, the SACL for AFP service determines whether you can connect for file service, and so on. Some services also determine whether a user is authorized to access particular resources. This authorization may require retrieving additional user account information from the directory domain. For example, AFP service needs the user ID and group membership information to determine which folders and files the user is authorized to read and/or write.

Open Directory Passwords

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. Kerberos is a network authentication system that uses credentials issued by a trusted server. Open Directory Password Server supports the traditional password authentication methods that some clients of network services require. (Kerberos isn't available on some Open Directory servers, such as an upgraded server with a shared NetInfo directory instead of an LDAP directory.)

Neither Kerberos nor Open Directory Password Server stores the password in the user's account. Both Kerberos and Open Directory Password Server store passwords in secure databases apart from 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 160.)

User accounts in the following directory domains can have Open Directory passwords:

- The LDAP directory of Mac OS X Server
- The local directory domain of Mac OS X Server upgraded from v10.2–10.3
- A shared NetInfo directory of a server upgraded from or still using Mac OS X Server v10.2

Note: Open Directory passwords can't be used to log in to Mac OS X version 10.1 or earlier. Users who need to log in using the login window of Mac OS X v10.1 or earlier must be configured to use crypt passwords. The password type doesn't matter for other services. For example, a user of Mac OS X v10.1 could authenticate for Apple file service with an Open Directory password.

Shadow Passwords

Shadow passwords support the same traditional authentication methods as Open Directory Password Server. These authentication methods are used to send shadow passwords over the network in a scrambled form, or hash.

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.

Shadow passwords also provide cached authentication for mobile user accounts. See the user management guide for complete information on mobile user accounts.

Crypt Passwords

A crypt password is stored an encrypted value, or hash, in the user account. 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 expect to find a crypt password stored in the user account.

Crypt authentication supports a maximum password length of only 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. Shadow passwords and Open Directory passwords are not subject to this length limit.

For secure transmission of passwords over a network, crypt supports the DHX authentication method.

Offline Attacks on Passwords

Because crypt passwords are stored directly in user accounts, they are potentially subject to cracking. User accounts in a shared directory domain are 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. Note that Open Directory passwords and shadow passwords aren't stored in user accounts, so these passwords can't be read from directory domains.

A malicious attacker, 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 decode crypt passwords stored in the user records. After decoding a crypt password, the cracker can log in unnoticed with a legitimate user name and crypt 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.

Important: Shadow passwords 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 change only Open Directory passwords and password policies.)

Crypt passwords are not considered secure. They should be used only for user accounts that must be compatible with UNIX clients that require them or Mac OS X v10.1 clients. Being stored in user accounts, they're too accessible and thus subject to offline attack (see "Offline Attacks on Passwords"). Although stored in an encoded form, they're relatively easy to decode.

How Crypt Passwords Are Encrypted

Crypt passwords are not stored in clear text; they are concealed and made unreadable by encryption. A crypt password is encrypted 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 or shadow file. If the values match, the password is considered valid.

Determining Which Authentication Option to Use

To authenticate a user, Open Directory must determine which authentication option to use—Kerberos, Open Directory Password Server, shadow password, or crypt password. 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.

You can change a user’s authentication authority attribute by changing the password type in the Advanced pane of Workgroup Manager, as shown in the following table. See “Changing a User’s Password Type” on page 96 for more information.

Password type	Authentication authority	Attribute in user record
Open Directory	Open Directory Password Server and/or Kerberos ¹	Either or both: <ul style="list-style-type: none">• ;ApplePasswordServer;• ;Kerberosv5;
Shadow password	Password file for each user, readable only by the root user	Either: <ul style="list-style-type: none">• ;ShadowHash;²• ;ShadowHash;<list of enabled authentication methods>
Crypt password	Encoded password in user record	Either: <ul style="list-style-type: none">• ;basic;• no attribute at all

¹ User accounts from Mac OS X Server v10.2 must be reset to include the Kerberos authentication authority attribute. See “Enabling Single Sign-On Kerberos Authentication for a User” on page 99.

² If the attribute in the user record is ;ShadowHash; without a list of enabled authentication methods, default authentication methods are enabled. The list of default authentication methods is different for Mac OS X Server and Mac OS X.

The authentication authority attribute can specify multiple authentication options. For example, a user account with an Open Directory password type normally has an authentication authority attribute that specifies both Kerberos and Open Directory Password Server.

A user account doesn't have to include 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.

Password Policies

Open Directory enforces password policies for users whose password type is Open Directory or Shadow Password. 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.

The password policy for a mobile user account applies when the account is used while disconnected from the network as well as while connected to the network. A mobile user account's password policy is cached for use while offline. For more information about mobile user accounts, see the user management guide.

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 could subject them to denial-of-service attacks.

Kerberos and Open Directory Password Server maintain password policies separately. An Open Directory server synchronizes the Kerberos password policy rules with Open Directory Password Server password policy rules.

Single Sign-On Authentication

Mac OS X Server uses Kerberos for *single sign-on* authentication, which relieves users from entering a name and password separately for every service. With single sign-on, 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 sign-on feature, users and services must be *Kerberized*—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 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 sign-on. This server's Kerberized services also use the server's built-in KDC and are automatically configured for single sign-on. This Mac OS X Server KDC can also authenticate users for services provided by other servers. Having additional servers with Mac OS X Server use the Mac OS X Server KDC requires only minimal configuration.

Kerberos Authentication

Kerberos was developed at MIT to provide secure authentication and communication over open networks like the Internet. It's named for the three-headed dog that guarded the entrance to the underworld of Greek mythology.

Kerberos provides proof of identity for two parties. It enables you to prove who you are to network services you want to use. It also proves to your applications that network services are genuine, not spoofed. Like other authentication systems, Kerberos does not provide authorization. Each network service determines for itself what it will allow you to do based on your proven identity.

Kerberos allows a client and a server to unambiguously identify each other much more securely than the typical challenge-response password authentication methods traditionally deployed. Kerberos also provides a single sign-on environment where users have to authenticate only once a day, week, or period of time, thereby easing authentication loads for the users.

Mac OS X Server offers integrated Kerberos support that virtually anyone can deploy. In fact, Kerberos deployment is so automatic that users and administrators may not realize it's deployed. Mac OS X v10.3 and later use Kerberos automatically when someone logs in using an account set for Open Directory authentication, and that is the default setting for user accounts in the Mac OS X Server LDAP directory. Other services provided by the LDAP directory server, such as AFP and mail service, also use Kerberos automatically. If your network has additional servers with Mac OS X Server v10.4, joining them to the Kerberos server is easy, and then most of their services use Kerberos automatically. Alternatively, if your network already has a Kerberos system such as Microsoft Active Directory, you can set up your Mac OS X Server and Mac OS X computers to use it for authentication.

Mac OS X Server and Mac OS X versions 10.3 and 10.4 support Kerberos version 5.

Breaking the Barriers to Kerberos Deployment

Until recently Kerberos was just a technology for universities and certain government sites. If Kerberos is so great then why isn't it more widely deployed? Answer: Adoption barriers needed to be taken down.

Mac OS X and Mac OS X Server v10.3 and later eliminate the following historical barriers to adoption of Kerberos.

- An Administrator had to setup a Kerberos Key Distribution Center (KDC). This was a nontrivial process to deploy and administer—not for the faint of heart.
- There was no standard integration with a directory system. Kerberos only does authentication, it doesn't store user account data such as user ID (UID), home directory location, or group membership. The administrator had to figure out how to integrate Kerberos with a directory system.
- All servers had to be registered with the Kerberos KDC. This added an extra step to the server setup process.
- After setting up a Kerberos server, the administrator had to visit all client machines and configure each one to use Kerberos. This wasn't difficult but was time-consuming and required editing configuration files and using command-line tools.
- You needed a suite of Kerberized applications (server and client software). Some of the basics are available but porting them and adapting them to work with your environment was difficult.
- Not all network protocols used for client-server authentication are Kerberos-enabled. Some network protocols still require traditional challenge-response authentication methods and there is no standard way to integrate Kerberos with these legacy network authentication methods.
- Kerberos client supports failover so if one KDC is offline it can use a replica, but the administrator had to figure out how to set up a Kerberos replica.
- Administration tools were never integrated. Tools for creating and editing user accounts in the directory domain didn't know anything about Kerberos, and the Kerberos tools knew nothing about user accounts in directories. Setting up a user record was a site-specific operation based on how the KDC was integrated with the directory system.

Single Sign-On Experience

Kerberos is a credential or ticket-based system. The user logs in once to the Kerberos system and is issued a ticket with a life span. During the life span of this ticket the user never need authenticate again to access a Kerberized service. The user's Kerberized client software, such as the Mac OS X Mail application, automatically presents a valid Kerberos ticket to authenticate the user for a Kerberized service. This provides a single sign-on experience.

A Kerberos ticket is like a press pass to a jazz festival held at multiple nightclubs over a three-day weekend. You prove your identity once to get the pass. Until the pass expires, you can show it at any nightclub to get a ticket for a performance. All the participating nightclubs accept your pass without seeing your proof of identity again.

Secure Authentication

The Internet is inherently insecure, yet many authentication protocols provide no real security. Malicious hackers can use readily available software tools to intercept passwords being sent over a network. Many applications send passwords unencrypted, and these are ready to use as soon as they're intercepted. Even encrypted passwords are not completely safe. Given enough time and computing power, encrypted passwords can be cracked.

A firewall can be used to isolate passwords on your private network, but this is no panacea. A firewall provides no security against disgruntled or malicious insiders.

Kerberos was designed to solve network security problems. It never transmits the user's password across the network, nor saves it in the user's computer memory or disk. Thus even if the Kerberos credentials are cracked or compromised, the attacker does not learn the original password and can potentially compromise only a small portion of the network rather than the whole network.

In addition to superior password management Kerberos is also mutually authenticated. The client authenticates to the service, and the service authenticates to the client. A man-in-the-middle or spoofing attack is impossible when you are using Kerberized services, and that means users can trust the services they are accessing.

Ready to Move Beyond Passwords

Network authentication is tricky business. In order to deploy a new network authentication method you have to have both the client and server agree on the method of authentication. And while it is possible for any client/server processes to agree on a custom authentication method, getting pervasive adoption across a suite of network protocols, platforms, and clients is virtually impossible.

For example, suppose you wanted to deploy smart cards as a network authentication method. Without Kerberos, you'd have to modify every client/server protocol to support the new method. The list of protocols includes SMTP, POP, IMAP, AFP, SMB, HTTP, FTP, IPP, SSH, QuickTime Streaming, DNS, LDAP, Netinfo, RPC, NFS, AFS, WebDAV, LPR, and goes on and on. Considering all the software that does network authentication, deploying a new authentication method across the entire suite of network protocols would be a daunting task. While this might be feasible for software from one vendor, you'd be unlikely to get all vendors to modify their client software to use your new smart card method. Furthermore, you'd probably also want your smart card authentication to work on multiple platforms: Mac OS X, Windows, and UNIX.

Due to the design of Kerberos, a client/server binary/protocol that supports Kerberos doesn't even know how the user proves identity—user name and password pair, smart card and PIN, you name it. Therefore you need only change the Kerberos client and the Kerberos server to accept a new proof of identity such as a smart card, and voilà your entire Kerberos network has now adopted the new proof-of-identity method, without deploying new versions of your client and server software.

Multiplatform Authentication

Kerberos is available on every major platform including Mac OS X, Windows, Linux, and other UNIX variants.

Centralized Authentication

Kerberos provides a central authentication authority for the network. All Kerberos-enabled services and clients on the network use this central authority. Administrators can centrally audit and control authentication policies and operations.

Kerberized Services

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

- Login window
- Mail service
- AFP file service
- FTP file service
- SMB/CIFS file service (as a member of an Active Directory Kerberos realm)
- VPN service
- Apache web service
- LDAP directory service

These services have been “Kerberized” whether running or not. Only services that have been Kerberized can use Kerberos to authenticate a user. Mac OS X Server includes command-line tools for Kerberizing additional services that are compatible with MIT-based Kerberos. For additional information, see the Open Directory chapter of the command-line administration guide.

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:

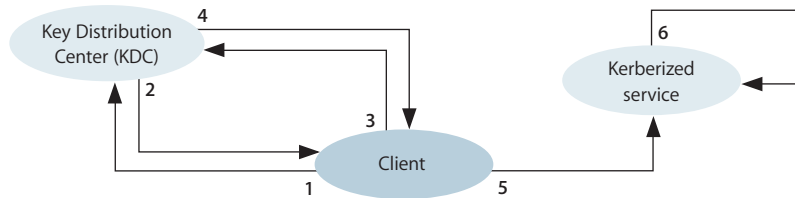
- Realm: MYREALM.EXAMPLE.COM

- User principal: jsanchez@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 Key Distribution Center (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 authenticates the client by verifying that the ticket is valid.

After authenticating the client, the service determines if the client is authorized to use the service. Kerberos only authenticates clients; it does not authorize them to use services. For example, many services use Mac OS X Server's service access control lists to determine whether a client is authorized to use the service.

Note that Kerberos never sends any password or password policy information to any service. 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

Open Directory Password Server and Shadow Password Authentication Methods

For compatibility with various services, Mac OS X Server can use a variety of authentication methods to validate Open Directory passwords and shadow passwords. For Open Directory passwords, Mac OS X Server uses the standard Simple Authentication and Security Layer (SASL) method to negotiate an authentication method between a client and a service. For shadow passwords, the use of SASL depends on the network protocol.

Authentication method	Network security	Storage security	Uses
APOP	Encrypted, with clear text fallback	Clear text	POP mail service
CRAM-MD5	Encrypted, with clear text fallback	Encrypted	IMAP mail service, LDAP service
DHX	Encrypted	Encrypted	AFP file service, Open Directory administration
Digest-MD5	Encrypted	Encrypted	Login window, email service
MS-CHAPv2	Encrypted	Encrypted	VPN service
NTLMv1 and NTLMv2	Encrypted	Encrypted	SMB/CIFS services (Windows NT/98 or later)
LAN Manager	Encrypted	Encrypted	SMB/CIFS services (Windows 95)
WebDAV-Digest	Encrypted	Clear text	WebDAV file service (iDisk)

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 stronger algorithms to encode the information that they transmit between client and server. The more secure authentication methods also store encrypted passwords, called hashes, which can't easily be recovered from the server. Less secure methods store a recoverable, clear text password.

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.

Note: If you connect Mac OS X Server v10.4 or later to a directory domain of Mac OS X Server v10.3 or earlier, be aware that users defined in the older directory domain cannot be authenticated with the NTLMv2 method. This method may be required to securely authenticate some Windows users for the Windows services of Mac OS X Server v10.4 and later. Open Directory Password Server in Mac OS X Server v10.4 and later supports NTLMv2 authentication, but Password Server in Mac OS X Server v10.3 and earlier does not support NTLMv2.

Similarly, if you connect Mac OS X Server v10.3 or later to a directory domain of Mac OS X Server v10.2 or earlier, 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 v10.3 and later. Open Directory Password Server in Mac OS X Server v10.3 and later supports MS-CHAPv2 authentication, but Password Server in Mac OS X Server v10.2 does not support MS-CHAPv2.

Disabling Open Directory Authentication Methods

Authentication methods can be selectively disabled to make Open Directory password storage on the server more secure. For example, if no clients will use Windows services, you can disable the NTLMv1, NTLMv2, and LAN Manager authentication methods to prevent storing passwords on the server using these methods. Then someone who somehow gains access to the server's password database can't exploit weaknesses in these authentication methods to crack passwords.

Important: If you disable an authentication method, its hash will be removed from the password database the next time the user authenticates. If you enable an authentication method that was disabled, every Open Directory password must be reset to add the newly enabled method's hash to the password database. Users can reset their own passwords, or a directory administrator can do it.

Disabling an authentication method makes the Open Directory Password Server database more secure in the event a malicious user gains physical access to an Open Directory server (master or replica) or to media containing a backup of the Open Directory master. Someone who gains access to the password database can try to crack a user's password by attacking the hash or recoverable text stored in the password database by any authentication method. Nothing will be stored in the password database by a disabled authentication method, leaving one less avenue of attack open to a cracker who has physical access to the Open Directory server or a backup of it.

The hashes stored in the password database by some authentication methods are easier to crack than others. The recoverable authentication methods actually store clear (plainly readable) text. Disabling authentication methods that store clear text or weaker hashes will increase password database security more than disabling methods that store stronger hashes.

If you believe your Open Directory master, replicas, and backups are secure, then all authentication methods should be selected. If you're concerned about the physical security of any Open Directory server or its backup media, you should disable some methods.

Note: Disabling authentication methods does not increase the security of passwords while they are transmitted over the network. Only the password database security is affected. In fact, disabling some authentication methods may require clients to configure their software to send passwords over the network in clear text, thereby compromising password security in a different way.

Disabling Shadow Password Authentication Methods

Authentication methods can be selectively disabled to make passwords stored in shadow password files more secure. For example, if a user doesn't use mail service or web services, you can disable the WebDAV-Digest and APOP methods for the user. Then someone who somehow gains access to the shadow password files on a server can't recover the user's password.

Important: If you disable a shadow password authentication method, its hash will be removed from a user's shadow password file the next time the user authenticates. If you enable an authentication method that was disabled, the newly enabled method's hash will be added to the user's shadow password file the next time the user authenticates for a service that can use a clear text password, such as login window or AFP. Alternatively, the user's password can be reset to add the newly enabled method's hash. The user can reset the password, or a directory administrator can do it.

Disabling an authentication method makes the shadow password more secure in the event a malicious user gains physical access to a server's shadow password files or to media containing a backup of the shadow password files. Someone who gains access to the password files can try to crack a user's password by attacking the hash or recoverable text stored by any authentication method. Nothing will be stored by a disabled authentication method, leaving one less avenue of attack open to a cracker who has physical access to a server's shadow password files or a backup of them.

The hashes stored by some authentication methods are easier to crack than others. With the recoverable authentication methods, the original clear text password can be reconstructed from what is stored in the file. Disabling the authentication methods that store recoverable or weaker hashes will increase shadow password file security more than disabling methods that store stronger hashes.

If you believe a server's shadow password files and backups are secure, then all authentication methods should be selected. If you're concerned about the physical security of the server or its backup media, you should disable some methods.

Note: Disabling authentication methods does not increase the security of passwords while they are transmitted over the network; only the password storage security is affected. In fact, disabling some authentication methods may require clients to configure their software to send passwords over the network in clear text, thereby compromising password security in a different way.

Contents of Open Directory Password Server Database

Open Directory Password Server maintains an authentication database separate from the directory domain. Open Directory tightly restricts access to the authentication database.

Open Directory Password Server stores the following information in its authentication database for each user account that has a password type of Open Directory:

- The user's password ID, 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, stored in recoverable (clear text) or hashed (encrypted) forms. 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.
- Time stamps and other usage information, such as last login time, last failed validation time, count of failed validations, and replication information.

LDAP Bind Authentication

For user accounts that reside in an LDAP directory on a non-Apple server, Open Directory attempts to use 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.

LDAP bind authentication may be insecure if the LDAP directory service and the client computer's connection to it are configured to allow sending clear text passwords over the network. Open Directory tries to use a secure authentication method with the LDAP directory. If the directory doesn't support secure LDAP bind and the client's LDAPv3 connection allows sending a clear text password, Open Directory will fall back to simple LDAP bind. In this case, 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. See "Changing the Security Policy for an LDAP Connection" on page 128 and "Changing the Connection Settings for an LDAP Directory" on page 127 for more information.

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 works only with user accounts that were created in a NetInfo domain of Mac OS X Server v10.0–10.2. Authentication Manager must have been enabled for the NetInfo domain.

When you upgrade a server to Mac OS X Server v10.4 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. Each existing user account in the server's local directory domain, which is a NetInfo domain, is automatically converted from crypt password to shadow password when the user or administrator changes the password or the user authenticates for a service that can use a recoverable authentication method.

After upgrading a server to Mac OS X Server v10.4, you can change existing user accounts to authenticate using Open Directory. If the upgraded server has a shared NetInfo domain and you migrate it to an LDAP directory, all user accounts are automatically converted to Open Directory passwords.

Open Directory passwords and shadow passwords are more secure than crypt passwords. Both Open Directory passwords and shadow passwords can be used for Windows file service. Open Directory passwords are required for domain login from a Windows workstation to a Mac OS X Server primary domain controller. New user accounts created in the LDAP directory of Mac OS X Server v10.4 are set to use Open Directory authentication.

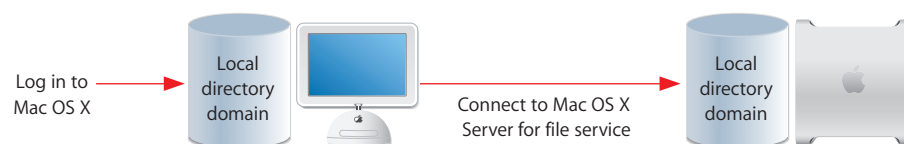
Like the plumbing and wiring in a building, directory services for a network must be planned in advance, not on an ad hoc basis.

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.

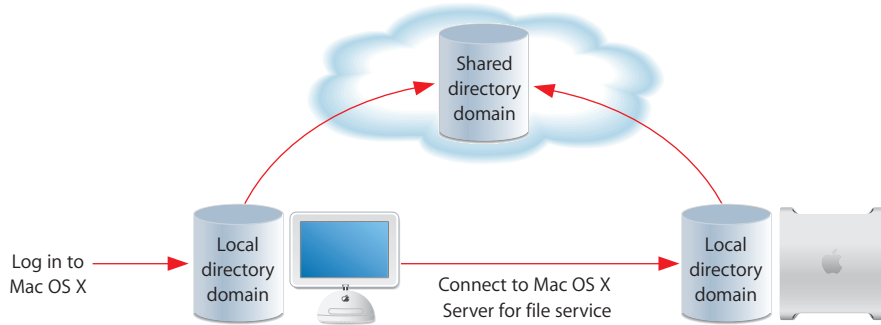
This chapter presents guidelines for planning Open Directory services and describes tools for managing them.

General Planning Guidelines

If you do not need to share user and resource information among multiple Mac OS X computers, very little directory domain planning is necessary; everything can be accessed from local directory domains. Just ensure that all individuals who need to use a particular Mac OS X computer have user accounts on that computer. These user accounts reside in the local directory domain on the computer. In addition, everyone who needs to use Mac OS X Server's file service, mail service, or other services that require authentication will need a user account in the server's local directory domain. With this arrangement, each user has two accounts, one for logging in to a computer and one for accessing services of Mac OS X Server.

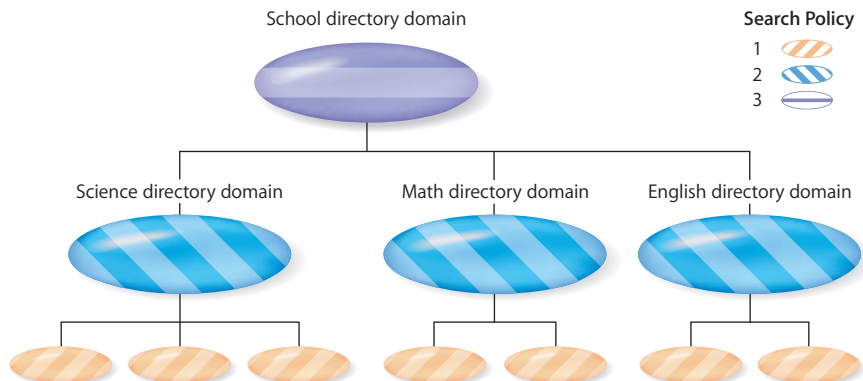


If you want to share information among Mac OS X computers and servers, you need to set up at least one shared directory domain. With this arrangement, each user needs only an account in the shared directory domain. With this one account, the user can log in to Mac OS X on any computer that's configured to access the shared directory domain. The user can also use this same account to access services of any Mac OS X Server that's configured to access the shared directory domain.



In many organizations, a single shared directory domain is completely adequate. It can allow hundreds of thousands of users and thousands of computers sharing the same resources, such as printer queues, share points for home directories, share points for applications, and share points for documents. Replicating the shared directory domain can increase the capacity or performance of the directory system by allowing multiple servers to handle the directory system load for the network.

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



Controlling Data Accessibility

With a single shared directory domain, you can manage client preferences and network views to isolate certain network services while making other services available to all users. For example, you might set up a managed network view that makes share points containing accounting applications and files visible only to accounting department computers. You might set up a different managed network view to show only technical writers the share points that contain publishing software and document files. If you want all employees to have access to one another's drop box folders, you would include the share point containing the drop box folders in all managed network views. See the user management guide for more information on managed clients and managed network views.

If your network has several shared directory domains, you can make each directory's information available only to a subset of the network computers. For example, students with user accounts in the science directory domain could log in only on computers whose search policies include the science domain.

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 won'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 uses the Berkeley DB database, which will remain efficient with 200,000 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, a server dedicated to Open Directory has a limit of 1000 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. An Open Directory server may be able to support well over 1000 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.

Avoid having only one replica on either side of a slow network link. If a replica is separated from all other replicas by a slow network link and the one replica fails, clients of the replica will fail over to a replica on the other side of the slow network link. Their directory services may slow down markedly.

If your network has a mix of Mac OS X Server versions 10.3 and 10.4, one version can't be a replica of a master of the other version. An Open Directory master of v10.4 won't replicate to Mac OS X Server v10.3. Nor will an Open Directory master of Mac OS X Server v10.3 replicate to Mac OS X Server v10.4.

	Mac OS X Server v10.4 master	Mac OS X Server v10.3 master
Mac OS X Server v10.4 replica	Yes	No
Mac OS X Server v10.3 replica	No	Yes

Load Balancing in Small, Medium, and Large Environments

Do not use service load-balancing software from third parties with Open Directory servers. Load-balancing software could cause unpredictable problems for Open Directory clients. It could interfere with the automatic load balancing and failover behavior of Open Directory in Mac OS X and Mac OS X Server. Mac OS X clients automatically seek the nearest available Open Directory server—master or replica. A client's nearest Open Directory master or replica is the one that responds most quickly to the client's request for an Open Directory connection.

Replication in a Multibuilding 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.

Using an Open Directory Master or Replica With NAT

If your network has an Open Directory server on the private network side of a NAT router (or gateway), including the NAT router of Mac OS X Server, only computers on the private network side of the NAT router can connect to the Open Directory server's LDAP directory domain. Computers on the public network side of the NAT router can't connect to the LDAP directory domain of an Open Directory master or replica that's on the private network side.

If an Open Directory server is on the public network side of a NAT router, computers on both the private network and the public network sides of the NAT router can connect to the Open Directory server's LDAP directory.

Avoiding Kerberos Conflicts With Multiple Directories

If you set up an Open Directory master on a network that already has an Active Directory domain, your network will have two Kerberos realms. It will have an Open Directory Kerberos realm and an Active Directory Kerberos realm. For all practical purposes, other servers on the network can use only one of the Kerberos realms. When you set up a file server, mail server, or other server that can use Kerberos authentication, you have to choose one Kerberos realm.

Mac OS X Server must belong to the same Kerberos realm as its client users. The Kerberos realm has just one authoritative Kerberos server. This one Kerberos server takes all responsibility for Kerberos authentication within the realm. The Kerberos server can only authenticate clients and servers in its realm. The Kerberos server can't authenticate clients or services that are part of a different realm.

Only user accounts in the chosen Kerberos realm will have single sign-on experience. Any user accounts in the other realm will still be able to authenticate, but they won't have single sign-on.

If you're configuring a server to access multiple directory systems each with a Kerberos realm, you have to think very carefully about the user accounts that will use Kerberized services. You have to know the intent of having access to two directory services. You need to join the server to the realm whose companion directory domain contains the user accounts that you want to use Kerberos and have single sign-on.

For example, you might want to configure access to an Active Directory realm for its user records and an Open Directory LDAP directory for the Mac OS X records and attributes that aren't in Active Directory, such as group and computer records. Other servers could join either the Active Directory Kerberos realm or the Open Directory Kerberos realm. In this case, the other servers should join the Active Directory Kerberos realm so the Active Directory user accounts have single sign-on. If you also have user accounts in the Open Directory server's LDAP directory, users can still authenticate with them. But the Open Directory user accounts won't use Kerberos or have single sign-on; they'll use Open Directory Password Server authentication methods. You could put all Mac users in the Open Directory domain and all Windows users in the Active Directory domain, and they'd all be able to authenticate. But only one of the populations would be able to use Kerberos.

Important: A serious problem occurs if you configure an Open Directory master or replica to also access an Active Directory domain. In this case, both the Open Directory Kerberos realm and the Active Directory Kerberos realm try to use the same configuration files on the Open Directory server, probably disrupting Open Directory Kerberos authentication. Therefore, you should not configure an Open Directory master or replica to also access an Active Directory domain or any other directory domain with a Kerberos realm.

To avoid a Kerberos configuration file conflict, don't use an Open Directory server as a workstation for managing users in another Kerberos server's directory domain, such as an Active Directory domain. Instead, use an administrator computer (a Mac OS X computer with server administration tools installed) that's configured to access all the directory domains of interest. If you must use an Open Directory server to manage users in another server's directory domain, make sure the other directory domain is not part of the Open Directory server's authentication search policy.

To further avoid a Kerberos configuration file conflict, don't use an Open Directory server to provide services that need to access a different Kerberos server's directory domain. For example, if you need to configure AFP file service to access both Open Directory and Active Directory, don't use an Open Directory server to provide the file service. Use another server and join it to the Kerberos realm of one directory service or the other.

Theoretically, servers or clients can belong to two Kerberos realms, such as an Open Directory realm and an Active Directory realm. Multiple-realm Kerberos authentication requires a very advanced configuration, which includes setting up the Kerberos servers and clients for cross-realm authentication, revising Kerberized service software so it can belong to multiple realms, and more. Documenting these advanced procedures is beyond the scope of this guide.

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 and the Open Directory logs to another disk.

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, 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.
- Set up service access control lists (SACLs) for login window and SSH to limit who can log in to an Open Directory master or replica.

- 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 those 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.
 - While creating an Open Directory replica, port 22 must be open between the master and prospective replica. This is the port used for secure shell (SSH) data transfer, which is used to transfer a complete, up-to-date copy of the LDAP database. After initial setup of the replica, only the LDAP port (389 or 636) is used for replication.
 - Workgroup Manager uses TCP port 311 and 625.
 - Server Admin uses TCP port 311.
 - SMB/CIFS 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.

Replicating directory and authentication data over the network is a minimal security risk. 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. NetInfo Manager is also available for legacy NetInfo domains (or you can use the Inspector in Workgroup Manager).

All these applications are included with Mac OS X Server and can be installed on another computer with Mac OS X v10.4 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

The Server Admin application provides access to tools you use to set up, manage, and monitor Open Directory services and other services. 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."

See the chapter on server administration in the getting started guide for basic information about using Server Admin, including:

- Opening and authenticating in Server Admin
- Working with specific servers
- Administering services
- Controlling access to services
- Using SSL for remote server administration
- Customizing the Server Admin environment

Server Admin is installed in `/Applications/Server/`.

Directory Access

Directory Access determines how a Mac OS X computer uses directory services, discovers network services, and searches directory services for authentication and contacts information. You use Directory Access to:

- Configure access to LDAP directories, an Active Directory domain, an NIS domain, and NetInfo domains.
- Configure data mapping for LDAP directories.
- Define policies for searching multiple directory services for authentication and contact information.
- Enable or disable kinds of directory services and kinds of network service discovery.

Directory Access can connect to other servers 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

The Workgroup Manager application provides comprehensive management of clients of Mac OS X Server. You use Workgroup Manager to:

- Set up and manage user accounts, group accounts, and computer lists. For instructions on managing user authentication, see Chapter 6, “Managing User Authentication.” For instructions on other user, group, and computer management topics, see the user management guide and the Windows services administration guide.
- Manage share points for file service and user home directories. For instructions, see the chapter on share points in the file services administration guide, the chapter on home directories in the user management guide, and the chapter on managing Windows services in the Windows services administration guide.
- Control what Mac OS X users see when they select the Network globe in a Finder sidebar. For instructions, see the chapter on managing network views in the user management guide.
- View directory entries in raw form by using the Inspector. For instructions, see “Directly Viewing and Editing Directory Data” on page 161.

See the chapter on server administration in the getting started guide for basic information about using Workgroup Manager including:

- Opening and authenticating in Workgroup Manager
- Administering accounts
- Customizing the Workgroup Manager environment

Workgroup Manager is installed in `/Applications/Server/`.

Command-Line Tools

A full range of command-line tools is 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 using the Terminal application, located in `/Applications/Utilities/`. For instructions, see the command-line administration guide.

NetInfo Manager

You use NetInfo Manager 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 sign-on Kerberos authentication service, 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 72 for a list of items to think about before you configure Open Directory on Mac OS X Server.

Step 2: Set up standalone servers

If you want to set up servers that won’t get authentication and other administrative information from a directory service, see “Setting Up a Standalone Server” on page 73.

Step 3: Set up an Open Directory master

If you want to set up a server to provide directory and authentication services, see “Open Directory Master and Replica Compatibility” on page 74 and “Setting Up an Open Directory Master” on page 75.

Step 4: Set up replicas of your Open Directory master

If you want to set up one or more servers to provide failover directory and authentication services or remote directory and authentication services for fast client interaction on distributed networks, see “Setting Up an Open Directory Replica” on page 77.

Step 5: Set up servers that connect to other directory systems

If you have file servers or other servers that need to access directory and authentication services, see “Setting Up a Connection to a Directory System” on page 80.

Step 6: Set up single sign-on Kerberos authentication

If you have set up an Open Directory master, you can configure other servers to join its Kerberos realm. If you set up an Open Directory master without Kerberos, you can set up Kerberos later. For instructions, see “Setting Up Single Sign-On Kerberos Authentication” on page 81.

Step 7: Migrate upgraded servers from NetInfo to LDAP

If you have servers that were upgraded from Mac OS X Server version 10.2 and are still using shared NetInfo directory domains, you can migrate them to LDAP. See “Migrating a Directory Domain From Netinfo to LDAP” on page 90 and “Disabling NetInfo After Migrating to LDAP” on page 92.

Step 8: Set up Directory Access on client computers

If you have set up an Open Directory master, you need to configure client computers to access its directory domain. You can also configure client computers to access other directory services such as Microsoft Active Directory. See Chapter 7, “Managing Directory Access.”

Step 9: Instruct users how to log in

See “Instructing Users How to Log In” on page 76.

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 With Open Directory.”
- Assess whether you need more than one shared domain. If so, decide which users will be defined in each shared domain. For more information, see “Multilevel Search Policies” on page 35 and “Simplifying Changes to Data in Directories” on page 59.
- Determine which authentication options users need. For descriptions of the available options, see Chapter 3, “Open Directory Authentication.”

- Decide whether to have replicas of your Open Directory master. Chapter 4, “Open Directory Planning,” provides some guidelines.
- Pick server administrators very carefully. Give administrator passwords only to people you trust. 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 v10.4 or later and use it to manage Open Directory on any server on your local network and beyond. 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 in the shared directory domain will be deleted.
- 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 archive a copy of the Open Directory master's directory and authentication data before changing it to an Open Directory standalone server. For instructions, see “Archiving an Open Directory Master” on page 167. You can also export users, groups, and computer lists from the Open Directory master before changing it to a standalone server. See the user management guide for more information.

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.

Open Directory Master and Replica Compatibility

Both Open Directory master and its replicas must use the same version of Mac OS X Server.

- An Open Directory master using Mac OS X Server v10.4 won't replicate to Mac OS X Server v10.3.
- Mac OS X Server v10.4 can't be a replica of an Open Directory master using Mac OS X Server v10.3.
- An Open Directory master using Mac OS X Server v10.4 can replicate to an Open Directory replica using Mac OS X Server v10.4.

If you have an Open Directory master and replicas that use Mac OS X Server v10.3, you need to upgrade them to v10.4 together. First you upgrade the master, and then you upgrade the replicas. Clients of the master and replicas will continue to receive directory and authentication services during the upgrade process. While you are upgrading the master, its clients will automatically fail over to the nearest replica. When you upgrade the replicas one by one, clients will fail back to the upgraded master.

Upgrading an Open Directory master from Mac OS X Server v10.3 to v10.4 will sever ties to its existing replicas. After upgrading each Open Directory replica to Mac OS X Server v10.4, it will be a standalone server and you'll need to make it a replica again. See the upgrading and migrating guide for instructions on upgrading to Mac OS X Server v10.4.

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.

An Open Directory master has an Open Directory Password Server, which supports all the conventional authentication methods required by Mac OS X Server services. In addition, an Open Directory master can provide Kerberos authentication for single sign-on.

If you want the Open Directory master to provide Kerberos authentication for single sign-on, DNS must be available on the network and must be correctly configured to resolve the fully qualified DNS name of the Open Directory master server to its IP address. DNS must also be configured to resolve the IP address to the server's fully qualified DNS name.

Important: If you're changing an Open Directory replica to an Open Directory master, the procedure you follow depends on whether the replica will replace the master or become an additional master.

- If you want to promote a replica to replace a nonfunctional master, follow the instructions in “Promoting an Open Directory Replica” on page 165 instead of the instructions here.
- If you want to change a replica to an additional master, first decommission the replica as described in “Decommissioning an Open Directory Replica” on page 166. Then make it a master by following the steps in this topic.

Note: If Mac OS X Server was connected to a directory system and you make it an Open Directory master, it 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 be an Open Directory master:

- 1 Open Server Admin, connect to the server, and select Open Directory for this server in the Computers & Services list.
- 2 Click Settings (near the bottom of the window), then click General (near the top).
- 3 If the Role option is set to Open Directory Replica and you want to make a new Open Directory master, you must change Role to Standalone Server and click Save.

If you don't change an Open Directory replica to a standalone server before making it a master, you end up promoting the replica to be a master instead of making a new master. See “Promoting an Open Directory Replica” on page 165 for more information.

- 4 Choose Open Directory Master from the Role pop-up menu and enter the requested information.

Name, Short Name, User ID, Password: You must create a new user account for the primary administrator of the LDAP directory. This account is not a copy of the administrator account in the server's local directory domain. You should make the names and User ID of the LDAP directory administrator different from the names and User IDs of user accounts in the local directory domain.

Kerberos Realm: This field is preset to be the same as the server's DNS name converted to capital letters. This is the convention for naming a Kerberos realm. You can enter a different name if necessary.

Search Base: This field is preset to a search base suffix for the new LDAP directory, derived from the domain portion of the server's DNS name. You can enter a different search base suffix or leave it blank. If you leave this field blank, the LDAP directory's default search base suffix will be used.

- 5 Click OK, then click Save.

You can confirm that the Open Directory master is functioning properly by clicking Overview (near the bottom of the Server Admin window, with Open Directory selected in the Computers & Services list). The status of all the items listed in the Open Directory overview pane should be "Running." If Kerberos remains stopped and you want it running, see "Kerberos is Stopped on an Open Directory Master or Replica" on page 169.

After setting up a Mac OS X Server computer to be an Open Directory master, you can change its binding policy, security policy, password policy, replication frequency, and LDAP protocol options. For instructions, see "Setting Options for an Open Directory Master or Replica" on page 86.

You can configure other computers with Mac OS X or Mac OS X Server to access the server's shared LDAP directory domain. For instructions, see "Accessing LDAP Directories" on page 117.

Instructing Users How to Log In

When a Mac OS X computer is connected to a directory domain and is configured to display a list of users in the Mac OS X login window, the list may include "Other." You might need to tell users who have never logged in with a network account that they need to click Other and then enter the account name and password.

Users can configure their computers not to display a list of users in the login window. Users change this setting by clicking Login Options in the Accounts pane of System Preferences.

You can have a computer's login window show network users in its list, or not display a list at all by managing computer preferences. Use Workgroup Manager to configure login preference settings for the computer list account that includes the computer. To manage computers that are not part of a particular computer list account, configure login preference settings for the Guest Computers account. For instructions, see the user management guide.

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 Open Directory Password Server 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 Mac OS X and Mac OS X Server v10.3–10.4.

Note: If your network has a mix of Mac OS X Server versions 10.3 and 10.4, one version can't be a replica of a master of the other version. An Open Directory master of v10.4 won't replicate to Mac OS X Server v10.3. Nor will an Open Directory master of Mac OS X Server v10.3 replicate to Mac OS X Server v10.4.

Important: When you first 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 Make sure the master, the prospective replica, and every firewall between them is configured to allow SSH communications (port 22).

You can enable SSH for Mac OS X Server in Server Admin. Select the server in the Computers & Services list, click Settings, click General, then select the SSH option. For additional information on SSH, see the getting started guide.

For instructions on allowing SSH communications through the Mac OS X Server firewall, see the network services administration guide.

- 2 Open Server Admin and select Open Directory for a server in the Computers & Services list.
- 3 Click Settings (near the bottom of the window), then click General (near the top).
- 4 Choose Open Directory Replica from the Role pop-up menu and enter the requested information.

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

“Root password on Open Directory master:” Enter the password of the Open Directory master system’s root user (user name System Administrator).

“Domain administrator’s short name on master:” Enter the name of an LDAP directory domain administrator account.

“Domain administrator’s password on master:” Enter the password of the administrator account whose name you entered.

- 5 Click OK, then click Save.
- 6 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 v10.3–10.4 of Mac OS X or Mac OS X Server maintain a list of Open Director replicas. 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. 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 117 and “Defining Automatic Search Policies” on page 114.

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 “Scheduling Replication of an Open Directory Master” on page 164.

Creating Multiple Replicas of an Open Directory Master

If you want to make more than one server a replica of an Open Directory master, create the replicas one at a time. If you try to create two replicas simultaneously, one attempt will succeed and the other will fail. A subsequent attempt to establish the second replica should succeed.

Setting Up Open Directory Failover

If an Open Directory master or any of its replicas become unavailable, its client computers with version 10.3–10.4 of Mac OS X or Mac OS X Server 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 the Open Directory master fails permanently and you have a current archive of its data, you can restore the data to a new master. Alternatively, you can promote a replica to be the master. For instructions, see “Restoring an Open Directory Master” on page 168 and “Promoting an Open Directory Replica” on page 165.

Note: If a failed Open Directory master or replica had client computers with Mac OS X or Mac OS X Server v10.2 or earlier, the v10.2 computers and servers will not automatically fail over to another replica. If you replace a failed master by promoting a replica to be the master, you can manually reconfigure each v10.2 computer and server to connect to this new master or one of its replicas. You do this by using Directory Access on each v10.2 computer or server to create an LDAPv3 configuration that specifies how the computer accesses the new master or an available replica. For instructions, see “Accessing LDAP Directories” on page 117.

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 local 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 information in the shared directory domain will be deleted.
- 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 v10.4 or later to a directory domain of Mac OS X Server v10.3 or earlier, be aware that users defined in the older directory domain cannot be authenticated with the NTLMv2 method. This method may be required to securely authenticate some Windows users for the Windows services of Mac OS X Server v10.4 and later. Open Directory Password Server in Mac OS X Server v10.4 and later supports NTLMv2 authentication, but Password Server in Mac OS X Server v10.3 and earlier does not support NTLMv2.

Similarly, if you configure Mac OS X Server v10.4 or later to access a directory domain of Mac OS X Server version 10.2 or earlier, 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 v10.4 and later. Open Directory in Mac OS X Server v10.4 supports MS-CHAPv2 authentication, but Password Server in Mac OS X Server version 10.2 does not support MS-CHAPv2.

- 6 If the server you're configuring has access a directory system that also hosts a Kerberos realm, you can join the server to the Kerberos realm.

To join the Kerberos realm, you need the name and password of a Kerberos administrator or a user who has been delegated the authority to join the realm. For instructions, see "Joining a Server to a Kerberos Realm" on page 85.

Setting Up Single Sign-On Kerberos Authentication

Setting up single sign-on Kerberos authentication involves these tasks:

- DNS must be available on the network and must be correctly configured to resolve the fully qualified DNS name of the Open Directory master server (or other Kerberos server) to its IP address. DNS must also be configured to resolve the IP address to the server's fully qualified DNS name.
- An administrator sets up a directory system to host a Kerberos realm. For instructions on setting up Mac OS X Server to host a Kerberos realm, see "Setting Up an Open Directory Kerberos Realm" (next).
- A Kerberos administrator of an Open Directory master can delegate the authority to join servers to the Open Directory master's Kerberos realm. (The administrator does not need delegated authority. A Kerberos administrator has implicit authority to join any server to the Kerberos realm.) See "Delegating Authority to Join an Open Directory Kerberos Realm" on page 83.
- A Kerberos administrator or users with delegated authority join servers to the Kerberos realm, which then provides single sign-on Kerberos authentication for services provided by the servers that have joined. See "Joining a Server to a Kerberos Realm" on page 85.
- All computers using 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.

The individual services of Mac OS X Server do not require any configuration for single sign-on or Kerberos. The following services are ready for single sign-on Kerberos authentication on every server with Mac OS X Server v10.4 and later that has joined or is an Open Directory master or replica: Login window, mail service, AFP, FTP, SMB/CIFS (as a member of an Active Directory Kerberos realm), VPN, Apache web service, and LDAPv3 directory service (on an Open Directory master or replica).

Setting Up an Open Directory Kerberos Realm

You can provide single sign-on 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. But 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. For instructions, see “Setting Up an Open Directory Master” on page 75 and “Starting Kerberos After Setting Up an Open Directory Master” on page 82.

A server that is an Open Directory master requires no additional configuration to support single sign-on Kerberos authentication for all the Kerberized services that the server itself provides. This server can also support single sign-on Kerberos authentication for Kerberized services of other servers on the network. The other servers must be set up to join the Open Directory Kerberos realm. For instructions, see “Delegating Authority to Join an Open Directory Kerberos Realm” on page 83, and “Joining a Server to a Kerberos Realm” on page 85.

Important: An Open Directory master requires properly configured DNS so it can provide Kerberos and single sign-on authentication.

- DNS service must be configured to resolve the fully qualified DNS names of all servers including the Open Directory master itself to their IP addresses and to provide the corresponding reverse lookups. For instructions on setting up DNS service, see the network services administration guide.
- The Open Directory master server’s Network preferences must be configured to use the DNS server that resolves the server’s name. (If the Open Directory master server provides its own DNS service, its Network preferences must be configured to use itself as a DNS server.)

Starting Kerberos After Setting Up an Open Directory Master

If Kerberos doesn’t start automatically when you set up an Open Directory master, you can use Server Admin to start it manually. First you have to fix the problem that prevented Kerberos from starting. Usually the problem is DNS service that isn’t configured correctly or isn’t running at all.

Note: After you manually start Kerberos, users whose accounts have Open Directory passwords and were created in the Open Directory master’s LDAP directory while Kerberos was stopped may have to reset their passwords the next time they log in. A user account is thus affected only if all the recoverable authentication methods for Open Directory passwords were disabled while Kerberos was stopped.

To start Kerberos manually on an Open Directory master:

- 1 Open Server Admin, connect to the Open Directory master, and select Open Directory for it in the Computers & Services list.

- 2 Click Refresh (or choose View > Refresh) and check the status of Kerberos as reported in the Overview pane.

If Kerberos is running, there's nothing more to do.

- 3 Use Network Utility (in /Applications/Utilities/) to do a DNS lookup of the Open Directory master's DNS name and a reverse lookup of the IP address.

If the server's DNS name or IP address don't resolve correctly:

- In the Network pane of System Preferences, look at the TCP/IP settings for the server's primary network interface (usually built-in Ethernet). Make sure the first DNS server listed is the one that resolves the Open Directory server's name.
- Check the configuration of DNS service and make sure it's running.

- 4 In Server Admin, select Open Directory for the master server, click Settings, then click General.

- 5 Click Kerberize, then enter the information requested.

Administrator Name and Password: You must authenticate as an administrator of the Open Directory master's LDAP directory.

Realm Name: This field is preset to be the same as the server's DNS name converted to capital letters. This is the convention for naming a Kerberos realm. You can enter a different name if necessary.

Delegating Authority to Join an Open Directory Kerberos Realm

Using Server Admin, you can delegate the authority to join a server to an Open Directory master server for single sign-on Kerberos authentication. You can delegate authority to one or more user accounts. The user accounts to which you delegate authority must have a password type of Open Directory and must reside in the LDAP directory of the Open Directory master server. The dependent server for which you are delegating authority must have Mac OS X Server v10.3 or later.

Note: If an account with delegated Kerberos authority is deleted and recreated on the Open Directory master server, the new account will not have authority to join the dependent server to the Open Directory master's Kerberos realm.

A Kerberos administrator (that is, an Open Directory LDAP administrator) doesn't need delegated authority to join dependent servers to the Open Directory Kerberos realm. A Kerberos administrator has implicit authority to join any server to the Kerberos realm.

To delegate authority to join an Open Directory Kerberos realm:

- 1 In Workgroup Manager, create a computer list in the LDAP directory domain of the Open Directory master server, or select an existing computer list in this directory.
 - To select an existing computer list in Workgroup Manager, click Accounts or choose View > Accounts, click the Computers button (above the accounts list), and select the computer list you want to use.

- If the LDAP server doesn't already have a computer list to which you want to add the dependent server, you can create one.
Click Accounts, then click the Computers button.
Click the small globe icon above the list of accounts and use the pop-up menu to open the Open Directory master's LDAP directory.
Click the lock and authenticate as an administrator of the LDAP directory.
Click List (on the right), then click New Computer List or choose Server > New Computer List.
Type a list name, for example Kerberized Servers.
- 2 Click the Add [+] button, then enter the dependent server's primary Ethernet address in the Address field and the server's fully qualified DNS name in the Name field.
 - *Address:* Enter the Ethernet address of the dependent server's primary Ethernet port. The primary Ethernet port is the first one listed in Network Status pane of the dependent server's Network preferences address. This port's address is displayed as the Ethernet ID in the Ethernet pane of Network preferences. If you do not enter the correct Ethernet address, the dependent server will be unable to join the Open Directory master for Kerberos authentication.
 - *Name:* Enter the fully qualified DNS name of the dependent server, not just its host name. For example, the name might be server2.example.com (not simply server2). This name is used to create Kerberos principals in the KDC. If this name is incorrect, users will be unable to authenticate using Kerberos.
Your DNS system must have an entry for the dependent server's name and a reverse lookup entry for the dependent server's IP address.
 - *"Use this name as the Computer Name:"* Does not affect Kerberos operations.
 - *Comment:* Is optional and purely informational.
 - 3 Click Save to save your changes to the computer list.
 - 4 Click Preferences and make sure the computer list doesn't have any managed preference settings.
If any item in the array of preference categories has a small arrow next to its icon, the item has managed preference settings. To remove managed preferences from an item, click the item, select Not Managed, and click Apply Now. If the item has multiple panes, select Not Managed in each pane, then click Apply Now.
 - 5 If you want to delegate Kerberos authority to one or more new user accounts, create them now.
 - Make sure you are working in the LDAP directory of the Open Directory master server. If necessary, click the small globe icon and use the pop-up menu to open this directory. Then click the lock and authenticate as an administrator of this directory.
 - Click the Users button (on the left), then click New User or choose Server > New User.

Enter a name, short name, and password. Neither “User can log in” nor “User may administer this server” need to be selected. You may change settings in other panes, but do not change the User Password Type setting in the Advanced pane. A user with delegated Kerberos authority must have an Open Directory password.

- 6 Click Save to save the new user account.
- 7 Open Server Admin, connect to the Open Directory master server, and select Open Directory for this server in the Computers & Services list.
- 8 Click Settings (near the bottom of the window), then click General (near the top).
- 9 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 fully qualified DNS name exactly as you entered it when adding the dependent server to the computer list in step 2.
 - *Delegated Administrators*: Enter a short name or a long name for each user account to which you want to delegate Kerberos authority for the specified server. In case this user account is deleted in the future, consider entering at least two names.
- 10 Click Save to delegate Kerberos authority as specified.

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

For instructions on joining a server to an Open Directory Kerberos realm, see “Joining a Server to a Kerberos Realm” (next).

Joining a Server to a Kerberos Realm

Using Server Admin, a Kerberos administrator or a user whose account has the properly delegated authority can join Mac OS X Server to a Kerberos realm. The server can join only one Kerberos realm. It can be an Open Directory Kerberos realm, an Active Directory Kerberos realm, or an existing realm based on MIT Kerberos.

To join an Open Directory Kerberos realm, you need a Kerberos administrator account or a user account with delegated Kerberos authority. See “Delegating Authority to Join an Open Directory Kerberos Realm” on page 83 for instructions.

To join a server to a Kerberos realm:

- 1 Make sure the server that you want to join to the Kerberos realm is configured to access the shared directory domain of the Kerberos server.

To confirm, open Directory Access on the server that you want to join to the Kerberos realm, or connect to the server using Directory Access on another computer. Click Authentication and make sure the Kerberos server's directory domain is listed. If it is not listed, see Chapter 7, "Managing Directory Access," for instructions on configuring access to the directory.

- 2 Open Server Admin, connect to the server that you want to join to the Kerberos realm, and select Open Directory for the this 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 Connected to a Directory System, then click Join Kerberos and enter the requested information.
 - For an Open Directory Kerberos realm or an Active Directory Kerberos realm, choose the realm from the pop-up menu and enter the name and password of a Kerberos administrator or a user with delegated Kerberos authority for the server.
 - For an MIT-based Kerberos realm, enter the name and password of a Kerberos administrator, the Kerberos realm name, and the DNS name of the Kerberos KDC server.

Setting Options for an Open Directory Master or Replica

You can set binding, security, and password policies for an Open Directory master and its replicas. You can also set several LDAP options for an Open Directory master or replica. For instructions, see the following:

- "Setting a Binding Policy for an Open Directory Master and Replicas" (next)
- "Setting a Security Policy for an Open Directory Master and Replicas" on page 87
- "Changing the Global Password Policy" on page 99
- "Changing the Location of an LDAP Database" on page 88
- "Limiting Search Results for LDAP Service" on page 88
- "Changing the Search Timeout for LDAP Service" on page 89
- "Setting up SSL for LDAP Service" on page 89

Setting a Binding Policy for an Open Directory Master and Replicas

Using Server Admin, you can configure an Open Directory master to allow or require trusted binding between the LDAP directory and the computers that access it. Replicas of the Open Directory master automatically inherit its binding policy.

Trusted LDAP binding is mutually authenticated. The computer proves its identity by using an LDAP directory administrator's name and password to authenticate to the LDAP directory. The LDAP directory proves its authenticity by means of an authenticated computer record that's created in the directory when you set up trusted binding.

Clients can't be configured to use both trusted LDAP binding and a DHCP-supplied LDAP server (also known as DHCP option 95). Trusted LDAP binding is inherently a static binding, but DHCP-supplied LDAP is a dynamic binding. See "Enabling or Disabling Use of a DHCP-Supplied LDAP Directory" on page 118 for more information.

Note: Clients need version 10.4 or later of Mac OS X or Mac OS X Server to use trusted LDAP binding. Clients using v10.3 or earlier won't be able to set up trusted binding.

To set the binding policy for an Open Directory master:

- 1 Open Server Admin, connect to the Open Directory master server, and select this server in the Computers & Services list.
- 2 Click Settings (near the bottom of the window), then click Policy (near the top).
- 3 Click Binding, then set the directory binding options you want.
 - To allow trusted binding, select "Enable directory binding."
 - To require trusted binding, also select "Require clients to bind to directory."
- 4 Click Save.

Setting a Security Policy for an Open Directory Master and Replicas

Using Server Admin, you can configure a security policy for access to the LDAP directory of an Open Directory master.

Replicas of the Open Directory master automatically inherit its security policy.

To set the security policy for an Open Directory master:

- 1 Open Server Admin, connect to the Open Directory master server, and select this server in the Computers & Services list.
- 2 Click Settings (near the bottom of the window), then click Policy (near the top).
- 3 Click Binding, then set the security options you want.
 - "*Disable clear text passwords*" determines whether clients can send passwords as clear text if the passwords can't be validated using any authentication method that sends an encrypted password. For more information, see "Selecting Authentication Methods for Shadow Password Users" on page 101 and "Selecting Authentication Methods for Open Directory Passwords" on page 102.
 - "*Digitally sign all packets (requires Kerberos)*" ensures directory data from the LDAP server won't be intercepted and modified by another computer while en route to client computers.
 - "*Encrypt all packets (requires SSL or Kerberos)*" requires the LDAP server to encrypt directory data using SSL or Kerberos before sending it to client computers.
 - "*Block man-in-the-middle attacks (requires Kerberos)*" protects against a rogue server posing as the LDAP server. Best if used with the "Digitally sign all packets" option.
- 4 Click Save.

Subject to the settings here, the security options can also be configured individually on each client of an Open Directory master or replica. If an option is selected here, it can't be deselected for a client. For instructions on configuring these options on a client, see "Changing the Security Policy for an LDAP Connection" on page 128.

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.

You can enter a folder path in the Database field or you can select a folder location by clicking the Browse (...) button.

- 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 enable Secure Sockets Layer (SSL) for encrypted communications between an Open Directory server's LDAP directory domain and computers that access it. SSL uses a digital certificate to provide a certified identity for the server. You can use a self-signed certificate or a certificate obtained from a certificate authority. See the mail service administration guide for complete information about defining, obtaining, and installing certificates on your server.

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 Enable Secure Sockets Layer (SSL).
- 4 Use the Certificate pop-up menu to choose an SSL certificate that you want LDAP service to use.

The menu lists all SSL certificates that have been installed on the server. To use a certificate not listed, choose Custom Configuration from the pop-up menu.

- 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 after you perform an upgrade installation to Mac OS X Server v10.4 or later. After you migrate the directory domain, client computers can continue to use NetInfo to access the directory domain or they can be configured to use LDAP to access it.

You can have client computers with Mac OS X v10.3–10.4 or Mac OS X Server v10.3–10.4 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 v10.3–10.4 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. For more information, see “Switching Directory Access From NetInfo to LDAP” (the next topic).

When none of the client computers on your network needs NetInfo access to a directory domain that has been migrated to LDAP, you can disable NetInfo access to this domain by clicking a button. After NetInfo is disabled, client computers can't switch automatically to LDAP. (Access to the local NetInfo directory domain is unaffected.) See “Disabling NetInfo After Migrating to LDAP” on page 92 for instructions.

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 sign-on Kerberos authentication.

Warning: 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 v10.3–10.4 to automatically reconfigure themselves to access the migrated directory domain using LDAP instead of NetInfo.

- 5 Click OK to begin migration.

The migration process can take a while.

After migration finishes, you can 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 v10.2–10.4 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 Mac OS X or Mac OS X Server v10.3–10.4 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. See "Migrating a Directory Domain From NetInfo to LDAP" on page 90.
- You can configure each computer to access the LDAP directory instead of the NetInfo directory domain. See Chapter 7, "Managing Directory Access," for instructions.
- Computers with an automatic authentication search policy may get the address of their directory server from DHCP service. If your Open Directory server has clients like this, you can change DHCP service to supply the address of the migrated LDAP directory's server.
- When 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. See "Disabling NetInfo After Migrating to LDAP" (next).

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 access to this domain. (Access to the local NetInfo directory domain is unaffected.)

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 Before you disable NetInfo on the server, make sure DHCP is not supplying the server's address for NetInfo binding.
- 2 Open Server Admin and select Open Directory for an Open Directory master server in the Computers & Services list.
- 3 Click Settings (near the bottom of the window), then click Protocols (near the top).
- 4 Choose NetInfo Migration from the Configure pop-up menu.
- 5 Click Disable NetInfo.

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

Learn how to reset user passwords, change password types, set password policies, select authentication methods, and more.

You can manage the user authentication information stored in directory domains. For task descriptions and instructions, see:

- “Composing a Password” on page 93
- “Changing a User’s Password” on page 94
- “Resetting the Passwords of Multiple Users” on page 95
- “Changing a User’s Password Type” on page 96
This includes changing the password type to Open Directory, shadow password, or crypt password; and enabling single sign-on Kerberos.
- “Enabling Single Sign-On Kerberos Authentication for a User” on page 99
- “Changing the Global Password Policy” on page 99
- “Setting Password Policies for Individual Users” on page 100
- “Selecting Authentication Methods for Shadow Password Users” on page 101
- “Selecting Authentication Methods for Open Directory Passwords” on page 102
- “Assigning Administrator Rights for Open Directory Authentication” on page 103
- “Keeping the Primary Administrator’s Passwords in Sync” on page 104
- “Enabling LDAP Bind Authentication for a User” on page 104
- “Setting Passwords of Exported or Imported Users” on page 105
- “Migrating Passwords From Mac OS X Server v10.1 or Earlier” on page 105

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 access, use only ASCII characters for passwords.

Changing a User's Password

You can use Workgroup Manager to change the password of a user account defined in another directory domain to which you have read/write access. For example, you can change the password of a user account in the LDAP directory of an LDAP master.

Important: If you change the password of a user account that's used to authenticate a computer's LDAP directory connection, you must either make the same change to the affected computer's LDAP connection settings or configure the LDAP directory and all connections to it to use trusted binding. For instructions, see "Changing the Password Used for Authenticating an LDAP Connection" on page 136 or "Setting a Binding Policy for an Open Directory Master and Replicas" on page 86 and "Setting Up Trusted Binding to an LDAP Directory" on page 132.

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 in 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 in the Basic pane, then set the User Password Type option in 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 a User's Password Type

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

- Open Directory enables multiple legacy authentication methods and also enables single sign-on Kerberos authentication if the user's account is in the LDAP directory of an Open Directory master or replica. Open Directory passwords are stored separately from the directory domain in the Open Directory Password Server database and the Kerberos KDC. See “Changing the Password Type to Open Directory” on page 96.
- Shadow password enables multiple legacy authentication methods for user accounts in the local directory domain. Shadow passwords are stored separately from the directory domain in files readable only by the root user. See “Changing the Password Type to Shadow Password” on page 98.
- Crypt password provides basic authentication for a user account in a shared directory domain. A crypt password is stored in the user account record in the directory domain. A crypt password is required for login to Mac OS X v10.1 and earlier. See “Changing the Password Type to Crypt Password” on page 97.

Changing the Password Type to Open Directory

Using Workgroup Manager, you can specify that a user account have an Open Directory password stored in secure databases apart from the directory domain. User accounts in the following directory domains can have Open Directory passwords:

- LDAP directory domain on Mac OS X Server v10.3–10.4
- Local directory domain of Mac OS X Server v10.3 or a server upgraded from v10.3
- Directory domain on Mac OS X Server v10.2 that is configured to use a Password Server

The Open Directory password type supports single sign-on using Kerberos authentication. It also supports the Open Directory Password Server, which offers Simple Authentication and Security Layer (SASL) authentication protocols including APOP, CRAM-MD5, DHX, Digest-MD5, MS-CHAPv2, NTLMv2, NTLM (also called Windows NT or SMB-NT), LAN Manager (LM), and WebDAV-Digest.

Note: 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 103.

To specify that a user account have an Open Directory password:

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

The directory domains that support Open Directory authentication are listed earlier in this topic.

- 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 When prompted, enter and verify a new password.

The password must contain no more than 512 bytes (512 characters or fewer, depending on the language), although the network authentication protocol can impose different limits; for example, 128 characters for NTLMv2 and NTLM and 14 for LAN Manager. "Composing a Password" on page 93 provides guidelines for choosing passwords.

- 5 In 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 select an option that requires resetting (changing) the password, 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. See "Viewing Open Directory Status and Logs" on page 160 for instructions. View this Open Directory log in Server Admin.

- 6 Click Save.

Changing the Password Type to Crypt Password

If necessary, you can use Workgroup Manager to specify that a user account have a crypt password stored in the user account. The user account can be part of an LDAP directory domain or a legacy shared NetInfo domain.

User accounts not used on computers that require a crypt password should have an Open Directory password or a shadow password. A crypt password is required only for login on a client computer with Mac OS X v10.1 and earlier and on client computers with some types of UNIX.

The crypt password is stored as an encrypted value, or hash, in the user account. Because the crypt password can be recovered 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 have 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 When prompted, enter and verify a new password.

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 have 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 When prompted, enter and verify a new password.

A long password may be truncated for some authentication methods. Up to 128 characters of the password are used for NTLMv2 and NTLM, and the first 14 characters are used for LAN Manager. "Composing a Password" on page 93 provides guidelines for choosing passwords.

- 4 In 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, 02/22/2005.

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.

- 5 In the Advanced pane, click Security to enable or disable authentication methods for the user, and click OK when you have finished.
See “Setting Password Policies for Individual Users” on page 100 for additional information.
- 6 Click Save.

Enabling Single Sign-On Kerberos Authentication for a User

You enable single sign-on Kerberos authentication for a user account in an LDAP directory of Mac OS X Server by setting the account’s password type to Open Directory in the Advanced pane of Workgroup Manager.

User accounts from Mac OS X Server v10.2 that already have a password type of Open Directory must be reset to enable Kerberos and single sign-on authentication. First set the password type to Crypt Password, then set it to Open Directory. For detailed instructions, see “Changing the Password Type to Crypt Password” on page 97 and “Changing the Password Type to Open Directory” on page 96.

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 user accounts that have an Open Directory password type in 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.

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 100.

Kerberos and Open Directory Password Server maintain password policies separately. Mac OS X Server synchronizes the Kerberos password policy rules with Open Directory Password Server password policy rules.

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

- 1 Open Server Admin, connect to an Open Directory master or replica, and select Open Directory for this server in the Computers & Services list.
- 2 Click Settings (near the bottom of the window), then click Policy (near the top).
- 3 Click Passwords, then set the password policy options you want enforced for users who do not have their own individual password policies.

If you select an option that requires resetting the password, 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.

- 4 Click Save.

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

From the Command Line

You can also set password policies by using the `pwdpolicy` command in Terminal. For more information, see the Open Directory chapter of the command-line administration guide.

Setting Password Policies for Individual Users

Using Workgroup Manager, you can set password policies for individual user accounts whose password type is Open Directory or Shadow Password. The password policy for a user overrides the global password policy defined in the Authentication Settings pane of Open Directory service in Server Admin.

The password policy for a mobile user account applies when the account is used while the mobile computer is disconnected from the network. The password policy from the corresponding network user account applies while the mobile computer is connected to the network. Administrator accounts are always exempt from password policies.

To set a password policy for a user account that has an Open Directory password, 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 103.

Kerberos and Open Directory Password Server maintain password policies separately. Mac OS X Server synchronizes the Kerberos password policy rules with Open Directory Password Server password policy rules.

Do not use the Options button in 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.

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 or Shadow Password.

- 3 Change password policy options, then click OK.

If you select an option that requires resetting (changing) the password, 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.

- 4 Click Save.

From the Command Line

You can also set password policies by using the `pwpolicy` command in Terminal. For more information, see the Open Directory chapter of the command-line administration guide.

Selecting Authentication Methods for Shadow Password Users

Using Workgroup Manager, you can select which authentication methods will be available for a user account whose password type is Shadow Password. The shadow password supports the available authentication methods for compatibility with client software. If you know the user will never use client software that requires a particular authentication method, you can disable the method. See "Disabling Shadow Password Authentication Methods" on page 52 for additional information.

Important: If you disable an authentication method, its hash will be removed from the user's shadow password file the next time the user authenticates. If you enable an authentication method that was disabled, the newly enabled method's hash will be added to the user's shadow password file the next time the user authenticates for a service that can use a clear text password, such as login window or AFP. Alternatively, the user's password can be reset to add the newly enabled method's hash. The user can reset the password, or a directory administrator can do it.

If you want to enable or disable authentications for user accounts whose password type is Open Directory, see the next topic.

To enable or disable authentication methods for a Shadow Password user:

- 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 click Security.

You can click Security only if the password type is Shadow Password or Open Directory.

- 3 Select the authentication methods you want enabled and deselect the authentication methods you want disabled, then click OK.
- 4 Click Save.

From the Command Line

You can also enable or disable authentication methods for a user with a shadow password by using the `pwdpolicy` command in Terminal. For more information, see the Open Directory chapter of the command-line administration guide.

Selecting Authentication Methods for Open Directory Passwords

Using Server Admin, you can select which authentication methods will be available for all user accounts whose password type is Open Directory. The Open Directory Password Server supports the available authentication methods for compatibility with client software. If you know that users will never use client software that requires a particular authentication method, you can disable the method. See "Disabling Open Directory Authentication Methods" on page 51 for additional information.

Important: If you disable an authentication method, its hash will be removed from the password database the next time the user authenticates. If you enable an authentication method that was disabled, every Open Directory password must be reset to add the newly enabled method's hash to the password database. Users can reset their own passwords, or a directory administrator can do it.

If you want to enable or disable authentications for user accounts whose password type is Shadow Password, see the previous topic.

To enable or disable authentication methods for Open Directory passwords:

- 1 Open Server Admin, connect to an Open Directory master, and select Open Directory for this server in the Computers & Services list.
- 2 Click Settings (near the bottom of the window), then click Policy (near the top).
- 3 Click Security, then select the authentication methods you want enabled and deselect the authentication methods you want disabled.
- 4 Click Save.

Replicas of the Open Directory master automatically inherit the authentication method settings for Open Directory passwords in the LDAP directory.

From the Command Line

You can also enable or disable Password Server authentication methods for all Open Directory passwords by using the `NeST` command with the `-getprotocols` and `-setprotocols` arguments in Terminal. For more information, see the Open Directory chapter of the command-line administration guide.

Assigning Administrator Rights for Open Directory Authentication

Using Workgroup Manager and an administrator account with rights to work with Open Directory password settings, you can assign these rights to other user accounts in the same directory domain. To assign these rights, your user account must have an Open Directory password and privileges to administer user accounts. This requirement protects the security of passwords stored in the Kerberos KDC and the Open Directory Password Server database.

To assign administrator rights for Open Directory authentication to a user account:

- 1 In Workgroup Manager, open the account of interest, click Advanced and make sure Password Type is set to Open Directory password.

For more information, see “Changing the Password Type to Open Directory” on page 96.

- 2 In the Basic pane, make sure “User can administer this directory domain” is selected.
- 3 Click Privileges and make sure “Edit user accounts” is selected.

For more information on setting administrator privileges, see the user management guide.

Keeping the Primary Administrator’s Passwords in Sync

On an Open Directory server upgraded from Mac OS X Server version 10.3, the primary administrator account normally exists in both the server’s local directory and its LDAP directory. This account was copied from the local directory to the LDAP directory when the Open Directory master was created with Mac OS X Server v10.3. Initially both copies of this account have user ID 501, the same name, and the same password. Each account is an administrator of its directory domain, and both are server administrators. When you connect to the server in Workgroup Manager using the accounts’ common name and password, you are automatically authenticated for both the local directory domain and the LDAP directory domain.

If you change either password, you will no longer be automatically authenticated for both directory domains. For example, if you use the local directory administrator’s password when you connect to the server in Workgroup Manager, you will be able to make changes only in the local directory. To make changes in the LDAP directory, you’ll have to click the lock and authenticate using the LDAP administrator’s password.

Having different passwords for the primary local administrator account and the LDAP administrator account (user ID 501) can be confusing. Therefore you should keep the passwords the same.

Note: An Open Directory server created with Mac OS X Server v10.4 has different administrator accounts for its local and LDAP directories. They have different names and user IDs, so their passwords can be different without causing confusion.

Enabling LDAP Bind Authentication for a User

You can 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:

- 1 Make sure the Mac OS X computer that needs to authenticate the user account has a connection to the LDAP directory in which the user account resides and that the computer’s search policy includes the LDAP directory connection.

See “Accessing LDAP Directories” on page 117 for information about configuring LDAP server connections and the search policy.

- 2 If you configure an LDAP connection that doesn't map the Password and authentication authority attributes, bind authentication will occur automatically. See "Configuring LDAP Searches and Mappings" on page 129 for instructions.
- 3 If the connection is configured to allow clear text passwords, it should also be configured to use SSL in order to protect the clear text password while it is in transit. See "Changing the Security Policy for an LDAP Connection" on page 128 and "Changing the Connection Settings for an LDAP Directory" on page 127 for instructions.

Setting Passwords of Exported or Imported Users

When you export user accounts whose password type is Open Directory or shadow password, passwords are not exported. This protects the security of the Open Directory Password Server database and shadow password files. 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. The user management guide has instructions for working with files of exported users.

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 95.
- You can set the password of each imported user account individually in the Basic pane of Workgroup Manager. For instructions, see "Changing a User's Password Type" on page 96.

Migrating Passwords From Mac OS X Server v10.1 or Earlier

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 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. If you import these user accounts into the server's local directory domain, which is a NetInfo domain, each is automatically converted from crypt password to shadow password when the user or administrator changes the password or the user authenticates for a service that can use a recoverable authentication method. See the user management guide for information on importing 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 have Open Directory passwords or shadow passwords. 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 96 and "Changing the Password Type to Shadow Password" on page 98.

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 first from Mac OS X Server v10.1 to v10.2 and then to v10.4, existing users can continue to use their same passwords.
 - You can change some or all upgraded user accounts to have Open Directory passwords or shadow passwords, which are more secure than crypt passwords. See "Exporting and Importing Authentication Manager Users" (next) for additional information.
 - If the upgraded server has a shared NetInfo domain and you migrate it to an LDAP directory, all user accounts are automatically converted to Open Directory passwords. See "Migrating a Directory Domain From Netinfo to LDAP" on page 90 for more information.
 - Each existing user account in the server's local directory domain, which is a NetInfo domain, is automatically converted from crypt password to shadow password when the user or administrator changes the password or the user authenticates for a service that can use a recoverable authentication method.
- If you import user accounts that use Authentication Manager into the LDAP directory, they will be converted during importing to have Open Directory passwords.

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 v10.4, 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 95.
- You can set the password of each imported user account individually in the Basic pane of Workgroup Manager. For instructions, see “Changing a User’s Password Type” on page 96.

Authentication Manager is a legacy technology for securely validating passwords of Windows file service users and Apple file service users whose Mac OS 8 computers have not been upgraded with AFP client software version 3.8.3 or later. Authentication Manager works only 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 “Authentication Manager” on page 54.

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 Directory Access on a Remote Server” (next)
- “Configuring Access to Services” on page 110
- “Setting Up Search Policies” on page 113
- “Accessing LDAP Directories” on page 117
- “Accessing an Active Directory Domain” on page 139
- “Accessing an NIS Domain” on page 151
- “Using BSD Configuration Files” on page 151
- “Accessing Legacy NetInfo Domains” on page 153

Setting Up Directory Access on a Remote Server

You can use the Directory Access application on your computer to set up and manage how Mac OS X Server on a remote server accesses directory services and discovers network services.

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.

Configuring Access to Services

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 SMB/CIFS, Mac OS X does not use it to discover file services, but you can still start Windows Sharing in the Sharing pane of System Preferences and connect to a Windows file server if you know its “smb://” address.

For task descriptions and instructions, see:

- “Enabling or Disabling Active Directory Service” on page 110
- “Enabling or Disabling AppleTalk Service Discovery” on page 111
- “Enabling or Disabling BSD Flat File and NIS Directory Services” on page 111
- “Enabling or Disabling LDAP Directory Services” on page 111
- “Enabling or Disabling NetInfo Directory Services” on page 112
- “Enabling Bonjour Service Discovery” on page 112
- “Enabling or Disabling SLP Service Discovery” on page 112
- “Enabling or Disabling SMB/CIFS Service Discovery” on page 113
- “Configuring SMB/CIFS Service Discovery” on page 113

Enabling or Disabling Active Directory Service

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

If you disable Active Directory services and any Active Directory domains are part of a custom search policy, they are listed in red in the Authentication or Contacts pane of Directory Access.

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 an Active Directory Domain” on page 139.

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.

If you disable BSD and NIS directory services and any BSD or NIS domains are part of a custom search policy, they are listed in red in the Authentication or Contacts pane of Directory Access.

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 151 and “Using BSD Configuration Files” on page 151.

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 directory services provided by Mac OS X Server use LDAPv3, as do many other servers. LDAPv3 is an open standard common in mixed networks of Macintosh, UNIX, and Windows systems. Some servers use the older version, LDAPv2, to provide directory service.

If you disable LDAP directory services and any LDAP directories are part of a custom search policy, they are listed in red in the Authentication or Contacts pane of Directory Access.

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 117.

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.

If you disable NetInfo directory services and any shared NetInfo directory domains are part of a custom search policy, they are listed in red in the Authentication or Contacts pane of Directory Access.

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 153.

Enabling Bonjour Service Discovery

Bonjour service discovery is always enabled. You can’t disable the use of Bonjour to discover network services.

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/CIFS Service Discovery

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

To enable or disable SMB/CIFS 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/CIFS Service Discovery” (next).

Configuring SMB/CIFS Service Discovery

You can configure how Mac OS X uses the Server Message Block (SMB/CIFS) 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/CIFS 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 Search Policies

Directory Access defines a search policy for authentication and a search policy for contact information.

- *Authentication:* Mac OS X uses the authentication search policy to locate and retrieve user authentication information and other administrative data from directory domains.

- *Contacts*: 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.

For task descriptions and instructions, see:

- “Defining Automatic Search Policies” (next)
- “Defining Custom Search Policies” on page 115
- “Defining Local Directory Search Policies” on page 116
- “Waiting for a Search Policy Change to Take Effect” on page 117

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.

Important: If you configure Mac OS X to use an automatic authentication search policy and a DHCP-supplied LDAP server or a DHCP-supplied NetInfo domain, you will increase the risk of a malicious user gaining control of your computer. The risk is higher if your computer is configured to connect to a wireless network. See “Protecting Computers From a Malicious DHCP Server” on page 117 for more information.

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 To include an LDAP server in the automatic search policy, make sure the use of a DHCP-supplied LDAP directory is enabled in Directory Access and the DHCP service is configured to supply the LDAP server’s address.

For instructions, see “Enabling or Disabling Use of a DHCP-Supplied LDAP Directory” on page 118.

For instructions on configuring the DHCP service of Mac OS X Server, see the network services administration guide.

- 6 To include an existing shared NetInfo domain in the automatic search policy, make sure the computer is configured to bind to the NetInfo domain.

For instructions, see “Configuring NetInfo Binding” on page 154.

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.

If a directory domain specified on a computer’s custom search policy is not available, a delay will occur when the computer starts up.

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.

If you want to add a directory that isn’t listed among the available directories, make sure the computer has been configured to access the directory. For instructions, see:

- “Accessing LDAP Directories” on page 117
- “Accessing an Active Directory Domain” on page 139
- “Accessing an NIS Domain” on page 151
- “Using BSD Configuration Files” on page 151
- “Accessing Legacy NetInfo Domains” on page 153

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.

Waiting for a Search Policy Change to Take Effect

After changing the search policy in the Authentication pane or the Contacts pane of Directory Access, you need to wait 10 or 15 seconds for the change to take effect. Attempts to log in using an account from a directory domain on the authentication search policy will be unsuccessful until changes to it take effect.

Protecting Computers From a Malicious DHCP Server

Apple recommends not using an automatic authentication search policy with a DHCP-supplied LDAP server and/or a DHCP-supplied NetInfo domain in an environment where security is a major concern. A malicious hacker with access to your network can use a sham DHCP server and a sham LDAP directory or NetInfo domain to control your computer as the root user.

First, the hacker's sham DHCP server must be part of your local network, or "subnet." Thus if your computers are the only ones on your local network and they get Internet access through the NAT service of Mac OS X Server or a NAT router, this type of security breach is not possible. However, a wireless local network increases the security risk because a hacker can join a wireless local network more easily than a wired local network.

You can protect your Mac against malicious attacks from a sham DHCP server by disabling use of a DHCP-supplied LDAP directory and disabling broadcast and DHCP binding for NetInfo (or disabling NetInfo altogether). For instructions, see "Enabling or Disabling Use of a DHCP-Supplied LDAP Directory" on page 118, and "Configuring NetInfo Binding" on page 154.

If you have a mobile computer that connects to an LDAP or NetInfo server when the computer is connected to a network and you change the computer's search policy from automatic to custom (in the Authentication pane of Directory Access), a startup delay will occur when the computer is not connected to the network. The delay occurs if the computer can't connect to a specific directory domain listed in the computer's custom search policy. No delay is noticeable when waking a computer that's been disconnected from the network while sleeping.

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:

- "Accessing LDAP Directories in Mail and Address Book" on page 118
- "Enabling or Disabling Use of a DHCP-Supplied LDAP Directory" on page 118
- "Showing or Hiding Configurations for LDAP Servers" on page 119
- "Configuring Access to an LDAP Directory" on page 120
- "Configuring Access to an LDAP Directory Manually" on page 122

- “Changing a Configuration for Accessing an LDAP Directory” on page 124
- “Duplicating a Configuration for Accessing an LDAP Directory” on page 125
- “Deleting a Configuration for Accessing an LDAP Directory” on page 126
- “Changing the Connection Settings for an LDAP Directory” on page 127
- “Changing the Security Policy for an LDAP Connection” on page 128
- “Configuring LDAP Searches and Mappings” on page 129
- “Setting Up Trusted Binding to an LDAP Directory” on page 132
- “Stopping Trusted Binding With an LDAP Directory” on page 133
- “Changing the Open/Close Timeout for an LDAP Connection” on page 133
- “Changing the Query Timeout for an LDAP Connection” on page 134
- “Changing the Rebind-Try Delay Time for an LDAP Connection” on page 134
- “Changing the Idle Timeout for an LDAP Connection” on page 134
- “Forcing Read-Only LDAPv2 Access” on page 135
- “Ignoring LDAP Server Referrals” on page 135
- “Authenticating an LDAP Connection” on page 135
- “Changing the Password Used for Authenticating an LDAP Connection” on page 136
- “Mapping Config Record Attributes for LDAP Directories” on page 136
- “Editing RFC 2307 Mapping to Enable Creating Users” on page 137
- “Preparing a Read-Only LDAP Directory for Mac OS X” on page 138
- “Populating LDAP Directories With Data for Mac OS X” on page 138

Accessing LDAP Directories in Mail and Address Book

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. The DHCP-supplied LDAP server also appears (dimmed) in the list of LDAP configurations. See “Defining Automatic Search Policies” on page 114 and “Changing a Configuration for Accessing an LDAP Directory” on page 124 for more information.

The computer can’t be configured to use both a DHCP-supplied LDAP directory and trusted LDAP binding. Trusted LDAP binding is inherently a static binding, but DHCP-supplied LDAP is a dynamic binding. See “Setting Up Trusted Binding to an LDAP Directory” on page 132 and “Setting a Binding Policy for an Open Directory Master and Replicas” on page 86 for more information.

Important: If you configure Mac OS X to use an automatic authentication search policy and a DHCP-supplied LDAP server or a DHCP-supplied NetInfo domain, you will increase the risk of a malicious user gaining control of your computer. The risk is higher if your computer is configured to connect to a wireless network. See “Protecting Computers From a Malicious DHCP Server” on page 117 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 Choose a network location from the Location pop-up menu.

The DHCP-supplied LDAP option can be enabled or disabled independently for each network location that is defined in the Network pane of System Preferences.

- 5 Click “Add DHCP-supplied LDAP servers to automatic search policies.”

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 153 for more information.

If you enable this setting, the server that provides this computer with 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 Configurations for LDAP Servers

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 that isn’t dimmed in the list. Dimmed LDAP configurations are supplied by DHCP, as described in “Enabling or Disabling Use of a DHCP-Supplied LDAP Directory” on page 118.

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

Directory Access can assist you in creating a configuration that specifies how Mac OS X accesses a particular LDAPv3 directory. You need to know the DNS name or IP address of the LDAP directory server. In addition, if the directory is not hosted by a server that supplies its own mappings, such as Mac OS X Server, you need to know the search base and the template for mapping Mac OS X data to the directory's data. The supported mapping templates are:

- Open Directory Server—for a directory that uses the Mac OS X Server schema
- Active Directory—for a directory hosted by a Windows 2000 or 2003 server
- RFC 2307—for most directories hosted by UNIX servers

The LDAPv3 plug-in fully supports Open Directory replication and failover. If the Open Directory master becomes unavailable, the plug-in automatically falls back to a nearby replica.

If you need to specify custom mappings for the directory data, follow the instructions in “Configuring Access to an LDAP Directory Manually” (next) instead of the instructions here.

To have Directory Access help you configure access to 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.

You can select LDAPv3 in the list of services without selecting the Enable checkbox for LDAPv3.

- 4 Click New and enter the LDAP server's DNS name or IP address.
- 5 Select the options for accessing the directory, then click Continue to have Directory Access get information from the LDAP server.
 - Select “Encrypt using SSL” if you want Open Directory to use Secure Sockets Layer (SSL) for connections with the LDAP directory.
 - Select “Use for authentication” if this directory contains users accounts that someone will use for login or to authenticate for services.
 - Select “Use for contacts” if this directory contains email addresses and other information that you want to use in Address Book.

If Directory Access can't contact the LDAP server, it displays a message and you have to configure access manually or cancel the setup process. For manual configuration instructions, see “Configuring Access to an LDAP Directory Manually” (next).

- 6 If the dialog expands to display mappings options, choose the mapping template from the pop-up menu, enter the search base suffix, then click Continue.

Typically, the search base suffix is derived from the server's DNS name. For example, the search base suffix could be "dc=ods,dc=example,dc=com" for a server whose DNS name is ods.example.com.

If none of the available mapping templates applies to the connection you're setting up, click Manual and see "Configuring Access to an LDAP Directory Manually" (next) for additional instructions.

- 7 If the dialog expands to display options for trusted binding, enter the name of the computer and the name and password of a directory administrator (binding may be optional).
 - The dialog tells you whether the LDAP directory requires trusted binding or makes it optional. Trusted binding is mutual: each time the computer connects to the LDAP directory, they authenticate each other.
 - If trusted binding is already set up or the LDAP directory doesn't support trusted binding, the Bind button is not displayed.
 - If you see an alert saying a computer record exists, you can click Overwrite to replace the existing computer record.

Before replacing an existing computer record, make sure you supplied the correct computer name. Click Cancel to go back and change the computer name.

The existing computer record may be abandoned or it may belong to another computer. If you decide to replace an existing computer record, you should notify the LDAP directory administrator in case replacing the record disables another computer. In this case, the LDAP directory administrator needs to add the disabled computer back to the computer list to which it belonged, using a different name for that computer. For instructions on adding a computer to a computer list, see the computer lists chapter of the user management guide.

- 8 If the dialog expands to display connection options, select "Use authentication when selecting" and enter the distinguished name and password of a user account in the directory.

The options for an authenticated connection appear if the LDAP server supports an authenticated connection but not trusted binding. An authentication connection is not mutual: the LDAP server authenticates the client but the client doesn't authenticate the server.

"Use authentication when selecting" is preselected but dimmed if the LDAP server requires you to enter a user account's distinguished name and password for an authenticated connection.

The distinguished name can specify any user account that has the privilege to see data in the directory. For example, a user account whose short name is “dirauth” on an LDAP server whose address is ods.example.com would have the distinguished name uid=dirauth,cn=users,dc=ods,dc=example,dc=com.

Important: If the distinguished name or password is incorrect no one will be able to log in on the computer using user accounts from the LDAP directory.

- 9 Click OK to finish creating the new LDAP connection, then click OK to finish configuring LDAPv3 options.

If you selected the “Use for authentication” option or the “Use for contacts” option in step 5, the LDAP directory configuration you just created is automatically added to a custom search policy in the Authentication or Contacts pane of Directory Access.

You need to make sure LDAPv3 is enabled in the Services pane so the computer will use the LDAP configuration you just created. For instructions, see “Enabling or Disabling LDAP Directory Services” on page 111.

Configuring Access to an LDAP Directory Manually

You can manually create a configuration that specifies how Mac OS X accesses a particular LDAPv3 or LDAPv2 directory. You need to know the DNS name or IP address of the LDAP directory server. In addition, if the directory is not hosted by Mac OS X Server, you need to know the search base and the template for mapping Mac OS X data to the directory’s data. The supported mapping templates are:

- From Server—for a directory that supplies its own mappings and search base, such as Mac OS X Server
- Open Directory Server—for a directory that uses the Mac OS X Server schema
- Active Directory—for a directory hosted by a Windows 2000 or 2003 server
- RFC 2307—for most directories hosted by UNIX servers
- Custom—for directories that don’t use any of the above mappings

The LDAPv3 plug-in fully supports Open Directory replication and failover. If the Open Directory master becomes unavailable, the plug-in automatically falls back to a nearby replica.

To manually configure access to 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.

You can select LDAPv3 in the list of services without selecting the Enable checkbox for LDAPv3.

- 4 Click New, then click Manual.
- 5 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 method:
 - *If you choose Server*, you don't need to enter a search base suffix. In this case, Open Directory assumes the search base suffix is the first level of the LDAP directory.
 - *If you choose a template*, enter the search base suffix for the LDAP directory and click OK. 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=ods, dc=example, dc=com" for a server whose DNS name is ods.example.com.
 - *If you choose Custom*, you now need to set up mappings between Mac OS X record types and attributes and the classes and attributes of the LDAP directory you're connecting to. For instructions, see "Configuring LDAP Searches and Mappings" on page 129.
- 8 Select the SSL checkbox if you want Open Directory to use Secure Sockets Layer (SSL) for connections with the LDAP directory.
- 9 If you want to change the settings for this LDAP configuration's trusted binding, connection options, or security policy, click Edit to display the options for the selected LDAP configuration, and click OK when you finish editing the LDAP configuration options.
 - Click Bind to set up trusted binding (if the LDAP directory supports it). For detailed instructions, see "Setting Up Trusted Binding to an LDAP Directory" on page 132.
 - Click Connection to set timeout options, specify a custom port, ignore server referrals, or force use of the LDAPv2 (read-only) protocol. For detailed instructions, see "Changing the Connection Settings for an LDAP Directory" on page 127.
 - Click Security to set up an authenticated connection (instead of trusted binding) and other security policy options. For detailed instructions, see "Changing the Security Policy for an LDAP Connection" on page 128.
- 10 Click OK to finish manually creating the configuration to access an LDAP directory.
- 11 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 111 and "Defining Custom Search Policies" on page 115.

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 137.

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. You can't change an LDAP configuration that was supplied by DHCP, and such a configuration appears dimmed in the LDAP configurations list.

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 suffix 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=ods, dc=example, dc=com" for a server whose DNS name is ods.example.com.

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

If you choose Custom, you now need to set up mappings between Mac OS X record types and attributes and the classes and attributes of the LDAP directory you're connecting to. For instructions, see "Configuring LDAP Searches and Mappings" on page 129.
 - *SSL*: Click a checkbox to enable or disable encrypted communications using the Secure Sockets Layer (SSL) protocol.
- 6 If you want to change the default settings for this LDAP configuration's trusted binding, connection options, or security policy, click Edit to display the options for the selected LDAP configuration, and click OK when you finish editing the LDAP configuration options.
 - Click the Bind button to set up trusted binding or the Unbind button to stop trusted binding. (You may not see these buttons if the LDAP directory doesn't allow trusted binding.) For detailed instructions, see "Setting Up Trusted Binding to an LDAP Directory" on page 132.

- Click Connection to set timeout options, specify a custom port, ignore server referrals, or force use of the LDAPv2 (read-only) protocol. For detailed instructions, see “Changing the Connection Settings for an LDAP Directory” on page 127.
 - Click Security to set up an authenticated connection (instead of trusted binding) and other security policy options. For detailed instructions, see “Changing the Security Policy for an LDAP Connection” on page 128.
- 7 Click OK to finish changing the configuration to access an LDAP directory.

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 suffix 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=ods, dc=example, dc=com” for a server whose DNS name is ods.example.com.

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

If you choose Custom, you now need to set up mappings between Mac OS X record types and attributes and the classes and attributes of the LDAP directory you're connecting to. For instructions, see “Configuring LDAP Searches and Mappings” on page 129.
 - *SSL*: Click a checkbox to enable or disable Secure Sockets Layer (SSL) connections.

- 7 If you want to change the default settings for the duplicate LDAP configuration's trusted binding, connection options, or security policy, click Edit to display the options, and click OK when you finish editing them.
 - Click the Bind button to set up trusted binding or the Unbind button to stop trusted binding. (You may not see these buttons if the LDAP directory doesn't allow trusted binding.) For detailed instructions, see "Setting Up Trusted Binding to an LDAP Directory" on page 132.
 - Click Connection to set up trusted binding (if the LDAP directory supports it), set timeout options, specify a custom port, ignore server referrals, or force use of the LDAPv2 (read-only) protocol. For detailed instructions, see "Changing the Connection Settings for an LDAP Directory" on page 127.
 - Click Security to set up an authenticated connection (instead of trusted binding) and other security policy options. For detailed instructions, see "Changing the Security Policy for an LDAP Connection" on page 128.
- 8 Click OK to finish changing the duplicate configuration.
- 9 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 111, and "Defining Custom Search Policies" on page 115.

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. You can't delete an LDAP configuration that was supplied by DHCP, and such a configuration appears dimmed in the LDAP configurations list.

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.
- 6 If you see an alert saying the computer is bound to the LDAP directory and you want to stop trusted binding: click OK, then enter the credentials requested.
 - Enter the name and password of an LDAP directory administrator (not a local computer administrator).
 - If you see an alert saying the computer can't contact the LDAP server, you can click OK to forcibly stop trusted binding.

If you forcibly stop trusted binding, this computer will still have a computer record in the LDAP directory. You should notify the LDAP directory administrator so the administrator knows to remove the computer from its computer list. For instructions on removing a computer from its computer list, see the computer lists chapter of the user management guide.

The deleted configuration is automatically removed from the custom search policies for authentication and contacts.

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 to wait before cancelling an attempt to connect to the LDAP server.
 - *"Query times out in"* specifies the number of seconds to wait before cancelling a query sent to the LDAP directory.
 - *"Re-bind attempted in"* specifies the number of seconds to wait before attempting to reconnect if the LDAP server fails to respond. You can increase this value to prevent continuous reconnect attempts.
 - *"Connection idles out in"* specifies the number of seconds to allow an idle or unresponsive connection to remain open.
 - *"Encrypt using SSL"* determines whether to encrypt 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).

- “Ignore server referrals” determines whether to ignore or follow an LDAP server’s referral to look on other LDAP servers or replicas for information. Server referrals can help a computer find information, but can also delay login or cause other delays if the computer ends up chasing referrals to many LDAP servers.
- “Use LDAPv2 (read only)” determines whether to use the older LDAPv2 protocol for read-only access to an LDAP directory.

Changing the Security Policy for an LDAP Connection

Using Directory Access, you can configure a stricter security policy for an LDAPv3 connection than the security policy of the LDAP directory. For example, if the LDAP directory’s security policy allows clear text passwords, you can set an LDAPv3 connection to not allow clear text passwords.

Setting a stricter security policy protects your computer against a malicious hacker trying to use a rogue LDAP server to gain control of your computer.

The computer needs to communicate with the LDAP server to accurately show the state of the security options. Therefore when you change security options for an LDAPv3 connection, the computer’s authentication search policy should include the LDAPv3 connection.

The permissible settings of an LDAPv3 connection’s security options are subject to the LDAP server’s security capabilities and requirements. For example, if the LDAP server doesn’t support Kerberos authentication, several of the LDAPv3 connection’s security options are disabled.

To change an LDAPv3 connection’s security options:

- 1 In Directory Access, click Authentication and make sure the LDAPv3 directory of interest is listed in the search policy.
For instructions on adding the LDAPv3 directory to the authentication search policy, see “Defining Custom Search Policies” on page 115.
- 2 If the lock icon is locked, click it and type the name and password of an administrator.
- 3 Click Services, 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 configuration for the directory of interest, then click Edit.

6 Click Security and change any of the settings.

If any of the last four options is selected but disabled, the LDAP directory requires it. If any of these options is unselected and disabled, the LDAP server doesn't support it. For instructions on setting these options for a Mac OS X Server LDAP directory, see "Setting a Security Policy for an Open Directory Master and Replicas" on page 87.

- *"Use authentication when connecting"* determines whether the LDAPv3 connection authenticates itself with the LDAP directory by supplying the specified Distinguished Name and Password. This option is not shown if the LDAPv3 connection uses trusted binding with the LDAP directory.
- *"Bound to the directory as"* specifies the credentials that the LDAPv3 connection uses for trusted binding with the LDAP directory. This option and the credentials can't be changed here. Instead, you can unbind and then bind again with different credentials. See "Stopping Trusted Binding With an LDAP Directory" on page 133 and "Setting Up Trusted Binding to an LDAP Directory" on page 132 for instructions. This option is not shown unless the LDAPv3 connection uses trusted binding.
- *"Disable clear text passwords"* determines whether the password is sent as clear text if the password can't be validated using any authentication method that sends an encrypted password. For more information, see "Selecting Authentication Methods for Shadow Password Users" on page 101 and "Selecting Authentication Methods for Open Directory Passwords" on page 102.
- *"Digitally sign all packets (requires Kerberos)"* ensures directory data from the LDAP server hasn't been intercepted and modified by another computer while en route to your computer.
- *"Encrypt all packets (requires SSL or Kerberos)"* requires the LDAP server to encrypt directory data using SSL or Kerberos before sending it to your computer.
- *"Block man-in-the-middle attacks (requires Kerberos)"* protects against a rogue server posing as the LDAP server. Best if used with the "Digitally sign all packets" option.

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 the Appendix, “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.
- 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 and search scope of a record type, select it in the Record Types and Attributes List. Then edit the “Search base” field. Select “all subtrees” to set the search scope to include the entire LDAP directory’s hierarchy from the search base down. Select “first level only” to set the search scope to include only the search base and one level below it in the LDAP directory’s hierarchy.
 - 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 Save Template if you want to save your mappings as a template.

Templates saved in the default location are listed in pop-up menus of LDAP mapping templates the next time the current user opens Directory Access. The default location for saved templates is in the current user's home folder at this path:

~/Library/Application Support/Directory Access/LDAPv3/Templates

- 11 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 or other user with write permission for the search base (for example, uid=diradmin,cn=users,dc=ods,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=ods,dc=example,dc=com”).

The LDAP directory supplies its mappings to Mac OS X clients whose custom search policy includes a connection that's configured to get mappings from the LDAP server. The LDAP directory also supplies its mappings to all Mac OS X clients that have an automatic search policy. For instructions, see "Configuring Access to an LDAP Directory" on page 120 and "Setting Up Search Policies" on page 113.

Setting Up Trusted Binding to an LDAP Directory

You can use Directory Access to set up trusted binding between the computer and an LDAP directory that supports trusted binding. The binding is mutually authenticated by means of an authenticated computer record that's created in the directory when you set up trusted binding.

The computer can't be configured to use both trusted LDAP binding and a DHCP-supplied LDAP directory. Trusted LDAP binding is inherently a static binding, but DHCP-supplied LDAP is a dynamic binding. See "Enabling or Disabling Use of a DHCP-Supplied LDAP Directory" on page 118 and "Setting a Binding Policy for an Open Directory Master and Replicas" on page 86 for more information.

To set up trusted binding to 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 the server configuration of interest, then click Edit.
- 6 Click Bind, then enter the credentials requested and click OK.

Enter the name of the computer and the name and password of an LDAP directory domain administrator. The computer name can't already be in use by another computer for trusted binding or other network services.

If the Bind button isn't displayed, the LDAP directory doesn't support trusted binding.

- 7 If you see an alert saying a computer record exists, you can click Overwrite to replace the existing computer record.

Before replacing an existing computer record, make sure you supplied the correct computer name in the previous step. Click Cancel to go back and change the computer name.

The existing computer record may be abandoned or it may belong to another computer. If you decide to replace an existing computer record, you should notify the LDAP directory administrator in case replacing the record disables another computer. In this case, the LDAP directory administrator needs to add the disabled computer back to the computer list to which it belonged, using a different name for that computer. For instructions on adding a computer to a computer list, see the computer lists chapter of the user management guide.

- 8 Click OK to finish setting up trusted binding.

Stopping Trusted Binding With an LDAP Directory

You can use Directory Access to stop trusted binding between a computer and an LDAP directory that allows but doesn't require trusted binding.

To stop trusted binding to 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 the server configuration of interest, then click Edit.
- 6 Click Unbind, then enter the credentials requested and click OK.

Enter the name and password of an LDAP directory administrator (not a local computer administrator).

- If trusted binding hasn't been set up on this computer, the Unbind button is not displayed.
- If you see an alert saying the computer can't contact the LDAP server, you can click OK to forcibly stop trusted binding.

If you forcibly stop trusted binding, this computer will still have a computer record in the LDAP directory. You should notify the LDAP directory administrator so the administrator knows to remove the computer from its computer list. For instructions on removing a computer from its computer list, see the computer lists chapter of the user management guide.

- 7 Click OK to finish stopping trusted binding.

Changing the Open/Close Timeout for an LDAP Connection

Using Directory Access, you can specify how long Open Directory waits before cancelling an attempt to connect to the LDAP server.

To set the open/close timeout for an LDAP connection:

- 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 enter a value for "Open/close times out in ___ seconds."

The default value is 15 seconds.

Changing the Query Timeout for an LDAP Connection

Using Directory Access, you can specify how long Open Directory waits before cancelling a query sent to the LDAP directory.

To set the query timeout for an LDAP connection:

- 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 enter a value for “Query times out in __ seconds.”
The default value is 120 seconds.

Changing the Rebind-Try Delay Time for an LDAP Connection

Using Directory Access, you can specify how long to wait before attempting to reconnect if the LDAP server fails to respond. You can increase this value to prevent continuous reconnect attempts.

To set the rebind delay for idle LDAP clients:

- 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 enter a value for “Rebind attempted in __ seconds.”
The default value is 120 seconds.

Changing the Idle Timeout for an LDAP Connection

Using Directory Access, you can specify how long an LDAP connection will remain idle before Open Directory closes the connection. You can adjust this setting to reduce the number of open connections on the LDAP server.

To set a timeout interval for an idle LDAP connection:

- 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 enter a value for “Connection idles out in __ minutes.”

The default value is 120 seconds.

Forcing Read-Only LDAPv2 Access

Using Directory Access, you can force a connection to an LDAP server using the legacy LDAPv2 protocol. This forced LDAPv2 connection is read-only (not read-write) and doesn't use SSL.

To force read-only LDAPv2 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 If the list of server configurations is hidden, click Show Options.
- 5 Select the server configuration in the list, then click Edit.
- 6 Click Connection and select "Use LDAPv2 (read only)."

Ignoring LDAP Server Referrals

Using Directory Access, you can specify whether the computer ignores or follows an LDAP server's referral to look on other LDAP servers or replicas for information. Server referrals can help a computer find information, but can also delay login or cause other delays if the computer ends up chasing referrals to many LDAP servers.

To specify whether to ignore LDAP server referrals:

- 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 select "Ignore server referrals."

Authenticating an LDAP Connection

Using Directory Access, you can set up an authenticated connection to an LDAP directory. This authentication is one way. The computer proves its identity to an LDAP directory, but the LDAP directory doesn't prove its authenticity to the computer. For mutual authentication, see "Setting Up Trusted Binding to an LDAP Directory" on page 132.

Note: If the trusted binding is already set up between the computer and the LDAP directory, an authenticated connection would be redundant and you can't set one up.

To set up an authenticated LDAPv3 connection:

- 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 Security, select “Use authentication when connecting,” and enter a user’s distinguished name and password.

The distinguished name can specify any user account that has the privilege to see data in the directory. For example, a user account whose short name is “authenticator” on an LDAP server whose address is ods.example.com would have the distinguished name `uid=authenticator,cn=users,dc=ods,dc=example,dc=com`.

Important: If the distinguished name or password is incorrect no one will be able to log in on the computer using user accounts from the LDAP directory.

Changing the Password Used for Authenticating an LDAP Connection

Using Directory Access, you can update an authenticated LDAP connection to use a new password that has been changed on the LDAP server. (All computers having an authenticated connection to an LDAP server need to be updated if the password used to authenticate the LDAP connection is changed on the server.)

To change the password for an LDAP connection:

- 1 In Directory Access, click the Services tab.
- 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 Security and change the Password setting.
 - If the Password setting is dimmed because “Use authentication when connecting” is unselected, see “Authenticating an LDAP Connection” on page 135.
 - If the Password setting is dimmed because “Bound to the directory as” is selected (albeit dimmed), the connection isn’t authenticated with a user password. Instead, the connection uses an authenticated computer record for trusted binding.

Mapping Config Record Attributes for LDAP Directories

If you want to store information for managed Mac OS X users in a non-Apple 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 the Appendix, "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 LDAP directories 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 users' Network browsers (what users see when they click Network in a Finder window sidebar). Use the Sharing and Network modules 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 host

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:

- “About the Active Directory Plug-in” (next)
- “Configuring Access to an Active Directory Domain” on page 141
- “Setting Up Mobile User Accounts in Active Directory” on page 143
- “Setting Up Home Folders for Active Directory User Accounts” on page 143
- “Setting a UNIX Shell for Active Directory User Accounts” on page 144
- “Mapping the UID to an Active Directory Attribute” on page 145
- “Mapping the Primary Group ID to an Active Directory Attribute” on page 145
- “Mapping the Group ID in Group Accounts to an Active Directory Attribute” on page 146
- “Specifying a Preferred Active Directory Server” on page 147
- “Changing the Active Directory Groups That Can Administer the Computer” on page 147
- “Controlling Authentication From All Domains in the Active Directory Forest” on page 148
- “Unbinding From the Active Directory Server” on page 149
- “Editing User Accounts and Other Records in Active Directory” on page 149

Alternative methods for accessing an Active Directory domain are appropriate for some networks. See “Setting Up LDAP Access to Active Directory Domains” on page 149.

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 in 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 user 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 Active Directory attributes that you specify.

Likewise, the Active Directory plug-in generates a group ID based on the Active Directory group account's GUID. You can also force the plug-in to map the group ID for group accounts to Active Directory attributes that you specify.

When someone logs in to Mac OS X with an Active Directory user account, the Active Directory plug-in can automatically mount the Windows network home directory that's specified in the Active Directory user account as the user's Mac OS X home folder. You can specify whether to use the network home specified by Active Directory's standard homeDirectory attribute or by Mac OS X's HomeDirectory attribute, if the Active Directory schema has been extended to include it.

Alternatively, you can configure the plug-in to create a local home folder on the startup volume of the Mac OS X client computer. In this case, the plug-in also mounts the user's Windows network home directory (specified in the Active Directory user account) as a network volume, like a share point. Using the Finder, the user can copy files between the Windows home directory network volume and the local Mac OS X home folder.

The Active Directory plug-in can also create mobile accounts for users. A mobile account caches the user's Active Directory authentication credentials on the Mac OS X client computer. The cached credentials allow the user to log in using the Active Directory name and password while the client computer is disconnected from the Active Directory server. A mobile account has a local home folder on the startup volume of the Mac OS X client computer. (The user also has a network home folder as specified in the user's Active Directory account.)

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 managed client attributes. This schema modification would enable the Active Directory plug-in to support managed client settings made using Mac OS X Server's Workgroup Manager application. 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 lists 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 multidomain 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.

Warning: Advanced options of the Active Directory plug-in allow you to map the Mac OS X unique user ID (UID), primary group ID (GID), and group GID attribute to appropriate attributes that have been added to the Active Directory schema. If you change the setting of any of these mapping options 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 name of the Active Directory domain to which you want to bind the computer you're configuring.

The administrator of the Active Directory domain can tell you the DNS name to enter.

- 5 If necessary, edit the Computer ID.

The Computer ID is the name by which the computer will be known in the Active Directory domain, and it's preset to the name of the computer. You may need to change this to conform to your organization's established scheme for naming computers in the Active Directory domain. If you're not sure, ask the Active Directory domain administrator.

- 6 Optionally, set the advanced options.

If the advanced options are hidden, click Show Advanced Options and set options in the User Experience, Mappings, and Administrative panes. You can also change the advanced option settings later. For detailed instructions on the advanced options, see:

- "Setting Up Mobile User Accounts in Active Directory" on page 143
- "Setting Up Home Folders for Active Directory User Accounts" on page 143
- "Setting a UNIX Shell for Active Directory User Accounts" on page 144

- “Mapping the UID to an Active Directory Attribute” on page 145
 - “Mapping the Primary Group ID to an Active Directory Attribute” on page 145
 - “Mapping the Group ID in Group Accounts to an Active Directory Attribute” on page 146
 - “Specifying a Preferred Active Directory Server” on page 147
 - “Changing the Active Directory Groups That Can Administer the Computer” on page 147
 - “Controlling Authentication From All Domains in the Active Directory Forest” on page 148
- 7 Click Bind, authenticate as a user who has rights to bind a computer to the Active Directory domain, select the search policies to which you want Active Directory added, and click OK.
- *Username 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.
 - *Computer OU:* Enter the organizational unit (OU) for the computer you’re configuring.
 - *Use for authentication:* Determines whether Active Directory is automatically added to the computer’s authentication search policy.
 - *Use for contacts:* Determines whether Active Directory is automatically added to the computer’s contacts search policy.

When you click OK, Directory Access sets up trusted binding between the computer you’re configuring and the Active Directory server. The computer’s search policies are set up according to the options you selected when you authenticated, and Active Directory is enabled in Directory Access’s Services pane.

With the default settings for Active Directory advanced options, the Active Directory forest is added to the computer’s authentication search policy and/or contacts search policy if you selected the “Use for authentication” option and/or the “Use for contacts” option. But if you deselect the “Allow authentication from any domain in the forest” option in the Administrative advanced options pane before clicking Bind, the nearest Active Directory domain is added instead of the forest. You can change the search policies later by adding or removing the Active Directory forest or individual domains. See “Defining Custom Search Policies” on page 115 for instructions.

- 8 If you’re configuring a server to access an Active Directory domain, you can also join the server to the Active Directory Kerberos realm.

On the server or an administrator computer that can connect to the server, open Server Admin and select Open Directory for the server. Click Settings, then click General. Click Join Kerberos, then choose the Active Directory Kerberos realm from the pop-up menu and enter credentials for a local administrator on this server. For detailed instructions, see “Joining a Server to a Kerberos Realm” on page 85.

Setting Up Mobile User Accounts in Active Directory

You can start or stop using mobile Active Directory user accounts on a computer that is configured to use Directory Access's Active Directory plug-in. Users with mobile accounts can log in using their Active Directory credentials while the computer is not connected to the Active Directory server. The Active Directory plug-in caches credentials for a user's mobile account when the user logs in while the computer is connected to the Active Directory domain. 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 mobile account setting.

You can have mobile accounts created automatically, or you can require Active Directory users to confirm creation of a mobile account.

To enable or disable mobile accounts 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 User Experience, then click "Create mobile account at login" and optionally click "Require confirmation before creating a mobile account."
 - If both options are selected, each user decides whether to create a mobile account during login. When a user logs in to Mac OS X using an Active Directory user account, the user sees a dialog with controls for creating a mobile account immediately or logging in as a network user.
 - If the first option is selected and the second option is unselected, mobile accounts are created automatically when users log in.
 - If the first option is unselected, the second option is disabled.
- 6 Click OK.

Setting Up Home Folders for Active Directory User Accounts

On a computer that's configured to use Directory Access's Active Directory plug-in, you can start or stop using network home folders or local home folders for Active Directory user accounts. With network home folders, a user's Windows network home directory is mounted as the Mac OS X home folder when the user logs in. You select whether the network home folder location comes from Active Directory's standard homeDirectory attribute or from Mac OS X's HomeDirectory attribute, if the Active Directory schema has been extended to include it.

With local home folders, each Active Directory user who logs in has a home folder on the Mac OS X startup disk. In addition, the user's network home folder is mounted as a network volume, like a share point. The user can copy files between this network volume and the local home folder.

To set up home folders for Active Directory user accounts:

- 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 User Experience.
- 6 Click "Force local home directory on startup disk" if you want Active Directory user accounts to have local home folders in the computer's /Users folder.

This option is not available if "Create mobile account at login" is selected.

- 7 To use Active Directory's standard attribute for the home folder location, select "Use UNC path from Active Directory to derive network home location" and choose the protocol for accessing the home folder.

- Choose "smb:" to use the standard Windows protocol, SMB/CIFS.
- Choose "afp:" to use the standard Macintosh protocol, AFP.

- 8 To use Mac OS X's attribute for the home folder location, deselect "Use UNC path from Active Directory to derive network home location."

To use the Mac OS X's attribute, the Active Directory schema must be extended to include it.

- 9 Click OK.

If you change the name of a user account in the Active Directory domain, the server will create a new home directory folder (and subfolders) for the user account the next time it is used for logging in to a Mac OS X computer.

The user can navigate to the old home directory and see its contents in the Finder.

You can prevent creation of a new home directory folder by renaming the old folder before the user next logs in.

Setting a UNIX Shell for Active Directory User Accounts

On a computer that's configured to use Directory Access's Active Directory plug-in, you can set the command-line shell that users with Active Directory accounts will use by default when interacting with Mac OS X in the Terminal application. The default shell is also used for remote interaction via SSH (Secure Shell) or Telnet. Each user can override the default shell by changing a Terminal preference.

To set a UNIX shell for Active Directory user accounts:

- 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 User Experience, then enter the default user shell's path.
- 6 Click OK.

Mapping the UID to an Active Directory Attribute

On a computer that's configured to use Directory Access's Active Directory plug-in, you can specify an Active Directory attribute that you want mapped to Mac OS X's unique user ID (UID) attribute.

Usually the Active Directory schema must be extended to include an attribute that's suitable for mapping to the UID.

- If the Active Directory administrator extends the Active Directory schema by installing Microsoft's Services for UNIX, you can map the UID to the msSFU-30-Uid-Number attribute.
- If the Active Directory administrator manually extends the Active Directory schema to include RFC 2307 attributes, you can map the UID to uidNumber.
- If the Active Directory administrator manually extends the Active Directory schema to include the Mac OS X UniqueID attribute, you can map the UID to it.

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

Warning: 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 Click Mappings, then select "Map UID to attribute" and enter the name of the Active Directory attribute you want mapped to the UID.

Mapping the Primary Group ID to an Active Directory Attribute

On a computer that's configured to use Directory Access's Active Directory plug-in, you can specify an Active Directory attribute that you want mapped to Mac OS X's primary group ID (GID) attribute in user accounts.

Usually the Active Directory schema must be extended to include an attribute that's suitable for mapping to the primary GID.

- If the Active Directory administrator extends the Active Directory schema by installing Microsoft's Services for UNIX, you can map the primary GID to the msSFU-30-Gid-Number attribute.
- If the Active Directory administrator manually extends the Active Directory schema to include RFC 2307 attributes, you can map the primary GID to gidNumber.
- If the Active Directory administrator manually extends the Active Directory schema to include the Mac OS X PrimaryGroupID attribute, you can map the primary GID to it.

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

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

To map the primary GID 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 Click Mappings, then select "Map user GID to attribute" and enter the name of the Active Directory attribute you want mapped to the primary group ID in user accounts.

Mapping the Group ID in Group Accounts to an Active Directory Attribute

On a computer that's configured to use Directory Access's Active Directory plug-in, you can specify an Active Directory attribute that you want mapped to Mac OS X's group ID (GID) attribute in group accounts.

Usually the Active Directory schema must be extended to include an attribute that's suitable for mapping to the group GID.

- If the Active Directory administrator extends the Active Directory schema by installing Microsoft's Services for UNIX, you can map the group GID to the msSFU-30-Gid-Number attribute.
- If the Active Directory administrator manually extends the Active Directory schema to include RFC 2307 attributes, you can map the group GID to gidNumber.
- If the Active Directory administrator manually extends the Active Directory schema to include the Mac OS X gidNumber attribute, you can map the group GID to it.

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

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

To map the group GID 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 Click Mappings, then select "Map group GID to attribute" and enter the name of the Active Directory attribute you want mapped to the group ID in group accounts.

Specifying a Preferred Active Directory Server

On a computer that's configured to use Directory Access's Active Directory plug-in, you can specify the DNS name of the server whose Active Directory domain you want the computer 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 Click Administrative, then select "Prefer this domain server" and enter the DNS name of the Active Directory server.

Changing the Active Directory Groups That Can Administer the Computer

On a computer that's configured to use Directory Access's Active Directory plug-in, you can identify Active Directory group accounts whose members you want to have administrator privileges for the computer. Users that are members of these Active Directory group accounts can perform administrative tasks such as installing software on the Mac OS X computer that you are configuring.

To add or remove Active Directory group accounts whose members 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 Click Administrative, select "Allow administration by," then change the list of Active Directory groups accounts whose members you want to have administrator privileges.
 - Add a group by clicking the Add button (+) and entering the Active Directory domain name, a backslash, and the group account name (for example, ADS\Domain Admins, IL2\Domain Admins).
 - Remove a group by selecting it in the list and clicking the Remove button (-).

Controlling Authentication From All Domains in the Active Directory Forest

On a computer that's configured to use Directory Access's Active Directory plug-in, you can allow users from all domains in the Active Directory forest to authenticate, or you can restrict authentication to users from individual domains.

To control whether users can authenticate from all domains in the forest:

- 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 Administrative, then click "Allow authentication from any domain in the forest."
- 6 After changing the setting of this option, you need to change the custom search policy in the Authentication pane and/or Contacts pane to include the Active Directory forest or selected domains as appropriate.

See "Defining Custom Search Policies" on page 115 for instructions on changing a custom search policy.

- If you select "Allow authentication from any domain in the forest," you can add the Active Directory forest to the computer's custom search policies for authentication and contacts. When adding to a custom search policy, the forest appears in the list of available directory domains as "/Active Directory/All Domains." (This is the default setting.)
- If you deselect "Allow authentication from any domain in the forest," you can add Active Directory domains individually to the computer's custom search policies for authentication and contacts. When adding to a custom search policy, each Active Directory domain appears separately in the list of available directory domains.

Unbinding From the Active Directory Server

If the computer is using Directory Access's Active Directory plug-in to bind to an Active Directory server, you can unbind the computer from the Active Directory server. You can forcibly unbind if the computer is currently unable to contact the server or the computer record has already been removed from the server.

To unbind the computer from the Active Directory 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 Active Directory in the list of services, then click Configure.
- 4 Click Unbind, then authenticate as a user who has rights to terminate a connection to the Active Directory domain and click OK.

If you see an alert saying the credentials weren't accepted or the computer can't contact Active Directory, you can click Force Unbind to forcibly break the connection.

If you forcibly unbind, Active Directory will still contain a computer record for this computer. You should notify the Active Directory administrator so the administrator knows to remove the computer record.

- 5 In the Services pane, deselect Active Directory's Enable setting, then click Apply.

Editing User Accounts and Other Records in Active Directory

You can use Workgroup Manager to make changes to user accounts, group accounts, computer lists, and other records in an Active Directory domain. You can also use Workgroup manager to delete records in an Active Directory domain. If the Active Directory schema has been extended to include standard Mac OS X record types (object classes) and attributes, you can use Workgroup Manager to create as well as edit computer lists in the Active Directory domain. For instructions on working with user accounts, group accounts, and computer lists, see the user management guide.

To create user accounts or group accounts 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; mobile user accounts with cached authentication credentials; discovery of all domains in an Active Directory forest; and support for Active Directory replication and failover. See “About the Active Directory Plug-in” on page 139 for more information.

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 Click New and enter the Active Directory server’s DNS name or IP address.
- 5 Select the options for accessing the directory, then click Continue to have Directory Access get information from the Active Directory server.
 - Select “Encrypt using SSL” if you want Open Directory to use Secure Sockets Layer (SSL) for connections with the Active Directory server.
 - Select “Use for authentication” if this directory contains users accounts that someone will use for login or to authenticate for services.
 - Select “Use for contacts” if this directory contains email addresses and other information that you want to use in Address Book.

If Directory Access can’t contact the Active Directory server, it displays a message and you have to configure access manually or cancel the setup process. For manual configuration instructions, see “Configuring Access to an LDAP Directory Manually” on page 122.

- 6 When the dialog expands to display mappings options, choose Active Directory from the pop-up menu, enter the search base, then click Continue.

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. See “Configuring Access to an Active Directory Domain” for instructions.)

- 7 When the dialog expands to display connection options, enter the distinguished name and password of an Active Directory user account.
- 8 Click OK to finish creating the new LDAP connection, then click OK to finish configuring LDAPv3 options.

If you selected the “Use for authentication” option or the “Use for contacts” option in step 5, the LDAPv3 connection to the Active Directory domain is automatically added to a custom search policy in the Authentication or Contacts pane of Directory Access.

You need to make sure LDAPv3 is enabled in the Services pane so the computer will use the connection you just set up. For instructions, see “Enabling or Disabling LDAP Directory Services” on page 111.

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

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.

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

- 6 Select “Use NIS domain for authentication,” then click OK.

The NIS domain is added to the computer’s authentication search policy as `/BSD/domain`, where *domain* is what you entered in step 4.

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 systems.

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 Make sure the BSD configuration files contain the data required by Mac OS X directory services.

See “Setting Up Data in BSD Configuration Files” on page 152 for instructions.

- 2 In Directory Access, click Services.

- 3 If the lock icon is locked, click it and type the name and password of an administrator.
- 4 Select “BSD Flat File and NIS” in the list of services, then click Configure.
- 5 Select “Use BSD local files (/etc) for authentication,” then click OK.

The BSD configuration files domain is added to the computer’s authentication search policy as /BSD/local.

Mac OS X Server 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.

BSD configuration file	Contains
/etc/master.passwd	User names, passwords, IDs, primary group IDs, and so forth
/etc/group	Group names, IDs, and members
/etc/fstab	NFS mounts
/etc/hosts	Computer names and addresses
/etc/networks	Network names and addresses
/etc/services	Service names, ports, and protocols
/etc/protocols	IP protocol names and numbers
/etc/rpcs	Open Network Computing RPC servers
/etc/printcap	Printer names and capabilities
/etc/bootparams	Bootparam settings
/etc/bootp	Bootp settings
/etc/aliases	Email aliases and distribution lists
/etc/netgroup	Network-wide group names and members

For detailed specifications of the data required by Mac OS X directory services, see the Appendix, “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:

- Mac OS X version 10.4 and Mac OS X Server version 10.4 can access any existing shared NetInfo domain.
- 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 10.4. However, an upgraded server's shared NetInfo domain can be converted to LDAP, and then other computers and servers must use LDAP instead of NetInfo to access the server's shared directory.

Note: You cannot create a new shared NetInfo domain with Mac OS X Server version 10.4 except by using command-line utilities. If you use Server Assistant or Server Admin to set up Mac OS X Server version 10.4 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 access to 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 155.
- 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 156.

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.

Important: If you configure Mac OS X to use an automatic authentication search policy and a DHCP-supplied LDAP server or a DHCP-supplied NetInfo domain, you will increase the risk of a malicious user gaining control of your computer. The risk is higher if your computer is configured to connect to a wireless network. See "Protecting Computers From a Malicious DHCP Server" on page 117 for more information.

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.
- 7 If you selected the DHCP binding protocol for NetInfo in step 4, make sure the DHCP server is configured to provide the address and NetInfo tag of the shared NetInfo domain.
- 8 If you selected the broadcast binding protocol for NetInfo in step 4, make sure the parent NetInfo domain has a machine record for the computer you’re configuring. For instructions, see “Adding a Machine Record to a Parent NetInfo Domain” (next).

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.

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.

Your ongoing tasks in managing and troubleshooting Open Directory services may include the following:

- “Controlling Access to Open Directory Servers” (next)
- “Monitoring Open Directory” on page 159
- “Directly Viewing and Editing Directory Data” on page 161
- “Importing Records of Any Type” on page 164
- “Managing Open Directory Replication” on page 164
- “Archiving an Open Directory Master” on page 167
- “Restoring an Open Directory Master” on page 168
- “Solving Open Directory Master and Replica Problems” on page 169
- “Solving Directory Access Problems” on page 170
- “Solving Authentication Problems” on page 171

Controlling Access to Open Directory Servers

You can control access to an Open Directory master or replica by restricting who can log in using the login window or the `ssh` command-line tool. For instructions, see:

- “Controlling Access to a Server’s Login Window” on page 157
- “Controlling Access to SSH Service” on page 158

Controlling Access to a Server’s Login Window

You can use Server Admin to control which users can log in to Mac OS X Server using the login window. Users with server administrator privileges are always allowed to log in to the server.

To restrict who can use the login window on a server:

- 1 Open Server Admin, connect to the server on which you want to control login window access, and select the server in the Computers & Services list.
Select the server, not a service under the server.
- 2 Click Settings, then click Access.
- 3 Deselect “Use same access for all services” and select Login Window in the list on the left.
- 4 Select “Allow only users and groups below” and edit the list of users and groups that you want to allow to log in using the server’s login window
 - Add users or groups that can use the login window by clicking the Add button (+) and supplying the requested information.
 - Remove users or groups from the list by selecting one or more and clicking the Remove button (–).
- 5 Click Save.

If “Allow all users and groups” is selected when you deselect “Use same access for all services” in step 3, all services except login window will allow access to all users and groups. If you want to restrict who can access a listed service besides login window, select the service in the list, select “Allow only users and groups below,” and add to the list of allowed users and groups.

If you want all users to be able to log in using the server’s login window, select Login Window, then select “Allow all users and groups.”

Controlling Access to SSH Service

You can use Server Admin to control which users can open a command-line connection to Mac OS X Server using the `ssh` command in Terminal. Users with server administrator privileges are always allowed to open a connection using `ssh`. The `ssh` command uses the Secure Shell (SSH) service. For information on using the `ssh` command, see the command-line administration guide.

To restrict who can open an SSH connection to a remote server:

- 1 Open Server Admin, connect to the server for which you want to control SSH access, and select the server in the Computers & Services list.
Select the server, not a service under the server.
- 2 Click Settings, then click Access.
- 3 Deselect “Use same access for all services” and select SSH in the list on the left.

- 4 Select “Allow only users and groups below” and edit the list of users and groups that you want to allow to open an SSH connection to the server.
 - Add users or groups that can open SSH connections by clicking the Add button (+) and supplying the requested information.
 - Remove users or groups from the list by selecting one or more and clicking the Remove button (-).
- 5 Click Save.

If “Allow all users and groups” is selected when you deselect “Use same access for all services” in step 3, all services except SSH will allow access to all users and groups. If you want to restrict who can access a listed service besides SSH, select the service in the list, select “Allow only users and groups below,” and add to the list of allowed users and groups.

If you want all users to be able to open an SSH connection to the server, select SSH, then select “Allow all users and groups.”

Monitoring Open Directory

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

For task instructions, see:

- “Checking the Status of an Open Directory Master or Replica” (next)
- “Monitoring Replicas of an Open Directory Master” on page 160
- “Viewing Open Directory Status and Logs” on page 160
- “Monitoring Open Directory Authentication” on page 160

Checking the Status of an Open Directory Master or Replica

You can confirm that the Open Directory master is functioning properly.

- 1 Open Server Admin and select Open Directory for a server in the Computers & Services list.
- 2 Click Overview (near the bottom of the window).
- 3 Make sure the status of all the items listed in the Open Directory overview pane is “Running.”

If any of the listed items is stopped, click Refresh (or choose View > Refresh). If Kerberos remains stopped, see “Kerberos is Stopped on an Open Directory Master or Replica” on page 169.

Monitoring Replicas of an Open Directory Master

Using Server Admin, you can check the status of replica creation and of ongoing replication.

To monitor replicas of an Open Directory master:

- 1 Open Server Admin and select Open Directory for the master in the Computers & Services list.
- 2 Click Settings to see a list of replicas and the status of each one.

The status for a new replica indicates whether it was created successfully. Thereafter, the status indicates whether the most recent replication attempt was successful.

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
- `kadmin` log
- `kdc` log
- `lookupd` log
- NetInfo log
- LDAP log
- Password service server log
- Password service error log
- Password service replication log
- `slapconfig` 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.
The path to the log file is displayed beneath the pop-up menu.
- 4 Optionally, enter some text in the Filter field and press return to show only lines containing the text you entered.

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 the `kdcc` log or 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. Furthermore, the Inspector allows you to edit directory data that you cannot otherwise change in Workgroup Manager. For example, you can use the Inspector to change a user's first short name.

For instructions, see:

- “Showing the Directory Inspector” on page 161
- “Hiding the Directory Inspector” on page 162
- “Changing a User's Short Name” on page 162
- “Deleting Records” on page 163

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.

Warning: 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 To see user, group, or computer list attributes, click the Users button, Group button, or Computer Lists button (on the left), then click Inspector (on the right).
- 4 To see other types of records, click the All Records button, which is next to the Computer Lists 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.

Warning: 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.

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.

Setting Directory Access Controls (DACs)

Open Directory provides the ability to define directory access controls (DACs) to all parts of the LDAP directory, providing fine-grained control of who has permission to modify what. Open Directory stores the DACs in an apple-acl record that you can edit using the Inspector in Workgroup Manager.

To change the directory access controls:

- 1 Open Workgroup Manager and make the Inspector visible if it is hidden.
- 2 Open the directory domain whose access controls you want to set, 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.

- 3 Click the All Records button (next to the Computer Lists button), then choose AccessControls from the pop-up menu at the top of the list.
- 4 Select “default” in the list of records.
- 5 Locate AccessControlEntry in the list of attributes, and if a triangle appears next to AccessControlEntry, click the triangle to see all access control entries.
- 6 Select an AccessControlEntry, then click Edit to change the value or click New Value to add an AccessControlEntry value.

You can also double-click a value to edit it in place.

- 7 Click Save.

Deleting Records

You can use the Inspector in Workgroup Manager to delete any kind of record.

Warning: Deleting records can cause the server to behave erratically or stop working. Don't delete any records unless you know they're not needed for proper server functioning.

To delete records with the Inspector:

- 1 Open Workgroup Manager and make the Inspector visible if it is hidden.
- 2 Open the directory domain in which you want to delete a record, 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.

- 3 Click the All Records button (next to the Computer Lists button), then choose the record type of interest from the pop-up menu at the top of the list.
- 4 Select one or more records that you want to delete in the list of records.
- 5 Click Delete (or choose Server > Delete Selected Records).

Importing Records of Any Type

Workgroup Manager can import all types of records into the LDAP directory of an Open Directory master. This includes users, groups, computer lists, computers, and all the other standard Mac OS X record types. For information on well-known record types and attributes, see “Standard Open Directory Record Types and Attributes” on page 216.

For a list of record types and attributes that can be imported, see the following file:

```
/System/Library/Frameworks/DirectoryService.framework/Headers/DirServicesConst.h
```

For instructions on importing records of any type, see the user management guide.

Managing Open Directory Replication

You can schedule Open Directory replication or replicate on demand, promote a replica to a master, or take a replica out of service. For instructions, see:

- “Scheduling Replication of an Open Directory Master” (next)
- “Synchronizing an Open Directory Replica on Demand” on page 165
- “Promoting an Open Directory Replica” on page 165
- “Decommissioning an Open Directory Replica” on page 166

Scheduling Replication 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.

Synchronizing an Open Directory Replica on Demand

Although an Open Directory master automatically synchronizes its directory and authentication data with registered replicas, you can use Server Admin to synchronize the data with a selected replica on demand.

To synchronize an Open Directory replica on demand:

- 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 Select a replica in the list, then click Replicate Now.

Promoting an Open Directory Replica

If an Open Directory master fails and you cannot recover it from a backup, you can promote a replica to be a master. The promoted master uses the existing directory and authentication databases of the replica. After doing this, you must convert all other replicas of the old master to standalone servers and then make them replicas of the new master.

Important: Use this procedure only to replace an Open Directory master with its replica. If you want to keep the Open Directory master in operation and make its replica another master, do not use this procedure. Instead, decommission the replica and then make it a master as described in “Decommissioning an Open Directory Replica” on page 166 and “Setting Up an Open Directory Master” on page 75.

To promote an Open Directory Replica:

- 1 In Server Admin, connect to the replica that you want to promote to be a master and select Open Directory for this server in the Computers & Services list.
- 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.
 - “Domain administrator’s short name:” The short name of an administrator of the server’s LDAP directory domain.
 - “Domain administrator’s password:” The password for the administrator account whose short name you entered.
- 4 Click OK, then click Save.
- 5 In Server Admin, connect to another replica of the old master and select Open Directory for this server in the Computers & Services list.
- 6 Click Settings, then click General.
- 7 Choose Standalone from the Role pop-up menu, then click Save.

- 8 Choose Open Directory Replica from the Role pop-up menu and enter the requested information.
 - *“IP address of Open Directory master:”* Enter the IP address of the server that is the new Open Directory master.
 - *“Root password on Open Directory master:”* Enter the password of the new Open Directory master system’s root user (user name System Administrator).
 - *“Domain administrator’s short name on master:”* Enter the name of an LDAP directory domain administrator.
 - *“Domain administrator’s password on master:”* Enter the password of the domain administrator whose name you entered.
- 9 Click OK, then click Save.
- 10 Repeat steps 5 – 9 for each additional replica of the old master.
- 11 Make sure the date, time, and time zone are correct on the replicas and the master.

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

If other computers were connected to the old Open Directory master’s LDAP directory, you need to reconfigure their connections to use the new master’s LDAP directory:

- Each Mac OS X and Mac OS X Server computer with a custom search policy that included the old master’s LDAP directory must be reconfigured to connect to the new master’s LDAP directory instead. Use the Services and Authentication panes of Directory Access. For instructions, see “Deleting a Configuration for Accessing an LDAP Directory” on page 126, and “Configuring Access to an LDAP Directory” on page 120.
- If DHCP service provided the old master’s LDAP URL to computers with automatic search policies, you need to reconfigure DHCP service to provide the new master’s LDAP URL instead. Mac OS X and Mac OS X Server computers with automatic search policies require no reconfiguration; they’ll get the correct LDAP URL from the updated DHCP service the next time they start up. See the DHCP chapter of the network services administration guide.

Decommissioning an Open Directory Replica

You can take an Open Directory replica out of service by making it a standalone server or connecting it to another system for directory and authentication services.

To decommission an Open Directory replica:

- 1 Verify that the network connection is working between the Open Directory master and the replica that you want to decommission.

Port 389 or 636 needs to be open between master and replica while decommissioning the replica. LDAP uses port 389 if SSL is disabled or port 636 if SSL is enabled on the master. (Port 22, used for SSH, does not need to be open to decommission a replica.)

Important: If you decommission a replica while there is no network connectivity between it and the master, the decommissioned replica will remain in the master's list of replicas. Furthermore, the master will try to replicate to the decommissioned replica as specified in the General settings pane for Open Directory service on the master server.

- 2 In Server Admin, connect to the replica that you want to decommission and select Open Directory for this server in the Computers & Services list.
- 3 Click Settings (near the bottom of the window), then click General (near the top).
- 4 Click the Role pop-up menu and choose Standalone Server or "Connected to a Directory System" and enter the requested information.
 - "Domain administrator's short name:" The short name of an administrator of the Open Directory master's LDAP directory.
 - "Domain administrator's password:" The password for the administrator account whose short name you entered.
 - "Root password on Open Directory master:" the password of the Open Directory master system's root user (user name System Administrator).
- 5 Click OK, then click Save.

Assuming there is network connection between the Open Directory master and the replica, the master is updated to no longer connect to the replica.

- 6 If you chose "Connected to a Directory System" from the role pop-up menu, 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."

Archiving an Open Directory Master

You can use Server Admin to archive a copy of an Open Directory master's directory and authentication data. You can archive a copy of the data while the Open Directory master is in service.

The following files are archived:

- LDAP directory database and configuration files
- Open Directory Password Server database
- Kerberos database and configuration files
- Local NetInfo domain and shadow password database

If you have a reliable archive of an Open Directory master, you effectively have an archive of all its replicas. If a replica develops a problem, you can just change its Open Directory 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.

Important: Carefully safeguard the archive media that contains a copy of the Open Directory Password database, the Kerberos database, and the Kerberos keytab file. The archive 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 archive media should be just as stringent as for the Open Directory master server.

To archive an Open Directory master:

- 1 Open Server Admin and in the Computers & Services list, select Open Directory for an Open Directory master server.
- 2 Click Archive (at the bottom of the window).
- 3 Enter the path to the folder where you want the Open Directory data archived, then click the Archive button.

You can type the folder path or click the Browse button (***) to select it.

- 4 Enter a name and password to use in encrypting the archive, then click OK.

Restoring an Open Directory Master

You can use Server Admin to restore an Open Directory master's directory and authentication data from an archive. The following files are restored:

- LDAP directory database and configuration files
- Open Directory Password Server database
- Kerberos database and configuration files
- Local NetInfo domain and shadow password database

Restoring an archive to an Open Directory standalone server makes it an Open Directory master with the same data as the master from which the archive was created.

Restoring an archive to an Open Directory master server merges the archive data with the existing master's data. If conflicts are encountered during the merge operation, the existing record takes precedence over the one in the archive; the archive record is ignored. Conflicts are recorded in the slapconfig log file (/Library/Logs/slapconfig.log), which you can view using Server Admin. See "Viewing Open Directory Status and Logs" on page 160.

Instead of restoring an Open Directory master from an archive, you may get better results by promoting a replica to be the master. The replica may have more recent directory and authentication data than the archive.

After restoring an Open Directory master from an archive, you must re-create your Open Directory replicas.

Important: Restoring an archive shouldn't be used as a means of porting directory and authentication data from one system into another. Instead, export from the source directory and import into the target directory. See the user management guide for more information on exporting and importing directory data.

To restore an Open Directory master from an archive:

- 1 Open Server Admin and in the Computers & Services list, select Open Directory for an Open Directory standalone server or an Open Directory master server.

If you select an Open Directory standalone server as the target of a restore operation, the server will become an Open Directory master with the directory and authentication data contained in the archive.

If you select an Open Directory master server as the target of a restore operation, the directory and authentication data from the archive is merged with the directory and authentication data of the target. The target server must have the same Kerberos realm name as the master from which the archive was created.

- 2 Click Archive (at the bottom of the window).
- 3 Enter the path to the Open Directory archive file, then click the Restore button.
You can type the path or click the Browse button (•••) to select the archive file.
- 4 Enter the password that was used to encrypt the archive when it was created, then click OK.
- 5 Convert all existing Open Directory replica server to Open Directory standalone servers and then make them replicas of the new master.

See "Setting Up a Standalone Server" on page 73 and "Setting Up an Open Directory Replica" on page 77 for instructions.

Solving Open Directory Master and Replica Problems Kerberos is Stopped on an Open Directory Master or Replica

An Open Directory master requires properly configured DNS so it can provide single sign-on Kerberos authentication.

To confirm that DNS is configured correctly for Kerberos:

- 1 Make sure DNS service is configured to resolve fully qualified DNS names and provide corresponding reverse lookups.

DNS must resolve the fully qualified DNS name and provide reverse lookups for the Open Directory master server, all replica servers, and other servers that are members of the Kerberos realm.

You can use the Lookup pane of Network Utility (in /Applications/Utilities/) to do a DNS lookup of a server's DNS name and a reverse lookup of the server's IP address.

For instructions on setting up DNS service, see the network services administration guide.

- 2 Make sure the Open Directory master server's host name is the correct fully qualified DNS name, not the server's local hostname.

For example, the host name might be `ods.example.com` but should not be `ods.local`. You can see the host name by opening Terminal, typing `hostname`, and pressing Return.

If the Open Directory server's host name isn't its fully qualified DNS name, try temporarily clearing the list of DNS servers and clicking Apply in the Open Directory server's Network preferences. Then re-enter one or more DNS server IP addresses, starting with the primary DNS server that resolves the Open Directory server's name, and clicking Apply again in Network Preferences.

If the Open Directory server's host name still isn't its fully qualified DNS name, try restarting the server.

- 3 Make sure the Open Directory master server's Network preferences is configured to use the DNS server that resolves the server's name.

If the Open Directory master server provides its own DNS service, the server's Network preferences must be configured to use itself as a DNS server.

After confirming the correct DNS configuration for the server, you can try starting Kerberos. See "Starting Kerberos After Setting Up an Open Directory Master" on page 82.

Can't Create an Open Directory Replica

If you try to create two replicas simultaneously, one attempt will succeed and the other will fail. A subsequent attempt to establish the second replica should succeed. If you still can't create the second replica, go to folder `/var/run/`, look for the file `slapconfig.lock`, and remove it if it exists. Alternatively, you can restart the server.

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.

- A pause during startup is normal when you disconnect a portable computer from the network that the LDAP server is connected to.
- 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.

You Can't Modify a User's Open Directory Password

To modify the password of a user whose password type is Open Directory, you must be an administrator of the directory domain in which the user's record resides. In addition, your user account must have a password type of Open Directory.

The user account specified when the Open Directory master was set up (using Server Assistant or the Open Directory service settings in Server Admin) normally has an Open Directory password. This account can be used to set up other user accounts as directory domain administrators with Open Directory passwords.

If all else fails, try using the root user account to set up a user account as directory administrator with an Open Directory password. (The root user account's name is "root" and the password is usually the same as the password first given to the administrator account created during initial server setup.)

A User Can't Access Some Services

If a user can access some services that require authentication but not others, try temporarily changing the user's password to a simple sequence of characters, such as "password." If this solves the problem, then the user's previous password contained characters that were not allowed by all services. For example, some services allow spaces in passwords while others don't.

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 v10.3–10.4. 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 v10.3–10.4. Alternatively, you can upgrade the older server to Mac OS X Server v10.4 or later.

You Can't Change a User's Password Type to Open Directory

To change a user's password type to 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. The user account specified when the Open Directory master was set up (using Server Assistant or the Open Directory service settings in Server Admin) has an Open Directory password. This account can be used to set up other user accounts as directory domain administrators with Open Directory passwords.

Users Relying on a Password Server Can't Log In

If your network has a server with Mac OS X Server version 10.2, it could be configured to get authentication from an Open Directory Password Server hosted by another server. If the Password Server's computer becomes disconnected from your network, for example because you unplug the cable from the computer's Ethernet port, users whose passwords are validated using the Password Server can't log in because its IP address isn't accessible.

Users can log in to Mac OS X Server if you reconnect the Password Server's computer to the network. Alternatively, while the Password Server's computer is offline, users can log in with user accounts whose password type is crypt password or shadow password.

Users Can't Log In With Accounts in a Shared Directory Domain

Users can't log in using accounts in a shared directory domain if the server hosting the directory isn't accessible. A server may become inaccessible due to a problem with the network, the server software, or the server hardware. Problems with the server hardware or software affect users trying to log in to Mac OS X computers and users trying to log in to the Windows domain of a Mac OS X Server PDC. Network problems may affect some users but not others, depending on where the network problem is.

Users with mobile user accounts can still log in to the Mac OS X computers they used previously. And users affected by these problems can log in by using a local user account defined on the computer, such as the user account created during initial setup after installing Mac OS X.

Can't Log In as Active Directory User

After configuring a connection to an Active Directory domain in the Service pane of Directory Access and adding it to a custom search policy in the Authentication pane, you need to wait 10 or 15 seconds for the change to take effect. Attempts to log in immediately with an Active Directory account will be unsuccessful.

Users Can't Authenticate Using Single Sign-On or Kerberos

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

- Kerberos authentication is based on encrypted time stamps. If there's more than a 5-minute difference between the KDC, client, and service computers, authentication may fail. Make sure that the clocks for all computers are synchronized using the Network Time Protocol (NTP) service of Mac OS X Server or another network time server. For information about the NTP service of Mac OS X Server, see the network services administration guide.
- Make sure that Kerberos is running on the Open Directory master and replicas. See "Kerberos is Stopped on an Open Directory Master or Replica" on page 169.
- 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 the Kerberos server's directory domain, and make sure this directory domain contains the accounts for users who are trying to authenticate using Kerberos. For information about configuring access to directory domains, see Chapter 7, "Managing Directory Access."
- For an Open Directory server's Kerberos realm, make sure the client's computer is configured to access the Open Directory server's LDAP directory using the correct search base suffix. The client's LDAPv3 search base suffix setting must match the LDAP directory's search base setting. The client's LDAPv3 search base suffix can be blank if it gets its LDAP mappings from the server, in which case the client uses the LDAP directory's default search base suffix.
 - To check the client's search base suffix setting, open Directory Access, show the list of LDAPv3 configurations, and choose the item from the LDAP Mappings pop-up menu that's already selected in the menu. For instructions, see "Changing a Configuration for Accessing an LDAP Directory" on page 124.
 - To check the LDAP directory's search base setting, open Server Admin and look in the Protocols pane of the Settings pane for Open Directory service.
- Refer to the KDC log for information that can help you solve problems. See "Viewing Open Directory Status and Logs" on page 160.
- If Kerberos was not running when user records were created, imported, or updated from an earlier Mac OS X version, they may not be enabled for Kerberos authentication.
 - A record isn't enabled for Kerberos if its authentication authority attribute lacks a ;Kerberosv5; value. You can use the Inspector in Workgroup Manager to see the value or values of a user record's authentication authority attribute.
 - You can enable Kerberos for a user record by changing its password type. First set the password type to Crypt Password, then set it to Open Directory. For detailed instructions, see "Changing the Password Type to Crypt Password" on page 97 and "Changing the Password Type to Open Directory" on page 96.

- If users can't authenticate using single sign-on or Kerberos for services provided by a server that is joined to an Open Directory master's Kerberos realm, the server's computer record might be incorrectly configured in the Open Directory master's LDAP directory. In particular, the server's name in the computer list account must be the server's fully qualified DNS name, not just the server's host name. For example, the name could be server2.example.com but not just server2.

To reconfigure a server's computer record for single sign-on Kerberos authentication:

- 1 Delete the server from the computer list account in the LDAP directory.
For instructions for this and the next step, see the user management guide.
- 2 Add the server to the computer list again.
- 3 Delegate authority again for joining the server to the Open Directory master's Kerberos realm.
For instructions, see "Delegating Authority to Join an Open Directory Kerberos Realm" on page 83.
- 4 Rejoin the server to the Open Directory Kerberos realm.
For instructions, see "Joining a Server to a Kerberos Realm" on page 85.

Users Can't Change Their Passwords

Users whose accounts reside in an LDAP directory not hosted by Mac OS X Server and have a password type of Crypt Password cannot change their passwords after logging in from a client computer with Mac OS X version 10.3. These users can change their passwords if you use Workgroup Manager's Advanced pane to change their accounts' User Password Type setting to Open Directory. When you make this change, you must also enter a new password. Then you should instruct users to log in using this new password and change it in the Accounts pane of System Preferences.

Can't Join a Server to an Open Directory Kerberos Realm

If a user with delegated Kerberos authority can't join a server to an Open Directory master's Kerberos realm, the server's computer record may be incorrectly configured in the Open Directory master's LDAP directory. In particular, the server's address in the computer list account must be the server's primary Ethernet address. The primary Ethernet address is the Ethernet ID of the first Ethernet port in the list of network port configurations shown in the server's Network preferences pane.

To reconfigure a server's computer record for joining a Kerberos realm:

- 1 Delete the server from the computer list account in the LDAP directory.
For instructions for this and the next step, see the user management guide.
- 2 Add the server to the computer list again.

- 3 Delegate authority again for joining the server to the Open Directory master's Kerberos realm.

You can skip this step if you can use a Kerberos administrator account (LDAP directory administrator account) to rejoin the server to the Kerberos realm. Otherwise, see "Delegating Authority to Join an Open Directory Kerberos Realm" on page 83 for instructions.

- 4 Rejoin the server to the Open Directory Kerberos realm.

For instructions, see "Joining a Server to a Kerberos Realm" on page 85.

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

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.

This appendix lists 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. Use this information for:

- Mapping object classes and attributes of non-Apple LDAP directories or Active Directory domains to Open Directory record types and attributes, as described in “Configuring LDAP Searches and Mappings” on page 129.
- Importing or exporting user or group accounts to an Open Directory domain, as described in the user management guide.
- Working in Workgroup Manager’s Inspector pane, as described in “Directly Viewing and Editing Directory Data” on page 161.

For details, see:

- Open Directory Extensions to LDAP Schema (p. 178)
 - Object Classes in Open Directory LDAP Schema (p. 178)
 - Attributes in Open Directory LDAP Schema (p. 185)
- Mapping Standard Record Types and Attributes to LDAP and Active Directory (p. 201)
 - Mappings for Users (p. 201)
 - Mappings for Groups (p. 205)
 - Mappings for Mounts (p. 206)
 - Mappings for Computers (p. 207)
 - Mappings for ComputerLists (p. 208)
 - Mappings for Config (p. 209)
 - Mappings for People (p. 210)
 - Mappings for PresetComputerLists (p. 211)
 - Mappings for PresetGroups (p. 212)
 - Mappings for PresetUsers (p. 213)
 - Mappings for Printers (p. 214)

- Mappings for AutoServerSetup (p. 215)
- Mappings for Locations (p. 216)
- Standard Attributes in User Records (p. 217)
 - User Data That Mac OS X Server Uses (p. 221)
- Standard Attributes in Group Records (p. 222)
- Standard Attributes in Computer Records (p. 223)
- Standard Attributes in Computer List Records (p. 224)
- Standard Attributes in Mount Records (p. 224)
- Standard Attributes in Config Records (p. 225)

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”

LDAP schema definitions specify syntax identifiers and matching rules that are defined in:

- RFC 2252 “LDAPv3 Attributes”

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:

- “Object Classes in Open Directory LDAP Schema” (next)
- “Attributes in Open Directory LDAP Schema” on page 185

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 ) )
```

Time to Live Object Class

```
attributetype (  
  1.3.6.1.4.1.250.1.60  
  NAME 'ttl'  
  EQUALITY integerMatch  
  SYNTAX '1.3.6.1.4.1.1466.115.121.1.27' SINGLE-VALUE )
```

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 kDSTdRecordTypeUsers 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 $ apple-imhandle $  
    apple-webloguri $  
    authAuthority $ acctFlags $ pwdLastSet $ logonTime $  
    logoffTime $ kickoffTime $ homeDrive $ scriptPath $  
    profilePath $ userWorkstations $ smbHome $ rid $  
    primaryGroupID $ sambaSID $ sambaPrimaryGroupSID $  
    userCertificate ) )
```

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 kDSTdRecordTypeGroups 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 $
        apple-group-nestedgroup $
        apple-group-memberguid $
        mail $
        rid $
        sambaSID $
        ttl ) )
```

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 $
        apple-machine-contactperson ) )
```

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-category $
        apple-computer-list-groups $
        apple-keyword $
        apple-mcxflags $
        apple-mcxsettings $
        apple-networkview $
        apple-xmlplist $
        apple-service-url $
        authAuthority $
        uidNumber $ gidNumber $ apple-generateduid $ ttl $
        acctFlags $ pwdLastSet $ logonTime $
        logoffTime $ kickoffTime $ rid $ primaryGroupID $
        sambaSID $ sambaPrimaryGroupSID ) )
```

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-generateduid $
        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 $
        ttl ) )
```

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-computer-list-groups $
        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 (  memberId $
      gidNumber $
      apple-group-homeurl $
      apple-group-homeowner $
      apple-mcxflags $
      apple-mcxsettings $
      apple-group-realname $
      apple-keyword $
      apple-group-nestedgroup $
      apple-group-memberguid $
      ttl ) )

```

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 $
        memberId $
        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 $
        description $
        shadowLastChange $
        shadowExpire $
        authAuthority $
        homeDrive $ scriptPath $ profilePath $ smbHome $

```

```
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 ) )
```

Service Object Class

```
objectclass (
  1.3.6.1.4.1.63.1000.1.1.2.19
  NAME 'apple-service'
  SUP top STRUCTURAL
  MUST ( cn $
        apple-service-type )
  MAY ( ipHostNumber $
        description $
        apple-service-location $
        apple-service-url $
        apple-service-port $
        apple-dnsname $
        apple-keyword ) )
```

Neighborhood Object Class

```
objectclass (
  1.3.6.1.4.1.63.1000.1.1.2.20
  NAME 'apple-neighborhood'
```



```

SUP top STRUCTURAL
MUST ( cn )
MAY ( description $
      apple-generateduid $
      apple-category $
      apple-nodexml $
      apple-neighborhoodalias $
      apple-computeraliases $
      apple-keyword $
      apple-realname $
      apple-xmlplist $
      ttl ) )

```

ACL Object Class

```

objectclass (
  1.3.6.1.4.1.63.1000.1.1.2.21
  NAME 'apple-acl'
  SUP top STRUCTURAL
  MUST ( cn $
        apple-acl-entry ) )

```

Attributes in Open Directory LDAP Schema

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

Time-to-Live Attribute

```

objectclass (
  1.3.6.1.4.1.250.3.18
  NAME 'cacheObject'
  AUXILIARY
  SUP top
  DESC 'Auxiliary object class to hold TTL caching information'
  MAY ( ttl ) )

```

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-imhandle

```
attributetype (
    1.3.6.1.4.1.63.1000.1.1.1.1.21
    NAME ( 'apple-imhandle' )
    DESC 'IM handle (service:account name)'
    EQUALITY caseExactMatch
    SUBSTR caseExactSubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

apple-webloguri

```
attributetype (
    1.3.6.1.4.1.63.1000.1.1.1.1.22
    NAME ( 'apple-webloguri' )
    DESC 'Weblog URI'
    EQUALITY caseExactIA5Match
    SUBSTR caseExactIA5SubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
```

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 homeDirectory attribute different 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-nestedgroup

```
attributetype (  
  1.3.6.1.4.1.63.1000.1.1.1.14.6  
  NAME 'apple-group-nestedgroup'
```

```
DESC 'group real name'  
EQUALITY caseExactMatch  
SUBSTR caseExactSubstringsMatch  
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

apple-group-memberguid

```
attributetype (  
  1.3.6.1.4.1.63.1000.1.1.1.14.7  
  NAME 'apple-group-memberguid'  
  DESC 'group real name'  
  EQUALITY caseExactMatch  
  SUBSTR caseExactSubstringsMatch  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

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 )
```

apple-machine-contactperson

```
attributeType (  
    1.3.6.1.4.1.63.1000.1.1.1.3.12  
    NAME 'apple-machine-contactperson'  
    DESC 'Name of contact person/owner of this machine'  
    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 )
```

apple-mount-name

```
# Alternative to using 'cn' when adding mount record schema to other
  LDAP servers
#attributetype (
# 1.3.6.1.4.1.63.1000.1.1.1.8.100
# NAME ( 'apple-mount-name' )
# DESC 'mount name'
# SUP name )
```

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 (  
  1.3.6.1.4.1.63.1000.1.1.1.9.6
```

```
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 )
```

apple-networkview

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.10.3
  NAME 'apple-networkview'
  DESC 'Network view for the computer'
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

apple-category

```
attributetype (
  1.3.6.1.4.1.63.1000.1.1.1.10.4
  NAME 'apple-category'
  DESC 'Category for the computer or neighborhood'
  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 )
```

XML Plist Attribute

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 )
```

Service URL Attribute

apple-service-url

```
attributetype (
    1.3.6.1.4.1.63.1000.1.1.1.19.2
    NAME 'apple-service-url'
    DESC 'URL of service'
    EQUALITY caseExactIA5Match
    SUBSTR caseExactIA5SubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
```

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 Attributes

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 )
```

authAuthority2

```
#attributetype (
# 1.3.6.1.4.1.63.1000.1.1.2.16.2
# NAME ( 'authAuthority' 'authAuthority2' )
# DESC 'password server authentication authority'
# EQUALITY caseExactMatch
# SUBSTR caseExactSubstringsMatch
# SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

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 )
```

Service Attributes

apple-service-type

```
attributetype (  
    1.3.6.1.4.1.63.1000.1.1.1.19.1  
    NAME 'apple-service-type'  
    DESC 'type of service'  
    EQUALITY caseExactIA5Match  
    SUBSTR caseExactIA5SubstringsMatch  
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
```

apple-service-url

```
#attributetype (  
# 1.3.6.1.4.1.63.1000.1.1.1.19.2  
# NAME 'apple-service-url'  
# DESC 'URL of service'  
# EQUALITY caseExactIA5Match  
# SUBSTR caseExactIA5SubstringsMatch  
# SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
```

apple-service-port

```
attributetype (  
    1.3.6.1.4.1.63.1000.1.1.1.19.3  
    NAME 'apple-service-port'  
    DESC 'Service port number'  
    EQUALITY integerMatch  
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 )
```

apple-dnsname

```
attributetype (  
    1.3.6.1.4.1.63.1000.1.1.1.19.4  
    NAME 'apple-dnsname'  
    DESC 'DNS name'  
    EQUALITY caseExactMatch  
    SUBSTR caseExactSubstringsMatch
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

apple-service-location

```
attributetype (  
  1.3.6.1.4.1.63.1000.1.1.1.19.5  
  NAME 'apple-service-location'  
  DESC 'Service location'  
  EQUALITY caseExactMatch  
  SUBSTR caseExactSubstringsMatch  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

Neighborhood Attributes

apple-nodepathxml

```
attributetype (  
  1.3.6.1.4.1.63.1000.1.1.1.20.1  
  NAME 'apple-nodepathxml'  
  DESC 'XML plist of directory node path'  
  EQUALITY caseExactMatch  
  SUBSTR caseExactSubstringsMatch  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

apple-neighborhoodalias

```
attributetype (  
  1.3.6.1.4.1.63.1000.1.1.1.20.2  
  NAME 'apple-neighborhoodalias'  
  DESC 'XML plist referring to another neighborhood record'  
  EQUALITY caseExactMatch  
  SUBSTR caseExactSubstringsMatch  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

apple-computeraliases

```
attributetype (  
  1.3.6.1.4.1.63.1000.1.1.1.20.3  
  NAME 'apple-computeraliases'  
  DESC 'XML plist referring to a computer record'  
  EQUALITY caseExactMatch  
  SUBSTR caseExactSubstringsMatch  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

ACL Attribute

apple-acl-entry

```
#attributetype (  
# 1.3.6.1.4.1.63.1000.1.1.1.21.1  
# NAME 'apple-acl-entry'
```



```
# DESC 'acl entry'
# EQUALITY caseExactMatch
# SUBSTR caseExactSubstringsMatch
# SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

Schema Attributes

apple-apple-attributeTypesConfig

```
#attributetype (
# 1.3.6.1.4.1.63.1000.1.1.1.22.1
# NAME 'attributeTypesConfig'
# DESC 'attribute type configuration'
# EQUALITY objectIdentifierFirstComponentMatch
# SYNTAX 1.3.6.1.4.1.1466.115.121.1.3 )
```

apple-objectClassesConfig

```
#attributetype (
# 1.3.6.1.4.1.63.1000.1.1.1.22.2
# NAME 'objectClassesConfig'
# DESC 'object class configuration'
# EQUALITY objectIdentifierFirstComponentMatch
# SYNTAX 1.3.6.1.4.1.1466.115.121.1.37 )
```

Mapping Standard Record Types and Attributes to LDAP and Active Directory

This section specifies how Open Directory record types and attributes map to LDAP object classes and attributes. It also specifies how the Active Directory object categories and attributes are mapped to and generated from Open Directory record types and attributes.

Mappings for Users

The following tables specify how 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	

Open Directory name, RFC/class	LDAP object class name OID	Active Directory plug-in
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 1.3.6.1.4.1.63.1000.1.1.1.1.6	Generated from homeDirectory
HomeDirectoryQuota, Apple registered	apple-user-homequota 1.3.6.1.4.1.63.1000.1.1.1.1.8	Apple extended schema
HomeDirectorySoftQuota, Apple registered	apple-user-homesoftquota 1.3.6.1.4.1.63.1000.1.1.1.1.17	Apple extended schema
MailAttribute, Apple registered	apple-user-mailattribute 1.3.6.1.4.1.63.1000.1.1.1.1.9	Apple extended schema
PrintServiceUserData, Apple registered	apple-user-printattribute 1.3.6.1.4.1.63.1000.1.1.1.1.13	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
AdminLimits, Apple registered	apple-user-adminlimits 1.3.6.1.4.1.63.1000.1.1.1.1.14	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 1.3.6.1.4.1.63.1000.1.1.1.1.15	Apple extended schema
PasswordPolicyOptions, Apple registered	apple-user-passwordpolicy 1.3.6.1.4.1.63.1000.1.1.1.1.18	Apple extended schema
Keywords, Apple registered	apple-keyword 1.3.6.1.4.1.63.1000.1.1.1.1.19	Apple extended schema
Picture, Apple registered	apple-user-picture 1.3.6.1.4.1.63.1000.1.1.1.1.12	Apple extended schema
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, sAMAccountName
RecordName, RFC 1274	uid 0.9.2342.19200300.100.1.1	N/A

Open Directory name, RFC/class, special purpose	LDAP attribute name OID	Active Directory plug-in
EEmailAddress, RFC 1274	mail 0.9.2342.19200300.100.13	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 1.3.6.1.1.1.4	Extended using RFC
Change, RFC 2307	shadowLastChange 1.3.6.1.1.1.5	No mapping
Expire, RFC 2307	shadowExpire 1.3.6.1.1.1.10	No mapping
UniqueID, RFC 2307	uidNumber 1.3.6.1.1.1.0	Generated from GUID
NFSHomeDirectory, RFC 2307	homeDirectory 1.3.6.1.1.1.3	Generated from homeDirectory
PrimaryGroupID, RFC 2307	gidNumber 1.3.6.1.1.1.1	Extended using RFC or generated from GUID
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)

Open Directory name, RFC/class, special purpose	LDAP attribute name OID	Active Directory plug-in
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
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

Open Directory name, RFC/class, special purpose	LDAP attribute name OID	Active Directory plug-in
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 1.3.6.1.4.1.63.1000.1.1.1.110	Apple extended schema
MCXSettings, Apple registered	apple-mcxsettings 1.3.6.1.4.1.63.1000.1.1.1.116	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)

Open Directory name, RFC/class	LDAP attribute name OID	Active Directory plug-in
Picture, Apple registered	apple-user-picture 1.3.6.1.4.1.63.1000.1.1.1.12	Apple extended schema
Keywords, Apple registered	apple-keyword 1.3.6.1.4.1.63.1000.1.1.1.19	Apple extended schema
GeneratedUID, Apple registered	apple-generateduid 1.3.6.1.4.1.63.1000.1.1.1.20	From GUID—formatted
GroupMembership, RFC 2307	memberUid 1.3.6.1.1.1.12	Generated from member
Member, RFC 2307	memberUid 1.3.6.1.1.1.12	Same as GroupMembership
PrimaryGroupID, RFC 2307	gidNumber 1.3.6.1.1.1.11	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 1.3.6.1.4.1.63.1000.1.1.1.1.4	Apple extended schema
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 1.3.6.1.4.1.63.1000.1.1.1.1.71	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)

Open Directory name, RFC/class, special purpose	LDAP attribute name OID	Active Directory plug-in
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 1.3.6.1.4.1.63.1000.1.1.1.110	Apple extended schema
MCXSettings, Apple registered	apple-mcxsettings 1.3.6.1.4.1.63.1000.1.1.1.116	Apple extended schema

Open Directory name, RFC/class,	LDAP attribute name OID	Active Directory plug-in
Computers, Apple registered	apple-computers 1.3.6.1.4.1.63.1000.1.1.11.3	Apple extended schema
Group, Apple registered	apple-computer-list-groups 1.3.6.1.4.1.63.1000.1.1.11.4	Apple extended schema
Keywords, Apple registered	apple-keyword 1.3.6.1.4.1.63.1000.1.1.11.19	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 1.3.6.1.4.1.63.1000.1.1.12.3	1.2.840.113556.1.2.13 (Microsoft)
DataStamp, Apple registered	apple-data-stamp 1.3.6.1.4.1.63.1000.1.1.12.2	Apple extended schema
KDCAuthKey, Apple registered, Apple KDC	apple-kdc-authkey 1.3.6.1.4.1.63.1000.1.1.12.7	No mapping
KDCConfigData, Apple registered, Apple KDC	apple-kdc-configdata 1.3.6.1.4.1.63.1000.1.1.12.8	No mapping
Keywords, Apple registered	apple-keyword 1.3.6.1.4.1.63.1000.1.1.11.19	Apple extended schema
LDAPReadReplicas, Apple registered, Apple LDAP Server	apple-ldap-replica 1.3.6.1.4.1.63.1000.1.1.12.5	No mapping

Open Directory name, RFC/class, special purpose	LDAP attribute name OID	Active Directory plug-in
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

Open Directory name, RFC/class	LDAP attribute name OID	Active Directory plug-in
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 1.3.6.1.4.1.63.1000.1.1.1.16	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 1.3.6.1.4.1.63.1000.1.1.1.10	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 1.3.6.1.4.1.63.1000.1.1.1.14.5	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

Open Directory name, RFC/class	LDAP attribute name OID	Active Directory plug-in
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 1.3.6.1.4.1.63.1000.1.1.1.1.6	N/A
HomeDirectoryQuota, Apple registered	apple-user-homequota 1.3.6.1.4.1.63.1000.1.1.1.1.8	Apple extended schema
HomeDirectorySoftQuota, Apple registered	apple-user-homesoftquota 1.3.6.1.4.1.63.1000.1.1.1.1.17	Apple extended schema
MailAttribute, Apple registered	apple-user-mailattribute 1.3.6.1.4.1.63.1000.1.1.1.1.9	Apple extended schema
PrintServiceUserData, Apple registered	apple-user-printattribute 1.3.6.1.4.1.63.1000.1.1.1.1.13	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
AdminLimits, Apple registered	apple-user-adminlimits 1.3.6.1.4.1.63.1000.1.1.1.1.14	Apple extended schema
Picture, Apple registered	apple-user-picture 1.3.6.1.4.1.63.1000.1.1.1.1.12	Apple extended schema
AuthenticationAuthority, Apple registered	authAuthority 1.3.6.1.4.1.63.1000.1.1.2.16.1	Apple extended schema

Open Directory name, RFC/class	LDAP attribute name OID	Active Directory plug-in
PasswordPolicyOptions, Apple registered	apple-user-passwordpolicy 1.3.6.1.4.1.63.1000.1.1.1.18	Apple extended schema
PresetUsersAdmin, Apple registered	apple-preset-user-is-admin 1.3.6.1.4.1.63.1000.1.1.1.15.1	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 1.3.6.1.1.1.12	Extended using RFC
PrimaryGroupID, RFC 2307	gidNumber 1.3.6.1.1.1.1	Extended using RFC
NFSHomeDirectory, RFC 2307	homeDirectory 1.3.6.1.1.1.3	N/A
UserShell, RFC 2307	loginShell 1.3.6.1.1.1.4	Extended using RFC
Change, RFC 2307	shadowLastChange 1.3.6.1.1.1.5	N/A
Expire, RFC 2307	shadowExpire 1.3.6.1.1.1.10	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
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 Open Directory Record Types and Attributes

For information about the standard attributes and record types in Open Directory domains, see:

- “Standard Attributes in User Records” on page 217
- “Standard Attributes in Group Records” on page 222
- “Standard Attributes in Computer Records” on page 223

- “Standard Attributes in Computer List Records” on page 224
- “Standard Attributes in Mount Records” on page 224
- “Standard Attributes in Config Records” on page 225

For a complete list of standard record types and attributes, see the following file:

/System/Library/Frameworks/DirectoryService.framework/Headers/DirServicesConst.h

Standard Attributes in User Records

The following table describes the standard attributes found in Open Directory user records. Use this information when working in Workgroup Manager’s Inspector pane or when mapping user record attributes with Directory Access.

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 <i>Important:</i> 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).
UniqueID: A unique user identifier, used for access privilege management	Signed 32-bit ASCII string of digits 0–9	Values below 500 may have special significance. 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.

Mac OS X user attribute	Format	Sample values
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	
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	

Mac OS X user attribute	Format	Sample values
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	
AuthenticationAuthority: Describes the user's authentication methods, such as Open Directory, shadow password, 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, ;ApplePasswordServer; and ;Kerberosv5;). Each value has the format <i>vers; tag; data</i> (where <i>vers</i> and <i>data</i> may be blank). Crypt password: ;basic; Open Directory password: ;ApplePasswordServer;HexID, server's public key IPAddress:port ;Kerberosv5;Kerberos data Shadow password (local directory domain only): • ;ShadowHash; • ;ShadowHash;<list of enabled authentication methods>

Mac OS X user attribute	Format	Sample values
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		
EEmailAddress: 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	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		
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/CIFS, 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)
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	EEmailAddress	Optional

Standard Attributes in Group Records

The following table describes the standard attributes found in Open Directory group records. Use this information when working in Workgroup Manager's Inspector pane or when mapping group attributes with Directory Access.

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).
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, _-	

Mac OS X group attribute	Format	Sample values
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 describes the standard attributes found in Open Directory computer records. Computer records associate the hardware address of a computer's primary 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 this information when working in Workgroup Manager's Inspector pane or when mapping computer record attributes with Directory Access.

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 describes the standard attributes found in Open Directory computer list records. A computer list record identifies a group of computers (much as a group record identifies a collection of users). Use this information when working in Workgroup Manager's Inspector pane or when mapping computer list record attributes with Directory Access.

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 describes the standard attributes found in Open Directory mount records. Use this information when working in Workgroup Manager's Inspector pane or when mapping mount record attributes with Directory Access.

Mac OS X mount attributes	Format	Sample values
RecordName: Host and path of the sharepoint	UTF-8 text	<i>hostname:/path on server</i> 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

Mac OS X mount attributes	Format	Sample values
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 describes the standard attributes found in the following two types of Open Directory 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 this information when working in Workgroup Manager's Inspector pane or when mapping config record attributes with Directory Access.

Mac OS X config attributes	Format	Sample values
<code>RecordName</code> : Name associated with a config	ASCII characters A–Z, a–z, 0–9, _-.,	<code>mcx_cache</code> <code>passwordserver</code> Non-zero length, maximum 255 bytes (85 triple-byte to 255 single-byte characters).
<code>PasswordServerLocation</code> : Identifies the host of the Password Server that's associated with the directory domain	IP address or host name	192.168.1.90
<code>RealName</code>		For the <code>mcx_cache</code> config record, <code>RealName</code> is a GUID
<code>DataStamp</code>		For the <code>mcx_cache</code> config record, <code>DataStamp</code> is a GUID

Active Directory The directory and authentication service of Microsoft Windows 2000 Server and Windows Server 2003.

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've 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.

attribute A named data item containing a specific type of information and belonging to an entry (record or object) in a directory domain. The actual data that an attribute contains is its value.

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.

binding (n.) A connection between a computer and a directory domain for the purpose of getting identification, authorization, and other administrative data. (v.) The process of making such a connection. See also **trusted binding**.

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.

CIFS Common Internet File System. See **SMB/CIFS**.

class See **object class**.

computer account See **computer list**.

computer list A list of computers that have the same preference settings and are available to the same users and groups.

cracker A malicious user who tries to gain unauthorized access to a computer system in order to disrupt computers and networks or steal information. Compare to hacker.

crypt password A type of password that's stored as a hash (using the standard UNIX encryption algorithm) directly in a user record.

DHCP Dynamic Host Configuration Protocol. A protocol used to dynamically 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.

distinguished name Specifies an entry (object) in an LDAP directory. Represented as a sequence of directory entries separated by commas, starting with the entry itself and followed by each entry that contains the previous entry in the sequence. Example: "cn=users, dc=example, dc=com."

entry A (usually short) article posted on a weblog. Readers can add comments to the entry, but the content associated with the entry can be changed only by the weblog owner. In an LDAP directory, an entry is a collection of attributes (data items) that has a unique distinguished name.

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 folder A directory that organizes documents and applications of special interest to group members and allows group members to pass information back and forth among themselves.

guest user A user who can log in to your server without a user name or password.

hacker An individual who enjoys programming, and explores ways to program new features and expand the capabilities of a computer system. See also **cracker**.

hash (noun) A scrambled, or 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.

KDC Kerberos Key Distribution Center. A trusted server that issues Kerberos tickets.

Kerberos realm The authentication domain comprising the users and services that are registered with the same Kerberos server. The registered services and users trust the Kerberos server to verify each other's identities.

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's possible to access additional services without retying a password (this is called single sign-on) 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.

local hostname A name that designates a computer on a local subnet. It can be used without a global DNS system to resolve names to IP addresses. It consists of lowercase letters, numbers, or hyphens (except as the last characters), and ends with “.local” (e.g, bills-computer.local). Although the name is derived by default from the computer name, a user can specify this name in the Network pane of System Preferences. It can be changed easily, and can be used anywhere a DNS name or fully qualified domain name is used. It can only resolve on the same subnet as the computer using it.

long name The long form of a user or group name. See also **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.

multicast DNS 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.” For more information, visit www.apple.com or www.zeroconf.org. To see how this protocol is used in Mac OS X Server, see **local hostname**.

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 Directory master A server that provides LDAP directory service, Kerberos authentication service, and Open Directory Password Server.

Open Directory password A password that's stored in secure databases on the server and can be authenticated using Open Directory Password Server or Kerberos (if Kerberos is available).

Open Directory Password Server An authentication service that validates passwords using a variety of conventional authentication methods required by the different services of Mac OS X Server. The authentication methods include APOP, CRAM-MD5, DHX, LAN Manager, NTLMv1, NTLMv2, and WebDAV-Digest.

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 owner of an item can change access permissions to the item. The owner may also change the group entry to any group in which the owner is a member. By default the owner has Read & Write permissions.

parent A computer whose shared directory domain provides configuration information to another computer.

password An alphanumeric string used to authenticate the identity of a user or to authorize access to files or services.

Password Server See **Open Directory Password Server**.

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.

principal, Kerberos The name and other identifying information of a client or service that Kerberos can authenticate. A user principal is usually a user's name or user's name and Kerberos realm. A service's principal is usually the service name, server's fully-qualified DNS name, and Kerberos realm.

protocol A set of rules that determines how data is sent back and forth between two applications.

schema The collection of attributes and record types or classes that provide a blueprint for the information in a directory domain.

search base A distinguished name that identifies where to start searching for information in an LDAP directory's hierarchy of entries.

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.

shadow password A password that's stored in a secure file on the server and can be authenticated using a variety of conventional authentication methods required by the different services of Mac OS X Server. The authentication methods include APOP, CRAM-MD5, DHX, LAN Manager, NTLMv1, NTLMv2, and WebDAV-Digest.

share point A folder, hard disk (or hard disk partition), or CD that's 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 sign-on 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 sign-on.

SLP DA Service Location Protocol 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/CIFS Server Message Block/Common Internet File System. 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/CIFS 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. More recent versions of SSL are known as TLS (Transport Level Security).

standalone server A server that provides services on a network but doesn't get directory services from another server or provide directory services to other computers.

ticket, Kerberos A temporary credential that proves a Kerberos client's identity to a service.

ticket-granting ticket A special Kerberos ticket that enables a client to obtain tickets for services within the same realm. A client gets a ticket-granting ticket by proving identity, for example by entering a valid name and password during login.

trusted binding A mutually authenticated connection between a computer and a directory domain. The computer provides credentials to prove its identity, and the directory domain provides credentials to prove its authenticity.

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's defined to provide access for WebDAV users and groups.

A

- access privileges, directory services and 24
- ACL attribute 200
- Active Directory
 - administrator groups in 147
 - binding 141
 - configuring access to 141
 - credential caching 143
 - editing accounts 149
 - forest 140
 - group GID mapping 146
 - home folders 143
 - Kerberos 63
 - LDAPv3 access to 149
 - mobile accounts 143
 - plug-in 139
 - preferred server 147
 - primary GID mapping 145
 - problem solving 172
 - replication 140
 - search policies and 142, 150
 - service, enabling or disabling 110
 - UID mapping 145
 - unbinding from 149
 - UNIX shell 144
- Address Book
 - LDAP directory access 118
- administrative data
 - See directory domains
- administrator
 - choosing for directory services 73
 - groups in Active Directory 147
 - Kerberos 81
 - NetInfo 153
 - Open Directory 103
 - Open Directory Password Server 103
 - password of 104, 175
 - password policy 44, 99, 100
- administrator computer 67
- APOP authentication 50
- AppleTalk
 - service discovery, enabling or disabling 111
 - service discovery protocol 26
- archive, Open Directory master 167
- attributes
 - See also specific attributes
 - about 28
 - ACL 200
 - adding 131
 - authentication authority 198
 - autoserver setup 216
 - computer 194, 207, 223
 - computer list 195, 208, 224
 - config 209, 225
 - configuration 196
 - group 190, 205, 222–223
 - LDAP 185
 - location 198, 216
 - machine 191
 - mapping Active Directory 145, 146
 - mapping LDAP 129
 - mount 192, 206, 224–225
 - neighborhood 200
 - people 210
 - preset computer list 212
 - preset group 212
 - preset user 198, 213
 - printer 193, 215
 - schema 201
 - service 199
 - service URL 196
 - time-to-live 185
 - user 185, 202, 217–220
 - xml plist 196
- authentication
 - authentication authority attribute 43
 - and authorization 40
 - crypt password 41
 - Kerberos 40, 44, 45, 81, 83, 85, 99
 - methods 50, 101, 102
 - monitoring 160
 - Open Directory 40
 - security 52, 53
 - shadow password 41
- authentication authority attribute 43, 198
- authentication authority object class 184

- Authentication Manager 54, 90, 107
- authentication search policy
 - about 33, 113
 - automatic 114
 - custom 115, 117, 122, 123, 126, 150
 - items in red 110, 111
 - local directory only 116
 - security and 117
- authorization 40
- automatic search policy
 - See also* search policies
 - about 36
 - LDAP mappings supplied by 132
 - mobile computer 117
 - security and 117
 - using 114
- automounting, directory services and 25

B

- basic authentication 41
- Berkeley DB 11
- binding
 - Active Directory 141
 - LDAP 37, 114, 132, 133
 - NetInfo 153
- Bonjour 26, 112
- BSD configuration files
 - history of 21
 - populating with data 152
 - service, enabling or disabling 111
 - using 151

C

- child NetInfo domain 153
- CIFS
 - See* SMB/CIFS
- clear text password 50
- command-line shell attribute, Active Directory 144
- computer attributes 194
- computer list attributes 195, 224
- computer list object class 182
- computer object class 181
- computer records, attributes of 223
- config record, non-Apple LDAP mapping of 136
- config records, attributes 225
- configuration attributes 196
- configuration files
 - See* BSD configuration files
- configuration object class 182
- connected to a directory system 80
- contacts search policy
 - about 33, 113
 - automatic 114
 - custom 115, 117, 122, 123, 126, 150
 - items in red 110, 111

- local directory only 116
- container object class 179
- CRAM-MD5 authentication 50
- credential caching, Active Directory 143
- crypt passwords 41, 97, 105
 - troubleshooting 174
- custom search policy
 - about 38
 - items in red 110, 111, 112
 - security and 117
 - using 115, 117, 122, 123, 126, 150

D

- database
 - Berkeley DB 11, 60
 - directory domain 20, 60
 - Kerberos 48, 65
 - LDAP 65, 88
 - Open Directory Password Server 53, 65
- delegated Kerberos authority 81, 83, 85, 174
- denial-of-service attack 88, 89
- DHCP
 - automatic search policy and 37, 114
 - directory service security 117
 - LDAP server for DHCP clients 37, 114, 117, 118
 - migrated LDAP directory and 92
 - NetInfo binding 114, 117, 154
 - Open Directory replica and 78
 - option 95 37
- DHX authentication 41, 50
- Digest-MD5 authentication 50
- Directory Access application
 - Active Directory binding 141
 - Active Directory enabling or disabling 110
 - Active Directory forest or domains 148
 - Active Directory group GID mapping 147
 - Active Directory home folders 144
 - Active Directory preferred server 147
 - Active Directory primary GID mapping 146
 - Active Directory UID mapping 145
 - Active Directory unbinding 149
 - Active Directory UNIX shell attribute 145
 - Active Directory via LDAPv3 150
- administrator groups in Active Directory 148
- AppleTalk enabling or disabling 111
- automatic search policy, defining 115
- BSD access 151
- BSD enabling or disabling 111
- custom search policies, defining 115
- LDAP access via DHCP 119
- LDAPv2 connections, forcing 135
- LDAPv3 configuration, adding 120, 122
- LDAPv3 configuration, changing 124
- LDAPv3 configuration, deleting 126
- LDAPv3 configuration, duplicating 125

- LDAPv3 configurations, showing and hiding 119
- LDAPv3 connection options 127
- LDAPv3 connections, authenticating 135, 136
- LDAPv3 enabling or disabling 112
- LDAPv3 idle timeout 134
- LDAPv3 open/close timeout 133
- LDAPv3 query timeout 134
- LDAPv3 rebind-try delay time 134
- LDAPv3 search bases and mappings 130, 137
- LDAPv3 security options 128
- LDAPv3 server referrals 135
- LDAPv3 trusted binding 132, 133
- local domain search policy 116
- NetInfo binding, configuring 154
- NetInfo enabling or disabling 112
- NIS access 151
- remote server configuration 109
- RFC 2307 137
- search policies 113–117
- SLP, enabling or disabling 112
- SMB/CIFS, configuring 113
- SMB/CIFS, enabling or disabling 113
- uses 68
- Directory Access Controls (DACs) 163
- directory domains
 - information storage in 20, 59
 - organization 28
 - planning 57
 - requirements 60
 - security 60
 - simplifying changes to 59
 - user accounts in 20
- directory services
 - See also* Open Directory
 - administrators for 73
 - benefits of 19
 - logs 160
 - network role of 20
 - planning 72
 - status 160
 - tools summary 67
- distinguished name (DN) 29
- DNS (Domain Name System)
 - Bonjour 26
 - Kerberos and 75, 81, 169
- documentation
 - guides 13, 15
 - onscreen help 14
 - resources 16
 - updates 16
 - using 14

E

- encryption
 - LDAP 89

- password 43, 51
- entries, LDAP 28

F

- failover
 - Active Directory 140
 - Open Directory 79, 120, 122

G

- global password policy 99
- group accounts, changing in Active Directory 149
- group attributes 190, 205, 222–223
- group GID mapping, Active Directory 146
- group object class 180
- group records 25, 205

H

- hash, password 41, 51
- help, using 14
- home directories 25
- home folders
 - Active Directory 140, 143

I

- idle timeout, LDAP 134
- importing and exporting
 - Authentication Manager users 107
 - passwords 105
 - records of any type 164
- Inspector
 - DACs, setting with 163
 - hiding 162
 - records, deleting with 163
 - short name, changing with 162
 - showing 161

J

- joining a Kerberos realm 85, 174

K

- kadmin log 160
- KDC
 - See* Kerberos
- kdc log 160
- Kerberized services 48
- Kerberos
 - about 45
 - Active Directory 63, 141, 142
 - archiving 167
 - authentication authority 43
 - authentication process 49
 - cross-realm 64
 - delegated authority 83, 85, 174
 - deployment barriers 46
 - DNS and 75, 81

- enabling 99
- enabling for users 99
- joining 83, 174
- LDAP digital signature 87, 129
- LDAP encryption 87, 129
- man-in-the-middle attacks, blocking 87, 129
- multiple realm 63
- Open Directory master 82
- password policy 44, 49, 99, 100
- principals 48
- realm 48, 76, 83
- replication 61
- restoring from archive 168
- security 47
- services supporting 48, 81
- setting up 81
- and smart card 47
- solving problems 173, 174
- starting 82
- stopped 169
- ticket 49
- ticket-granting ticket 49
- time sync 50, 173
- troubleshooting 173
- using 48

L

- LAN Manager authentication 50

LDAP

- See also* directory domains
- accessing directories 120, 122
- accessing directories in Mail and Address Book 118
- Active Directory access via 149
- archiving 167
- attributes 185
- authentication 129, 135, 136
- automatic search policy and 37
- binding to 37, 54
- changing a configuration for accessing 124
- clear text passwords 87, 129
- connection settings, changing 127
- database location 88
- deleting access configurations 126
- DHCP-supplied 117, 118, 132
- digitally signed 87, 129
- directory access controls (DACs) 163
- directory service protocol 25
- distinguished name 29
- duplicating an access configuration 125
- encrypted 87, 129
- entries 28
- hiding configurations 119
- idle timeout 134
- LDAPv2 forced 135

- log 160
- man-in-the-middle attacks, blocking 87, 129
- manually configuring 122
- mapping objects and attributes 129
- mappings from server 131
- migrating from NetInfo 90
- non-Apple 136, 137
- object classes 28, 178
- open/close timeout 133
- populating with data 138
- port configuration 127
- ports used 66
- query timeout 134
- read-only 138
- rebind-try delay 134
- relative distinguished name 29
- replication 61
- restoring from archive 168
- RFC 2307 137
- schema extensions 178
- search base 29
- search policies and 122, 123, 126
- search results, limiting 88
- search scope 29
- search timeout 89
- security 117, 128
- server referrals 135
- service, enabling or disabling 111
- shared domains 36
- showing configurations 119
- SSL 89, 124, 127
- structure 29
- switching clients from NetInfo 92
- trusted binding 118, 129, 132, 133
- LDAP bind authentication 54, 104
- Lightweight Directory Access Protocol (LDAP)
 - See* LDAP
- load balancing 62
- local directory domain
 - in automatic search policy 36
 - information storage 30
 - NetInfo 153
 - search policy 34, 116
 - standalone server 73
- local home, Active Directory 143
- local search policy 34
- location attributes 198
- location object class 184
- login
 - authenticating 21, 24
 - solving problems 172
 - user instructions 76
- login window
 - controlling access to a server's 158
- logs
 - directory services 160

- kadmin 160
- kdc 160
- LDAP 160
- lookupd 160
- NetInfo 160
- password service 160
- slapconfig 160
- lookupd log 160

M

- machine attributes 191
- machine object class 180
- Mac OS X Server
 - administration applications 67
 - data items used by 221
 - shared directory domains 31
 - what's new 12
- Mail
 - LDAP directory access 118
- managed client, non-Apple LDAP mapping for 136
- managed client data 25
- managed network view 25
- man-in-the-middle attacks 87, 129
- mapping
 - Active Directory attributes 145, 146
 - autoserver setup records 215
 - computer list records 208, 224
 - computer records 207, 223
 - config records 209, 225
 - group records 205, 222–223
 - LDAP objects and attributes 129
 - location records 216
 - mount records 206, 224–225
 - people records 210
 - preset computer list records 211
 - preset group records 212
 - preset user records 213
 - printer records 214
 - user records 201, 217–220
- master
 - See Open Directory master
- migration
 - NetInfo to LDAP 90
- mobile accounts
 - Active Directory 140, 143
- mount attributes 192
- mount object class 180
- mount records 206, 224–225
- MS-CHAPv2 authentication 50, 51, 81, 171
- multicast DNS 26

N

- NAT router, Open Directory and 63
- neighborhood attributes 200
- neighborhood object class 184

NetInfo

- See also directory domains
- binding 117, 153
- child 153
- directory service protocol 25
- disabling domain 90, 92
- log 160
- migrating to LDAP 90
- parent 153
- port configuration 156
- security 117
- service, enabling or disabling 112
- shared domain 36
- switching clients to LDAP 92
- NetInfo Manager 69, 155, 156
- network home, Active Directory 143
- network services
 - data items used by 221
 - discovery protocols 26
- NIS, accessing 151
- NT authentication 50
- NTLM authentication 50
- NTLMv2 authentication 51, 80

O

- object classes 28, 178
- offline attack 42
- open/close timeout, LDAP 133
- Open Directory
 - See also directory services, Open Directory master,
 - Open Directory replica
 - access privileges and 24
 - administrator rights 103
 - automount share points and 25
 - compared to UNIX systems 23
 - configuring protocols 110
 - group records and 25
 - home directories and 25
 - information management 24, 26
 - information storage in 26
 - mail settings and 25
 - managed client data and 25
 - managed network views 25
 - monitoring 160
 - NAT router and 63
 - performance 65
 - planning 57
 - quotas and 25
 - replication 120, 122
 - schema 178
 - searching non-Apple domains 32
 - search policies 33
 - security 65
 - service discovery and 26
 - setup 71

- UNIX heritage 21
- uses of 24–25
- Open Directory master
 - about 61
 - archiving 167
 - controlling access to 157
 - failover to replica 79
 - Kerberos 82
 - Kerberos stopped 169
 - promoted from replica 165
 - restoring from archive 168
 - setting up 75
 - single sign-on 82
 - status check 159
- Open Directory password
 - about 40
 - authentication methods 51, 102
 - changing 96
 - solving problems 171, 172
- Open Directory Password Server
 - archiving 167
 - authentication authority 43
 - database 53
 - hosting 75, 77
 - password policy 44, 99, 100
 - replication 61
 - restoring from archive 168
 - security 52
 - setting up 75, 77
 - Windows authentication 11
- Open Directory replica
 - about 61
 - controlling access to 157
 - decommissioning 166
 - failover from master 79
 - Kerberos stopped 169
 - monitoring 160
 - multiple 79
 - password policy 61
 - promoting to master 165
 - setting up 77, 170
 - status check 159
- OpenLDAP 11
- Option 95, DHCP 37

P

- parent NetInfo domain 153
- password policy
 - administrator 44, 101
 - global 99
 - individual user 100
 - Kerberos 44, 49
 - mobile user 44
 - Open Directory Password Server 44
 - replicas 61

- passwords
 - administrator 104
 - changing 94
 - clear text 50, 87, 129
 - composing 93
 - cracking 42
 - crypt password type 41, 97
 - imported users 105
 - incompatible 171
 - migrating to Open Directory 105
 - offline attacks 42
 - Open Directory password type 40, 96
 - resetting multiple 95
 - shadow password type 41, 98
 - synchronizing changes in replicas 79
 - troubleshooting 171, 174
 - unable to modify 171
- Password Server
 - See Open Directory Password Server
- password service logs 160
- password type
 - about 39
 - changing 96
 - crypt password 41, 97
 - Open Directory password 40, 96
 - shadow password 41, 98
- performance, Open Directory 65
- planning 57
- preset computer list object class 182
- preset group object class 182
- preset user attributes 198
- preset user object class 183
- primary GID mapping, Active Directory 145
- principals, Kerberos 48
- printer attributes 193
- printer object class 181
- protocols
 - See also specific protocols
 - directory services 110
 - Open Directory 110
 - service discovery 26

Q

- query timeout, LDAP 134
- quotas, user settings 25

R

- raw directory data, editing 161
- realm, Kerberos 48, 76, 83
- RealName, mapping to LDAP attribute 130
- rebind-try delay, LDAP 134
- RecordName, mapping to LDAP attribute 130
- record types
 - See also specific record types
 - about 28

- mapping to LDAP objects 129
- redundancy, Open Directory 65
- relative distinguished name (RDN) 29
- remote administration 67
- remote server configuration 109
- replica
 - See Open Directory replica
- replication
 - Active Directory 140
 - failover 79
 - frequency 164
 - multibuilding 62
 - Open Directory 120, 122
 - planning 61
 - slow network link and 61
- requirements
 - directory and authentication 60
- RFC 2252 178
- RFC 2307 137, 178
- RFC 2798 178
- root domain, NetInfo 153

S

- SASL (Simple Authentication and Security Layer) 50
- schema
 - Active Directory 139, 140, 145, 146
 - attributes 201
 - LDAP mapping 129
 - Open Directory extensions 178
- search base
 - LDAP directory 29, 76
 - mappings stored on server 131
 - object class, for 129, 130
 - suffix 121, 123, 124, 125
- search policies
 - about 33, 113
 - adding Active Directory 142, 150
 - adding an LDAP directory 122, 123, 126
 - adding BSD files 152
 - adding NIS 151
 - automatic 36
 - changing 117
 - custom 38, 115, 117, 122, 123, 126
 - local directory 34, 116
 - mobile computer 117
 - security and 117
- search results, limiting LDAP 88
- search scope, LDAP 29, 129
- search timeout, LDAP 89
- security
 - authentication methods 52, 53
 - automatic search policy 117
 - DHCP-supplied directory servers 117
 - LDAP connection 117, 128
 - NetInfo binding 117

- Open Directory 65
- passwords 42
- of server hardware 60
- Server Admin
 - connecting to existing directory domain 80
 - directory services information 160
 - login window access control 158
 - Open Directory access control 157
 - Open Directory master 75
 - Open Directory replica 77, 165, 167
 - Open Directory replica monitoring 160
 - Open Directory status 159
 - SSH access control 158
 - uses 67
- server administration guides 15
- Server Assistant 73
- server assistant configuration object class 184
- server referrals, LDAP 135
- servers, security of 60
- service access control lists 40
- service attributes 199
- service discovery 26, 110
- service object class 184
- service URL attributes 196
- shadow password
 - about 41
 - authentication authority 43
 - authentication methods 52, 101
 - setting 98
- shared directory domain
 - See also LDAP, NetInfo
 - connecting to existing 80
 - hosting 75, 77
 - information storage 31
- short name, changing 162
- single sign-on
 - See also Kerberos
 - about 44
 - Kerberos tickets 46
- slapconfig.lock 170
- slapconfig log 160
- SLP (Service Location Protocol) 26, 112
- smart card authentication 47
- SMB/CIFS
 - service discovery, configuring 113
 - Windows protocol 26
- SMB-LAN Manager authentication 50
- SMB-NT authentication 50
- SSH
 - restricting 158
- SSL
 - LDAPv3 directory access 124, 127
 - Open Directory service 89
- standalone server 73
- startup delay 117, 170
- status

- Open Directory master 159
- Open Directory replica 159
- suffix, search base 76, 121, 123, 124, 125

T

- templates, LDAP mapping 129
- ticket, Kerberos 49
- ticket-granting ticket, Kerberos 49
- time, sync for Kerberos 50
- time-to-live attribute 185
- time-to-live object class 179
- trusted binding, LDAP 132, 133

U

- UID mapping, Active Directory 145
- unbinding
 - Active Directory 149
 - LDAP 133
- UNIX
 - BSD configuration files 151
 - compared to Open Directory 21
 - configuration files 22
 - shell attribute, Active Directory 144
- user accounts
 - changing in Active Directory 149

- in directory domains 20
- user object class 179
- user records
 - attributes 185, 202, 217–220
 - how used by server 221
 - mapping 201, 221
- users
 - login instructions 76

V

- VPN authentication 51, 81, 171

W

- WebDAV-Digest authentication 50
- Windows services
 - authentication 11, 51, 80
 - discovering via SMB/CIFS 113
- Windows workgroup, changing 113
- WINS, configuring 113
- Workgroup Manager
 - populating LDAP domains with 138
 - uses 68

X

- xml plist attributes 196