



## Understanding and Using NetInfo

Includes information on setting up Mac OS X Server and NetInfo to increase the power of your Mac OS X network

#### **É** Apple Computer, Inc.

© 2001 Apple Computer, Inc. All rights reserved.

Under the copyright laws, this publication may not be copied, in whole or in part, without the written consent of Apple.

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, AppleShare, 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.

© 1995-2001 The Apache Group. All rights reserved.

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

062-8432/06-16-01

### Contents

#### Preface About This Document 7

What's in This Document 7 Where to Find More Information 8

#### 1 What Is NetInfo? 9

NetInfo: A Service for Mac OS X Processes 9 A Historical Perspective 10 Data Consolidation 10 Data Distribution 11 The Power of NetInfo: Software That Uses It 12 Folder and File Ownership 12 Home Directories 13 Mounts 14 Architectural Elements of NetInfo 14 Local Data 14 Shared Data 15 NetInfo Hierarchies 17 Binding 19 Replication 21 Inside NetInfo 22 Accessing and Manipulating NetInfo Data 24 Defining NetInfo Domains 24 Configuring NetInfo Hierarchies 24 Setting Up Search Policies 25 Managing NetInfo Data 25

Viewing NetInfo Data 25 Using Command Line Utilities 27 The Importance of Planning 27 **NetInfo Planning** 2 29 General Planning Guidelines 29 Controlling NetInfo Data Visibility 31 Simplifying Changes to NetInfo Data 31 Identifying Computers for Hosting Shared Domains Devising a Binding Strategy 32 User Data Planning 32 Understanding the Login Environment 32 Contrasting Logging In and Connecting 35 Managing Names 35 Managing UIDs 38 Setting Up Home Directories 38 Group Data Planning - 39 Ensuring Group Visibility 39 Avoiding Duplicate Short Names 40 The Next Step 41 **Setting Up NetInfo Hierarchies** 3 43 The Overall Process 43 Setting Up the Root Domain of a Simple Hierarchy 44 Setting Up Shared Domains in Deeper Hierarchies 45 Understanding Machine Records 45 Defining Shared Domains 47 Setting Up Local Domains of Network Users 51 Static Binding - 51 DHCP Binding 52 Broadcast Binding 52 Setting Up Replication 53 Distinguishing Masters 53

31

Locating and Using Masters and Clones 54

Creating Masters 54

Creating Clones 54 Replacing a Master With a Clone 55 Setting Up Windows User Authentication 56 Simple Hierarchies With No Clones 56 Other Hierarchies 57 Disabling Authentication Manager 60 Populating Domains 60 Setting Up Mounts and Automounting 60 Defining Users and Groups 61 Sharing Printers 62

#### PREFACE

## About This Document

#### What's in This Document

If you're a system or network administrator whose responsibilities include Mac OS X administration, this document will help you understand and implement NetInfo.

NetInfo is the directory system that is built into computers running Mac OS X and Mac OS X Server. NetInfo facilitates the management of administrative information used by Mac OS X computers.

For example, NetInfo lets you centralize information about users, printers, servers, and other network devices so that all Mac OS X computers on your network, or only some of them, have access to it. It helps you set up and manage home directories for Mac OS X users on multiple, integrated Mac OS X Servers. And it simplifies the day-to-day management of administrative information by letting you update information that's used across the network in one central place.

- Chapter 1, "What Is NetInfo?," introduces NetInfo. It tells you how NetInfo is used by Mac OS X computers and highlights key aspects of its external and internal architecture. It also introduces you to the various ways you can access and manipulate NetInfo data.
- Chapter 2, "NetInfo Planning," provides guidelines to help you decide how to implement NetInfo in your environment. Use the information in this chapter to design a NetInfo hierarchy that gives your Mac OS X users easy access to the network resources they need, yet minimizes the time you spend maintaining NetInfo data.
- Chapter 3, "Setting Up NetInfo Hierarchies," tells you how to create and configure the components of a NetInfo directory system.

#### Where to Find More Information

The following information is available for Mac OS X Server administrators. Mac OS X Server is a powerful server platform that delivers a complete range of services to network users, including applications that help you set up and manage your NetInfo data:

- *Mac OS X Server Administrator's Guide* provides information about Mac OS X Server's administrative applications and how to use them to set up your server.
- The online help for each server administration program provides step-by-step instructions for everyday server management.
- *Mac OS X Server Migration Guide* provides instructions for upgrading to Mac OS X Server from AppleShare IP, Macintosh Manager, and Mac OS X Server 1.2.

If you would like help planning, designing, and implementing NetInfo, contact Apple iServices at iservices@apple.com or call 800-848-6398.

# C H A P T E R

## What Is NetInfo?

NetInfo is the built-in Mac OS X directory system. A directory system is software that system and application processes can use to store and find administrative information about resources and users.

The Mac OS X login process, for example, consults user information in NetInfo to determine whether the name and password entered in the login window are those of a valid user. Other processes need information about the location of such resources as home directories, printers, file servers, and other devices available from a particular Mac OS X computer.

#### **NetInfo: A Service for Mac OS X Processes**

NetInfo stores information about users and resources and makes it available to Mac OS X processes that want to use it.



Processes running on Mac OS X computers can save information in NetInfo, and processes that need the information can retrieve it from NetInfo. For example, when you set up a user account, the application you use to do so stores information about the user in NetInfo:

- On a computer running Mac OS X, you use the Users pane of System Preferences to set up user accounts.
- *On Mac OS X Server*, use the Users & Groups module of Server Admin, which lets you set up additional user attributes, such as the user's home directory.

No matter which application you use, the user information is stored in NetInfo. When a user attempts to log in to a Mac OS X computer, the login process consults the information in NetInfo to authenticate the user.



This chapter introduces NetInfo. It briefly describes how it evolved. It samples some of the common ways NetInfo is used, illustrating how it makes Mac OS X one of the world's most advanced operating systems. And it highlights key elements of NetInfo's visible and behind-the-scenes architecture.

#### **A Historical Perspective**

Like Mac OS X, NetInfo has a UNIX heritage. Much of what it manages is the same administrative data formerly kept in UNIX configuration files, but it consolidates the data and distributes it for ease of access and maintenance.

#### **Data Consolidation**

In early UNIX systems, administrative information was stored in a collection of files located in the /etc directory. Every computer had its own set of these files, and processes read the files when they needed administrative information. If you're experienced with UNIX, you'll likely recall the files in the /etc directory—group, hosts, hosts.eq, passwd, and so forth.



When a process needed to retrieve a password, it used one kind of call to consult the /etc/ passwd file, which contained a record for each user. When a process needed group information, it used a different call to read the group file.

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



Processes no longer need to be aware of how and where administrative data is stored. NetInfo does that for them. If a process needs the home directory for a user, it simply retrieves it from NetInfo. NetInfo finds the requested information, then returns it, insulating the process from the details of how the information is stored. And when you take advantage of NetInfo's ability to store administrative data in several NetInfo databases, NetInfo automatically consults them when needed.



Much of the data NetInfo stores is identical to data stored on earlier UNIX systems. The crypt password, the home directory, the real name, short name, UID, GID—all stored in NetInfo user records—have corresponding entries in the standard /etc/passwd file. However, much of the data stored by NetInfo supports functions unique to Mac OS X, such as support for Apple Filing Protocol (AFP) directories.

#### **Data Distribution**

Another characteristic of early storage strategies for administrative data is that the data was stored *locally*. If you wanted to use a specific computer, your user account information had to be stored on that computer. To configure a computer's network settings, the administrator needed to go to each computer and manually enter the IP address and all the other information needed to identify the computer on the network.

Likewise, user or network information needed to be changed on the computer where it resided. Some changes, such as network settings, had to be made on multiple computers. As networks grew in size and complexity, it became unwieldy to maintain administrative information using this approach.

NetInfo solves this problem by letting you store administrative data in such a way that it can be managed by a system administrator from one location. NetInfo lets you distribute the information so that it is visible on a network to both computers that need it and administrators who manage it:



#### The Power of NetInfo: Software That Uses It

Although NetInfo provides an easy-to-maintain database that lets you consolidate and distribute network information, that information is useful only if it can be accessed by processes that need it. The real power of NetInfo is not NetInfo itself, but the fact that Mac OS X software takes full advantage of data stored in NetInfo.

You have already seen how the Users pane of System Preferences or the Users & Groups module of Server Admin creates NetInfo user records, and how these records, in turn, are used to authenticate users who log in to Mac OS X computers. This section highlights a few additional ways that NetInfo data is created and used.

#### **Folder and File Ownership**

The Mac OS X file system uses a particular data item in the user record—the user ID (UID)— to keep track of directory and file ownership.

When a user creates a directory or file, the file system stores the creator's UID. When a user with that UID accesses the directory or file, the user is granted read and write privileges to it. Any process started by the creator is granted read and write privileges to any files associated with the creator's UID.

If an administrator changes a user's UID, the user may no longer be able to modify or even access files and directories she created. Likewise, if the user logs in as a user whose UID is different from the UID used to create the files and directories, the user will no longer have owner access privileges for them.

When you define a user, the UID for the user is automatically assigned and stored in the user's record in NetInfo. The Server Admin Users & Groups module lets you change the UID of users if you need to. You might, for example, need to change a user's UID when merging users created on different servers into one new server or cluster of servers; the same UID may have been associated with different users on the previous servers.

#### **Home Directories**

A home directory is a location for storing a user's personal files and system preferences. Other users can see your home directory and read files in its Public folder, but they can't (by default) access anything else in your home directory.

Home directories are defined using the same applications you use to set up user accounts:

- If you set up the account using the Users pane of System Preferences, a local home directory named using the user's short name is created in the /Users directory.
- If you create a user on Mac OS X Server with the Users & Groups module, you have more control over the user's home directory name and location. For example, you can store the home directory on a remote computer, or you can specify a name for the home directory. You can also set up home directories to mount automatically on the computer where the user logs in, using the Sharing module of Server Admin.

When you define a user's home directory, its location is stored in NetInfo. Various Mac OS X processes use the home directory location. Here are several examples of Mac OS X activities that depend on home directory data stored in NetInfo:

- A user's home directory is displayed when the user clicks Home in a Finder window or chooses Home from the Finder's Go menu.
- Home directories that are set up for mounting automatically appear in the Finder on the computer where the user logs in.
- System preferences you set up, such as Desktop and folder backgrounds, take effect as soon as you log in. These preferences are stored in the Preferences folder in your home directory.

Home directories are an example of how some Mac OS X processes collaborate to define and use NetInfo data. The Finder can display your home directory automatically because it retrieves its location from your NetInfo user record. But making home directories available is more complicated than simply adding data to a NetInfo user record. It involves such file system actions as creating folders with particular privileges on an available file server. And for a remote home directory to be made visible on a user's Desktop, the partition (or share) containing that home directory must be defined as a mount (or share point) and the mount must also have a NetInfo record.

#### **Mounts**

*Mounts* are Network File System (NFS) or AFP directories that have been set up as share points so that their contents are visible to other computers on the network.

You can set up a NetInfo record that makes a share point automatically visible in the Finder of a Mac OS X computer by using the Sharing module of Server Admin. For example, you can make volumes and files associated with share points visible in

/Network/Applications /Network/Library /Network/Servers /Network/Users

#### **Architectural Elements of NetInfo**

The way you make NetInfo data accessible to processes that run on individual Mac OS X computers is by distributing the data among domains that are visible to those computers. A *domain* is a collection of administrative information that is stored in a NetInfo database.

#### **Local Data**

Every Mac OS X computer has a local NetInfo domain. A local domain's administrative data is visible *only* to processes 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 uses data stored in NetInfo.

When the user logs in to a computer running Mac OS X, the login process on that computer consults the local NetInfo domain on that computer. If the user's record is found, the user is granted access to the computer.



After login, if the user chooses Connect To Server from the Go menu to access a computer running Mac OS X Server, the local domain *on the server* is consulted to authenticate the user. Again, if a record for the user is found, the user is granted access to the server.

When you first set up a Mac OS X computer, its local NetInfo 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 UID and the location of the user's home directory.

#### **Shared Data**

While any process running on a Mac OS X computer can use the data stored in its local domain, the real power of NetInfo is that it lets you share administrative data among multiple Mac OS X computers by storing it in shared domains. When a computer is configured to use a shared domain, any administrative data in the shared domain is *also* visible to processes running on that computer.

If a user's record is not found in the local domain of a Mac OS X computer, a NetInfo process automatically searches for the user's record in any shared domains that the computer has access to. 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 Mac OS X Servers, because servers are equipped with tools such as Server Admin for managing network resources and network users.

Similarly, you can make network resources such as printers visible to certain computers by setting up printer records in a shared domain accessed by those computers. For example, graphic artists in a company might need to access color printers and scanners, while copy center personnel need to use high-speed laser printers. Rather than configuring printer access for each computer individually, you could use the Print module of Server Admin to add printers to two shared domains: Graphics and Repro.



Printers visible in the Print Center application on graphic artists' computers would be those in the Graphics domain, while printers in the Repro domain would be visible to computers used by copy center personnel. Printers that have records in shared domains appear in the Directory Services list in Print Center. While some devices may need to be used only by specific departments, some resources, such as personnel forms, may need to be shared by *all* employees. You could make a directory of those forms visible to everybody by setting up a share point for the directory in a shared domain known as the *root* domain, which is always named "/".



Because the root domain is a shared domain that is visible to *all* computers that use a particular NetInfo hierarchy, all graphic and copy center personnel can access the forms.

#### **NetInfo Hierarchies**

Local and shared domains are organized into *hierarchies*, tree-like topologies that have a root domain at the top and local domains at the bottom of the tree.

A hierarchy can be as simple as a local domain and a root domain, or it can contain one or more shared domains between the local and root domains, as in this education example.



Each shared domain is called a *parent* domain, and the domain immediately below it in the hierarchy is called a *child* domain. In this example, the local domain on each undergraduate computer is a child of the parent domain Undergraduates. Undergraduates, in turn, is a child of the parent domain Students, which is a child of the root domain.

A Mac OS X computer has access to NetInfo data stored in any of the parents of its local domain:

- When a Mac OS X login or connection process needs to authenticate a user, the local domain is searched first. If the user is not found in the local domain, its parent domain is searched. If the user is still not found and the parent domain also has a parent, the second parent is searched, and so on up through the hierarchy.
- Printers defined in any of a computer's parent domains appear in the Directory Services list in Print Center.
- All the mounts defined in a computer's parent domains can be visible in one of the Finder's /Network folders.

A NetInfo hierarchy controls which Mac OS X computers can see particular administrative data. The "subtrees" of the hierarchy essentially hide information from other subtrees in the hierarchy. In the education example, computers using the subtree that includes the Graduates domain do not have access to records in the Undergraduates domain. But records in the root domain are visible to any computer that is configured to access the Undergraduates, Graduates, or Faculty domain.

Domain visibility depends on the computer, not the user. So when a user logs in to a different computer, different NetInfo administrative data may be visible to that computer. In the educational scenario, an undergraduate can log in to a graduate student's computer if the undergraduate's user record resides in the Students domain. But the devices that are defined in the Undergraduates domain are not visible unless they are also defined in the Graduates, Students, or root domain.

You can affect an entire network or just a group of computers by choosing which domain to publish administrative data in. The higher the administrative data resides in a NetInfo hierarchy, the fewer places it needs to be changed as users and system resources change. Probably the most important aspect of NetInfo for administrators is planning NetInfo domains and hierarchies. They 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 NetInfo data.

#### **Binding**

*Binding* is the technique that sets up the subtree of domains visible to a Mac OS X computer.

Binding associates a child domain with a particular parent domain. In the education example, when an undergraduate's computer starts up, the local domain on the computer binds to the Undergraduates domain, the Undergraduates domain binds to the Students domain, and the Students domain binds to the root domain.

Because the subtree is initially set up at login, it is sometimes called a login hierarchy.

All the shared domains in a hierarchy could reside on the same server, or they could be distributed among multiple servers. The way you set up the binding would determine the actual NetInfo hierarchy.





There are three binding choices, each of which offers a different way for the computer hosting the child domain to locate a computer hosting its parent domain. The protocol you use to bind any two domains depends mainly on the topology of the network:

 Broadcast binding. This protocol, which is the default, is best for binding domains on two computers on the same subnet or on a local area network (LAN) configured for IP broadcast forwarding.

The child computer sends out an IP broadcast request for the computer hosting its parent. A NetInfo process on the parent computer recognizes its child and responds to the request, and binding occurs. If no parent computer responds to the child's broadcast, the child computer uses only its local NetInfo domain.

You must set up both the child and parent computers if you want to use broadcast binding. The remaining two protocols only require setting up the child computer.

• *Static binding*. The computer hosting the child domain locates its parent by using the parent computer's IP address and the NetInfo tag of the parent domain. The NetInfo tag identifies the directory where the domain's database resides. The tag is needed because there may be more than one domain hosted by the computer at the IP address.

With this technique, a child can locate its parent regardless of where the child computer is on a network, but may experience delays if the parent computer doesn't respond.

• *DHCP binding*. As with static binding, the child locates the parent computer by using the parent's IP address and NetInfo tag. But you configure your DHCP server to provide this information rather than enter it statically on each child computer. You can configure any DHCP server to provide this information. If your DHCP server is hosted by a Mac OS X Server, you can use Server Admin to configure the DHCP server for DHCP binding.

"Setting Up Local Domains of Network Users" on page 51 describes how to configure a Mac OS X computer to bind to a parent NetInfo domain. Binding two shared domains is described in "Setting Up Shared Domains in Deeper Hierarchies" on page 45.

You can adjust a computer's binding as required to support different users and environments:

- When you add a new computer to a network in which parent domains have been set up, you configure it to bind into the appropriate part of the NetInfo hierarchy.
- If the computer is transferred to a different user, you can change its binding to support the new user's needs.
- You can use the Network pane of System Preferences to define different locations when you want to use different NetInfo parents from the same computer. This approach is useful when you're using a portable computer and the bindings used at work are not appropriate when working offsite. You would select the offsite location in the Network pane before shutting down and disconnecting from the network, so when you turn your computer on at home, it does not try indefinitely to find its parent computer.

Once binding has occurred, Mac OS X processes interact transparently with NetInfo. Rebinding occurs when any network location or settings change or when network connections are lost, then re-established. If a parent domain becomes unavailable for any reason, many local processes, even opening applications that reside locally, may be delayed. You can use *replication* to minimize the potential of such delays.

#### Replication

To ensure the availability of your shared administrative data as well as improve the speed with which multiple computers can retrieve it, you can replicate NetInfo parent domains. When you replicate a shared domain, you set up a *master* and mirror its data to one or more *clones*, which reside on different computers.



Computers in the network can bind to any of the computers hosting the master or its clones. If one of those hosts is unavailable, another is used automatically. Closer computers offer faster response times, but any computer that hosts the domain can act as a backup resource for administrative data when any other computer becomes unavailable.

If the master computer is completely lost, one of the clones can be converted into a new master.

The master is the only version of the domain that can be modified. When administrative data needs to change, only the master is changed. The changes are automatically propagated to the clones, usually within seconds.

"Setting Up Replication" on page 53 provides more information about replication, including how to create and manage clones.

#### **Inside NetInfo**

When a Mac OS X computer starts up and domain binding occurs, a NetInfo daemon called *nibindd* starts. The nibindd daemon starts another daemon—netinfod—for each domain on the computer. Then nibindd listens for requests from netinfod processes asking for parents, checking for the appropriate netinfod process and initiating binding as required. Both nibindd and netinfod run in the background.

A third process related to NetInfo is called *lookupd*. It's the process used to interact with NetInfo when legacy UNIX software (such as the Terminal application) requests administrative information now stored in NetInfo. The lookupd process makes it possible for software that uses Posix or BSD calls to retrieve administrative information from NetInfo. lookupd searches through the NetInfo hierarchy as required to locate the information needed, then returns it to the process that requested it.

Every Mac OS X computer uses one instance of nibindd and lookupd, and one instance of netinfod for each domain on the computer. The netinfod process is sometimes referred to as a *NetInfo server*.

Each netinfod process manages interactions with a domain's NetInfo database. Information in a NetInfo database is organized into directories, which are specific categories of NetInfo records, such as users, machines, and mounts. For example, the users directory contains a record for each user defined in the domain. Here is the record for a user with the short name "admin," as viewed in NetInfo Manager:

	loca	al @ localhost.company	.com - /MyDomain/(null)	
	Q			
Directory Browser				/users/admin
1		config groups locations machines mounts networks printers protocols rpcs services users	Admin daemon john karen new nobody root unknown www	4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
irectory: admin		_		•
irectory: admin	Value(c)	_	)	)+(
irectory: admin Property uid	Value(s)	=		)+(
irectory: admin Property uid name	Value(s) 101 admin	-		)+  
irectory: admin Property uid name _writers_passwd	Value(s) 101 admin admin	_		)+
irectory: admin Property uid _writers_passwd realname	Value(s) 101 admin admin Server Administ	trator		)+[
irectory: admin Property uid name _writers_passwd realname passwd	Value(s) 101 admin admin Server Administ 39tFCDCtCmSv	trator Q		
irectory: admin Property uid name _writers_passwd realname passwd homedirstyletype	Value(s) 101 admin admin Server Administ 39tFCDCtCmSv 1282368364	trator Q		)+
irectory: admin Property uid name _writers_passwd realname passwd homedirstyletype home	Value(s) 101 admin admin Server Administ 39tFCDCtCm5y 1282368364 /Network/Serve	trator Q ers/MyComputer/Users/a	dmin	)+( 
irectory: admin Property uid name _writers_passwd realname passwd homedirstyletype home home_loc	Value(s) 101 admin Server Administ 39tFCDCtCmSv 1282368364 /Network/Serve <home_dir> <u< td=""><td>trator Q ers/MyComputer/Users/a Irl&gt;afp://MyComputer.co</td><td>dmin mpany.com/Users/<url> &lt; path</url></td><td>&gt;admin</td></u<></home_dir>	trator Q ers/MyComputer/Users/a Irl>afp://MyComputer.co	dmin mpany.com/Users/ <url> &lt; path</url>	>admin
irrectory: admin Property uid name _writers_passwd realname passwd homedirstyletype home_loc gid	Value(s) 101 admin Server Administ 39tFCDCtCm5v 1282368364 /Network/Serve <home_dir> <u< td=""><td>trator Q ers/MyComputer/Users/a rrl&gt;afp://MyComputer.co</td><td>dmin mpany.com/Users/<url> <path< td=""><td>&gt;admin</td></path<></url></td></u<></home_dir>	trator Q ers/MyComputer/Users/a rrl>afp://MyComputer.co	dmin mpany.com/Users/ <url> <path< td=""><td>&gt;admin</td></path<></url>	>admin
irrectory: admin Property uid name _writers_passwd realname passwd homedirstyletype home home_loc gid	Value(s) 101 admin admin Server Administ 39tFCDCtCm5v 1282368364 /Network/Serve <home_dir> <u 20</u </home_dir>	trator Q ers/MyComputer/Users/a rrl>afp://MyComputer.co	dmin mpany.com/Users/ <url> &lt; path</url>	>admin
irrectory: admin Property uid name _writers_passwd realname passwd homedirstyletype home home_loc gid	Value(s) 101 admin admin Server Administ 39tFCDCtCm5v 1282368364 /Network/Serve <home_dir> <u< td=""><td>trator Q ers/MyComputer/Users/a rrl&gt;afp://MyComputer.co</td><td>dmin mpany.com/Users/<url> &lt; path</url></td><td>&gt;admin</td></u<></home_dir>	trator Q ers/MyComputer/Users/a rrl>afp://MyComputer.co	dmin mpany.com/Users/ <url> &lt; path</url>	>admin
irrectory: admin Property uid name _writers_passwd realname passwd homedirstyletype home home_loc gid	Value(s) 101 admin admin Server Administ 39tFCDCtCm5v 1282368364 /Network/Serve <home_dir> <u< td=""><td>trator Q ers/MyComputer/Users/a ırl&gt; afp://MyComputer.co</td><td>dmin mpany.com/Users/<url> &lt; path</url></td><td>&gt; admin</td></u<></home_dir>	trator Q ers/MyComputer/Users/a ırl> afp://MyComputer.co	dmin mpany.com/Users/ <url> &lt; path</url>	> admin
irrectory: admin Property uid name _writers_passwd realname passwd homedirstyletype home home_loc gid Click the lock to	Value(s) 101 admin admin Server Administ 39tFCDCtCm5v 1282368364 /Network/Serve <home_dir> <u 20 make changes.</u </home_dir>	trator Q ers/MyComputer/Users/a rrl> afp://MyComputer.co	dmin mpany.com/Users/ <url> &lt; path</url>	>admin

Each record is a collection of properties. Each property has a key (listed in the Property column) and one or more values (shown in the Value(s) column). The key is used by processes to retrieve values. This user record, set up by defining the user with the Users & Groups module of Server Admin, has properties that are used to authenticate the user and locate the user's home directory, which resides on a Mac OS X Server:

- uid is the user ID of the user.
- name is the short name.
- realname is the user's full name.
- passwd is the user's password, encrypted using a one-way encryption algorithm so that it cannot be decrypted.

- homedirstyletype is used by Server Admin to distinguish among home directory styles none, local, and custom.
- home is the absolute path to the user's home directory.
- home\_loc is present if the home directory is on an Apple file server. Its value is a Mac OS X property list that contains the domain name of the AFP server where the home directory share point resides and the path, relative to the share point, to the home directory.
- gid is the user's primary group.

The user named "root" in a domain can change any of its properties or add new ones. Properties with the prefix "\_writers\_" list the short names of other users authorized to change the value of a particular property. For example, \_writers\_passwd is the short name of the user who can change this user's password (in this example, the user named "admin").

You can use NetInfo Manager, located in /Applications/Utilities, on any Mac OS X computer to view the administrative data in a NetInfo domain. It is one of several applications (discussed in the next section) that interact with NetInfo.

#### **Accessing and Manipulating NetInfo Data**

Mac OS X provides several applications for setting up NetInfo hierarchies, managing administrative data stored in NetInfo, and viewing NetInfo data.

#### **Defining NetInfo Domains**

When you first set up a Mac OS X computer, the local NetInfo domain is automatically created and populated. After initial setup, you can create shared domains:

- If your NetInfo hierarchy will consist of only local domains and a root domain, see "Setting Up the Root Domain of a Simple Hierarchy" on page 44.
- If your hierarchy will contain multiple levels of parent domains, see "Setting Up Shared Domains in Deeper Hierarchies" on page 45.

After creating the shared domains, you configure them into hierarchies.

#### **Configuring NetInfo Hierarchies**

"Setting Up Local Domains of Network Users" on page 51 describes how to configure a Mac OS X computer to bind to a parent NetInfo domain. Binding two shared domains is described in "Setting Up Shared Domains in Deeper Hierarchies" on page 45.

#### **Setting Up Search Policies**

When a process requests NetInfo administrative data, the default search policy is to search the login hierarchy, starting with the local domain, then proceeding toward the root domain until the needed data is located. Binding determines the order in which parent domains are searched.

If you want to extend administrative data searches on a particular Mac OS X computer, you have these options:

- Using Directory Setup, you can specify other NetInfo domains to search if the default search path does not lead to the data needed.
- You can also specify LDAP servers to search after searching the default hierarchy.

The *Mac OS X Server Administrator's Guide* tells you how to customize a computer's search policy.

*Note:* In the remainder of this document, "login hierarchy" is used to refer to the NetInfo domains visible after a user logs in, even though additional domains may be visible if the user has set up a custom search policy.

#### **Managing NetInfo Data**

NetInfo administrative data is created and updated automatically when you use Mac OS X administration applications:

- On computers running Mac OS X, settings you define using System Preferences automatically create and update NetInfo data used by various Mac OS X processes.
- On Mac OS X Server, you can also use System Preferences, but the preferred application for managing server users and services is Server Admin. For information on using Server Admin, refer to the online help for Server Admin and to the Mac OS X Server Administrator's Guide.

#### **Viewing NetInfo Data**

While you can use the Mac OS X administration applications to view NetInfo settings indirectly, if you want to view the exact information stored in a particular domain, use NetInfo Manager.

When you first start NetInfo Manager, which is located in /Applications/Utilities/, it displays information for the local domain. If the domain has a parent, you can click the globe at the top of the pane to view it. If that domain also has a parent, click the globe again, and so on until the root domain is displayed. Alternatively, you can open a domain by choosing Open from the Domain menu.

When the domain is open, select a directory. The illustration below shows the window for a domain called MyDomain. At the bottom of the window, you see the properties of the machines record of a computer named computer01. This computer has a record in the machines directory because it hosts a master or clone of the domain.

Directory Prowser			/machines/co	moutor0
/	<ul> <li>aliases config groups locations machines mounts printers users</li> </ul>	4 4 4 4 4 4	computer01 computer02 computer03 computer04 computer05 computer06 computer07 computer08 computer09	
4			computer 10	)+(
irectory: compute	r01		comnuter10	)+
<ul> <li>irectory: compute</li> <li>Property</li> <li>ip_address</li> </ul>	rr01 [Value(s) 192.16	8.12.12	computer 10	
<ul> <li>irectory: compute</li> <li>Property</li> <li>ip_address</li> <li>name</li> </ul>	rr01	8.12.12 er01	computer 10	)+

There are usually three properties associated with a machines record. The name property is the computer's host name. The ip\_address property is the IP address of the computer. The serves property identifies one or more NetInfo databases stored on that computer that are related to the domain being viewed. The serves property value includes the NetInfo tag, which is the directory where the database for a domain resides:

- The NetInfo tag of the domain being viewed, "network" in this example, is preceded by the notation "./".
- If the domain has a parent on the same computer, the parent's tag is also listed, preceded by "../". In this example, the domain has a parent that resides on computer01 in a domain with the tag "Company."

• If the domain has a child on the same computer, the child's tag is listed, preceded by the domain name and a /. In this example, computer01/local indicates that the domain has a child named computer01 that has the NetInfo tag "local."

**Important** While NetInfo Manager can be used to change NetInfo data, it is easy to make a mistake that can disable your computer. For example, a Mac OS X process may expect to find a property with a single value; if the property has multiple values, the process's response is unpredictable. Use NetInfo Manager as described in the Mac OS X documentation.

#### **Using Command Line Utilities**

Several command line utilities that interact with NetInfo are available through the Terminal application. To find out more about them, view their man pages.

Utility	Description
niload	Loads data from UNIX configuration files (such as /etc/passwd) into a NetInfo database.
nidump	Outputs data in a NetInfo database to a UNIX configuration file.
niutil	Reads from a NetInfo database and writes to one.
nidomain	Creates and destroys NetInfo databases. Tells you which domains are served from which databases by servers running on a particular computer.
nigrep	Searches all NetInfo domains for all instances of a string you specify.
nifind	Determines whether a particular directory exists. If it does, lists its number and contents for a particular domain or all domains in the hierarchy.
nireport	Lists values of all properties in all subdirectories as well as those in a specific directory and domain.
nicl	Creates, reads, or manages NetInfo data.

#### **The Importance of Planning**

The more information you store in NetInfo, the more control you have over your network, and the more users you can share the data with, while changing it in only one or very few locations. But the amount of control you enjoy depends on the effort you put into planning your NetInfo domains and hierarchies. The next chapter in this document provides information that should help you plan your NetInfo hierarchies.

C H A P T E R

## NetInfo Planning

The goal of NetInfo planning is to design a hierarchy of NetInfo domains that gives your Mac OS X users easy access to the network resources they need *and* minimizes the time you spend maintaining NetInfo data.

This chapter provides information that will help you decide what your NetInfo hierarchy should look like. It presents some general planning considerations, then focuses on issues related to distributing user and group information among NetInfo domains.

#### **General Planning Guidelines**

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



If you want to share information among Mac OS X computers, you need to set up at least a root domain.



Hierarchies this simple may be completely adequate when all your network computer users share the same resources, such as printers or share points that contain home directories or applications.

Larger, more complex organizations can benefit from a deeper NetInfo hierarchy.



NetInfo hierarchies that contain at least one shared domain between the local and root domains let you make NetInfo information visible to only subsets of a network's computers. In this example hierarchy, the administrator can tailor the users and resources visible to the community of Mac OS X computers by distributing them among six shared domains.

#### **Controlling NetInfo Data Visibility**

If you want certain NetInfo data to be visible to *all* computers in a NetInfo hierarchy, you'd store that data in the root domain of the hierarchy. To make NetInfo data visible only to a subset of computers, store it in a shared domain below the root domain.

You might want to set up shared domains to support computers used by specific groups within an organization. For example, you could make directories containing programming applications and files visible only to engineering computers. On the other hand, you could give technical writers access to directories that store publishing software and document files. If you want all employees to have access to each other's home directories, you would store mount records for all the home directories in the root domain.

#### **Simplifying Changes to NetInfo Data**

Organize NetInfo hierarchies so you minimize the number of places data has to change over time. 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
- directories being created and reorganized
- printers being moved among locations

Try to make each domain applicable to all the computers that bind to it so you don't have to change or add information in multiple domains. In the educational hierarchy example, all students may have user records in the Students domain and all employees have accounts in the Employees domain. As undergraduate students leave or become graduate students, or as employees are hired or retire, the administrator can make adjustments to user information by simply editing one domain.

If you have a widespread or complex NetInfo hierarchy in a network that is managed by several administrators, devise strategies to minimize conflicts. For example, set aside ranges of user IDs (UIDs) for specific groups of users within the organization to avoid granting file access to the wrong users (see "User Data Planning" on page 32 for more information).

#### **Identifying Computers for Hosting Shared Domains**

Identify the computers on which shared domains should reside. Shared domains affect many users, so they should reside on Mac OS X computers that

- have restricted physical access
- have limited network access
- are configured with features that make them constantly available, such as uninterruptible power supplies

It's best to locate shared domains on Mac OS X Servers. They provide more NetInfo data management applications than computers running Mac OS X.

Select computers that will not be replaced frequently and that have adequate capacity for growing domains. While you can move a domain after it has been set up, you may need to reconfigure computers that bind to the shared domains to ensure that the login hierarchies you originally established remain intact.

If a shared domain will support more than 100 Mac OS X computers, plan to clone the domain. Most of the time, you should store a clone on a different computer from the master domain's computer so that if one of the computers experiences problems, the domain on the other computer will still be available.

#### **Devising a Binding Strategy**

If you don't want certain computers to have access to certain information, do not store the information in shared domains the computer binds to. Conversely, if you want a computer to be able to access certain information, ensure the computer binds into domains that make it accessible.

There are three techniques for binding: broadcast, static, and DHCP. The strategies available for binding depend on your network topology. (Refer to "Binding" on page 19 for information about these strategies.) Broadcast and DHCP binding require a little extra planning:

- When you want a computer to locate its parent using broadcast binding, the parent must have the NetInfo tag "network." Since every domain that resides on a particular computer must have a unique NetInfo tag, this requirement affects how you spread your domains among computers and how you name their NetInfo tags.
- If you want to use DHCP binding, you first need to set up a DHCP server. You can use your existing DHCP server or use the DHCP module of Server Admin to configure the DHCP server that comes with Mac OS X Server.

#### **User Data Planning**

NetInfo user records have an important impact on a Mac OS X user's experience. They control which Mac OS X computer the user can log in to or connect to, and which home directory is mounted. They also affect the user's file access privileges.

This section provides information that will help you decide how and in which domain(s) to place NetInfo user records.

#### **Understanding the Login Environment**

When a Mac OS X computer starts, its NetInfo login hierarchy is set up. The computer's local domain binds to its parent domain (if it has one), and the parent and any parents it has are likewise bound together, creating a "tree" of NetInfo domains visible to processes running on the computer.

When a user logs in to the computer, all the user records in the login hierarchy are available for authenticating the user. NetInfo searches for a user record that contains the user name entered by the user in the login window, starting with the local domain and proceeding through the login hierarchy. If a user record that contains the login name is found, the password in the record is compared with the password entered by the user. If they match, the user is authenticated. If they don't match, NetInfo stops searching for a matching user record, and the user can't log in.

After a user is authenticated, the user is granted access to the computer, and all the resources in the login hierarchy are visible to the user. For example, any printer in the login hierarchy is visible in the Directory Services list in Print Center. And any mount records in the login hierarchy make directories visible in the user's Finder window under /Network.

In this example, the folder "homes," which resides on a computer named computer01, is visible to the user under /Network.



Each automatically mounted directory has a NetInfo mount record in one of the domains in the login hierarchy. You can view a mount record using NetInfo Manager by selecting the mounts directory, then selecting a computer and share point. Here is one of the mount records for the automatically mounted directories visible under /Network/Servers in the Finder window on the previous page.

		/(				
Directory Browser		/mounts/computer01:/homes/dir				
*	config databases fax_modems groups licenses locations machines	computer01:/homes/dir2 computer02:/Users computer03:/homes/dir1 computer03:/homes/dir2 computer04:/Users computer05:/Users				
•	mounts	computer06:/Users	1			
↓	mounts	computer06:/Users				
irectory: gimli     Property     vístype	value of s	computer06:/Users				
<ul> <li>Interctory: gimli</li> <li>Property</li> <li>vfstype</li> <li>passno</li> </ul>	value nfs 0	(s)				
Property Vistype passno dir	Value 0 (Value	(s)				
Property Vfstype passno dir dumo_freg	Value nfs 0 /Net	(s)				
	Value nfs 0 /Net 0 com	(s) work/Servers				

The value of the "dir" property controls where in the Finder the directory is visible to the user. For example, a directory visible under /Network/Applications in the Finder would have a mount record with a "dir" property value of /Network/Applications in at least one of the domains of the login hierarchy.

Although *any* user who can log in to a particular Mac OS X computer can view the directories and resources associated with domains in the computer's login hierarchy, *each* user's NetInfo user record determines several aspects of the login environment:

- The UID in the record determines the files or operations the user has access to.
- The primary group ID associated with the user record also affects a user's file access privileges. If the user accesses a file that isn't owned by the user, the file system checks the file's group privileges. If group privileges have been granted to the user's primary group, the user inherits those privileges.
- The home directory associated with the user determines system preferences and access to the user's personal directories and files. The home directory is visible when you click Home in a Finder window or in the Finder's Go menu.

Because the NetInfo user record that is used to authenticate a user plays an important role, be sure to create user records in NetInfo domains accessible from any Mac OS X computer you want the user to be able to log in to.

#### **Contrasting Logging In and Connecting**

Also plan to create user records in NetInfo domains accessible from any Mac OS X computer you want the user to be able to *connect to after login*.

When a user chooses Connect To Server from the Finder's Go menu, NetInfo looks for a user record starting in the local domain on the remote server and proceeding through the NetInfo hierarchy used by the remote server. If a matching user record is found, the server is mounted on the user's login computer.

The UID or primary group ID associated with the user record used to authenticate the user on the remote server determines the user's access privileges to directories and files on that server, but has no effect on the user's login hierarchy or home directory.

#### **Managing Names**

This section provides some guidelines to remember when defining the names associated with any user record.

#### **Avoid Duplicate Names**

If separate NetInfo user records have the same name and password, a Mac OS X computer may authenticate a user different from the one you want it to authenticate, or mask the user record that should be used for authentication.

For example, user records for Tony Smith and Tom Smith contain the short name "tsmith" and the password "smitty."



When Tony logs in to his computer with a user name "tsmith" and the password "smitty," he is authenticated using the record in the Students domain. Similarly, Tom can use the same login entries at his computer and be authenticated using his record in the root domain. If Tony and Tom ever logged into each other's computers using tsmith and smitty, they would both be authenticated, but not with the desired results. Tony could access Tom's files, and vice versa.

Now let's say that Tony and Tom have the same short name, but different passwords.



If Tom attempts to log in to Tony's computer using the short name "tsmith" and his password (smitty), his user record is masked by Tony's user record in the Students domain. NetInfo finds "tsmith" in Students, but its password does not match the one Tom used to log in. Tom is denied access to Tony's computer, and his record in the root domain is never found.

If Tony has a user record in his local domain that has the same names and password as his record in the Students domain, the Students domain's record for Tony would be masked. Tony's local domain should offer a name/password combination that distinguishes it from the Students domain's record. If the Students domain is not accessible (when Tony works at home, for example), he can log in using the local name and continue using his computer. Tony can still access local files created when he logged in using the Students domain if the UID in both records is the same.

Duplicate short names also have undesirable effects in group records, described in "Avoiding Duplicate Short Names" on page 40.

#### **Choose Stable Short Names**

Try to use short names that won't change even if a user changes his or her real name. When you create groups, users in them are identified by their short names. When a user's short name changes, any groups to which the user belongs must be edited to reflect the change so that directory and file access remain consistent.

If a short name change is unavoidable, you can create a new record for the user (in the same domain) that contains the new short name, but retains all other information (UID, primary group, home directory, and so forth). Then disable login for the old user record. Now the user can log in using the changed name, yet have the same access to files and other network resources as before.

#### **Managing UIDs**

The UID is a critical element in ensuring users have full access to the directories and files they create as well as to their home directories.

Most of the time, individual users should have unique UIDs. Assigning the same UID to different user records is risky unless you have a specific reason for doing so, such as to support a new short name as described above. Two users with the same UID have identical directory and file access privileges.

Devise a UID strategy that will minimize the likelihood of different users having the same UID. You can, for example, reserve a range of UIDs for use in each shared domain. Remember the following points:

- The UID 0 is reserved for the root user.
- The maximum UID is 2,147,483,647.
- UIDs below 100 are reserved for system use.
- Users created in the Users pane of System Preferences are automatically assigned UIDs starting with 500. You can change these UIDs using NetInfo Manager.
- Users created in the Users & Groups module of Server Admin are automatically assigned UIDs starting with 100. These UIDs can be changed in the Users & Groups module.

Once UIDs have been assigned and users start creating files and directories throughout a network, you shouldn't change UIDs. So ensure that from the beginning you have a plan for UID management.

#### **Setting Up Home Directories**

Identify the computers on which you want user home directories to reside. You may want to store home directories for users with last names from A to F on one computer, G to J on another, and so on. Whatever strategy you pick, decide on one before creating a lot of users. You can move home directories, but if you do, you may need to change a large number of user and share point records.

Once you have decided how many computers you want to use for home directories, plan the domain name or IP address of each computer. Also determine the names and any share points on the computer that will be used for home directories. You'll need this information to set up home directories when you define users with the Users & Groups module of Server Admin on a Mac OS X Server.

Now you are ready to define users, either individually or by importing them. Even if the computers on which the user home directories will reside will not be available for a while, you can begin the process of setting up users.

Refer to the *Mac OS X Server Administrator's Guide* for information about using Server Admin to define users, home directories, and share points.

#### **Group Data Planning**

Groups are used to assign directory and file access privileges to collections of users.

Here is what a group record looks like in NetInfo Manager. It is a simple record that contains only the name of the group, the group ID (GID), and a list of the short names of users who are members of the group.

	$\odot$	39			
Directory Brow	ser			/group	os/Students
/	Þ	afpuser_aliases aliases config groups locations machines mounts networks printers		nogroup operator staff Students sys Teachers tty unknown utmp	
) irectory: Stude	nts	_			
Property	nts	Value	(5)		
Pirectory: Stude Property name	nts	Value Stude	(s) ents		
Frectory: Stude Property name gid	nts	Value Stude 102	(s) ents		
Pirectory: Stude Property name gid ▶users	nts	Value Studi 102 (john	<b>(s)</b> ents , mary)		

This section provides information to help you decide how and in which domain(s) to place NetInfo group records.

#### **Ensuring Group Visibility**

All directories and files on Mac OS X computers have access privileges for the file's owner, a group, and everyone else.

ĺ	
l	MyDoc
l	

owner 127 can: Read & Write group 2017 can: Read only everyone else can: None When a user attempts to access a directory or file the user doesn't own, group privileges are checked:

- First the GID of the user's primary group is compared with the GID associated with the directory or file. If they match, the user is granted group access privileges.
- If they do not match, NetInfo searches through the login hierarchy for a group record with a matching GID, starting with the local domain and proceeding toward the root domain.

If NetInfo finds a matching group record, it searches the login hierarchy to map each short name in the group record to a UID. If the user's UID matches one of the UIDs found, the user is granted group access privileges.

Plan to create group and related user records in NetInfo domains accessible from any Mac OS X computer you want the user to be able to log in to or connect to.

#### **Avoiding Duplicate Short Names**

Since short names are used to find UIDs of group members, duplicate short names can result in file access being granted to users you hadn't intended to give access.

Return to the example of Tony and Tom Smith, who have duplicate short names. Assume that the administrator has created a group in the root domain to which all students belong. The group—AllStudents—has a GID of 2017.



Now suppose that a file, MyDoc, resides on a computer accessible to both Tony and Tom. The file is owned by a user with the UID 127. It has read-only access privileges for AllStudents. Tom is not a member of AllStudents, but the short name in his user record, "tsmith," is the same as Tony's, who *is* in AllStudents.

When Tom attempts to access MyDoc, NetInfo searches the login hierarchy for user records with short names that match those associated with AllStudents. Tom's user record is found because it resides in the login hierarchy, and the UID in the record is compared with Tom's login UID. They match, so Tom is allowed to read MyDoc, even though he's not actually a member of AllStudents.

#### **The Next Step**

After you have decided what the logical and physical topology of your NetInfo hierarchy should look like and which administrative information to store in shared domains, you are ready to set up your hierarchy. The next chapter tells you how.

CHAPTER 3

## Setting Up NetInfo Hierarchies

After you have decided what the topology of your NetInfo hierarchy should look like and identified which computers will host shared domains, you are ready to create the hierarchy and populate its domains with records.

#### **The Overall Process**

These are the main steps for setting up NetInfo hierarchies:

#### Step 1: Set up shared domains

On each computer you want to host shared domains, create the domains and configure them so that they are able to bind into the hierarchy you want.

- If you want a hierarchy that contains only one shared domain, a root domain, follow the instructions in "Setting Up the Root Domain of a Simple Hierarchy" on page 44.
- If you want a hierarchy with at least one shared domain between local and root domains, see "Setting Up Shared Domains in Deeper Hierarchies" on page 45.

#### Step 2: Set up local domains

Set up the local domain on each Mac OS X computer so that it binds to the appropriate shared domain. See "Setting Up Local Domains of Network Users" on page 51.

#### **Step 3: Set up replication**

If you want to replicate shared domains for performance and reliability, define clones. See "Setting Up Replication" on page 53.

#### Step 4: Set up Windows user authentication

If Windows users need to be authenticated using NetInfo, follow the instructions in "Setting Up Windows User Authentication" on page 56.

#### **Step 5: Populate domains**

Add user, group, mount, and printer records to the appropriate domains in your hierarchy. See "Populating Domains" on page 60.

#### **Setting Up the Root Domain of a Simple Hierarchy**

The simplest NetInfo hierarchy consists of a root domain and one or more local domains that bind to it.



To create the root domain, use NetInfo Domain Setup on Mac OS X Server:

- 1 As the root user, log in to the server where the root domain will reside.
- 2 Ensure that the server has a valid Domain Name System (DNS) entry in your DNS server. For example, if the IP address of the server that will host the root domain is 192.168.12.12 and its DNS name is server.apple.com, you need a corresponding entry in the DNS server that maps the server's IP address to the DNS host name. *Mac OS X Server Administrator's Guide* provides information on DNS.
- 3 Open NetInfo Domain Setup, located in /Applications/Utilities. Click the lock icon to log in. In the first authentication dialog, enter a server administrator name and password. In the second dialog, enter the root user name and password.
- 4 Choose "is a NetInfo Parent" from the "This machine" pop-up menu. "Static Address" should be selected in the "Find NetInfo Parent via" pop-up menu and "network" should appear in the NetInfo Server Tag field. Enter the server's IP address in the NetInfo Parent Address field.
- **5** If users of Windows computers need to be authenticated using NetInfo, follow the instructions in "Setting Up Windows User Authentication" on page 56.

6 Click Save, then click OK when a message tells you to restart the computer. It will take a few minutes for your changes to be completed. When changes are complete, click the Close button in the NetInfo Domain Setup window and restart the computer.

#### **Setting Up Shared Domains in Deeper Hierarchies**

When you need to constrain the visibility of some of your NetInfo data, use a hierarchy with more shared domains than just a root domain.

In this example, two shared domains beneath the root domain—Math and English—limit the data visible to processes running on Math and English student computers. The root domain and the Math domain reside on one Mac OS X Server (server1) and the English domain resides on a second server (server2).



To set up the shared domains in hierarchies such as these, you create the domains on the computers where they will reside, then configure NetInfo machine records for each parent domain.

#### **Understanding Machine Records**

Machine records provide the information needed for a child domain to bind to its parent:

- The child domain needs to find the shared domain that can serve as its parent, and
- The shared domain needs to determine whether it matches the description of the parent the child is looking for.

Machine records are located in the machines directory of a domain's NetInfo database.

Here is a summary of requirements for machine records in a NetInfo domain:

• The machines directory of every shared domain must have a record for every computer where the domain resides.

These machine records contain a serves property having the value ./*tag*, where ./ refers to the current domain and *tag* is the NetInfo tag of the domain.

• The machines directory of every shared domain that has a parent must have a record for every computer hosting its parent.

These machine records contain a serves property having the value ../*tag*, where ../ refers to the parent domain and *tag* is the NetInfo tag of the parent domain.

- The machines directory of every shared domain that has one or more children must contain a machine record for each child's computer if the child is one of the following:
  - a shared domain
  - a local domain that uses broadcast binding

The serves property of these machine records has the value *domain/tag*, where *domain* is the name of the child domain and *tag* is the child's NetInfo tag.

The root domain on server1 in the example above needs two machine records, one for server1 (to identify the root domain and one of its children, Math) and one for server2 (to identify its other child, English). Because it is the root, it has no machine record for a parent domain.

Property	Value(s)	
ip_address	192.168.12.12	
name	server1	- 8
▼ serves	(Math/network, ./myschool)	
	Math/network	
	./myschool	- 8
-		
	Value(s)	
Property		
Property ip_address	192.168.12.13	
Property ip_address name	192.168.12.13 server2	

In this example, "myschool" is the NetInfo tag of the root domain, and "network" is the NetInfo tag of the Math and English domains.

The English domain, which resides on server2, also needs two machine records—one to identify its parent (the root domain) and one for itself. Here is the English domain's machine record that identifies its parent. The NetInfo server for the English domain can find the NetInfo server for its parent domain by sending a message to server1 that looks for a netinfod process for "myschool."

Property	Value(s)
ip_address	192.168.12.12
name	server1
serves	/myschool

The English domain also needs a machine record for each local domain that uses broadcast to find it.

#### **Defining Shared Domains**

This section uses the example above. The steps below create the shared domains that reside on server1 and server2 and ensure that they contain the required machine records.

Several of the steps instruct you to use the command line utilities nidomain and niutil. For complete information about these commands, view their man pages through the Terminal application.

#### Step 1: Create the future Math domain

In this step, you create a root domain on server1 using NetInfo Domain Setup. Later, this domain will become the Math domain.

- 1 Ensure that server1 has a valid DNS entry in your DNS server.
- 2 Log in as root to server1.
- 3 Open NetInfo Domain Setup. Click the lock icon to log in. In the first authentication dialog, enter a server administrator name and password. In the second dialog, enter the root user name and password.
- 4 Choose "is a NetInfo Parent" from the "This machine" pop-up menu. "Static Address" should be selected in the "Find NetInfo Parent via" pop-up menu and "network" should appear in the NetInfo Server Tag field. Enter the server's IP address in the NetInfo Parent Address field.
- 5 Click Save, then click Quit when it becomes active. Restart server1.

#### Step 2: Define the root domain and its relationship to the Math domain

On server1, you now have a local domain that is configured to bind statically to a root domain. Use the following procedure to create the actual root domain you want and define the Math domain as a child of the root domain:

- 1 Log in as root to server1.
- 2 Open the Terminal application, located in /Applications/Utilities/, and enter the following niutil command to create the new root domain. The argument "myschool" will be the root domain's NetInfo tag:

```
[server1:~] root# nidomain -m myschool
```

NetInfo creates a domain that contains a machine record for server1. The name property of the record is "server1," and its ip\_address property is the IP address of server1.

3 Add a serves property to server1's machine record in the new domain to indicate it serves a domain named Math that has the NetInfo tag "network:"

[server1:~] root# niutil -createprop -t server1/myschool
/machines/server1 serves./myschool Math/network

- "-createprop" is the niutil command for creating a new property (or overwriting an existing one).
- "-t server1/myschool" identifies the database in which you want to create the property. Since it is not yet connected to a hierarchy, you must identify it by host name and NetInfo tag.
- "/machines/server1" indicates you want to create the new property in the record named server1 in the machines directory of the new domain.
- "serves" indicates that you want to create a serves property.
- "./myschool" provides the first value for the new serves property. The value indicates that server1 serves the current domain (.) from the database with the NetInfo tag "myschool."
- "Math/network" provides the second value for the serves property. It indicates that server1 also serves a domain named Math from the database with the NetInfo tag "network."
- 4 Now add a serves property with three values to the machine record named server1 in the Math domain, referred to using its NetInfo tag, network, and the notation for current (.):

```
[server1:~] root# niutil -createprop -t server1/network
/machines/server1 serves server1/local ./network ../myschool
```

The three serves property values indicate that server1 serves the Math domain's child (server1) from the database tagged local, the current domain (.) from the database tagged network, and the Math domain's parent (..) from the database tagged myschool.

**5** Use nidomain's list command to verify that you now have three domains on server 1:

```
[server1:~] root# nidomain -l
tag=network udp=768 tcp=769
tag=local udp=766 tcp=767
tag=myschool udp=854 tcp=855
```

You can also list all the netinfod processes running on server1. There is one for each domain:

[server1:~] root# ps aux | grep netinfod 164 0.0 0.4 1784 504 ?? S 0:00.47 root netinfod local (master) 165 0.0 0.3 2684 452 ?? S 0:00.87 root netinfod myschool (master) 166 0.0 0.3 root 1784 424 ?? S 0:00.30 netinfod network (master) 272 0.0 0.1 root 5708 196 std RV 0:00.00 grep netinfod

6 When you create a new domain using niutil, it has only two directories in its database: / and machines. Add additional directories to the root domain's database:

[server1:~]	root#	niutil	-create	-t	server1/myschool	/users
[server1:~]	root#	niutil	-create	-t	server1/myschool	/groups
[server1:~]	root#	niutil	-create	-t	server1/myschool	/aliases
[server1:~]	root#	niutil	-create	-t	server1/myschool	/mounts
[server1:~]	root#	niutil	-create	-t	server1/myschool	/printers

- 7 Restart server1 and create a root and an administrator account in the root domain so that it can be modified from anywhere on the network:
  - a Open Server Admin, click the General tab, then click Users & Groups.
  - Choose New User and select /NetInfo/root from the pop-up menu. Enter the information that describes the root user, including a short name of "root" and a UID and primary group ID of 0. Then click Save.
  - c Repeat step 7b to create a user who is an administrator.

#### Step 3: Create the future English domain

In this step, you create a root domain on server2 using NetInfo Domain Setup. Later, this domain will become the English domain.

- 1 Ensure that server2 has a valid DNS entry in your DNS server.
- 2 Log in as root to server2.
- 3 Open NetInfo Domain Setup. Click the lock icon to log in. In the first authentication dialog, enter a server administrator name and password. In the second dialog, enter the root user name and password.
- 4 Choose "is a NetInfo Parent" from the "This machine" pop-up menu. "Static Address" should be selected in the "Find NetInfo Parent via" pop-up menu and "network" should appear in the NetInfo Server Tag field. Enter the server's IP address in the NetInfo Parent Address field.
- 5 Click Save, close NetInfo Domain Setup, and restart server2.

#### Step 4: Add a machine record for the root domain to the future English domain

Because the parent of the English domain resides on a different computer, you must add a machine record for server1 to the English domain to identify the parent.

- 1 Open NetInfo Manager on server2 and open the network domain. You can open the network domain by clicking the globe at the top of the NetInfo Manager window or choosing Open Parent from the Domain menu.
- 2 Click the lock icon and log in as the root user.
- 3 Select the machines directory in the Directory Browser list.
- 4 Choose New Subdirectory from the Directory menu. A property called "name" is created, with a default value of new\_directory. Double-click new\_directory in the lower list and enter server1.
- 5 Choose New Property from the Directory menu. Double-click new\_property and change it to "ip\_address."
- 6 Choose New Value from the Directory menu. Double-click new\_value and enter the IP address of server1.
- 7 Choose New Property from the Directory menu. Double-click new\_property and change it to "serves."
- 8 Choose New Value from the Directory menu. Double-click new\_value and enter "../ myschool".
- 9 Choose Save from the Domain menu and click Update Copy in the dialog that appears.

#### Step 5: Add a machine record for the English domain to the root domain

On server1, add a machine record to the root domain to identify the English domain. Use the same process as in step 4, but set up the machine record properties like this:

- The name should be server2.
- The IP address should be server2's IP address.
- The serves property should be English/network.

#### **Step 6: Restart both servers**

Restart server1, then restart server2 to ensure that all your NetInfo changes take effect.

#### Step 7: Set up Windows user authentication

If users of Windows computers need to be authenticated using NetInfo, set up authentication as described in "Setting Up Windows User Authentication" on page 56.

#### **Setting Up Local Domains of Network Users**

To configure the local domain of a Mac OS X computer to bind to the shared domain that should be its parent, use the Directory Setup application on the local domain's computer.

A computer can bind to its parent using broadcast, static, or DHCP binding. You can also configure a computer with multiple binding options so that if a parent is not located with one, another one is used. NetInfo attempts to locate the parent using chosen options in this order: static, DHCP, broadcast.

#### **Static Binding**

Static binding is most commonly used when the parent domain's computer is not on the same IP subnet as the computer that needs to access it.

Use this procedure to configure a Mac OS X computer to use static binding:

- 1 On the Mac OS X computer, open Directory Setup, located in /Applications/Utilities. Click the lock icon and log in as the local administrator.
- 2 Select NetInfo and click Configure.
- 3 Choose "Attempt to connect to a specific NetInfo server."
- 4 Enter the IP address of the parent domain's computer in the Server Address field.
- **5** Enter the parent domain's NetInfo tag in the Server Tag field.
- 6 Click OK.
- **7** Restart the computer.

#### **DHCP Binding**

When you configure a Mac OS X computer to locate its parent using DHCP binding, the parent's IP address and NetInfo tag are provided by a DHCP server rather than using information you enter using Directory Setup. To use DHCP binding, you must first configure a DHCP server to provide this information.

Use this procedure to configure a Mac OS X computer to use DHCP binding:

- 1 Configure a DHCP server to provide the IP address and NetInfo tag of the parent domain. If the DHCP server is hosted by a Mac OS X Server, use these steps.
  - a On the Mac OS X Server hosting the DHCP server, open Server Admin and log in as a server administrator.
  - b Click the Network tab, click DHCP/NetBoot, and choose Configure DHCP.
  - c Select a subnet, then click Edit.
  - d Click the NetInfo tab.
  - In the NetInfo Tag field, enter the NetInfo tag of the parent domain you are setting up binding for.
  - **f** In the NetInfo Parents field, enter the IP address of the computer on which the parent domain resides.
  - g Click Save.
- 2 On the Mac OS X computer for which you want to configure binding, open Directory Setup. Click the lock icon and log in as the local administrator.
- 3 Select NetInfo and click Configure.
- 4 Choose "Attempt to connect using DHCP protocol."
- 5 Click OK.
- 6 Restart the computer.

#### **Broadcast Binding**

This technique is the default if multiple binding options are not configured. The two computers that need to bind must be on the same subnet or on a LAN configured for IP broadcast forwarding. Also, the parent domain must have the NetInfo tag "network."

Follow these steps to configure a Mac OS X computer to bind to a parent domain using broadcast binding:

- 1 Add a machine record for the Mac OS X computer to the parent you want it to bind to:
  - a Open NetInfo Manager on the computer where the parent domain resides, then open the domain.
  - **b** Click the lock icon and log in using the user name and password specified when the domain was created.

- c Select the machines directory in the Directory Browser list.
- d Choose New Subdirectory from the Directory menu. Double-click new\_directory in the lower list and enter the DNS name of the child computer.
- Choose New Property from the Directory menu. Double-click new\_property and change it to "ip\_address."
- f Choose New Value from the Directory menu. Double-click new\_value and enter the IP address of the child computer.
- g Choose New Property from the Directory menu. Double-click new\_property and change it to "serves."
- h Choose New Value from the Directory menu. Double-click new\_value and enter the name and NetInfo tag of the child's local domain, separated by a "/", for example, marketing.demo/local. Press Return.
- i Choose Save from the Domain menu, then click Update This Copy.
- 2 On the Mac OS X computer for which you want to configure binding, open Directory Setup. Click the lock icon and log in as the local administrator.
- 3 Select NetInfo and click Configure.
- 4 Choose "Attempt to connect using Broadcast protocol."
- 5 Click OK, then click Apply.
- 6 Restart the computer.

#### **Setting Up Replication**

NetInfo lets you replicate shared domains to improve reliability and speed of access to their data. Each domain has a master server. Additional servers for the domain are called *clones*.

Usually you configure at least one clone for every shared domain. You need multiple clones when a shared domain is needed by more computers than a master and a single clone can support.

This section briefly describes some of the characteristics of NetInfo replication, then tells you how to create clones and how to replace a master with a clone.

#### **Distinguishing Masters**

The master is distinguished by a property named "master" in the root directory of every domain. The master property's value consists of the DNS name of the master's computer, followed by a "/", then the NetInfo tag of the master domain.

*Note:* The values for the master property and a machine record's serves property for a child domain appear identical, but they are different. The name that precedes the "/" is a *bost name* for a master property but a *domain name* for a serves property.

#### **Locating and Using Masters and Clones**

When a Mac OS X process requests information from NetInfo:

- The parent for the local domain is located using the binding information set up in Directory Setup.
- Other parent domains are located by searching the machines directory of the domain for records with a serves property of the form *../parent-tag*, where *parent-tag* identifies the NetInfo tag of the parent domain.

If the parent has clones, multiple machine records identify computers serving the domain.

When attempting to communicate with a domain, NetInfo first uses servers for the domain running on the local computer, then other computers on the same subnet, and finally other computers that serve the domain.

If a connection between domains is lost, NetInfo re-establishes a new connection the next time NetInfo data is requested.

#### **Creating Masters**

The first time you create an instance of a domain, NetInfo sets up the master property for it. You do not have to manage this property yourself. When you follow the instructions in "Setting Up Shared Domains in Deeper Hierarchies" on page 45, you will automatically set up all your masters.

#### **Creating Clones**

These steps create a clone of the root domain in the example that has been used in this chapter. Since the root domain resides on server1, the clone will be created on server2.

- Log in as root to server2.
- 2 Open the Terminal application. Enter the following niutil command to add a serves property to the machines/server2 record in the root domain on server1 identifying server2 as a clone server for ./myschool.

```
[server2:~] root# niutil -createprop -u root / /machines/server2
serves English/network ./myschool
```

Password: enter-server1-root-password

3 Overwrite the serves property of the English domain's server2 record to identify the future clone by adding a new value, ../myschool:

[server2:~] root# niutil -createprop -t server1/network
/machines/server2 serves ../myschool ./network server2/local

4 Create the clone:

```
[server2:~] root# nidomain -c myschool server1/myschool
```

This command copies the database tagged "myschool" from its host (server1) to a database having the same tag on the current computer, server2.

**5** Restart server2. Now server2 hosts two master domains and a clone:

[server2	:~] ro	ot# p	s aux	grep net	infod			
root netin	183 nfod l	0.0 ocal	0.4 (master	1784 c)	500	??	S	0:00.34
root netin	184 nfod m	0.0 yschod	0.3 pl (clo	2168 one)	404	??	S	0:00.12
root netin	185 nfod n	0.0 etwor	0.3 c (mast	1784 ter)	408	??	S	0:00.16
root grep	271 netin	0.0 fod	0.2	1384	304	std	S+	0:00.02

#### **Replacing a Master With a Clone**

If a computer hosting a NetInfo master database becomes unusable, a clone can be used to replace it. Follow these steps:

- 1 Disconnect the computer hosting the damaged master from the network.
- 2 Log in as the root user to one of the computers hosting a clone of the damaged master.
- 3 Edit /etc/hostconfig and change the values of the name variable to the name of the master database.
- 4 Use the Network pane of System Preferences to change the computer's IP address to match that of the computer hosting the damaged master domain.
- 5 Restart the computer. The computer takes the name and IP address of the former master's computer and now hosts the master domain. Depending on your network configuration, this computer might need to be moved to the location of the former master's computer.
- 6 If the former master computer hosted home directories, ensure that they are re-created on the new master's computer.
- 7 Although the clone now functions as a master, NetInfo has an entry for it as a clone. To remove the entry, in NetInfo Manager open the former clone and delete the "./network" value from the serves property in the machines subdirectory. Since there is no longer a machines entry with the former clone's name, it should no longer be recorded as a clone server for the master domain.

If you reconnect the former master's computer to the network, use a different host name and IP address so they don't conflict with those of the new master domain's computer.

#### **Setting Up Windows User Authentication**

To authenticate Windows users using NetInfo so they can take advantage of the Windows services on Mac OS X Server, ensure that all the shared domains in your hierarchy reside on Mac OS X Servers.

Mac OS X Server has two ways to validate Windows users' passwords:

• *Encrypted password validation* is preferred because it is the safest and because it is the default technique supported by Windows computers on a local area network (LAN). This technique transmits encrypted passwords between a Windows computer and Mac OS X Server.

To use encrypted password validation, you enable Authentication Manager for *all* domains in the hierarchy and define an encryption key for each domain. When Authentication Manager is enabled, a tim\_passwd property is stored in NetInfo user/Manager records. It can be decrypted to get the cleartext password using the encryption key, which is stored in a file on the server that is readable only by root.

 Cleartext password validation should be used only when encrypted transmission of user authentication information is not important. Windows computers must be configured individually to support cleartext password validation. See the Windows documentation for information on how to set up cleartext password validation.

When you use cleartext password validation, passwords are not stored in a recoverable format. The NetInfo password value, associated with the "passwd" property, is derived using a one-way hash, which can't be easily decoded. The one-way hash ensures that each time it's used for the same password, the same result occurs.

To set up encrypted password validation, enable Authentication Manager on *every* Mac OS X computer that participates in the hierarchy. How you accomplish this depends on how many shared domains are in your hierarchy:

- If a hierarchy has only a root domain that is not cloned, use the procedure in the next section.
- Otherwise, use the procedure in "Other Hierarchies" on page 57.

For information about Windows services on Mac OS X Servers, refer to *Mac OS X Server Administrator's Guide.* 

#### **Simple Hierarchies With No Clones**

Enable Authentication Manager on the root domain's server:

- 1 Log in to the server as the root user.
- 2 Open NetInfo Domain Setup.
- 3 Click the lock icon. In the first authentication dialog, enter a server administrator name and password. In the second dialog, enter the root user name and password.

- 4 Check the Authentication Manager box.
- **5** Click Save and close NetInfo Domain Setup.
- 6 Restart the server. Authentication Manager is now enabled for both the local and the root domains.
- 7 Open Server Admin and use the Users & Groups module to reset passwords of existing users who will be using Windows computers. New users are automatically set up for encrypted password validation.

Then enable Authentication Manager on each additional Mac OS X computer whose local domain binds to the root domain:

- 1 As the root user, log in to the computer.
- 2 Open the Terminal application, located in /Applications/Utilities.
- 3 Enter the following command line, where "local" is the NetInfo tag for a local domain:

[computer1:~] root# tim -init -auto local

4 When prompted, enter and re-enter an encryption key:

Password for local:

Re-enter to verify:

Initialize service for local: Operation Succeeded

Enable autostart for local: Operation Succeeded

**5** In the computer's /etc/hostconfig file, ensure that this line exists:

AUTHSERVER=-YES-

6 Restart the computer or start Authentication Manager by entering this command line in the Terminal application:

[computer1:~] root# tim

7 Repeat steps 1 through 6 for each additional Mac OS X computer that uses the root domain.

#### **Other Hierarchies**

In hierarchies that have several levels of shared domains and in hierarchies that use clones, first enable Authentication Manager on every server hosting a shared domain that is *not* a clone, then enable Authentication Manager on each clone's server. Finally, enable Authentication Manager on every additional Mac OS X computer that has a local domain that binds into the hierarchy.

#### **Set Up Masters**

Use this procedure for servers hosting masters:

- 1 Log in as the root user to the server where the master resides.
- 2 Open the Terminal application, located in /Applications/Utilities.
- 3 Enter the following command line, substituting the NetInfo tag name of the master for tag: [server1:~] root# tim -init -auto tag
- 4 When prompted, enter and re-enter an encryption key:

```
Password for tag:
```

```
Re-enter to verify:
```

Initialize service for tag: Operation Succeeded

Enable autostart for tag: Operation Succeeded

- **5** Repeat steps 3 and 4 for the local domain and each additional master on the same server.
- 6 In the server's /etc/hostconfig file, ensure that this line exists:

AUTHSERVER=-YES-

7 Restart the server or start Authentication Manager by entering this command line in the Terminal application:

[server1:~] root# tim

8 Repeat steps 1 through 7 for each additional server hosting a shared domain in the hierarchy that is not a clone.

#### **Set Up Clones**

Enabling Authentication Manager for each clone is somewhat simpler, because NetInfo masters propagate information to their clones automatically. To set up Authentication Manager for clones:

- 1 Copy the file containing the master's encryption key to the clone's /var/db/netinfo/ directory. You'll find the master's file in /var/db/netinfo/.tag.tim, where "tag" is the NetInfo tag of the master.
- 2 Ensure that the /etc/hostconfig file on the clone's server contains this line:

AUTHSERVER=-YES-

3 Restart the clone's computer or start Authentication Manager by entering this command line in the Terminal application:

[server1:~] root# tim

4 Repeat steps 1 through 3 for each server on which clones reside.

#### Set Up Local Domains on Other Mac OS X Computers

Use the following procedure to enable Authentication Manager for the local domain on any other Mac OS X computers that will bind into the hierarchy:

- 1 Log in to the computer as the root user.
- 2 Open the Terminal application, located in /Applications/Utilities.
- 3 Enter the following command line, where "local" is the NetInfo tag for a local domain:

[computer1:~] root# tim -init -auto local

4 When prompted, enter and re-enter an encryption key:

Password for local:

Re-enter to verify:

Initialize service for local: Operation Succeeded

Enable autostart for local: Operation Succeeded

**5** In the computer's /etc/hostconfig file, ensure that this line exists:

AUTHSERVER=-YES-

6 Restart the computer or start Authentication Manager by entering this command line in the Terminal application:

[computer1:~] root# tim

7 Repeat steps 1 through 6 for each additional Mac OS X computer whose local domain is part of the hierarchy.

#### **Reset Existing User Passwords**

When you add a new user to a domain that resides on Mac OS X Server, the user is automatically set up for encrypted password validation. However, passwords of existing users in each domain residing on the server must be reset. Use the Users & Groups module of Server Admin:

- First reset the root user's password. The root user is the user named System Administrator, which is listed when you select Show System Users & Groups in the Users & Groups List window.
- Then reset the password of existing users that will be using Windows computers.

If you receive a 5015 error when adding users or changing a user's password, you most likely have not enabled Authentication Manager properly for all of a hierarchy's domains.

#### **Disabling Authentication Manager**

Follow these steps if you no longer want to use Authentication Manager:

- 1 In the /etc/hostconfig file, ensure that this line exists: AUTHSERVER=-NO-
- 2 Remove all files that have a ".tim" extension from /var/db/netinfo.
- 3 Using NetInfo Manager, remove the /config/authentication\_server records from each domain, and remove the tim password property from all users.
- 4 Restart the server.
- **5** Repeat steps 1 through 4 for all other servers that host NetInfo domains.

#### **Populating Domains**

Use Server Admin on Mac OS X Server to populate domains that reside on the server. Server Admin provides a consolidated, easy-to-use interface for managing NetInfo records for users, groups, printers, and mounts.

This section provides a brief summary of the procedures involved in populating domains. For complete details, see *Mac OS X Server Administrator's Guide* and online help for Server Admin.

#### **Setting Up Mounts and Automounting**

Use the Sharing module of Server Admin to identify share points in shared domains that you want to mount automatically in a user's /Network directory:

- 1 Open Server Admin on a server that has the shared domain in the login hierarchy. Click the General tab, then click Sharing.
- 2 To define a share point, choose Set Sharing Attributes, select the item you want to share, and click Choose. Then configure the share point for AFP, Windows, and FTP access and/or for NFS access:
  - a To set up the share point for AFP, Windows, and FTP access, click "Share this item and its contents" in the General pane, then set up privileges.
  - **b** To set up the share point for NFS access, choose NFS Access Control from the pop-up menu in the General pane. Use the NFS Access pane to set up export criteria.
- 3 To set up automounting, choose Automount from the pop-up menu.
- 4 Choose the domain in which you want to make the automount available; you will be prompted for the user name and password of a user authorized to change the domain. After you are authenticated, click "Automount this item to clients in domain."

- 5 Select an automount option. If you choose "Mount dynamically in /Network/Servers," share points are listed in the /Network/Servers folder and mount when the user selects them. If you choose "Mount statically in," share points mount automatically at client startup in the location you specify, usually/Network/Servers, /Network/Applications, /Network/Library, or /Network/Users.
- 6 If you've set up the share point for access using AFP and NFS, select the protocol you want to use to mount the share point.
- 7 Click Save.
- 8 If you are setting up a share point so that home directories are visible to network users and you mount the share point using AFP, use the Server Admin Apple file service module to make sure that users will not be automatically disconnected when they do not use the server for a while. In the Idle Users tab, do not select "Disconnect idle users after \_\_\_\_\_ minutes."

When you set up home directories using the Users & Groups module of Server Admin, choose the share point for the home directory location.

#### **Defining Users and Groups**

Use the Users & Groups module of Server Admin to define user and group records in NetInfo domains:

- 1 Open Server Admin on the server where the domain to which you want to add the user or group resides.
- 2 In the General tab, click Users & Groups. Choose Show Users & Groups List and select the domain in which you want to define user or group records.
- 3 Click New User or New Group to add a user or group to the domain. After entering information for the user or group, click Save.

To create the user or group by copying an existing user or group in a different NetInfo domain:

- 1 In the General tab, click Users & Groups and choose Find Users & Groups.
- 2 Enter criteria that match the user or group and click Find.
- 3 Select the user or group and drag it to the Users & Groups List opened in step 2.
- 4 Click Copy to indicate you want to add a user, not an alias.

#### **Sharing Printers**

Use the Print module of Server Admin to create a record for a printer in a shared NetInfo domain:

- 1 Open Server Admin on a server that has the domain in its login hierarchy.
- 2 In the File & Print tab, click Print and choose Show Print Monitor.
- 3 If the printer you want to add to the domain is not listed, click New Queue to add a queue for it.
- 4 In the Print Monitor window, select the printer and click Edit.
- 5 Choose the domain from the Share LPR Queue in Domain pop-up menu.
- 6 Click OK.