Tips and Tricks For Macintosh Management

Version 14.0
Leopard

Summer 2008

As they say in the publishing business:

“This page left intentionally blank”

except for the obligatory picture of Tenshi:

“Is it nap time yet?”

Contents

Overview

Welcome to the first edition of the Leopard Tips and Tricks. This paper is a highlight of the best practices and tips & tricks of managing Macs in a Leopard environment. The previous version of the T&T doc covered Tiger management, and if you are mixing systems, you will need to use the older paper to get more information on legacy settings.

In a change from previous papers, I have dropped several sections that relate to basic server configuration, imaging, and other services separate from client management. The imaging issues alone have reached a level of requiring their own documentation. I suggest the official imaging document, as well as a visit to the Apple Discussion pages for more answers in that area.

What we are covering in this paper:

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

The T&T docs have been a personal project of mine since my first paper on Macintosh management in 1996 - covering how to set up At Ease for Workgroups. Since then, I have tried to both continue these papers and spent a lot of time working with the documentation folks internally at Apple writing portions of the official server documentation. Please check out the official User Management guide at the Apple web site: http://www.apple.com/server/documentation for the latest version, as well as the docs covering directory services, collaboration services, and many other features.

In my position as a senior field engineer for Apple, I spend most of my time embedded in systems and client management. This also means that I do not have the luxury of being able to provide technical support for anyone reading this guide, outside of official assignments with the Apple Education sales team. If you get hung up on a section of the guide, feel free to drop me a note; but I cannot guarantee a response within any specific timeframe. Using Applecare and Apple Professional Services is a much better path to getting support in a timely manner.

Please send along any comments, corrections, or kudos as you see fit.

John DeTroye

Sr. Consulting Engineer

Apple Education

Summer 2008

johnd@apple.com & johnd@me.com

1. Defining client management for Mac OS X

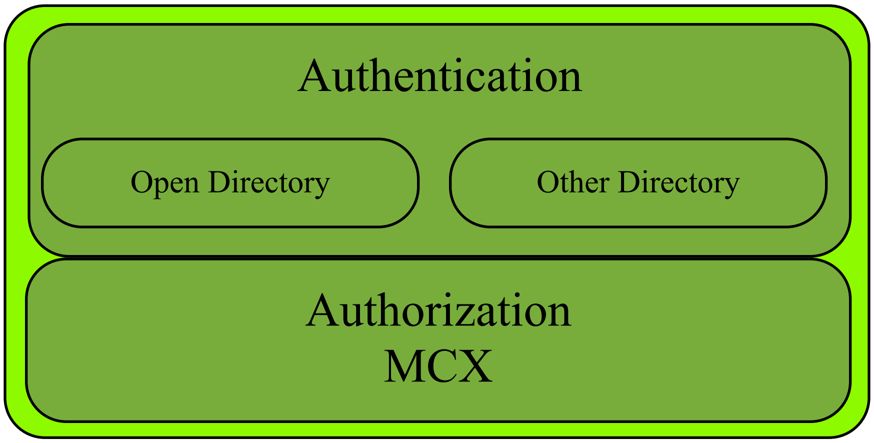
Being able to establish a stable user experience is a core definition of client management. MCX, or Managed Client for Mac OS X, is a subset of Open Directory, our directory service. The policies set for client systems are stored within a directory as part of either a computer, group, or user record. Using centralized management to storing management policies on a network database, sysadmins can easily define the user experience for a large number of computers owned by the institution. MCX settings are actively cached onto the client computers, allowing the management settings to stick to the system when away from the network, as very useful practice in the growing use of 1-1 deployments.

Being part of a directory, more specifically an LDAP (lightweight directory access protocol) directory, MCX is considered as the follow-on portion of the user experience when accessing a client computer. The first thing a user generally has to do is authenticate to a directory, whether that directory is stored locally or on the network. This authentication portion of the directory contains, at a minimum, the user's name and password. Once the user has authenticated to the directory, the user's authorization, or policy, is checked to see what items that user actually has permission to use. Let's dig into this a little deeper.

a. Directory Services - authentication to authorization

For a Leopard client, there are numerous methods to provide the necessary authentication and authorization databases. The three most common network directories are OpenLDAP (Apple's default on OS X server), Active Directory (Microsoft), and eDirectory (Novell). While the entire process of login and policy management can easily be performed using Apple's directory services, some sites choose to use one of the other directories to provide user account information, and sometimes even extend their directory schema, or mappings, to include the MCX settings. This paper will touch briefly on how a client may need to be configured to support a non-Apple directory focus; but for detailed information on these foreign directories, check out the Open Directory Admin guide on the Apple site.

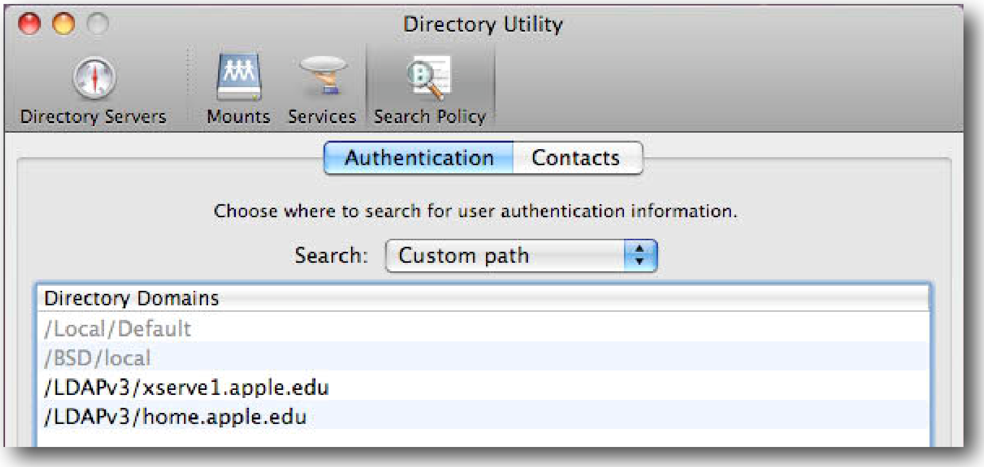
Here's where the services fit together:



Directory Services

While we (Apple) would prefer that you use only our directory services, I am sure that some of you have your hearts set on using AD/eD/etcD. Luckily, this capability was built into Leopard client, allowing the interaction of the different directories. Usually, this means that your users' authentication information - name and password - are stored in one directory, with the authorization information (MCX) being stored in OD on an OS X server. The process of using more than Apple's Open Directory environment is usually referred to as the "Golden Triangle" where the client system and two differing directories provide the information needed for both login and management.

If you set up binding for two directory services, make sure you list the MCX server first, then the authentication server. That way, when the user logs in, the search path always passes through the MCX directory on its way to the user authentication. For example, here's the search path for a setup with 'xserve1' carrying the MCX settings and the user accounts stored on 'home':



Choose where to search for user authentication information.

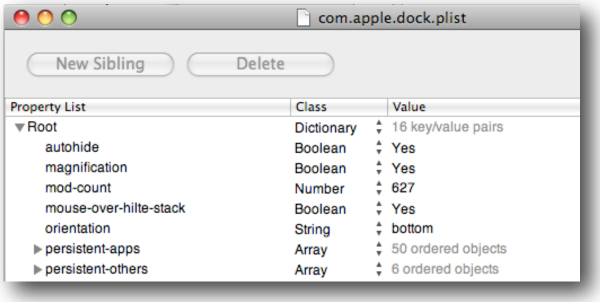
Search: Custom path

Directory Domains
/Local/Default
/BSD/local
/LDAPv3/xserve1.apple.edu
/LDAPv3/home.apple.edu

Set the search order to access the MCX directory first, then the user data

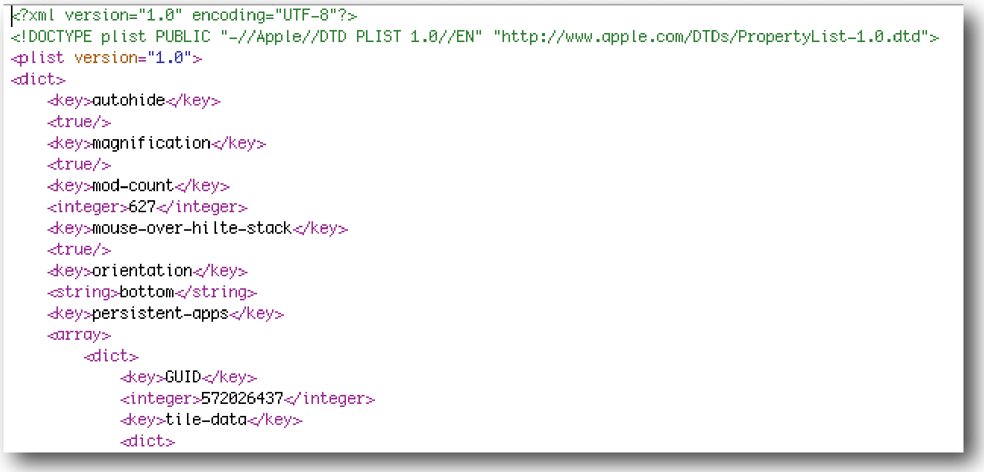
b. Managing preferences

Managed client settings are really nothing more than property values stored in the directory. Locally, you see the preferences stored in /Library/Preferences for the computer, and in ~/Library/Preferences for a specific user. When using a network directory to store these settings, they live inside a specific domain, such as a managed group (also referred to as a workgroup) or a computer group. The values, in both cases, are stored in XML format. Here's several ways to look at the same type of data, depending on where it is stored:



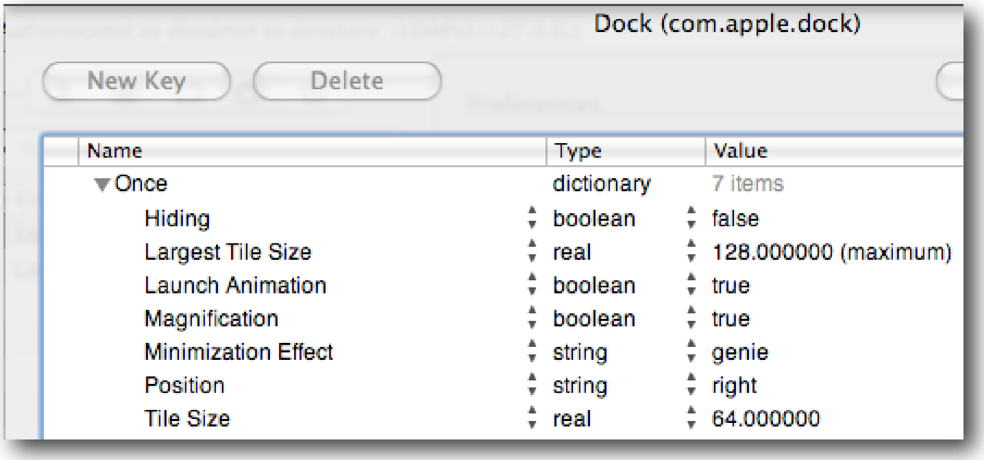
Property List	Class	Value
▼Root	Dictionary	16 key/value pairs
autohide	Boolean	Yes
magnification	Boolean	Yes
mod-count	Number	627
mouse-over-hilte-stack	Boolean	Yes
orientation	String	bottom
▶persistent-apps	Array	50 ordered objects
▶persistent-others	Array	6 ordered objects

Local user's Dock settings viewed from Property List Editor



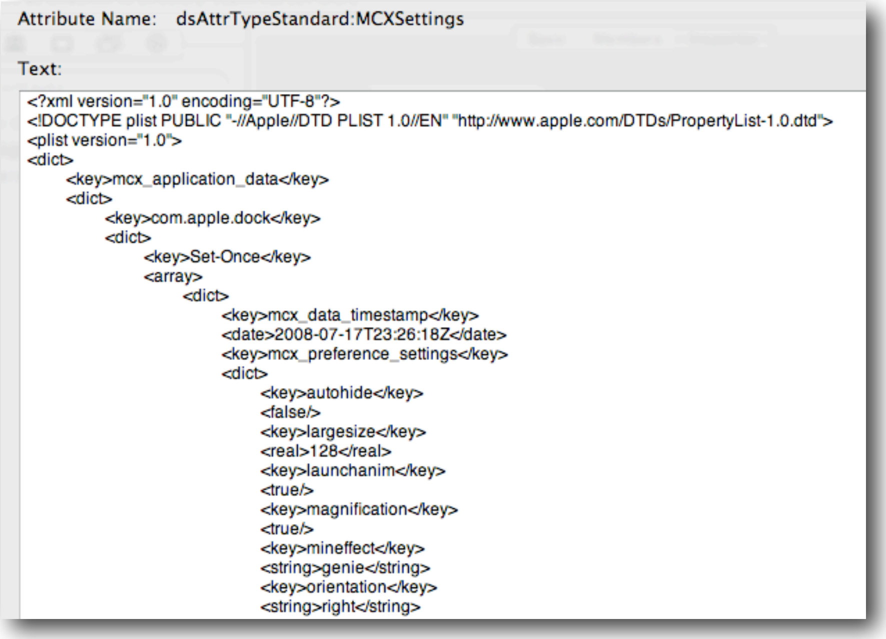
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>autohide</key>
    <true/>
    <key>magnification</key>
    <true/>
    <key>mod-count</key>
    <integer>627</integer>
    <key>mouse-over-hilte-stack</key>
    <true/>
    <key>orientation</key>
    <string>bottom</string>
    <key>persistent-apps</key>
    <array>
        <dict>
            <key>GUID</key>
            <integer>572026437</integer>
            <key>tile-data</key>
            <dict>
```

User’s Dock settings viewed as raw XML



Name	Type	Value
▼Once	dictionary	7 items
Hiding	boolean	false
Largest Tile Size	real	128.000000 (maximum)
Launch Animation	boolean	true
Magnification	boolean	true
Minimization Effect	string	genie
Position	string	right
Tile Size	real	64.000000

Network-managed Dock settings from inside Workgroup Manager



Attribute Name: dsAttrTypeStandard:MCXSettings

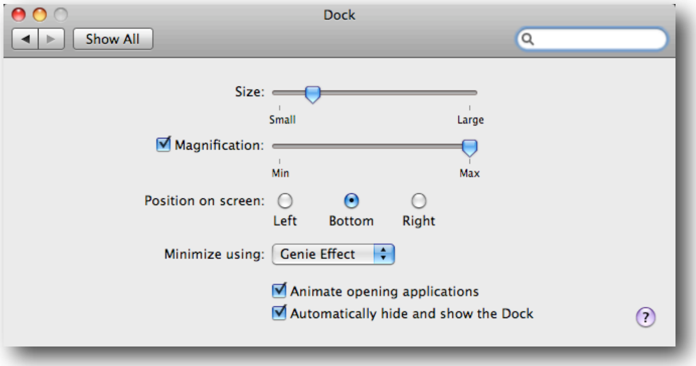
Text:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>mcx_application_data</key>
        <dict>
                <key>com.apple.dock</key>
                <dict>
                        <key>Set-Once</key>
                        <array>
                                <dict>
                                        <key>mcx_data_timestamp</key>
                                        <date>2008-07-17T23:26:18Z</date>
                                        <key>mcx_preference_settings</key>
                                        <dict>
                                                <key>autohide</key>
                                                <false/>
                                                <key>largesize</key>
                                                <real>128</real>
                                                <key>launchanim</key>
                                                <true/>
                                                <key>magnification</key>
                                                <true/>
                                                <key>mineffect</key>
                                                <string>genie</string>
                                                <key>orientation</key>
                                                <string>right</string>
```

Network-managed Dock settings viewed raw using Inspector

Setting the values locally for a single computer makes sense; but setting the value as it will impact hundreds or thousands of users may not. While you can set certain preferences on a base system - what I refer to as ‘client 0’ - then duplicate that computer for deployment; you might want to keep the base system pretty clean, then use managed preferences to establish the rules for the deployed systems based on who gets access to those computers.

The way most preferences are set in Workgroup Manager (WGM) for a set of systems isn’t that much different from the way those same values are set on a local machine. For example, here are the local and network Dock settings:



Size: Small — Large

☑ Magnification: Min — Max

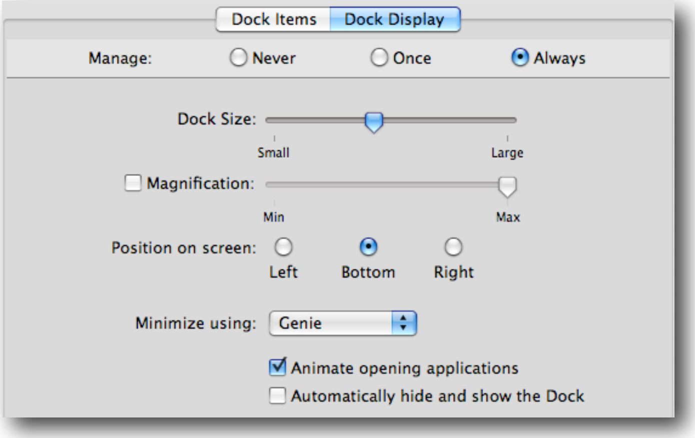
Position on screen: ○ Left ● Bottom ○ Right

Minimize using: Genie Effect

☑ Animate opening applications

☑ Automatically hide and show the Dock

Local Dock settings



Manage: ○ Never ○ Once ● Always

Dock Size: Small — Large

□ Magnification: Min — Max

Position on screen: ○ Left ● Bottom ○ Right

Minimize using: Genie

☑ Animate opening applications

□ Automatically hide and show the Dock

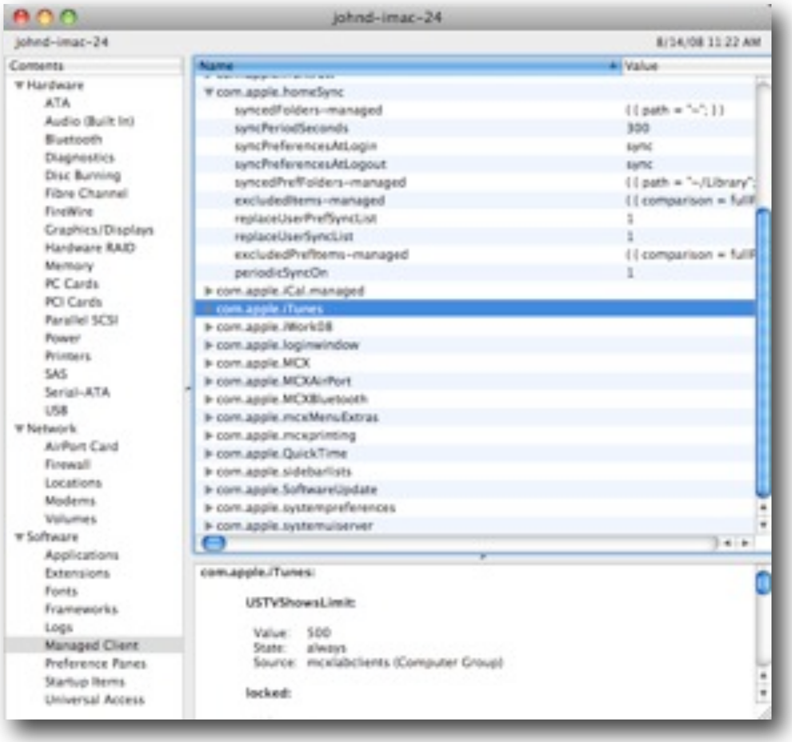
Network (WGM) Dock settings

c. Preference interaction - the rules of MCX

Before we go leaping into the actual setup of managed preferences on our system, we need to have a basic understanding of how all the parts interact. The preferences set within Workgroup Manager (WGM) are stored in one of four domains - user, (work)group, computer, or computer group. Some values can only be set at computer or computer group level, such as Energy Saver (doesn’t make any sense trying to set a user’s Energy Saver level, does it?).

There is a specific hierarchy for the preference settings. This ordering determines which items win out in a “left, right, center” or “on / off” argument, as well as establishing the order of listed items, such as in the Dock. The user account has the highest priority, the group the lowest. Preferences that a user sets for himself will be stored inside his home directory. Preferences set at directory level for a user get stored inside /Library/Managed Preferences/<username>. All other preferences are stored inside the cached “mcxsettings” in the local directory. The account order in which these preferences are obeyed are (highest to lowest): user, computer, computer group, workgroup (a user group with managed settings). The MCX system takes all of the preference settings and pulls them together using a tool called the compositor. This tool takes all the different preference settings and pulls them into a single property list (plist).

Some of the preferences are set to load at the loginwindow, such as the display of the login pane (list or name/password) and the loginwindow message. The rest of the preferences are applied at login for a specific user. You can view the final preference set in a number of different ways. One is with the Terminal command **mcxquery** which will allow you to dump the output of the compositor for any specific directory combination. Details on how this works can be found by typing **man mcxquery** into a Terminal window. The best way, in my opinion, to view the compositor results for a specific user is to log in as that user then launch “System Profiler” (easily accessible from the Apple Menu item “About this Mac” then “More Info…”). Here is what a set of values looks like for one user logged into a managed client system:



Name	Value
▼ com.apple.homeSync	
syncedFolders-managed	({ path = "~"; })
syncPeriodSeconds	300
syncPreferencesAtLogin	sync
syncPreferencesAtLogout	sync
syncedPrefFolders-managed	({ path = "~/Library";
excludedItems-managed	({ comparison = fullP
replaceUserPrefSyncList	1
replaceUserSyncList	1
excludedPrefItems-managed	({ comparison = fullP
periodicSyncOn	1
▶ com.apple.iCal.managed	
com.apple.iTunes	
▶ com.apple.iWork08	
▶ com.apple.loginwindow	
▶ com.apple.MCX	
▶ com.apple.MCXAirPort	
▶ com.apple.MCXBluetooth	
▶ com.apple.mcxMenuExtras	
▶ com.apple.mcxprinting	
▶ com.apple.QuickTime	
▶ com.apple.sidebarlists	
▶ com.apple.SoftwareUpdate	
▶ com.apple.systempreferences	
▶ com.apple.systemuiserver	

com.apple.iTunes:

USTVShowsLimit:

Value: 500
State: always
Source: mcxlabclients (Computer Group)

locked:

Managed user’s MCX settings in System profiler

Note that the settings show not only exactly what was set, but also where each setting came from. For example, in this picture, the source of the iTunes setting is the computer group “mcxlabclients” and the setting is for “Always”. This is a great way to be able to compare what you thought you set in Workgroup Manager versus what the compositor finally ends up with for the end user.

The hierarchy followed by the compositor is defined by the different types of accounts that can hold preference settings. As the compositor works, it looks through the different settings and resolves them in one of two ways. First, it takes any direct conflicts, such as Dock (left) / Dock (right), and sets the value to that of the preference belonging to the highest account priority. Second, it takes any lists, such as Dock items or printers, and concatenates them into a single list, tossing out any duplicates and ordering the list into a sequence from the highest priority to lowest.

The dominance is how well the preference *sticks* or retains its settings. There are four settings, three of which are actual domains. "Always" means that a preference is set by the admin to be permanent. Settings made in this domain usually cannot be altered by the end user. "Always" is often used for those settings that a user should not change, such as locking down items in the Dock that must be there always. "Often" is the domain provided inside the **Details** section of the preference editor. This domain allows the admin to establish a setting, yet the user may be allowed to edit that preference during the session. I say 'may' because the user must have the ability and permission to access the setting and change it. At logout, the settings are reverted to the initial values set by the admin. This domain is also the default location for non-standard settings, such as a preference file added outside of those in the GUI. "Often" is very useful in training where you may want the users to experiment with settings; but revert them to the presets at logout. "Once" is the domain that a admin uses to establish a starting point for a preference. The value has a timestamp attached to it, and if the user is allowed to make changes to the setting, then when they do so, the timestamp changes. As long as the admin doesn't touch that setting again, the user will retain ownership of the setting. If the admin changes the setting, the timestamp is now newer than that of the user, so the setting now displays the value set by the admin - 'once again'. "Once" is best used as a guideline for new users, or as a template. A group of users in training may be exposed to a series of settings. If you use "Once" to define those same settings for their production machines, they will be starting off with the settings from training; but will change them to suit their wants/needs.

d. A few thoughts on preference enforcement and AUP's

While it may seem like a really good idea to lock down the client systems, you should consider that the computer is there to help the end user accomplish assigned tasks. Try to set as few preferences as necessary to get the job done versus turning the computer into a featureless box (let's leave that to Dell, ok?). The phrase 'pick your battles' comes to mind. I have found that turning on every possible restriction only drives many users to spend an inordinate amount of time trying to get around the restrictions. If you need to enforce restrictions, start with the Acceptable Use Policy. No amount of technology is ever going to secure your client systems if you do not aggressively enforce the AUP.

2. Configuration tips for Mac OS X server

Setting up your environment for management involves some basic configuration of both the back end architecture and the client systems. This section deals with the basic settings to make sure your server(s) are properly configured to support MCX.

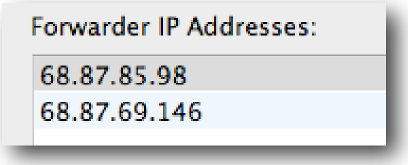
The services you need to provide will depend entirely on the size and scope of your deployment. I am not going to get into a full scale walkthrough of the entire Mac OS X server setup; but will cover those settings that are required. I have included several idea for extending the architecture for workflow and collaboration later in this paper. What you do need to provide is both authentication and authorization. To support those two key components, you must provide the following services:

- DNS (to resolve server names for clients)
- NTP (Network Time Protocol, for keeping everyone on the clock)
- DHCP (or fixed IP services)
- Open Directory (or some directory service)
- AFP (preferred, or SMB, for home directories and workflow)
- Other services as desired, such as Web, Software Update, iCal, etc.

a. DNS (Domain Name Services)

Every server should have a static IP address and should have its name registered with a DNS server. This allows the clients to locate the server, and for the directory services to store paths to settings in a clear, concise manner. Mac OS X, being UNIX, requires DNS, so don't bother trying to work around it. If your upstream ISP or IT folks will provide you with a fixed address and name, then you are all set. If they do so, make sure that the name and IP can be resolved from within the network you are operating on. If they don't want to help, you can set up your own DNS easily.

Server Admin has a totally new DNS interface for Leopard. When you first set up your server, you might have to enter an upstream server for your DNS setting; but you can change that after we get this set up. It will not interfere with any DNS service being provided by your ISP. First, launch **Server Admin** and connect as the local admin to your server. Select the **DNS** service (if you don't see it, add the service under the "Settings / Services" tab when looking at the server itself). Select the "Settings" tool under "DNS". Add your upstream DNS settings to the "Forwarder IP Addresses" window:



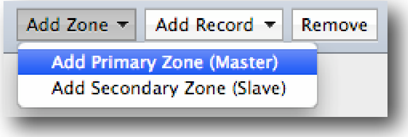
Forwarder IP Addresses:

68.87.85.98
68.87.69.146

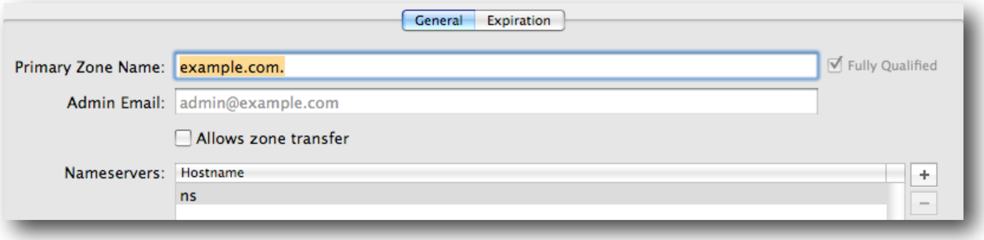
Upstream DNS servers

Note: When making entries inside Server Admin -always- hit Tab after any entry. This makes sure the value gets written properly. Don't believe me? Good luck…

The forwarder addresses allow you to point your clients at your DNS server, yet allows them to look up any Internet address out there in the tubes. Next, select "Zones" and click on the "Add Zone" popup. Choose "Add Primary…"



The initial values need to be replaced with your own settings (hit Tab at each entry!):



Primary Zone Name: example.com. ☑ Fully Qualified

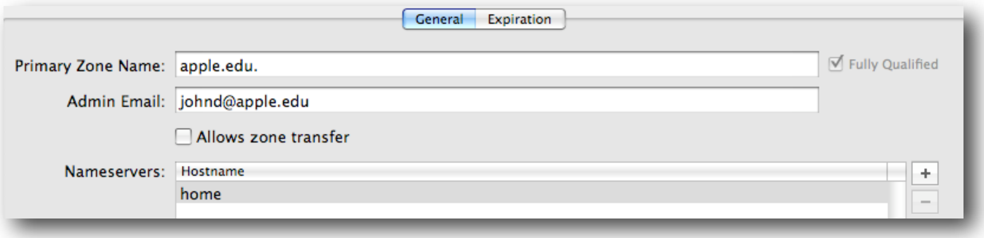
Admin Email: admin@example.com

☐ Allows zone transfer

Nameservers: Hostname

ns

Change the example values to those of your actual server



Primary Zone Name: apple.edu. ☑ Fully Qualified

Admin Email: johnd@apple.edu

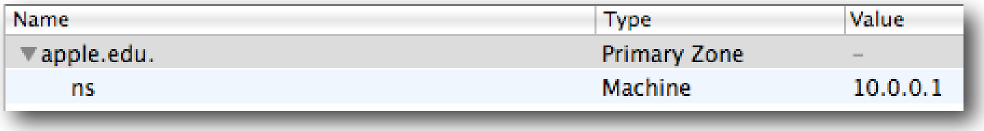
☐ Allows zone transfer

Nameservers: Hostname

home

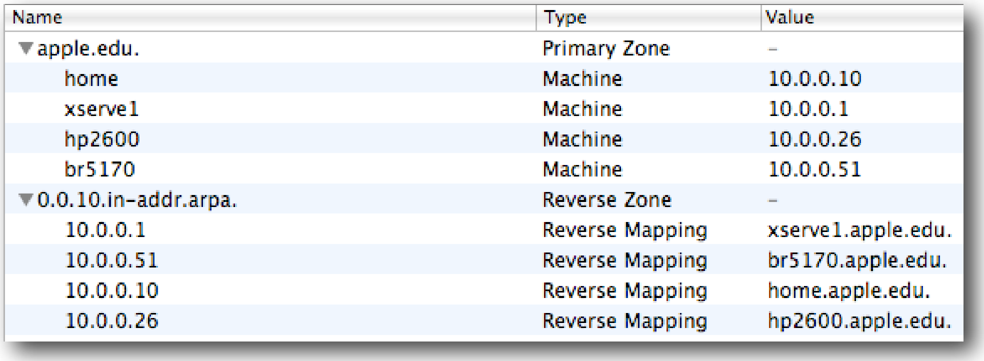
… like this.

Next, select the new zone, turn down the triangle to edit, and change the "ns" value to match your server. (Tab!)



Name	Type	Value
▼apple.edu.	Primary Zone	–
ns	Machine	10.0.0.1

Add any other "A" records you need to. Note that the reverse settings are automatically applied. When you are all done, you will have something roughly like this:



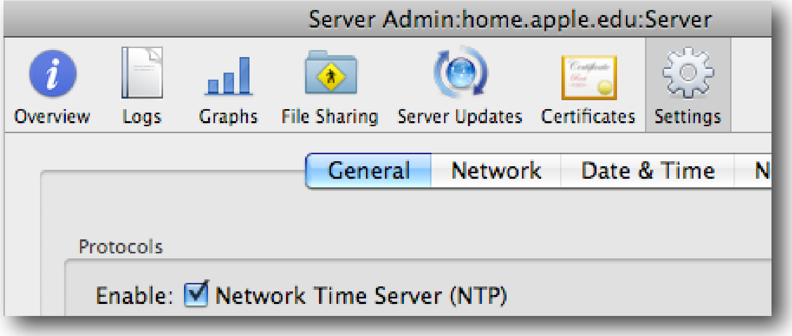
Name	Type	Value
▼apple.edu.	Primary Zone	–
home	Machine	10.0.0.10
xserve1	Machine	10.0.0.1
hp2600	Machine	10.0.0.26
br5170	Machine	10.0.0.51
▼0.0.10.in-addr.arpa.	Reverse Zone	–
10.0.0.1	Reverse Mapping	xserve1.apple.edu.
10.0.0.51	Reverse Mapping	br5170.apple.edu.
10.0.0.10	Reverse Mapping	home.apple.edu.
10.0.0.26	Reverse Mapping	hp2600.apple.edu.

A properly configured DNS

Your final action will be to go into System Preferences on your server and change the DNS settings for your network interface to point back to itself. This is needed to make sure all other services work properly. Since your server has the ability to forward any queries it can't answer, it will become a very good local DNS for your users.

b. NTP (Time Services)

You can turn on NTP on your server, if you wish, to provide a clock check for your clients. If you don't do this, then make sure this server, as well as all clients, are pointing to the same NTP server.



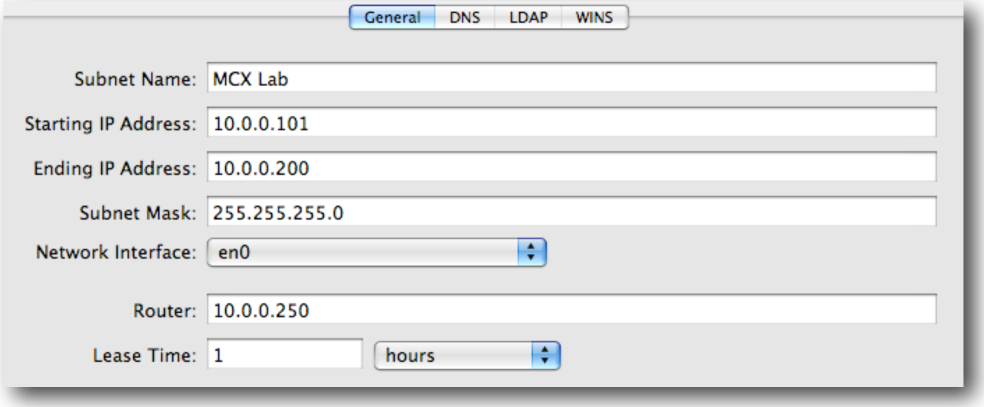
Enable: ☑ Network Time Server (NTP)

Not rocket science - but essential

The reason NTP is so important is that directory services uses Kerberos for user authentication. This service demands that all clients have their clock within 5 minutes or less than that of the server. If the clocks are off, users cannot log in. Period.

c. DHCP (IP services)

All networked clients require an IP address, and OS X server works as a very nice DHCP server. If you are already getting IP addresses from another device on the network, don't worry about setting this up. For those of you who want to see it - here you go. In **Server Admin**, select your server, then the **DHCP** service. Choose the "Subnets" tool and click on the "+" to create a new set.



Subnet Name: MCX Lab

Starting IP Address: 10.0.0.101

Ending IP Address: 10.0.0.200

Subnet Mask: 255.255.255.0

Network Interface: en0

Router: 10.0.0.250

Lease Time: 1 hours

Fill in the settings so they support your network

Do **not** create your own DHCP server if there is already one running on your network. Go ahead and add the DNS entries, and leave the rest alone for now.

d. Open Directory

To provide centralized management policies, you must have a directory to hold those policies. If you are using AD or eDirectory, you can extend the schema to add MCX settings - but that is something left to our Professional Services gang and will not be covered here. What we are going to do is create an Open Directory master (ODM) to contain the management settings used by all the client systems. If you are in a small enough environment, less than a thousand client systems, you can use the same server for your user accounts that you use for your management settings. I am going to cover setting up the authentication directory separate from the authorization directory. I do this because it allows you to learn to keep these items separate in your head for planning purposes, and it allows you to expand as needed when your installed base grows.

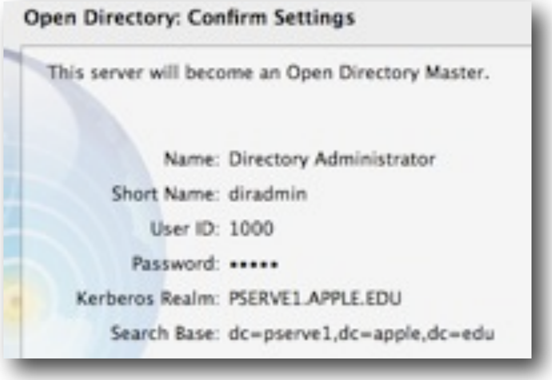
Suppose, for example, that someone forced you to use Active Directory for your user accounts. If you had no access to manage those accounts, the concept of adding MCX settings could be daunting. If you deploy an OS X server, however, to provide the policies to the Macs; then you can easily work with that environment. This concept works even if you have a large OD directory, such as a single district wide personnel database, and differing management needs in each school. You can deploy an ODM as the MCX server in each school to provide unique management for that environment; but still require the users to authenticate to the district db. Just a thought.

1. Setting up the user directory server

If you are providing your own ODM for user authentication, with or without the MCX settings on this system, you need to follow these steps. If you are using a different directory for user accounts, just skip to the next section.

First, you need to make sure your server has a valid FQDN (fully qualified domain name). Using the Network Utility or any tool of choice, check to make sure you can do forward and reverse lookups of your server both from itself and a client system.

Next, in **Server Admin**, connect to your server and select the “Open Directory” setting in the sidebar (add it first in the Server/Settings/Services tab if necessary). In the “General” pane, click on the “Change” button. Select “Open Directory Master” and add a password for the ‘diradmin’ account. In the ‘Master Domain Info’ you should see both a Kerberos realm and a search base. Confirm the settings.



Open Directory: Confirm Settings

This server will become an Open Directory Master.

Name: Directory Administrator
Short Name: diradmin
User ID: 1000
Password: •••••
Kerberos Realm: PSERVE1.APPLE.EDU
Search Base: dc=pserve1,dc=apple,dc=edu

Example settings for an ODM at setup

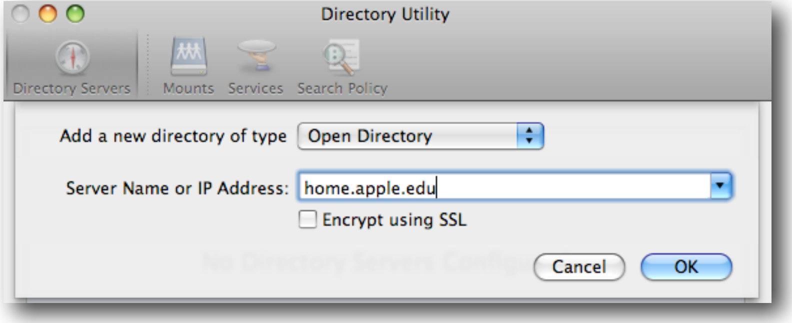
If you don't get settings similar to this, go back and make sure your DNS setup is working.

Note: Do not use the previous setup if you are going to have an upstream directory server. That server would have Kerberos running on it, and you must do a few extra steps before setting up your ODM.

2. Setting up the managed client directory server

This setup would be used when the OD server is going to be part of a larger directory environment. This serves for a MCX server being secondary to any other directory server that would provide authentication, such as an OD and AD server that would contain the user records. For this setup, I will use an upstream OD server that contains the user records.

First, you must log into the MCX server as the local admin and launch **Directory Utility**. Unlock the window and click on the "+" sign. Select the type of directory your upstream server uses. In our case, I selected "Open Directory" and entered "home.apple.edu" which is my user account server.



Add a new directory of type: Open Directory

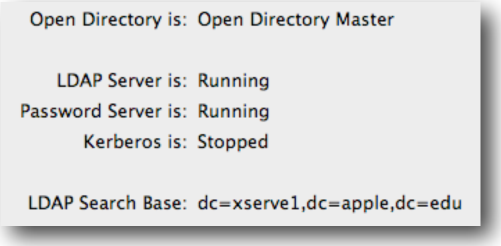
Server Name or IP Address: home.apple.edu

□ Encrypt using SSL

Binding the MCX server to the OD server with user accounts

You do not need to make this an authenticated bind unless you will also be using the MCX server as the wiki/blog and/or collaboration server. Anonymous binding will work fine for now.

Now you can go into **Server Admin**, into the Open Directory settings and make this server an OD master. The process is almost the same as in the section above; but you won't get any Kerberos information. OS X server senses that it is bound to a Kerberos server and won't activate the code on your MCX server. When you create the 'diradmin' account, you need to change it be different from the directory administrator account on its parent server. I suggest something easy such as 'MCX Admin' with a short name of 'mcxadmin' - otherwise, you'll get authentication errors trying to connect. The final setup for my MCX server looks like:



Open Directory is: Open Directory Master

LDAP Server is: Running
Password Server is: Running
Kerberos is: Stopped

LDAP Search Base: dc=xserve1,dc=apple,dc=edu

The MCX server does not have Kerberos running

e. AFP (Appleshare file protocol)

The MCX server may not have any sharepoints on it; but we still need to go through the process for general knowledge. In Appendix A, we will discuss several criteria for load planning, plus user account types and usage is in section 6. One of these criteria is making sure you do not overload any of your servers in such a way as to impact other services. If you have a server providing policies, it can handle some other tasks; however, one of those tasks should not be providing home directory support. If you are using dedicated network user accounts with network based home directories, the load on the file server can be immense. If you are using mobile accounts with portable home directories, the load can still be pretty heavy. Best practice for this is to have dedicated home directory servers.

This doesn't mean that you can't use the MCX server for other purposes. You can house common or group sharepoints on this server, as well as possibly support some of the new collaboration services, such as wiki or blog. In this section, I'll cover setting up the basic file sharing to support all of these possibilities, and leave it to you to decide which ones you activate on your various servers.

1. Basic AFP service

Basic AFP service doesn't need any hand holding in Leopard. You just turn it on. Guest access is off by default, so any sharepoints will be off limits to anyone but authenticated users. If you wish to allow guests, just click in the "Enable Guest Access" checkbox in the 'Settings/Access' pane.

2. Creating sharepoints

In Leopard, the sharepoint setup has moved from Workgroup Manager to Server Admin. It really makes sense being here - it's where all the services are established. To create a sharepoint, you open **Server Admin**. connect to your server, and select the "File Sharing" toolbar item. Using the "Browse" tab, you can locate, or create the folder you wish to share. Just click on the "Share" button then save. Setting permissions, and more advanced ACL's is covered in the official documentation.

3. Automounts

Under Leopard, the new AutoFS code is much more reliable than that for Tiger. Leopard clients do not misbehave the way Tiger clients did (or do if you still use Tiger clients with a Leopard server). One thing you can do is make sure the number of automounts you configure per server is as small as possible. When you have many servers on the same network, the clients still have to keep track of all of them; but under Leopard, the sharepoints can mount and dismount without creating the problems we saw in Tiger where a logged out user's sharepoint refused to disconnect, leaving the next user unable to log in.

To set up a sharepoint as an automount in Leopard, you use **Server Admin** to select the sharepoint within the "File Sharing" tool, then select the "Enable Automount" checkbox. Choose the type of automount - usually "User home folders" - then click "Ok". Authenticate as the directory administrator on that server (be careful, if it's on the MCX server we set that as 'mcxadmin', not 'diradmin').

This completes the basic setup of the server. We will expand on these configurations later in the paper.

3. Configuration tips for Mac OS X clients

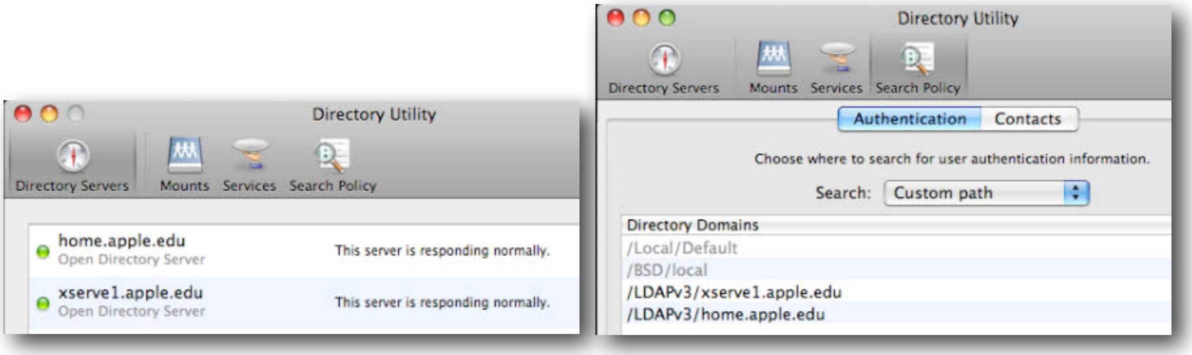
Setting up Mac OS X clients for MCX is pretty straight forward. This paper is not going to cover imaging; so all that you need to bring to the table here is a client system running Mac OS X v10.5.4 (highly recommended); we also support 10.4.11 and 10.3.9). I will cover a few of the aspects of Tiger client behavior within the Leopard MCX world; but I won't be mentioning Panther systems after this, other than to say that yes, the basic management settings do work.

Clients should have all the appropriate software installed. If a system will be used by many people, then install all the needed software and use MCX to control access to the software as needed. At least one client system should have the Server Admin software installed to allow this system to act as "Client Zero" for purposes of setting specialized MCX configurations.

All clients should have a local admin account, hidden or not, so that basic maintenance can be performed on and off network. This will also be the account you would log into to run Workgroup Manager and perform MCX setup. All clients need to have unique names - and you should avoid special characters in the names.

The most difficult task we have to perform here is binding the client to the management server(s). If you have only one server providing both user authentication and management authorization, then a single bind is all you need. Dual binding is needed if you are attaching to a directory for access to user account information as well as to a MCX server for management settings. The tool for this is **Directory Utility** located in the /Applications/Utilities folder (not to be confused with the "Directory" tool, also located in the Utilities folder…).

Log into the client as a local admin and launch the **Directory Utility** application. Unlock, and click on the "+" sign. Enter the MCX server's domain information in the window, click "Ok" and wait for the dialog to go away. Click on the "+" again and enter the user directory server info. When done, you need to check to make sure the search path for directory services is correct. What you want is for the system to search through the local directory first, then through the MCX server, and finally to the user directory. This insures that all MCX settings are picked up while trying to authenticate as a user. If you switched the order, the user may log in; but have no management settings.



home.apple.edu
Open Directory Server
This server is responding normally.

xserve1.apple.edu
Open Directory Server
This server is responding normally.

Choose where to search for user authentication information.

Search: Custom path

Directory Domains
/Local/Default
/BSD/local
/LDAPv3/xserve1.apple.edu
/LDAPv3/home.apple.edu

Note that the list is alphabetical - but the search order is correct

With this done, we can now set the MCX values for our client systems.

4. Basic MCX configuration setup

Setting up basic management policies for Mac OS X requires a few basic considerations. First, determining the level at which management will be applied. Second, establishing baseline preference settings. Third, testing the setup and tuning.

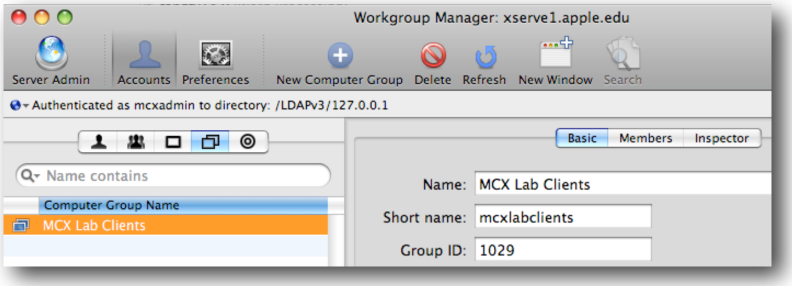
The first consideration is based on which domain, or domains, you desire to use for management. The domains are user, group, computer, and computer group. Under earlier versions of MCX, as well as Macintosh Manager for OS9, I usually recommended setting policies at group level. This was because we relied heavily on workgroups for both user management and for workflow. What this resulted in was users having to choose among several different workgroups at login in order to have access to a group folder and group settings. This setup could still be used except for a radical improvement in Leopard for group policy management. Under Leopard MCX, workgroup preference settings are combined by default into a single set of values. This means that instead of having to choose between the Math, Science, or Language Arts workgroups when logging in, you would just authenticate and be taken directly to the Desktop, All the settings for each of those workgroups would be composited together giving you all the Dock items, and the composite of all the other settings.

Is this a better way to do things? I think so. Over the past few years, I have heard from many people asking for something better than asking the users, usually teachers and students, to have to log out and in again between classes, or to have to pick a specific workgroup for every class. With Leopard, we can now have a student or teacher, or other faculty member, log in as themselves and get a single work environment that encompasses all they need to do for that day. This is especially important as we increase the number of 1-1 deployments with users carrying their own computers from class to class. Log in once and be done with it for the day.

That said, how do we set up the management scheme to best support this new capability, yet allow for workflow and use of both Leopard and Tiger systems? I feel that the best practice is to use computer group level management most of the time, with some use of workgroup settings for unique circumstances. Why this works best is that we can streamline the user experience, and make it similar between Leopard and Tiger clients. Under the new group combining setting, Leopard users will get a single user setup at the Desktop; but Tiger users would still get the workgroup picker. If the management was more computer group based, both sets of the systems would respond the same.

So what are these different domains, and how do they relate? The user domain is the user account. Setting preferences from the network for a user is possible; but it is probably much too granular. If you set preferences for a specific user, or mass set them for a group of users selected together, you will have edit those preferences later on one by one. Where individual preferences can work well is allowing special permissions for a user outside what that user's group(s) or computer group(s) allow. The user domain has the highest rank in the hierarchy, so settings made at this level will override any made to the same preference at any of the other levels.

Computer groups are new to Leopard. Under Tiger, we had computer lists. A set of computers could belong to a single list, such as 'teacher_computers'. With Leopard, we can assign computers to as many groups as we need, the same as assigning users to different groups for various admin and management needs. Now we can have a single system in the computer groups 'teachers', 'portables', 'Middle School', and 'Science' all at once. These groups may or may not have any management assigned. Computers that are bound to the directory can be added to any computer list. The computer is tracked by its Sharing name and the primary (onboard) MAC address. If a computer has more than one network interface, the computer record will contain multiple MAC addresses.



Workgroup Manager: xserve1.apple.edu

Authenticated as mcxadmin to directory: /LDAPv3/127.0.0.1

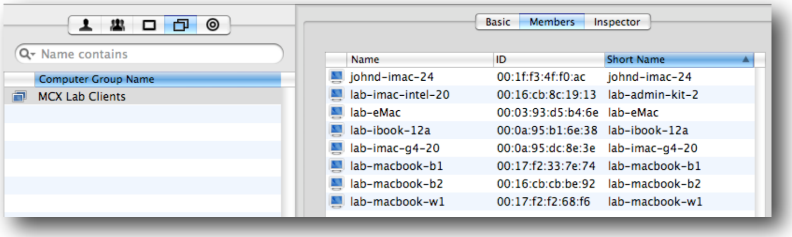
Computer Group Name
MCX Lab Clients

Name: MCX Lab Clients

Short name: mcxlabclients

Group ID: 1029

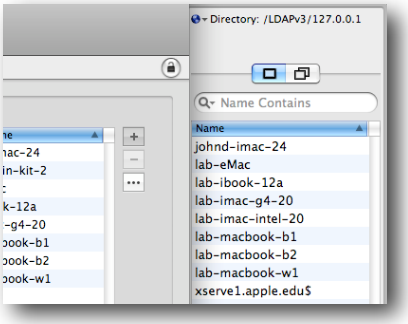
A computer group with…



Computer Group Name
MCX Lab Clients

Name	ID	Short Name
johnd-imac-24	00:1f:f3:4f:f0:ac	johnd-imac-24
lab-imac-intel-20	00:16:cb:8c:19:13	lab-admin-kit-2
lab-eMac	00:03:93:d5:b4:6e	lab-eMac
lab-ibook-12a	00:0a:95:b1:6e:38	lab-ibook-12a
lab-imac-g4-20	00:0a:95:dc:8e:3e	lab-imac-g4-20
lab-macbook-b1	00:17:f2:33:7e:74	lab-macbook-b1
lab-macbook-b2	00:16:cb:cb:be:92	lab-macbook-b2
lab-macbook-w1	00:17:f2:f2:68:f6	lab-macbook-w1

many members, added from…



Directory: /LDAPv3/127.0.0.1

Name
johnd-imac-24
lab-eMac
lab-ibook-12a
lab-imac-g4-20
lab-imac-intel-20
lab-macbook-b1
lab-macbook-b2
lab-macbook-w1
xserve1.apple.edu$

the list of bound computers.

The computer domain is just that - the individual computer record. Just like a user record, a computer record can contain unique settings to make it vary from the rest of the computers in any specific computer group. A special case is the 'Guest Computer' which is defined as any computer bound to the directory server but not specifically designated.

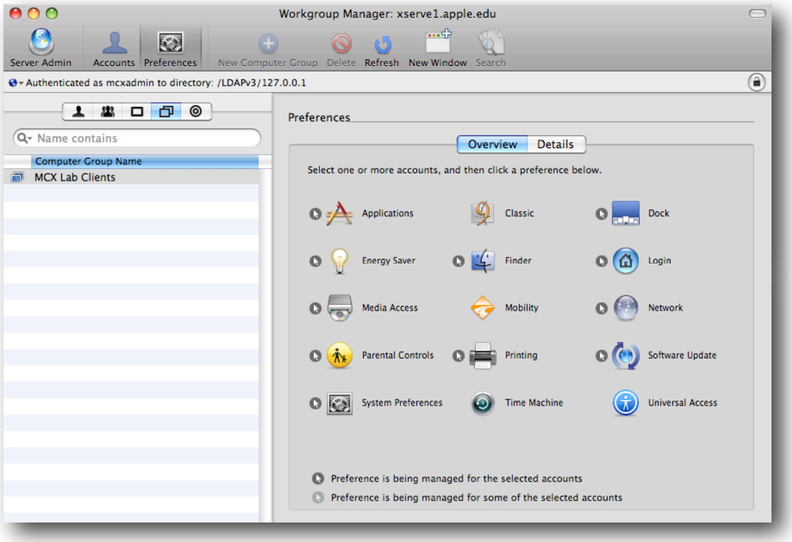
Note: In Leopard, the Guest computer account is no longer available by default as it was in Tiger. To enable and use the Guest computer account go to the 'Server' menu in Workgroup Manager (WGM) and select "Create Guest Computer"

Workgroups are user groups with preferences assigned. This group holds the lowest priority, and should be used only when the settings must be very unique for a specific set of users. Since we can create ‘group’ folders that reside together in a common sharepoint (see the section on workflow and collaboration), we don’t need workgroups just to provide the standard ‘hand in folder’ any more. Where a workgroup might come in handy is when setting up a special workgroup for loading sharepoints at login that aren’t normally used. Then again, you could also set that same setting at computer group level - it depends on the usage and client deployment model.

With all that specific info (now you know why we say “It depends” a lot…), let’s look at the basic settings we need to employ to begin managing our workstations. We begin by logging into an admin station or client with server tools installed as the local admin. Do **not** run Workgroup Manager from a server. Not that it won’t run; but the server isn’t configured or loaded as a client, so you’ll have a much more difficult, if not impossible, time of setting many of the preferences. Just think of having to install all the applications, widgets, and other tools on the server just to be able to set management values. You also wouldn’t be able to mount sharepoints that you’ll want to add to Dock or Login items later on.

a. Login preferences

The first indication a user gets of being managed is at the loginwindow. It is also a very good way of insuring that MCX settings have made it to the client. To set up Login preferences, select either the Guest Computer account or one of the computer group accounts. Select the Preferences tool.



Setting preferences at computer group

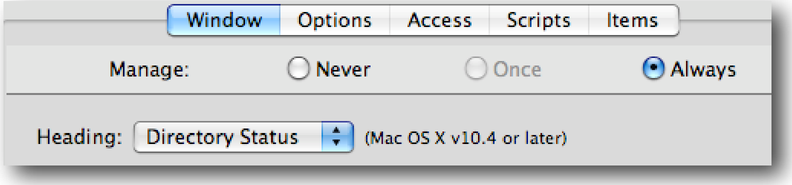
Now select the Login item. There are five different sets of values that can be added to the Login mcx domain - Window, Options, Access, Scripts, and Items. For the basic setup, we’ll explore a few of these, leaving the rest for later.

Login preference tabs

Let's tackle 'Window' first. It is here that we can set the view of the loginwindow, add a message for the end user, and display system information.

1. Window

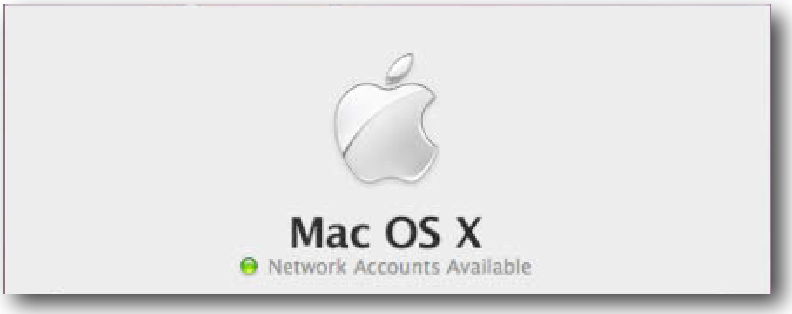
In the 'Window' tab, select 'Always' and set the "Heading' to display 'Directory Status'. The 'Heading' is also called 'AdminHostInfo' in the database and is the set of values you see in the login window underneath the large "Mac OS X". If you click on these values, you will rotate through the set - Name, OS version, OS build, serial number, IP address, directory status, and time. The default for any system is the name of the computer. We're setting this to 'Directory Status' so we can see at a glance if the client is properly bound to its directory server(s).



Manage: ○ Never ○ Once ● Always

Heading: Directory Status (Mac OS X v10.4 or later)

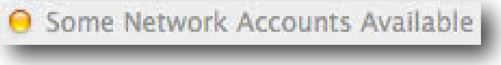
Setting the login window to display directory status gives us …


Mac OS X

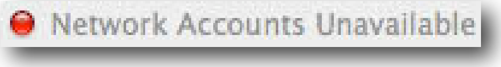
● Network Accounts Available

the fact that we are bound to our servers correctly.

This 'jelly' usually starts out red at boot time, then will turn green within 30-45 seconds (or faster). There are four possible values for this window:


● Some Network Accounts Available

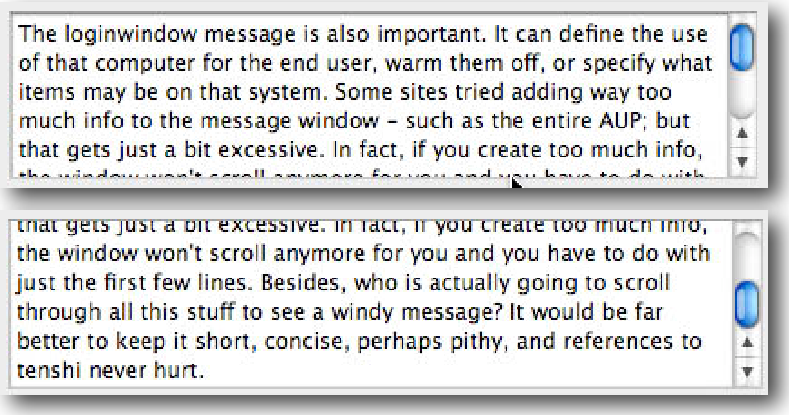
One or more of my directory servers isn't reachable


● Network Accounts Unavailable

None of my directory servers is reachable

The fourth value is "Network Login Required", a yellow tag, showing that your client is managed with 802.1x (Radius). Can't get a screen shot of that since to do it I need to have my ARD client connected to the network; but if it's connected the value changes. ;-)

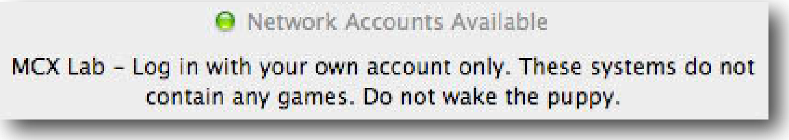
The loginwindow message is also important. It can define the use of that computer for the end user, warm them off, or specify what items may be on that system. Some sites tried adding way too much info to the message window - such as the entire AUP; but that gets just a bit excessive. Here's an example of a long message, followed by one a bit more appropriate. If the message is longer than the basic window, scroll bars appear.



The loginwindow message is also important. It can define the use of that computer for the end user, warm them off, or specify what items may be on that system. Some sites tried adding way too much info to the message window – such as the entire AUP; but that gets just a bit excessive. In fact, if you create too much info,

the window won't scroll anymore for you and you have to do with just the first few lines. Besides, who is actually going to scroll through all this stuff to see a windy message? It would be far better to keep it short, concise, perhaps pithy, and references to tenshi never hurt.

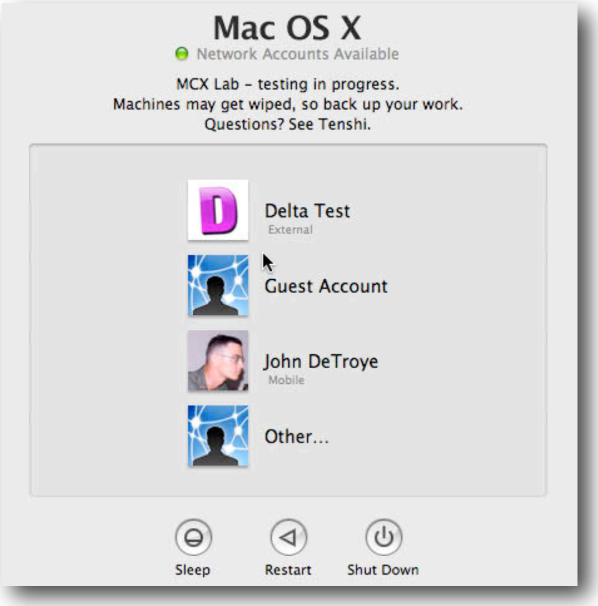
A bit much…



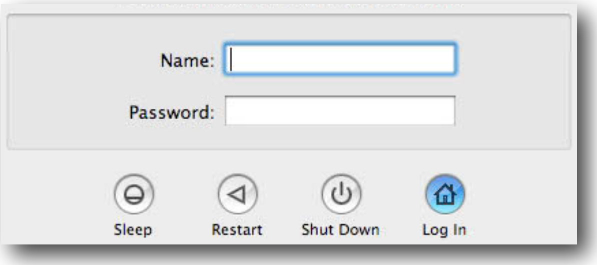
MCX Lab – Log in with your own account only. These systems do not contain any games. Do not wake the puppy.

An excellent and cogent loginwindow message.

The final portion of the loginwindow to define is the display style. You can choose either to view just a name/password field set, or display user names with a picture in a list. If you are using 802.1x authentication, you must use only the name/password set since the user list isn't cached until after the network connection is established. In some schools, or even just certain grades, having the student picture and name as part of the login window is a good idea. Adding a 'Picture' field to the user directory with a path to a small image jazzes up the user experience. If you have thousands of users though, this could create a delay while the loginwindow loads, as well as forcing users to scroll through a long list. You can also choose whether or not to display the 'Restart' and/or 'Shutdown' buttons.



MCX Lab – testing in progress.
Machines may get wiped, so back up your work.
Questions? See Tenshi.

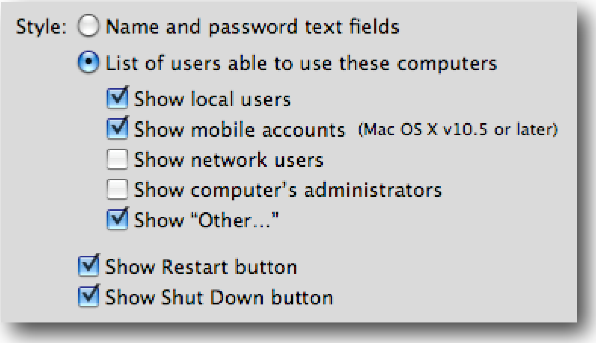


Name:

Password:

A couple of style variations on Login Window

The list view can be trimmed to display only local accounts, such as the Guest account, mobile accounts, and hide admins and network users. If a user was logging in for the first time as a network user with a mobile account, in this case, they would use the "Other…" selection. After that, their account would be visible in the window.



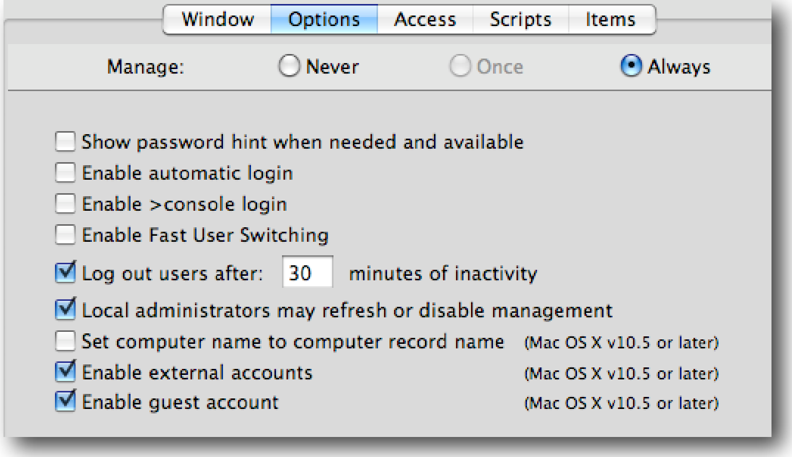
Style: ○ Name and password text fields
● List of users able to use these computers
☑ Show local users
☑ Show mobile accounts (Mac OS X v10.5 or later)
☐ Show network users
☐ Show computer's administrators
☑ Show "Other…"

☑ Show Restart button
☑ Show Shut Down button

Style options

2. Options

The 'Options' tab contains settings that limit non-standard logins, force logouts, and determine is local admins are managed or not. It also contains the flags for globally enabling guest and external accounts. These two items will be discussed in detail in the user account section.



Manage: ○ Never ○ Once ● Always

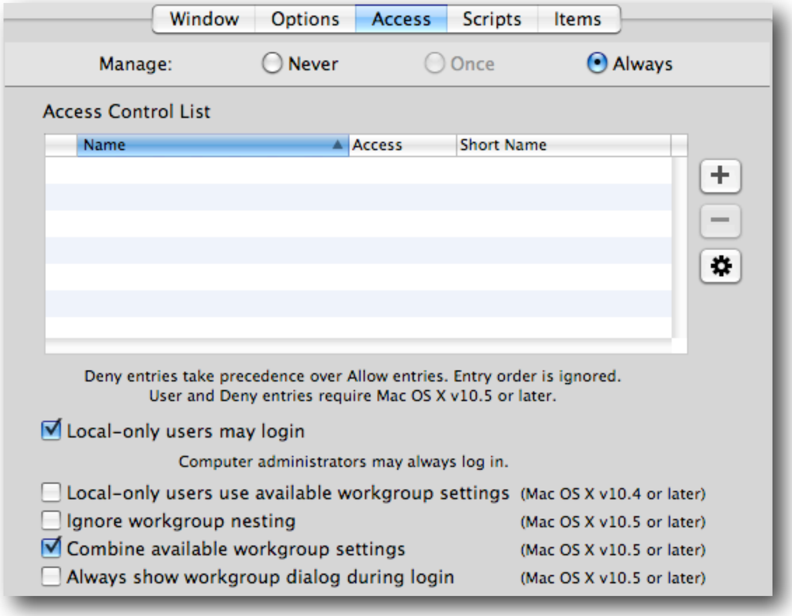
☐ Show password hint when needed and available
☐ Enable automatic login
☐ Enable >console login
☐ Enable Fast User Switching
☑ Log out users after: 30 minutes of inactivity
☑ Local administrators may refresh or disable management
☐ Set computer name to computer record name (Mac OS X v10.5 or later)
☑ Enable external accounts (Mac OS X v10.5 or later)
☑ Enable guest account (Mac OS X v10.5 or later)

Note the v10.5 or later tags

Note that you can force even local admins to be managed. This could work well in a school where politics overrode common sense and some users are allowed to be local admins of school owned computers, when those users have absolutely no clue what they are doing. (It was done because “they have to be able to install software” - right.) With this checkbox, that user can remain a local admin, will be able to install software; but you can insure that nothing else gets done to that computer. Just to be ‘in charge’, you could allow them to install software, but deny that software from running. Ouch.

3. Access

For now, we’ll leave the ‘Access’ settings in their default mode. This is the pane where you force local users to adhere to network management settings, allow or ignore nesting, combine workgroup settings, and allow or suppress the workgroup picker. The difference between nesting and combining is that you can create a managed group with preferences, then place that workgroup inside another workgroup. Nesting allows all values to be taken into effect when the compositor runs. Not allowing nesting is the normal behavior for Tiger clients. Combining takes all the workgroups at the same level and composites all of their settings together into a single experience. The Access Control List allows you to specify groups and/or users who are allowed/denied from logging into that computer group. If you have several workgroups, you can place the few that should have access to those clients; then only those group members will be able to log in, and only those workgroup settings will be combined.



Manage: ○ Never ○ Once ● Always

Access Control List

Name	Access	Short Name

Deny entries take precedence over Allow entries. Entry order is ignored.
User and Deny entries require Mac OS X v10.5 or later.

☑ Local-only users may login

Computer administrators may always log in.

☐ Local-only users use available workgroup settings (Mac OS X v10.4 or later)
☐ Ignore workgroup nesting (Mac OS X v10.5 or later)
☑ Combine available workgroup settings (Mac OS X v10.5 or later)
☐ Always show workgroup dialog during login (Mac OS X v10.5 or later)

The default settings for ‘Options’

4. Scripts

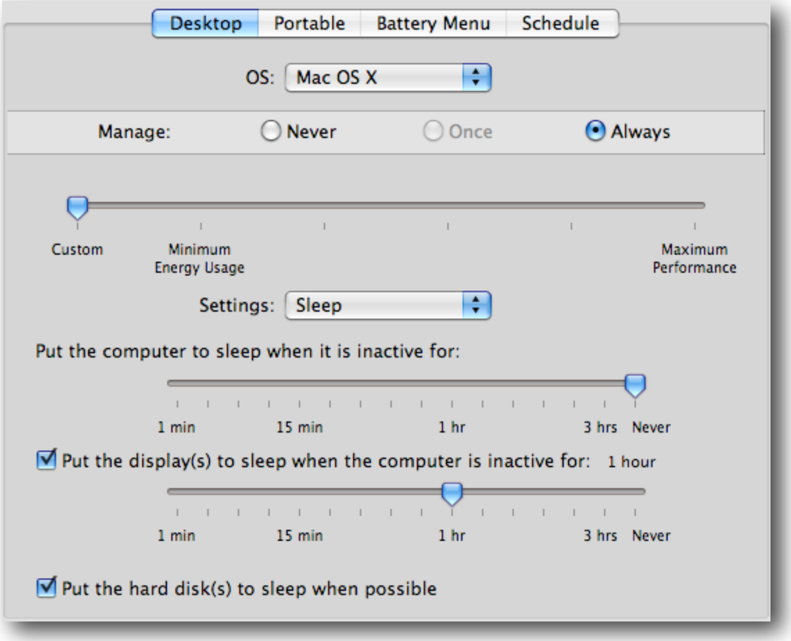
If you have a special action that needs to take place at login, or some housekeeping to do, you can use the ‘Scripts’ settings. The script gets stored inside the directory, so you do not need to copy it onto your image, and you can change it as often as you need. More info on this is in the official User Management guide.

5. Items

We’ll discuss this in the workflow/collaboration section. Here is where you add non-automount sharepoints that need to be loaded. It also works well for having an application, document or URL that needs to launch at login.

b. Energy Saver

This setting can only be done at computer or computer group - which actually makes sense. Energy Saver is one of the few MCX settings that takes effect at the login window (or at least is supposed to - there are bugs concerning the sleep activating…). The settings affect all users. One benefit of the settings is that you can force the client systems to stay awake so you can reach the systems with tools like Remote Desktop (ARD).



OS: Mac OS X

Manage: ○ Never ○ Once ● Always

Settings: Sleep

Put the computer to sleep when it is inactive for: Never

☑ Put the display(s) to sleep when the computer is inactive for: 1 hour

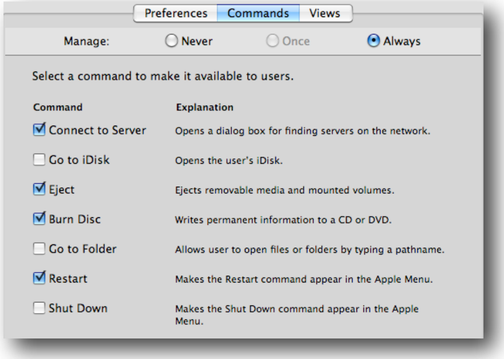
☑ Put the hard disk(s) to sleep when possible

Defining Energy Saver settings

If you use the 'Schedule' settings, try to stick to the sleep/wake setup versus the shutdown/boot setup. Be aware that if you set portables to sleep that it shuts off the Airport card - so reaching those systems afterward isn't easy.

Finder

The Finder, by default, is chaotic. No organization, nothing in order - enough to drive Lana Evitneter crazy. Some of the most important settings are being able to force new Finder windows, set which Finder commands are allowed, and put some order into the Desktop. For example, you can turn of the "Go to Folder" and "Shutdown" commands, stopping users from exploring places they shouldn't look (/etc anyone?) and requiring logouts versus accidental shutdowns.



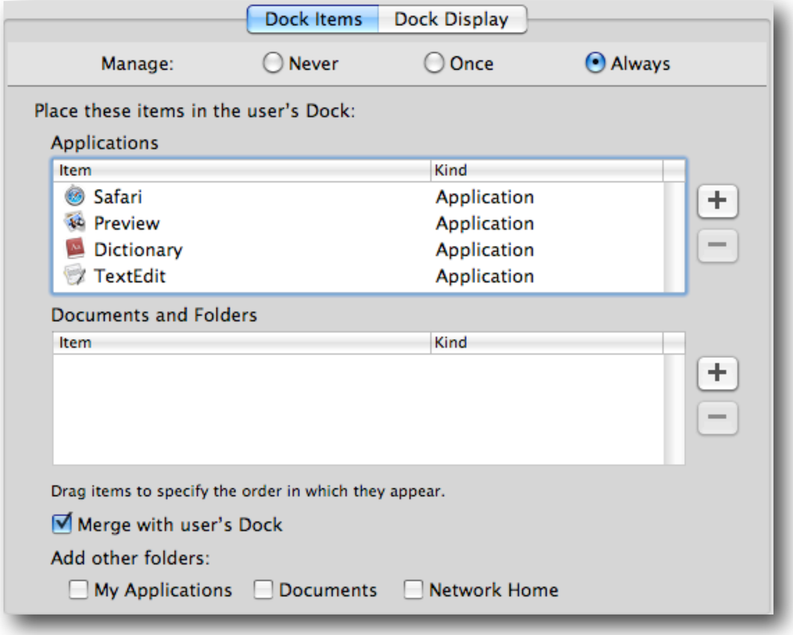
Manage: ○ Never ○ Once ● Always

Select a command to make it available to users.

Command	Explanation
☑ Connect to Server	Opens a dialog box for finding servers on the network.
□ Go to iDisk	Opens the user's iDisk.
☑ Eject	Ejects removable media and mounted volumes.
☑ Burn Disc	Writes permanent information to a CD or DVD.
□ Go to Folder	Allows user to open files or folders by typing a pathname.
☑ Restart	Makes the Restart command appear in the Apple Menu.
□ Shut Down	Makes the Shut Down command appear in the Apple Menu.

c. Dock

The Dock can get set at several levels. Globally, you can assign items to the Dock that all users need to have, yet also allow users to add/delete their own items.



Manage: ○ Never ○ Once ● Always

Place these items in the user's Dock:

Applications

Item	Kind
Safari	Application
Preview	Application
Dictionary	Application
TextEdit	Application

Documents and Folders

Item	Kind

Drag items to specify the order in which they appear.

☑ Merge with user's Dock

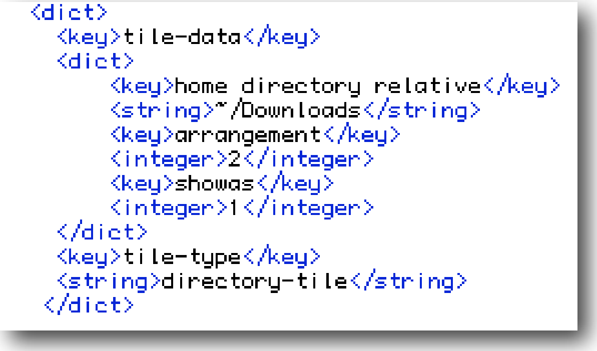
Add other folders:

□ My Applications □ Documents □ Network Home

Basic Dock settings with merge enabled

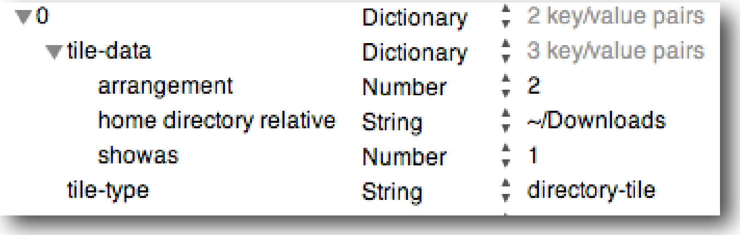
Note: if you set the Dock to merge with the user's settings, you'll also get the System defaults. Those values are stored inside the Dock application itself. Here's how to edit the defaults:

Log into a client system as admin or root. You can use TextWrangler (freeware) as admin, or use Property List Editor (from the DevTools) as root. Locate the 'default.plist' - /System/Library/CoreServices/Dock.app/Contents/Resources/English.lproj/default.plist - and open. TextWrangler works well to view the plist as xml,



```
<dict>
  <key>tile-data</key>
  <dict>
      <key>home directory relative</key>
      <string>~/Downloads</string>
      <key>arrangement</key>
      <integer>2</integer>
      <key>showas</key>
      <integer>1</integer>
  </dict>
  <key>tile-type</key>
  <string>directory-tile</string>
 </dict>
```

Property List Editor lets you edit the keys a little easier.



▼0	Dictionary	2 key/value pairs
▼tile-data	Dictionary	3 key/value pairs
arrangement	Number	2
home directory relative	String	~/Downloads
showas	Number	1
tile-type	String	directory-tile

Remove any of the items that you don't want to show up for every new user. Save the file and logout. The changes will stick until the next time the Dock gets updated.

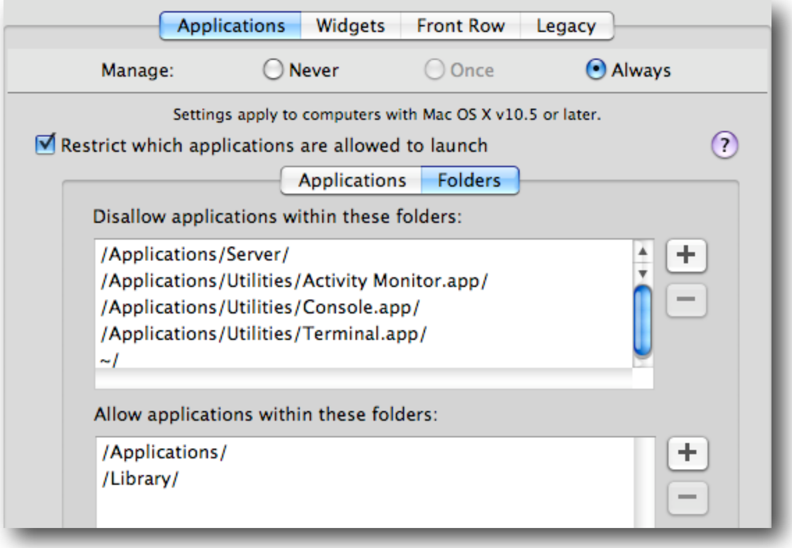
d. Applications

Management and control over applications has improved radically since Tiger. Whereas before, a user could drag an application into their home directory and alter it to bypass restrictions, and where schools had to totally disable Dashboard to stop users from running unapproved Widgets, Leopard MCX has much better control.

1. Applications themselves

There are two different settings for application management, one is the ability to digitally sign applications to keep them from becoming altered. While this is a great setting, it does not provide application restrictions. The setting is missing the ability to set "don't allow anything but signed apps to run" or something to that effect. This renders the signing ineffective as a control mechanism.

What works best for application management is path or folder restriction. You can set the locations where applications are allowed to run, and locations that are forbidden. Check the example:



Manage: ○ Never ○ Once ● Always

Settings apply to computers with Mac OS X v10.5 or later.

☑ Restrict which applications are allowed to launch

Disallow applications within these folders:

/Applications/Server/
/Applications/Utilities/Activity Monitor.app/
/Applications/Utilities/Console.app/
/Applications/Utilities/Terminal.app/
~/

Allow applications within these folders:

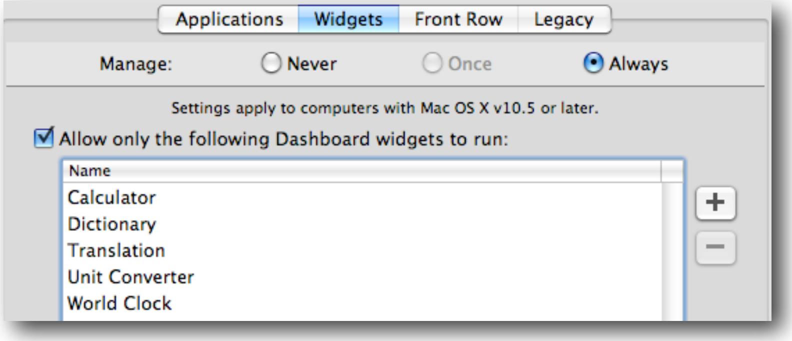
/Applications/
/Library/

Setting what locations a user can run apps from

The idea here is that the default Applications folder is safe, and the /Library folder often contains sub-launched apps needed by mainstream ones. The user does not have permission to mess with either of these folders. You could further restrict /Library by designating just the Application Support folder. I found that a few sys admin apps, such as Sassafras' K2 tools need to run as a user task from /Library itself. Note that the user's homedir is not allowed - so even if someone downloads an application to their home, they can't launch it. You could also deny use of Installer. While the dialog box says 'Folders', what it really means is the path. This means that you can add the path directly to an application as denied. The rules work like a firewall; so denies always win, and you can allow an app inside a denied folder - **except** - if you add the application to the 'Applications' pane, it can be launched anywhere, including from inside the user's home directory.

2. Widgets

The reaction from Education to Dashboard and Widgets was to turn them off. Not an optimal way to take advantage of the cool widgets out there. Here's a better way:



Manage: ○ Never ○ Once ● Always

Settings apply to computers with Mac OS X v10.5 or later.

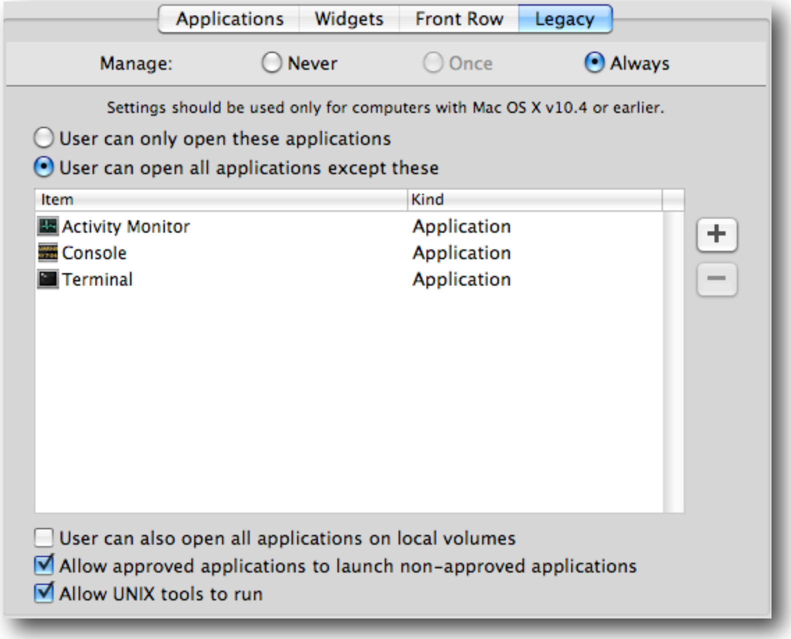
☑ Allow only the following Dashboard widgets to run:

Name
Calculator
Dictionary
Translation
Unit Converter
World Clock

Set the allowed widgets -all others are denied

3. Legacy

The controls here work as they did in Tiger. This is the only pane you can use to establish application restrictions for non-Leopard clients. See my earlier Tips and Tricks for info on setting Tiger restrictions.



Manage: ○ Never ○ Once ● Always

Settings should be used only for computers with Mac OS X v10.4 or earlier.

○ User can only open these applications

● User can open all applications except these

Item	Kind
Activity Monitor	Application
Console	Application
Terminal	Application

□ User can also open all applications on local volumes

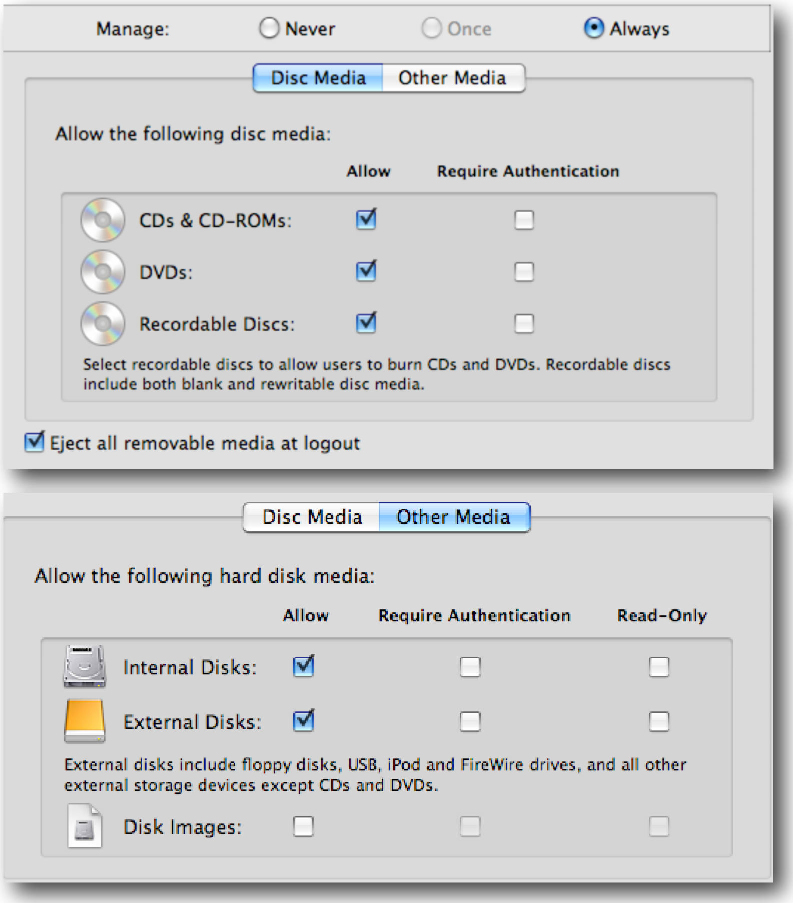
☑ Allow approved applications to launch non-approved applications

☑ Allow UNIX tools to run

Tiger and Panther application restrictions must be set here.

e. Media Access

This preference pane allows you to control access to r/w media such as CD-R's and recordable DVD's. You can also control access to external drives. My favorite item is the "Eject all removable media at logout". That setting alone should relieve the strain on many lab teachers who try to locate the two dozen clipart CD's handed out at the beginning of the class period.


Manage: ○ Never ○ Once ● Always

Disc Media | Other Media

Allow the following disc media:

	Allow	Require Authentication
CDs & CD-ROMs:	☑	□
DVDs:	☑	□
Recordable Discs:	☑	□

Select recordable discs to allow users to burn CDs and DVDs. Recordable discs include both blank and rewritable disc media.

☑ Eject all removable media at logout

Disc Media | Other Media

Allow the following hard disk media:

	Allow	Require Authentication	Read-Only
Internal Disks:	☑	□	□
External Disks:	☑	□	□

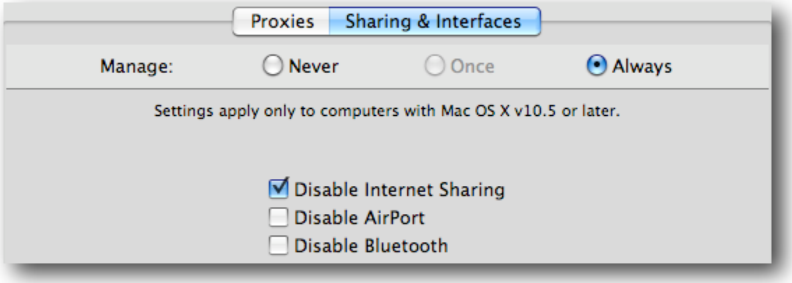
External disks include floppy disks, USB, iPod and FireWire drives, and all other external storage devices except CDs and DVDs.

Disk Images:	□	□	□

Controlling access to disk images, external media, etc.

f. Network

The Network settings under Leopard allow for proxy settings and controlling access to Internet Sharing, Airport, and Bluetooth. The settings that existed under Tiger to set the default web and mail apps is gone - since only Safari obeyed the settings. Be careful with the interfaces. Turning off Airport for a group of desktop lab systems may make sense; accidentally doing that to a group of portables may be counter-productive.


Proxies | Sharing & Interfaces

Manage: ○ Never ○ Once ● Always

Settings apply only to computers with Mac OS X v10.5 or later.

☑ Disable Internet Sharing
□ Disable AirPort
□ Disable Bluetooth

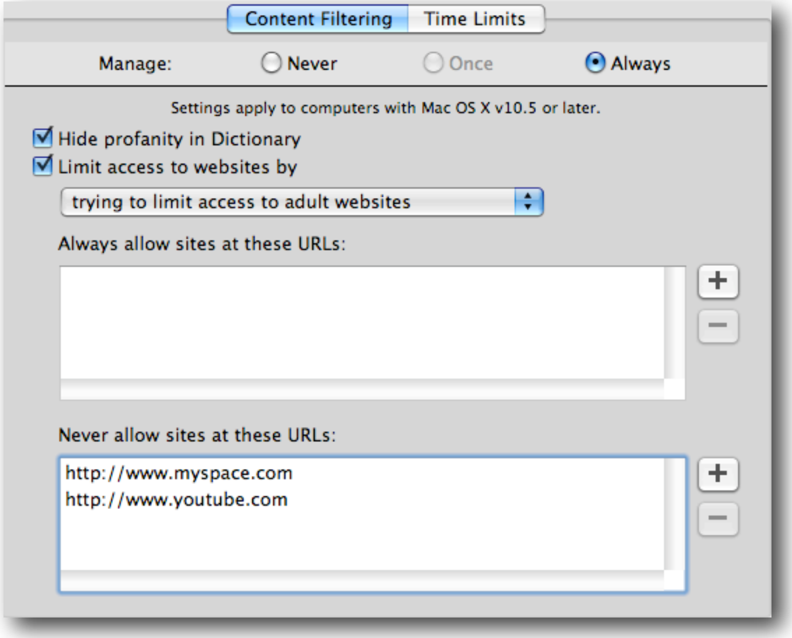
Setting global proxies and interface access.

What everyone needs to do is escalate the need for one more control here - "Disable Create Network in Airport" - just a thought.

g. Mobility is in Section 6 (User Accounts).

h. Parental Controls

New for Leopard is the **Parental Controls** settings. Think of Parental Controls as "comfort food" for that parent who asks "So, you're going to give my child a laptop. How do you intend on keeping her off the Internet at 2am?" Here are some of the key settings:



Manage: ○ Never ○ Once ● Always

Settings apply to computers with Mac OS X v10.5 or later.

☑ Hide profanity in Dictionary

☑ Limit access to websites by

trying to limit access to adult websites

Always allow sites at these URLs:

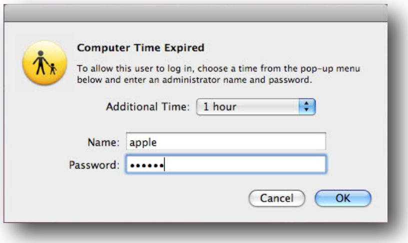
Never allow sites at these URLs:

http://www.myspace.com
http://www.youtube.com

Content filtering and white/blacklisting

The filtering run by Parental Controls is a local proxy which uses the same kind of logic we use for our spam filtering. It has a huge database of what are 'good' and what are 'bad' things to find on the Internet, and can determine what's acceptable with a high degree of certainty. It provides you with a mechanism to filter web content even when the users are outside the school network. Note: Don't bother putting www.apple.com into the whitelist - we hard coded it in there already. :-)

Setting time limits and curfews are the next most important items. You can specify a specificc amount of time a user is allowed on the computer on the weekend - this may be something the parents want in the AUP for a 1-1. Being able to set a curfew resolves the question asked previously about late night users. At the specified time, or if the usage limit has been reached, the user gets:



Computer Time Expired

To allow this user to log in, choose a time from the pop-up menu below and enter an administrator name and password.

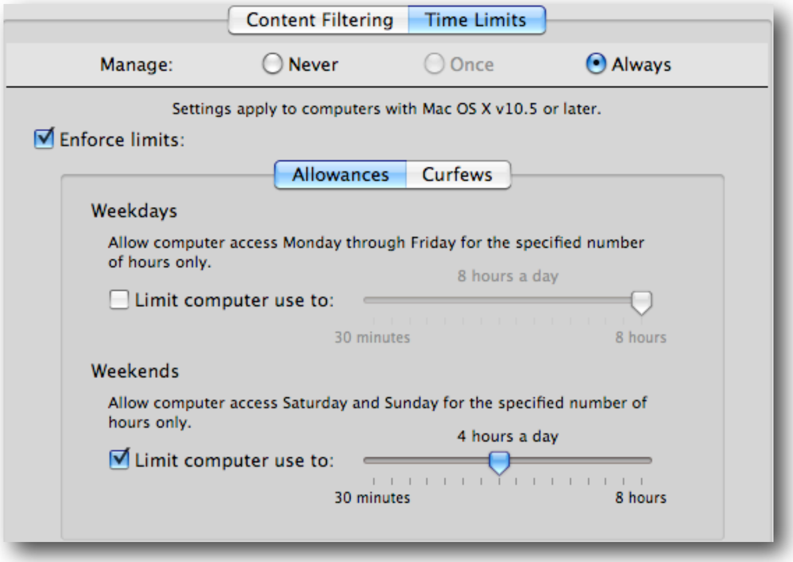
Additional Time: 1 hour

Name: apple

Password: ••••••

Time's up - Parental Controls

And yes, it takes a local admin account password to bypass the restriction. They do get a series of warnings as the expiration time approaches.



Content Filtering | Time Limits

Manage: ○ Never ○ Once ● Always

Settings apply to computers with Mac OS X v10.5 or later.

☑ Enforce limits:

Allowances | Curfews

Weekdays

Allow computer access Monday through Friday for the specified number of hours only.

8 hours a day

□ Limit computer use to:

30 minutes 8 hours

Weekends

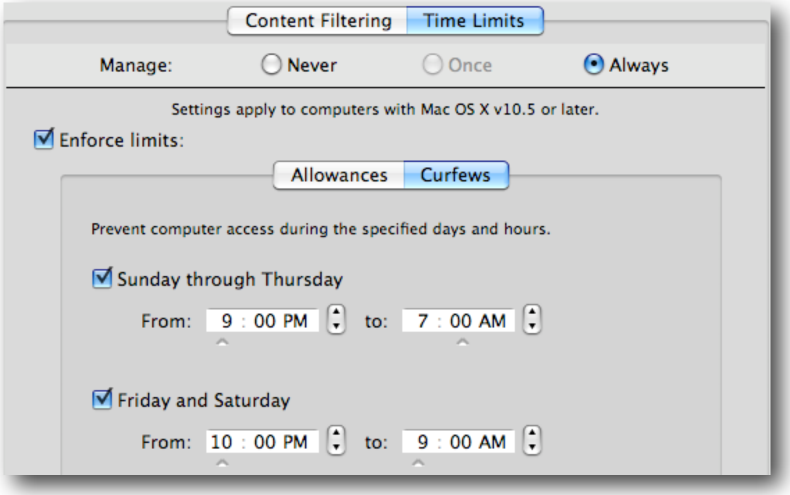
Allow computer access Saturday and Sunday for the specified number of hours only.

4 hours a day

☑ Limit computer use to:

30 minutes 8 hours

Setting a usage time limit



Content Filtering | Time Limits

Manage: ○ Never ○ Once ● Always

Settings apply to computers with Mac OS X v10.5 or later.

☑ Enforce limits:

Allowances | Curfews

Prevent computer access during the specified days and hours.

☑ Sunday through Thursday

From: 9 : 00 PM to: 7 : 00 AM

☑ Friday and Saturday

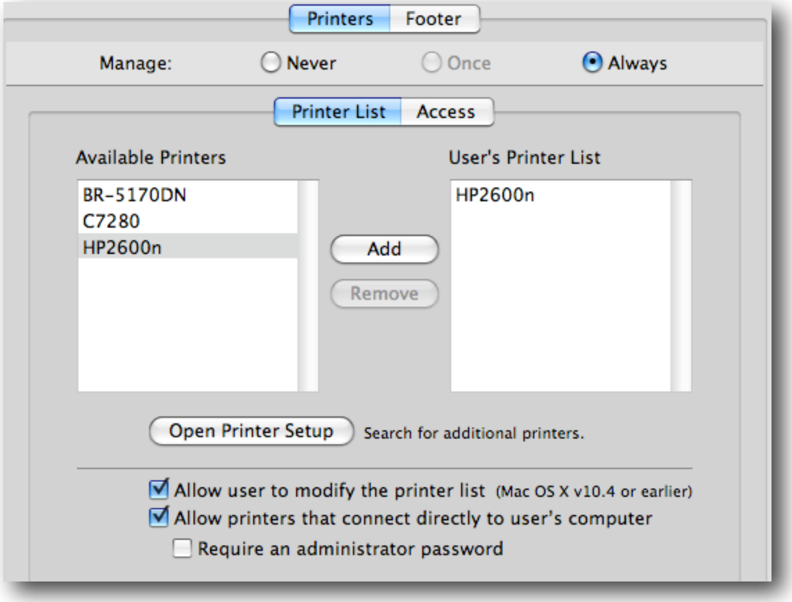
From: 10 : 00 PM to: 9 : 00 AM

Curfews for your 1-1 users

Being able to implement a series of distinct controls on deployed portables, and even on in-school systems is a good idea. You can set the lab computers to be unable to accept user logins after school, except for admin work.

i. Printing

The print management has improved somewhat since Tiger with the addition of a setting that had disappeared back with the end of Macintosh Manager. Running WGM from either an admin or client system, you will get a list of the printers captured by that system. If you are going to add many different printers from different areas, you will need to either run WGM from each of those locations, or capture all the printers on your admin/client system.



Manage: ○ Never ○ Once ● Always

Available Printers: BR-5170DN, C7280, HP2600n

User's Printer List: HP2600n

Search for additional printers.

☑ Allow user to modify the printer list (Mac OS X v10.4 or earlier)
☑ Allow printers that connect directly to user's computer
□ Require an administrator password

Assigning printers to managed clients

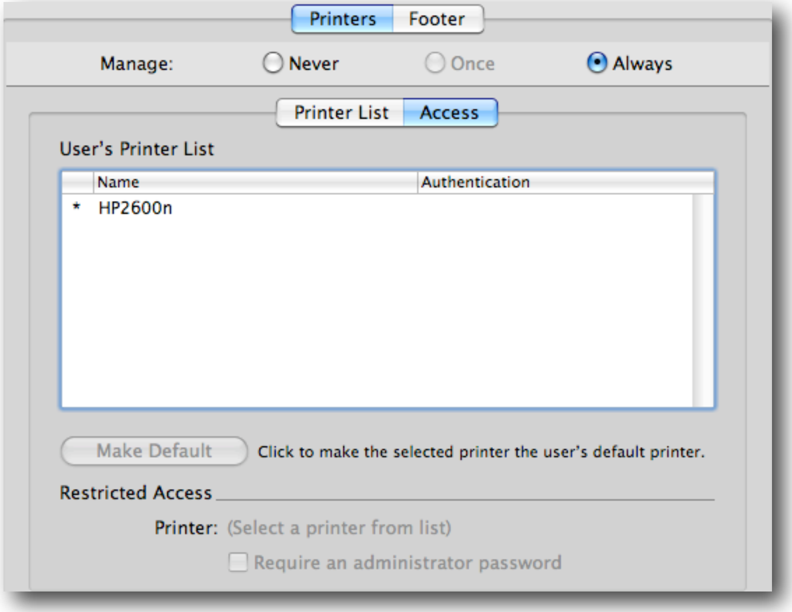
Note that the selection to allow users to modify the printer list applies to only 10.4 and **below**. This is due to a change in the Leopard printer system preferences to require local admin access to add/remove printers. Funny thing is, this came about because of all the schools screaming at us because the students and teachers kept adding printers all the time. Now those of you who didn't care about that then are now screaming. So… if you would like your users to be able to add their own printers, you can make a change to a file on the client system. To fix this for now, until we get it fixed in a future update, you need to locate the "/etc/cups/cupsd.conf" file on your admin system and open it with TextWrangler (or use terminal and your favorite editor). Locate the line:

```
# All administration operations require an administrator to
authenticate...
```

Change the following lines to:

```
<Limit CUPS-Add-Modify-Printer CUPS-Delete-Printer CUPS-Add-Modify-Class
CUPS-Delete-Class CUPS-Set-Default>

    # AuthType Default

    # Require user @SYSTEM

    Require valid-user

    Order deny,allow

</Limit>
```

Save your changes and restart the computer. You can then use ARD or any sys mgmt tool to deploy that new cupsd.conf file to all of your clients (or if you are ahead of the game, make it part of the image). The user still won't be able to add/remove printers in the system preference; but they will be able to add a printer when in the Print Dialog.



Manage: ○ Never ○ Once ● Always

User's Printer List

	Name	Authentication
*	HP2600n	

Click to make the selected printer the user's default printer.

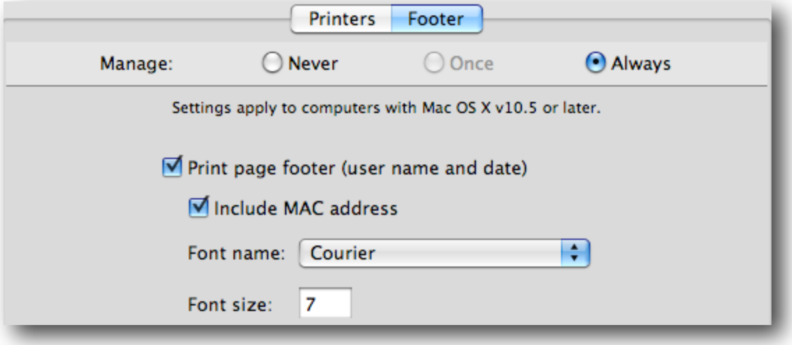
Restricted Access

Printer: (Select a printer from list)

□ Require an administrator password

Setting the default printer - plus authentication if needed

You can specify the default printer, and choose to require an admin password to print to a certain printer. Most importantly, to me anyhow, is that you can now specify that all print jobs will have a footer attached with the user's info and a timestamp.



Manage: ○ Never ○ Once ● Always

Settings apply to computers with Mac OS X v10.5 or later.

☑ Print page footer (user name and date)

☑ Include MAC address

Font name: Courier

Font size: 7

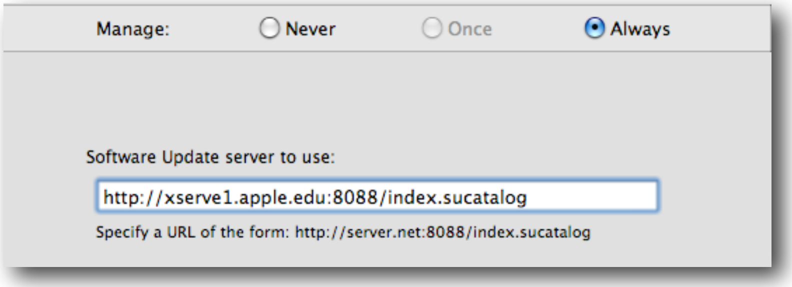
Adding a water-marked footer to print jobs

The footer is printed over top of anything else on the page. Since it is done as a watermark, if you have URL info, doc footers, etc. these will get printed over.

Final note on printing - any printers added by the users will show up for other users, so use caution allowing users to attach their own printers. The watermark is printed at the bottom left of the page at the extreme limit of the printer's page limit. This may or may not be below any user-defined footers. Finally, if you set a footer and the user prints a photo - it may not be what you wanted on the picture.

j. Software Update

The MCX setup for software update binds the clients to a specific software update server - hopefully inside your network.



Manage: ○ Never ○ Once ● Always

Software Update server to use:

http://xserve1.apple.edu:8088/index.sucatalog

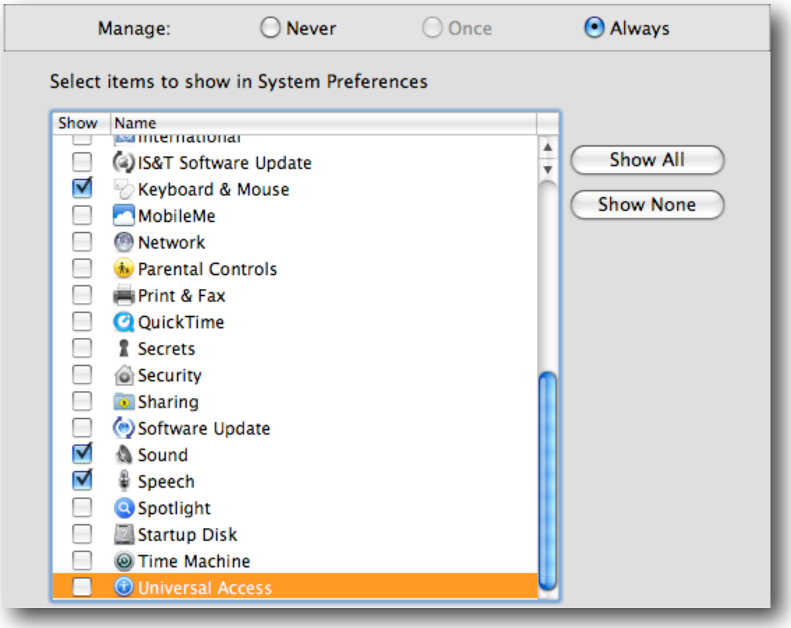
Specify a URL of the form: http://server.net:8088/index.sucatalog

Pointing your clients to an internal SuS

This works for servers as well as client systems.

k. System Preferences

Establishing access to specific system preferences has a couple of uses. First, you might want to just avoid the curiosity factor. While some sys prefs are locked to non-admin users, they can view the settings. Second, if you use MCX to populate some sys prefs settings, such as the Quicktime Pro license, you may not want users to see what the settings actually are - to avoid them 'borrowing' the license for use at home. Finally, even if you grant a user access to a sys pref, they may not be able to edit the item due to admin access requirements. That capability would make a great update.



Manage: ○ Never ○ Once ● Always

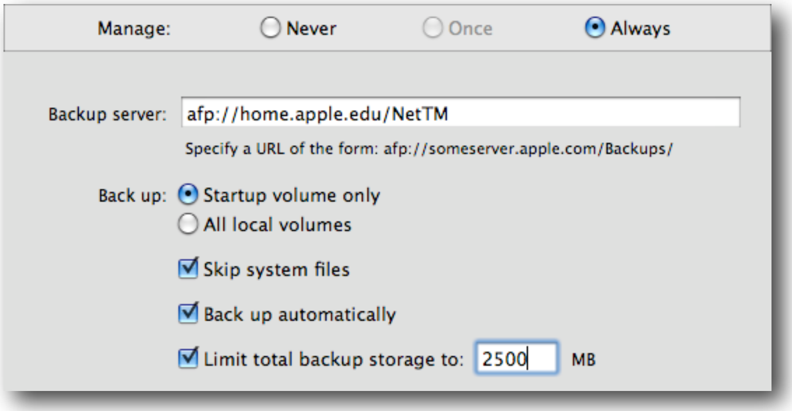
Select items to show in System Preferences

Show	Name
☐	International
☐	IS&T Software Update
☑	Keyboard & Mouse
☐	MobileMe
☐	Network
☐	Parental Controls
☐	Print & Fax
☐	QuickTime
☐	Secrets
☐	Security
☐	Sharing
☐	Software Update
☑	Sound
☑	Speech
☐	Spotlight
☐	Startup Disk
☐	Time Machine
☐	Universal Access

l. Time Machine

TM is a very interesting, very new capability in Leopard. As such, it is still undergoing a few growing pains. I am not going to recommend setting this up for large deployments yet. The reasons I am reluctant to do so are based on the behavior over the network of the backup mechanism. First, the disk image that gets created at the TM sharepoint is named after the MAC address of the client. This means that a user could end up with backups stored on several images as they change computers, making restoration very, very difficult. Second, while the TM backup can be set to ignore System files, the first backup still copies the entire Applications from the client system into that user's backup - every user on that client gets a copy of the Applications folder.

Until network TM is based entirely on the user account and not the computer, using Time Machine on a large deployment just isn't practical.


Manage: ○ Never ○ Once ● Always

Backup server: afp://home.apple.edu/NetTM

Specify a URL of the form: afp://someserver.apple.com/Backups/

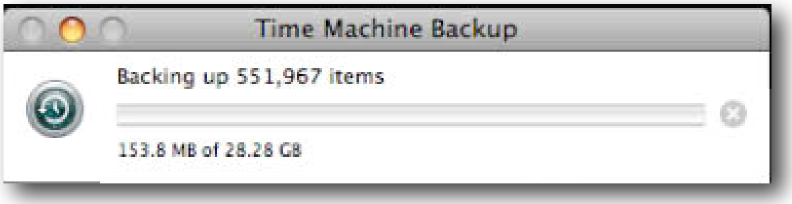
Back up: ● Startup volume only
○ All local volumes

☑ Skip system files

☑ Back up automatically

☑ Limit total backup storage to: 2500 MB

MCX settings for Time Machine


Time Machine Backup

Backing up 551,967 items

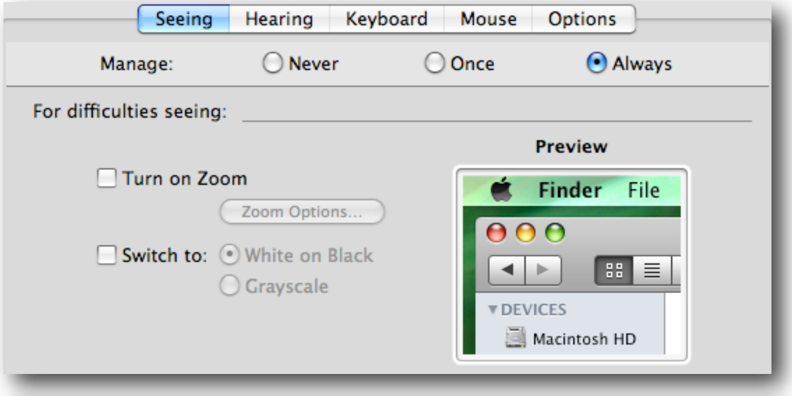
153.8 MB of 28.28 GB

Alpha's first backup - his home is not 29GB!

Note that in this example, Alpha's TM backup is trying to gather over half a million items. My test user has 88MB's of stuff in his home directory. The rest of that is the entire Applications folder being backed up. Nuf said.

m. Universal Access

In some circumstances, you may need to activate portions of the Universal Access sys prefs. Some external devices require the assistive device flag set. In a lab, you might want to set the screens to flash when a sysbeep is sounded. You can also turn on zoom as a default - however, under leopard, to zoom now all you need to do is hold down the Control key and use the scroll wheel or mouse dimple to zoom in/out.

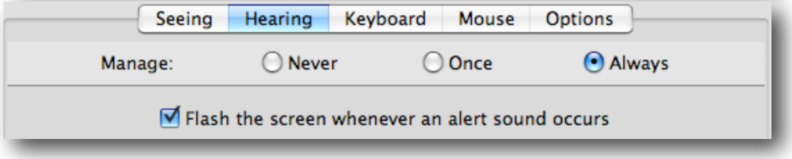


Manage: ○ Never ○ Once ● Always

For difficulties seeing:

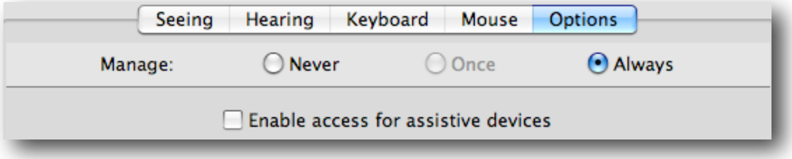
□ Turn on Zoom

□ Switch to: ● White on Black
○ Grayscale



Manage: ○ Never ○ Once ● Always

☑ Flash the screen whenever an alert sound occurs



Seeing | Hearing | Keyboard | Mouse | Options

Manage: ○ Never ○ Once ● Always

□ Enable access for assistive devices

Some of the Universal Access MCX settings

n. MCX in Action - an example of the hierarchy

Now that we've walked through the different MCX settings visible in the GUI, we can look at an example of these items in action. For this example, we are going to create a series of Dock values for two workgroups, a computer group, and two users. Then we'll display the difference each of the users gets at login. For grins, I'll also toss in a series of screenshots for a Tiger client for comparison.

1. The setup

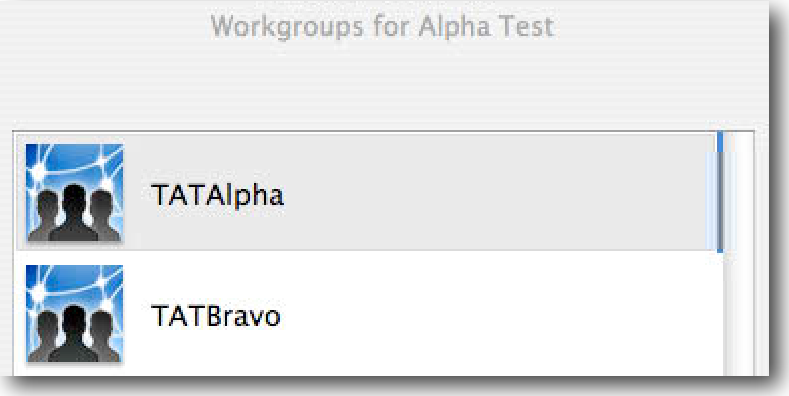
For this demo, we have the following values set for the Dock at each domain:

- Computer Group (Tips & Tricks) - Dock items are: Safari, Preview, Dictionary, and TextEdit. The user's settings will be merged. Dock items are set to 'Always'. Dock display is set to 'Once' on the bottom (for easier screenshots).
- Workgroup (TATAlpha) - Dock items are: Chess and Address Book. (Always)
- Workgroup (TATBravo) - Dock items are: Directory and Grapher. (Always)
- User (Alpha) - Dock item is iTunes. (Always)
- User (Bravo) - Dock item is Mail. (Once)

Note that the 'merge' setting will also populate the Dock with the system defaults, so we'll look at the Dock both before and after one of the users has dumped all the items they are allowed to remove.

2. The results

The Leopard clients won't get a workgroup picker because the combine flag was set when we set up the loginwindow (and it's on by default). The Tiger client, however, gets this:

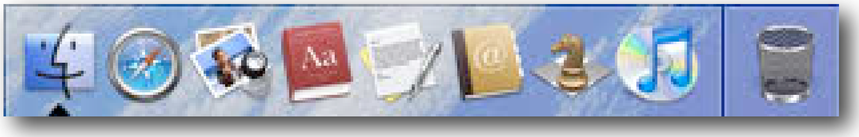


Workgroups for Alpha Test

TATAlpha

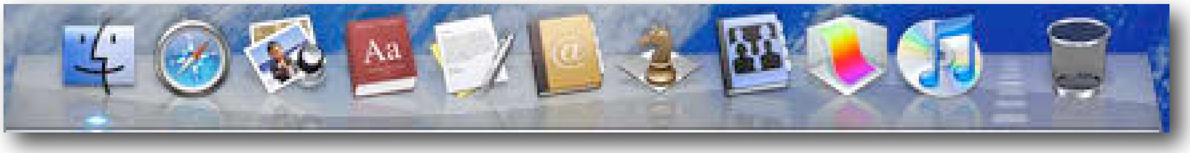
TATBravo

Because of this, the 'Alpha' user must choose one of the workgroups. His Dock will contain only the items set there from that workgroup as well as items from the computer group and user MCX settings.



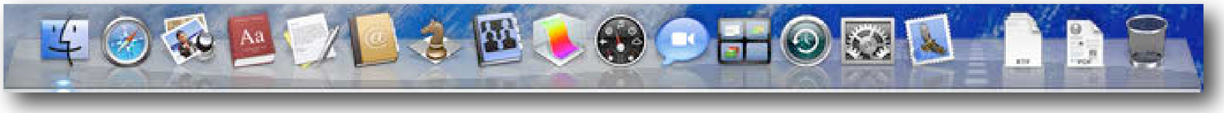
Tiger Dock for Alpha in the TATAlpha workgroup

When Alpha logs out and moves to a Leopard client, his first Dock looks like this:



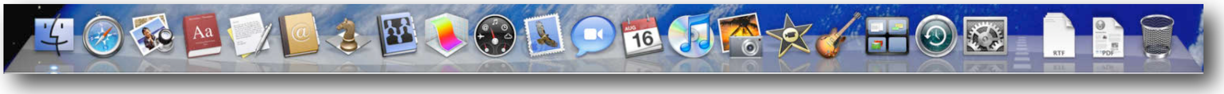
Alpha's Dock as a Leopard user - note that both workgroups' items are present

Note that the items are staged in the Dock from (left to right), Computer group, First workgroup alphabetically, second workgroup, then user. Items within the workgroup are staged alphabetically. What's also noticeable is that the System items are not present. They were set to merge with the user's items; but since the user account didn't have the merge turned on, they aren't here. If I let Bravo have the merge flag set, his initial Dock at login looks like this:



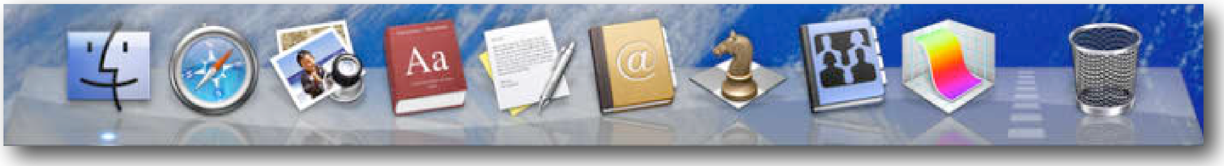
Bravo has the Dock items from the System also

Taking that to the extreme, if my image contains all the iLife apps too, then the initial Dock would look like this:



Bravo's full Dock with all system defaults

You can see where cleaning up the default Dock property list might be a really good idea. After tossing everything that Bravo can get rid of, the Dock is reduced to this:



These are the items set to 'Always' - all the 'Once' items have been 'poofed away'

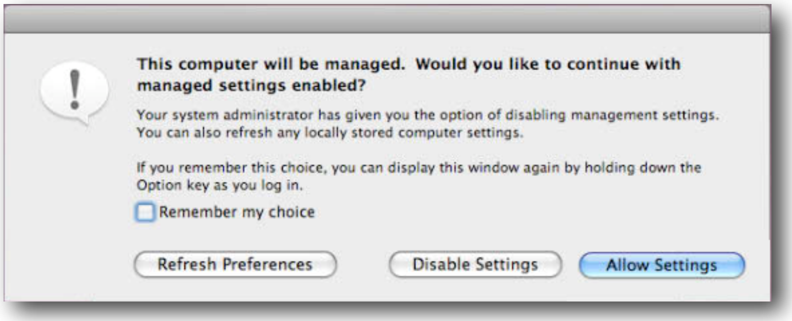
This is just one example of how the combining and MCX hierarchy works. Testing out your settings can save many hours of troubleshooting later on.

o. Admin tips for MCX

These are a few tips for admins when deploying the MCX settings.

1. Bypassing MCX settings

When you set up your managed client environment, you can choose to allow local admins to bypass MCX settings. You can also use that setting to refresh the MCX preferences if you don't want to wait for a restart or login/logout cycle. If the local admin holds down the 'Option' key at the loginwindow when clicking on the 'Log In' button, they get this:



This computer will be managed. Would you like to continue with managed settings enabled?

Your system administrator has given you the option of disabling management settings. You can also refresh any locally stored computer settings.

If you remember this choice, you can display this window again by holding down the Option key as you log in.

☐ Remember my choice

Admin bypass dialog

If you select to remember the setting, you're going to bypass MCX settings until the next time you hold down the 'Option' key at login. This is also the dialog you can use to refresh the management settings, which brings us to the discussion of the MCX cache…

2. MCX and cached settings

I almost left this section out; mostly because it gets so abused. Under Tiger, the sysadmin could set MCX cache values. No longer. The MCX system works by caching the directory settings onto the client inside the local directory in two places. The first is in a /Computers record in the local directory. You can view this directory by opening Terminal and typing in:

```
sudo dscl . -readall /Computers
```

Resulting in a data dump like this (just a portion):

```
johnd-imac-24:~ johnd$ sudo dscl . -readall /Computers

dsAttrTypeNative:cached_groups:

 <?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://
www.apple.com/DTDs/PropertyList-1.0.dtd">

<plist version="1.0">

<dict>

      <key>dsAttrTypeStandard:AppleMetaNodeLocation</key>

      <array>

            <string>/LDAPv3/xserve1.apple.edu</string>

      </array>

      <key>dsAttrTypeStandard:CopyTimestamp</key>
```

This info matches the information you can view in the System Profiler under the Managed Preferences tag. The rest of the locally cached information is stored inside the

‘Managed Preferences’ folder located at /Library/Managed Preferences. These values are updated on a very regular and consistent basis. The MCX cache is now refreshed:

- at boot time
- at each login
- at each logout
- at each restart
- at periodic intervals while sitting at the loginwindow
- every time there is a network transition (EN to Airport, etc)

The MCX cache is flagged as ‘dirty’ as soon as it comes into existence. This means that the system is always looking to refresh the cache at every possible circumstance. What this does not mean is that the cache is ever just deleted when the client cannot see the server. Users do not become magically ‘unmanaged’ just because they left the network. They definitely do not suddenly become admins. If you are experiencing problems with the MCX settings, and think that the cache is out of whack with what you set up, you need to perform a few basic troubleshooting tricks.

One - open up Workgroup Manager along side System Profiler on a client system. Compare the values listed for MCX/Managed Client carefully. What is getting to the client should exactly match what you set up.

Two - you can flush the cache and reboot the machine to force a complete rewrite with the following command: (if you use ARD, leave out ‘sudo’)

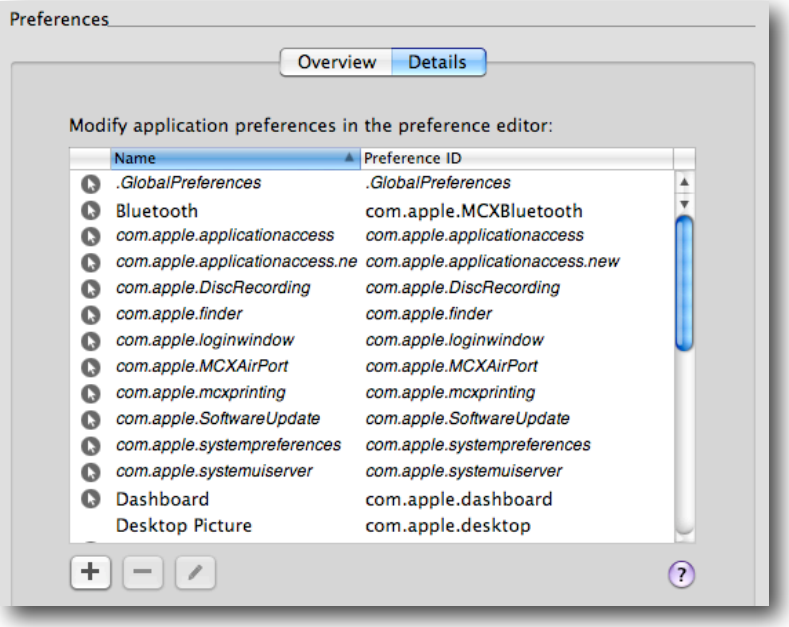
```
sudo dscl . -delete /Computers
sudo rm -rf /Library/Managed\ Preferences
```

Do **NOT** run these commands as part of a login script or you will definitely end up with unmanaged users. That all said, now we’ll move on to the really complex MCX stuff.

My brain really hurts now…

5. Advanced MCX setup - adding “Details”

While using Workgroup Manager’s GUI to edit MCX settings is relatively simple; there are many times you need a setting or value that “just ought to be there”. For that purpose, the MCX engineers gave us the ‘Details’ pane within WGM/Preferences.



Modify application preferences in the preference editor:

Name	Preference ID
.GlobalPreferences	.GlobalPreferences
Bluetooth	com.apple.MCXBluetooth
com.apple.applicationaccess	com.apple.applicationaccess
com.apple.applicationaccess.ne	com.apple.applicationaccess.new
com.apple.DiscRecording	com.apple.DiscRecording
com.apple.finder	com.apple.finder
com.apple.loginwindow	com.apple.loginwindow
com.apple.MCXAirPort	com.apple.MCXAirPort
com.apple.mcxprinting	com.apple.mcxprinting
com.apple.SoftwareUpdate	com.apple.SoftwareUpdate
com.apple.systempreferences	com.apple.systempreferences
com.apple.systemuiserver	com.apple.systemuiserver
Dashboard	com.apple.dashboard
Desktop Picture	com.apple.desktop

Getting into the Details of MCX

All the settings created under the GUI are visible here in property list format. What makes this section of the settings invaluable is that it provides access to settings that extend both currently available MCX GUI settings, or allow you to add settings that would never make it into the GUI.

a. Preference Manifests and other hidden settings

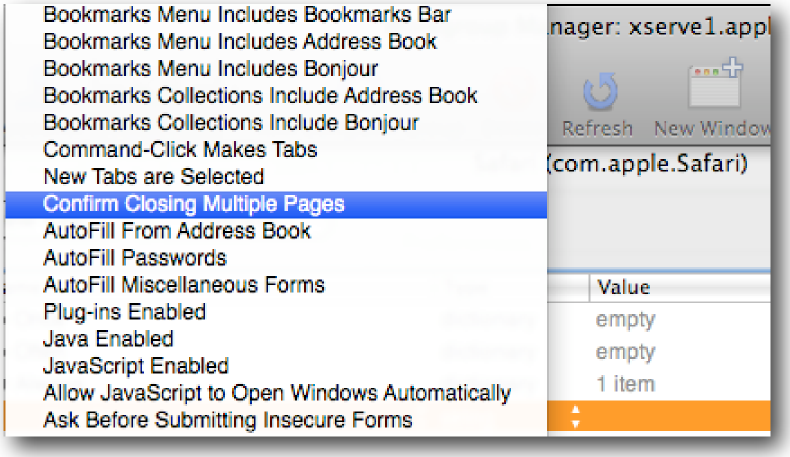
Many applications contain settings, or **keys**, that allow us to configure the user experience for ourselves. These settings can, in many cases, be stored in a directory so that they can be applied to many users. We have already looked at several of these settings in the previous section. Where MCX shines is in the ability to use the ‘other’ keys that many applications have outside of the system preferences we have already begun setting up. The best case for these keys is a **preference manifest** which is a database of values that the developer has imbedded into the application itself. This manifest lists the keys, and often their default values, that can be preset and are supported for inclusion in a directory.

Safari is a great example of the use of preference manifests. The application contains a large set of values that can be added to the MCX domain, and will then apply to any user set to be managed with that domain. Many preferences contain a set of keys that are known to work with MCX from a network. By adding the preference file to the MCX domain, the settings will apply across all users. Setting the mouse is a good example of this.

Finally, there are many 'tweaks' that we perform on a system to enhance the user experience, or to better manage the system. These settings are often altered with the "defaults" command. Many, if not most, of these type of adjustments can be made through the Details section versus running a script, or poking each machine individually with a command line call.

Here's a quick look at an example of each type of entry for Details:

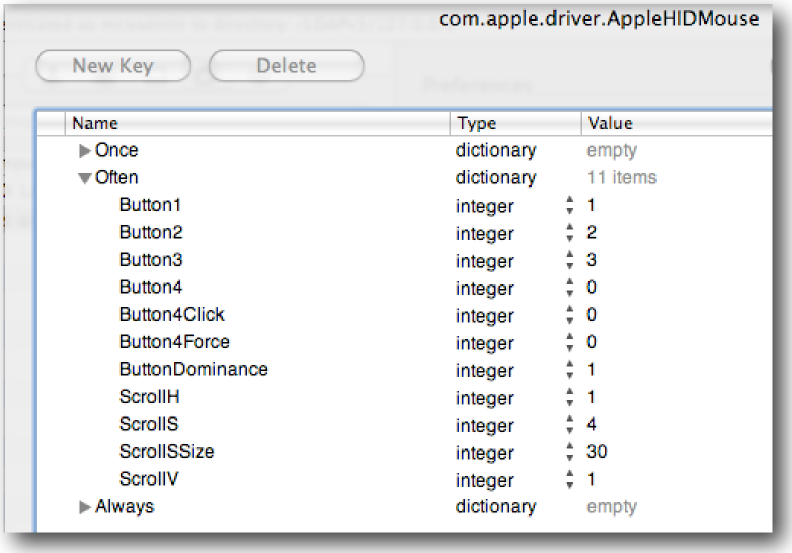
1. Preference Manifest - Safari



Bookmarks Menu Includes Bookmarks Bar
Bookmarks Menu Includes Address Book
Bookmarks Menu Includes Bonjour
Bookmarks Collections Include Address Book
Bookmarks Collections Include Bonjour
Command-Click Makes Tabs
New Tabs are Selected
Confirm Closing Multiple Pages
AutoFill From Address Book
AutoFill Passwords
AutoFill Miscellaneous Forms
Plug-ins Enabled
Java Enabled
JavaScript Enabled
Allow JavaScript to Open Windows Automatically
Ask Before Submitting Insecure Forms

Safari contains dozens of MCX aware keys in its manifest

2. Keys in a preference file - Mighty Mouse



Name	Type	Value
▶ Once	dictionary	empty
▼ Often	dictionary	11 items
Button1	integer	1
Button2	integer	2
Button3	integer	3
Button4	integer	0
Button4Click	integer	0
Button4Force	integer	0
ButtonDominance	integer	1
ScrollH	integer	1
ScrollS	integer	4
ScrollSSize	integer	30
ScrollV	integer	1
▶ Always	dictionary	empty

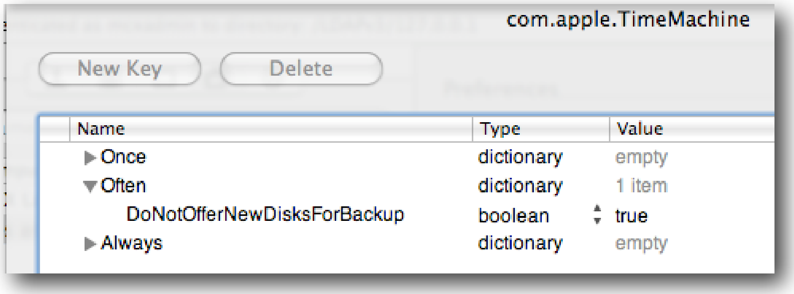
These keys allow the settings from the mouse to be default for all users

3. Turning a default into an MCX setting

Time Machine driving your users crazy? One solution would be to have each user run the following command:

```
defaults write com.apple.TimeMachine DoNotOfferNewDisksForBackup -bool YES
```

or… you could set the following information into a key in MCX/Details:



Name	Type	Value
▶ Once	dictionary	empty
▼ Often	dictionary	1 item
DoNotOfferNewDisksForBackup	boolean	true
▶ Always	dictionary	empty

Setting TM to stop bugging me at every disk mount

These were just a few examples. Now I'll go into as many tweaks as I can remember so you can learn more about the Details section.

b. Details tricks

There are two ways to add settings to Details. The first is to configure a preference the way you expect it to behave, then add that preference to the Details section. In order for this to work, the preference file must be in 'plist' or property list format. The second method is to locate the application itself and add it to the Details section. When this is done, the MCX code checks for the presence of a preference manifest first, then it looks for the preference file used by the application. It searches for the file inside the user's home directory in the path **~/Library/Preferences/<reverse domain>.<application>** (such as com.apple.Calculator).

The preference file may contain many items that are either unique to a specific user, or may be of no use in a managed setup. You can edit the imported plist to remove all but the keys you really plan to use.

A preference manifest looks empty at first glance. What is contains is the keys in a database, and to employ these keys you have add each key to the domain. Look at the iTunes or Safari examples to see how this is done.

Here are just a few examples of the cool MCX settings that can be done with Details:

1. Mousing around

The Mighty Mouse, by default, treats both sides of the front - the left/right buttons - as the left click location. If you set up a MM on your admin system, you can import those values into Details to allow all users to use the same settings. The values can be brought into 'Always' to force them, 'Often' to test them, and 'Once' to allow the user to change them as they see fit.

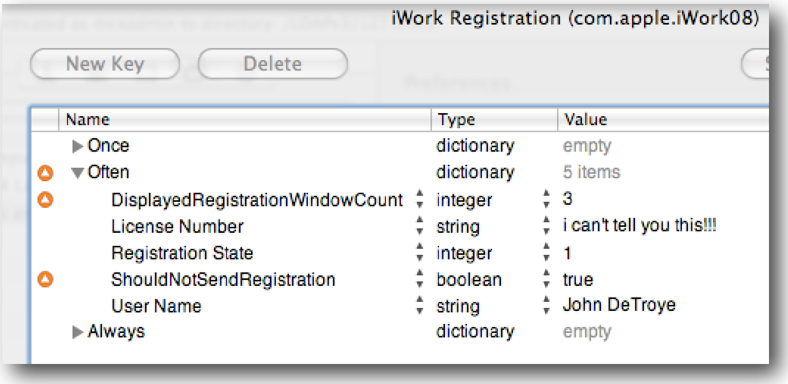
Set up the mouse sys prefs the way you want. In Details, select the "+" and locate "com.apple.driver.AppleHIDMouse". Select either the 'Always', 'Often', or 'Once' domain to add the entries. Test the settings by logging into a managed client as a non-admin user.

Note: You'll learn to use the different domains depending on the restriction you want to set; but you'll also find that some preference files only work when added to the 'Often' domain. Usually, these are preferences that have no user accessible settings.

2. iWork

If you are deploying iWork (and why wouldn't you?), you might find that entering all the different preferences and entering the serial number can become a bit much on lots of workstations. Using Details, you can grab all the settings at once and add them into your managed set. Note: The tutorial window will stiff pop up the first time the user launches on the apps; but not after that.

Set up iWork08 on your admin system. Make sure you launch each application at least three times to stop the tutorial windows from popping up at launch (unless you want your users to keep seeing them). Within Details, select the "+" and locate the "com.apple.iWork08.plist" inside /Library/Preferences. Add the item to the 'Often' domain.



Name	Type	Value
▶ Once	dictionary	empty
▼ Often	dictionary	5 items
DisplayedRegistrationWindowCount	integer	3
License Number	string	i can't tell you this!!!
Registration State	integer	1
ShouldNotSendRegistration	boolean	true
User Name	string	John DeTroye
▶ Always	dictionary	empty

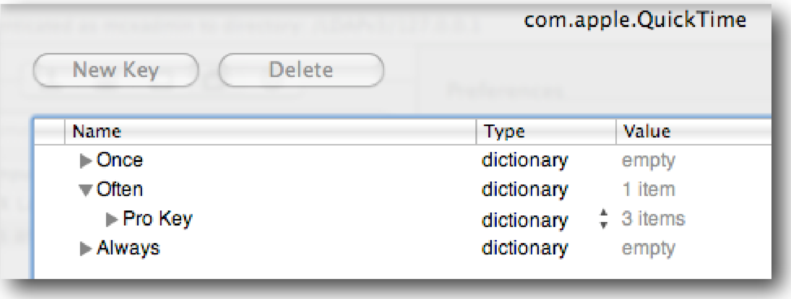
Make sure you own a site license for iWork before doing this!

3. Managed Client

In Leopard, the MCX team built in a huge number of default manifest items into the Managed Client application itself. So many, that I have dedicated section for this coming up later.

4. Quicktime

Instead of running around and setting the Quicktime Pro license on all of your systems, you can add the license to the computer group that has as members those systems you have licensed for QTPro. Just use Details to add the "com.apple.Quicktime.plist" to the 'Often' domain for your computer group.



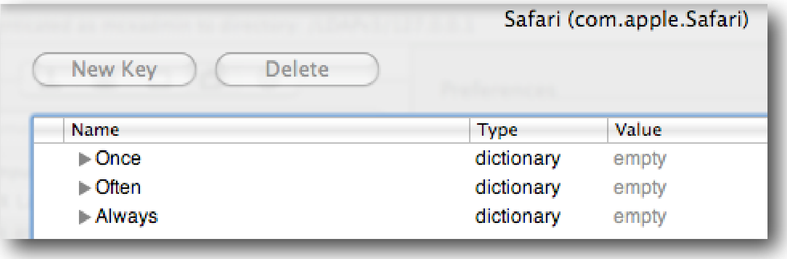
Name	Type	Value
▶ Once	dictionary	empty
▼ Often	dictionary	1 item
▶ Pro Key	dictionary	3 items
▶ Always	dictionary	empty

The QTPro key will automatically apply to all managed systems

Hint - make sure you hide the Quicktime system preference from prying eyes after this to avoid someone accidentally borrowing the serial number for their home use.

5. Safari

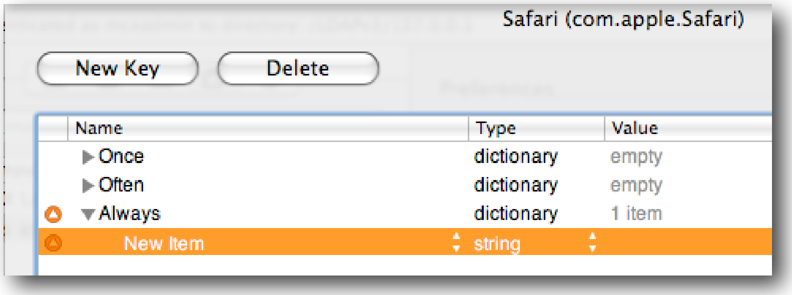
Safari contains the most extensive preference manifest of any Apple applications, followed by iTunes. Using Details, you can select the "+" and add Safari itself. In this case, you can turn off the "Import my preferences for this application" checkbox. If you didn't, the settings would contain all the extra stuff from your admin system's Safari preferences. This way, we start clean. The first thing you'll notice is that the domain looks empty:



Name	Type	Value
▶ Once	dictionary	empty
▶ Often	dictionary	empty
▶ Always	dictionary	empty

Only the manifest was imported - no values yet

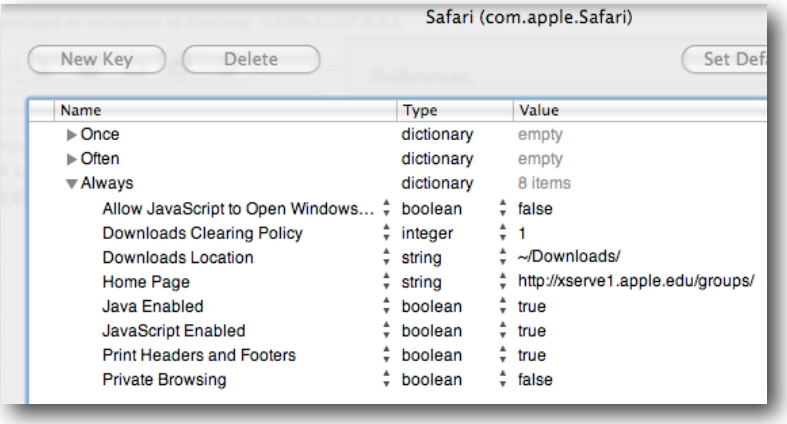
Then you begin adding **keys** to the settings. Start by selecting the 'Always' domain and turning down its subset triangle. With 'Always' highlighted, select the 'New Key' button.



Name	Type	Value
▶ Once	dictionary	empty
▶ Often	dictionary	empty
▼ Always	dictionary	1 item
New Item	string	

Adding a new key to the Safari settings

Next, click on the new item and you'll see a huge popup list of available keys. Select the 'Home Page' item near the top of the list. You can now edit the 'Value' filed to enter the default web site you want your managed users to get. Select the 'Always' domain again and add a few more keys. You'll note that you have a lot of control over Safari's behavior. Now if only Firefox had a preference manifest…

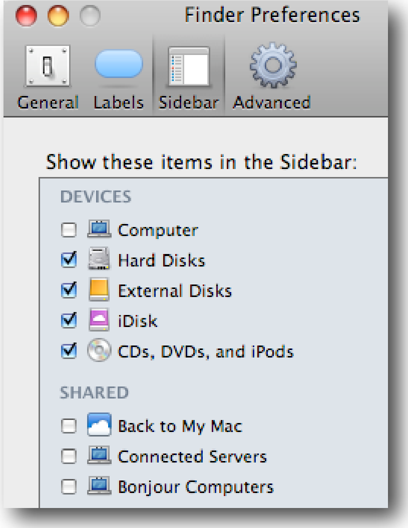


Name	Type	Value
▶ Once	dictionary	empty
▶ Often	dictionary	empty
▼ Always	dictionary	8 items
Allow JavaScript to Open Windows…	boolean	false
Downloads Clearing Policy	integer	1
Downloads Location	string	~/Downloads/
Home Page	string	http://xserve1.apple.edu/groups/
Java Enabled	boolean	true
JavaScript Enabled	boolean	true
Print Headers and Footers	boolean	true
Private Browsing	boolean	false

Note that private browsing is now denied

6. Sidebar

Many school sites have been asking how to suppress the “Shared” section of the Sidebar. If you have many sharepoints on the network, as well as all the clients with Remote Desktop agent active, you will see many entries in that section. To suppress the Sidebar ‘Shared’ section, you have start with your admin system’s Finder preferences. Select the Finder preferences on your admin system and turn off the checkboxes within the Sidebar settings.



Show these items in the Sidebar:

DEVICES

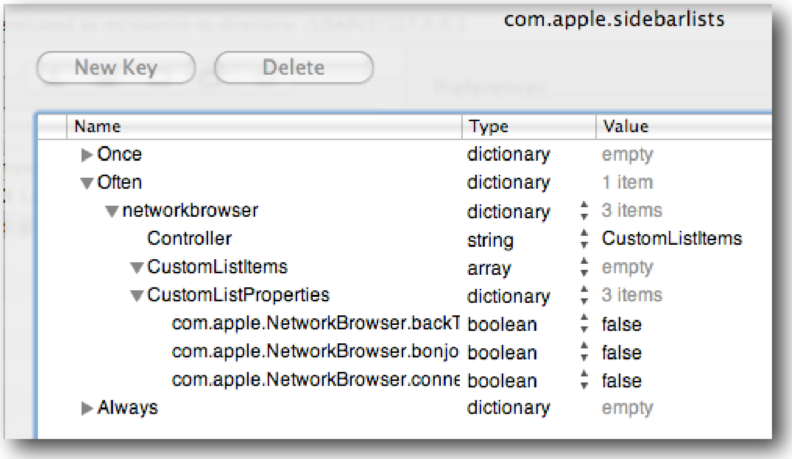
- ☐ Computer
- ☑ Hard Disks
- ☑ External Disks
- ☑ iDisk
- ☑ CDs, DVDs, and iPods

SHARED

- ☐ Back to My Mac
- ☐ Connected Servers
- ☐ Bonjour Computers

Turn off all Shared settings

Next, in Details, select the “+” and locate the “com.apple.sidebarlists.plist” inside ~/Library/Preferences. Add the item to the ‘Often’ domain, then select the edit tool (the pencil). Remove most of the items listed leaving this set:



Name	Type	Value
▶ Once	dictionary	empty
▼ Often	dictionary	1 item
▼ networkbrowser	dictionary	3 items
Controller	string	CustomListItems
▼ CustomListItems	array	empty
▼ CustomListProperties	dictionary	3 items
com.apple.NetworkBrowser.backT	boolean	false
com.apple.NetworkBrowser.bonjo	boolean	false
com.apple.NetworkBrowser.conne	boolean	false
▶ Always	dictionary	empty

Sidebar MCX settings to turn off Shared items

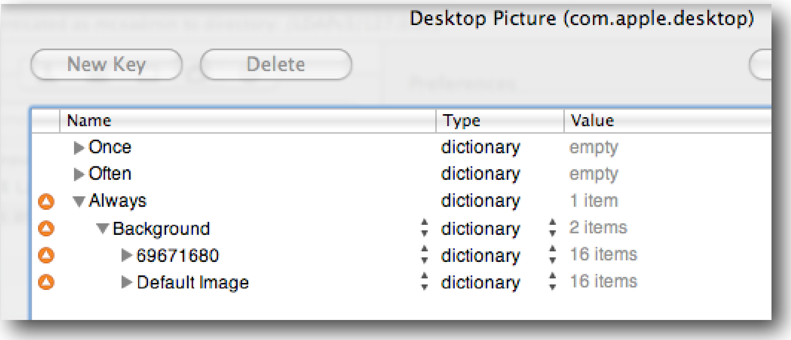
You can leave the saved searches, if desired, but the other items should be removed. If you end up with odd items in your client’s Sidebar, this is where to look for the problem. **Note:** The Sidebar isn’t very manageable, and if a user turns these items back on their own Finder preferences, the settings will not revert. Placing the settings into the ‘Always’ domain will not work.

7. Desktop Picture

In some cases, you may want to lock down a user's Desktop picture. While I don't have a problem with users displaying dumb pictures of their own, your site rules may not allow it. The problem has been that there are so many ways to set the Desktop picture, and under Tiger, it wouldn't stick. Details helps get past that.

First, if you are going to assign a default Desktop picture, make sure the image is stored on all clients in the same path. We're not going to store the actual picture in the directory, only the path to it. Hence, copy the image you want to use into the /Library/Desktop Pictures folder. Next, set it on your admin system.

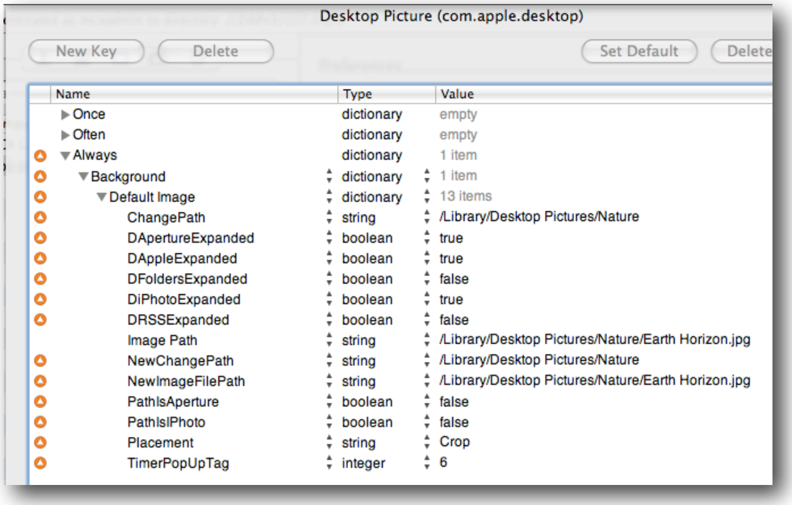
Now, in Details, select the "+" and locate the "com.apple.desktop" inside your ~/Library/Preferences folder. Add the file into the 'Always' domain (choosing the 'Often' domain would be cruel - they'll change it, and then it'll change back at every login…) Select the edit button and you'll see this:



	Name	Type	Value
	▶ Once	dictionary	empty
	▶ Often	dictionary	empty
⚠	▼ Always	dictionary	1 item
⚠	▼ Background	dictionary	2 items
⚠	▶ 69671680	dictionary	16 items
⚠	▶ Default Image	dictionary	16 items

Desktop plist with all custom settings intact

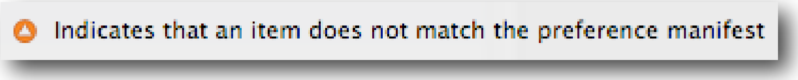
The extra settings you see are the ones belonging to every monitor and display configuration you've ever attached to your admin system. Delete all but the 'Default Image' key from the 'Background' key. The final setting will look like this:



	Name	Type	Value
	▶ Once	dictionary	empty
	▶ Often	dictionary	empty
⚠	▼ Always	dictionary	1 item
⚠	▼ Background	dictionary	1 item
⚠	▼ Default Image	dictionary	13 items
⚠	ChangePath	string	/Library/Desktop Pictures/Nature
⚠	DApertureExpanded	boolean	true
⚠	DAppleExpanded	boolean	true
⚠	DFoldersExpanded	boolean	false
⚠	DiPhotoExpanded	boolean	true
⚠	DRSSExpanded	boolean	false
	Image Path	string	/Library/Desktop Pictures/Nature/Earth Horizon.jpg
⚠	NewChangePath	string	/Library/Desktop Pictures/Nature
⚠	NewImageFilePath	string	/Library/Desktop Pictures/Nature/Earth Horizon.jpg
⚠	PathIsAperture	boolean	false
⚠	PathIsIPhoto	boolean	false
⚠	Placement	string	Crop
⚠	TimerPopUpTag	integer	6

The empty settings were deleted - the final looks like this

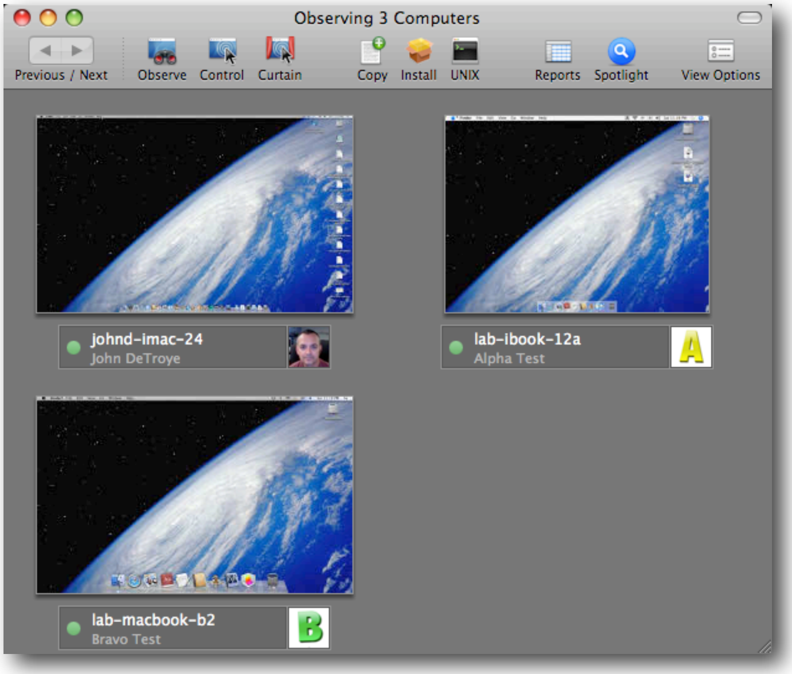
The rest of the items are flagged to show that they are not part of a manifest:



⚠ Indicates that an item does not match the preference manifest

You will see this orange triangle a lot as you edit preferences. It doesn't mean that the settings won't work; just that there is no manifest database item to correspond to that entry.

The user will not be able to change the Desktop picture at this point. They'll try; but beyond seeing the preview image in the sys pref change, nothing else will work. Even iPhoto will respect this setting in Leopard. All the user's Desktops will now look like this:



Consistent Desktop pictures

These are just some of the ideas where you can use Details to enhance the MCX experience. Test out the concept for any applications and/or plist files you have.

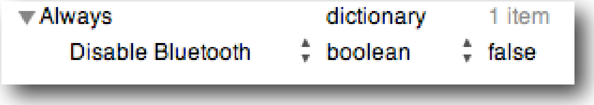
c. The Managed Client.app preference manifest

One of the most powerful portions of the Details setup in Leopard is hidden away by default. This is the preference manifest set built into the ManagedClient application itself. The ManagedClient app resides inside /System/Library/CoreServices and contains the code used to run the entire MCX process. The Compositor lives in here, along with the code to run mobile accounts, portable home directories, and much more. The MCX team built a huge preference manifest list in here that is still undergoing shakedown.

The manifest is made visible by going into Details, selecting the /System/Library/CoreServices/ManagedClient.app, and adding it to the set. The items show up as bold, grey items compared to the italicized items we already have. Some of the imports ask to replace ones we have already entered, such as Desktop. That's ok; the values we have already entered won't be disturbed. We will, however, be able to streamline some of our settings. Here's a walkthrough of the manifest items brought to us by the ManagedClient app:

1. Bluetooth

This setting contains only one key - ‘Disable Bluetooth’ (boolean)



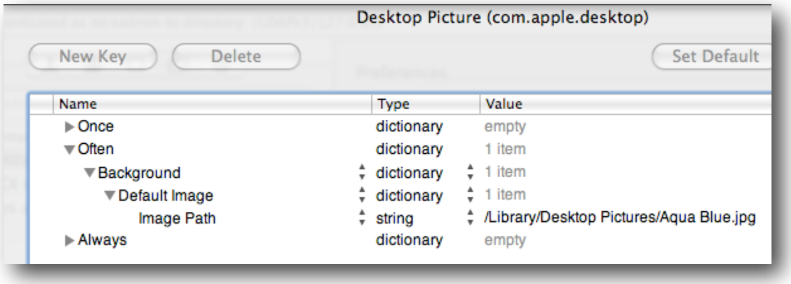
▼Always	dictionary	1 item
Disable Bluetooth	boolean	false

2. Dashboard

This manifest contains only one key - ‘Disable Dashboard’ (boolean) in respect for the Tiger systems that can’t use Widget management. If you did set Widget management for your Leopard systems, the values you set will show up here too.

3. Desktop Picture

This manifest contains keys to set the basic required info to establish a Desktop picture:



Desktop Picture (com.apple.desktop)

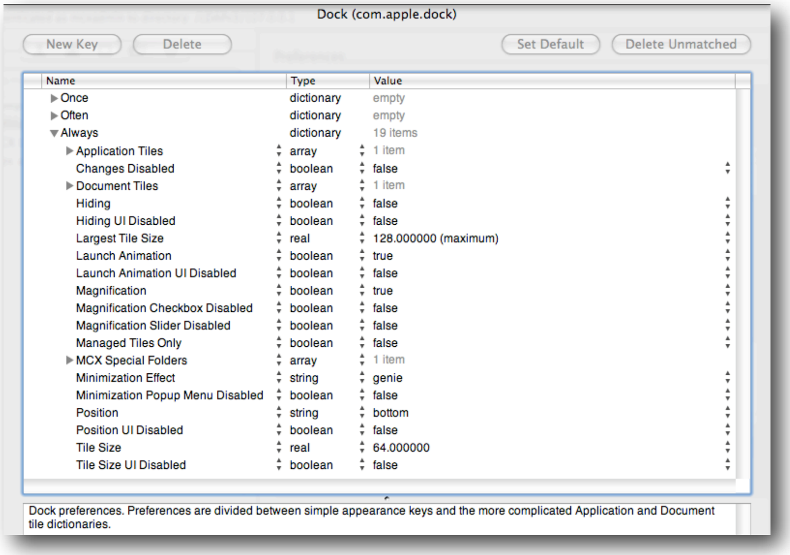
Name	Type	Value
▶Once	dictionary	empty
▼Often	dictionary	1 item
▼Background	dictionary	1 item
▼Default Image	dictionary	1 item
Image Path	string	/Library/Desktop Pictures/Aqua Blue.jpg
▶Always	dictionary	empty

‘Image Path’ is the only required item

Note that the manifest keys can only be created inside the ‘Often’ domain here. If you move them, or import them, in the ‘Always’ domain, they might not work. Test your settings often before locking them down.

4. Dock

The Dock manifest has a huge set of keys:



Dock (com.apple.dock)

Name	Type	Value
▶Once	dictionary	empty
▶Often	dictionary	empty
▼Always	dictionary	19 items
▶Application Tiles	array	1 item
Changes Disabled	boolean	false
▶Document Tiles	array	1 item
Hiding	boolean	false
Hiding UI Disabled	boolean	false
Largest Tile Size	real	128.000000 (maximum)
Launch Animation	boolean	true
Launch Animation UI Disabled	boolean	false
Magnification	boolean	true
Magnification Checkbox Disabled	boolean	false
Magnification Slider Disabled	boolean	false
Managed Tiles Only	boolean	false
▶MCX Special Folders	array	1 item
Minimization Effect	string	genie
Minimization Popup Menu Disabled	boolean	false
Position	string	bottom
Position UI Disabled	boolean	false
Tile Size	real	64.000000
Tile Size UI Disabled	boolean	false

Dock preferences. Preferences are divided between simple appearance keys and the more complicated Application and Document tile dictionaries.

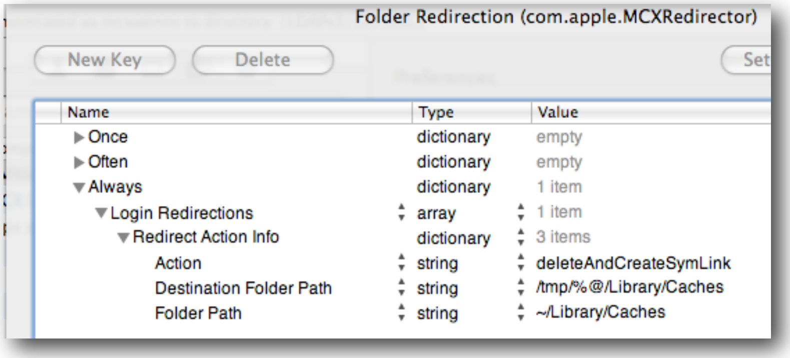
Note the definition window below the key settings

Always check the definitions of each key as you add it. Some will not need to be used, and if not, their default setting will take effect.

5. Folder Redirection

This setting will help take some of the performance load off of network home directory users. It is designed to force the current user's Cache folder into /tmp on the local machine. When you must use network home accounts, this can mean a huge reduction in network traffic. More on that in the user accounts section coming up.

Meanwhile, to set this up, you open the setting to edit, select the 'Always' domain and add a new key. Select 'Login Redirections' from the popup menu. Now select that new sub-key. Turn down the triangle, and add a new key to that key. This sub-key will be called 'Redirect Action Info'. When you open that key, you'll see that the defaults are already filled in for you:



Name	Type	Value
▶ Once	dictionary	empty
▶ Often	dictionary	empty
▼ Always	dictionary	1 item
▼ Login Redirections	array	1 item
▼ Redirect Action Info	dictionary	3 items
Action	string	deleteAndCreateSymLink
Destination Folder Path	string	/tmp/%@/Library/Caches
Folder Path	string	~/Library/Caches

The default login redirection item

The other actions are 'Logout Redirection' and 'Other Redirections'. One key point here is that you cannot create a redirection to force the user's home, or a sub-folder, to an upstream or network location. These redirections take place before any mount points are available, so the redirection would fail.

6. Home Sync

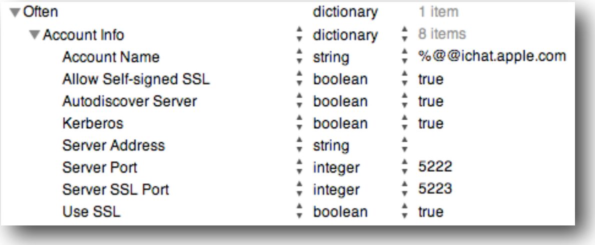
This setting gets its own discussion in the user accounts section.

7. iCal

The setting supports adding an 'Imported Accounts' key with values. It is designed to preset iCal accounts for users; but has had problems so far. Stay tuned for an update.

8. iChat

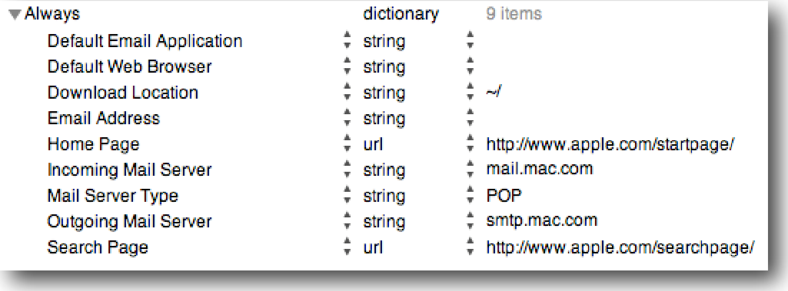
As with the iCal setting, the iChat setting is designed to preload account info for local iChat server users. Current behavior is twitchy.



▼ Often	dictionary	1 item
▼ Account Info	dictionary	8 items
Account Name	string	%@@ichat.apple.com
Allow Self-signed SSL	boolean	true
Autodiscover Server	boolean	true
Kerberos	boolean	true
Server Address	string	
Server Port	integer	5222
Server SSL Port	integer	5223
Use SSL	boolean	true

9. Internet Configuration

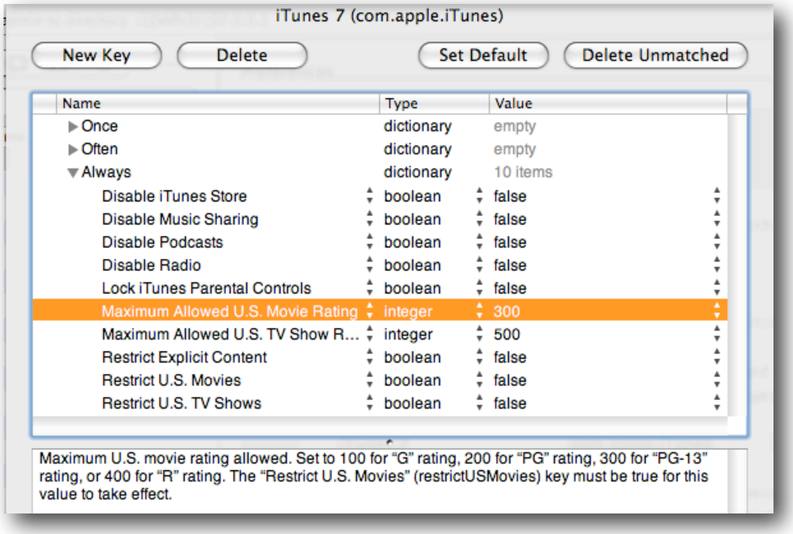
These are the settings and values that were removed from the GUI for Leopard, and will generally work only on Safari and Mail. Worth testing out.



▼Always	dictionary	9 items
Default Email Application	string	
Default Web Browser	string	
Download Location	string	~/
Email Address	string	
Home Page	url	http://www.apple.com/startpage/
Incoming Mail Server	string	mail.mac.com
Mail Server Type	string	POP
Outgoing Mail Server	string	smtp.mac.com
Search Page	url	http://www.apple.com/searchpage/

10. iTunes 7

These are the settings that allow granular control over the entire suite of settings in iTunes.



iTunes 7 (com.apple.iTunes)

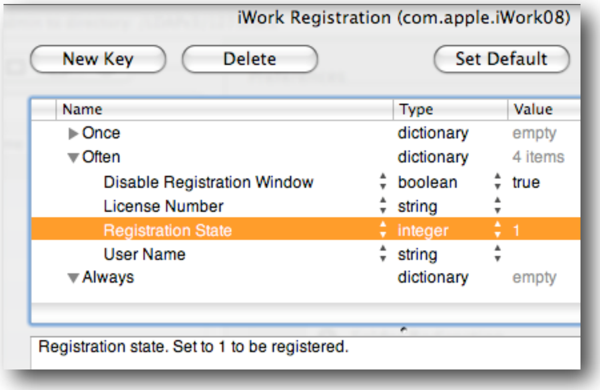
Name	Type	Value
▶Once	dictionary	empty
▶Often	dictionary	empty
▼Always	dictionary	10 items
Disable iTunes Store	boolean	false
Disable Music Sharing	boolean	false
Disable Podcasts	boolean	false
Disable Radio	boolean	false
Lock iTunes Parental Controls	boolean	false
Maximum Allowed U.S. Movie Rating	integer	300
Maximum Allowed U.S. TV Show R…	integer	500
Restrict Explicit Content	boolean	false
Restrict U.S. Movies	boolean	false
Restrict U.S. TV Shows	boolean	false

Maximum U.S. movie rating allowed. Set to 100 for "G" rating, 200 for "PG" rating, 300 for "PG-13" rating, or 400 for "R" rating. The "Restrict U.S. Movies" (restrictUSMovies) key must be true for this value to take effect.

Access to the iTunes Store, restricted movies, etc.

11. iWork Registration

These are the basic settings to establish the registration keys for network managed systems.



iWork Registration (com.apple.iWork08)

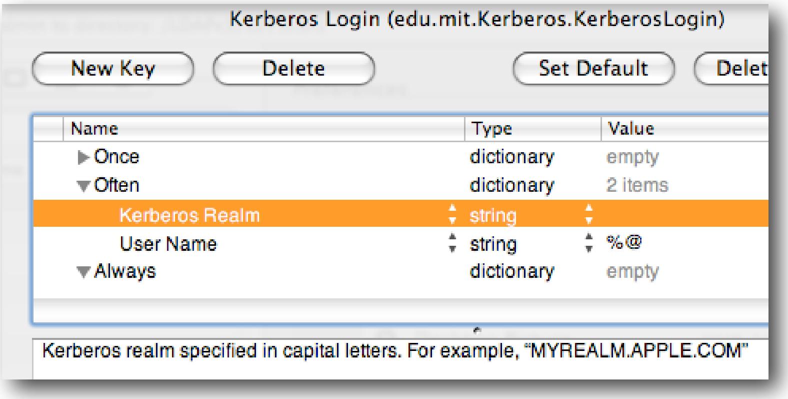
Name	Type	Value
▶Once	dictionary	empty
▼Often	dictionary	4 items
Disable Registration Window	boolean	true
License Number	string	
Registration State	integer	1
User Name	string	
▼Always	dictionary	empty

Registration state. Set to 1 to be registered.

Basic iWork registration info

12. Kerberos Login

Values to preset the Kerberos login (I have not tested this one at all).



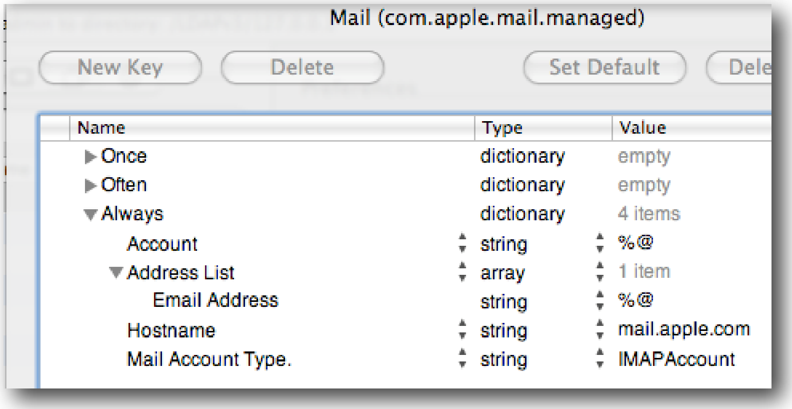
Name	Type	Value
▶ Once	dictionary	empty
▼ Often	dictionary	2 items
Kerberos Realm	string	
User Name	string	%@
▼ Always	dictionary	empty

Kerberos realm specified in capital letters. For example, "MYREALM.APPLE.COM"

The "%@" means the current logged in user

13. Mail

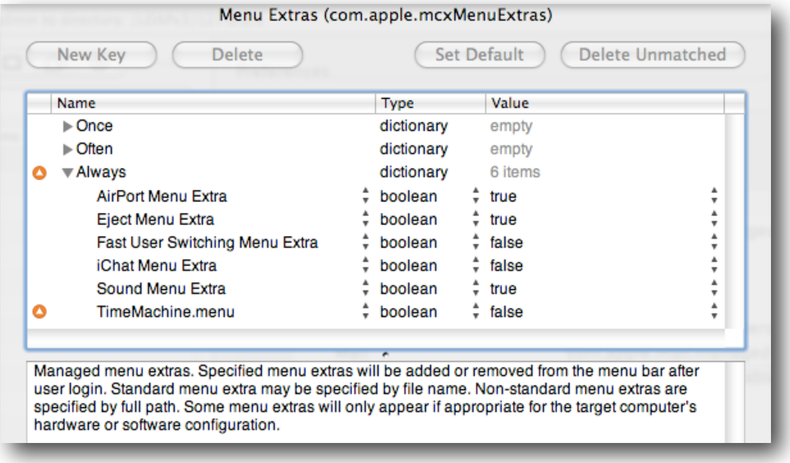
Preset values for user's Mail account.



Name	Type	Value
▶ Once	dictionary	empty
▶ Often	dictionary	empty
▼ Always	dictionary	4 items
Account	string	%@
▼ Address List	array	1 item
Email Address	string	%@
Hostname	string	mail.apple.com
Mail Account Type.	string	IMAPAccount

14. Menu Extras

This setting allows you to add menu items, or disable them. There are a lot of items in the manifest. For items not in the manifest, you can add the name or path, such as the Time Machine menu item shown here:



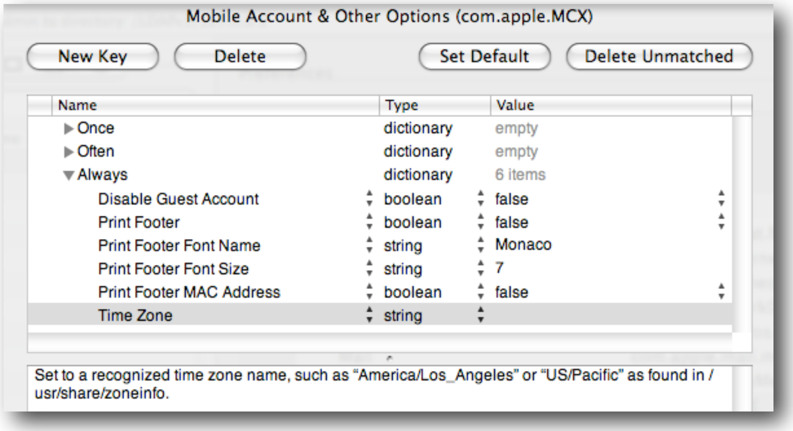
Name	Type	Value
▶ Once	dictionary	empty
▶ Often	dictionary	empty
▼ Always	dictionary	6 items
AirPort Menu Extra	boolean	true
Eject Menu Extra	boolean	true
Fast User Switching Menu Extra	boolean	false
iChat Menu Extra	boolean	false
Sound Menu Extra	boolean	true
TimeMachine.menu	boolean	false

Managed menu extras. Specified menu extras will be added or removed from the menu bar after user login. Standard menu extra may be specified by file name. Non-standard menu extras are specified by full path. Some menu extras will only appear if appropriate for the target computer's hardware or software configuration.

Menu Extras - note that TM item is disabled

15. Mobile Account & Other Options

The values here support mobility plus other items tossed in. The mobile items will be covered in section 6, the others are shown here:



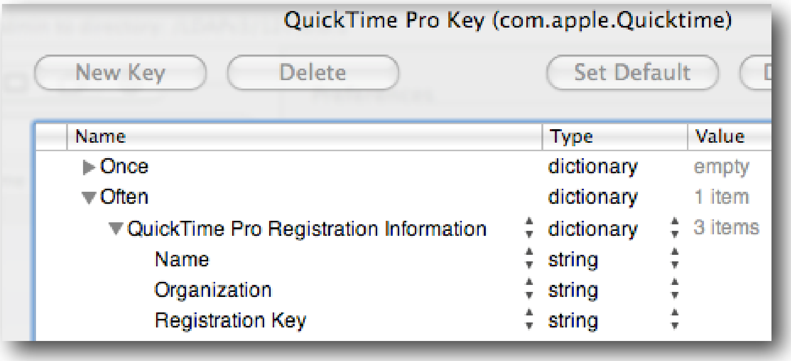
Name	Type	Value
▶ Once	dictionary	empty
▶ Often	dictionary	empty
▼ Always	dictionary	6 items
Disable Guest Account	boolean	false
Print Footer	boolean	false
Print Footer Font Name	string	Monaco
Print Footer Font Size	string	7
Print Footer MAC Address	boolean	false
Time Zone	string	

Set to a recognized time zone name, such as "America/Los_Angeles" or "US/Pacific" as found in /usr/share/zoneinfo.

Didn't like the choices for footer? Change them here.

16. QuickTime Pro Key

A simple way of adding the QTPro registration info from an admin station to a managed client set.



Name	Type	Value
▶ Once	dictionary	empty
▼ Often	dictionary	1 item
▼ QuickTime Pro Registration Information	dictionary	3 items
Name	string	
Organization	string	
Registration Key	string	

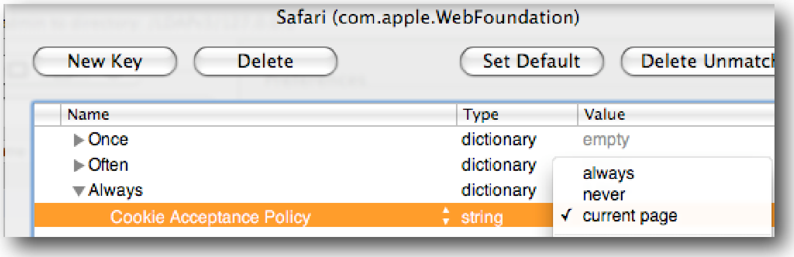
You can fill in the info yourself…

17. Safari

We've covered this already. It's a whole lot of information. Safari has a huge preference manifest.

18. Safari (WebFoundation)

This setting contains Safari's cookie policy.

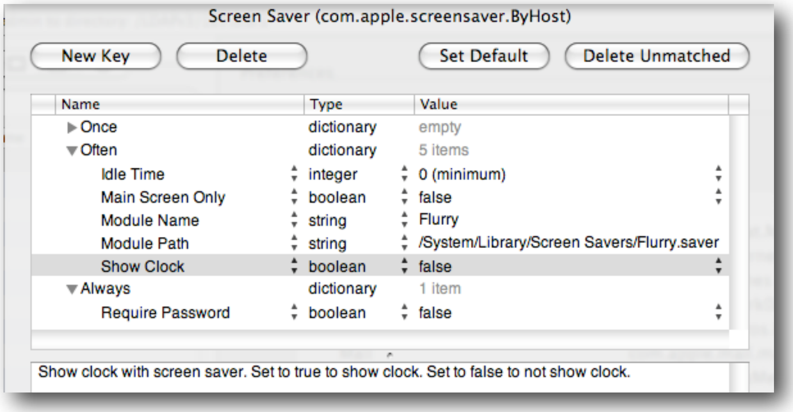


Name	Type	Value
▶ Once	dictionary	empty
▶ Often	dictionary	
▼ Always	dictionary	
Cookie Acceptance Policy	string	current page

Like cookies?

19. Screen Saver

Use this setting to establish a common screen saver for all managed clients.



Screen Saver (com.apple.screensaver.ByHost)

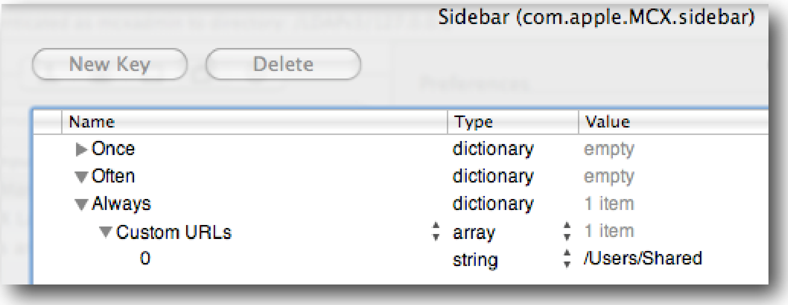
Name	Type	Value
▶ Once	dictionary	empty
▼ Often	dictionary	5 items
Idle Time	integer	0 (minimum)
Main Screen Only	boolean	false
Module Name	string	Flurry
Module Path	string	/System/Library/Screen Savers/Flurry.saver
Show Clock	boolean	false
▼ Always	dictionary	1 item
Require Password	boolean	false

Show clock with screen saver. Set to true to show clock. Set to false to not show clock.

Note that the password key shows up only in the 'Always' domain

20. Sidebar

This setting is for adding custom URLs to the Sidebar. It is not the same as the com.apple.sidebarlists settings.



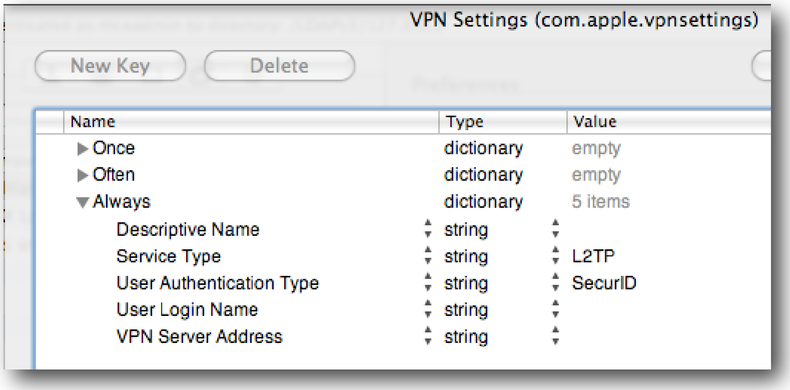
Sidebar (com.apple.MCX.sidebar)

Name	Type	Value
▶ Once	dictionary	empty
▼ Often	dictionary	empty
▼ Always	dictionary	1 item
▼ Custom URLs	array	1 item
0	string	/Users/Shared

Try adding a web site URL

21. VPN Settings

Settings to preload the VPN values.



VPN Settings (com.apple.vpnsettings)

Name	Type	Value
▶ Once	dictionary	empty
▶ Often	dictionary	empty
▼ Always	dictionary	5 items
Descriptive Name	string	
Service Type	string	L2TP
User Authentication Type	string	SecurID
User Login Name	string	
VPN Server Address	string	

Several of the settings here are still experimental. The MCX team added them to enhance management. In some cases, they aren't fully baked yet, so test your configs before deployment. Now on to user accounts, mobility, and the Portable Home Directory.

6. User accounts - MA's, PHD's, and other degrees of exertion

There are three classes of user accounts - local, network, and mobile. Under Leopard, information on these accounts is stored in either the local directory, on the network directory, or in both places at the same time. Along with these accounts are two locations for the user's home directory, on the network or on a locally attached drive, usually the client system's boot drive. Keeping the combination of the user's directory information and their home directory data functional is the challenge. Let's look at each class of account and where it is best employed.

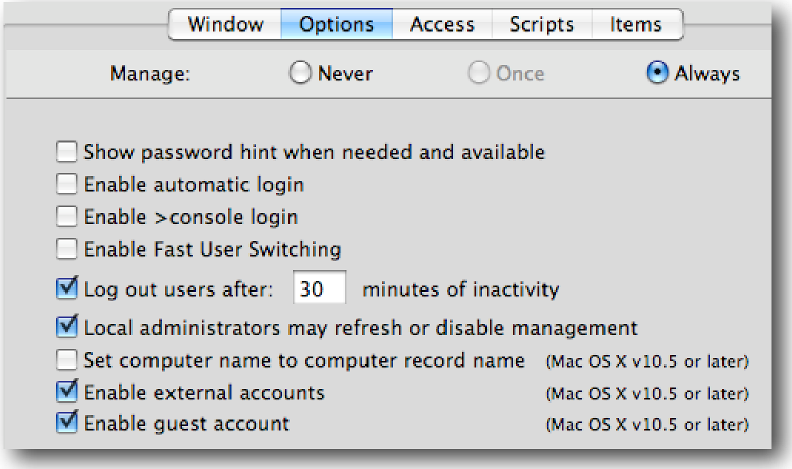
a. Local accounts

Local accounts are just that - user accounts established on a local directory. The home directory for this class of account is stored in a path that can be reached without moving across the network. Local accounts have the benefit of being able to function when there is no network connection, and are totally portable, in that the computer can be relocated without any adverse impact on the account. Three types of local accounts exist - guest, non-admin, and admin.

1. Guest account

New to Leopard is the 'Guest' account. It was created in response to the need for an anonymous user account that can be used in circumstances where user tracking and logging isn't needed. Good examples of this would be a locked down visitor kiosk, and kindergarten computer, or other systems that are configured in such a way that an anonymous user logging in would provide more benefit than possible harm.

In MCX, the Guest account is activated in the Login preference setting:

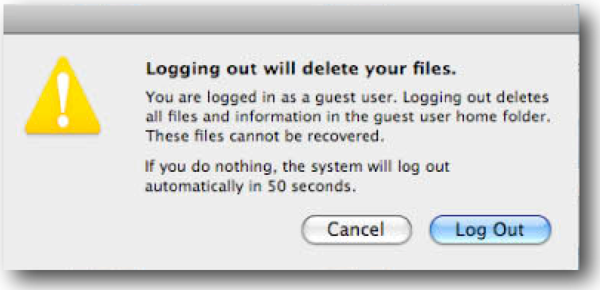


Manage: ○ Never ○ Once ● Always

□ Show password hint when needed and available
□ Enable automatic login
□ Enable >console login
□ Enable Fast User Switching
☑ Log out users after: 30 minutes of inactivity
☑ Local administrators may refresh or disable management
□ Set computer name to computer record name (Mac OS X v10.5 or later)
☑ Enable external accounts (Mac OS X v10.5 or later)
☑ Enable guest account (Mac OS X v10.5 or later)

Globally activating Guest account

The Guest account is only functional on Leopard clients; Tiger need not apply. When a user logs in as the Guest account, a complete home directory is created for them from scratch using the local home directory template. A key feature of the Guest account is its temporary nature. At logout, the contents of the Guest account's home directory are deleted. There is no way to go back into the system to recover anything accidentally left behind. If a Guest account is being used for work that needs to be saved, the user needs to either save to an external device, or mount a network sharepoint and copy the relevant data to the network.



Logging out will delete your files.

You are logged in as a guest user. Logging out deletes all files and information in the guest user home folder. These files cannot be recovered.

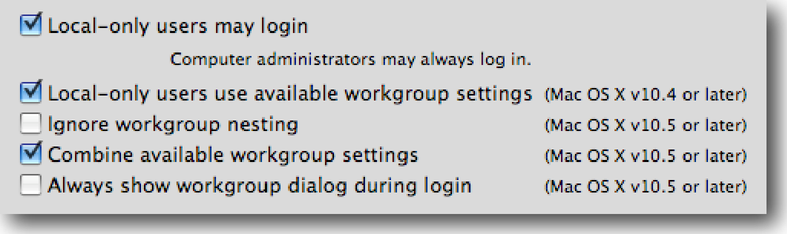
If you do nothing, the system will log out automatically in 50 seconds.

If you need your data, make sure you relocate it!

At the loginwindow, the Guest account either appears as a single icon labelled "Guest Account" or, in the name/password fields, you would type in "guest1" with no password. The Guest account is treated as any local account for management purposes.

2. Non-Admin account

On a client system, a non-admin account would be a local account with no administrative privileges. Examples of this are the generic 'student' and 'teacher' accounts, or the 'maclab-11' account to go with the portable labelled 'maclab-11' in the cart. This type of account is easy to set up on one to a few systems; but rapidly becomes an administrative hassle when trying to deploy and manage across hundreds, or even thousands, of client machines. MCX settings for this type of account is usually done at computer level, and can use workgroup settings if the flag is set to force local users to respond to them.



☑ Local-only users may login

Computer administrators may always log in.

☑ Local-only users use available workgroup settings (Mac OS X v10.4 or later)

□ Ignore workgroup nesting (Mac OS X v10.5 or later)

☑ Combine available workgroup settings (Mac OS X v10.5 or later)

□ Always show workgroup dialog during login (Mac OS X v10.5 or later)

Local user management set in Login/Access

Local user accounts are best used when network logging and tracking of users is not required, and/or the client systems are often removed from the network.

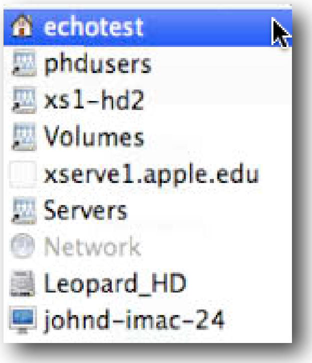
3. Admin account

A local administrative account is pretty much required on current systems. The account can be hidden (contains a UID less than 500), or just inaccessible to non-admin users. The local admin account is used to set up the ARD agent, or any other systems management tool, plus it is used for local maintenance. In MCX, the local admin account can bypass MCX settings, normally, by holding down the 'Option' key when logging in. The directory administrator can also force local admins to respond to MCX settings in the Login/Options pane. While under Tiger you could add a local non-admin account to the network admin group, essentially making that user a local admin; no such capability exists in Leopard.

b. Network Accounts

Network home directory accounts were created under Mac OS 9 for Macintosh Manager. The concept was for a network user to be able to log into a workstation, yet maintain their Documents on a network sharepoint. All the user's preferences were copied to the local machine, and all the work the user did was stored in this single folder on the net. Under Mac OS X, things got complicated. Now the user's entire existence is stored in a network home directory - a folder containing not just the user's documents, but their preferences, music, movies, web caches, and everything else.

While it is possible to tweak the network account setting to force the network user account's home directory to exist on a local drive, we're not going there this way. That would be better served by using the mobile account setup in the next section. A network user account consists of a network directory entry containing the user's identity, password, and any unique contact and management information for that account. The directory entry also includes the location information for that user's home directory. Under Leopard (and Tiger), this home directory usually lives on an automount sharepoint on an OS X server. It can live on any network sharepoint using AFP, SMB, or NFS to provide the network filesharing.



Network home location

Network accounts are extremely easy to set up and manage. All account information resides in a network directory, management settings and home directory also reside on the network. The user account can, theoretically, log into any workstation bound to the directory server and be working inside their home directory at any time. Changes made to the user account info, MCX settings, or home directory are immediately available to the user at next login. Another advantage is that the account could be used to back up a local Guest or generic account. The user logs in as Guest or 'Student', then mounts their network home directory to save items that they create locally. The downside here is that users often get into the habit of just opening the items from the network store; bringing down the problems we encounter using pure network accounts.

Unfortunately, there is a saying "In theory, theory and practice are identical; in practice, they are not". We (Apple) do not support network home directory accounts on networks less than 100Mb/switched/wired. Note that this does not include over Airport. Many applications behave very badly with network homes, creating massive amounts of traffic to/from the home directory server just for basic needs. For example, just launching a web browser creates open log files and cache files with immense traffic before the user even gets to their first web site. Using iLife over the network - imagine trying to perform a movie capture…

Best use for network accounts is in high-speed networks with low demand on the user's home directory. When network accounts are required; you should activate the "Folder Redirection" setting in MCX to keep the cache files local. While that may not totally resolve application traffic problems, it should reduce them significantly.

c. Mobile Accounts

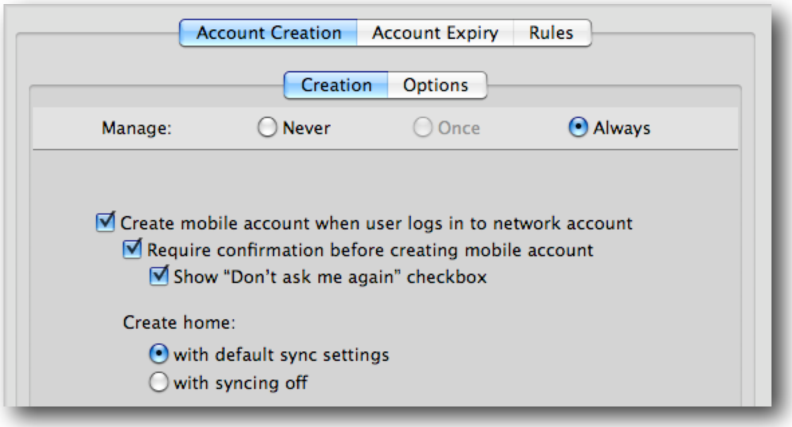
Combining the ease of management in network accounts with the performance and portability of local home directories, mobile user accounts are literally the "best of both worlds". The idea is that your user account information is stored in a network directory and at login, cloned to the local directory on a client system. Your network home directory contents are cloned onto the local system, and you have the option of mirroring your work so that your network home directory and your local home directory always contain the same data. Leopard introduces several options to enhance mobile accounts, such as expiry, Filevault, external accounts, and granular sync settings. Let's walk through setting up mobility and see how all the parts play together.

Who should be mobile account users? Practically everyone, really. Think about it - every person who uses the same computer every day; all the admins, secretaries, support folks, teachers, and other faculty. If that person uses primarily one computer all the time, they could have their account information managed on the network - passwords, policies, collaboration settings - and have their home directory on the computer they go to or carry all the time. Of course, mobile accounts are perfect for 1-1's. The user has a network managed account with all of their work on the computers they use all the time. Even if a user switches between a couple different computers during the day, being set up as a mobile account makes both productivity and management much easier.

1. Setting mobility - account creation

Mobility can be set at user, computer, or group level. Doing it at user level for testing is fine; but you'll find your self running back to the user account off and on to adjust settings. That can become a bit bothersome. Setting mobility at group level will work fine for pure Leopard environment; but remember that Tiger systems will see the mobility workgroup as a choice, not a requirement. Using the computer group as the focus for the mobility settings gives you both a more centralized focus and the ability to define which computers get mobile accounts while some others may remain in use for network users. The idea here is that "it depends" is a reality. You may want a 1-1 project set up where every computer in that set gets nothing but mobile account settings; yet your library machines may still support only network user accounts. Carefully setting up syncing and other mobility options may help you have a very well run environment.

To set mobility, log into an admin system as local admin, launch Workgroup Manager (WGM), and create or locate your computer group. (If you are going to set mobility by group or user, the steps are the same, just the account type is different.)



Manage: ○ Never ○ Once ● Always

☑ Create mobile account when user logs in to network account

☑ Require confirmation before creating mobile account

☑ Show "Don't ask me again" checkbox

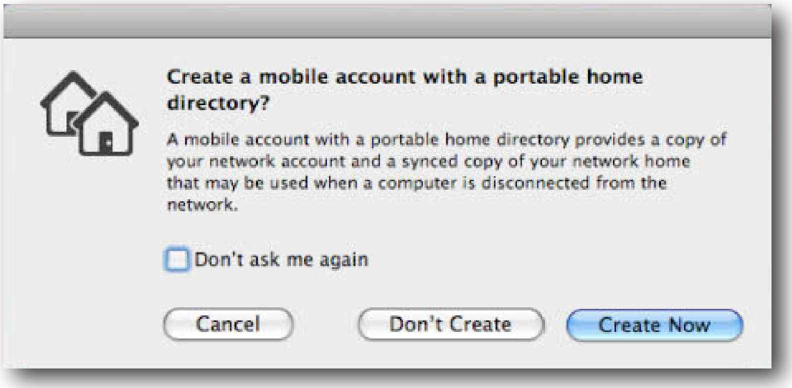
Create home:

● with default sync settings

○ with syncing off

The initial MCX Mobility window

Select the "Create mobile…" checkbox. I recommend turning off both of the sub-settings - "Require confirmation…" and "Show…" because they give the end user the option of backing out of the setting. You decide; here's what those dialogs look like to the end user at login:



Create a mobile account with a portable home directory?

A mobile account with a portable home directory provides a copy of your network account and a synced copy of your network home that may be used when a computer is disconnected from the network.

□ Don't ask me again

This might be too many choices for an end user

The "Don't Create" button causes the user to be logged in as a true network account and mounts their network home directory. The "Don't ask…" checkbox flags the system to never try to create a mobile account. If the user selects only the first option, the next time they log in, they'll get the dialog again. If they select the checkbox, it'll never show up - and you'll have to figure out why so many of your users are complaining that they can't log into their computer when they go offline.

2. To Sync, or not to Sync - that is the question

The setting at the bottom of the Creation pane needs the most consideration, planning, and work. The default is to create a mobile account with Portable Home Directory (PHD) support. The other choice is to create the mobile account without any synchronization options. Which is best? It depends.

a. MA/PHD's support RRtS

Now that you know this, you'll just jump right on using them right? Here's what I mean by all this. Mobile accounts with portable home directories support Rapid Return to Service. The idea is that a user logs into a client system and their network home is cloned completely to the local system. That user's account info is cached locally, allowing that user to log into the system both on and off the network. While they are working on their machine, the filesync mechanism is maintaining a mirror of the data in their local home directory with their network home directory. If anything happens to their computer - the backpack gets tossed down the stairs, the portable gets dropped in the snow, the teacher leaves their Macbook on the roof of the car and drives off - that user can get a loaner/replacement from the tech support folks, log in, and have all of their data resynced from the mirror (the network home) in a matter of minutes.

b. Capability with a cost

This capability does not come without a price. The infrastructure to support PHD's must be robust, and requires a lot of storage. Just look at the size of your own home directory on that laptop you are using. (Mine is currently ~110GB) A student using all the tools we offer in iLife to create projects for school can easily gather 6-8GB of material in a matter of weeks. Teachers can gather 10-20GB a semester. Trying to decide what is critical data is almost impossible for the IT staff. What is critical to an 8th grader is way different from what is critical to the network admin. We'll delve into that much more as we go along. The concept is solid though. What you are offering the end user is the security of knowing that their data always resides in at least two locations, and if anything happens to the computer the school lent them, they can get back up and running in a hurry.

c. What if I don't sync?

That said, there are reasons why you may not want to turn on Sync. First, the computer may spend far more time off the network than on it. Having the user's account information synced - which is always done - may be enough. If the user works in an office where they can backup, archive, or sync their own material may work for them. If you have a huge number of 1-1 users and your infrastructure cannot support more than providing them with basic network filesharing. A current Macbook ships with a 120, 160, or 250GB hard drive. If you provided PHD support for a thousand users and covered only 10% of their possible home directory space, you may need to set up as much as 23 terrabytes of storage on the network. (Yes, the computer may have 20-40GB of system files and applications; but that still leaves 100-200+GB of free space. Nature abhors a vacuum, and hard drives hate free space.)

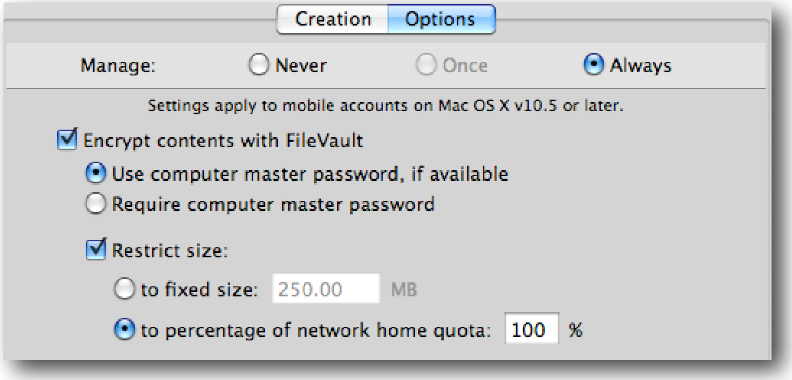
To close out this section, let's review the choices. If I turn on PHD sync, then I could mirror my local working directory back to the network. From there, I may even add backup to the network storage giving me true survivability. If I decide not to sync, then it is up to my users to provide a backup or mirror of their own data.

Let's continue this by exploring the various options to tune mobility.

3. Mobility options

a. FileVault

Leopard mobility provides several powerful options. First is FileVault. You can select to create a FileVault container for the user at first login, encrypting the contents of their home directory. The FileVault container is opened at login; if sync is enabled, the contents are mirrored to the network drive, then at logout FV recovers any extra space and closes itself up tight. If you have users who keep sensitive data on their school owned computer, and you want to maintain network manageability, then you will want to activate this option. Another point for this option is that you can define the size of the local home directory with this setup. Think of this as a locally controlled quota. (Be careful here - we'll talk about quotas later…)



Manage: ○ Never ○ Once ● Always

Settings apply to mobile accounts on Mac OS X v10.5 or later.

☑ Encrypt contents with FileVault

● Use computer master password, if available

○ Require computer master password

☑ Restrict size:

○ to fixed size: 250.00 MB

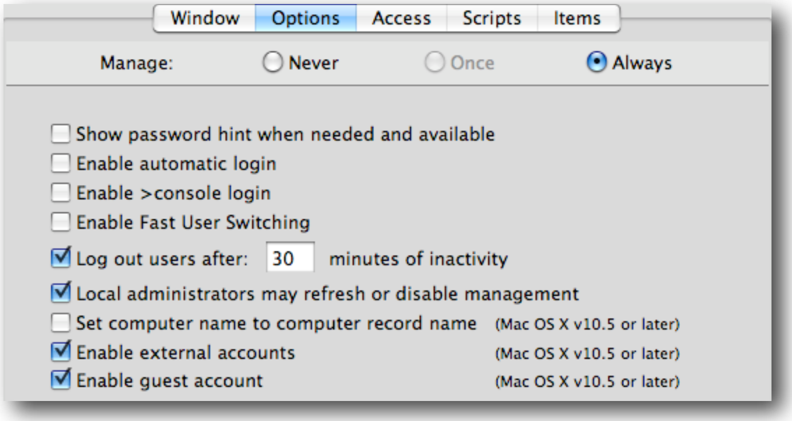
● to percentage of network home quota: 100 %

FileVault options for mobility

b. Home folder location - external accounts

The home folder location option is the sleeper of the mobility settings in Leopard. By default, the mobile account's home directory is created in the /Users folder on the boot drive of the client system. Using the home folder location setting, you push the home folder creation to a different partition, drive, or even an external device. This functionality is referred to as **external accounts** where the mobile account user logging into a managed client is not using the boot drive of that system for their local home directory location.

Setting this capability up requires two parts - enabling external accounts, and selecting a home folder location. Part one is done from within the Login settings:



Manage: ○ Never ○ Once ● Always

□ Show password hint when needed and available

□ Enable automatic login

□ Enable >console login

□ Enable Fast User Switching

☑ Log out users after: 30 minutes of inactivity

☑ Local administrators may refresh or disable management

□ Set computer name to computer record name (Mac OS X v10.5 or later)

☑ Enable external accounts (Mac OS X v10.5 or later)

☑ Enable guest account (Mac OS X v10.5 or later)

Note the "Enable external accounts" flag

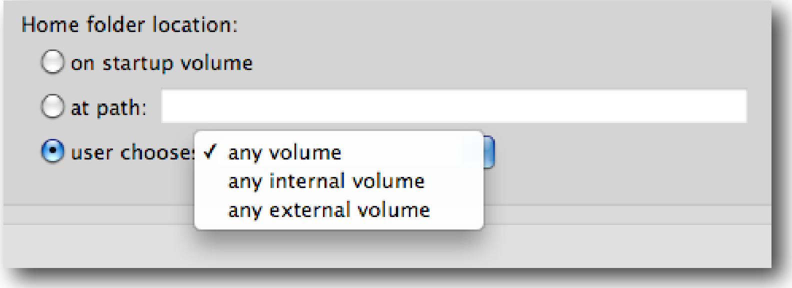
This setting allows MCX to support the use of a non-startup location for the mobile account user.

How does the use of external accounts affect the deployment of mobile accounts? Quite significantly. Here's a couple of examples:

- Students use their USB keys as their home directory location for their mobile accounts. When they use one of the Macbooks in the portable cart, they don't have to log in as a generic account, or rely on PHD sync to get their home directory. They just plug in the USB key and log in. The account is managed across the network, their home directory is on the USB key. If they switch computers, it doesn't matter since their home travels with them. Benefits include an automatic quota limited to the size of the USB key, and if they have PHD sync turned on, their USB home is mirrored back to the network. This provides RRtS in case they drop the key in a puddle.

- Teachers and students in a 1-1 have external accounts enabled. Their primary mobile home is on the Macbook they were issued. The teacher and students go to the new high-end video lab, all decked out with new 24" iMacs. Instead of just logging into the iMacs and relying on PHD sync to clone all of their info; they reboot their Macbooks in target disk mode ("T" key down at boot), making their 1-1 portables now external Firewire drives. The teacher goes to the iMac in the front of the room with all the AV/projection cables attached, plugs in her Macbook to the FW cable and MagSafe power adapter so thoughtfully provided by the tech support team, and logs in. Her home directory is still on her Macbook, so she can get to work right away. The students do the same. After class, they can log out and carry away with them all the work they did since it never went anywhere but into the designated home directory on the Macbook. So cool.

Setting this up in MCX is a matter of choosing the initial home folder location:



Home folder location:

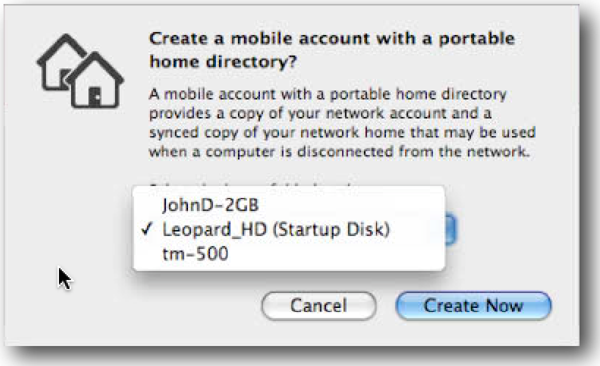
○ on startup volume

○ at path:

● user choose: ✓ any volume

Home Folder Location

Choosing the setting location is important. For a 1-1, you would want to select either 'any volume' or 'any internal volume' (or leave it set to 'on startup volume'). To force the users to choose an external device initially, select 'any external volume', and if you are using a system with multiple partitions, you can set the path to the partition. This would come in handy, for example, if you created a boot partition with the System and Applications, then wanted the user's home folder to be on the second partition. Here's an example of an open setting where the user can choose:

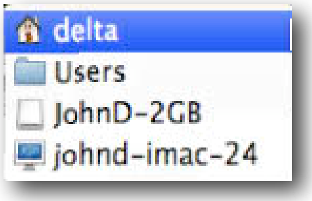


Create a mobile account with a portable home directory?

A mobile account with a portable home directory provides a copy of your network account and a synced copy of your network home that may be used when a computer is disconnected from the network.

User chooses between startup, FW, or USB drive

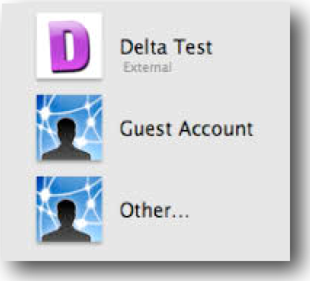
Once logged in, the user sees a normal Desktop and other settings. The key is where their home directory is now located:



Mobile account with external home directory

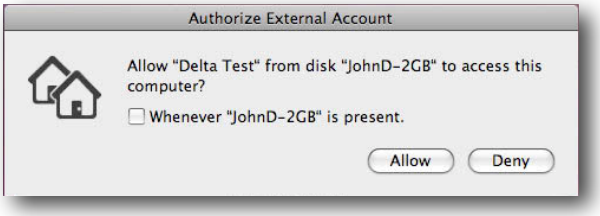
c. External account behaviors

The basic behavior of an external account is to login and see the user's home folder as normal. If you view the loginwindow as a list, you'll see the account as:



External account at Login

The external device can be unplugged without error at the loginwindow. If you go to another bound computer and plug in the device, the icon of the user will reappear in list view, otherwise there will be no indication the device is plugged in. Behavior changes if you go to an unbound system, for example, your home computer. If your home system is at the loginwindow, you will see this dialog:


Authorize External Account

Allow "Delta Test" from disk "JohnD-2GB" to access this computer?

□ Whenever "JohnD-2GB" is present.

External account device attached to unbound computer

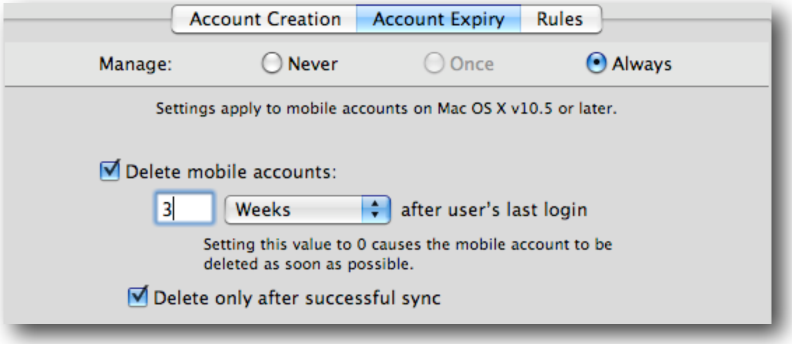
You must then authenticate as a local admin to allow login to continue. The benefit of this is that you can continue to use your external account's mobile home directory as your current working directory. **Note** - if you had a Windows machine, you'd get nothing - the external device must be formatted HFS+ for this to work. This also means that those generic USB keys our kids get for school will need to be reformatted to work as external account devices.

If you plug the device into a computer already logged in, the drive shows up as a normal external storage device. The external account's home will be on the device inside /Users/<username>. Unless you enable FileVault protection, your external device will be open for access, so be careful with your data.

4. Account Expiry

One side effect of working with mobile accounts is that each time a user logs into a client system a cached copy of that user's account is created on that computer as well as a local home directory. In a 1-1 environment, this can work really well. In an open lab, or portable cart setup, you could end up with many instances of the same user having logged into random machines over time, leaving little cloned copies of themselves all over. Luckily, there are two solutions to this problem. The first one was invented by Greek scholars about 2500 years ago - it's called the seating chart. Insuring the student uses the same computer each time they come to the lab, or scheduling and assigning specific users to specific portables keeps the 'rabbit droppings' down.

The second solution is **account expiry**. This setting is designed to age a mobile account, and once it has reached the expiration date, the account will be deleted, along with the local home directory, at the login event by another user.



Manage: ○ Never ○ Once ● Always

Settings apply to mobile accounts on Mac OS X v10.5 or later.

☑ Delete mobile accounts:

3 Weeks after user's last login

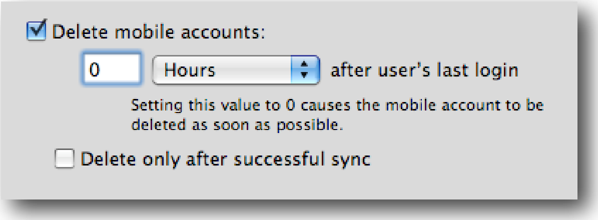
Setting this value to 0 causes the mobile account to be deleted as soon as possible.

☑ Delete only after successful sync

Mobile account expiration settings

Figuring out what the expiration setting should be is important. If you set the interval too short, you'll force users to go through a complete re-sync at next login. A good practice would be to find out what the longest break is during the year, including flu season, then double that number. Taking the expiration to the extreme, you can set the value to zero and turn off the "Delete only after successful sync" setting. That done, your mobile accounts will be deleted at logout. This could come in handy in non-sync setups where you want your users to have the benefit of running as mobile users with local homes, yet do not want the systems to have any leftovers after use. Users would have to store any files they want to keep on another device, use a workflow sharepoint with hand-in folders, or in the case of lower grades, print their work before logging out.

Turning the expiration setting on for 1-1's with their home directory on the startup disk is not a good idea. External accounts with the home directory on an external, removable device are not affected by the expiration settings.



☑ Delete mobile accounts:

0 Hours after user's last login

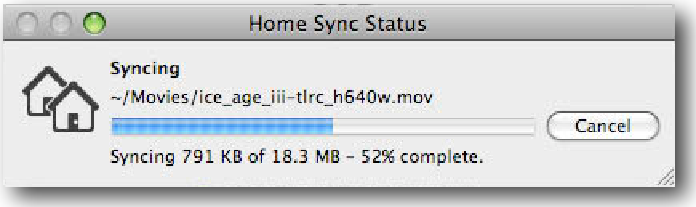
Setting this value to 0 causes the mobile account to be deleted as soon as possible.

☐ Delete only after successful sync

Immediate elimination of mobile accounts

5. Rules for Portable Home Directory sync

Setting up PHD sync can be as simple or as complicated as you want to make it. The idea is very basic - at first login, the entire contents of the user's network home directory plus the contents of any local home directory with that user's name are merged. If there is no network home directory yet, it is created using the network home server's home directory template, then cloned to the local drive. While the user is logged in, at a defined interval, the entire contents of that user's working directory (the local one) is synced with the network home, minus the user's Library folder and a few other active files. At logout, the entire contents of the local home is synced, minus a few designated files.

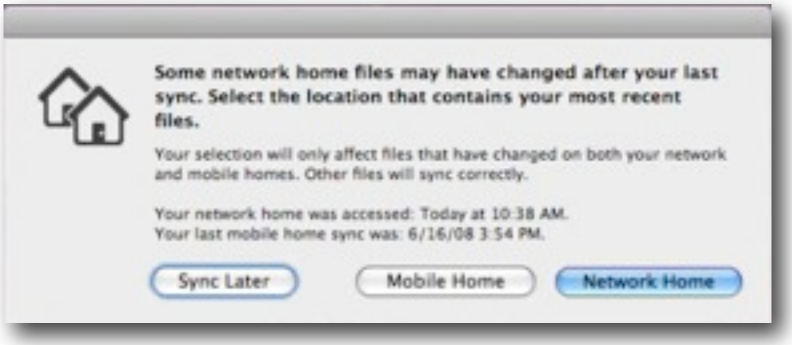


Syncing
~/Movies/ice_age_iii-tlrc_h640w.mov
Syncing 791 KB of 18.3 MB – 52% complete.

First time login sync brings everything together

Subsequent logins will sync only the user's Library folder and the MS User Data folder, if it exists. While logged in, the user's entire home folder, minus the Library, will be synced. Only files that change, are created, or deleted, will be synced. This is the default behavior, and if the network infrastructure is up to speed, it will work very well.

Conflicts might happen. If you add items to a user's network home directory when they are offline, the next time they go to log in, they'll get a conflict dialog. Under Tiger, it just left you with a "Yeah, and…?" but in Leopard, you get a warning when you last touched your network home versus your local (mobile) home:



Some network home files may have changed after your last sync. Select the location that contains your most recent files.

Your selection will only affect files that have changed on both your network and mobile homes. Other files will sync correctly.

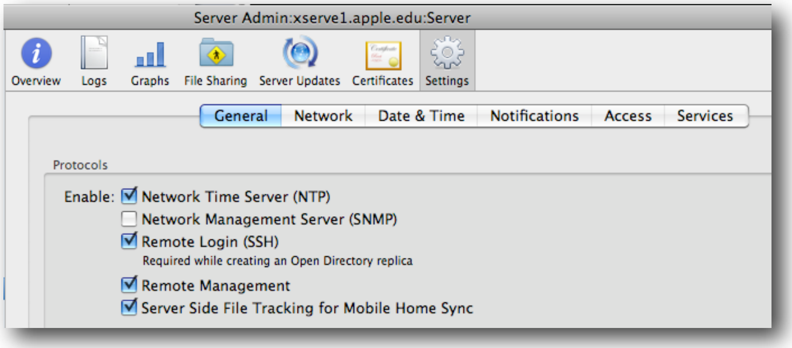
Your network home was accessed: Today at 10:38 AM.
Your last mobile home sync was: 6/16/08 3:54 PM.

New Sync conflict dialog

At least now, you can check and say "Hey, the IT folks did say they were adding stuff to our homes. I should click on 'Network Home'."

a. Server side sync

Leopard contains a serious improvement in PHD sync capability over Tiger. Besides a whole new filesync mechanism, the server now supports "server side tracking" for mobile home sync. This can seriously improve PHD sync performance for Leopard clients. Under Tiger, when PHD sync kicked in, the local home directory was watched over by a filesync database. It kept track of all file system changes. When the sync interval hit, this database reported the files that had changed, been added, or deleted. At that point, Filesync had to scan the entire network home directory for that user, file by file, looking for the files that matched the changes reported by the database. If you had a few hundred files, you wouldn't notice. Change that to a few hundred thousand files and things got really slow.



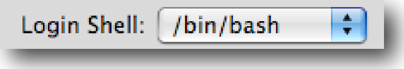
Protocols

Enable: ☑ Network Time Server (NTP)
□ Network Management Server (SNMP)
☑ Remote Login (SSH)
Required while creating an Open Directory replica
☑ Remote Management
☑ Server Side File Tracking for Mobile Home Sync

Server side file tracking for PHD sync

Leopard allows you to set the same filesync database on the user's network home directory. When file system changes take place, a filesync database is maintained at both ends. The PHD sync process now consists of two databases being compared. It no longer matters how many files you have in your home directory.

In order for this to work for your mobile accounts, all users must have a valid shell assigned in the network directory, using /dev/null won't work.

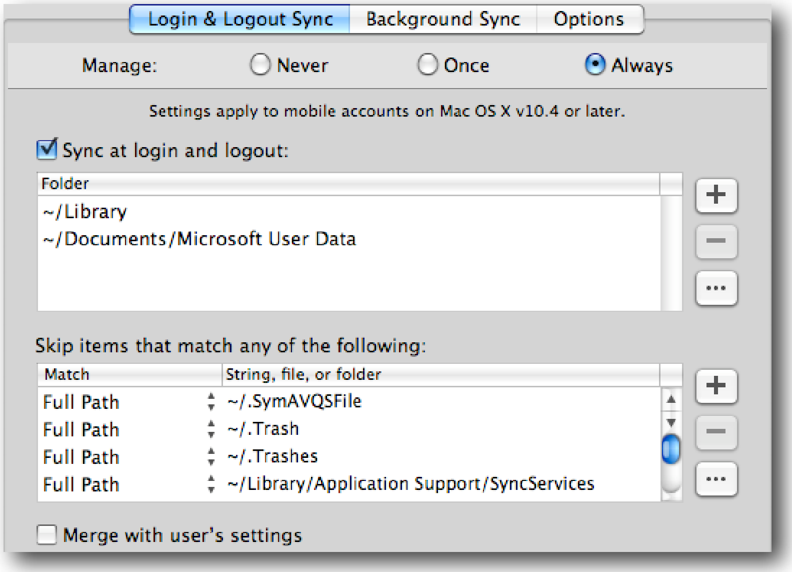


Login Shell: /bin/bash

User login shell set to allow server side PHD sync

b. Tuning Login/Logout sync

The default set for login/logout sync looks like this:



Manage: ○ Never ○ Once ● Always

Settings apply to mobile accounts on Mac OS X v10.4 or later.

☑ Sync at login and logout:

Folder
~/Library
~/Documents/Microsoft User Data

Skip items that match any of the following:

Match	String, file, or folder
Full Path	~/.SymAVQSFile
Full Path	~/.Trash
Full Path	~/.Trashes
Full Path	~/Library/Application Support/SyncServices

□ Merge with user's settings

Default Login/logout sync settings

If you turn on these settings, you can override the default sync settings. There are two primary scenarios for using PHD's - 24/7 users and 9-5 users. The first case is the student/faculty member who has been provided with a system for full time use. They take the computer home, use it all day, weekends, and during non-school days. Those people should have the least load placed on them when returning to the network for performance reasons, and they always treat their mobile account's local home directory as their primary storage location. If this person came back to school from a long weekend and had uploaded a few hundred MB's of podcasts, worked on an iMovie project, and updated their iPhoto collection (plus the movie trailers, etc.), their initial login at school would consist of a lot of waiting for a very long sync period.

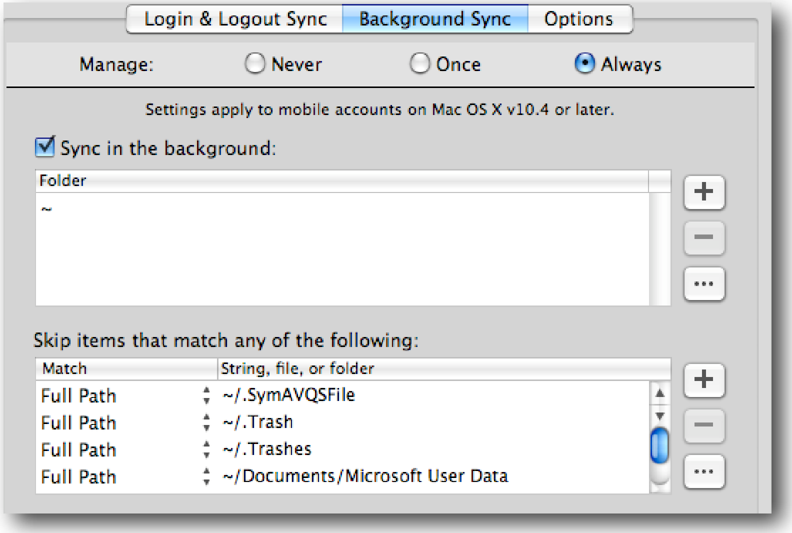
Since they are already holding onto their primary home directory, you would want to get them logged in as fast as possible, then let their offline additions to their home directory sync in the background.

Users who work in the office all day, then head out at 5pm can have even more open settings. For these users, you can set their login/logout sync to be "~" since they haven't added anything to their systems overnight. If you add some files to their network homes during off hours, those files will then sync at login.

One file that does need to be added to the Login/Logout sync set, if you use the defaults as a starting point, is the iTunes Library. I'd suggest either adding the iPhoto Library itself, or the Pictures folder to the Login/Logout set. This is because the sync process throws an error when a user is running iPhoto and background sync tries to mirror the picture library.

c. Tuning Background sync

Background sync needs very little tuning. You might want to add the iPhoto Library or the Pictures folder to the "Skip" set to avoid the issue noted above.



Manage: ○ Never ○ Once ● Always

Settings apply to mobile accounts on Mac OS X v10.4 or later.

☑ Sync in the background:

Folder
~

Skip items that match any of the following:

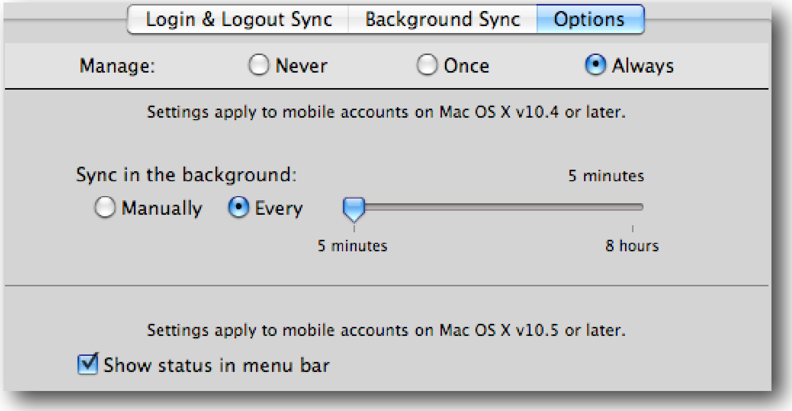
Match	String, file, or folder
Full Path	~/.SymAVQSFile
Full Path	~/.Trash
Full Path	~/.Trashes
Full Path	~/Documents/Microsoft User Data

Note that the MS User Data folder is not synced in the background

One thing of note is the exclusion of the "Microsoft User Data" folder. This is due to the behavior of Entourage. Since filesync acts on files whose modified timestamp have changed, it would constantly be trying to mirror the Entourage mail database. Since the Entourage db is one large flat file, every time you check mail, or select the Entourage app window, the file shows that it changed. This would result in a constant state of being mirrored. If you are not using Entourage, then you could delete that item from the "Skip" set.

d. Tuning the timing (Options)

The default interval for PHD sync is 20 minutes, and that may be too long.



Manage: ○ Never ○ Once ● Always

Settings apply to mobile accounts on Mac OS X v10.4 or later.

Sync in the background: 5 minutes

○ Manually ● Every

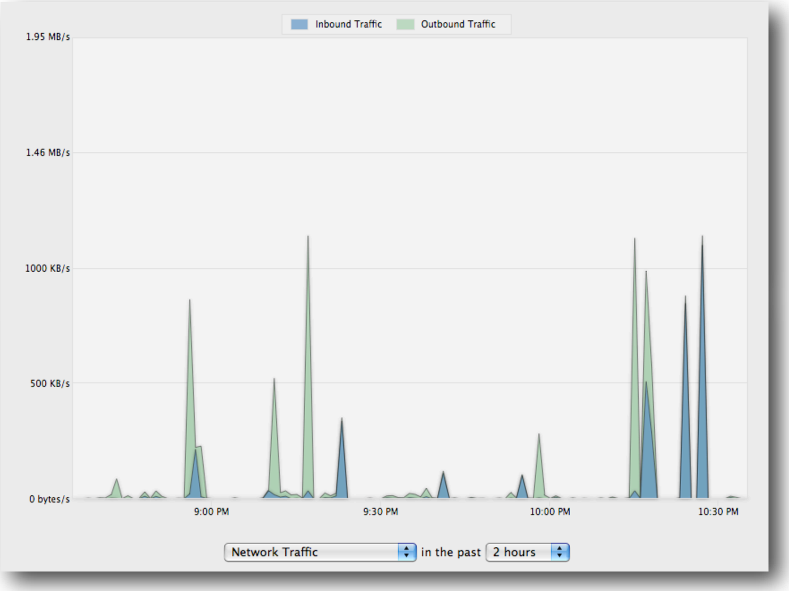
5 minutes … 8 hours

Settings apply to mobile accounts on Mac OS X v10.5 or later.

☑ Show status in menu bar

Keep the sync interval short

This time may be too long for many of the sites to handle with their infrastructure. Several sites set their sync interval to manual. My recommendation is to set the interval as short as possible for your network - perhaps in the 9-11 minute range. I set the interval in my labs to 5 minutes. Leaving it set to manual can work if the users are trained to invoke it often, perhaps in an AV lab where they don't want PHD sync to interfere with video capture. Where the problem pops up is when the users wait until just before the bell rings, then rush to sync. The traffic would be a tsunami of filesharing.



Short sync intervals keep the traffic down in the long run

e. Other options - getting restrictive

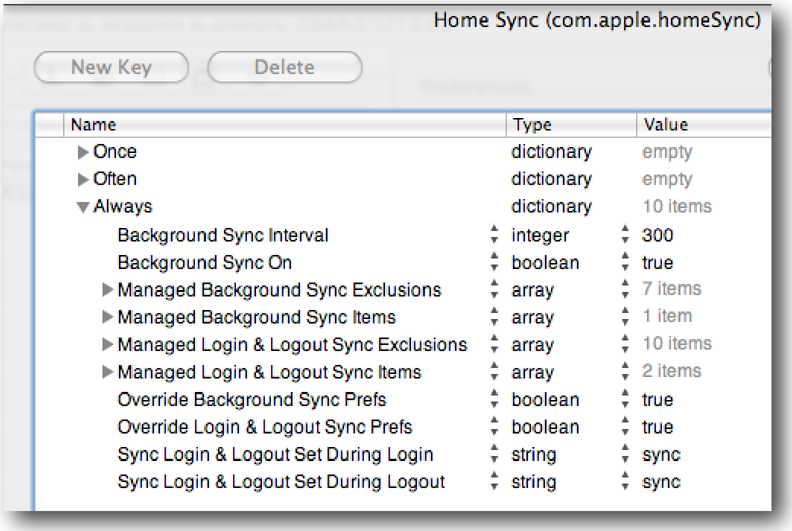
Many IT shops decide to manage PHD sync by shutting it down to just syncing the user's Documents folder, or skipping all .mp3 files. I do not recommend doing this for a couple of reasons. First, if you don't know what the data is, or where the users store it, making a call like that can result in lost data - voiding the concept of RRtS. Second, if you try to filter by file type, say .mp3's, you may be skipping critical podcasts, or an audio interview a student did that is imbedded in a Keynote presentation. That would cause the Keynote presentation to be rendered worthless. Remember, many files today are bundles, with dozens of smaller files inside.

Finally, trying to force the users to store their critical items in the Documents folder because all you'll agree to mirror is that folder will only result in the user putting everything into that folder, including their iTunes library. If it comes down to a matter of network storage space, you might be better off setting up FileVault containers with a specified size, or using external accounts on USB keys issued by the school. Then again, you could always add more online storage. Your Apple account exec would be glad to help you there.

What is mission critical to an 8th grader or a teacher often has no bearing on what the IT support staff thinks is critical. Support the mission - support the end user. :-)

6. Digging deeper - Details and mobility

Many of the required settings for mobility are exposed in the GUI. However, there are also just as many settings buried in the Details. These settings are hidden on purpose. There are some really serious problems that can be created by playing with these without careful consideration. That said, let's look through the briar patch:



Home Sync (com.apple.homeSync)

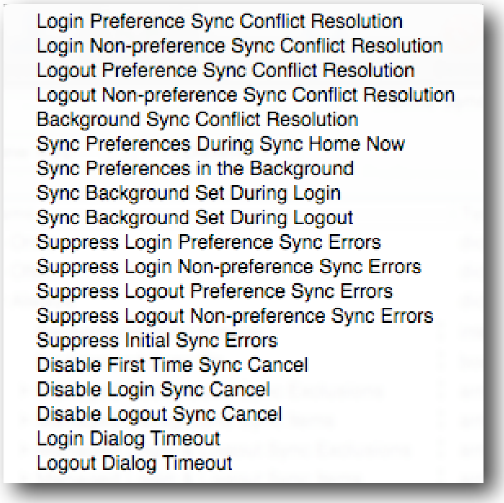
Name	Type	Value
▶ Once	dictionary	empty
▶ Often	dictionary	empty
▼ Always	dictionary	10 items
Background Sync Interval	integer	300
Background Sync On	boolean	true
▶ Managed Background Sync Exclusions	array	7 items
▶ Managed Background Sync Items	array	1 item
▶ Managed Login & Logout Sync Exclusions	array	10 items
▶ Managed Login & Logout Sync Items	array	2 items
Override Background Sync Prefs	boolean	true
Override Login & Logout Sync Prefs	boolean	true
Sync Login & Logout Set During Login	string	sync
Sync Login & Logout Set During Logout	string	sync

The 'visible' sync options from underneath

These are the items that you can see from above. There are a few settings that could be tweaked; specifically, the last two listed. The options there are 'sync', 'not sync' and 'automatic'. You can set the sync option to skip sync at login and only sync at logout, or vice versa. The 'Override…' settings mean to override the local settings made in System Preferences by that user. This would be if you allowed the user to set their own sync prefs. By overriding them, you are forcing the use of the MCX settings from the network directory.

a. Hidden sync prefs

The settings that are hidden are listed below. These are very dangerous settings to play with. I seriously suggest testing any adjustments to the defaults with a non-production set of systems.



Login Preference Sync Conflict Resolution
Login Non-preference Sync Conflict Resolution
Logout Preference Sync Conflict Resolution
Logout Non-preference Sync Conflict Resolution
Background Sync Conflict Resolution
Sync Preferences During Sync Home Now
Sync Preferences in the Background
Sync Background Set During Login
Sync Background Set During Logout
Suppress Login Preference Sync Errors
Suppress Login Non-preference Sync Errors
Suppress Logout Preference Sync Errors
Suppress Logout Non-preference Sync Errors
Suppress Initial Sync Errors
Disable First Time Sync Cancel
Disable Login Sync Cancel
Disable Logout Sync Cancel
Login Dialog Timeout
Logout Dialog Timeout

Hidden sync prefs (keys)

Note: Playing hard and fast with these settings may result in data loss for your users! Guessing which location is right for your users is going to be wrong 50% of the time. It's better to teach them to pay attention to where they put their stuff, and report really odd behavior to the sys admin.

Let's take a cautious look at each of these with the qualifier that employing these keys is not supported in any way outside of the default values. I have pulled the descriptions of the actions directly from the hints in WGM.

1. Login Preference Sync Conflict Resolution

This setting affects syncing ~/Library at login. Set to "mobileHomeWins" to merge homes and have the local (mobile) home win conflicts, "mobileHomeCopy" to copy the local home to the network home, "automatic" or "networkHomeWins" to merge homes and have the network home win conflicts, or "networkHomeCopy" to copy the network home to the local home.

2. Login Non-preference Sync Conflict Resolution

This setting affects syncing everything besides ~/Library at login. Set to "showConflictDialogs" to show dialogs when conflicts occur, "mobileHomeWins" to merge homes and have the local (mobile) home win conflicts, "mobileHomeCopy" to copy the local home to the network home, "automatic" or "networkHomeWins" to merge homes and have the network home win conflicts, or "networkHomeCopy" to copy the network home to the local home.

3. Logout Sync keys - behave the same as the Login keys

There is a consistency in the pattern here…

4. Background Sync Conflict Resolution

This setting affects syncing everything besides ~/Library in the background. Set to “automatic” or “showConflictDialogs” to show dialogs when conflicts occur, “mobileHomeWins” to merge homes and have the local (mobile) home win conflicts, “mobileHomeCopy” to copy the local home to the network home, “networkHomeWins” to merge homes and have the network home win conflicts, or “networkHomeCopy” to copy the network home to the local home.

5. Sync Preferences during Sync Home Now

This setting allows you to sync ~/Library during a Sync Home Now sync. Set to “automatic” for the best choice, “sync” to sync preferences during Sync Home Now, or “dontSync” to not sync preferences during Sync Home Now.

I’m skipping a few now to show you this one:

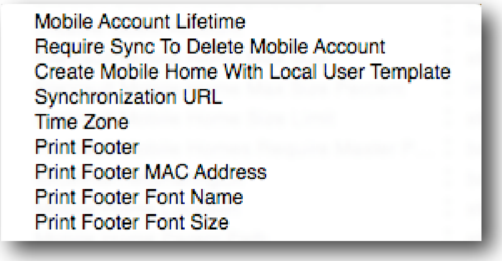
6. Suppress Login Preference Sync Errors

Set to true to suppress error dialogs during login sync of ~/Library.

The “suppression” keys can result in problems for users where an actual problem between two files may exist. While some errors get reported and are not really problems, if you suppress the conflict dialogs, you may create a bigger problem.

b. Hidden Mobile Account keys

These are settings come from the **Mobile Account & Other Options** (com.apple.MCX) settings and can enhance performance of the mobile account, or set additional restrictions. Any of the ‘Options’ not set in the GUI show up here as unused keys. In this picture, the “Expiry” keys are unused:



Mobile Account Lifetime
Require Sync To Delete Mobile Account
Create Mobile Home With Local User Template
Synchronization URL
Time Zone
Print Footer
Print Footer MAC Address
Print Footer Font Name
Print Footer Font Size

Unused mobile account and ‘Other’ keys

The first two listed wouldn’t be here if I had enabled “Expiry” settings, so we’ll ignore those. The last four are the values used in the “Printing” preference as we have already covered. The important ones are the middle three -

1. Create Mobile Home With Local User Template

When creating a mobile account, create the local home folder using the computer’s user template. (This allows you pre-populate the home template of your master image, then use that as the source of mobile account home directories versus pulling the template from the network server. The end result will be that the network home directory and the local template will be combined. The user will have a local home directory consisting of the contents of both templates. What this allows you to do is keep the network template very light, e.g. empty, and put all the needed items into a local template that is cloned as part of the image.)

2. Synchronization URL

URL of the network home used for home sync. Only settable for mobile account creation. The string “%@” will be substituted with the user record name before use. Example: afp://myserver.apple.com/Users/BuildingA/%@ .

3. Time Zone

Set to a recognized time zone name, such as “America/Los_Angeles” or “US/Pacific” as found in /usr/share/zoneinfo.

8. FileSync troubleshooting

Remember, PHD sync is mirroring, not backup! So the first troubleshooting tip you get is to make sure the user didn’t throw something away expecting “the copy on the server to be there.” Files deleted at either end are deleted at next sync interval.

If you need to try to track down sync problems, the client log lives at ~/Library/Logs/FileSyncAgent/FileSyncAgent-verbose.log. For PHDs with server side tracking, the log on server lives at ~/Library/Logs/FileSync-server/FileSync-server-verbose.log.

To reset FileSync, there are two methods. The “soft” one is to log onto the client as local admin and delete the mobile user account and home directory from within System Preferences. The user can log in again and have their mobile account and home directory resynced from the network. The “hard” reset involves the “soft” steps plus:

- on the server, delete the user's ~/.FileSync folder

- on the server, delete the user's ~/Library/FileSync folder

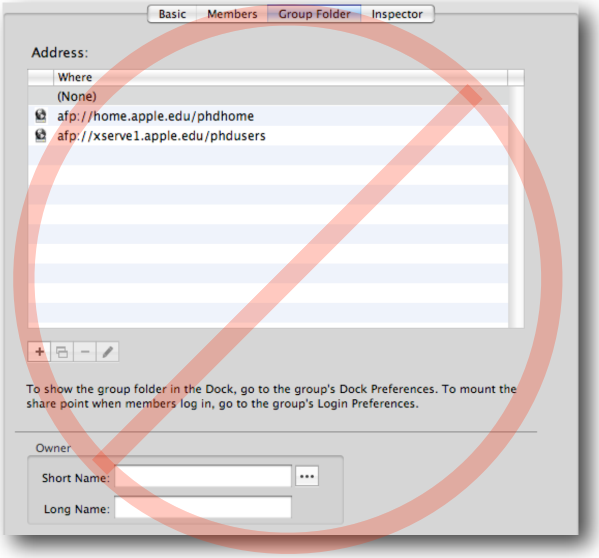
7. Workflow and Collaboration tips

One of the biggest reasons schools used Macintosh Manager, and set up management by workgroup, was to take advantage of the group folder functionality. Having a sharepoint mount at login for the user with a hand-in folder and one or more folders for class materials is a great way to encourage workflow. The idea of workflow isn't new; it's been used for many, many years. The advantage Leopard has is that we can now create a single workflow for many users simultaneously, without the requirement for multiple workgroups for the user to choose between. Add to that the new collaboration tools within Leopard to provide a true ability to share and distribute knowledge, and we are one step closer to the goal of meeting the standards asked of our students by the world - *be able to work together in teams, be able to solve complex problems, be able to present your solutions clearly and concisely in both written and oral form.*

What we are going to set up is an environment where the students and faculty can share information using common storage, yet allow for specific access to information by group as needed.

a. Setting up the workflow

The parts needed to make this work are: a set of user groups reflecting classes or activities for the students and staff, a common sharepoint with folders depicting the same set of groups, and the management settings to bring it all to the user's Desktop. A workflow server would be the server, or servers, that contain common access sharepoints for school use. By using 'common' sharepoints with sub-folders, we no longer need to create dedicated 'group folders'.



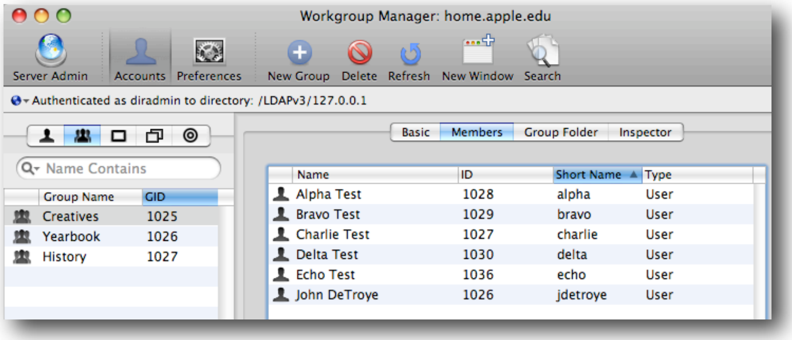
(None)
afp://home.apple.edu/phdhome
afp://xserve1.apple.edu/phdusers

To show the group folder in the Dock, go to the group's Dock Preferences. To mount the share point when members log in, go to the group's Login Preferences.

Not creating group folders anymore

1. Creating groups, not workgroups

Instead of creating a series of workgroups with group folders for users to log into, we are going to create only UNIX groups on our 'workflow' server. These groups will contain users from our primary directory server. The groups, for this example, are going to be: History, Creatives, and Yearbook. In WGM, connected to our workgroup server as a directory admin, we create these three groups and populate them.



Authenticated as diradmin to directory: /LDAPv3/127.0.0.1

Group Name	GID
Creatives	1025
Yearbook	1026
History	1027

Name	ID	Short Name	Type
Alpha Test	1028	alpha	User
Bravo Test	1029	bravo	User
Charlie Test	1027	charlie	User
Delta Test	1030	delta	User
Echo Test	1036	echo	User
John DeTroye	1026	jdetroye	User

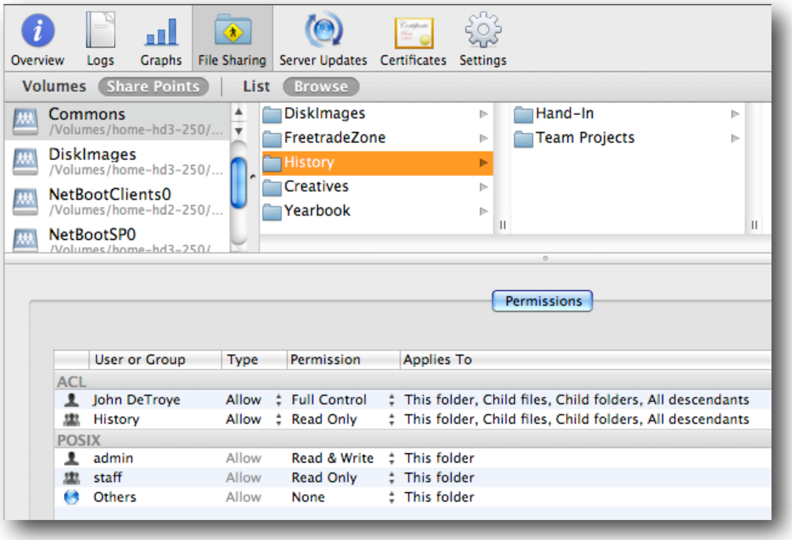
Three groups created on the workflow server

Now we move to **Server Admin** to finish this setup.

2. Building the "Commons"

In Server Admin, we select the workflow server, then the File Sharing tool. We then create a 'Commons' folder on one of the drive volumes. Within that folder, we create sub-folders for each of the groups. Keeping ahead of the game, we can also create sub-folders for each group folder for 'Hand-In' and other tasks as needed by the faculty member who will own the folder. We can also just show the owner of the folder how to manage it themselves, passing on the responsibility of keeping the folder in working order to the owner.

Next, we select one of the new group folders and add it's owner and group into the **ACL** section of the Server Admin/File Sharing pane.

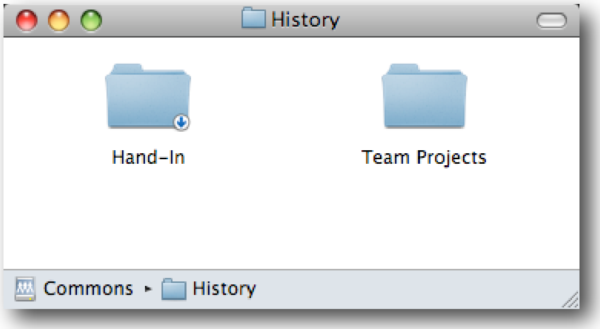


User or Group	Type	Permission	Applies To
ACL			
John DeTroye	Allow	Full Control	This folder, Child files, Child folders, All descendants
History	Allow	Read Only	This folder, Child files, Child folders, All descendants
POSIX			
admin	Allow	Read & Write	This folder
staff	Allow	Read Only	This folder
Others	Allow	None	This folder

Owner gets "Full Control", group gets "Read Only", Others get zip

Next, you need to select the 'Hand-In' folder inside "History" and turn it into an actual hand in folder - the write only folder - by setting the basic permissions to 'write only' for 'Others'.

Repeat these steps with the other folders. Note that the only folder that is actually "shared" is the 'Commons' folder. This gives us a really streamlined method of providing our workflow environment. Test the setup by mounting the sharepoint as various users onto your admin system. Access to the folders and sub-folders should match the ACL's you set.



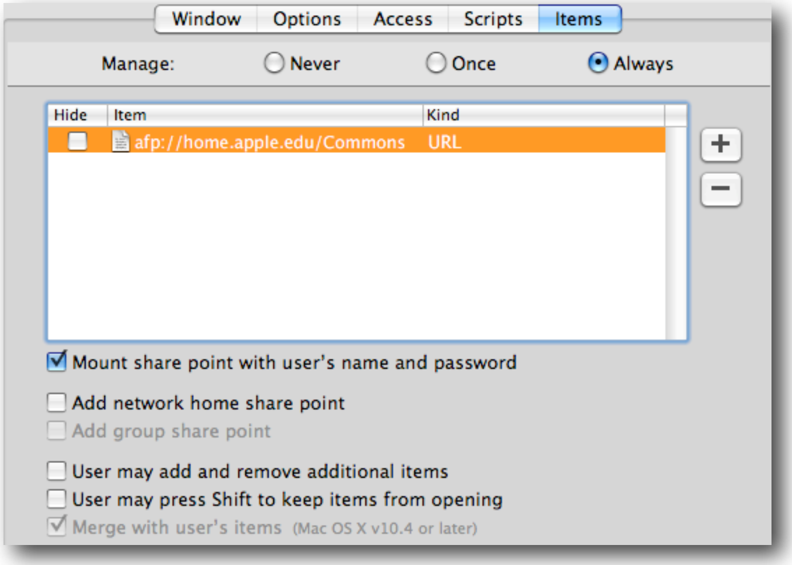
History folder from user "nova's" perspective

Note that the hand-In folder looks like one with the cute little arrow in the corner. If you had logged in as the user 'johnd', you'd have full access to all folders. You can experiment with the various permissions settings to achieve exactly the effect you want. I set the 'Team Projects' folder, for example, to allow all group member r/w access, but denied them delete permissions. This keeps people from accidentally trashing someone else's work. They can ask me to have items deleted.

3. ACLs and MCX together at last

To make this workflow complete, we need to have the 'Commons' folder show up when the users login. To do this, we use the same technique we used all the way back to MacMgr. First, on an admin system, we mount the 'Commons' sharepoint on the Desktop. Second, we launch and authenticate to our MCX server, then select the managed computer group we set up.

In Preferences/Login, select the 'Items' pane, and select 'Always', then drag the image of 'Commons' from the Desktop into the window. Highlight the sharepoint item, then select the "Mount share point…" checkbox.



Manage: ○ Never ○ Once ● Always

Hide	Item	Kind
□	afp://home.apple.edu/Commons	URL

☑ Mount share point with user's name and password

□ Add network home share point

□ Add group share point

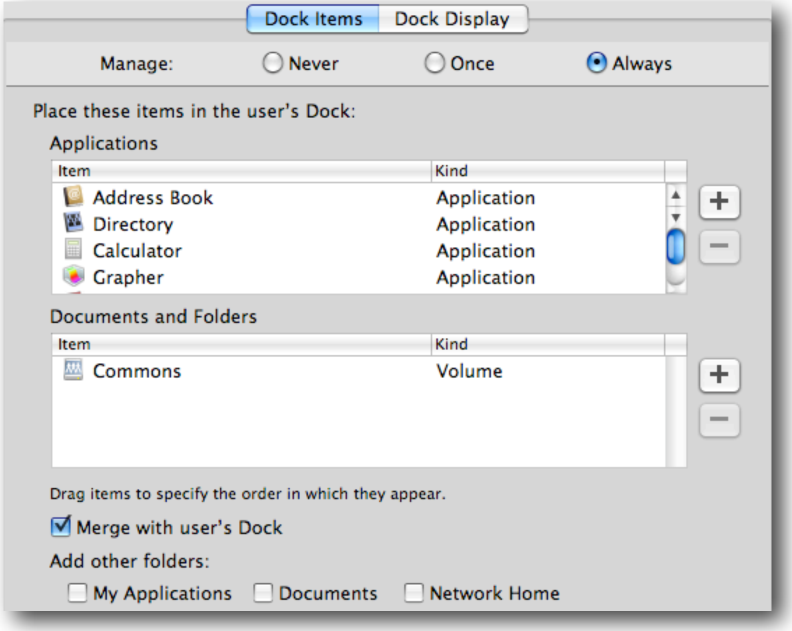
□ User may add and remove additional items

□ User may press Shift to keep items from opening

☑ Merge with user's items (Mac OS X v10.4 or later)

'Commons' will be mounted at login with the current user's credentials

In case the user ejects the volume, we'll also add the sharepoint to the Dock.



Manage: ○ Never ○ Once ● Always

Place these items in the user's Dock:

Applications

Item	Kind
Address Book	Application
Directory	Application
Calculator	Application
Grapher	Application

Documents and Folders

Item	Kind
Commons	Volume

Drag items to specify the order in which they appear.

☑ Merge with user's Dock

Add other folders:

□ My Applications □ Documents □ Network Home

'Commons' is now a Dock item for all users.

Note: If we had only added the sharepoint to the Dock, then at login, the user would have seen a "?" in the Dock. This is because the Dock item is only a URL. At login, Dock setting is checked by the Dock to resolve all the components. The URL would show a sharepoint; but the Dock wouldn't see it - hence the "?".

You can use this trick to create and mount sharepoints with databases used by educational applications, such as TypeToLearn, in order to premount the db for the application. If you do work with network databases for users, make sure you set the ACL on the folder to allow the user to write but not delete. Otherwise, the user could either toss the db in the Trash, or not be able to save any test results.

b. Collaboration tools - a simple beginning

Leopard introduced a whole suite of collaboration tools for group or team work. Each of the tools has a use in the educational workflow. This quick look is only a taste - the world of collaborations services, podcasts, etc., is huge - and well worth exploring.

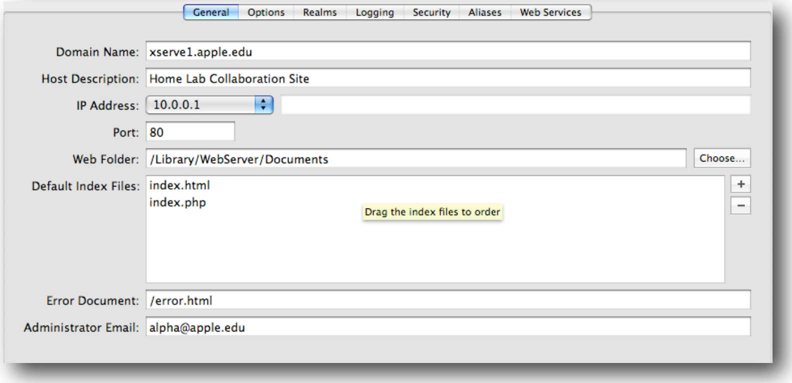
Who has not spent some time browsing Wikipedia? What teacher, being interested in educational technology, would love to have their own wiki? Imagine not having to post and repost all the class materials, or having the ability to let students help flesh out a class project online. Let's look at setting up the bare minimum we would need to start this process. Imagine a language arts teacher being able to post a reading assignment, then having the students comment on the assignment in a class blog? Think about an internal activities calendar run by the various groups at school?

1. Server setup for collaboration

First, if your wiki server won't be on the ODM, you must use an authenticated bind to attach your server to the directory. Do that using **Directory** Utility**.** Using

Server Admin, select the "Web" service. Choose the "Web Services" pane in "Settings". Choose a template for your wiki site(s).

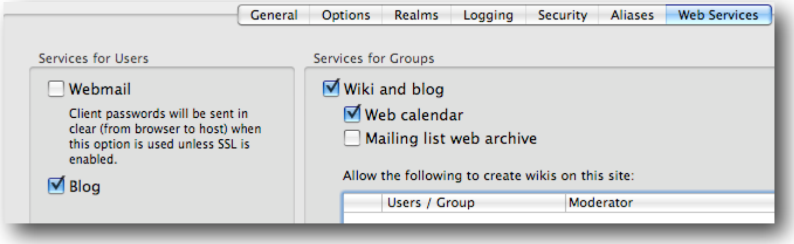
Next, select the "Sites" tool and select the current site "*". Edit the entries so that your site information is fixed versus blank and select the "Enable" checkbox, as in this example:



Domain Name: xserve1.apple.edu
Host Description: Home Lab Collaboration Site
IP Address: 10.0.0.1
Port: 80
Web Folder: /Library/WebServer/Documents
Default Index Files: index.html
index.php
Error Document: /error.html
Administrator Email: alpha@apple.edu

Cleaning up the entries for your wiki site

In the "Options" pane, make sure the *Performance Cache* is not enabled. Select the "Web Services" pane and turn on the collaboration items you will use.



Services for Users

□ Webmail

Client passwords will be sent in clear (from browser to host) when this option is used unless SSL is enabled.

☑ Blog

Services for Groups

☑ Wiki and blog

☑ Web calendar

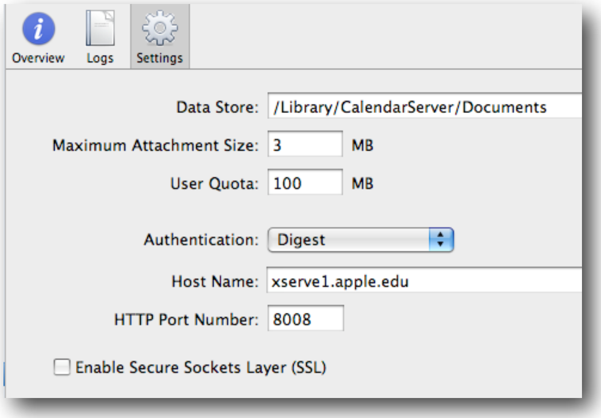
□ Mailing list web archive

Allow the following to create wikis on this site:

	Users / Group	Moderator

Establishing collaboration services

Next, locate the "iCal" service and turn it on. I suggest setting the authentication to "Digest" at this time.



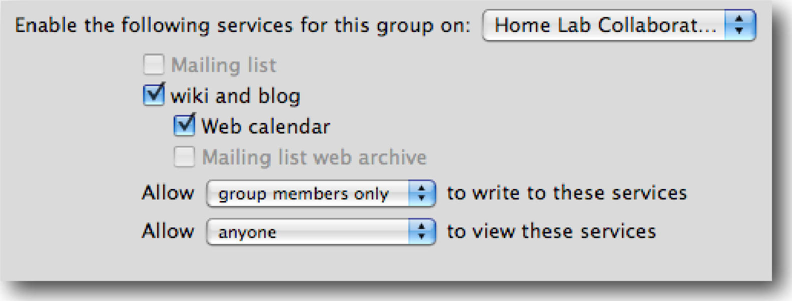
Data Store: /Library/CalendarServer/Documents
Maximum Attachment Size: 3 MB
User Quota: 100 MB
Authentication: Digest
Host Name: xserve1.apple.edu
HTTP Port Number: 8008
□ Enable Secure Sockets Layer (SSL)

iCal service settings - use Digest for authentication

The services are using the boot drive of the server for their storage for now. You might want to change that before going into production, or just play with it this way for now.

2. Setting the group(s) to use the collaboration services

In **Workgroup Manager**, locate the group(s) you are going to activate collaboration for. In the "Accounts/Basic" pane, enable services as in this example:



Enable the following services for this group on: Home Lab Collaborat...

□ Mailing list
☑ wiki and blog
☑ Web calendar
□ Mailing list web archive

Allow group members only to write to these services

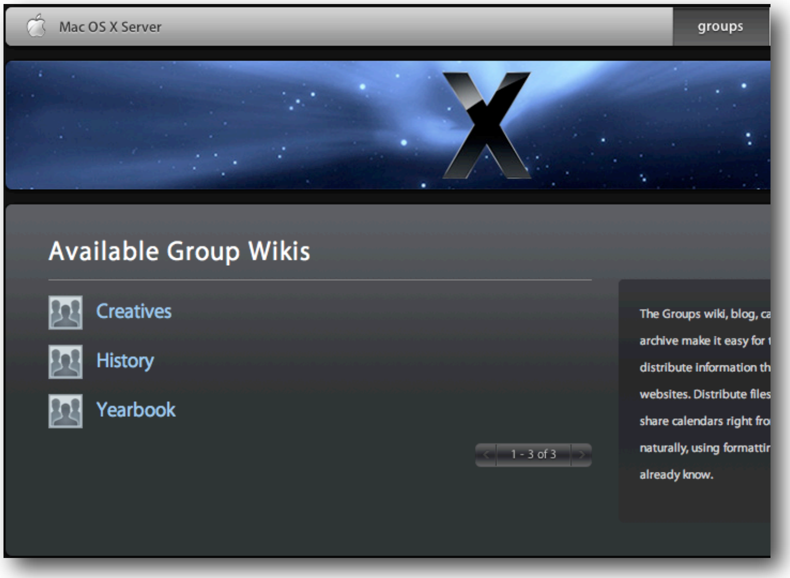
Allow anyone to view these services

Enabling collaboration services for a group

When you have added the settings for your groups, go back to Server Admin and stop/restart the web services.

3. Testing it out

With everything up and running now, launch Safari (or your favorite web browser), and enter the url to your new collaboration server as follows - http://<webserver.domain.name>/Groups - and you'll see something like this:



Available Group Wikis

Creatives
History
Yearbook

1 - 3 of 3

The Groups wiki, blog, ca
archive make it easy for t
distribute information th
websites. Distribute files
share calendars right fro
naturally, using formattin
already know.

The group page of my new collaboration server

From this point on, you can click, select, edit, update, and enhance to your heart's content. You'll find the links to the blogs and calendars inside each group's wiki page. Have fun and for more info, check out the official documentation on collaboration services.

8. Additional tips and tricks for management

These are just a few tips that I have gathered while working with both Apple Education field folks and customers. I am sure there are many more yet to come.

a. Home directory templates

There are two locations for the templates used to create user's home directories. One is located on the client system inside /System/Library/User Template; the other is on the home directory server in the same relative location. Root access is required to edit the template manually, or you can use command line. If you open the template for editing, you'll see two key folders - "English.lproj" and "Non_localized". You can add items, such as a pre-populated iTunes library, or iPhoto set, to the template so that new users get these items when their accounts are created.

Local users get their home directory from the local template. Network users and mobile accounts, by default, get their home directory from the home directory server's template. You can use the Details section in WGM to change the location where the mobile account pulls their home from to the local drive. This may make setting up users in a 1-1, or for large deployments, much faster by populating the local home from the local system versus pulling all the information across the network at first sync.

b. Importing users

This may be a silly question; but did you know that Workgroup Manager supports text file imports of user records? And that it has since Tiger came out? Just create a text file with the record values you need, save it with unix line endings (easiest - you can use Mac line endings but need to know the hex code), then select "Import" in WGM. I'd suggest using Passenger from MacinMind Software to format the import file first; but the raw method can work too. You will need to add a special field to be able to import passwords.

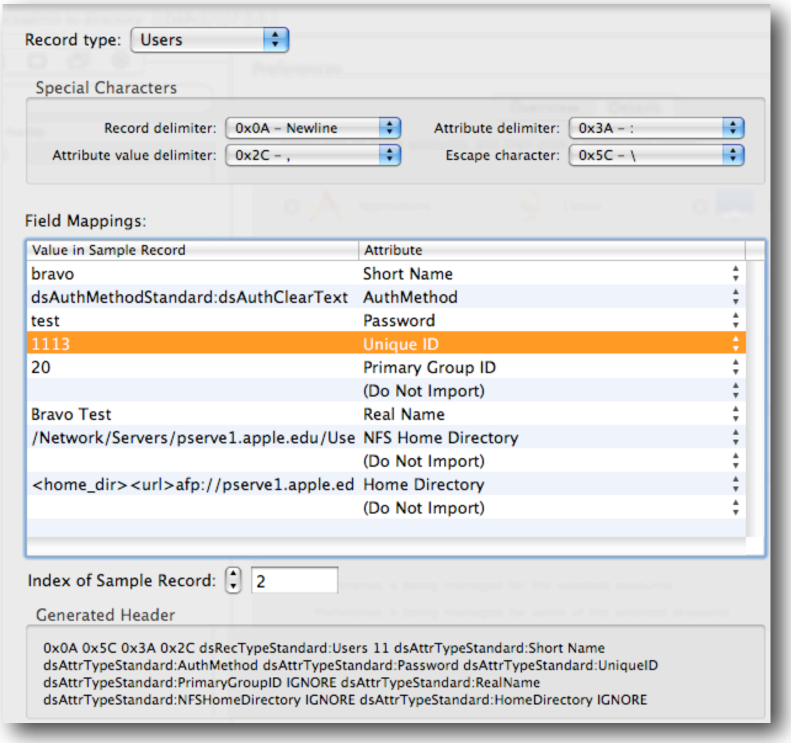
This mechanism supports importing computer records, computer groups, groups, users - the whole ball of wax. I created a simple spreadsheet in Numbers to allow me to drop user names plus a few other tidbits, and it spit out a text file. The spreadsheet is posted in my iDisk.

While I am mentioning imports, don't forget to use **Presets** with WGM. You can create a single account, then fill in all the common items that user's peers will need. Save that account as a preset, then select that preset when importing the rest of the accounts. A real fast way for activating collaboration services, Mail, and other key settings for a lot of users at once.

Here's an example set of records and the import window:

```
alpha:dsAuthMethodStandard\:dsAuthClearText:test:1112:20::Alpha Test:/Network/Servers/pserve1.apple.edu/Users/alpha::<home_dir><url>afp\://pserve1.apple.edu/Users</url><path>alpha</path></home_dir>:

bravo:dsAuthMethodStandard\:dsAuthClearText:test:1113:20::Bravo Test:/Network/Servers/pserve1.apple.edu/Users/bravo::<home_dir><url>afp\://pserve1.apple.edu/Users</url><path>bravo</path></home_dir>:
```

```
charlie:dsAuthMethodStandard\:dsAuthClearText:test:1114:20::Charlie Test:/
Network/Servers/pserve1.apple.edu/Users/charlie::<home_dir><url>afp\://
pserve1.apple.edu/Users</url><path>charlie</path></home_dir>:
```



Record type: Users

Special Characters

Record delimiter: 0x0A – Newline

Attribute delimiter: 0x3A – :

Attribute value delimiter: 0x2C – ,

Escape character: 0x5C – \

Field Mappings:

Value in Sample Record	Attribute
bravo	Short Name
dsAuthMethodStandard:dsAuthClearText	AuthMethod
test	Password
1113	Unique ID
20	Primary Group ID
	(Do Not Import)
Bravo Test	Real Name
/Network/Servers/pserve1.apple.edu/Use	NFS Home Directory
	(Do Not Import)
<home_dir><url>afp://pserve1.apple.ed	Home Directory
	(Do Not Import)

Index of Sample Record: 2

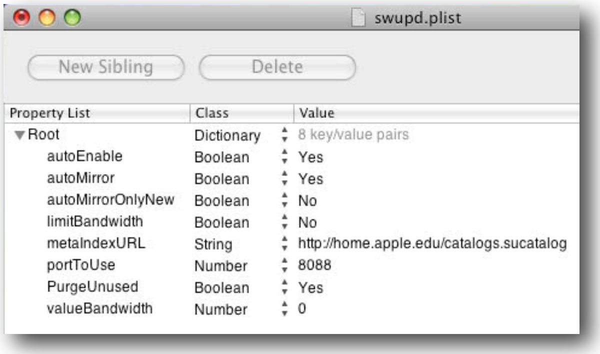
Generated Header

```
0x0A 0x5C 0x3A 0x2C dsRecTypeStandard:Users 11 dsAttrTypeStandard:Short Name
dsAttrTypeStandard:AuthMethod dsAttrTypeStandard:Password dsAttrTypeStandard:UniqueID
dsAttrTypeStandard:PrimaryGroupID IGNORE dsAttrTypeStandard:RealName
dsAttrTypeStandard:NFSHomeDirectory IGNORE dsAttrTypeStandard:HomeDirectory IGNORE
```

WGM import function

c. Software Update server (cascading too)

Setting up an internal Software Update Server on your private school network may save you many hassles. What is even better is setting one up at the district office, then setting the other Software Update Servers in the schools to get their info from the master server at district. This can be done by setting each lower server to cascade from the upper one. To set this up, log into the lower level (downstream) server as admin and edit **/etc/swupd/swupd.plist**. Set the “metaIndexURL to point to your upstream server. The URL needs to end with ‘<domain>.catalogs.sucatalog’ instead of the ‘index.sucatalog’ used in MCX for the clients. You can then point the clients to the closest SuS; but know that all servers will get the same info.



Property List	Class	Value
▼Root	Dictionary	8 key/value pairs
autoEnable	Boolean	Yes
autoMirror	Boolean	Yes
autoMirrorOnlyNew	Boolean	No
limitBandwidth	Boolean	No
metaIndexURL	String	http://home.apple.edu/catalogs.sucatalog
portToUse	Number	8088
PurgeUnused	Boolean	Yes
valueBandwidth	Number	0

Setting a downstream SuS to talk upstream

That's it for the time being. There are probably dozens of things I forgot to cover. Please make sure you review the official documentation and contact Apple Support for problems.

If I don't get a nap, I'll just have a cow - or two…