Tips and Tricks For Macintosh Management

Version 12.1b

Tiger Release

Summer 2005

As they say in the publishing business:

“This page left intentionally blank”

except for the obligatory picture of Tenshi:



“If I sit at the table, it means I get fed, right?”

Contents

Overview.....8

1. Workflow and Managing Tiger(s).....9

a. What is workflow management?.....9

b. How does this differ from ‘workgroup management’?.....10

c. Groups versus workgroups.....11

d. Deployment scenarios.....12

2. Tiger server setup.....13

a. Service planning.....13

i. How many services per server?.....13

ii. AFP (Apple File Protocol or Appleshare).....13

iii. DHCP.....13

iv. DNS.....14

v. iChat.....14

vi. NetBoot.....14

vii. Open Directory.....14

viii. Software Update.....14

b. Installation tips.....14

i. Clean Install - using Server Assistant.....14

ii. Upgrade (‘dirty’) install (“I need MacMgr...”).....15

c. Key sticking points.....15

i. Setting up DNS on Tiger.....15

ii. DHCP tips.....17

iii. AFP tips.....18

iv. NetBoot (see the client setup section).....18

v. Software Update server.....18

vi. Open Directory tips..... 20

- How many servers do I need?..... 20

- Where do my servers need to go?.....21

- What about authenticated binding?..... 21

- What's a replica gain me?..... 21

3. Tiger client setup.....22

a. Basic configuration ideas.....22

b. Imaging thoughts.....22

c. NetBoot - think about it.....23

i. New Imaging utility..... 24

ii. Custom Installer image sets.....24

iii. Model Filter.....24

iv. Computer naming.....25

v. Directory Binding in NB images..... 26

vi. A few tips for better NetBoot experience.....26

d. Classic and other historical needs..... 26

4. Client Management basics 28

a. Managed Client for X (MCX) essentials.....28

b. Tiger management features.....29

i. Managed Network views.....29

ii. Basic Preference management..... 30

iii. Details and the Preference Editor.....34

c. Mobile Accounts and Portable Home Directories basics.....34

i. Defining mobile accounts.....35

ii. Portable home directories..... 35

iii. Synchronization options..... 36

5. Directory Services tips 38

a. Account planning.....38

b. Home directory considerations.....39

c. Mobile Accounts versus everything else..... 40

d. Groups and workgroups.....43

e. Computer accounts.....46

6. Managed Client for X (mcx) in detail..... 48

a. Preferences versus Security.....48

b. The logic of Managed Client..... 48

c. Understanding the 'mcx' stuff.....51

d. Tips for setting up managed Preferences..... 56

i. Computer account setup..... 56

ii. The Preference settings.....58

a. Applications.....59

b. Classic.....59

c. Dock..... 60

d. Energy Saver..... 60

e. Finder.....60

f. Internet.....62

g. Login..... 63

h. Media Access..... 64

i. Mobility.....65

j. Network..... 65

k. Printing.....65

l. Software Update.....65

m. System Preferences..... 65

n. Universal Access..... 65

e. Working with the Preference Editor.....65

i. Example: Setting up a Quicktime preference..... 66

ii. Other interesting preference tricks.....67

f. Preference Manifests..... 70

g. Portable Home Directories and synchronization..... 71

i. Why and when to turn on ~/Library sync.....71

ii. Digging up the parts to turn on ~/Library sync..... 72

a. Step 1 - Login as 'root' and create a copy..... 72

b. Step 2 - Open the app copy bundle.....72

c. Step 3 - Edit the Compositor.plist.....72

d. Step 4 - Edit the CinchDefaults.plist.....73

e. Step 5 - Replace old mcxd.app and deploy.....76

h. Dealing with the mobile account 'rabbit effect'.....76

7. Workflow management - getting in deep (coming in v12.2)....78

a. Enhanced group folders.....78

b. ACL's - the new permissions.....78

c. SACL's for server admins..... 78

8. Apple Remote Desktop tips.....79

a. Mass system changes - 'Send UNIX command'.....79

b. Reports - a plethora of information.....80

i. Install ARD on remote system and set as remote collector.....80

ii. Set admin system to use remote collection system.....81

iii. Set clients to report on a schedule.....81

c. Troubleshooting using ARD.....82

9. Wrap up 83

Appendix A. Reference files 84

Documentation from Apple on Tiger server: Located at http://www.apple.com/server/documentation/ 84

A1. systemsetup commands for ARD 85

A2. networksetup commands for ARD 89

A3. Modified CinchDefaults.plist in XML format 95

Overview

This is the first Tiger-based version of the T&T anthology. With the release of MacOS X v10.4 (Tiger), our ability to deploy and manage large numbers of systems has increased dramatically. Tiger has not only provided many improvements in performance and reliability, but has also introduced many new services for collaboration, workgroup management, and imaging. We are approaching the 3 year point since Panther (10.3) was released, and at this point, sticking with Jaguar (10.2) or earlier seems to be an exercise in futility. The improvements in the client aside, the performance of Tiger server is an order of magnitude better than what we saw when Panther was released.

This paper is based on the current Tiger release (10.4.2), and will contain references to working with previous releases where it makes sense. In all cases, my examples and scenarios are dependent on using Tiger server. With the improvements in server components, I see no need to work with earlier server software, regardless of the client.

In order to pull all of the concepts discussed in this paper together, I have included a 'walkthrough' setup. It is only an example, and can't possibly cover all possibilities - but many people asked for something like this, so it's there. (Actually - it's not in the initial release, it'll be in v12.2) The admin guides are very useful for deriving detailed information about specific services and some technical underpinnings that I don't have the room for in this document. The admin guides can be found at the main OS X server web page - http://www.apple.com/server and should be referenced when possible.

As with all of the previous T&T docs, I am extremely thankful for the enormous amount of help I have gotten from both inside and outside of Apple. The Desktop Management Solutions (DMS) team is a great group to work with, and their insight and tolerance of my constant questioning of everything is appreciated. The many contributors on the various lists, and those who pop up in iChat at midnight with ideas have made my job much easier.

As with the previous Tips and Tricks, dating all the way back to 1996, I am wide open to constructive comments, spell checking, and just ideas for better papers going forward. I am also constrained by my job in the Education division from being able to provide more than cursory assistance to those who need technical support as they go forth deploying Macs in their environment. Please use Applecare and the professional services group for more detailed assistance.

Summer, 2005

John DeTroye, Sr Consulting Engineer

johnd@apple.com

1. Workflow and Managing Tiger(s)

By default, every Mac that ships is designed to be a family computer, meant to go on the kitchen table. In Education, that means gathering dozens or even thousands of new systems from the warehouse and converting those systems into educational technology tools. If you have the capability and funding, you might get ahead of the curve by arranging to have new systems pre-imaged for your use in manufacturing. This process is called 'CDM' or customer driven manufacturing.

Even if your systems show up with the hard drives configured with a clean OS, your defined list of applications, and even your school's Desktop picture, you may still need to establish a plan to deploy and manage these systems. This section is dedicated to that planning process, specifically, to a concept for promoting collaborative learning called workflow management. The other key aspect of deploying large numbers of systems is establishing management practices. Most sites use an AUP (Acceptable Use Policy) to establish rules of behavior for computers. However, this does not mean that you should just trust the end users to 'do the right thing' all the time. Setting up management for school-owned computers makes sense, if only as an aid to the end users. Stepping in front of either accidental or malicious damage to systems is makes sense.

a. What is workflow management?

Workflow management is the process by which students and teachers collaborate to accomplish the tasks assigned. It isn't a radical new way of doing day to day work, it is just a better way for defining the process that many educators already, albeit unconsciously, perform. The idea stems from the way most work is performed in the 'real world' as defined by the general public - i.e., the business world. The idea is that workers (students) are given a project (homework, thesis, paper, etc.) to complete and the manager (teacher) provides guidance, direction, and feedback on the project until it is accomplished. At that time, the project is evaluated (graded). In business, much of the project work is done by teams; but there are many projects performed by individuals as part of a larger project.

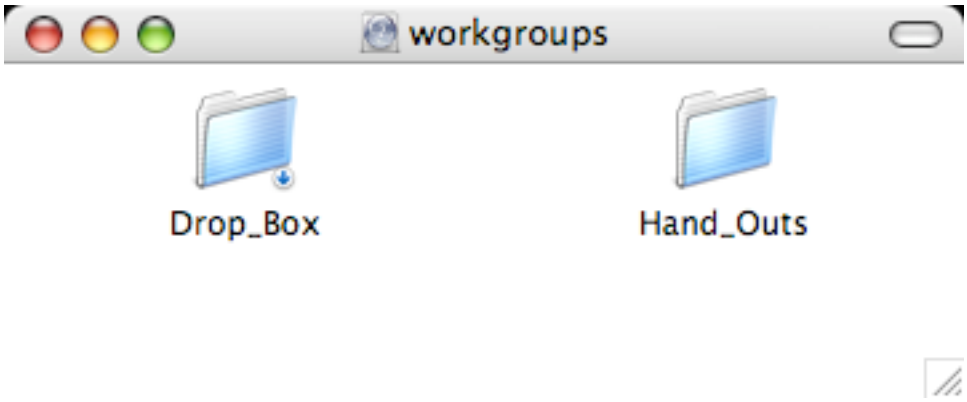
Let's examine this workflow in the light of a History class whose current project is to study the fall of the Roman Empire (I love ancient history...). The teacher assigns each student a portion of a timeline to research and provide feedback on. The portions of history are overlapped so that several students are given the same time period. In the classroom, the teacher would hand out a general timeline of major events to the entire class, and ask that each student dig into their own designated period to provide some specific event or events to fill out the significance of that period. While the teacher hands out documents that encourage exploration, the students turn in documents that enhance the experience, such as reports and presentations. Doing this manually is what we have seen going on for decades in school. Doing it electronically is merely bringing the process into the present.

Where once upon a time, a teacher would have an in-basket on their desk for students to turn in papers, and also a basket with hand-out material for them, now a teacher can use the electronic equivalent of the same baskets. This can be done for any class, and at many various degrees of complexity.

b. How does this differ from 'workgroup management'?

Actually, workflow management is accomplished through workgroup management. Using designated managed groups (workgroups) and group folders, a teacher can actually manage the flow of information to and from their students. For example, a school may use simple workgroups such as 'student' and 'teacher', each having a group folder for those respective users to share common items. At a more complex level, the same school may create detailed sets of workgroups for students and teachers with many sub-groups that are used to refine the various classes or courses into unique entities.

Here's an example of a very basic group shared folder with only the minimum capabilities:

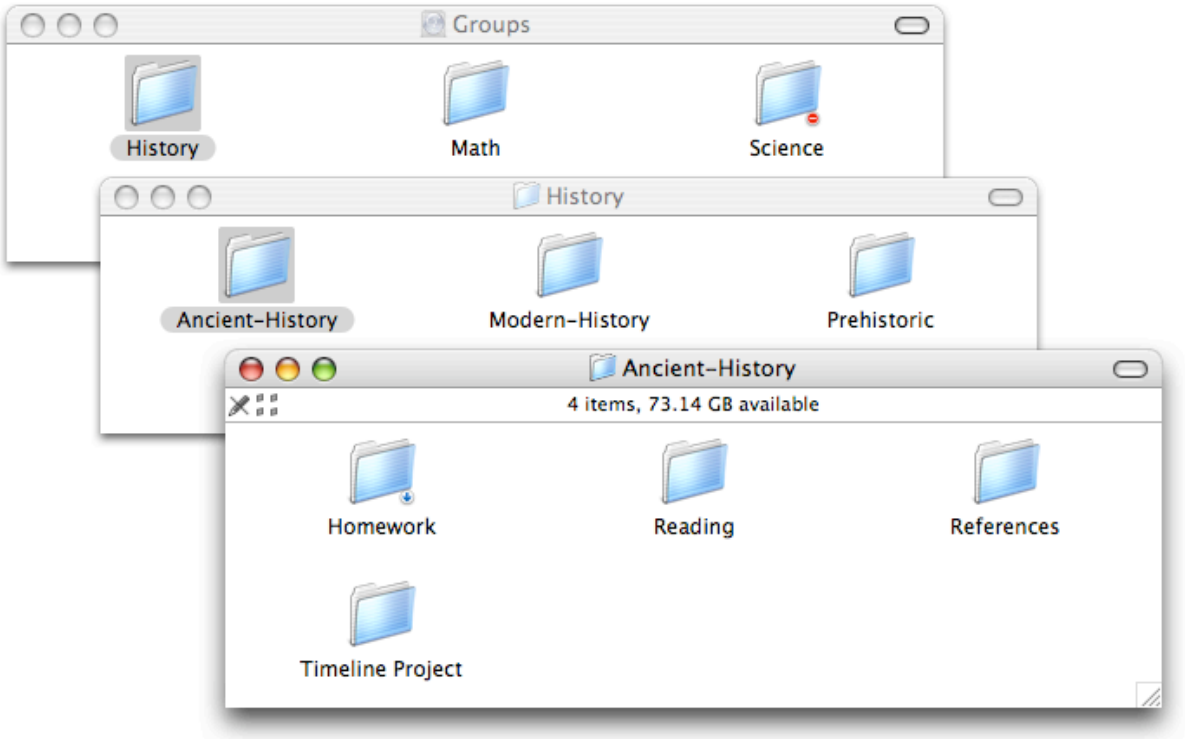


A basic group folder with a dropbox and a read-only sub-folder

You will note that the folder does not look like the default group folder built when the system uses the 'CreateGroupFolder' command. The reason for this is that that command generates a set of system calls to create a folder at the group folder location with 'Documents', 'Library', and 'Public' folders. I, personally, have no use for that template, so I delete those items and create my own. It is possible to create as simple or as complex a group folder environment as you see the need for. My preference is to mimic the method I would use in real life - create a series of folders and sub-folders the same way I would do it if I was using manila folders to accomplish the same task.

Remember - the idea is to create a working environment electronically that your users would immediately recognize as useful. Sometimes just following the templates doesn't cut it.

Now, here's a more complex group sharepoint with many sub-folders, each belonging to a specific portion of the curriculum:



Layered group folders for better workflow management

Each of the group folder sets can be accessed at any time by any of the members of each specific group of users. This is because the group folders we use are not dedicated to any specific login outside of the user belonging to the group that is allowed access to that group folder. Think about having a common area where each of the teachers and students can locate the class material they need, all at once, without having to log into different class groups. This would mean that Joey could turn in his math paper while he was working on an iMovie project for History - assuming his History teacher wasn't standing over his shoulder making sure he was deep into proper concentration for the project....

c. Groups versus workgroups

Groups in Open Directory are collections of users. Workgroups are groups with managed preferences. You assign users to groups for administrative, permissions, and tracking reasons. Users are assigned to workgroups to establish differing sets of preferences for each group.

An example of each of these would be assigning the same user to a group called 'Students' and to a workgroup called 'Mobile users'. In the first case, the user belongs to an administrative group that can be given permission to access certain folders, sharepoints, and computers. In the second, or workgroup, case, the user belongs to a set of individuals who are given certain preference settings - such as access to specific applications, Dock settings, and system preferences. Group folders are part of a group, and a user can have access to a group folder regardless of their workgroup assignments. More on group folders as we go along.

The key to remember is that groups help define permissions, workgroups define preferences.

d. Deployment scenarios

There are many ways to deploy systems and many ways to describe those deployments. For this paper, I have decided to pick four distinct scenarios that outline common deployments. They are:

- Wired/Shared

- Wireless/Shared

- Wired/Unique

- Wireless/Unique

Each of these deployments may be mixed with any of the others. Let's look at what each of these offers in terms of capability and flexibility.

Wired/Shared is most commonly known as the 'computer lab' deployment. It is a set of systems on a fixed, wired network. All of the units are shared resources, with several users having access to any one of the systems at any time during the day.

Wireless/Shared is the iBook cart scenario. Shared portable systems that are used by several people during the course of a day.

Wired/Unique is normally desktop systems that are under the daily use of a single person. Teacher desktops, staff computers, and any other 'cube rat' systems fit this category.

Wireless/Unique is normally referred to as the 1:1 deployment. This is providing a user with their own portable computer. There are a significant number of variations on this scenario, and we will go into a lot of those variations in this paper.

Each of these scenarios provides a baseline to the actual deployment you will probably use. For example, a lot of schools will deploy iBook carts along with a fixed lab for more I/O intensive work, such as iDVD or ProApps projects. Later in this paper we will go over these scenarios as they relate to user account management and home directory usage. I am presenting these scenarios now as a way to define the basic operation of a specific deployment.

2. Tiger server setup

Setting up the server(s) is/are usually done parallel to the client imaging or master image creation. This section is not a complete teaching guide for setting up Tiger server; but a set of tips and guidelines of things to cover during the process. Once again, I strongly recommend familiarity with the various server administration guides posted online at the Apple site. http://www.apple.com/server/documentation/

a. Service planning

First of all, Tiger server is extremely DNS dependent. All of the services are very happy when they can locate the server by FQDN (fully qualified domain name - such as 'xserve1.apple.edu'), and very unhappy when they can't. This means that if you don't have a DNS server with the server's DNS name and IP address available to all clients, odds are your setup will fail. Period.

i. How many services per server?

The most often asked question is 'how many things can I run on my server at once?' - followed by 'how many clients can I connect to this service/server?' Well, there is no pat answer - but I will tell you that more servers are better than less. There are some guidelines that I can give, based on feedback and experience. Unfortunately, many of the services are not used by me for this paper, and I don't have numbers for everything. Let's walk through the Server Admin tool and look at some the services and a few possible numbers.

ii. AFP (Apple File Protocol or Appleshare)

If the server is used for general file sharing, where the clients connect randomly and seldom do more than download an item or two, will support up to 1000 concurrent connections. That is 1000 clients mounting the server for random usage. If the server is going to be used for group folders or for an online database, such as a reading assessment application, then 3-400 concurrent connections would probably work fine. For dedicated network home directories, where each user is bound to that server for all of their documents and preferences, you should have no more than 150 concurrent connections. As for mobile accounts with portable home directories - the new thing in Tiger - I don't have numbers for that yet. I would assume that the number of concurrent connections ranges between the 150-450 range, and that is going to be based on how much synchronization is taking place.

iii. DHCP

The DHCP server actually takes very small hits, and a single DHCP service should be able to handle a Class C subnet (254 clients) very easily. I am also guessing that the service could probably scale to handle several subnets, or even a portion of a Class B subnet easily. Many sites set up DHCP at their router, so make sure you check to insure that you aren't duplicating efforts. Also, NetBoot servers need DHCP; but do not need to be running that service on the NetBoot server itself.

iv. DNS

My recommendation is to set up DNS at every site/building/location so that there is a DNS server close to the clients. If this server is acting as a secondary server to a district master, that works too. The idea is to reduce the latency of a DNS request to as short a time as possible.

v. iChat

The new Jabber server hasn't gotten a lot of stress testing yet in the field. I would recommend that if you are going to be using this heavily, then plan on setting up a dedicated collaborative server with iChat, WebLog, and QTSS on the same box by themselves. There is a lot of caching and encryption going on when these are running.

vi. NetBoot

If you are going to use NetBoot heavily with dedicated images to actually boot clients, then dedicate a server running this to each of 75-150 client systems. If the service is going to be used for one-off NetInstalls, then you should be able to run this service on any fast server without interfering with other processes. During that once/twice a year re-imaging marathon, you might even shut off all other non-essential services and use all of your servers as NetBoot/NetInstall systems.

vii. Open Directory

There is a hard-coded limit of 1000 simultaneous logins on the OD server process. Therefore, you should make good use of the master/replica configuration capability of Tiger server for more than 1000 clients. I'd stay on the safe side of 900.

viii. Software Update

The new software update server mechanism can get pretty swamped when first setting itself up, and when you send a 'softwareupdate' command to all of the clients using ARD. If you are going to set this process up to avoid your clients pinging the 'mothership' for all updates, I recommend running this on a non-homedir server at each physical site. If you are going to have several thousand systems in action at a site, then look at running this process on 2-3 servers to keep the load down. You can set sections of clients to use different swupdate servers.

b. Installation tips

Installing Tiger server seems to take one of two tracks: upgrading a Panther server to retain older settings and/or Macintosh Manager services, or clean installs. We will discuss both here; but will focus on clean installs first. As always, a clean install is the best way to clear out the old 'stuff' and allow for a new start.

i. Clean Install - using Server Assistant

Direct CD/DVD installation is pretty straight forward; but the real trick is using the Server Assistant to do remote installs. Since the servers you ought to be buying (hint) are Xserves, hence headless systems by default, then using the Server Assistant to install and configure these units is a useful capability. You can also use this utility to setup and configure the server after installation. Note: You will need to boot the server



Server Assistant

from the installer CD/DVD to make this work - or you should be able to create a NetInstall set from the Server DVD to accomplish the same task. (I have not tried to create a NetInstall set from the Server CD set. You should be able to create a custom install kit using the different CDs though.)



ii. Upgrade ('dirty') install ("I need MacMgr...")

I discourage upgrade installs, if at all possible. You should be able to back up key information, export user/group/computer account information, and perform a clean install and restore data to insure the best performance. However, if you are heavily invested with custom configuration information and/or need to keep the Macintosh Manager service, then you need to upgrade from 10.3.9 to 10.4.2 (as of this writing).

It is a good idea to make sure your server has been upgraded to 10.3.9 before jumping to Tiger. If you are running Jaguar (10.2.x) then a clean install is pretty much mandatory - your server info is probably way out of date.

If you are doing an upgrade install, then boot from the latest server install CD/DVD you can find. If you have only the 10.4 kit, contact Applecare and request a newer build of the CD/DVD set - there was a bad bug in the 10.4 server install that caused problems with upgrade installs for several services. Once you have done the install, and should have ARD running on the server, immediately connect to the server with the ARD admin app and send a 'softwareupdate -i -a' command as 'root'. This will push all of the latest updates onto the server and after you reboot, your server will be at peak configuration. Go back in and examine all of your settings to make sure your specific site settings are intact.

c. Key sticking points

There are a few things I would pay very close attention to when setting up a Tiger server and activating some of the services. Below are some tips I have run across that will help. Note: These tips are focused on setting up a managed client world - they are not for the cluster/QTSS/web/mail crowd - sorry

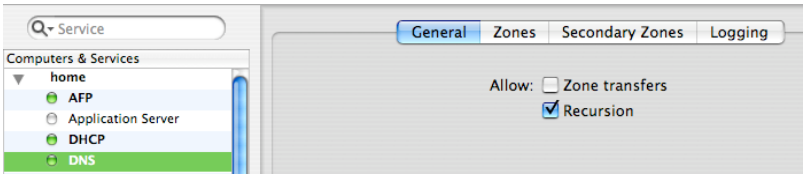
i. Setting up DNS on Tiger

Tiger DNS is massively improved from what we have used before. It's still very much Bind; but the user interface is much better. This new interface was designed for basic usage though - some of you may still want to dig into the Bind config files. If so, have fun - you're on your own. (I suggest "DNS&BIND" from O'Reilly - http://www.oreilly.com/catalog/dns4/)

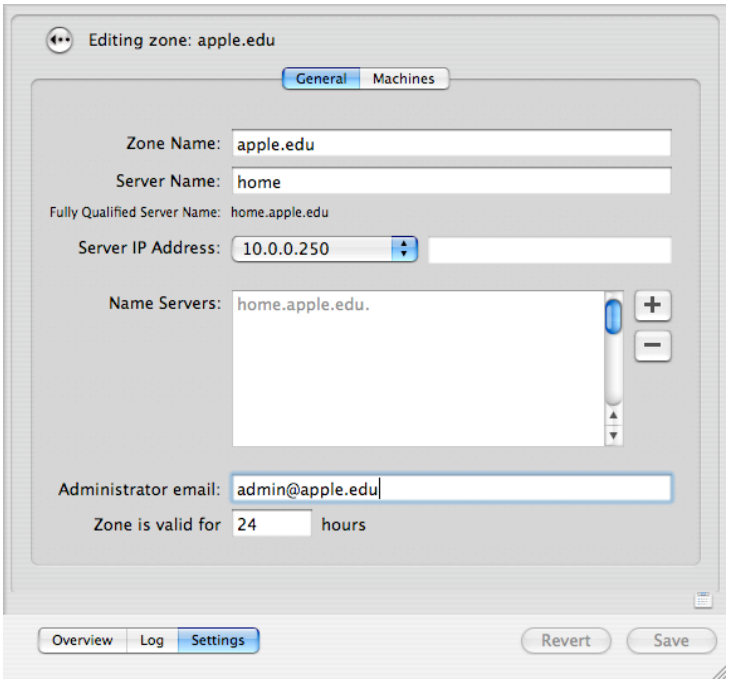
The basic interface for DNS allows for setting up a zone that defines the domain names for the servers and other systems on your network that need to have fixed names and

IP addresses. By and large, your clients won't need this; but it is mandatory for your servers to be configured with DNS.

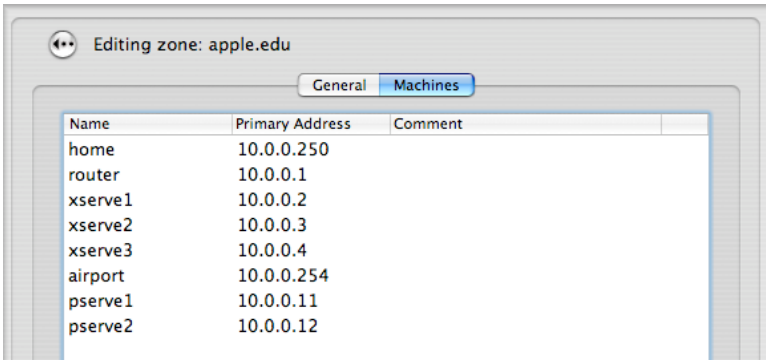
The first window you set up contains two entries: 'Zone transfers' and 'Recursion'. Zone transfers allow you to set up secondary servers to help spread the load and share all of the DNS zone information. Recursion allows the DNS server to walk back up through the Internet to look up information outside of your network. It also makes your DNS server responsible for actually finding the information, instead of passing the query off to another server. I recommend using recursion.



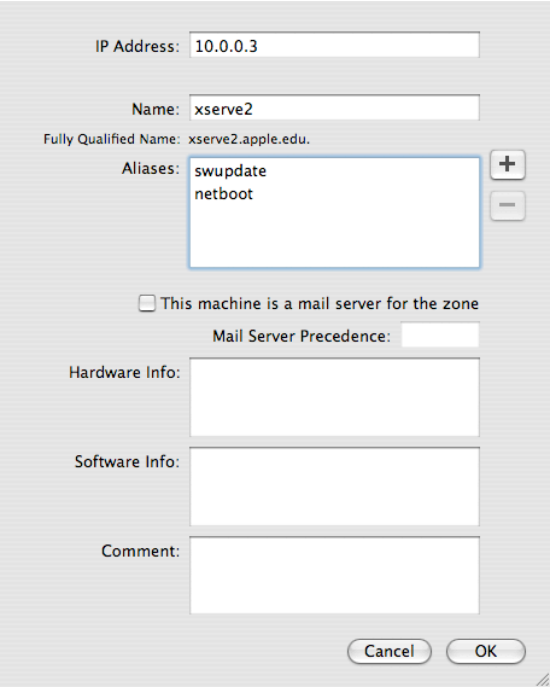
In the second window, you will define the zone for your network. I will admit that I have not gone past setting up one zone per network - despite the large amount of screen space begging for someone to set up lots of zones, just to see what happens. Click on the "+" sign to create a new zone, and fill in the blanks as needed. If you are only setting up one name server for your network, then the settings are pretty simple - as you can see in the picture. The advantage of the new DNS setup in Tiger is that it automatically creates all of the extra information needed - such as the reverse DNS lookup information.



After creating the zone, you can then add your servers and other key network devices to the list. This is done by going to the 'Machines' pane of the zone tab. Here's the window I have for my testing lab:



Adding machine entries is pretty straight forward. Click the "+" button, then fill in the blanks. Since the server automatically creates the reverse DNS entries, you are only required to enter the IP address, the name, and any



aliases you want to use.

Something to watch out for - if you change the IP address of an entry, the reverse entry gets messed up. I suggest that you delete the entire machine entry and start from scratch.

For additional information on setting up DNS, check out the Server Admin guide:

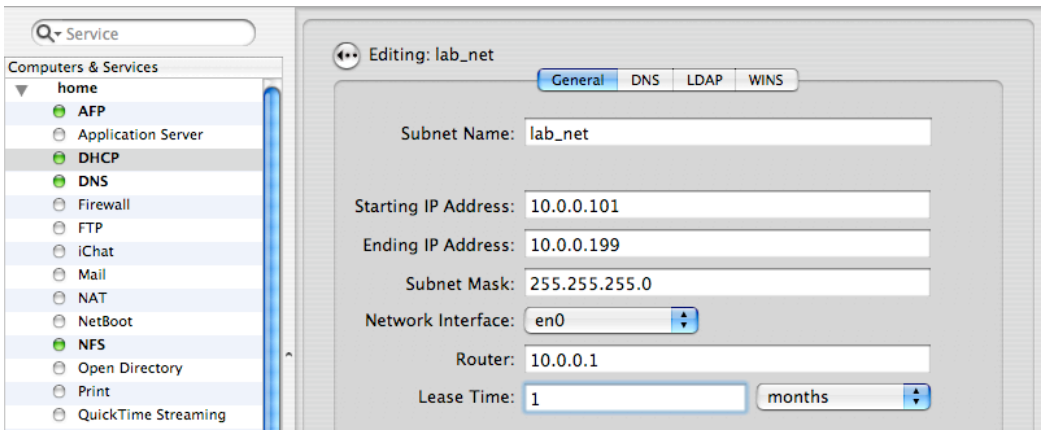
Network Services Administration

ii. DHCP tips

Many network admins provide DHCP through a router, or other device already; but you may have to provide IP addresses for your clients yourself. Setting up DHCP on Tiger server is kept simple through the user interface (GUI). The service can provide a range of IP addresses for systems that are not in need of a fixed IP address, as well as provide a specific IP address for a machine, or machines, as needed. DHCP server also lets you assign LDAP binding information and DNS information.

Before setting up DHCP service on your network make absolutely sure that no other device is already providing the information. That will create lots of problems.

Basically, you create a new subnet and add the requisite information as shown in the window below:



Basic DHCP server - setting up a subnet

The Lease time is one way to make your clients keep their IP addresses for an extended period. Setting the lease to a month, for example, means that the client will keep renewing the same IP address unless it is actually off the network for more than a month.

The remaining settings are self-explanatory, as well as covered very well in the Server Admin guide. No tips for those.

iii. AFP tips

Under Tiger server, the AFP (Appleshare) service no longer uses Appletalk for either browsing or connections. Clients that used to be able to locate an AFP server using Appletalk when running Panther or earlier OS's will no longer be able to find the server except over IP now. This may affect scripts or login items that you may have been using. So, even if you turn on the 'Enable browsing with Appletalk', it won't work.

By default, 'Guest' access is off and you should leave that alone for now - at least until we see what the effect is on automounts.

One of the settings that is not adjustable in the GUI is a value to set how long the server waits to drop a disconnected session. The idea here is that AFP under Tiger (and Panther) supports what is termed as 'aggressive reconnect.' This was implemented to allow for minor network interruptions. When a client drops the connection to the server accidentally (a network burp), the server will allow the session to remain active for a period of time. Meanwhile, the client goes into panic mode (not 'kernel panic' mode - just a low state of paranoia) and aggressively reattaches itself to the server, if possible. The problem is that this disconnect may be an idle timeout/logout - and the client will try to reconnect. Or it may be a logout and sleep by a portable, and the server interprets it as an accident. The default timeout for this is 1440 minutes - a tad bit long. You can adjust this setting by sending the following command to the server using either ssh or ARD:

```
serveradmin settings afp:reconnectTTLInMin = 20
```

The rest of the AFP settings we are concerned with will be discussed as part of the Workgroup Manager section when we cover home directories.

iv. NetBoot (see the client setup section)

I'll give NetBoot it's own coverage in the next section on client setup.

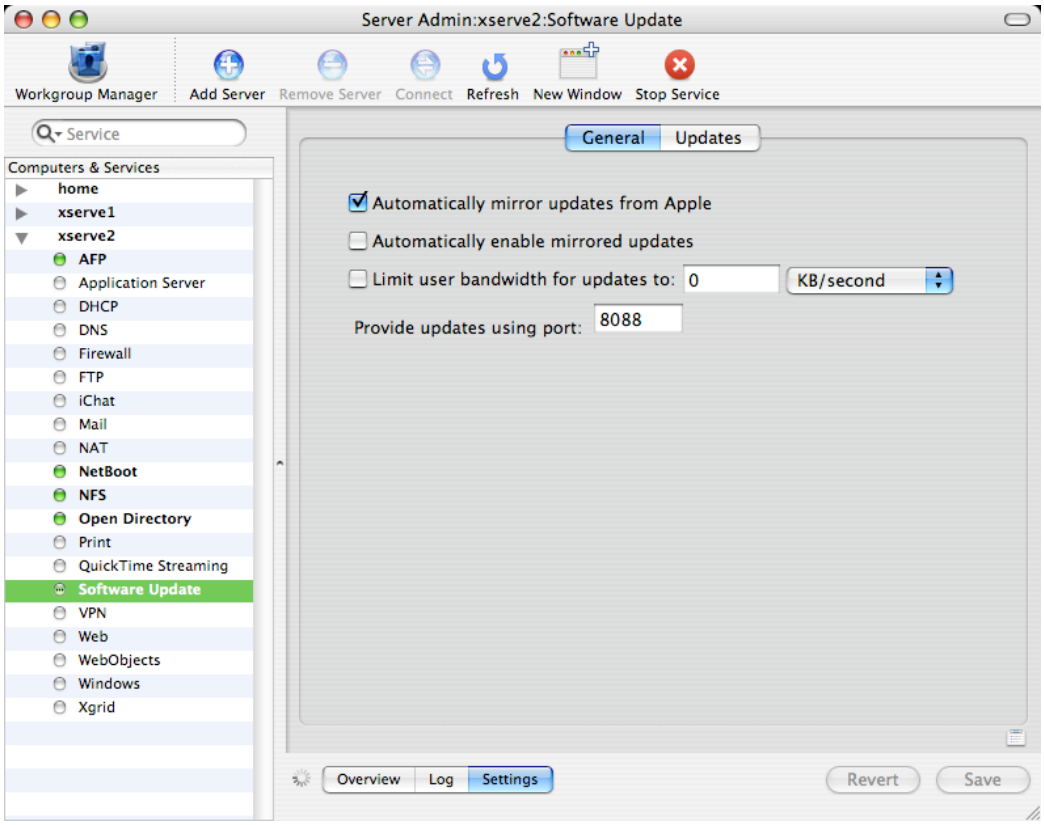
v. Software Update server

New to Tiger server is the local software update server process. This process was designed to act as a mirror for the actual Apple Software Update server maintained at the Mothership. The idea behind the process is to allow all available updates to be relocated onto a server inside your network. Since most school networks are inside a NAT/Firewall, this means that you can configure your clients to look at this internal server for updates instead of having to connect to the outside. Another key benefit of the process is that you now control which updates become available for your clients. You can allow the ones you have tested, and not allow ones that don't need to be posted - such as not seeing the iPod updates show up for all of your lab machines. Yes, you can control this by setting the disable flag for each un-needed update at each client; but why bother?

The server process, in its first incarnation, is very carefully controlled. First of all, the server process will not run if it cannot contact the 'real' software update server(s) upstream. Second, the packages it downloads are encrypted to insure that they are the actual, approved

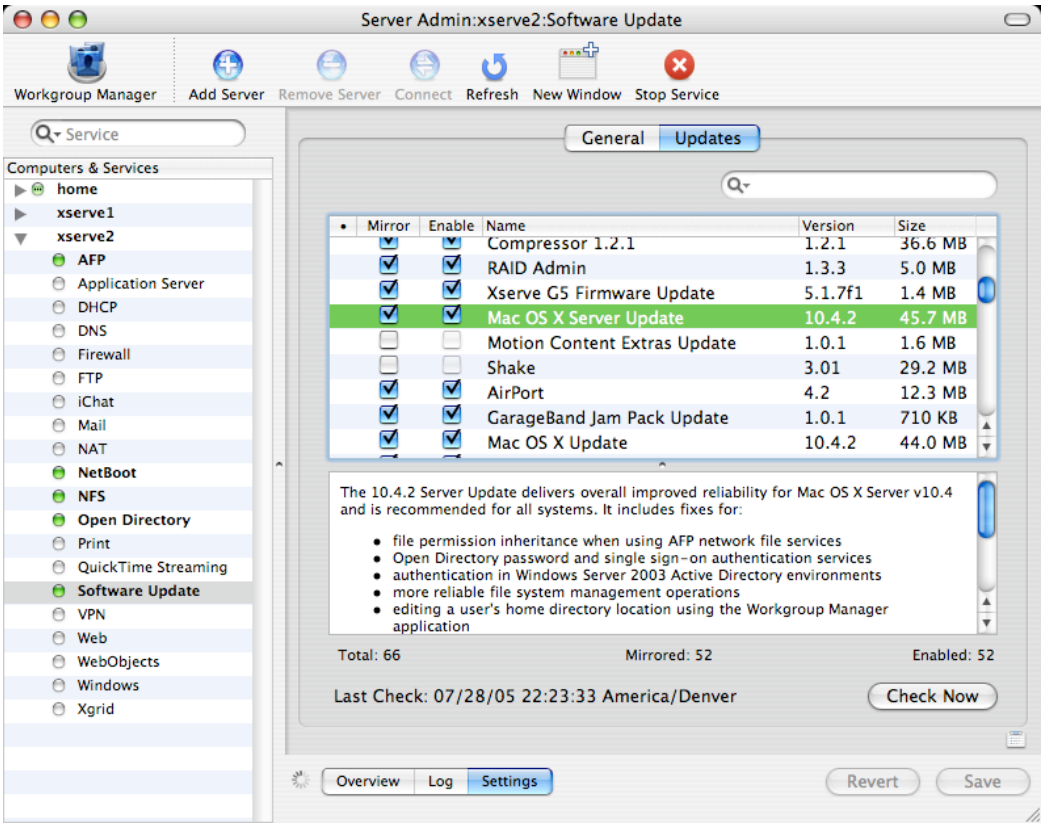
updates from Apple. This means that you cannot move the packages from the server, nor can you add your own to the mix.

All that aside, the process is very basic to setup, and the clients need only a slight push to make them use that server instead of trying to go outside for updates. The server process is activated, and it will take a while to 'phone home' and download all of the available updates. Once that is done, you can pick and choose the ones you want to make available.



Software Update server initial setup

Once the updates have become active, you can turn off the ones you don't want to mirror. That will mean that any future versions of those updates won't be mirrored either. If you turn off 'Automatically enable...' then you will need to visit the update server configuration every once in a while to check to see if new updates have been made available. One way to do this is to keep one system in your testing lab set to look upstream so it will catch the new updates. You can then activate the ones you have tested. Under 10.4.2, you need to check through the update list yourself to see which are new and which are ones you meant to leave off.



Updates made available or not depending on your network/client needs

Setting the clients to connect to the update server can be done in two ways. First, you can manually configure all of the clients to connect by adding the URL to the local software update server to their preferences. Second, you can use the MCX (managed client) settings and set the location of the local server for all managed computers centrally.

To set the location to the 'URLCatalog' manually, you can use either ssh or ARD to send the following command to all affected systems:

```
defaults write com.apple.SoftwareUpdate CatalogURL http://
<myserver.myschool.edu>:8088/index.sucatalog
```

Do **not** send this command to the server running the software update server process!

Setting the command under MCX will be covered in the client management section. Using managed client settings allows you to easily alter the software update server, or change locations one time, and have those effects ripple across all clients without changing individual property lists on thousands of machines.

vi. Open Directory tips

Considering that there is a 250 page manual on Directory Services, I think just pointers to a few key items are best here.

- How many servers do I need?

The number of Open Directory servers depends on the total number of clients. A rule of thumb is that each OD server can handle ~900 concurrent logins. Since you can use

dozens of OD replicas with a single Open Directory Master (ODM), you can support thousands of systems for authentication.

- Where do my servers need to go?

While it is possible to place an ODM at district level and replicas at each site/ building, you may also ask yourself if having one big master list of all users makes sense. If so, then great. If not; i.e., "How many of the K-3 kids visit the high school and vice versa?", then you may want to set up ODM's at each school, or at least shared among a few schools with replicas at each site.

- What about authenticated binding?

Tiger server introduced the ability to bind a client to a specific OD server with authentication - meaning that the server is guaranteed to be the one you attach to, versus setting up an OD master by DNS name and possibly changing it later on. This capability totally stops any way of someone introducing a bogus OD server in the middle of the network to affect your clients. It is also required if you are going to use Login/Logout scripts from within the MCX/managed client settings.

- What's a replica gain me?

If you are deploying lots of clients, you can take several of your servers and turn them from just slaves of the OD master into replicas. This would mean that each of these systems contains a duplicate of the master LDAP database. It also means that users' systems will learn about all of the different authentication servers and will be able to use any of them as an authentication server. The process is detailed very well in the Open Directory Admin Guide.

3. Tiger client setup

There are a lot of ways to create client systems, and much more detailed documents on that than I have time to go into with this paper. Therefore, I am going to defer to the imaging experts and just offer some basic tips to consider when putting the client systems together.

a. Basic configuration ideas

First of all, you need to decide what software and hardware are going into the client systems. Making sure you have enough OS licenses, as well as licenses for all of the other applications is a detail that you really can't ignore. Also, do you have enough ram to run all of the operations you want to on the clients? Even though Tiger can boot on a system with 128Mb of ram, that doesn't mean you can actually run anything once it's booted - at least not at a speed that would make you glad you deployed it.

Second, you need to determine which systems get all the high end software versus the basic stuff. There is no reason not to deploy basic systems to writing labs and Internet research stations, where only a word processor and browser are needed. It's also a good way to keep some older systems alive versus junking them (although I think replacing all old tower systems with new Mac mini's isn't a bad idea).

Finally, remember that the Mac ships as a consumer system. While it may seem easier to add your own software over top of what ships, there may be a budding set of problems with leaving all of the default-installed games and tools in place for curious young minds to find.

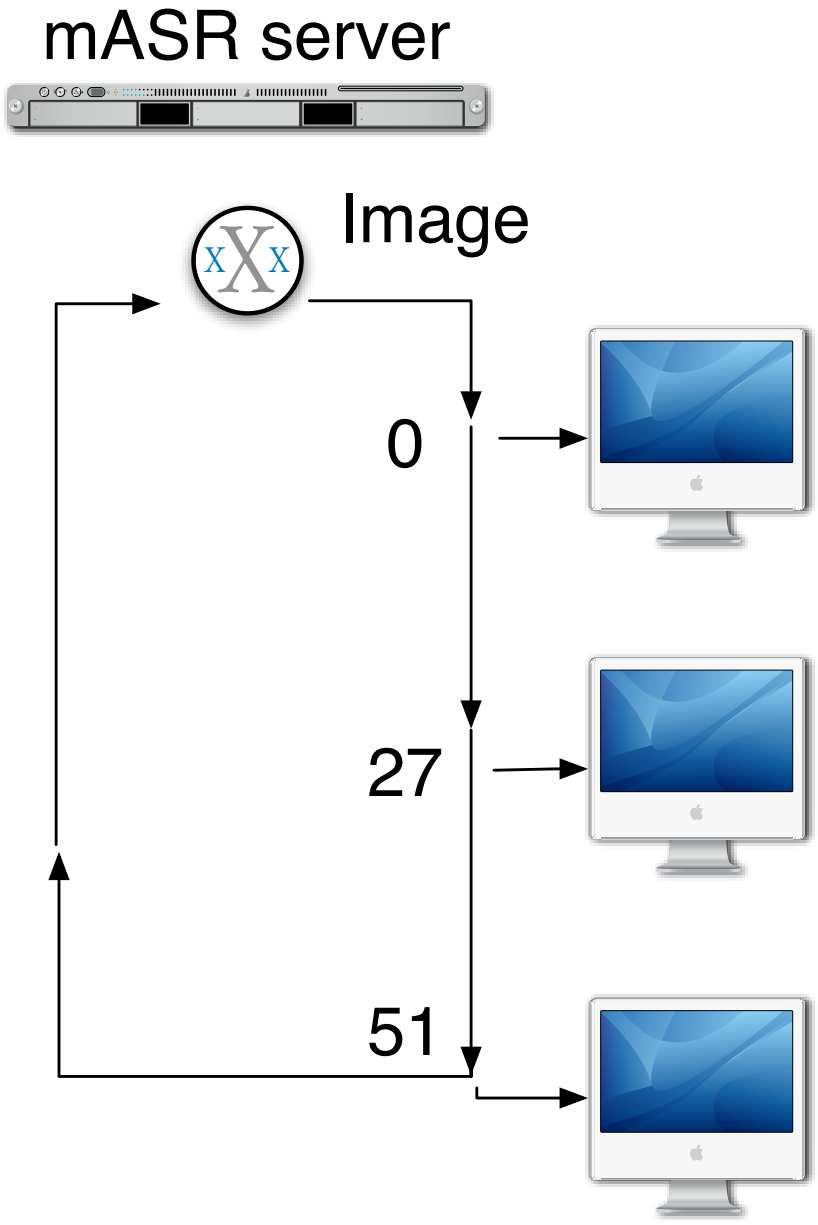
b. Imaging thoughts

The options for imaging systems comes down to two basic concepts - installation of all items or restoration of a system from a known good master. Combinations of these concepts could include installing the core OS and key apps, then copying unique files, or cloning core systems and installing additional applications and settings.

Installation, using CD/DVD or NetInstall, has the benefit of being hardware independent. You can take the latest installer set you have and usually install all of the Mac OS X components onto any system we ship. Sometimes the installer sets are built for a specific class of hardware; but that is less and less the issue with Tiger. The downside is the amount of time it takes to do the installation, as well as the time it takes to customize or 'unique' the system for use.

Cloning, using Apple Software Restore, NetRestore (from Mike Bombich), Radmind, Casper, or any of several other tools, allows you to create a clean master system then clone that onto other systems. The upside is that it tends to be much faster than package installs, and usually includes customization by way of post-action scripts. The downside is that you will usually need to create several image sets to match the different key system classes - iMac/G4 versus iMac/G5, Powerbook versus iBook, and so one. This is because many of the extensions and system components are not installed for one class of system versus another.

Another option is the new Multicast ASR that is part of Tiger. This tool is actually imbedded into every Tiger system, client and server. It works by taking a properly prepared image and sending it out across the network in a continuous loop, bit by bit. A client systems is told to image from the 'stream' and that client just begins collecting the bits as they go by. When the client finally collects all of the bits, it places them into the right order and reboots, re-imaged as needed. For a complete description of the process look at the man pages on 'ASR' (type 'man asr' in Terminal on any Tiger system).



Multicast ASR flowchart

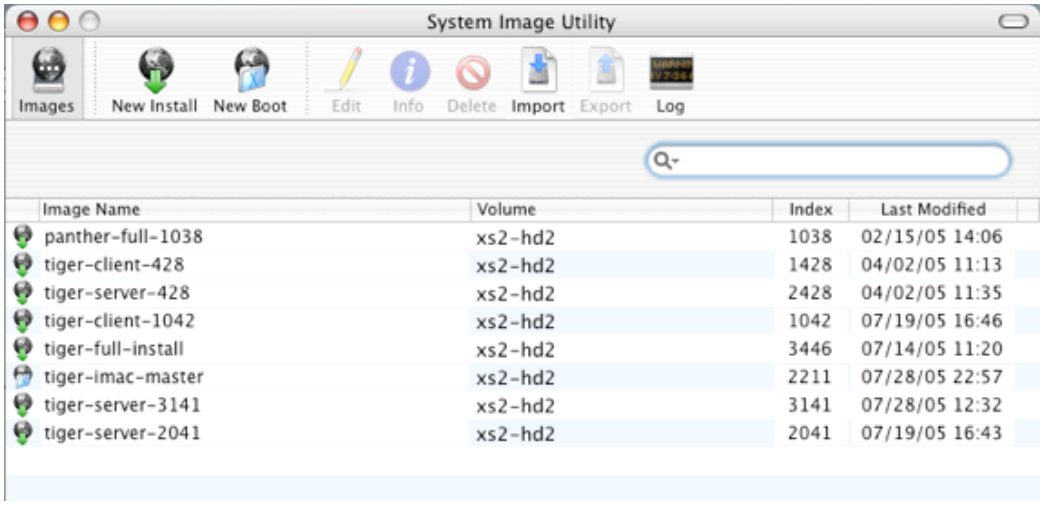
c. NetBoot - think about it

A much-ignored capability in Mac OS X server is NetBoot, and it's companion, NetInstall. The technology behind NetBoot is the ability to boot client systems on a network

from a locked image on a network server. The benefit of this capability is that the client system cannot get damaged, at least from a software perspective. If anything happens to interfere with normal operations on the system, merely rebooting restores the unit to full functionality. Extend that capability to being able to create network-bootable installer sets and you have a powerful set of tools. Let's look at several of the capabilities and settings that are part of the Tiger server.

i. New Imaging utility

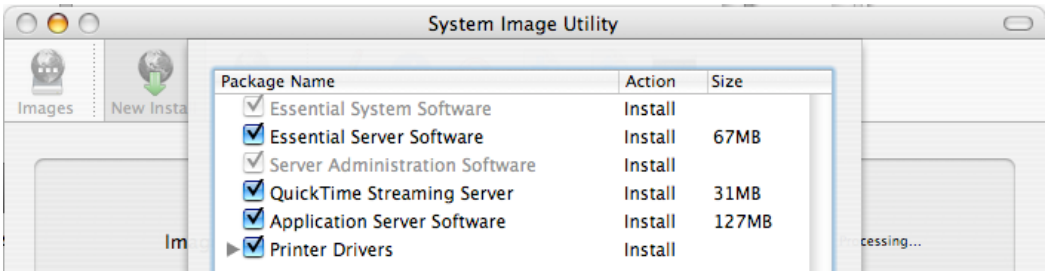
Tiger server comes with 'System Image Utility' which replaces the older 'Network Image Utility' from Panther. The new utility has several new capabilities. The utility can be run either at the server, or from any Tiger client that has the Server Admin tools installed.



SIU - editing images in place at the NetBoot server

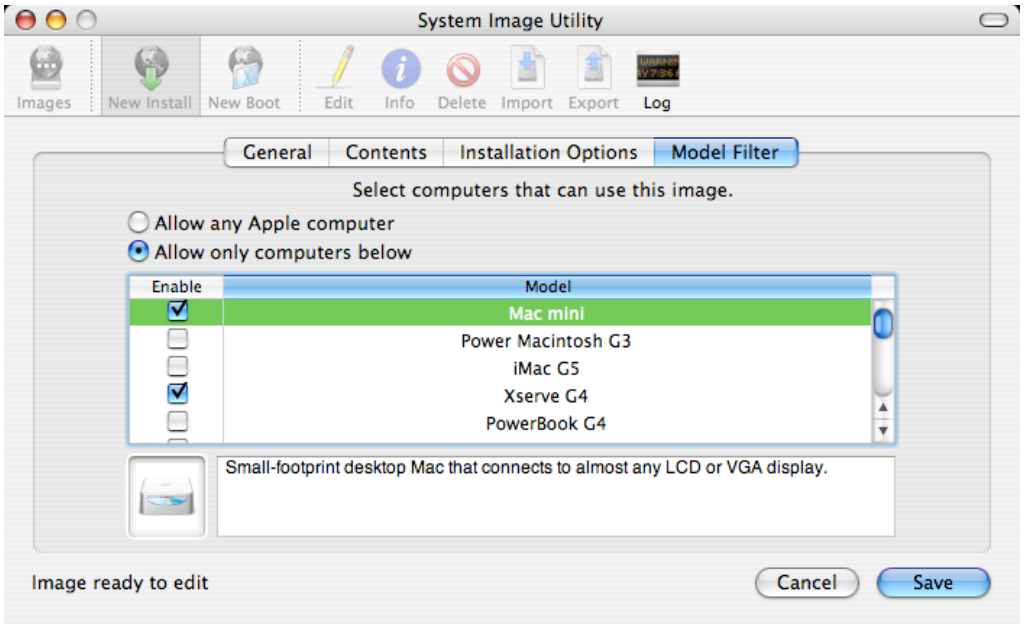
ii. Custom Installer image sets

The new SIU allows you to create custom install sets by editing out the un-needed parts, or adding other parts, to your NetInstall image. You mount the main installer CD/DVD and select it in the source window. then you add/subtract the packages you desire to create the image with just the parts you need.



iii. Model Filter

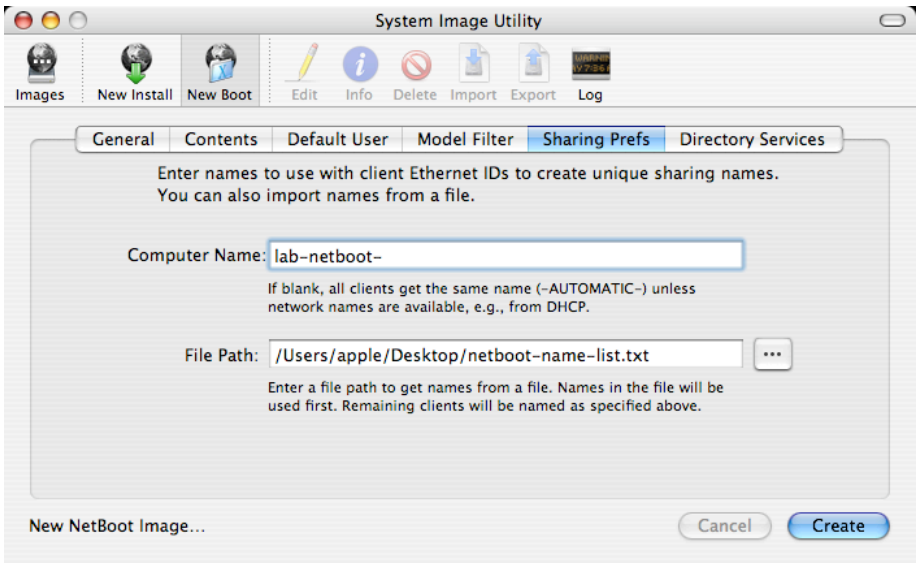
SIU includes a 'model filter' that let's you set what specific types of Macs can boot from the image. This would allow you to keep iBooks from booting off an image created for iMac/G5's, or from booting a server installation image on anything but systems meant to be used as servers.



SIU model filter

iv. Computer naming

SIU now includes the mechanism for setting unique names for clients. It can be done with one of two methods. The first is to use a generic beginning to the name that the system will append with the computer's MAC address.



Naming setup - 'lab-netboot-D0622E'

The format for the name file is simple. It must be text, and must include the following fields: MAC address, '-AUTOMATIC-', computer name, and a blank entry. For example, here is the contents of the file used for my screen shot - note that it does not contain the actual field names, nor can you see the extra tab and blank entry at the end.

```
00:0a:95:b1:6e:38 -AUTOMATIC- lab-ibook-g4-12
00:0a:95:bb:fa:82 -AUTOMATIC- lab-ibook-g4-14
00:03:93:8f:58:f0 -AUTOMATIC- lab-imac-g4-15
00:0a:95:7a:37:82 -AUTOMATIC- lab-imac-g4-17
00:0a:95:dc:8e:3e -AUTOMATIC- lab-imac-g4-20
00:11:24:76:99:a2 -AUTOMATIC- lab-mini-1
00:03:93:d0:10:ac -AUTOMATIC- lab-pbg4-12
00:0a:95:d0:62:2e -AUTOMATIC- lab-pbg4-15-1
00:0a:95:c8:f5:92 -AUTOMATIC- lab-pbg4-15-2
```

v. Directory Binding in NB images

SIU also now supports adding the current directory binding information used by the system you are running the utility from. This would allow you to match all of the your systems up quickly with the master station to be bound to your OD master.



Directory mapping 'borrowed' from the master system

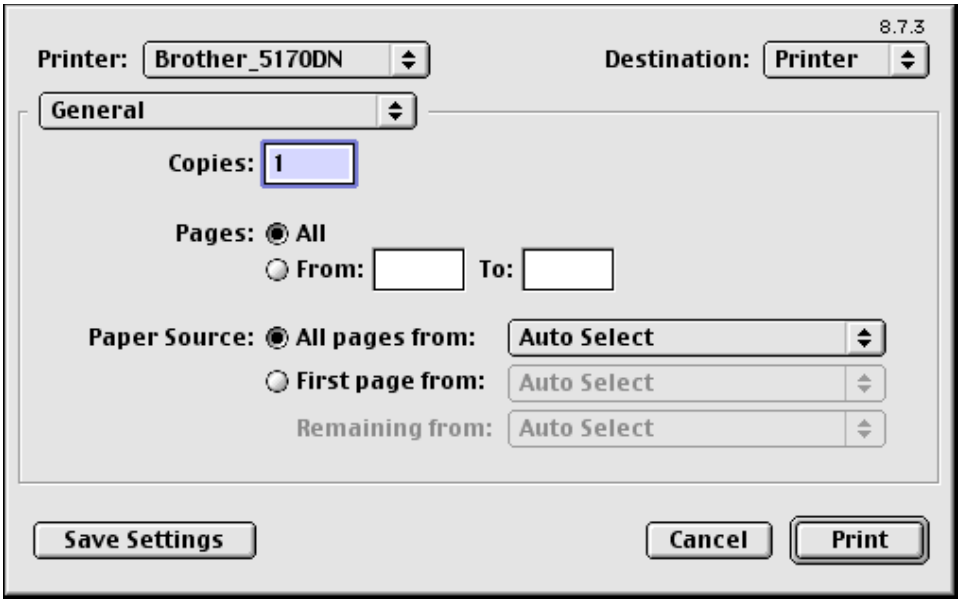
vi. A few tips for better NetBoot experience

Just a few guidelines. First, if at all possible use a dedicated server with lots (lots and lots) of ram as the NetBoot server. Second, store the NetBoot images on a different drive from the core OS. Finally, follow the guidelines in the System Imaging Guide.

d. Classic and other historical needs

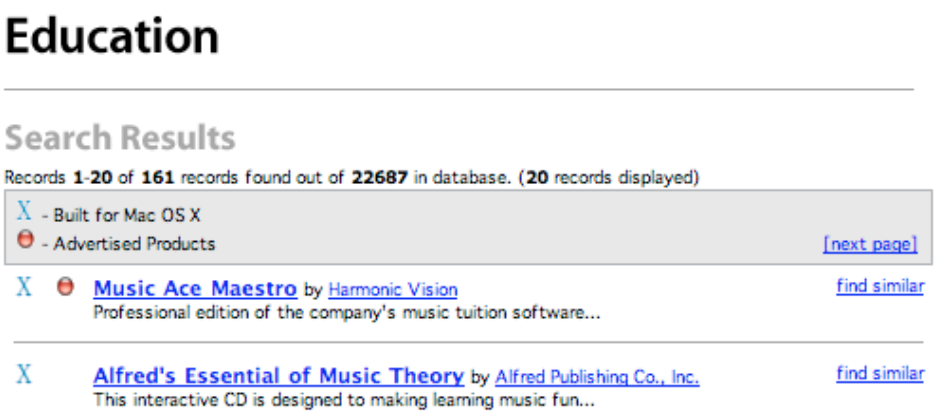
Just a couple of thoughts here. One, many of the old applications that are keeping your systems from being upgraded to Tiger are probably taking more time to support than the rest of your systems together. Second, Mac OS X v10.4.2 is current - Panther (10.3) is already three years old (not to mention how old Jaguar is...). Third, Mac OS 9 hasn't seen an update in almost 6 years, and isn't going to get any newer. Meanwhile, Classic is still

supported in Tiger, and there are a few improvements. For example, the Classic printing engine now uses the default Tiger printer.



This is the default OSX printer for my system - never opened Chooser. :-)

Otherwise, take a really good look at the 'MadeForMacs' section of Apple's web site. You'll find tons of applications that perform the same tasks as many of your older Classic apps - but native to X.



Looking for music education software - for example

4. Client Management basics

The whole idea behind deploying lots of computers that are owned by a school for use by lots of staff, faculty, and students on a daily basis for their work and study is a daunting challenge. Often, much thought goes into designing the 'perfect' network, setting up file services, providing for backup, redundancy, filtering, proxies, and lots of other things that the network and system admins are very good at. However, a couple small things tend to come in at the very end of the process - or not at all - and those are "How are these computers going to be used by the end users?" and "How can I help configure the systems to be both safe from out-of-bounds use and yet enhance the ability of my users to accomplish their work/studies?"

There are two major schools of thought behind client management. The first says "We will set up the computers the way we want, and the teachers/students/staff will like it." The second says "Let's work with the curriculum folks to determine the day to day workflow of a teacher/student and configure the entire process to aid and enhance that process." I can't help the first group, and this paper is designed for the second group. Also, I am not at all in favor of what I call 'management through imaging' which is the idea of building a very complex arrangement around the workstations to allow users to do whatever they want, then just re-image the computers daily (or in some cases, at every login!). The focus here will be on setting up computers to guide the users, assist the users, and control the users enough to keep them from doing 'Dumb Things™' to the computers.

"What about keeping them from <pick your favorite kid activity> during my class?" That comes under the purview of the Acceptable Use Policy (AUP). While we can design lots of technology to keep kids from accessing rude web sites or deleting the applications from a shared computer; we cannot cover all the bases. If a student (or teacher) abuses the privileges of using an expensive piece of equipment provided to them by the school, then take it away from them or tell them to stop. Period. Tenshi says so.

a. Managed Client for X (MCX) essentials

Once upon a time there was Macintosh Manager. Then came Mac OSX and with it we have managed client for Mac OS X - MCX. I use the term 'mcx' because it is the term used for all of the various code segments, as well as being much easier to say than 'managed client for Mac OS X, a component of the Workgroup Manager and Mac OS X server.'

The basic concept behind MCX is to provide a directory-based mechanism for storing preferences for the OS and applications. This is not really a security model as much as way to establish common settings for sets of computers, groups of users, or even for individual users. While many of the features are duplicated by the 'Parental Controls' that are part of each Mac's local account management, the focal point of the managed client world is setting and controlling the settings from a central directory - preferably an Open Directory LDAP server.

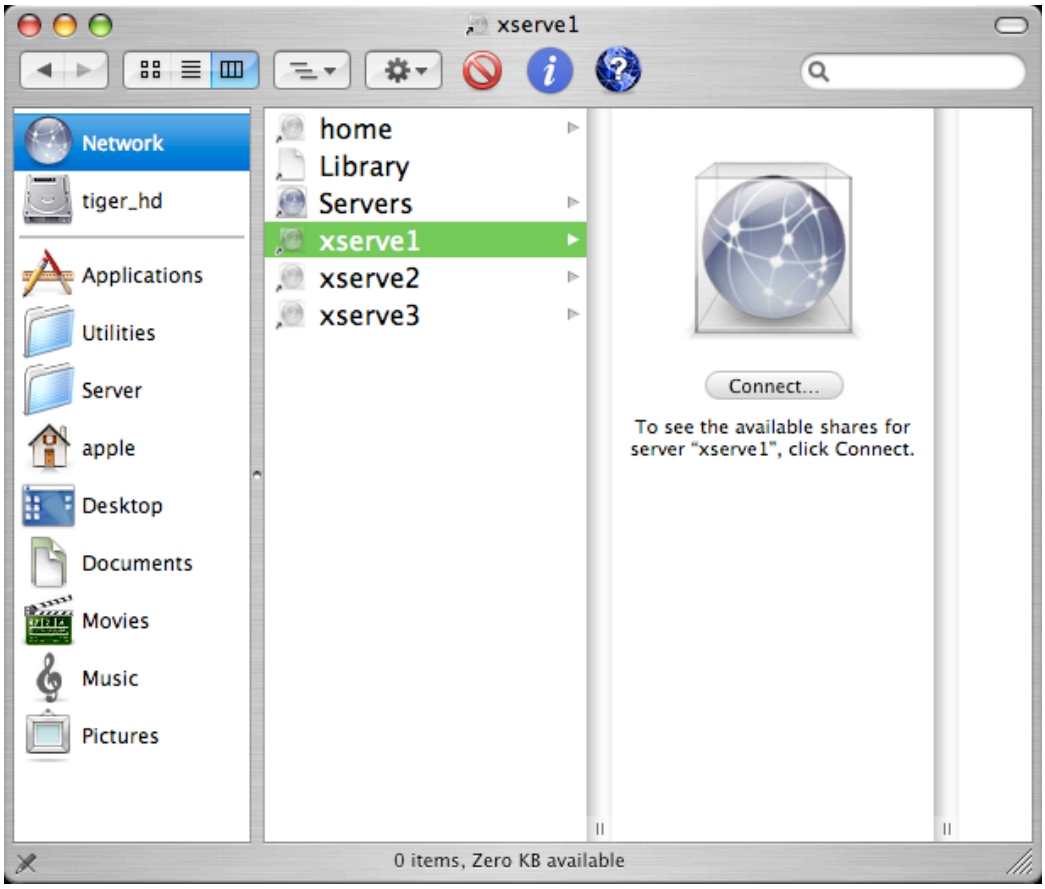
The section 'Preference Management in Detail' goes into much more of the science behind MCX operations. This section covers basic definitions and tips for usage of the essentials.

b. Tiger management features

While many of the features for managed client work with Panther as well as Tiger, we are going to concentrate on the items that will make you want to be using Tiger. :-)

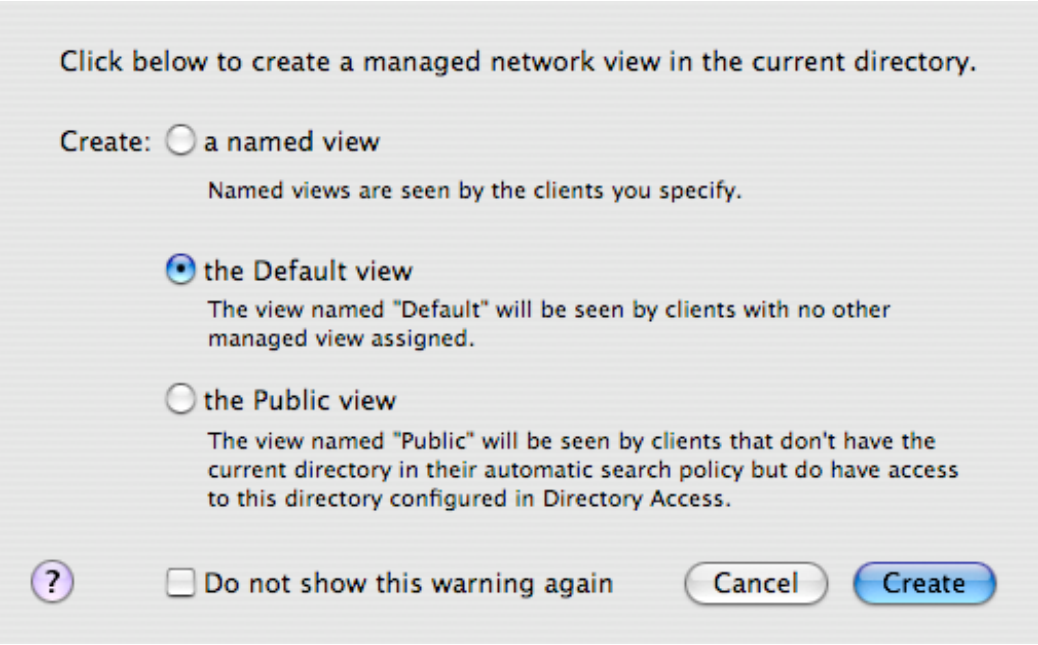
i. Managed Network views

One interesting feature introduced with Tiger is Managed Network views. This feature allows the administrator to organize the view a user has of the network based on accessing the 'Network' sidebar icon or menu item. For example, a user has access to all file servers in the 'Network' view by default.

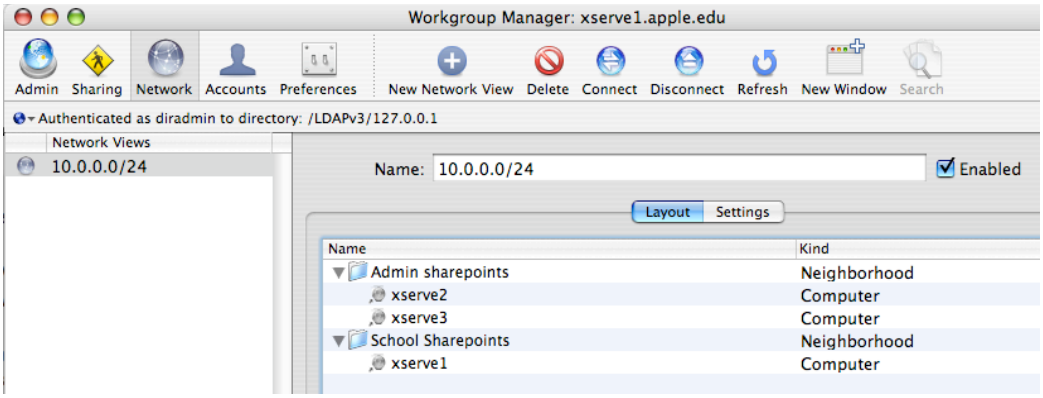


Default view of a sample network - all servers visible

Using Workgroup Manager to change the default Network view, the user's experience can be changed to contain only the items the network admin wants to be visible to specific users, groups, or sets of computers. Different views can be created based on subnet also.



The 'New Network view' dialog from WGM

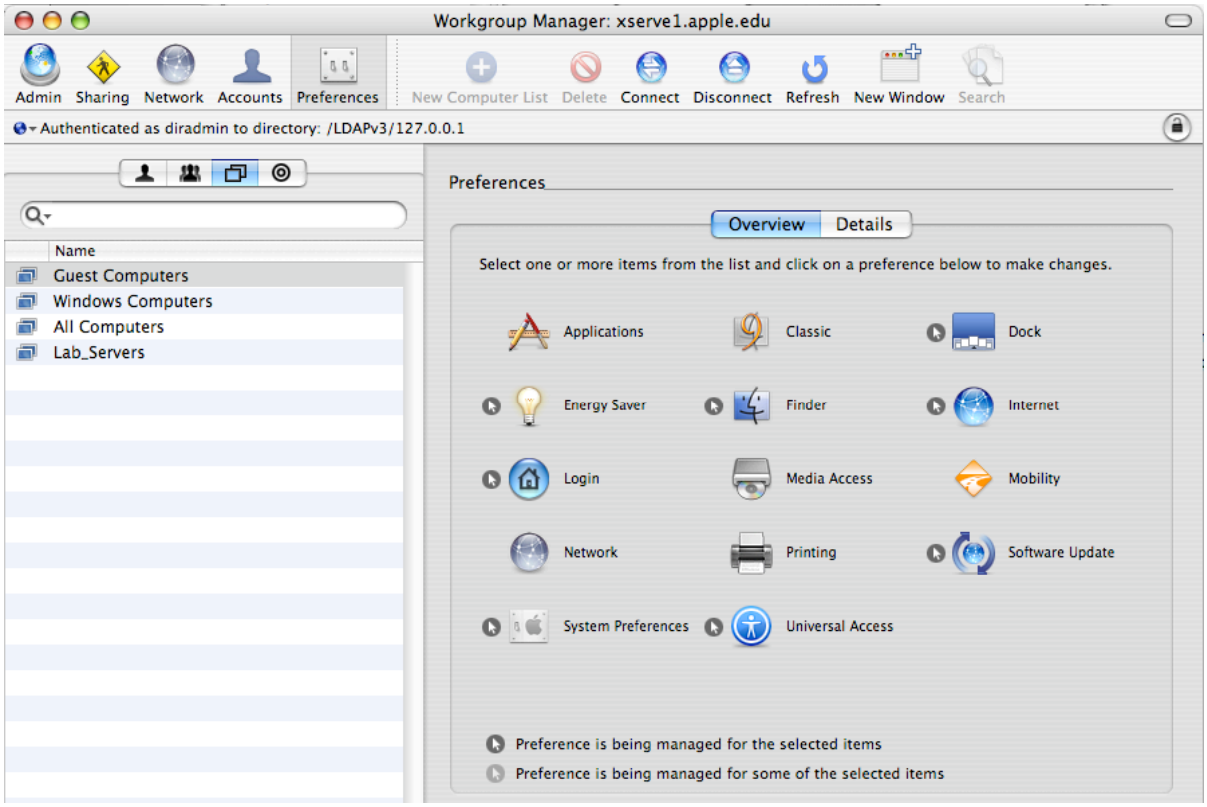


A 'named view' using the local subnet

Once the network view is in effect, the client system will see only the items chosen for view. This does not stop a user from connecting to a specific server by name or IP, it merely hides the servers or redisplays them from the user perspective. The other settings for all the various options for Network managed views is laid out in the User Management Guide.

ii. Basic Preference management

The most used feature in MCX is the basic preference management process. Most of the System preferences, as well as Finder, Dock and Application settings, can be managed by a central administrator. These settings can be applied to sets of computers, groups of users, and even at individual user level. Settings are also cached at local workstations for offline use. This means that if the computer is removed from the network, management settings will still apply.

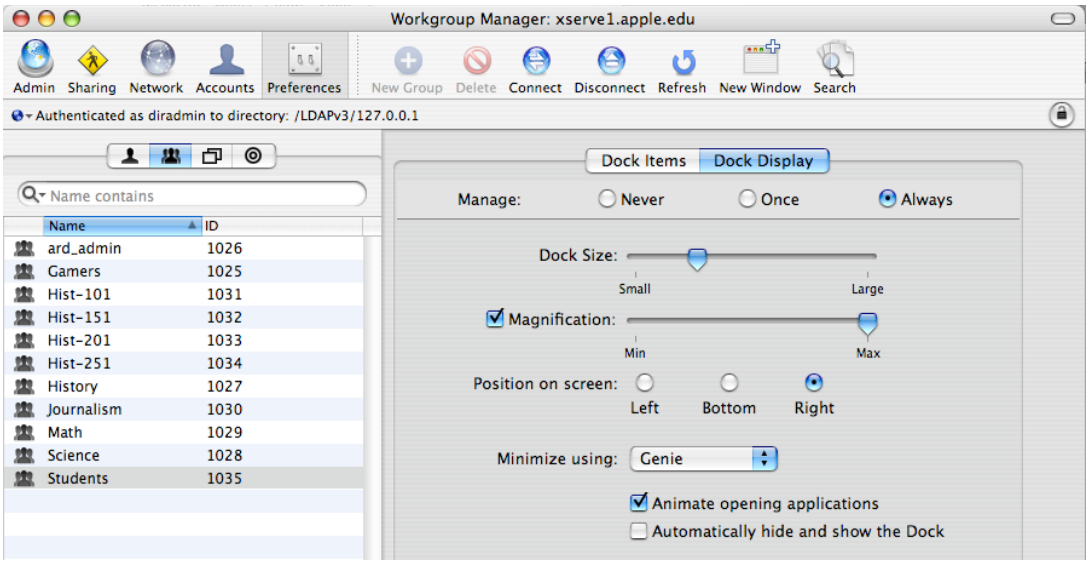


The primary preference management pane

The settings made are applied through a 'chain of command' with the settings for an individual user account having the highest priority, followed by computer account settings and finally by workgroup level settings. There are also two different types of settings that can be made - list or binary. This means that you can establish a list of items to be displayed or allowed, or you can establish a specific value for an item, such as 'on' or 'off'. Let's look at an example of all of this by exploring the most visible of all preferences - the Dock settings.

In this example, we will set the Dock display and the Dock items for all three of the account types - users, workgroups, and computers - just to view the final effect of managing these preferences. First, we set the baseline at the lowest level - workgroup.

A workgroup is a group account that has preferences set within that group. By creating a group with managed preferences, the users belonging to that workgroup will be subject to the settings when they log into a computer using that group. Users can belong to many groups and workgroups; but they can only choose workgroups at login. At that time, the settings of that specific workgroup will be applied to their user experience. Workgroups have the lowest priority for managed settings. If you set preferences at workgroup level, they can be over-ridden by setting differing preferences at computer list, or user account level. The managed client system (mcx) will actually work out all of the differences during the login of a user, presenting that user with only the final settings they are allowed to have.



Dock Display setting managed for 'Students'

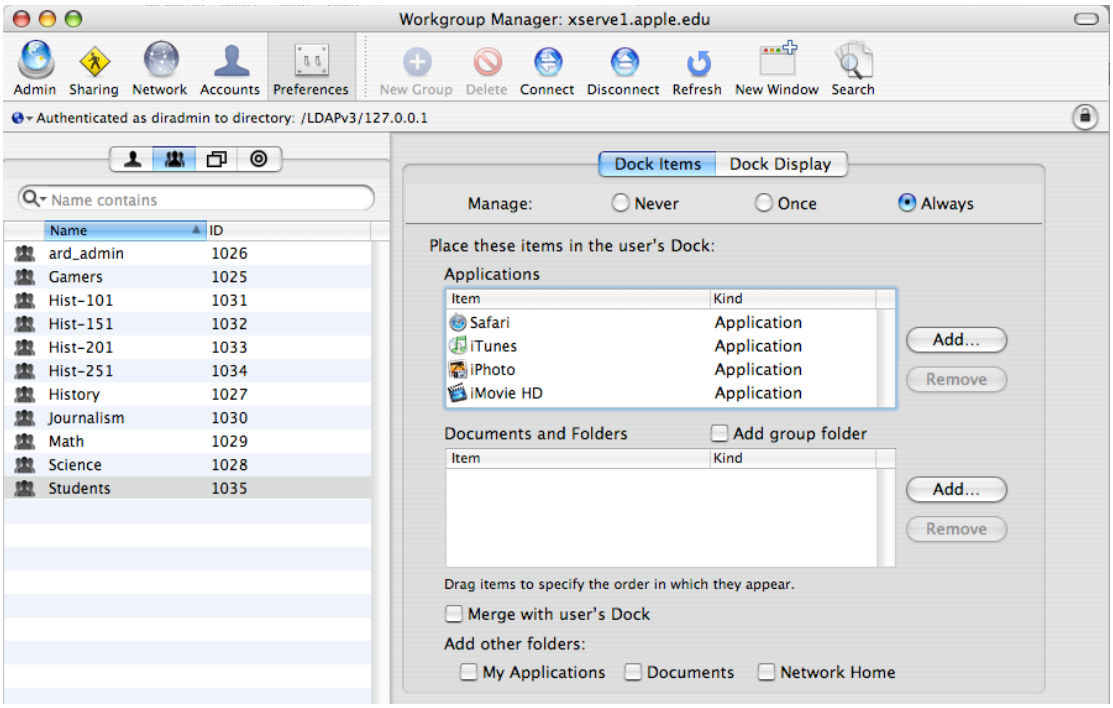


The example setting is to determine the Dock display for the users. In this case, for 'Student', we will set it to display the Dock on the right side of the Desktop and make it specific size.

If you then create a different group, say 'Teachers' and set their Dock to be on the left side (feels sort of like the Lilliputian thing - Big Endian vs Little Endian), a user logging in as a member of the 'teacher's workgroup will get the Dock on the left, a user logging in as a member of 'Students' will get their Dock on the right.

This shows the 'binary' setting, in that the Dock can only be on one side or the other. There is the other type of setting - 'list' which contains items assigned to a user, such as Dock items or applications. The next example is using that list setting in the Dock preferences.





Workgroup 'Students' with Dock set to display four items

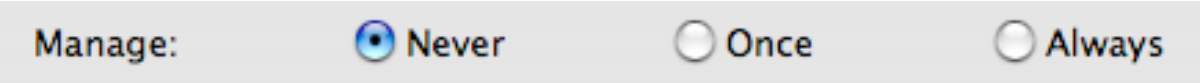
List settings follow a specific hierarchy of layout, in that a computer account, a workgroup, and a user account can all have lists of items available to them. Each of these lists will be pulled together at login to present a unified view of all of the lists.

For example, if the 'Students' workgroup has for its Dock items - Safari, iTunes, iPhoto, and iMovie - and you add the items - Preview, iChat, and TextEdit - to the computer account, then the final list might look like this (I moved the Dock back to the center for publishing space...):



Note the items are in order from computer to workgroup

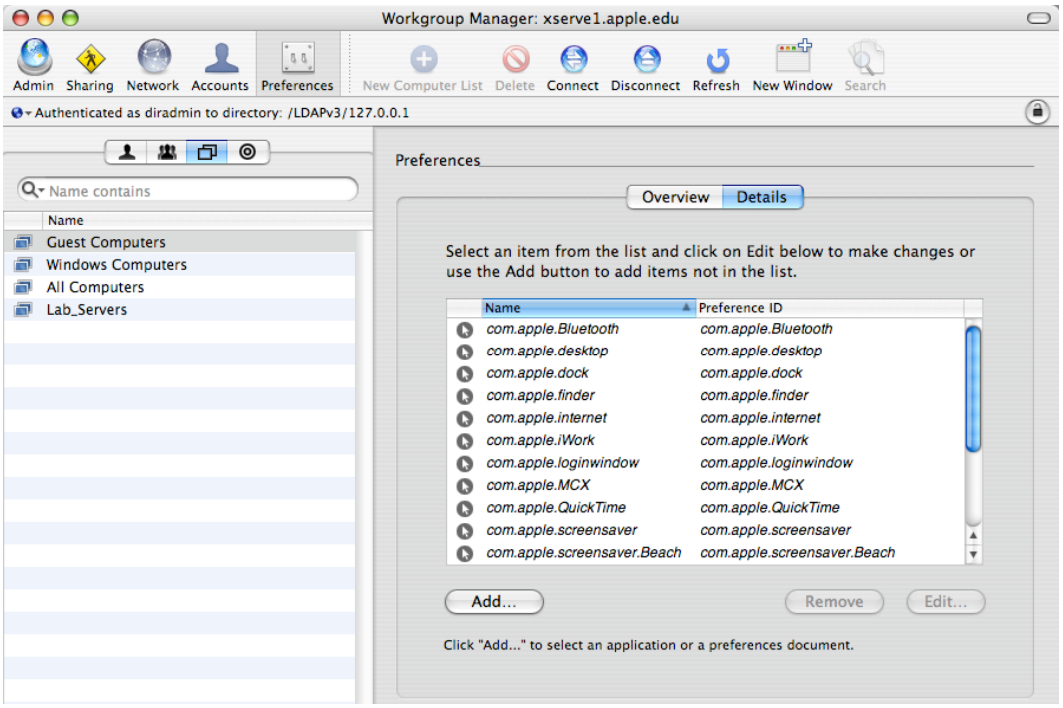
Another key to setting up managed preferences is understanding the use of the two primary levels of control - 'Once' and 'Always'. 'Once' is a setting that provides a guideline by allowing the user to change the settings. 'Always' dictates that the settings made are not to be changed by the user. If the user is allowed to change the settings made by a 'Once' configuration, if the admin changes the settings at a later date then they are "once' again set to what the admin wants. Consider using 'Once' if you are only trying to guide the user's experience, versus using 'Always' to fix their experience.



In the next section, I'll go into details on how each preference setting can be used and how they interact. Meanwhile, let's go look at the latest Tiger feature - Preference Editor.

iii. Details and the Preference Editor

One limit of the Panther MCX setup was that only the designated System and application preferences could be managed. Even though there was a way to go 'under the hood' to set preferences, the need for a formal, supported process was important. Tiger provides that with the Preference Editor. This tool is located behind the 'Details' pane in the 'Preferences' section of Workgroup Manager. As with the basic settings, the Preference Editor sets items for computer, workgroup, and user accounts. Items are added to the Preference Editor list and can be altered or adjusted as needed. Much more detail on this feature is in a later section of the paper.



Preference Editor in Workgroup Manager

The Preference Editor also adds a new preference setting called 'Often' which allows the user to edit the settings; but reverts to admin-defined settings at each login.

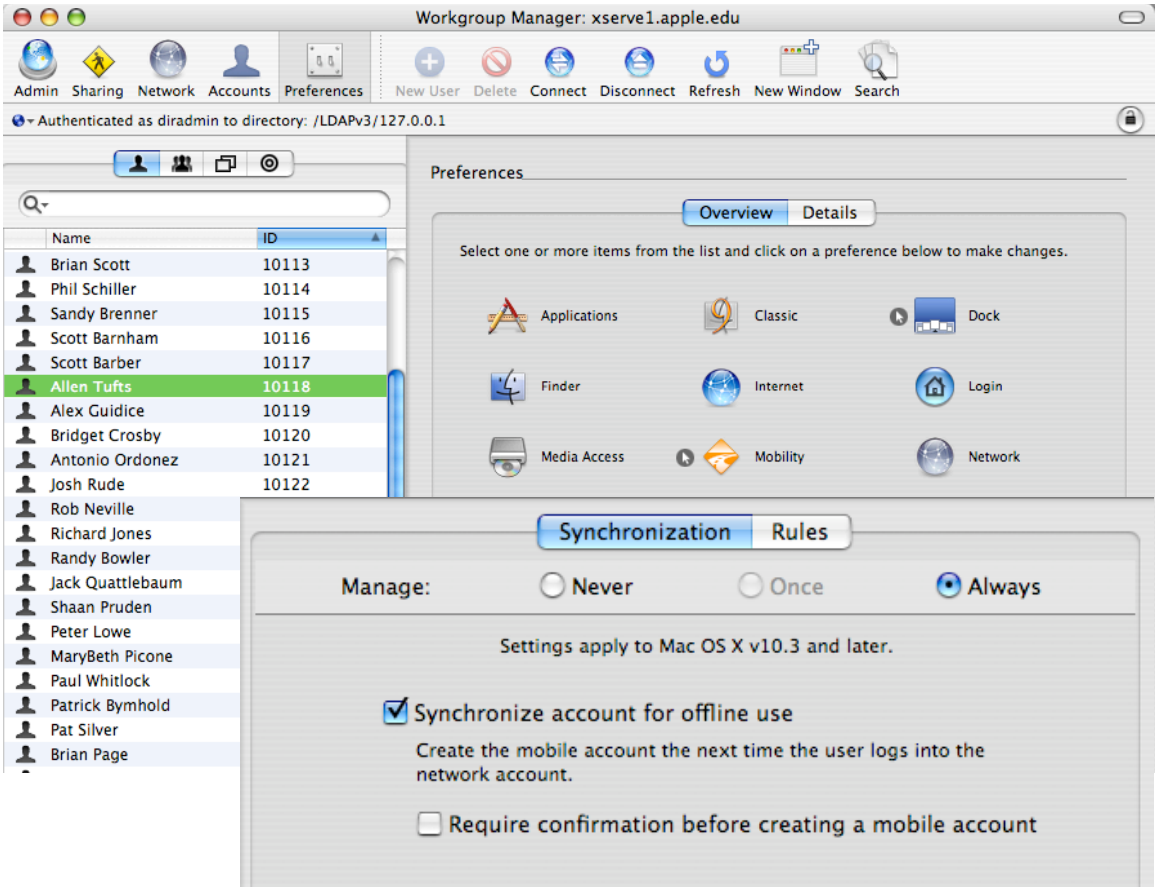
c. Mobile Accounts and Portable Home Directories basics

Mobile accounts were introduced with Panther server/client. The idea is that a network home directory may be too bandwidth intensive for most of the newer applications, and may be really detrimental when using wireless clients. Consequently, the concept of a network-based account with its home directory on the local hard drive came into existence. The account is managed centrally as part of the network directory for password and preference management; but the user's bound home directory lives on the local computer.

This is a very good idea for 1:1 deployments, and can be used to great advantage in shared labs of heavy duty systems running iLife or ProApps. When you think about it, it just makes sense - you have a user launching Safari and all of the caches and downloads are kept local. Tiger even adds the ability to synchronize files with a network home directory to allow a user the ability to move from one system to another while keeping their key documents and settings.

i. Defining mobile accounts

Users who need to have their home directories local can be set to do so by either defining the preference for that user specifically, or by setting the computers to create a mobile account for all users who log into that system. You could even establish mobility at group level, perhaps creating a group of users designated as 'mobileusers.'



Designated user set as 'mobile'

The 'synchronization' defined at this point is the user account information only. The data stored in the user's network directory account is cloned to a local user account and will be synchronized on a regular basis. This does not include any of the files stored in the user's home directory. For that to happen, we have to add 'Portable Home Directories.'

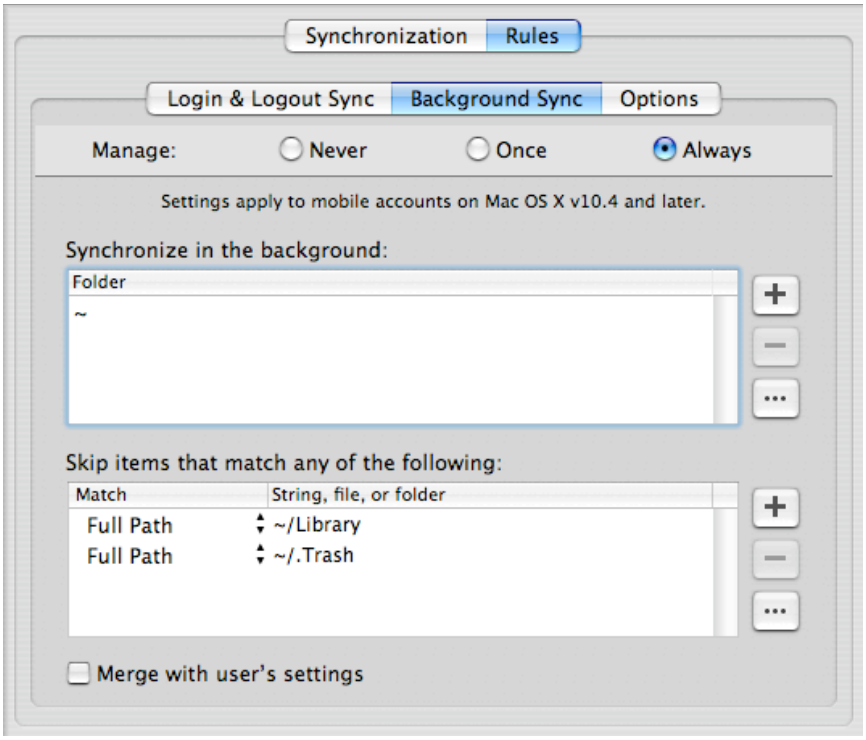
ii. Portable home directories

This mechanism allows the files and folders inside a user's home directory to be synchronized with their network-based home directory. The ability to sync with the network

homedir allows user's to change machines and keep the same documents from system to system. For example, a user with their own dedicated portable or desktop system can sync their Documents folder onto the network homedir. Then, if that user goes to a shared lab and logs onto another computer, they would have that same Documents folder available to them. Note - this is **not** a backup mechanism as designed. If you sync a bunch of files to the network homedir, then delete them from your local mobile account's homedir, at the next sync operation, the files at the network homedir will be deleted also. It's a true file-file synchronization.

iii. Synchronization options

Within the Mobility preferences are rules for managing portable home directories. The two basic sync options are 'Login/Logout' and 'Background' or periodic. Under the current version of Tiger (10.4.2), sync for the user's /Library (referred to as '~/Library') is not supported; however, I am providing a way to activate that process later in this paper. Being able to sync specific items at login/logout or in the background depend on when you need those items to be available. If the files you want need to be there as soon as you login, then setting them up to be sync'd at login is a good idea. Background sync can get around the problem of having a lot of files to sync at any one time.



Synchronization options

For example, you might want to have the items in a user's Desktop folder available to them at login, while wanting to sync their Movies folder during the time they are using the system. This would help when you were creating a large iMovie or iDVD project. Since the sync process copies files, a large project in iMovie would consist of many smaller sections, so it would be sync'd a few pieces at a time.

You can also set up manual sync, where the user decides when the best time to sync is. All of these options will impact the user in different ways, as well as determine the load on the network.



Background sync can be manual or timed - or both.

All of this is covered in much more detail in the section on 'Preference Management in Detail.'



"This makes me really sleepy..."

5. Directory Services tips

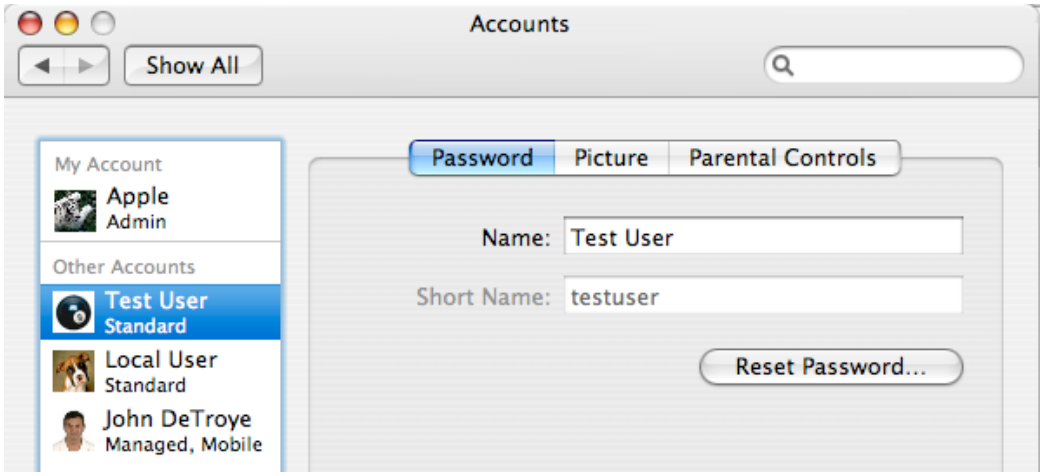
Setting up directory services properly will make or break a deployment. All of the management settings and preferences set for managing clients mean nothing if your directory service configuration is out of whack (a technical term). There is no such thing as “If I turn off management, I might be able to get to my home directory.” Since all of the management stuff (the reason for this paper) is dependent on the proper setup of directory services, I will spend some time on doing it right.

I am also not going anywhere near any other directory services than OpenLDAP as we (Apple) use it on Tiger server. There are lots of people out there much smarter than I am who know how to deploy, and explain, non-Apple directory services.

a. Account planning

When you begin planning for large system deployments, the thought always surfaces as whether or not network accounts should be used - or should you just stick with generic accounts for yet one more year. There are a couple of key points to consider when determining how and what you are going to use for your accounts.

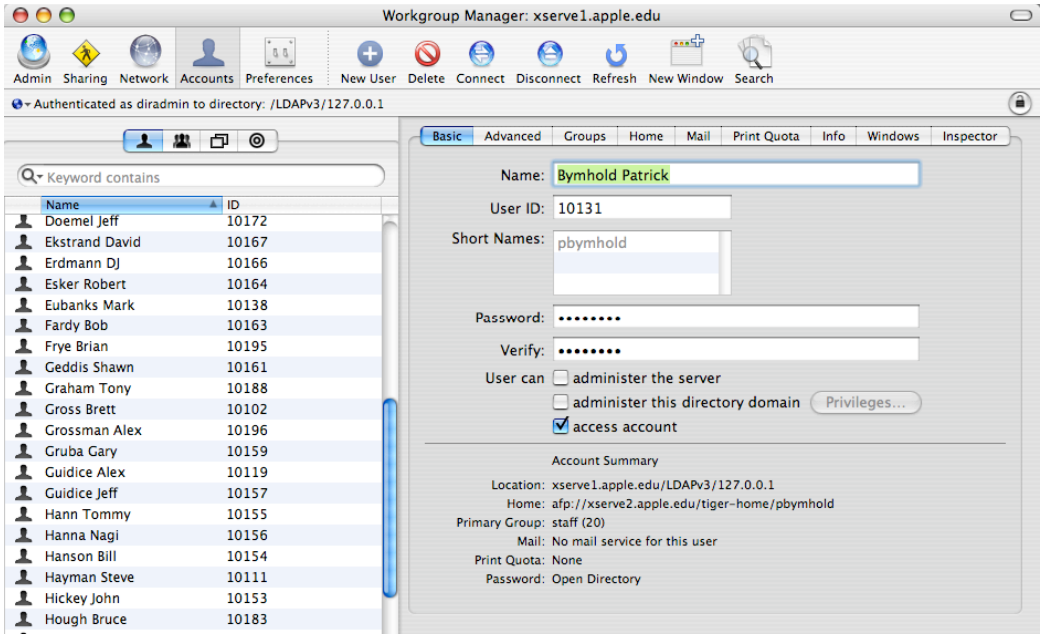
Local accounts, or generic accounts, such as ‘Student’, are very useful for common systems that are used for basic services. The upside for these is that you can easily create a simple basic account on a system and clone that system to many others with the same settings. The downside is that you will have no idea which ‘Student’ is on that system at any specific time. In fact, some sites have gone as far as to set up generic accounts on the workstations, yet created network accounts too so the users’ could log into a file server to store documents. Local accounts tend to have a single password that is common across all cloned workstations. This also means that if someone discovers one password for an iBook, they now have all of the passwords for all of the iBooks.



Local accounts are usually generic

Network accounts require more planning; but are also manageable over the network. The biggest consideration with network accounts that I see is determining how far away from the user is the account managed. It is possible to create a single massive directory at district

level that may cover all school sites. This makes the account list reachable from a single point, manageable from a single point, and prone to possible failure from a single point. Luckily, the use of OD replicas can provide a measure of safety and performance. Centralized management can also happen if there are several distinct master databases. Having a different db at the high school, middle schools, and elementary schools can still be managed from a single location on the district WAN.



Network accounts defined in Open Directory

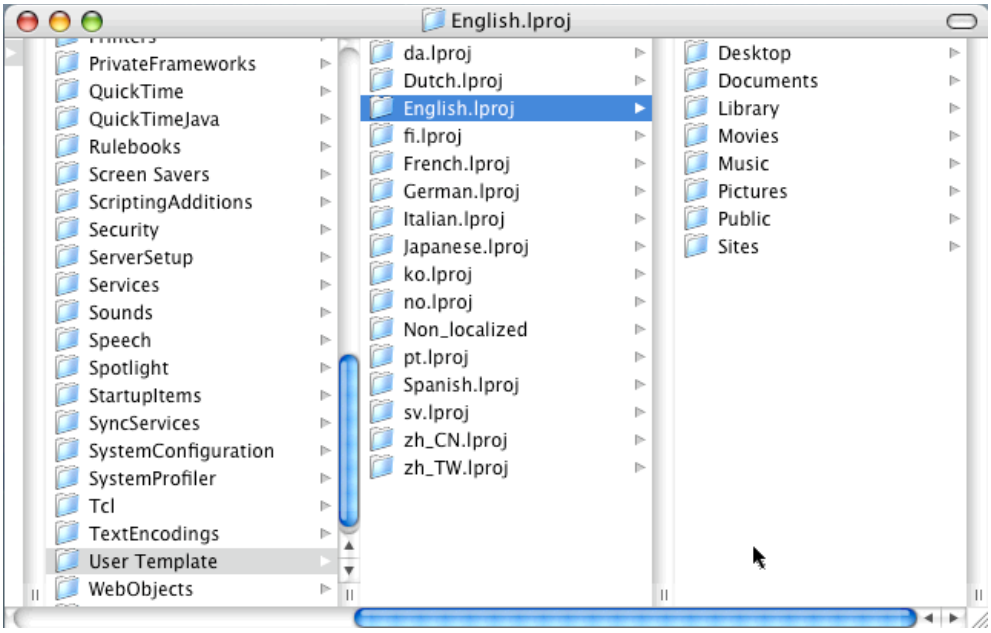
As for sizing, the LDAP directory architecture in Tiger server will support over 100,000 user records. Each authentication server can support a thousand concurrent logins, and a site can easily support a dozen or more replicas to provide parallelism for the authentication server. This means you could deploy 10's of thousands of Macs all using Tiger from the back to the front of the network. Cool. Since setting up local accounts for generic users is too easy, we are going to concentrate on creating network based accounts for the duration of this section.

b. Home directory considerations

One of the biggest concerns with deploying user accounts is dealing with home directories. If you use a local account, then that account has a single home directory located on that specific workstation. This works great if the user is always the same, or always uses the same machine. If more than one person uses this system, each one may change or delete the other user's documents. When you use network home directories, you are at the mercy of network traffic and bandwidth. Since all of the user's temporary files and caches, as well as documents, are created and saved by default inside the home directory, this means literally thousands of files being opened and edited across the network.

When planning for home directories, you should look at the template used by the System for creating these. While there is an identical template located on each workstation, it

actually isn't used at all by Tiger when working with network accounts. The template is located on the server you are using for your home directories. If you have multiple home directory servers, then you will have multiple templates to deal with. In order to edit the template, you will need to log onto the server as 'root' - which means you need physical access to the server. (For those of you waiting for me to jump right into using Terminal to edit the template, give up....)

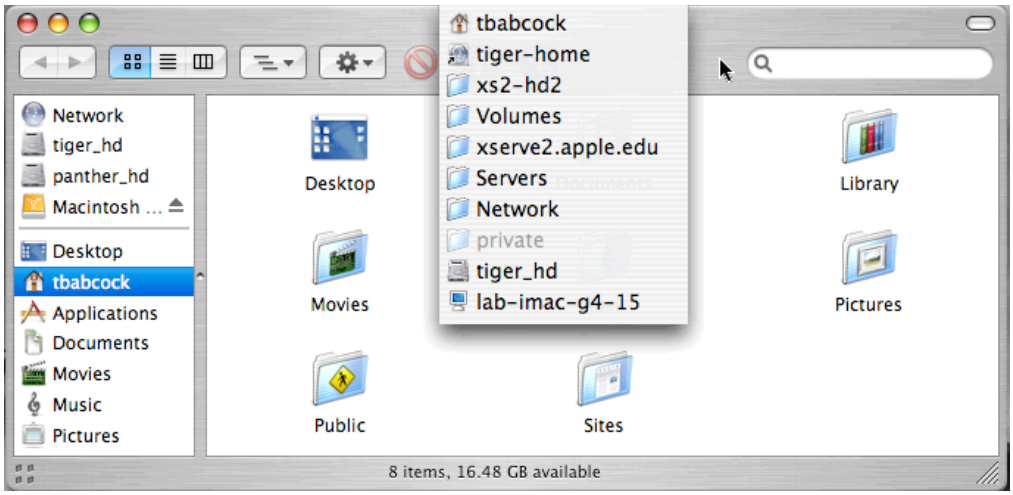


/System/Library/User Template on homedir server

This template can be edited as you see fit - with some conditions. The folders used by the iApps are very aggressive - if you launch iTunes, for example, and you had decided to do away with the 'Music' folder, it will be recreated. You can safely get rid of 'Public' or the 'Explorer' folder inside of 'Library'. The idea is to mold the user's home template so it reflects the environment needed for teaching/learning.

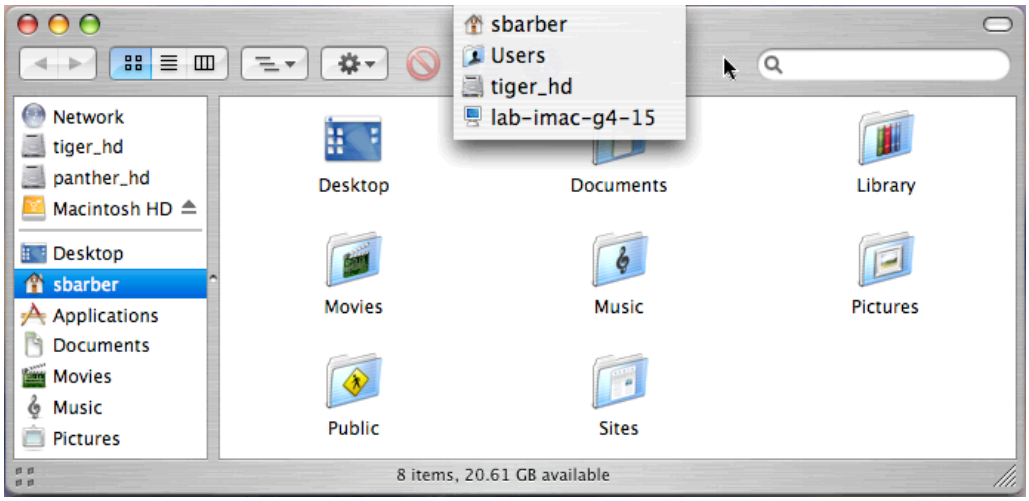
c. Mobile Accounts versus everything else

Since the focus of Tiger server, and this paper, is on managed clients, I want to keep the noise level up on mobile accounts. When the concept of true network home directories was introduced by Apple, no one else in the industry was actually mounted a complete network drive for the logged in user. The idea was that the computer would become secondary to the process, and that the user could easily float from one machine to another with all of their document, settings, their 'essence' intact. The problem is that in order to achieve the best performance, the amount of network bandwidth required is pretty substantial. When running high load applications, such as any of the iLife suite, or even a browser, the amount of traffic sent back and forth between the client system and the user's home directory can become overwhelming. Some sites got around the bandwidth issue by using only local, generic user accounts. Others resolved the problem by creating scripts that relocated many of the cache folders and high traffic sets to the local drive in places like /tmp.

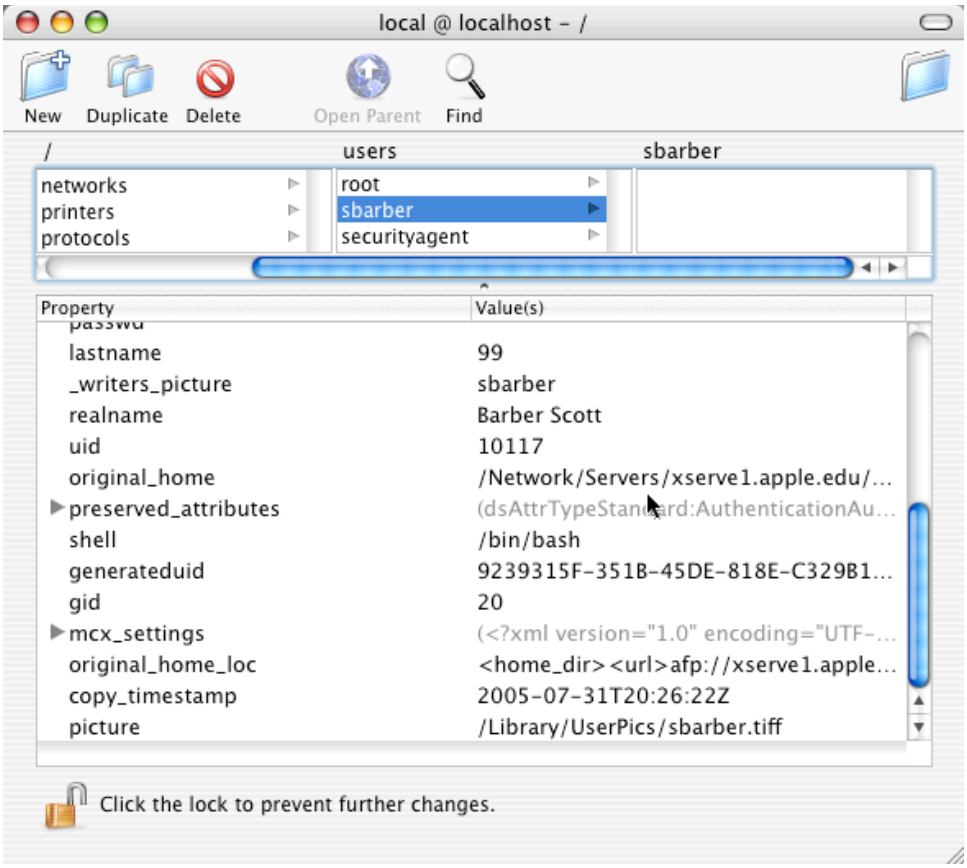


Network user with true network home directory

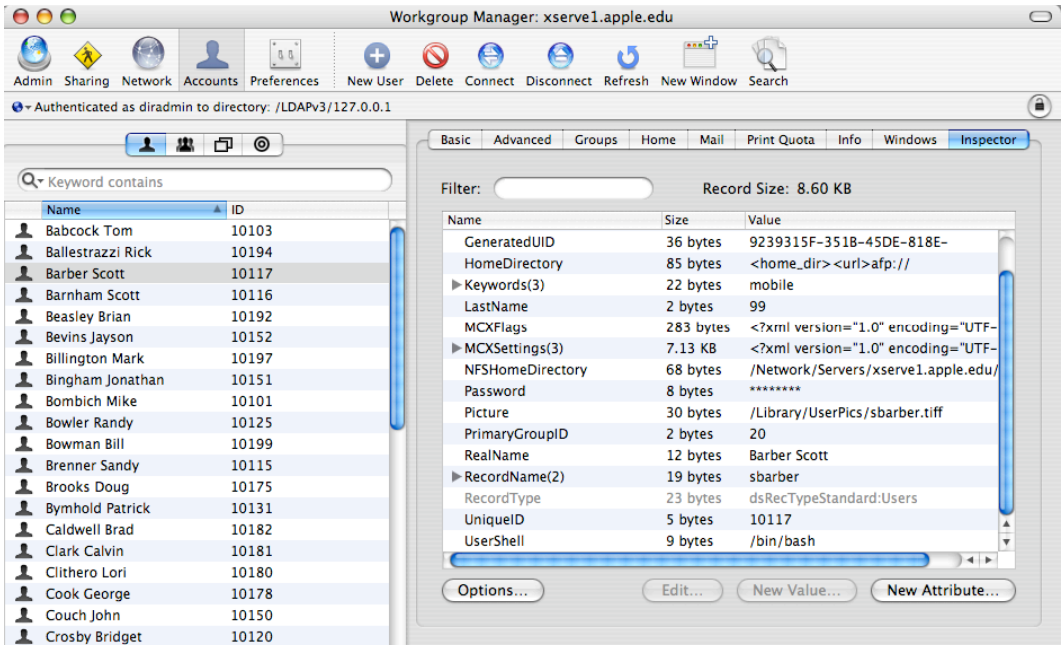
In Panther(10.3), we introduced the concept of the 'mobile account.' The strength of a mobile account is that the account is centrally managed as part of a network directory, and that the user's home directory resides on the local client. The two flaws in the concept under Panther are that mobile accounts are created on the local client with no aging mechanism to delete them automatically after some set period of time, and that when the user moves from one client system to another, none of their settings of documents moved with them. Tiger resolved one of these issues, making mobile accounts possibly the best idea since chicken liver spaghetti. :-)



Network user set up as mobile account with local home directory



Local NetInfo data on a mobile account



Open Directory / LDAP data on a mobile account - note that all of the key data points match in both locations

The data that comprises a user's account is synchronized between the network and the local directory databases. This is done through a datetime stamp that is kept in both locations. This information is checked several times, and anytime the datetime stamp get out of sync, the systems resync and the local machine caches the correct info. For example, if the user changes their password when they are logged in as a mobile user, that change is sync'd at the network directory at logout. If the network admin changes the user's record, that information is changed at the client the next time the mcx_cache is refreshed, such as at boot time, login, logout, or network service change.

Under Tiger, we added 'portable home directories' to the mobile account. This adds the ability to synchronize the user's document and settings with their network home directory. The idea is that the user lives on a local home directory, created as part of their mobile account, and the traffic to their 'stuff' is very localized. At login/logout and/or periodic intervals, certain portions of their home directory are sync'd with the network home. This allows the user to move from one client system to another, such as from their dedicated portable in a 1:1 to a shared lab system, yet keep all of their critical files with them.

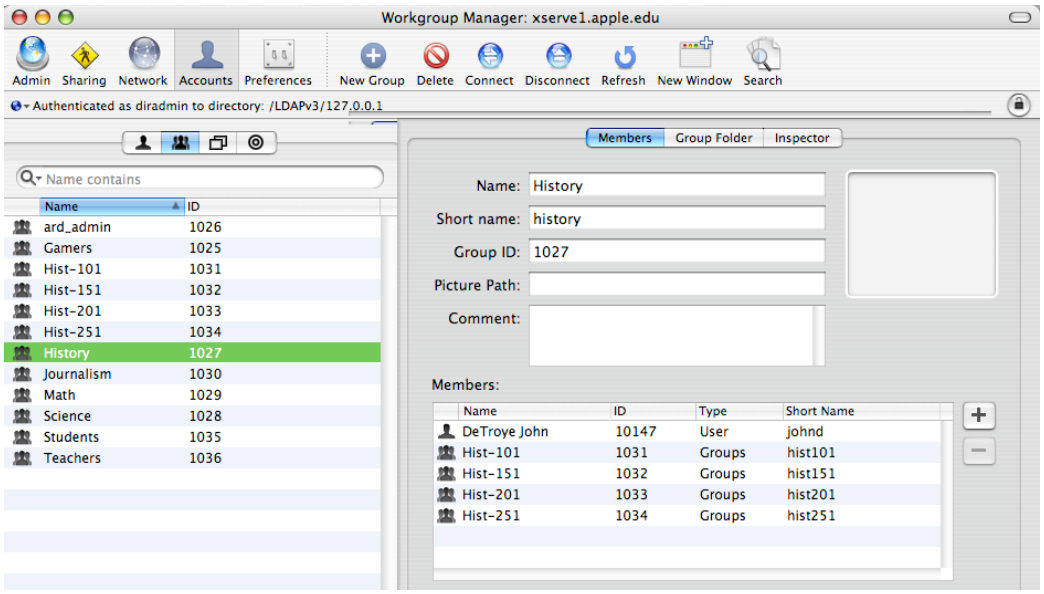


Mobile user with Portable Home Directory - syncing active

With careful setup, and a few hints (to come in later sections), you can set up a very dynamic mobile account/portable home directory environment that allows the best performance you have ever seen on your network. It makes sense to use this setup over generic accounts, and especially over pure network home directory accounts.

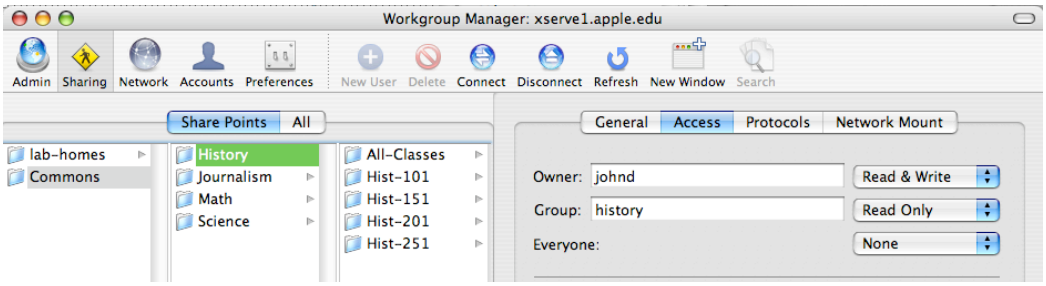
d. Groups and workgroups

Groups of users are set up to help define the permissions those users have when accessing folders or sharepoints. For example, a group of students may belong to a group titled 'History' so that they can access any sharepoints or folders that were created for that group.



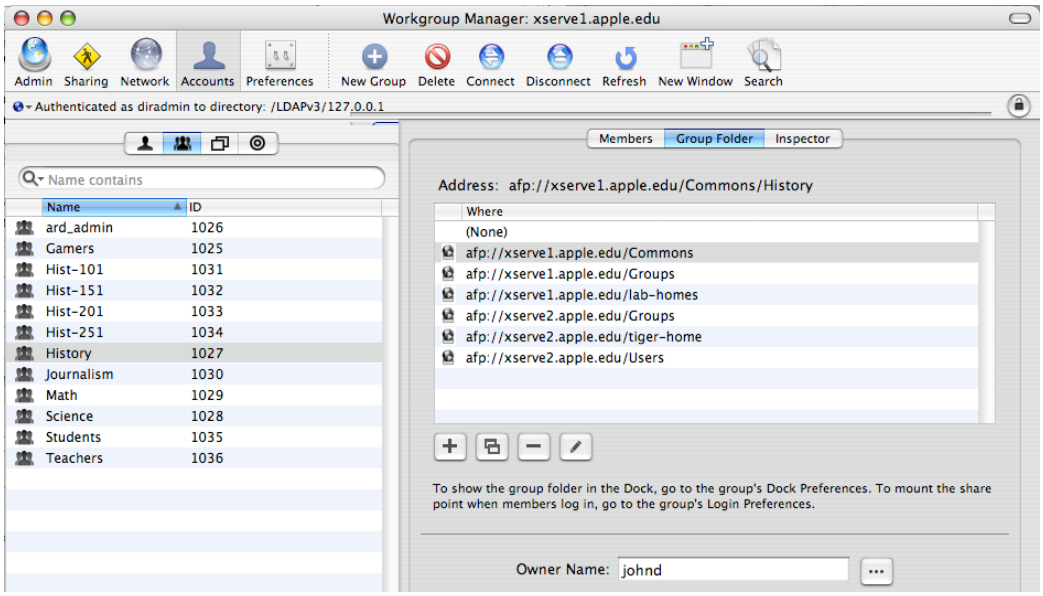
Group with owner and nested sub-groups

Note that under Tiger, you can now assign groups to other groups. This is called ‘embedding’ and make management of groups much more powerful. **Note:** Under the current version of Tiger, the use of embedded groups does not allow the preferences of the parent group to be inherited by the sub-group. This diagram is accurate for group permissions; but does not function for management.



Basic folder permissions set for a group - ‘History’

Another feature that belongs to groups is the ability to have a group folder. This folder is used as a common network folder for that group, with all members automatically having access to that folder. The folder itself is not, by default, automatically mounted for a user at login as the home directory is. In order to accomplish that, you need to use managed preferences (mcx) settings. The directory admin establishes the group folder location and owner, then the server that holds the group folder will automatically create that folder overnight during its periodic maintenance, or you can use the ‘CreateGroupFolder’ command to force create the folder on the file server. One of the problems with group folders currently is that the default content of the folder does not exist as part of a template anywhere. The items the system places inside the default folder are ‘Documents,’ ‘Library,’ and ‘Public.’ First thing I always do is delete those items and create the contents as I see fit. Group folders are a huge part of workflow management.



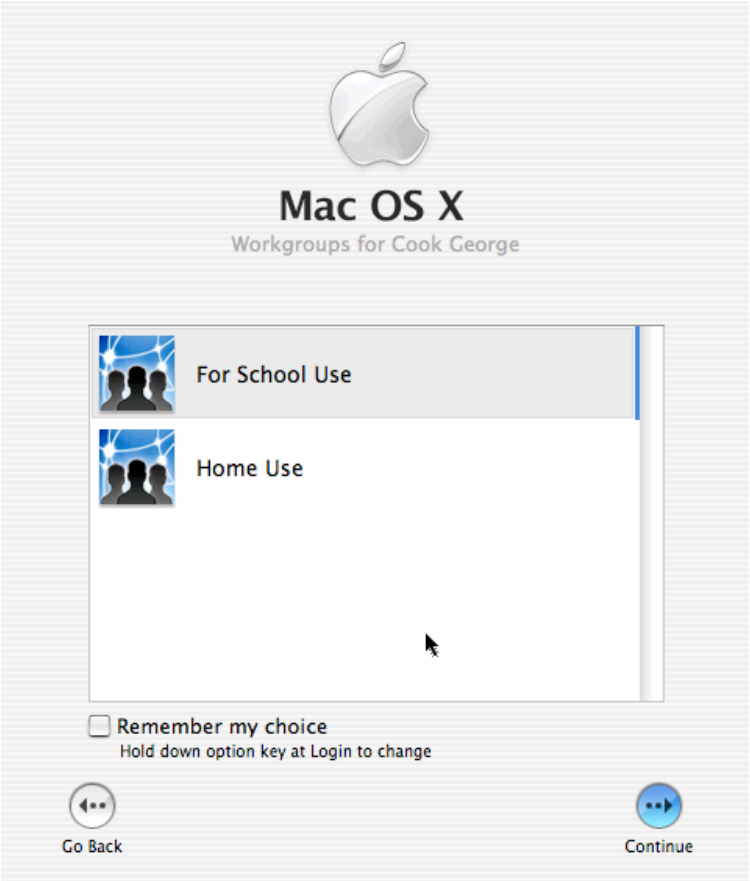
*Group folder for 'History' - note that an owner **must** be defined*

Workgroups are the next level of progression for group. When a group has managed preferences assigned, it becomes a workgroup. The key here is that a group controls only folder/file permissions. A workgroup may control the entire user experience, as well as folder/file permissions. In a managed client deployment, a user may belong to dozens of groups - but can only log into and select workgroups. For example, we will create two workgroups to manage basic functions for users while they are at school or at home. The two workgroups are 'For School use' and 'Home Use' (this was done so that the default group will be the school one, since it shows up highlighted at the top of the window).

When a user belongs to both workgroups, they will be allowed to choose which one they wish to use at that time. If the user is only allowed to use one workgroup while on that specific computer, then they will go directly to the Finder. The only exception to this rule is if local administrators are allowed to bypass management when logging in.

Access to workgroups can be set per set of computers (next section). This insures that the workgroup chosen makes sense for that client system when a user logs in.

For example, you may create two workgroups, one for use at home and one for use at school. The home one should only apply to systems that will leave the building, while the school one should apply to all computers. The key difference may be that the school workgroup will automatically mount various network folders when the users log in, as well as allow all users to access it, versus the home workgroup which may only apply to specific set of users and have no network volumes mounted when in use.



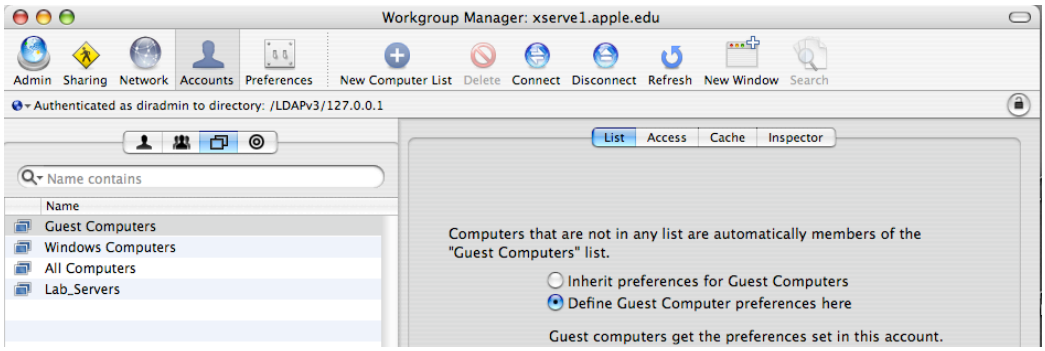
Workgroup picker for a user with two choices

e. Computer accounts

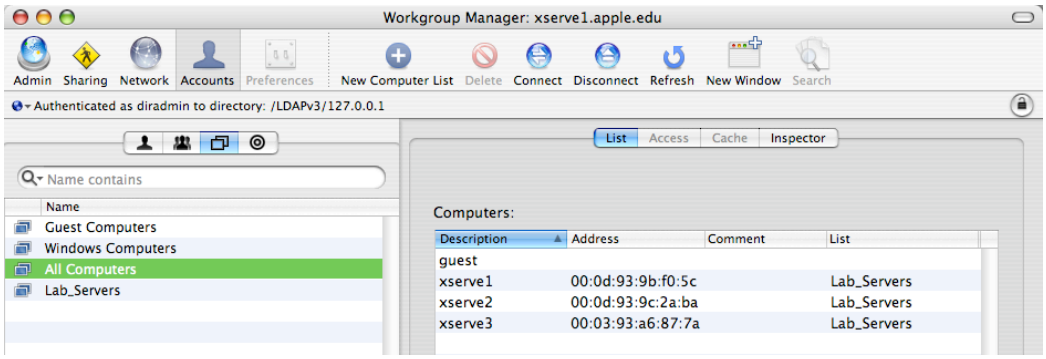
At the base of the directory for a network of managed computers is the computer account. There is a generic ‘Guest Computers’, ‘Windows Computers’, and ‘All Computers’ set. The ‘Guest’ set covers every computer that is not called out uniquely by another set, as long as the computer is bound to the directory you are working in. The ‘All Computers’ set shows a machine record for every computer that has been added to a unique set.

Computer accounts let you segment your computers by type, or by function, so that specific management actions can be defined for different sets of computers. For example, you might want to isolate your servers from the rest of the computers as far as management settings are defined, so that a user cannot actually log into the server expecting it to become just a really fast workstation. “Gee whiz, this one runs iMovie really fast!”

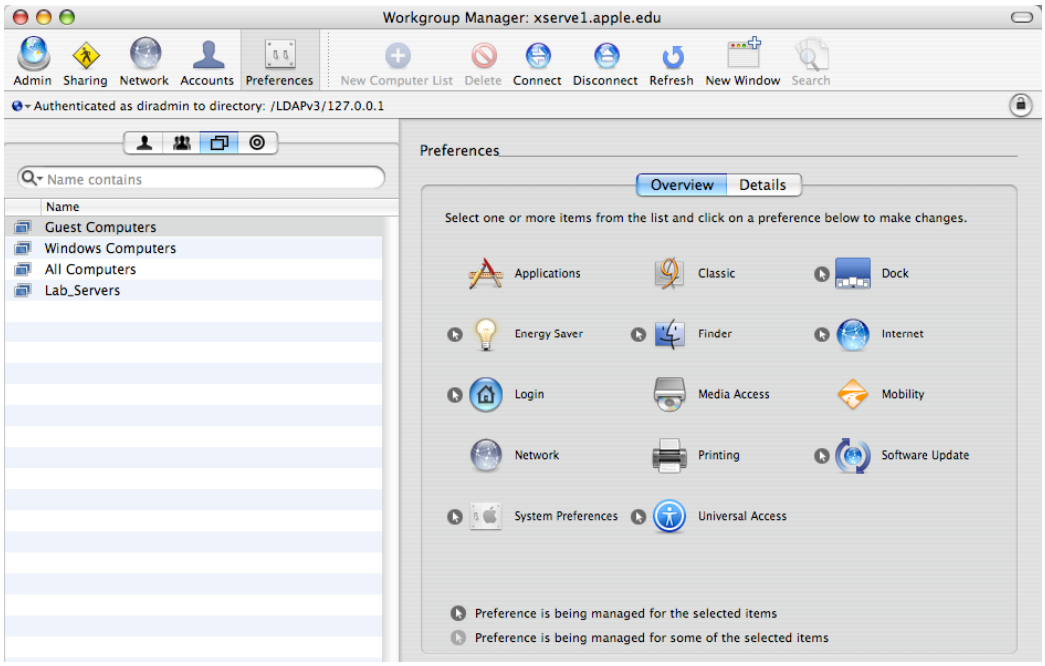
Client management settings are also kicked into operation by use of computer accounts. The ‘Guest’ set can be set to ‘define’ management, or you could use a unique list in a set to establish management. The walkthrough shows how this done.



Guest Computers set to define management



'All Computers' displays every computer named uniquely



Management is also set at computer level

6. Managed Client for X (mcx) in detail

While it is very possible to set up a very well-defined and clean deployment with managed systems without knowing the deep, grungy details of how it all works, a brief exploration of the process is pretty useful. In this section, I will try to explain the various parts that comprise the management system, as well as how they all work together to establish a user's experience at a computer when they log in.

a. Preferences versus Security

The MCX process is defined as preference management model, not a security model. The reason for this difference is that for every method of setting controls for a user's access or use of a preference or application, there may something in the OS or an application that overrides this. I bring this up only for informational sake, and to make sure everyone understands that the harder you try to make to make a system foolproof, someone out there is developing a better fool.

A good example is in trying to lock down the Desktop picture (there are better examples - but I don't want to give away the really ugly secrets...). While you can lock the preference file that controls the Desktop picture setting, and even go under the hood to lock down the actual picture files, just opening iPhoto and selecting a button to change the Desktop picture overrides that setting. It's a matter of having some applications just not behave according to the wishes and desires of the Desktop Management engineering team. (If you find a pet peeve with the system, escalate to Applecare - make your wishes known and engineering will try to fix it!)

b. The logic of Managed Client

There is a very well-defined logic to the process used in setting up a managed client. Essentially, the client computer boots to a login window with a copy of the preferences set by the network directory. At login time, a user gets their user and workgroup management data combined with the information defined for that computer. A final, single list of managed items is presented to the system by the management process, and the user will get the benefit of all of those settings.

Let's walk through the process in a little more detail:

- Client boots to login window

- During boot, the workstation's local mcx code checks the timestamp of the cached management information against the timestamp of the management information stored in the network directory.

- Any discrepancy in the timestamp forces a refresh of the cached management values from the network directory.

- Pre-login managed values are applied to the login window and to the system running underneath. These values include any text to display in the login window, as well as the specific layout of the login window. It may also contain other core process settings.

- User begins login process

- All authentication processes kick in. This is not part of the managed client process however. This is the name/password challenge, as well as a check for authority for that user to be logging into this specific computer. If this part fails, the system kicks back to the login window.

- A valid user is then presented with the 'workgroup picker' window if; a. that user is an administrator, or b. that user has more than one workgroup available to them at that computer.

- If the user gets the workgroup picker, they may choose to bypass management (if they are a local admin), choose a specific workgroup and select the option to be locked into that workgroup at that computer, or just choose a workgroup and continue to login.

- The mcx data is once more checked, and the timestamp verified. If there is still a discrepancy at this point, the system will forego using the locally cached values and just use the settings directly from the network directory.

- User sees Finder load and Desktop appears

- Login synchronization, if set, runs and compares the user's local home directory files to that of the user's network home directory. Files that need to be copied in either direction, or deleted, are handled as necessary.

- The management values are (and have been getting) collated and compared, presenting the user and the system with a single, unified set of management preferences that apply to everything from Dock placement to automounting of group folders.

- User is logged in and working

- Background (periodic) sync activates, or manual sync is activated by user, causing the file comparison between network and local homes. Appropriate swapping, deletion, and changes made as needed.

- User logs out

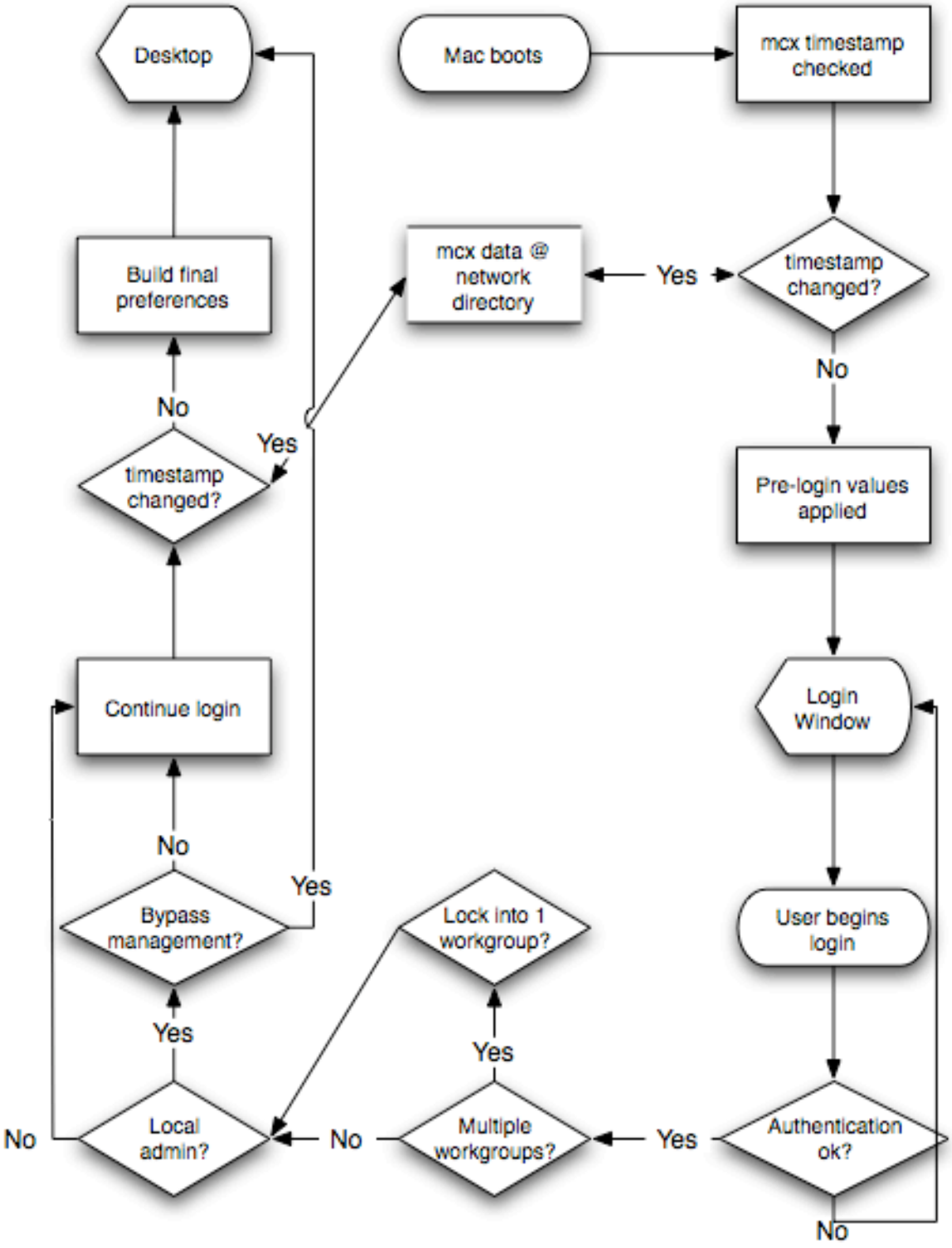
- Logout sync activates, repeating the login sync settings.

- The mcx timestamp is checked again, and any discrepancy forces a re-sync of the mcx data from the network directory down the local cache.

- Login window reappears.

This process is very dependent on a couple things working very cleanly. First, the client must have seen the management server at least once during a boot cycle in order to load the local caches with data from the network directory. Second, the user's network home directory must be available if that user is going to synchronize their local mobile account home directory with their network account home directory. Key to the process is that setting management has nothing to do with the user being able to either authenticate or access their home directory. The management settings are part of the network directory, and as such, if the user can't log into the workstation, they'll never get far enough for their management settings to take effect.

Now, I'm not an artist - but the login process should look like sort of like this:



The managed client (mcx) process for login

c. Understanding the ‘mcx’ stuff

The ‘under the hood’ parts of the managed client process are complex; but a few of them need to be understood if you are going to spend a significant amount of time tweaking or adjusting the settings. The key parts of the managed client process are:

- MCXsettings: The field inside the network directory that contains the management settings for an account (user, workgroup, or computer). An example of an mcx_setting for a workgroup that has some basic Dock settings looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://
www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
	<key>mcx_application_data</key>
	<dict>
		<key>com.apple.dock</key>
		<dict>
			<key>Set-Once</key>
			<array>
				<dict>
					<key>mcx_data_timestamp</key>
					<date>2005-07-14T21:59:39Z</date>
					<key>mcx_preference_settings</key>
					<dict>
						<key>autohide</key>
						<true/>
						<key>largesize</key>
						<real>128</real>
						<key>launchanim</key>
						<true/>
						<key>magnification</key>
						<true/>
						<key>mineffect</key>
						<string>genie</string>
						<key>orientation</key>
						<string>bottom</string>
						<key>tilesize</key>
						<real>40.208332777023315</real>
					</dict>
				</dict>
			</array>
			<key>Unset</key>
			<array>
				<dict>
					<key>mcx_preference_settings</key>
					<dict>
						<key>autohide-immutable</key>
```

```
                                <false/>
                                <key>launchanim-immutable</key>
                                <false/>
                                <key>magnify-immutable</key>
                                <false/>
                                <key>magsize-immutable</key>
                                <false/>
                                <key>mineffect-immutable</key>
                                <false/>
                                <key>position-immutable</key>
                                <false/>
                                <key>size-immutable</key>
                                <false/>
                            </dict>
                        </dict>
                    </array>
                </dict>
        </dict>
</dict>
</plist>
```

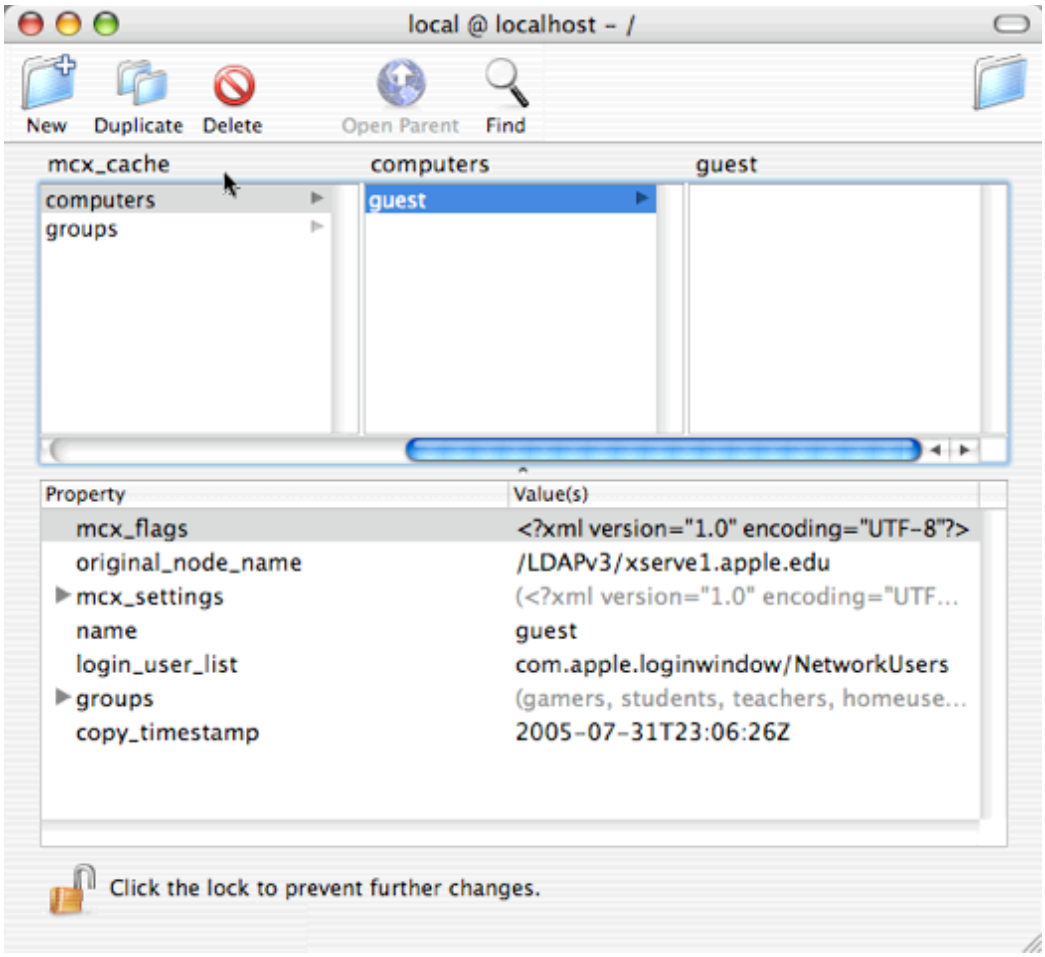
Note that the timestamp is inside this record, as well as the specific settings to be applied to the Dock. Many of the records for fancier configurations are significantly longer.

- MCXflags: The field in the network directory that contains the core settings to determine if management is enabled at that system. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://
www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>has_mcx_settings</key>
        <true/>
</dict>
</plist>
```

In this example, the flag merely dictates that this group has management settings, which turn this group into a workgroup and make it selectable in the workgroup picker.

- mcx_cache: This is the local data record on a workstation that mirrors the settings made at the network directory. There are actually two different mcx_caches. The first one is the true mirror of the network directory with all of the values for computer and workgroup accounts that match the values in the network directory. Here's an example from a workstation's local NetInfo database:



NetInfo /mcx_cache/computers/guest settings

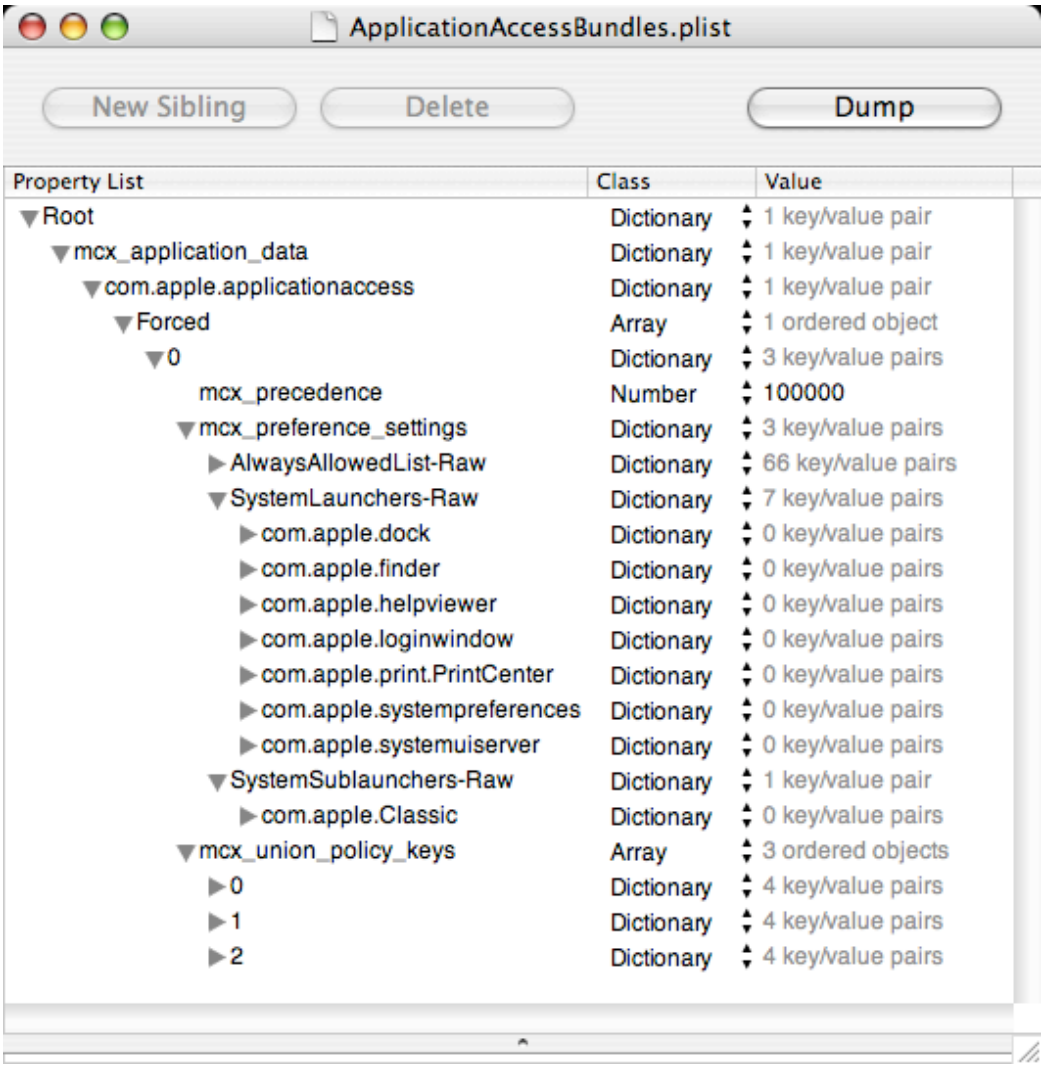
This record contains a duplicate of the mcx_flags and mcx_settings that are contained in the 'Guest Computers' record at the network directory.

The second mcx_cache is located inside the NetInfo /config record, and it contains the computer binding information for that system.

- mcxd: The core system process that contains the key code and settings to make all of this 'stuff' work on the local workstation. It contains a few key files and executables that can be modified or used by a knowledgeable administrator. (**Note - messing around inside here can cause extreme damage to your clients!**)

- ApplicationAccessBundles.plist: Contains the list of applications that are allowed to launch regardless of mcx settings, such as low level OS items. It also contains the priority ordering for negotiating conflicts between settings.

- cinchd: The portable home directory synchronization tool.

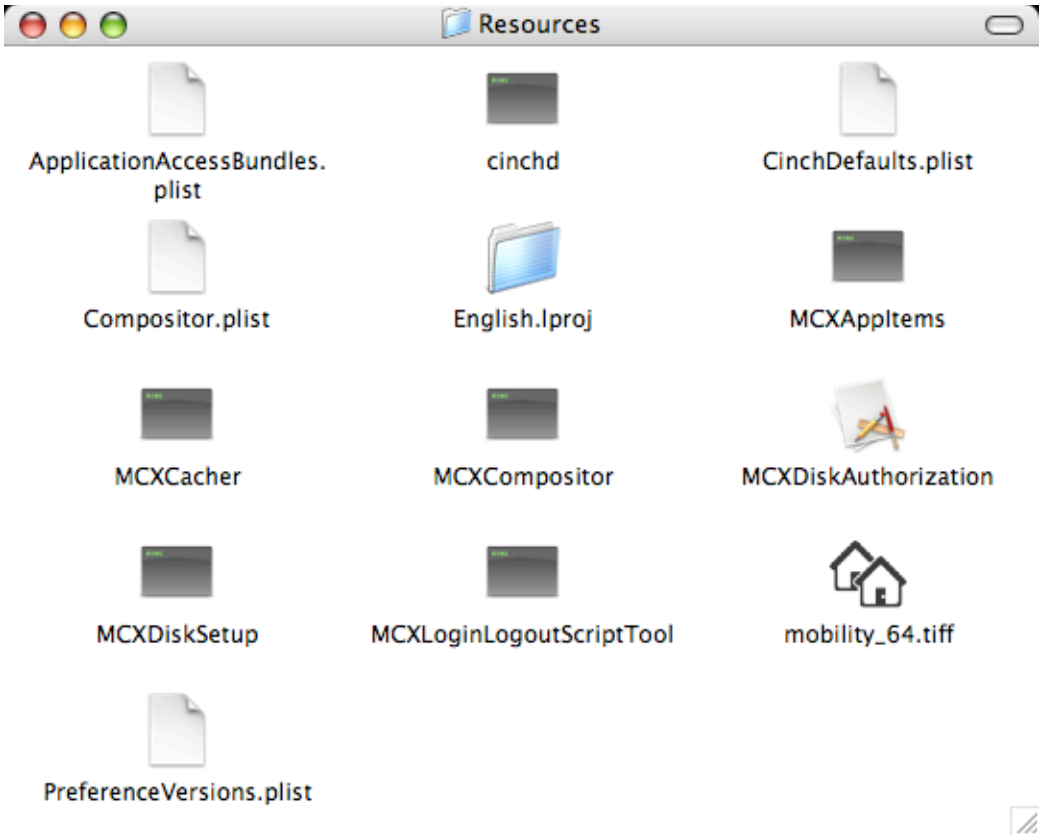


ApplicationAccessBundles.plist

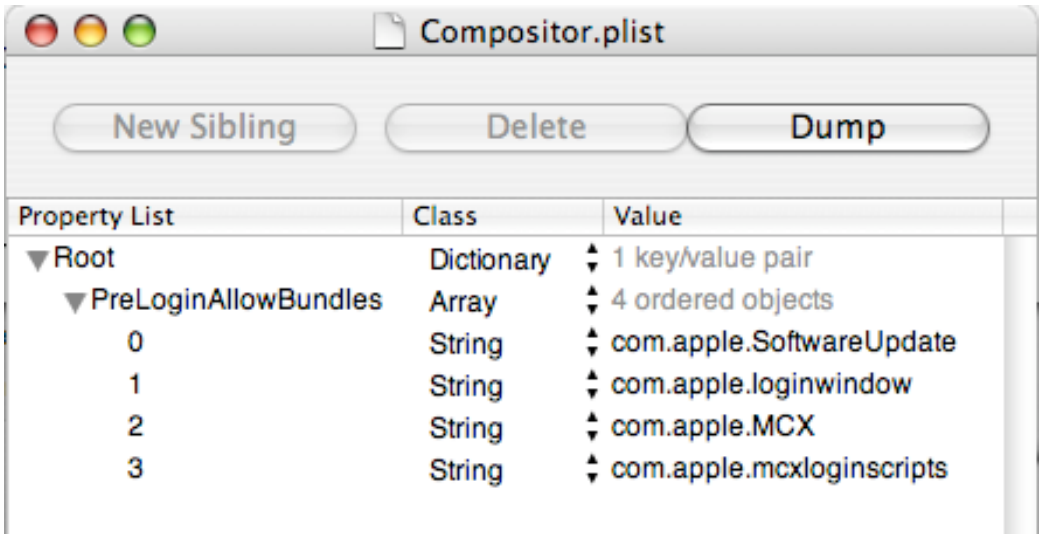
- CinchDefaults.plist: The master list of items used in the Login/Logout and Background (periodic) synchronization. This is the list that needs to be altered in order to activate ~/Library sync'ing for 10.4.2. See the Sync section for details on editing this.

- Compositor.plist: This contains a list of items designated as the 'pre-login' items, that is, the items that are allowed to be managed prior to a user login. By default, the items in the list are: com.apple.loginwindow, com.apple.MCX, and com.apple.mcxloginscripts. In order to get the new Software Update server process to work while the client is at the login screen, you need to add 'com.apple.SoftwareUpdate' to the list. I use Property List Editor and add it as a 'New Child' to the PreLoginAllowBundles' set. A picture is on the next page.

- MCXCacher: This is the executable that forces the mcx cache refresh, as well as having the ability to flush the cache completely. Sending the command using ARD to totally flush the cache is possible; but MCXCacher actually runs on its own.



The pieces of mcxd -inside /System/Library/CoreServices/mcxd.app



Compositor.plist with software update added

- MCXCompositor: This executable runs during login to build the master preferences list used by the system to present the user with a unified set of preferences.

The rest of the items are important; but not to be discussed in this paper.

d. Tips for setting up managed Preferences

The process of setting up managed preferences needs to follow a very logical progression. The sequence needs to run as follows:

- Set up the computer list(s) to be used.

- Create and set up and workgroups to be used.

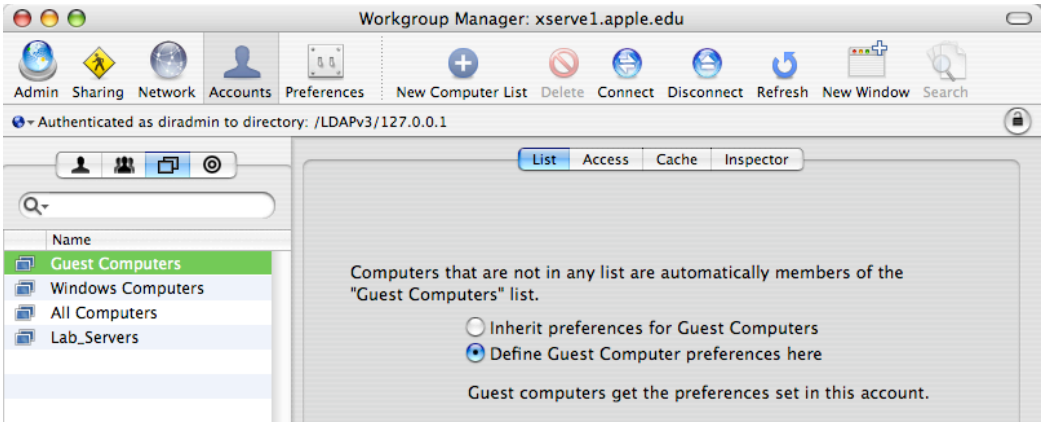
- Set up and users that will be allowed exceptions to the above settings.

- Test, test, test. Then test again.

Let's go through the various portions of the managed client settings in Workgroup Manager and point out key items to be careful with.

i. Computer account setup

There are two basic setups for computer accounts. Use the 'Guest Computers' or create a dedicated list of systems. For setting up 'Guest' you need to activate the managed preferences by turning on 'Define...' within the Accounts pane:

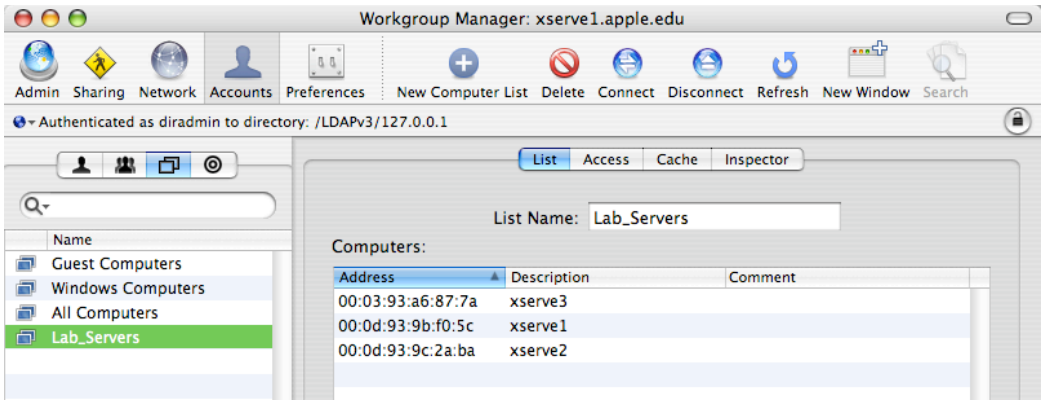


Setting 'Define...' allows all non-specific, bound clients to be managed

This management setting will only apply to computers bound to this server for management/authentication. If there is another server upstream providing authentication, then this server would be the second one in the directory list.

A specific computer list would be created to establish unique settings for a sub-set of 'all' or "guest' computers. One consideration is that if you use only the 'Guest' setting, then if someone accidentally binds their computer to your network, or you use DHCP binding, then that system is subject to your management settings from then on out - and they cache the settings to apply when they go away. (And that's a bad thing?....)

Setting up a computer list account is relatively easy. In Workgroup Manager, you create a new computer account, then establish the preference settings and other settings the same as for Guest computers. Note that when you add computers to the list, the UI for this task still asks for 'servers' - it's kidding, any client system can be added.



A unique list of computers for different management settings

One possible use for this setup would be to isolate your servers so they don't have all of the management settings the clients get by default when you use 'Guest' for them. Another possibility is to separate the desktops from the portables for differing preference settings.

Next, you set the access level for who gets to use the client, and any options they might get:

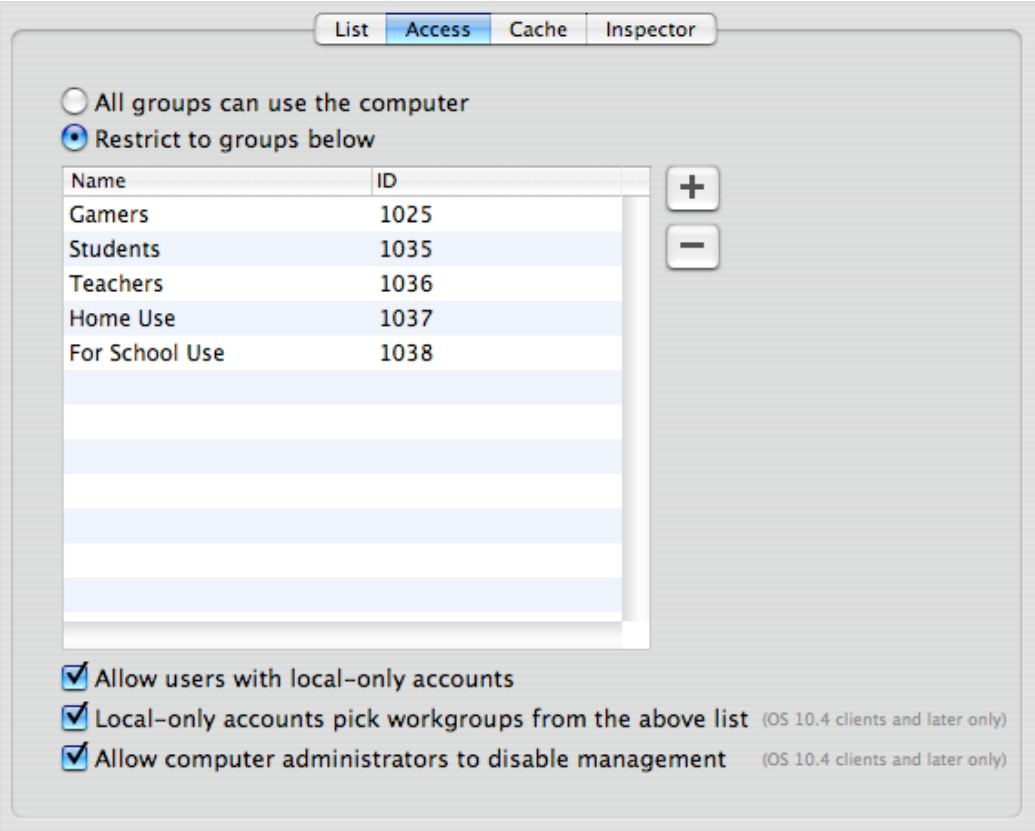
- All groups can use the computer: Allows any user to log in and will display all available workgroups for that user.

- Restrict to groups below: Sets the limits for which users can log in based on group membership. It also determines what workgroups will show up in the workgroup picker for that user. Note that you can choose access by groups, and do not need to select workgroups. This means that if you have created a group called 'Students' but have no management settings for them, then they can log into this computer; but will have only computer and user level management settings.

- Allow users with local-only accounts: This lets you isolate computers from use by any local accounts. If this is checked and the following box is not, then local users will have access to all workgroups.

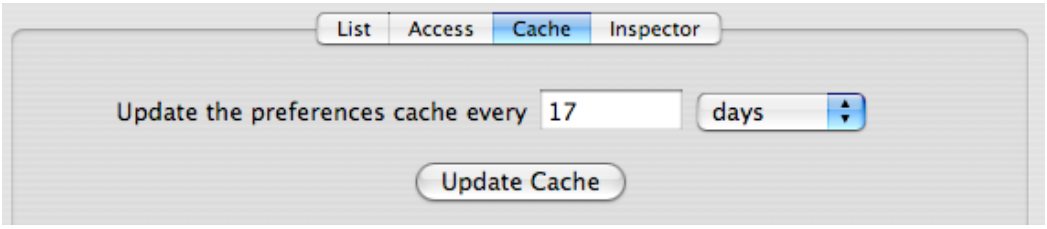
- Local-only accounts pick workgroups from the above list: For Tiger clients, checking this will allow local users to pick workgroups from the list (or from all workgroups if the that radio button is selected). Panther local clients always pick workgroups from the above list.

- Allow computer administrators to disable management: This setting is there to allow the local administrator to choose to disable management for their login, or not. If you have to run applications that require local admin users, this checkbox allows you still provide management for those users.



Access settings for Computer accounts

The next setting is to establish the 'Last Ditch' cache setting, and to keep your clients from becoming unmanaged when disconnected from the network.



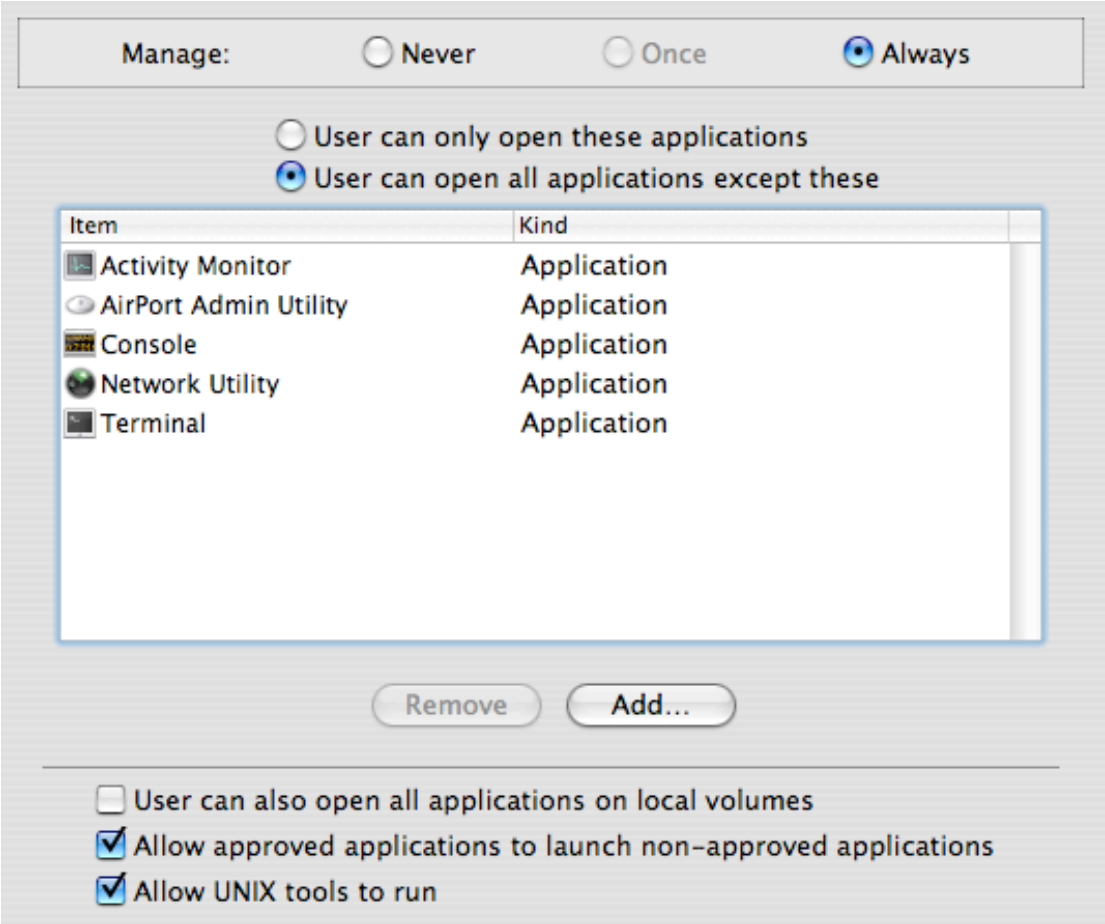
Last resort cache setting

This setting is now default at 17 days. It is designed to force a cache refresh when absolutely nothing else has happened to make the MCXCacher run. That includes no startups, reboots, logouts, network location changes, or other acts of nature, including a lack of any changes to directory records and preferences. However, if this is set too low, it creates lots of network traffic. Set it to -0- and the caching mechanism shuts off and all users become unmanaged when they disconnect from the network. That's because it allows the local mcx system to talk directly with the network directory all the time, and never creates the mcx_cache records. Great for testing, lousy for deployment.

ii. The Preference settings

Now let's tour the individual preference settings and look at some settings tips.

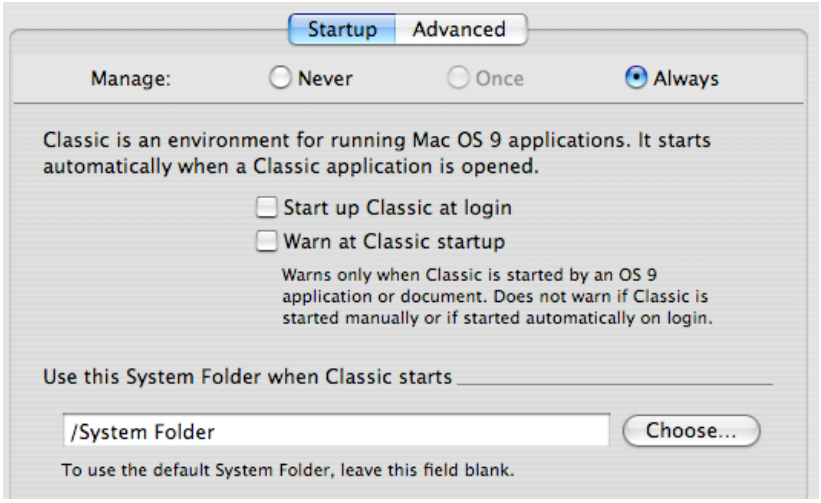
a. Applications



Applications Preference pane

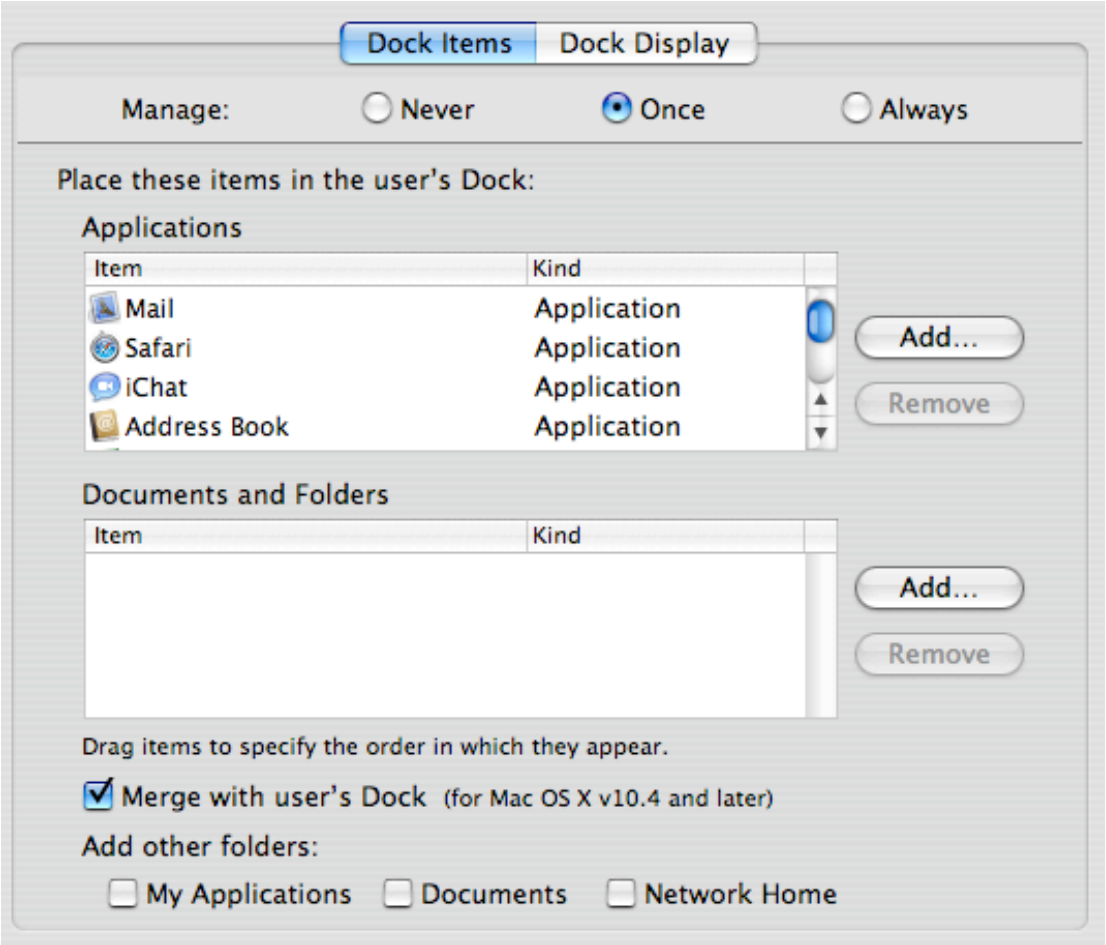
You can set applications for approval, or for denial. Also, turning on the '...open all applications on local volumes' defeats the purpose for managing apps to begin with.

b. Classic



Set the boot System Folder to avoid rogue OS 9 systems

c. Dock



Dock Items preference pane

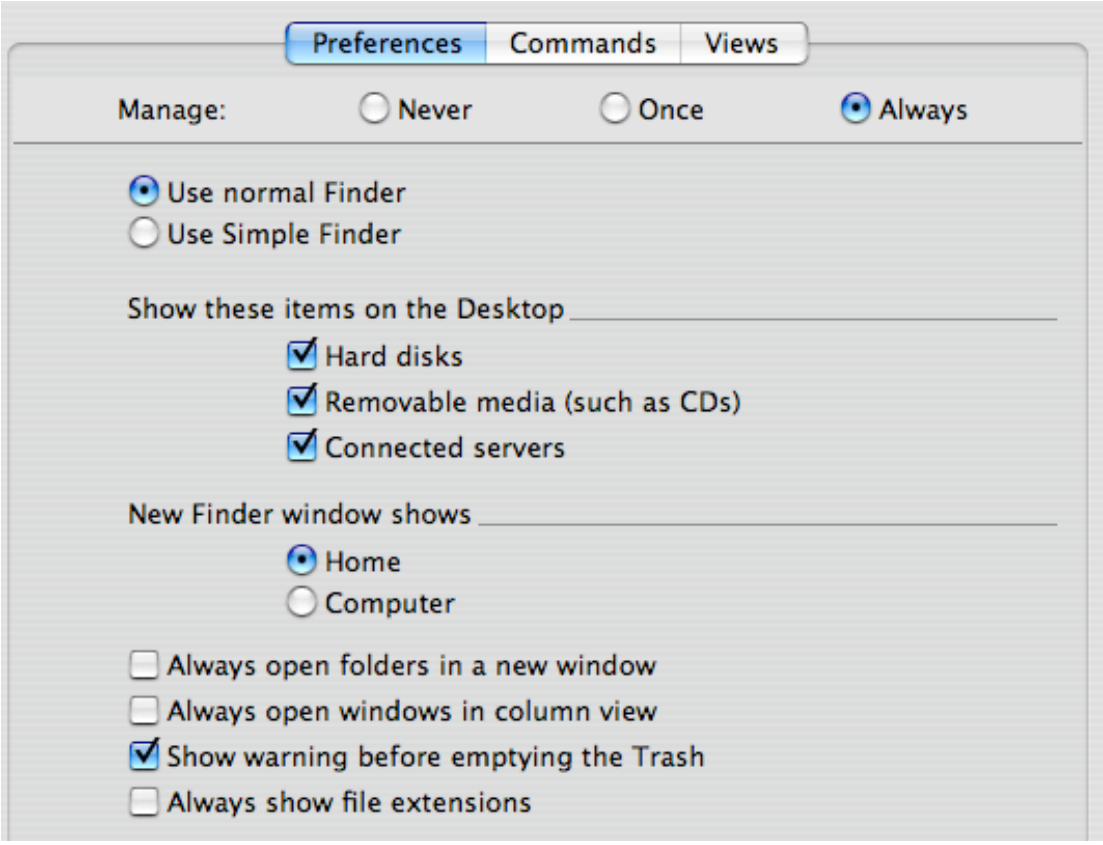
Using 'Once' lets you set the Dock to look like it did in training. After that, the user's get to modify as they see fit. You might not add the 'Network Home' if this is a workgroup that will be used offline, or if you are using Portable Home Directories. The Dock display settings can be used to differentiate workgroups by placing the Dock in one location for one workgroup, and a totally different location for the other.

d. Energy Saver

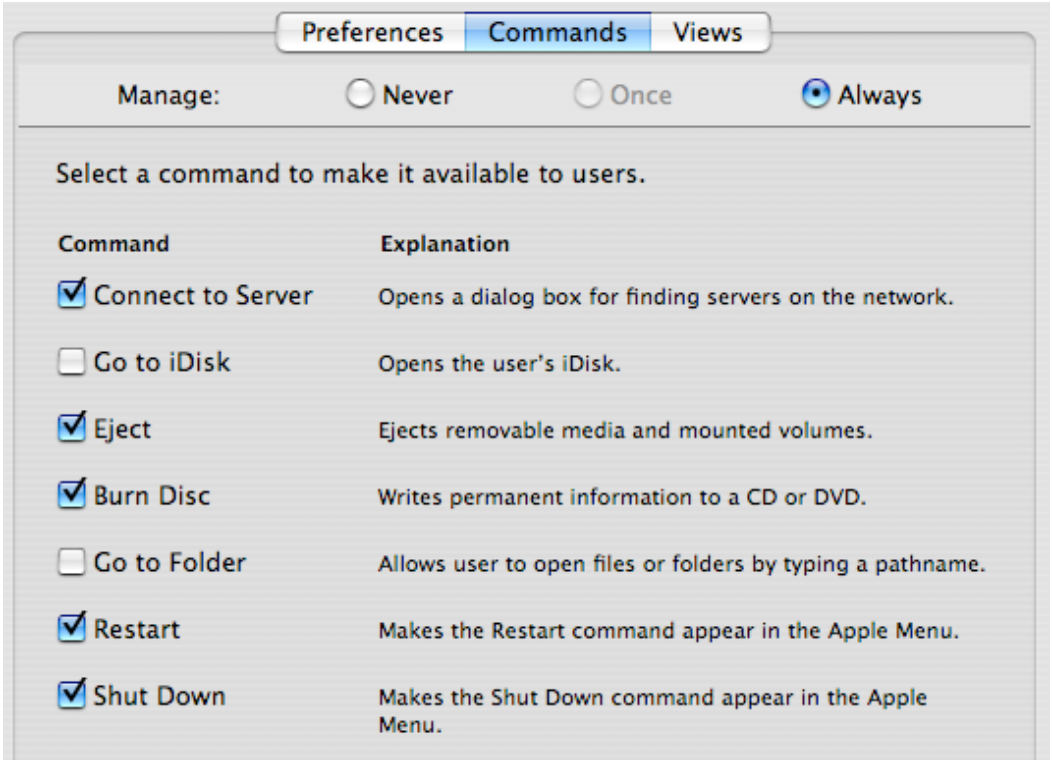
Energy Saver contains a lot of myriad settings. Most importantly, you can specify the settings for power consumption for desktops versus portables. You can also set the various times of sleep for system, hard disk, and screen. Recommendation - set the system sleep to never on shared systems so that you can always locate and access them with ARD. Also, note that the Startup/Shutdown settings do not apply when the system is at the Login Window.

e. Finder

Setting the Finder is often ignored; but can help users tremendously. You can specify the view settings for your systems in order to avoid the default 'Chaos' view (no alignment, ordering, or spacing). Sure would be nice if you could set the Finder to always open to list or icon view.

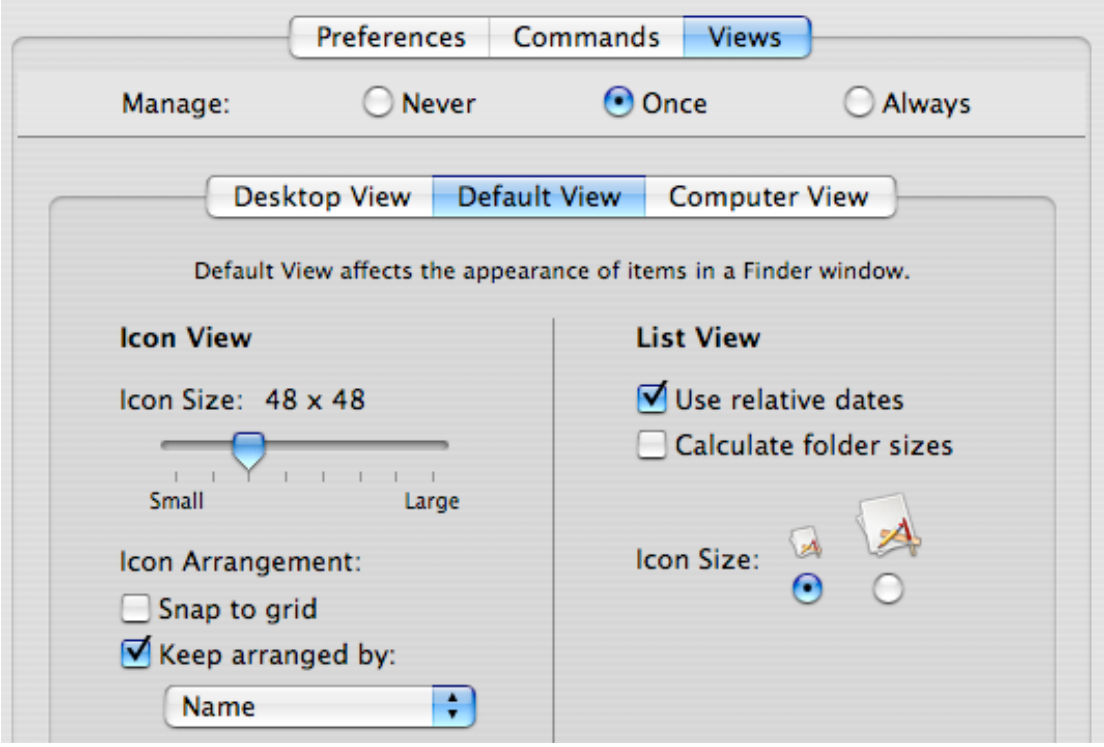


Finder preference pane - part 1



Finder preference pane - part deux

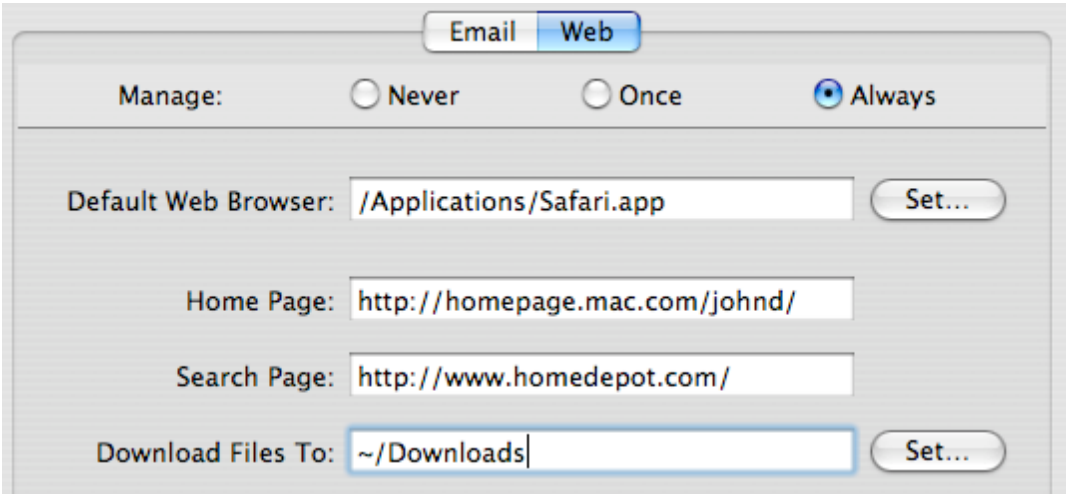
A note about Simple Finder: It's not even close to what Panels was under MacMgr, so don't try to make it act like that. Simple Finder (or Finder for Kids - FFK) does not support mounting network volumes, creating new folders, deleting anything, or basically working as any educator would need it to. Since the escalations about it not performing as needed (versus performing as designed) have dropped to nothing for the past year or so, don't count on this changing either.



Finder preference pane - part III

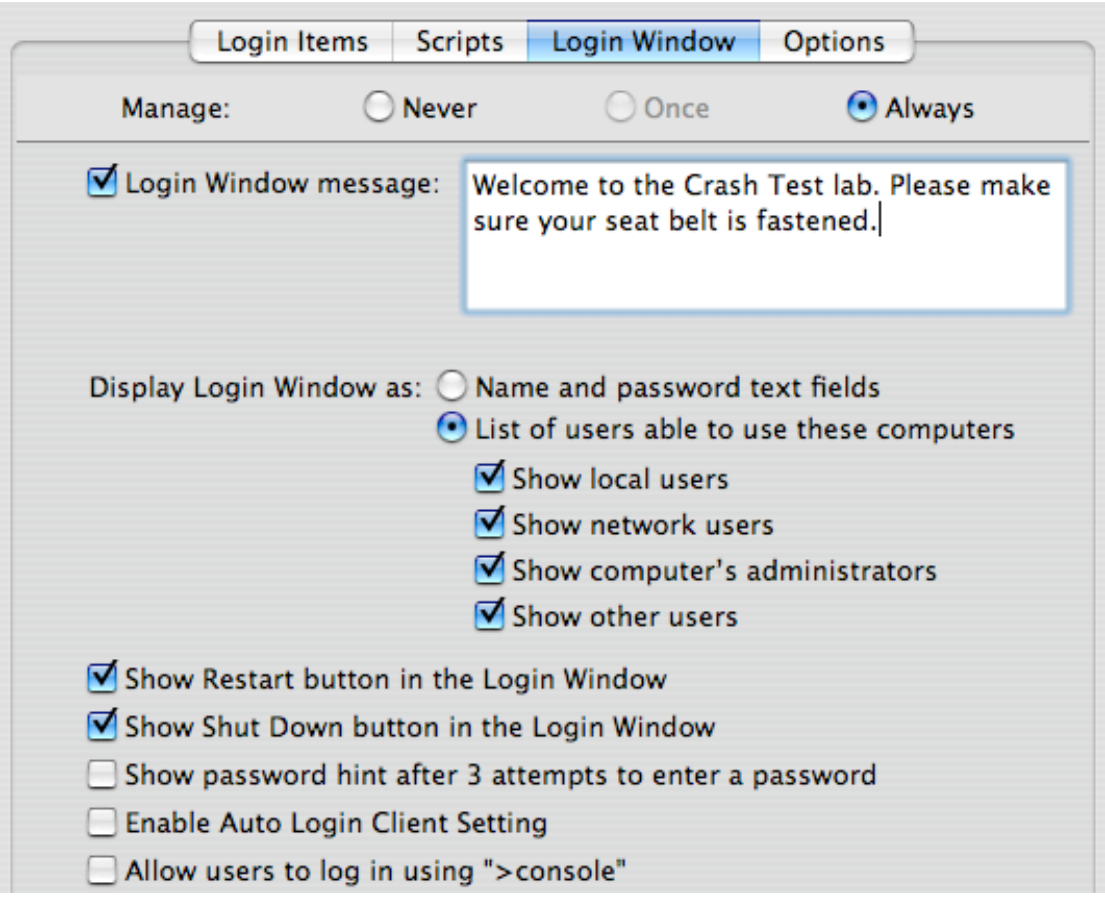
f. Internet

Use this preference pane to set the default download location for user files, default web browser, and mail server settings. This preference allows the use of '~/...' to point to a location inside the user's home directory.



g. Login

The Login preference pane allows two levels of control. First, you set the items for the actual Login Window appearance as a user would see their client system. Second, you establish the login settings for items to be activated or mounted at login.



Login window settings used at Computer account level only

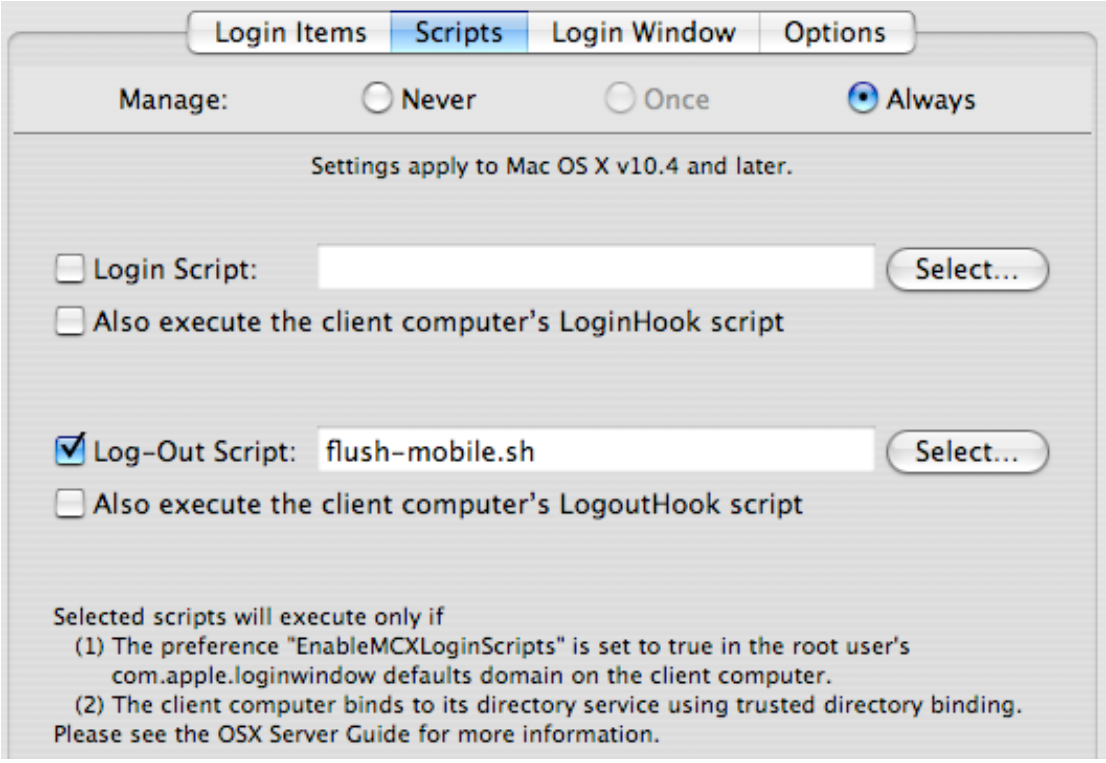
Tiger managed client now supports scripts in the login/logout process. When these scripts are enabled, they are actually stored inside the directory record for the computer list account. The client(s) that run these scripts may be bound to the managed client server by authenticated bind (see the Directory Access application help). This mechanism also allows the administrator to lock out the possibility of someone running a rogue script on a system.

The workstations must have their com.apple.loginwindow preferences edited to include the key 'EnableMCXLoginScripts' set to true. From the 'User Management' guide, pg. 169 -

Set the key "EnableMCXLoginScripts" in ~root/Library/Preferences/ com.apple.loginwidow.plist to TRUE.

```
$ sudo defaults write com.apple.loginwindow.plist EnableMCXLoginScripts -
bool TRUE
```

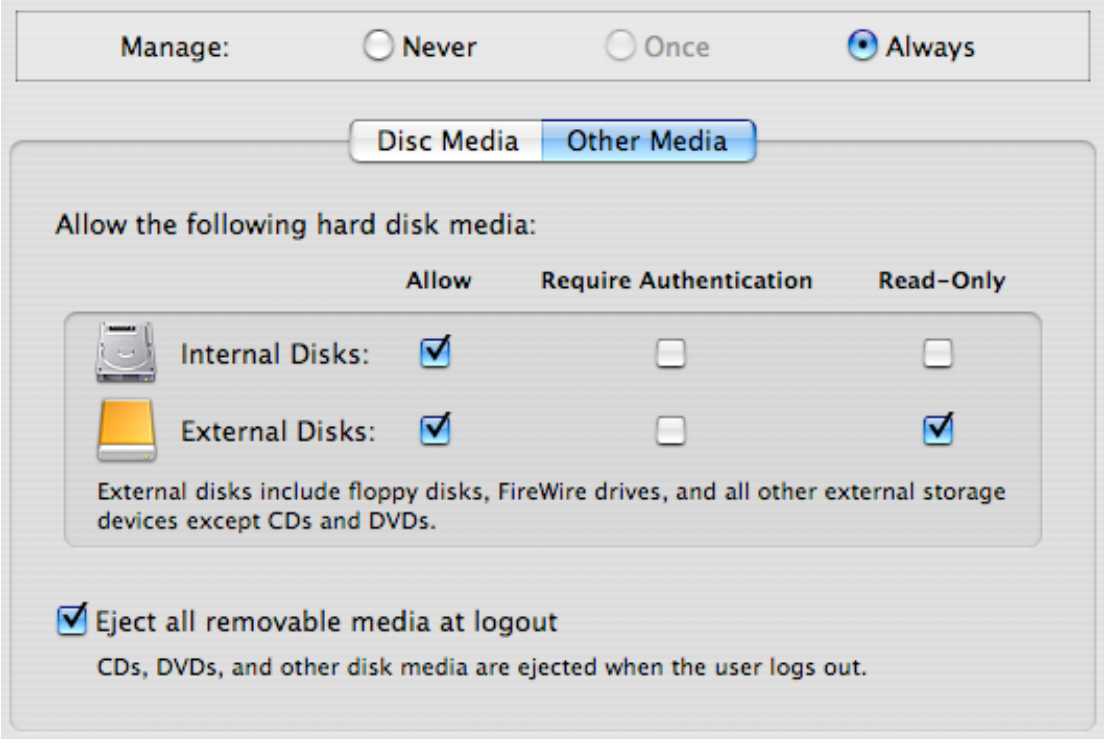
This can be done using ARD or individually. MCX settings cannot be used to force this value.



Enabling script actions as part of the mcx settings

h. Media Access

This preference pane allows you to control access to removable media, as well as read-write media.



i. Mobility

This preference pane gets its own entire section in this paper.

j. Network

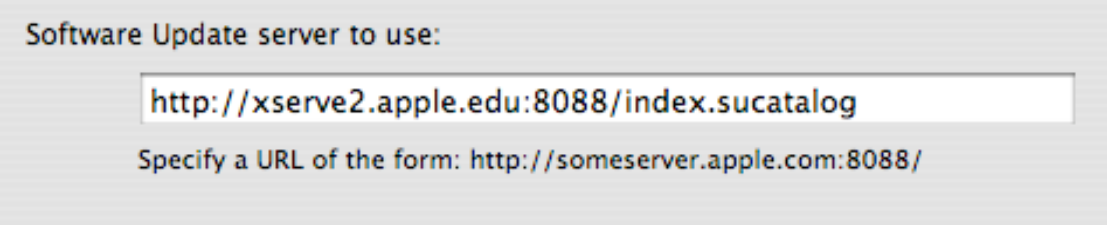
Use this preference pane to set up global proxy settings for sets of computers.

k. Printing

This preference pane allows you to set the specific printers available to a computer list, workgroup, or user.

l. Software Update

One of the newest settings in Workgroup Manager is the ability to set access to the local software update server. The preference sets the catalog used by the software update process to look up available updates. Currently, the setting is only active when a user is logged in, unless you alter a portion of the 'mcxd' application itself to include the software update property list being active at the login window. See the section on preference management in detail for more info.



Note that the tooltip is not complete. You need the 'index.sucatalog' ending.

m. System Preferences

Use this preference setting to determine the specific System preferences that your users should have access to. Items restricted here are also unavailable through any other means, such as menubars.

n. Universal Access

This preference pane allows you to set up the various user enhancement options, such as screen flashing when an alert is made, zoom, mouse controls, and more.

e. Working with the Preference Editor

New with Tiger is the ability to go behind the normal Preference pane and add, edit, and create unique management settings. The Preference Editor is located under the 'Details' pane. This tool allows you to work with any property list on the system, as well as a new concept known as 'preference manifests' which will covered in the next section.

Setting up a preference that is not included in the main Preferences pane is a matter of tracking down the preference file, adding it to the list, then editing the file to include only the items that are key to your deployment. There are some key points to understand before using the Editor.

- The application may not respond to the preferences you set. Some apps only look in certain places, or do not follow the approved 'CFPreferences' model. (Bad Apps™)

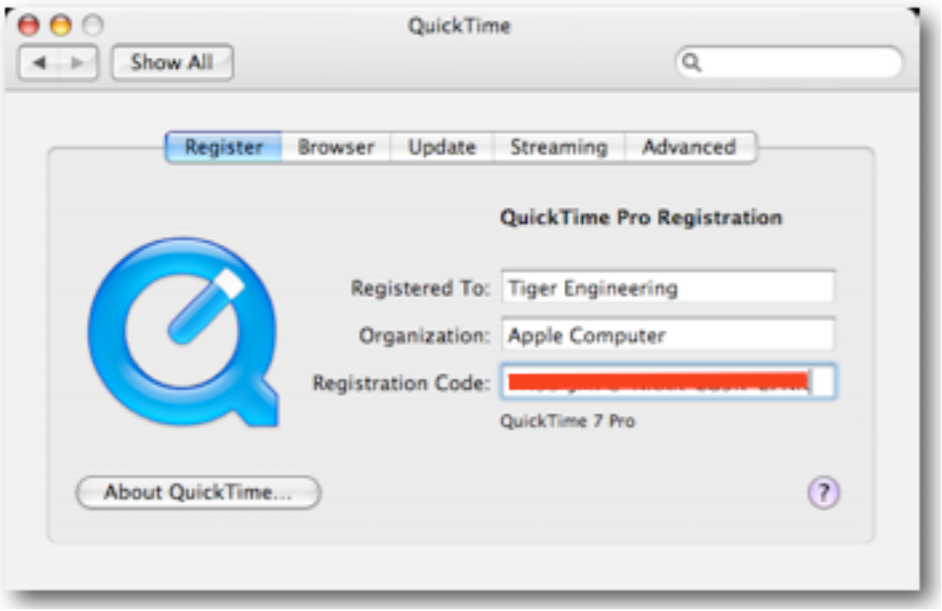
- Some of the settings may not apply across managed users. Be careful to remove any suspicious direct path names, such as /mac-hd/Users/skippy/Desktop.

Let's go through an example, then I'll clue you in to several tricks I have learned while working with the editor.

i. Example: Setting up a Quicktime preference

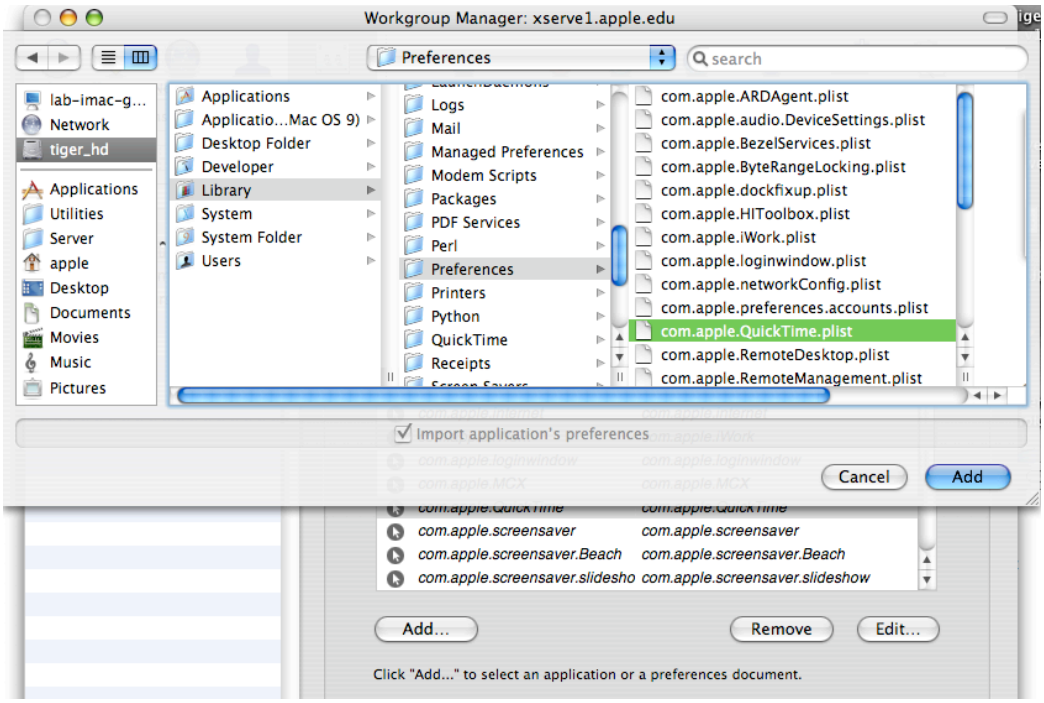
One problem I saw at several sites was that the network admin had bought a license for QTPro for all clients; but couldn't apply it to all workstations. Under QTv7, the license (finally!) works when applied at root level of the system to all logged in users. Using the Preference Editor, you can set the QTPro license, and other settings, to apply to your clients - and best of all, the end user can't see it, even if they have access to the QT preference.

First, you need to go into the QT system preference on your admin/master system and set the license and other preferences. Save the changes.



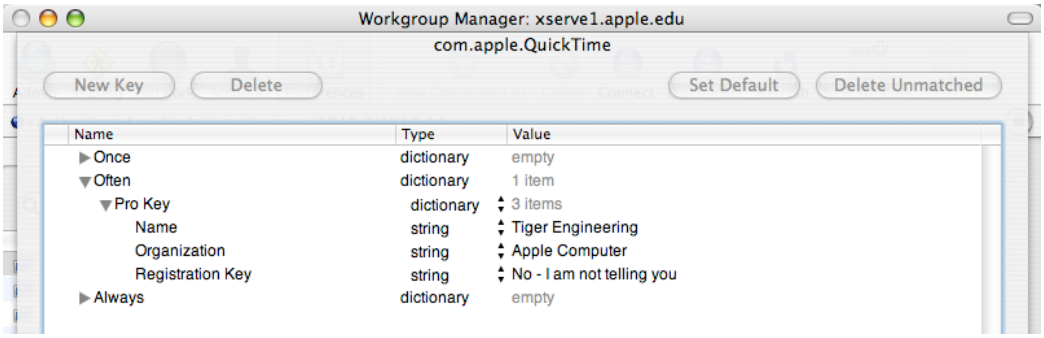
Don't you wish....

In Workgroup Manager, go to the appropriate computer account, select Preferences, then Details. Click on the 'Add' button at the bottom of the window and locate the 'com.apple.Quicktime.plist' inside the /Library/Preferences folder (not your own ~/Library/Preferences folder - use the one for the computer).



Quicktime preference added to Editor

The preference when opened shows up under the new listing of ‘Often’ in the Editor window.



Quicktime preference details

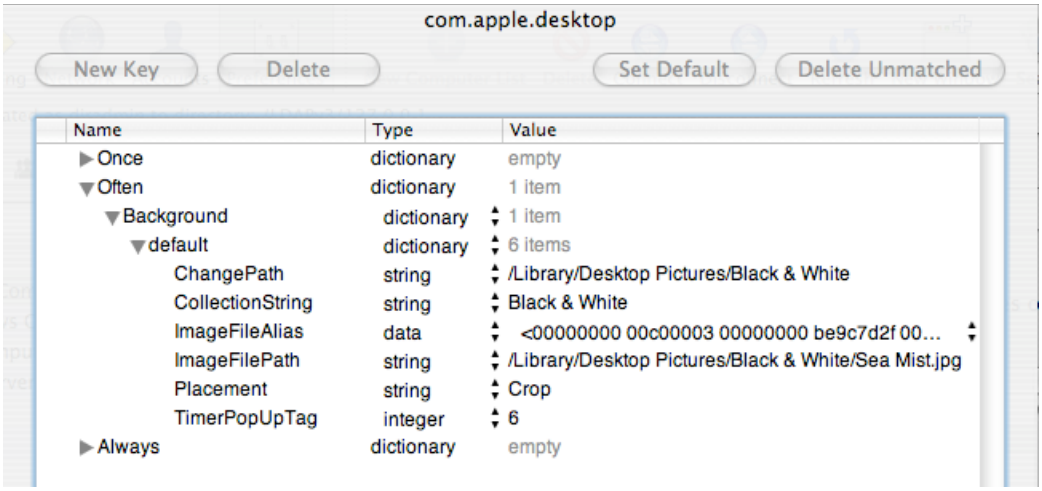
Remember - ‘Often’ establishes the setting as editable by the user while they are logged in (if you allow access to whatever the preference is); but reverts to the admin-defined preference at next login. I think of it as a ‘perception of freedom’ for the user.

ii. Other interesting preference tricks

The strength of the Preference Editor is being able to set preferences that will affect lots of users; but don’t exist as specific settings in the regular Preference pane. Here are some tips for setting some of the more interesting items:

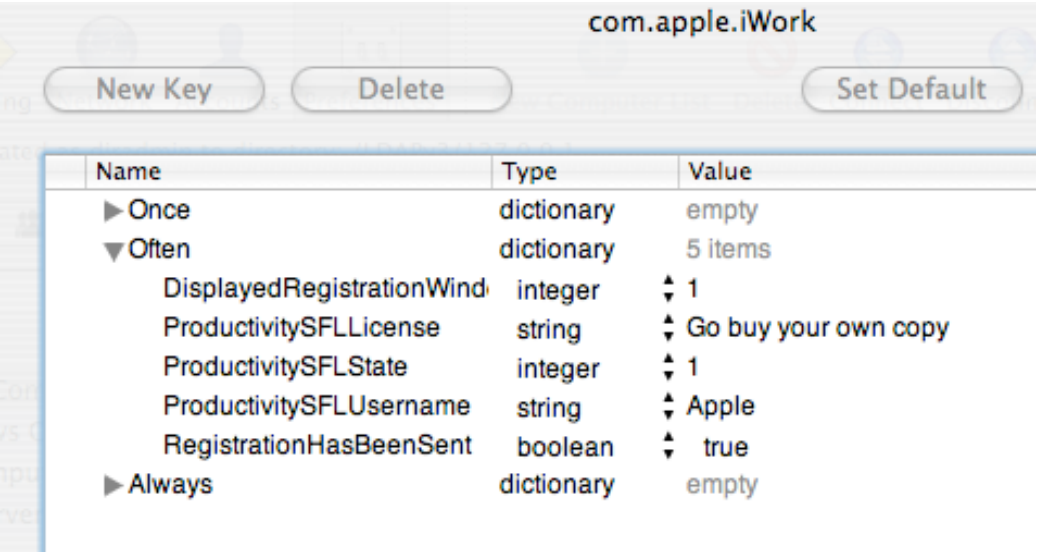
- Desktop: Using portions of the com.apple.desktop plist, you can set a Desktop picture that will be established at every login. So, even if the user changes the Desktop from within iPhoto, it’ll be reset to this picture. Just pick the one you like for yourself, then grab your

own com.apple.desktop.plist from ~/Library/Preferences. The picture must reside on their workstation. You can use ARD to blast a set of 'allowed' pictures to all your systems.



They can change it - but at next login...

- iWork: Comes with a requirement to register at first launch. You can get around this for your site (if you own the proper # of licenses) by capturing the com.apple.iWork plist after you have set up a system and entered the serial number info.



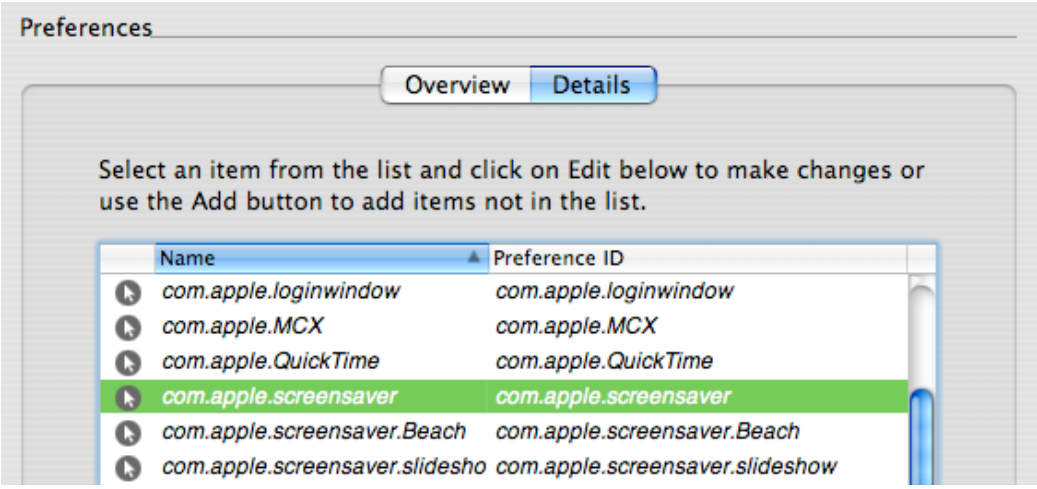
Use your site license serial number to preset all your clients

- Screensaver: You can preset a screensaver for your users by capturing the values you use on the admin/master system. You'll need to make sure you grab all three sets of files:

- com.apple.screensaver
- com.apple.screensaver.<saver type>
- com.apple.screensaver.<saver motion>

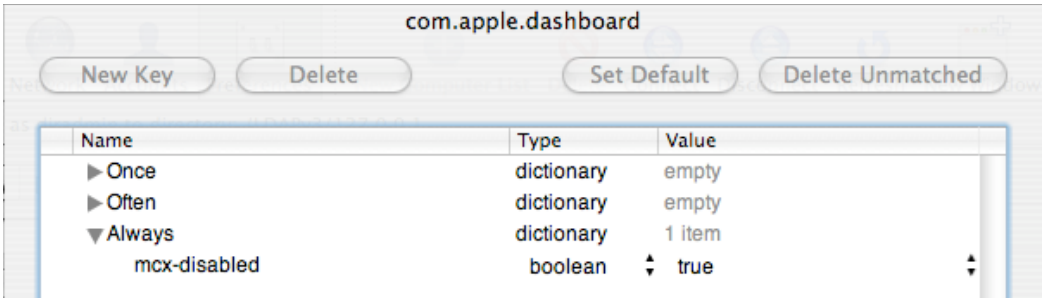
The files, as you may have noticed, exist inside the /ByHost folder and have your system's MAC address attached to the end of the filename. Option-drag these files out to the Desktop, remove the MAC address from the end of the filename, and then add those items to the Preference Editor. Once they are added to the 'Often' section, you can cut/paste them over to the 'Always' section.

By getting all of these, you can establish the screensaver for all logged in clients. What I haven't tried is working out the pre-login setting to make this work (if it will) with the Login Window.



Grab all three parts of the screensaver setup

- Dashboard: Some people have a requirement, desire, or dictate, to turn off Dashboard support on their network. Until we have an item-by-item control of Widgets, the best that can be done is to disable Dashboard entirely for managed users. This can be done through the Preference Editor by creating a new preference in the editor and adding a single key. In the Editor, click on the 'Add' at the bottom of the window and locate your com.apple.dashboard preference inside your own Preference folder. With that preference highlighted, click on 'Edit...'. Delete all portions of the preference, and click on 'Always' then click on 'New Key'. Enter 'mcx-disabled' as the key name, make its type 'boolean' and set the value to 'true.'



Dashboard disabled for all managed clients

Obviously, just grabbing property lists from applications and plopping them into the Preference Editor won't always work. For example, the Bluetooth settings just refuse to play.

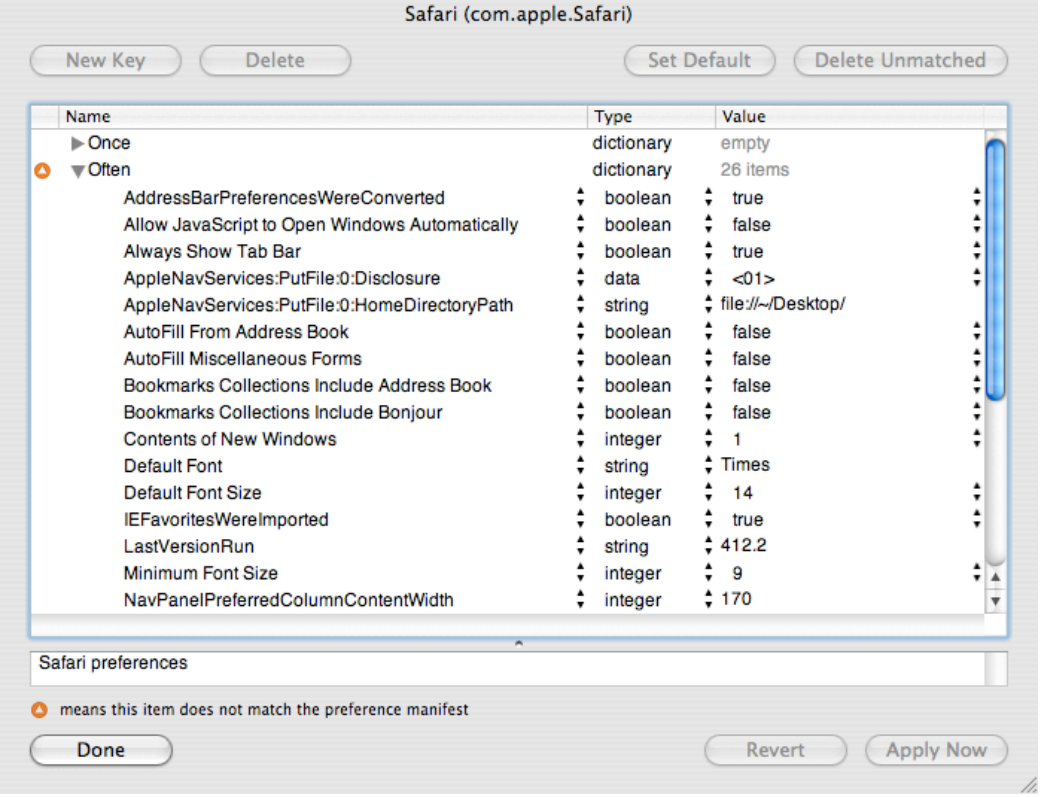
(Shame on them...) At the same time, settings for applications such as KidPix3 Deluxe from MacKiev work really well. Test , test, test.

f. Preference Manifests

Another addition to Tiger management is the idea of Preference Manifests. This is a new capability that actually involves the application developer writing a database within their application that defines the managed items, or 'keys', for insertion into the Preference Editor. There is a huge amount of detail on this downloadable from the Developer web site - [Preference Manifest Files for Managed Clients Overview]

Instead of loading an entire plist of items for an application only to find that one or two of the keys work for your network, you load the preference manifest. The developer did the work for you already by defining the items within their application that actually work for managed users. Here's an example using Safari.

- Safari's preference manifest contains only the keys that can be set for multiple users:



Note the non-hard pathnames and more readable settings

The manifest is added by selecting the application itself and adding it to the Editor. If you also select the checkbox at the bottom of the window to add all of the preferences, you'll get both the manifest and plist. Many of the items in Safari's standard plist are machine specific, hence not items you would want to add normally to a managed set.

g. Portable Home Directories and synchronization

The biggest concern over Portable Home Directories (PHDs) in Tiger is being able to set them up correctly, and the issue of being able to sync everything - meaning the user's Library folder as well as their documents. The idea is that a computer should become ubiquitous for that user. If they drop their iBook, then handing over a replacement system and the user logging in, seeing their own work and settings, is the goal.

Some of the testing that has gone on so far has been over the use of Login/Logout sync versus Background sync. The sync process is essentially the same as used for .Mac's iDisk syncing, and the speed of the sync operation is entirely dependent on the amount of changed material. This is significantly different from the old preference management model employed with Macintosh Manager. In that model, the entire contents of the Preference folder could be copied to and from the workstation at login/logout. Under Tiger, the sync operation will copy only changed files. This means that if you alter only one document out of hundreds on your system when logged in, then only that document will need to be synchronized at logout, or in the background.

i. Why and when to turn on ~/Library sync

~/Library syncing has been the biggest issue with Tiger's PHD implementation so far. While it is possible to require your users place items into a 'safe folder' that has been established for automatic sync, it doesn't cover for some of the more essential items that may need to be kept in tow. These are:

- iPhoto libraries. While the Pictures folder can be sync'd, the preferences that define the actual photo libraries and albums are stored inside the user's Preferences folder.

- iTunes. Same goes here. Music is in the Music folder, playlist definitions and settings are in the Library folder.

- Safari. Bookmarks are in the Library folder.

- Mail. The Apple mail application stores everything inside the user's Library folder. You might use IMAP to, sort of, get around this - but most portable users download a large portion of their email to their local system.

That said (and those are just the Apple-specific cases), there are also reasons why the ~/Library sync wasn't turned on up to this point. Many applications write hard-coded paths into their preferences still - even after years of being told not to. Most applications still store caches inside the user's Library folder - despite the screams of frustration from customers who know that this can crater a wireless network. I guess this comes from developers who never test on anything other than local accounts on really fast computers....

There are also folders that can be sync'd, such as ~/Documents, that may contain items you don't want to sync. This could include the ~/Documents/Appleworks User Data folder which contains a lot of extra files that have no purpose in being sync'd. Or large portions of the ~/Documents/Microsoft User Data folder for the same reason.

Finally, when would you turn on ~/Library sync? Since the process needs to have the user's preferences sync'd when they begin working, then ~/Library sync needs to be

activated when the user logs in. That was the way it was designed, and to imagine trying to use backround/periodic sync for the preferences is an exercise in futility.

That said, let's venture into the realm of experimental science and turn on ~/Library syncing for clients using 10.4.2.

ii. Digging up the parts to turn on ~/Library sync

In order to make this work, we have to make changes to the 'mcxd.app' and then deploy that changed application to all of our clients. The components of the 'mcxd.app' that need to change are 'CinchDefaults.plist' and the 'Compositor.plist.' The key is the 'CinchDefaults.plist' since it contains all of the settings that determine what items are allowed to be added to the Workgroup Manager windows, as well as setting the default items that make sense. The 'Compositor.plist' is included only so we can activate the ability to remotely perform software updates while the client systems are at the login window. **Note - I had to dig up lots of interesting things to put this together; but make sure you try this in a limited test environment first!!! This is not supported by Apple, Applecare, me, or Tenshi.**

a. Step 1 - Login as 'root' and create a copy

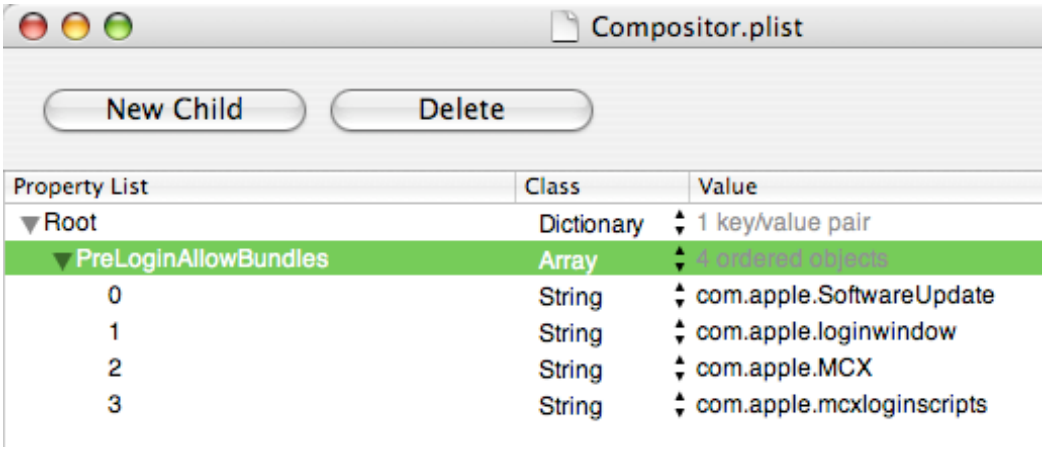
I shouldn't have to say this; but create a copy of the **/System/Library/CoreServices/mcxd** application to work on. I placed mine on the Desktop of the admin/master I was using to make all of the managed client settings. Logging in as 'root' keeps you from having to mess with permissions later on. (Yes, the smarter people will be doing all of this in Terminal. I use Property List Editor - it comes with the Server Admin install and with Developer Tools.)

b. Step 2 - Open the app copy bundle

Ok, this one is simple; but yes, you need to open the application bundle to get to the parts.

c. Step 3 - Edit the Compositor.plist

The file is located inside the /Resources folder. Click on the 'Root' property, then the 'PreLoginAllowBundles'. Click the 'New Child' button, and enter a new string titled 'com.apple.SoftwareUpdate'. Adding this new value allows the sysadmin to send softwareupdate commands to

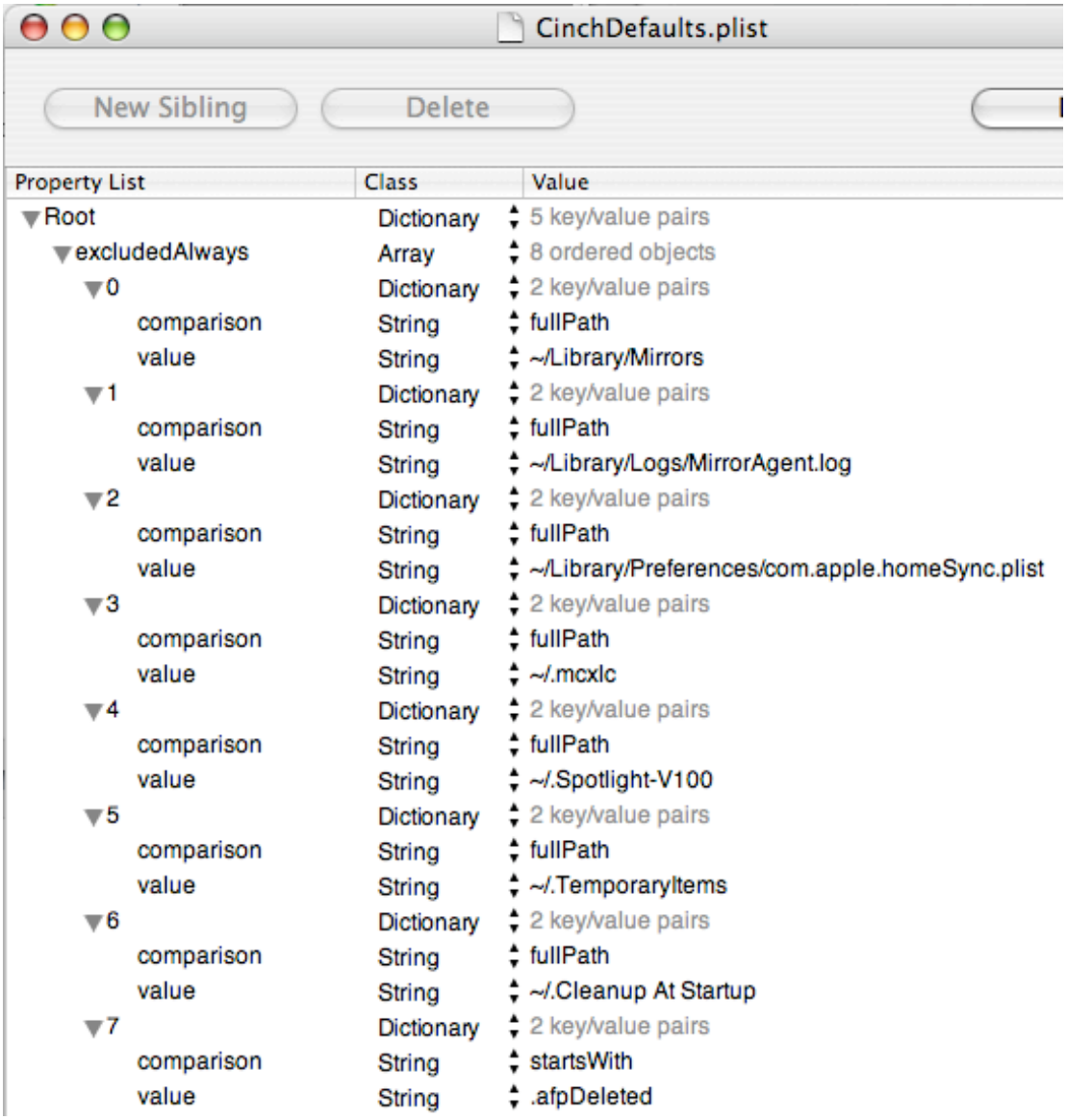


d. Step 4 - Edit the CinchDefaults.plist

This property list contains several components. Each fulfills a mission during either Login/Logout sync, background sync, or globally. (A text version of the edited files is included at the end of this paper.)

- excludedAlways: The list of items 'you can't touch' - these are the items that should be left alone because syncing them may cause damage to the user's settings, rendering them unable to log in or work. It includes such goodies as '~/Library/Mirrors' which contains the mirror files that determine what gets sync'd. Ever stand between two mirrors and see how they tend to look into infinity? Syncing this folder would be a Bad Idea™

The files that should be in this list are as follows (you can use Property List editor to make sure you have these included:



- excludedItems: These are the items that show up by default in the ‘Skip items...’ list for background/periodic sync. They are the ~/Library folder and the Trash can folder. You can leave this list alone, since we do not want to sync either of these while we are logged in. It’d be sort of like changing a tire at 75 mph.

- excludedPrefItems: These are the items that are in the ‘Skip items...’ window for Login/Logout sync. This is where we will do most of our editing. You can also add any items that you know to be bad for traffic and/or performance on your own. I suggest you keep this list tight, and add any you are not sure of to a preset user account in WGM later on.

Property List	Class	Value
▼Root	Dictionary	5 key/value pairs
▶excludedAlways	Array	8 ordered objects
▶excludedItems	Array	2 ordered objects
▼excludedPrefItems	Array	12 ordered objects
▼0	Dictionary	2 key/value pairs
comparison	String	fullPath
value	String	~/Library/Application Support/SyncServices
▼1	Dictionary	2 key/value pairs
comparison	String	fullPath
value	String	~/Library/Caches
▼2	Dictionary	2 key/value pairs
comparison	String	fullPath
value	String	~/Library/Logs
▼3	Dictionary	2 key/value pairs
comparison	String	fullPath
value	String	~/Library/Preferences/ByHost
▼4	Dictionary	2 key/value pairs
comparison	String	fullPath
value	String	~/Library/Printers
▼5	Dictionary	2 key/value pairs
comparison	String	fullPath
value	String	~/Library/Safari/Icons
▼6	Dictionary	2 key/value pairs
comparison	String	startsWith
value	String	IMAP-
▼7	Dictionary	2 key/value pairs
comparison	String	startsWith
value	String	Mac-
▼8	Dictionary	2 key/value pairs
comparison	String	fullPath
value	String	~/Library/Preferences/com.apple.dock.plist
▼9	Dictionary	2 key/value pairs
comparison	String	fullPath
value	String	~/Library/Preferences/com.apple.iChatAgent.plist
▼10	Dictionary	2 key/value pairs
comparison	String	fullPath
value	String	~/Library/Preferences/com.apple.sidebarlists.plist
▼11	Dictionary	2 key/value pairs
comparison	String	fullPath
value	String	~/Library/Preferences/loginwindow.plist

ExcludedPrefItems in CinchDefault.plist

- syncedFolders: The item(s) to be synced by default during a background sync. This is currently set to only '~' which is everything (minus the exceptions noted in the 'excludedItems' above.

- syncedPrefFolders: Currently contains nothing by default. Since Login/Logout sync is up to you, you might want to add items that you always want to sync then. I leave it blank and add the items inside WGM.

e. Step 5 - Replace old mcxd.app and deploy

Once you have made these changes, you can then use ARD to copy the modified mcxd application to all of your workstations and/or place it into the master image(s) you will deploy. Test first! Note that the file permissions for the mcxd.app are:

```
drwxr-xr-x    3 root  wheel     102 Jul 13 13:55 mcxd.app
```

h. Dealing with the mobile account 'rabbit effect'

One other issue to deal with when deploying mobile accounts is the 'rabbit effect' or the problem of having lots of local home directories left all over your workstations by users who are no longer around. If you have only shared systems, such as iBook carts, then you may have users logging in as mobile accounts and leaving their home directories behind. Over the space of a week or so, this may result in dozens of orphaned home directories on every machine.

One possible solution is to assign specific users to specific shared systems, so that you keep down the proliferation of home directories. Another option is to borrow a script from a really smart person who figured out a way to query the workstation for accounts that are mobile, then remove that account and delete their home directory on that computer. While this script could be used as a logout script, I caution using it too liberally, or too often. My suggestion is to run this as a scheduled task under ARD once a week, or once every so often, to cleanse the shared systems. This would make a good logout script for kiosk systems. Be careful running this without control - imagine nuking all of the newly deployed 1:1 systems by mistake.

Here's the script (thanks to Steve Hayman):

```
#!/bin/sh

# nonnull.sh currentUser

# remove accounts of mobile account users, other than the currently-
logging-in-user ($1)

# and anybody who's logged in

PATH=/bin:/sbin:/usr/bin:/usr/sbin export PATH

users=$( nireport . /users name original_home | grep -v NoValue | sed -e
's/[     ,].*$//' | sed -e '/^$/d' )

for user in $users; do

    # Skip $1 - which is the user logging in (assuming this is a login
hook)

    # and skip anybody else who seems to be logged in

    if [[ "$user" != "$1" ]]; then

        if ! who | grep -w "^$user" >/dev/null; then

            echo Removing home folder of $user

            eval hd="~$user"
```

```
            # sanity check!  users with home folders in /Users only
            if [[ "$hd" != /Users/* ]]; then
                rm -r "$hd"
                echo Removing $user from netinfo
                niutil -destroy . /users/$user
            fi
        fi
    fi
done
# Dirty the mcx cache so that the login list shows recently deleted
accounts
/System/Library/CoreServices/mcxd.app/Contents/Resources/MCXCacher -d
exit 0
--------
```

Once again, as with all of the alterations -and even the approved settings - make sure you test each change on systems that can be rebuilt. Like the Vice Principal's powerbook. :-)

7. Workflow management - getting in deep (coming in v12.2)

a. Enhanced group folders

b. ACL's - the new permissions

c. SACL's for server admins

8. Apple Remote Desktop tips

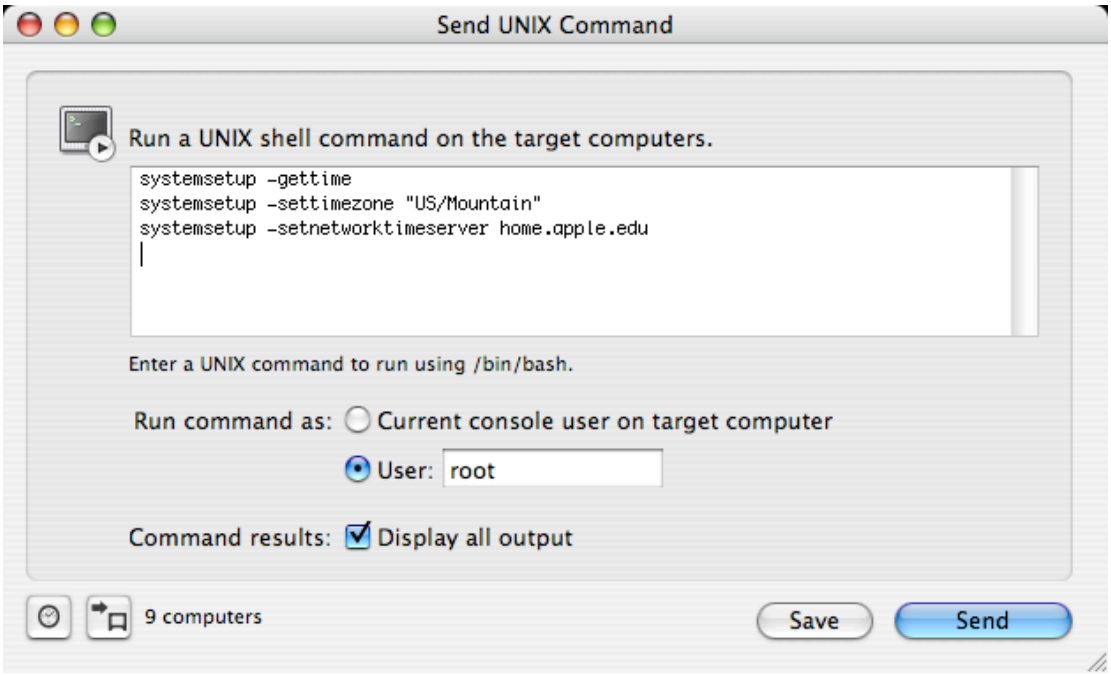
Very little of this paper, or my network/client management could have taken place without the use of Apple Remote Desktop version 2.2. This section is here to show you just a few of the things that ARD can do to make managing lots of systems a much easier task.

a. Mass system changes - 'Send UNIX command'

ARD supports a task called 'Send UNIX command' which allows the administrator the ability to interfere on a grand scale with the clients and servers on his network. For example, after I created the user accounts and specified the home directories, I sent a command to all of the servers at once - 'CreateHomeDir -a' which kicks in the home directory creation process. Doing this across several servers simultaneously is a dream.

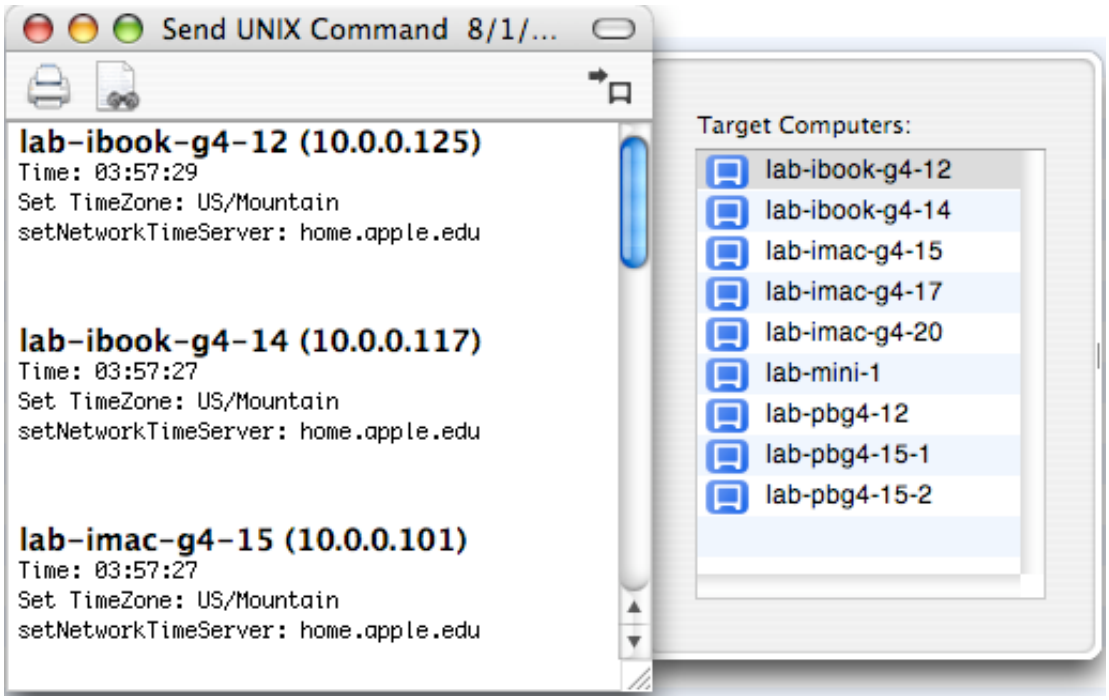
The same idea goes for the 'CreateGroupFolder' command. It can be sent to all servers at once instead of waiting for the periodic maintenance session in the middle of the night.

Finally, sending commands to workstations for troubleshooting helps a lot. Besides the typical ones like 'top', there is an entire suite of commands built into ARD for system and network maintenance. They are the 'systemsetup' and 'networksetup' commands. These commands allow the sysadmin the ability to view or edit a huge amount of data across all connected workstations. For example - are you sure all of your clocks are set the same, that all workstations have the same timezone settings, and that they are all using the same time server? Using ARD, you can check all of this at once across all of your clients.



Setting and checking times makes sure all of the services work well

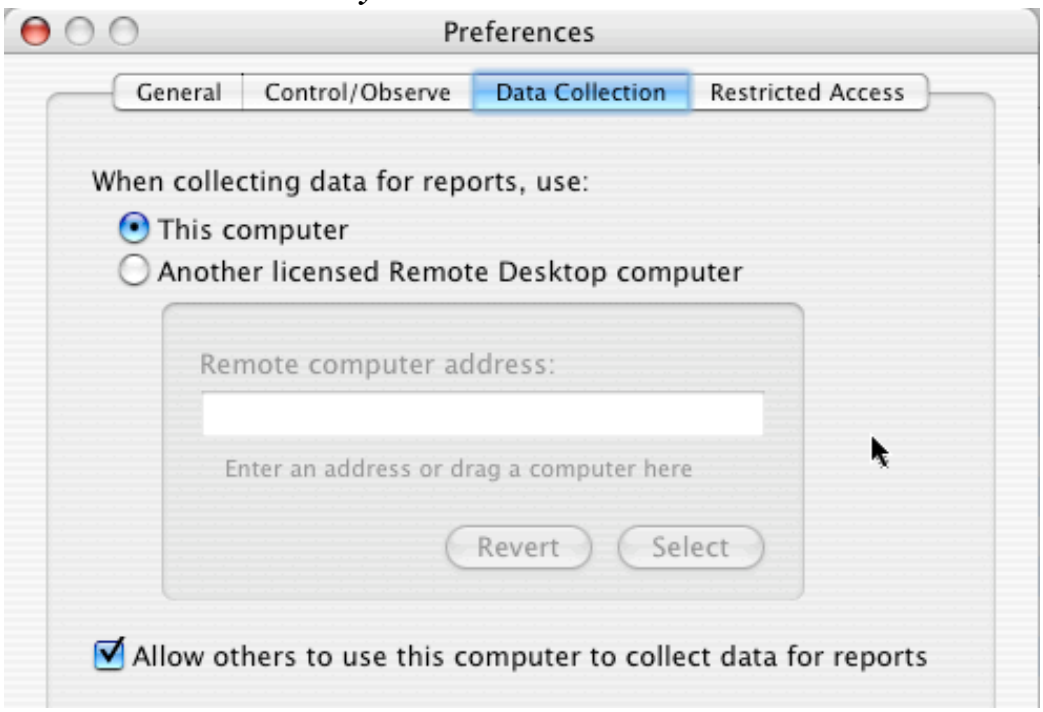
The end result of sending this is a response from all workstations together in a report that can be scanned, saved, or repeated.



b. Reports - a plethora of information

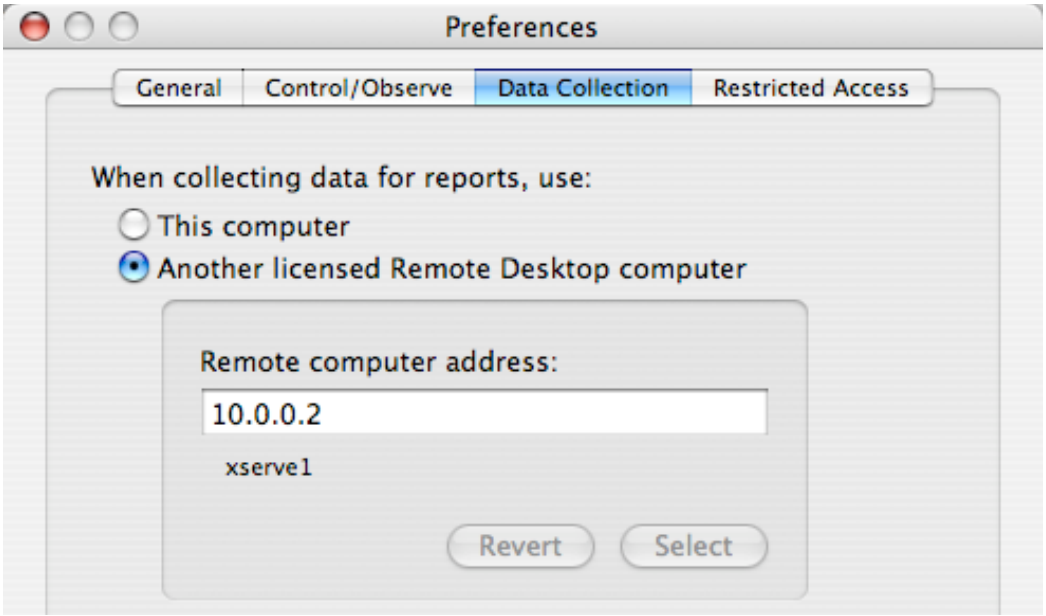
ARD supports remote reports. Clients can be set to send their information to a remote collection station that has ARD installed. This frees up the admin workstation from having to stick around looking for all of the client systems. Here's how it is set up, and how it looks in action:

i. Install ARD on remote system and set as remote collector.



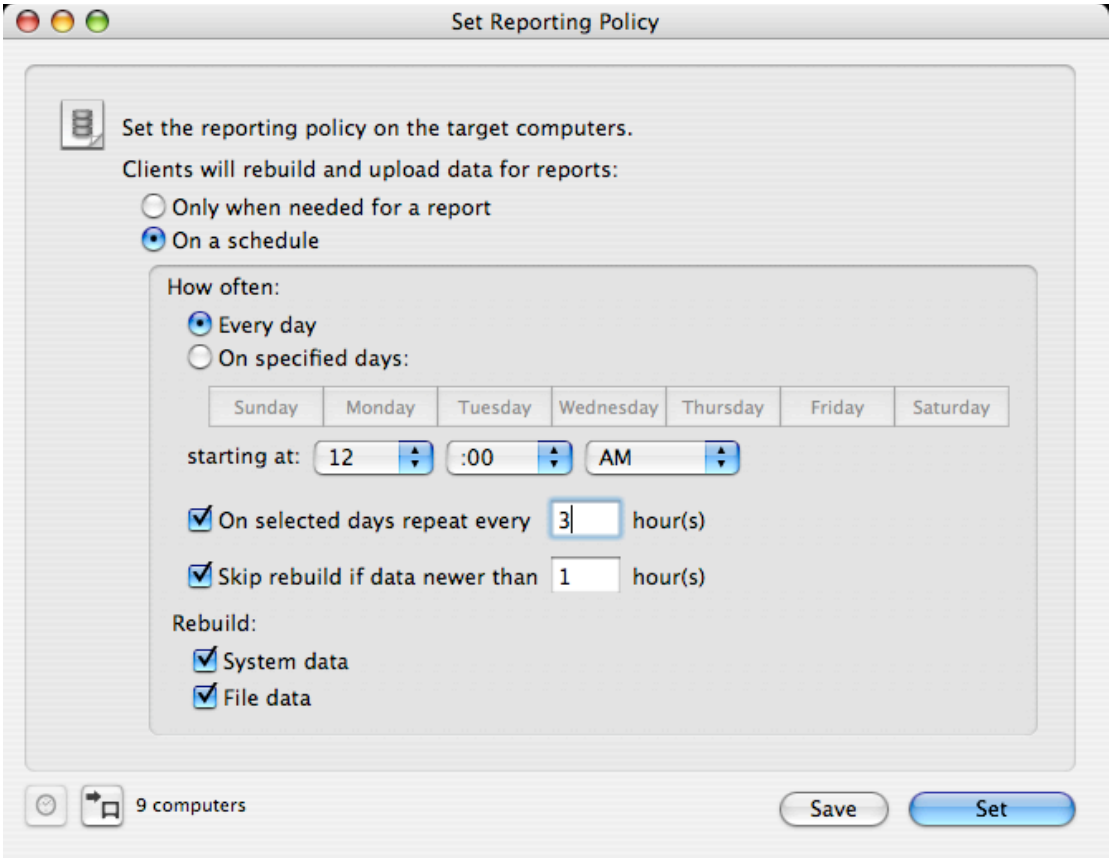
Remote system set to allow others to store report data

ii. Set admin system to use remote collection system



Admin/master is set to use remote system as data collection site

iii. Set clients to report on a schedule

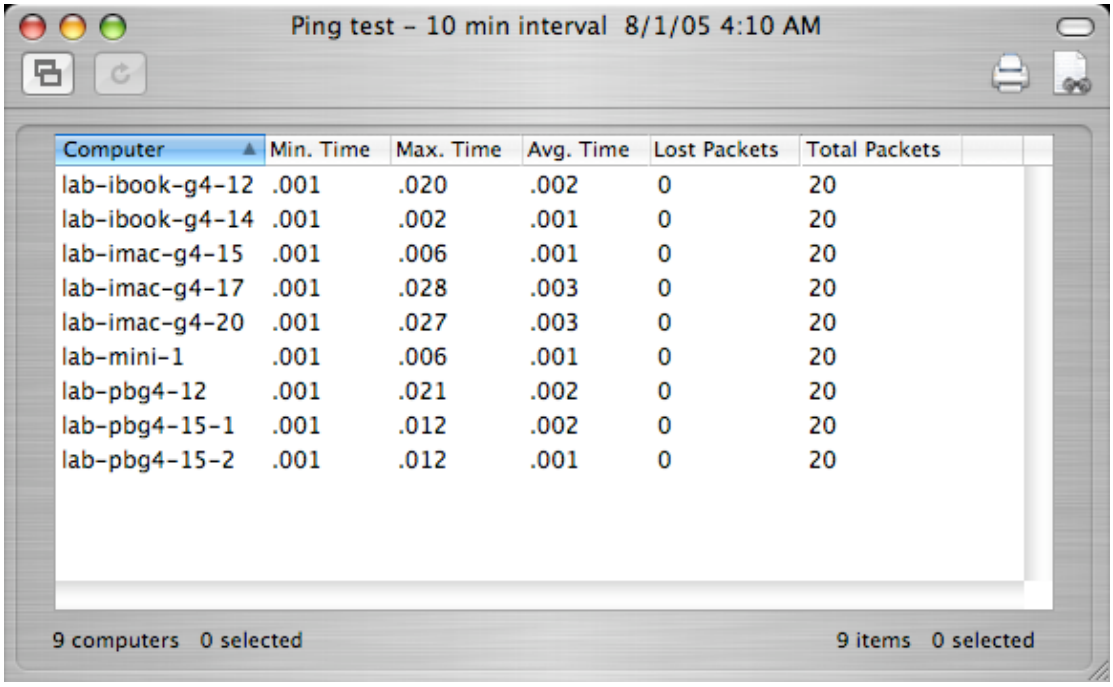


Clients will generate report data and send it to the designated collection system

At this point, when the admin needs a report on any of the datapoints collected by ARD, it will be able to get the information even if the clients are offline. As long as the client has reported at least once to the collection system, its information is available to the administrator.

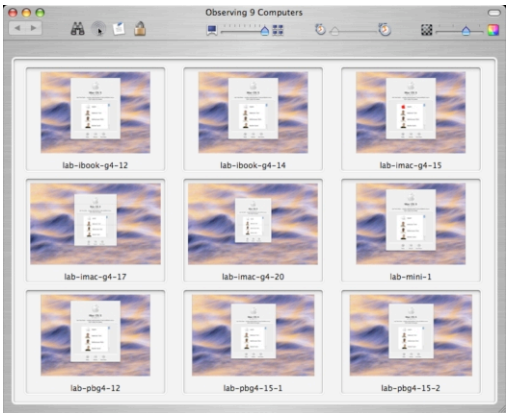
c. Troubleshooting using ARD

Another key feature of ARD is the ability to troubleshot systems for performance or reliability. For example, you can send packets to all of your systems using the 'Network Performance' test and check the network for noticeable interference. Here's an example of a quick test - the iMacs are on the 100bT wired switch, the iBooks and Powerbooks are on an Airport Extreme set to support b/g:



Simple network performance test

These are only a few of the capabilities of ARD, without even going into the screen sharing, control, observer, lock screen, ...



9. Wrap up

There are probably a hundred pages of material that didn't make it into this version of the Tips and Tricks. The next version, actually the next update, will contain a walkthrough of the entire process of setting up a managed client environment from start to finish. I also need to finish up the detailed section on advanced workflow.

The key to using the managed client controls is testing and evaluating the needs of the end users. There are many steps in client management, and making sure that the goal "Support the end users" stay at the top of your pile.

Hopefully, you enjoy this paper, and let me know if you'd like to see more.

JohnD

Summer, 2005

Appendix A. Reference files

Documentation from Apple on Tiger server: Located at http://www.apple.com/server/documentation/

Mac OS X Server Documentation

- Getting Started
- Upgrading and Migrating
- Migrating to Mac OS X Server from Windows NT
- User Management
- File Services Administration
- Print Service Administration
- System Imaging and Software Update Administration
- Mail Service Administration
- Web Technologies Administration
- Collaboration Services Administration
- Network Services Administration
- Open Directory Administration
- QuickTime Streaming Server 5.5 Administration
- Windows Services Administration
- Command-Line Administration
- High Availability Administration
- Xgrid Administration
- Mac OS X Server Worksheet
- Mac OS X Server Glossary

A1. systemsetup commands for ARD

```
systemsetup Help Information
------------------------------------
Usage: systemsetup -getdate
     Display current date.

Usage: systemsetup -setdate <mm:dd:yy>
     Set current date to <mm:dd:yy>.

Usage: systemsetup -gettime
     Display current time.

Usage: systemsetup -settime <hh:mm:ss>
     Set current time to <hh:mm:ss>.

Usage: systemsetup -gettimezone
     Display current time zone.

Usage: systemsetup -settimezone <timezone>
     Set current time zone to <timezone>. Use "-listtimezones" to list time
zones.

Usage: systemsetup -listtimezones
     List time zones supported by this machine.

Usage: systemsetup -getusingnetworktime
     Display whether network time is on or off.

Usage: systemsetup -setusingnetworktime <on off>
     Set using network time to either <on> or <off>.

Usage: systemsetup -getnetworktimeserver
     Display network time server.

Usage: systemsetup -setnetworktimeserver <timeserver>
     Set network time server to <timeserver>.

Usage: systemsetup -getsleep
     Display amount of idle time until computer, display and hard disk
sleep.

Usage: systemsetup -setsleep <minutes>
     Set amount of idle time until computer, display and hard disk sleep to
<minutes>.
     Specify "Never" or "Off" for never.
```

```
Usage: systemsetup -getcomputersleep
     Display amount of idle time until computer sleeps.

Usage: systemsetup -setcomputersleep <minutes>
     Set amount of idle time until compputer sleeps to <minutes>.
     Specify "Never" or "Off" for never.

Usage: systemsetup -getdisplaysleep
     Display amount of idle time until display sleeps.

Usage: systemsetup -setdisplaysleep <minutes>
     Set amount of idle time until display sleeps to <minutes>.
     Specify "Never" or "Off" for never.

Usage: systemsetup -getharddisksleep
     Display amount of idle time until hard disk sleeps.

Usage: systemsetup -setharddisksleep <minutes>
     Set amount of idle time until hard disk sleeps to <minutes>.
     Specify "Never" or "Off" for never.

Usage: systemsetup -getwakeonmodem
     Display whether wake on modem is on or off.

Usage: systemsetup -setwakeonmodem <on off>
     Set wake on modem to either <on> or <off>.

Usage: systemsetup -getwakeonnetworkaccess
     Display whether wake on network access is on or off.

Usage: systemsetup -setwakeonnetworkaccess <on off>
     Set wake on network access to either <on> or <off>.

Usage: systemsetup -getrestartpowerfailure
     Display whether restart on power failure is on or off.

Usage: systemsetup -setrestartpowerfailure <on off>
     Set restart on power failure to either <on> or <off>.

Usage: systemsetup -getrestartfreeze
     Display whether restart on freeze is on or off.

Usage: systemsetup -setrestartfreeze <on off>
     Set restart on freeze to either <on> or <off>.

Usage: systemsetup -getallowpowerbuttontosleepcomputer
```

```
     Display whether the power button is able to sleep the computer.

Usage: systemsetup -setallowpowerbuttontosleepcomputer <on off>
     Enable or disable whether the power button can sleep the computer.

Usage: systemsetup -getremotelogin
     Display whether remote login is on or off.

Usage: systemsetup -setremotelogin <on off>
     Set remote login to either <on> or <off>. Use "systemsetup -f -
setremotelogin off" to suppress prompting when turning remote login off.

Usage: systemsetup -getremoteappleevents
     Display whether remote apple events are on or off.

Usage: systemsetup -setremoteappleevents <on off>
     Set remote apple events to either <on> or <off>.

Usage: systemsetup -getcomputername
     Display computer name.

Usage: systemsetup -setcomputername <computername>
     Set computer name to <computername>.

Usage: systemsetup -getlocalsubnetname
     Display local subnet name.

Usage: systemsetup -setlocalsubnetname <name>
     Set local subnet name to <name>.

Usage: systemsetup -getstartupdisk
     Display current startup disk.

Usage: systemsetup -setstartupdisk <disk>
     Set current startup disk to <disk>.

Usage: systemsetup -liststartupdisks
     List startup disks on this machine.

Usage: systemsetup -getWaitForStartupAfterPowerFailure
     Get the number of seconds after which the computer will start up after
a power failure.

Usage: systemsetup -setWaitForStartupAfterPowerFailure <seconds>
     Set the number of seconds after which the computer will start up after
a power failure. The <seconds> value must be a multiple of 30 seconds.
```

```
Usage: systemsetup -version
     Display version of systemsetup tool.

Usage: systemsetup -help
     Display help.

Usage: systemsetup -printCommands
     Display commands.
```

A2. networksetup commands for ARD

```
networksetup Help Information
------------------------------
Usage: networksetup -listnetworkserviceorder
     Display services with corresponding port and device in order they are
tried for connecting
     to a network. An asterisk (*) denotes that a service is disabled.

Usage: networksetup -listallnetworkservices
     Display list of services. An asterisk (*) denotes that a network
service is disabled.

Usage: networksetup -listallhardwareports
     Display list of hardware ports with corresponding device name and
ethernet address.

Usage: networksetup -detectnewhardware
     Detect new network hardware and create a default network service on
the hardware.

Usage: networksetup -getmacaddress <hardwareport or device name>
     Display ethernet (or AirPort) address for hardwareport or device
specified.

Usage: networksetup -getcomputername
     Display the computer name.

Usage: networksetup -setcomputername <name>
     Set the computer's name (if valid) to <name>.

Usage: networksetup -getinfo <networkservice>
     Display IPv4 address, IPv6 address, subnet mask,
     router address, ethernet address for <networkservice>.

Usage: networksetup -setmanual <networkservice> <ip> <subnet> <router>
     Set the <networkservice> TCP/IP configuration to manual with IP
address set to ip,
     Subnet Mask set to subnet, and Router address set to router.

Usage: networksetup -setdhcp <networkservice> [clientid]
     Set the <networkservice> TCP/IP configuration to DHCP. You can set the
     DHCP client id to the optional [clientid]. Specify "Empty" for
[clientid]
     to clear the DHCP client id.
```

```
Usage: networksetup -setbootp <networkservice>
     Set the <networkservice> TCP/IP configuration to BOOTP.

Usage: networksetup -setmanualwithdhcprouter <networkservice> <ip>
     Set the <networkservice> TCP/IP configuration to manual with DHCP
router with IP address set
     to ip.

Usage: networksetup -setv4off <networkservice>
     Turn IPv4 off on <networkservice>.

Usage: networksetup -setv6off <networkservice>
     Turn IPv6 off on <networkservice>.

Usage: networksetup -setv6automatic <networkservice>
     Set the service to get its IPv6 info automatically.

Usage: networksetup -setv6manual <networkservice> <address> <prefixlength>
<router>
     Set the service to get its IPv6 info manually.
     Specify <address> <prefixLength> and <router>.

Usage: networksetup -getdnsservers <networkservice>
     Display DNS info for <networkservice>.

Usage: networksetup -setdnsservers <networkservice> <dns1> [dns2] [...]
     Set the <networkservice> DNS servers to <dns1> [dns2] [...]. Any
number of dns servers can be
     specified. Specify "Empty" for <dns1> to clear all DSN entries.

Usage: networksetup -getsearchdomains <networkservice>
     Display Domain Name info for <networkservice>.

Usage: networksetup -setsearchdomains <networkservice> <domain1> [domain2]
[...]
     Set the <networkservice> Domain Name servers to <domain1> [domain2]
[...]. Any number of Domain Name
     servers can be specified. Specify "Empty" for <domain1> to clear all
Domain Name entries.

Usage: networksetup -create6to4service <newnetworkservicename>
     Create a 6 to 4 service with name <newnetworkservicename>.

Usage: networksetup -set6to4automatic <networkservice>
     Set the service to get its 6 to 4 info automatically.

Usage: networksetup -set6to4manual <networkservice> <relayaddress>
```

```
     Set the service to get its 6 to 4 info manually.
     Specify <relayaddress> for the relay address.

Usage: networksetup -getftpproxy <networkservice>
     Display FTP proxy (server, port, enabled value) info for
<networkservice>.

Usage: networksetup -setftpproxy <networkservice> <domain> <port number>
<authenticated> <username> <password>
     Set FTP proxy for <networkservice> with <domain> and <port number>.
Turns proxy on. Optionally, specify <on> or <off> for <authenticated> to
enable and disable authenticated proxy support. Specify <username> and
<password> if you turn authenticated proxy support on.

Usage: networksetup -setftpproxystate <networkservice> <on off>
     Set FTP proxy to  either <on> or <off>.

Usage: networksetup -getwebproxy <networkservice>
     Display Web proxy (server, port, enabled value) info for
<networkservice>.

Usage: networksetup -setwebproxy <networkservice> <domain> <port number>
<authenticated> <username> <password>
     Set Web proxy for <networkservice> with <domain> and <port number>.
Turns proxy on. Optionally, specify <on> or <off> for <authenticated> to
enable and disable authenticated proxy support. Specify <username> and
<password> if you turn authenticated proxy support on.

Usage: networksetup -setwebproxystate <networkservice> <on off>
     Set Web proxy to  either <on> or <off>.

Usage: networksetup -getsecurewebproxy <networkservice>
     Display Secure Web proxy (server, port, enabled value) info for
<networkservice>.

Usage: networksetup -setsecurewebproxy <networkservice> <domain> <port
number> <authenticated> <username> <password>
     Set Secure Web proxy for <networkservice> with <domain> and <port
number>. Turns proxy on. Optionally, specify <on> or <off> for
<authenticated> to enable and disable authenticated proxy support. Specify
<username> and <password> if you turn authenticated proxy support on.

Usage: networksetup -setsecurewebproxystate <networkservice> <on off>
     Set SecureWeb proxy to  either <on> or <off>.

Usage: networksetup -getstreamingproxy <networkservice>
```

```
    Display Streaming proxy (server, port, enabled value) info for
<networkservice>.

Usage: networksetup -setstreamingproxy <networkservice> <domain> <port
number> <authenticated> <username> <password>
    Set Streaming proxy for <networkservice> with <domain> and <port
number>. Turns proxy on. Optionally, specify <on> or <off> for
<authenticated> to enable and disable authenticated proxy support. Specify
<username> and <password> if you turn authenticated proxy support on.

Usage: networksetup -setstreamingproxystate <networkservice> <on off>
    Set Streaming proxy to  either <on> or <off>.

Usage: networksetup -getgopherproxy <networkservice>
    Display Gopher proxy (server, port, enabled value) info for
<networkservice>.

Usage: networksetup -setgopherproxy <networkservice> <domain> <port
number> <authenticated> <username> <password>
    Set Gopher proxy for <networkservice> with <domain> and <port number>.
Turns proxy on. Optionally, specify <on> or <off> for <authenticated> to
enable and disable authenticated proxy support. Specify <username> and
<password> if you turn authenticated proxy support on.

Usage: networksetup -setgopherproxystate <networkservice> <on off>
    Set Gopher proxy to  either <on> or <off>.

Usage: networksetup -getsocksfirewallproxy <networkservice>
    Display SOCKS Firewall proxy (server, port, enabled value) info for
<networkservice>.

Usage: networksetup -setsocksfirewallproxy <networkservice> <domain> <port
number> <authenticated> <username> <password>
    Set SOCKS Firewall proxy for <networkservice> with <domain> and <port
number>. Turns proxy on. Optionally, specify <on> or <off> for
<authenticated> to enable and disable authenticated proxy support. Specify
<username> and <password> if you turn authenticated proxy support on.

Usage: networksetup -setsocksfirewallproxystate <networkservice> <on off>
    Set SOCKS Firewall proxy to  either <on> or <off>.

Usage: networksetup -getproxybypassdomains <networkservice>
    Display Bypass Domain Names for <networkservice>.

Usage: networksetup -setproxybypassdomains <networkservice> <domain1>
[domain2] [...]
```

```
     Set the Bypass Domain Name Servers for <networkservice> to <domain1>
[domain2] [...]. Any number of
     Domain Name servers can be specified. Specify "Empty" for <domain1> to
clear all
     Domain Name entries.

Usage: networksetup -getpassiveftp <networkservice>
     Display whether Passive FTP is on or off for <networkservice>.

Usage: networksetup -setpassiveftp <networkservice> <on off>
     Set Passive FTP to either <on> or <off>.

Usage: networksetup -getairportnetwork
     Display current AirPort Network.

Usage: networksetup -setairportnework <network> [password]
     Set AirPort Network to <network> using optional [password] if
specified.

Usage: networksetup -getairportpower
     Display whether AirPort power is on or off.

Usage: networksetup -setairportpower <on off>
     Set AirPort power to either <on> or <off>.

Usage: networksetup -getnetworkserviceenabled <networkservice>
     Display whether a service is on or off (enabled or disabled).

Usage: networksetup -setnetworkserviceenabled <networkservice> <on off>
     Set <networkservice> to either <on> or <off> (enabled or disabled).

Usage: networksetup -createnetworkservice <newnetworkservicename>
<hardwareport>
     Create a service named <networkservice> on port <hardwareport>. The
new service will be enabled by default.

Usage: networksetup -renamenetworkservice <networkservice>
<newnetworkservicename>
     Rename <networkservice> to <newnetworkservicename>.

Usage: networksetup -duplicatenetworkservice <networkservice>
<newnetworkservicename>
     Duplicate <networkservice> and name it with <newnetworkservicename>.

Usage: networksetup -removenetworkservice <networkservice>
     Remove the service named <networkservice>. Will fail if this is the
only service on the hardware port that <networkservice> is on.
```

```
Usage: networksetup -ordernetworkservices <service1> <service2> <service3>
<...>
     Order the services in order specified. Use "-listnetworkserviceorder"
to view service order.

Usage: networksetup -getappletalk <networkservice>
     Display whether AppleTalk is on or off (enabled or disabled) on
<networkservice>.

Usage: networksetup -setappletalk <networkservice> <on off>
     Set AppleTalk to either <on> or <off> (enabled or disabled) on
<networkservice>.

Usage: networksetup -getMTU <hardwareport or device name>
     Get the MTU value for hardwareport or device specified.

Usage: networksetup -setMTU <hardwareport or device name> <value>
     Set MTU for hardwareport or device specified.

Usage: networksetup -listValidMTURange <hardwareport or device name>
     List the valid MTU range for hardwareport or device specified.

Usage: networksetup -getMedia <hardwareport or device name>
     Show both the current setting for media and the active media on
hardwareport or device specified.

Usage: networksetup -setMedia <hardwareport or device name> <subtype>
[option1] [option2] [...]
     Set media for hardwareport or device specified to subtype. Specify
optional [option1] and additional options depending on subtype. Any number
of valid options can be specified.

Usage: networksetup -listValidMedia <hardwareport or device name>
     List valid media options for hardwareport or device name. Enumerates
available subtypes and options per subtype.

Usage: networksetup -version
     Display version of networksetup tool.

Usage: networksetup -help
     Display these help listings.

Usage: networksetup -printcommands
     Displays a quick listing of commands (without explanations).
```

A3. Modified CinchDefaults.plist in XML format

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://
www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
	<key>excludedAlways</key>
	<array>
		<dict>
			<key>comparison</key>
			<string>fullPath</string>
			<key>value</key>
			<string>~/Library/Mirrors</string>
		</dict>
		<dict>
			<key>comparison</key>
			<string>fullPath</string>
			<key>value</key>
			<string>~/Library/Logs/MirrorAgent.log</string>
		</dict>
		<dict>
			<key>comparison</key>
			<string>fullPath</string>
			<key>value</key>
			<string>~/Library/Preferences/com.apple.homeSync.plist</string>
		</dict>
		<dict>
			<key>comparison</key>
			<string>fullPath</string>
			<key>value</key>
			<string>~/.mcxlc</string>
		</dict>
		<dict>
			<key>comparison</key>
			<string>fullPath</string>
			<key>value</key>
			<string>~/.Spotlight-V100</string>
		</dict>
		<dict>
			<key>comparison</key>
			<string>fullPath</string>
			<key>value</key>
			<string>~/.TemporaryItems</string>
		</dict>
		<dict>
```

```
            <key>comparison</key>
            <string>fullPath</string>
            <key>value</key>
            <string>~/.Cleanup At Startup</string>
        </dict>
        <dict>
            <key>comparison</key>
            <string>startsWith</string>
            <key>value</key>
            <string>.afpDeleted</string>
        </dict>
    </array>
    <key>excludedItems</key>
    <array>
        <dict>
            <key>comparison</key>
            <string>fullPath</string>
            <key>value</key>
            <string>~/Library</string>
        </dict>
        <dict>
            <key>comparison</key>
            <string>fullPath</string>
            <key>value</key>
            <string>~/.Trash</string>
        </dict>
    </array>
    <key>excludedPrefItems</key>
    <array>
        <dict>
            <key>comparison</key>
            <string>fullPath</string>
            <key>value</key>
            <string>~/Library/Application Support/SyncServices</string>
        </dict>
        <dict>
            <key>comparison</key>
            <string>fullPath</string>
            <key>value</key>
            <string>~/Library/Caches</string>
        </dict>
        <dict>
            <key>comparison</key>
            <string>fullPath</string>
            <key>value</key>
            <string>~/Library/Logs</string>
        </dict>
```

```
<dict>
	<key>comparison</key>
	<string>fullPath</string>
	<key>value</key>
	<string>~/Library/Preferences/ByHost</string>
</dict>
<dict>
	<key>comparison</key>
	<string>fullPath</string>
	<key>value</key>
	<string>~/Library/Printers</string>
</dict>
<dict>
	<key>comparison</key>
	<string>fullPath</string>
	<key>value</key>
	<string>~/Library/Safari/Icons</string>
</dict>
<dict>
	<key>comparison</key>
	<string>startsWith</string>
	<key>value</key>
	<string>IMAP-</string>
</dict>
<dict>
	<key>comparison</key>
	<string>startsWith</string>
	<key>value</key>
	<string>Mac-</string>
</dict>
<dict>
	<key>comparison</key>
	<string>fullPath</string>
	<key>value</key>
	<string>~/Library/Preferences/com.apple.dock.plist</string>
</dict>
<dict>
	<key>comparison</key>
	<string>fullPath</string>
	<key>value</key>
	<string>~/Library/Preferences/com.apple.iChatAgent.plist</string>
</dict>
<dict>
	<key>comparison</key>
	<string>fullPath</string>
	<key>value</key>
	<string>~/Library/Preferences/com.apple.sidebarlists.plist</string>
```

```
            </dict>
            <dict>
                  <key>comparison</key>
                  <string>fullPath</string>
                  <key>value</key>
                  <string>~/Library/Preferences/loginwindow.plist</string>
            </dict>
      </array>
      <key>syncedFolders</key>
      <array>
            <dict>
                  <key>path</key>
                  <string>~</string>
            </dict>
      </array>
      <key>syncedPrefFolders</key>
      <array>
      </array>
</dict>
</plist>
```