# CONTENTS

**ACKNOWLEDGEMENTS**

1. **INTRODUCTION** 5

2. **APPLE INTEL MACS IN COMPUTER LABS** 6
   - LAB USAGE STATISTICS 6
   - FUTURE OF INTEL MACS 7

3. **MAC OS X Server** 8
   - HARDWARE REQUIREMENTS 8
   - INSTALLATION 8
   - CONFIGURATION 9

4. **NETBOOT SERVICE** 12
   - MULTIPLE ARCHITECTURES 15
   - MAC FILTERING 15

5. **NETRESTORE BY BOMBICH** 16
   - NetInstall-Restore Set 16
   - CONFIGURATION 17

6. **MASTER IMAGE** 19
   - TIPS AND TRICKS 19
   - IN HOUSE SCRIPTS 19
   - THIRD-PARTY SOFTWARE 20
   - CREATING IMAGES 22

7. **IMAGING** 23
   - START NETINSTALL-RESTORE 23
   - NETBOOT IN ACTION 24

8. **APPLE REMOTE DESKTOP 3.2** 26

9. **LEVERAGING ACTIVE DIRECTORY** 28
   - BINDING TO ACTIVE DIRECTORY 28
10. Troubleshooting

Time synchronization problem between Windows and Mac

Random breakage of Active Directory

Overheating of Intel Macs on Windows XP side

11. Additional Resources
ACKNOWLEDGEMENTS

My name is Muhammad Omar Alam and I am a System Administrator at Student Affairs Technology, Weber State University. I graduated from Weber State University with a Bachelors of Computer Science with an emphasis in Network Security and Administration in spring 2009. I am proud of the fact that I have financed my entire education through scholarships and full time jobs. My five year work experience related to my field of study has enhanced my appreciation for the education I have pursued and has given me an edge in the field of Information Technology.

Three out of my five years of professional experience, were spent working at Weber State University. I have mastered computer hardware and software troubleshooting and repair. I have hands-on experience with imaging software such as Symantec Ghost, Netrestore Mac imaging, desktop security software Faronics DeepFreeze, Apple Remote Desktop, Pharos print management system, Novell services, Weber State’s network infrastructure and all of the major operating systems. I am currently managing three dual-boot Intel Mac and eleven Windows computer labs on campus. I am managing a Mac server which helps me to manage the three dual-boot Intel Mac labs.

I would like to first thank a friend Erin Hill and Professor Mali Subbiah (English Dept, Weber State University) in helping me to write this technical manual.

I would also like to thank Network Management at Weber State University, who has been really involved in getting Intel Macs working in our computer labs and the Campus Bookstore at Weber State University (Apple Authorized Dealers) who initiated a team of Apple Representatives and IT team to facilitate working of Active Directory on iMacs.

Please contact me by e-mail omaralam@weber.edu or by phone at 801-626-7485. Let me know if you have any questions or concerns.
1. INTRODUCTION

The goal of the document is to provide an understanding of how Intel Macs can be implemented in a computer lab environment. Currently not a lot of people are familiar with the Intel Macs technology. When you do a search online, there are very few institutions that have Intel Macs installed in student computer labs.

This manual is for anyone who administers a computer lab to a technician working in a computer lab. This will allow institutions to benefit from this document in setting up Intel Macs in the computer labs. This will save time for anyone in troubleshooting the problems that they might face when deploying the dual-boot Intel Macs.

Part of this document will be “exercise” oriented to get the users familiar on how all the software works on the Mac OS X server, while other parts will provide documentation specific to integrate Intel Macs in a computer lab environment. The users of this document will need two Intel Mac machines. One machine will have Mac OS X operating system installed with all available updates and on the other machine; users will install Mac OS Server and all available updates.

By using the manual, users will learn to:
- Set up a dual-boot Mac, running both Windows and Mac.
- Configure a Mac OS X server and NetBoot service
- Create a Master Image
- Deploy Intel Macs with the master image in a computer lab through the network.
- Manage a computer lab using Apple Remote Desktop 3.2.
- Leverage the full management capabilities of Mac OS X by using an Active Directory Server.
- Implement Faronics Deep Freeze in a computer lab environment.

Figure 1.1 – Union Computer Lab (Weber State University)
There are a number of reasons why Weber State University moved from PCs to iMacs in open student computer labs. One of the main reasons was to implement the multiplatform functionality in which students or faculty can choose whether to use the Windows or Mac operating system. This also gives students and faculty the opportunity to become literate in both the operating systems. Apart from the multiplatform functionality, hardware support is faster for Macs. The reason for this is that the campus bookstore is an authorized Apple dealer and also has authorized technicians who can replace hardware quickly.

Based on student input through surveys and studying student computer usage, we found that students not only needed easier access to the computers but also more desk space to spread out their books. Faculty using computer classrooms wanted classrooms in which they did not have to compete over the sound of whirring computers. Additionally, curriculums are now increasingly based on emphasis with working in teams and group projects which necessitated the need for areas in which a group of three or four students could work around a single machine.

As a result we have been looking at solutions that not only meet the needs of faculty and students but administrators as well. They had to be systems that were reliable, robust and integrated with the overall campus IT environment.

LAB USAGE STATISTICS

Below is the comparison of logins between Windows XP and Mac OS X in one of our computer labs running lab both Windows and Mac (Elizabeth Hall computer lab) in the month of February, 2009.
We had 843 individual logins on the Windows side and 774 individual logins on the Mac side. It clearly shows that students really like the multiplatform functionality in which they can become literate in two “languages,” Windows and Mac OS X.

FUTURE OF INTEL MACS

The future of dual boot iMac solution looks very promising. The machine meets a number needs for our students, faculty and staff. It provides premium desk space, a sleeker appearance, a quieter environment and the options of a varying operating system. The success of these machines in our labs depends upon the availability of Windows operating system, its smooth, easy integration, Mac support, and the cost of the machines. Over the years, the cost of a non-Mac PC has dropped considerably, which could provide a cheaper alternative in a difficult financial environment.
3. MAC OS X SERVER

First of we will need a machine with the Mac OS X Server installed. Apple has made it very straightforward to install the Mac OS X Server from a DVD or CD's. You can find detailed descriptive manuals on Apple's website which will guide you to install the Mac OS Server and get started.

http://images.apple.com/server/documentation/

In Weber State University, we use a centralized Mac OS X Server (Mac Pro), which runs on Dual-Core Intel Xeon 2.66 GHz processor and 6 GB RAM. So far, it has worked great for us. If we are imaging the machines in remote locations of campus then we use our remote Mac OS X Server (Macbook Pro), which runs on Intel Core 2 Duo 2.16 GHz processor and 2 GB RAM.

![Figure: 3.1 – Mac OS X Server](image)

HARDWARE REQUIREMENTS

To install the Mac OS Server, you need a Macintosh desktop computer or server with:
- An Intel Processor or PowerPC G5 or G4 (867 MHz or faster) processor
- At least 1 GB of RAM
- At least 20 GB of disk space available
- A secure Ethernet network

INSTALLATION

This technical manual will only help you with the installation of Mac OS X Server 10.5.4 or higher. We have purchased the Mac OS X Server v10.5.4 unlimited-client license for our departmental use. It helps in running multiple servers on campus without worrying too much about licenses.

Installing a server is not hard and you can follow the step by step instructions from the following PDF document provided by Apple:

CONFIGURATION

After the server has been installed, you need to configure the services. You can do this by going into the Server Admin Utility, which is a powerful built-in tool.

Figure 3.2 shows what the Server Admin utility looks like. As you can see it shows the overall information about the hardware, software, services and the status of the server. For this manual we have taken screenshots from our Macbook Pro running Mac OS Server.

You can easily check for “Server Updates” from this utility. As shown in Figure 3.3 you click on “Server Updates” and it will show you all the recent updates that you can download and install. Currently it shows “Remote Desktop Admin Update and iTunes update, which are ready to install. It also shows what version of the software it will be updating so you can keep track.
You can also select the services you need to configure on the Mac OS X Server by clicking on the Settings tab up at the top and then clicking on “Services”. This shows the list of services you can select to configure. The Mac OS X Server offers a wide variety of services. Since our purpose right now is to image Intel Macs over the network we will be concentrating on the “NetBoot” service. Therefore we need to select NetBoot service. However there is no harm in selecting all of the services and clicking save. This is because all the services are ‘off’ by default. You can then turn ‘on’ the service you want and configure it. NetBoot service requires DHCP. If your campus doesn’t already have a DHCP server, then you can select DHCP service to run on your Mac OS Server. However in our case, Weber State University has a DHCP server so we don’t have to use the DHCP service.
For our purposes we only selected NAT, NetBoot and NFS services. We have dedicated, as shown in the figure, this server as a NetBoot server. So far it has been doing a good job.

In our next section we will configure the NetBoot service and get down to business.
4. NETBOOT SERVICE

NetBoot service is a built in service in a Mac OS X Server. Thanks to Server Admin Utility by Apple, it is very easy to manage this service. Figure 4.1 shows the default view of the NetBoot service. As you can see there is only one Netinstall Mac OS image. This is because we use this server only for one of the computer labs.

![NetBoot Service](image)

Figure 4.1 – NetBoot Service

You can have more than one Netinstall Mac OS X image depending on the architecture of the Intel Macs.

You can also view the log file by clicking on the “log” file. This is very useful for troubleshooting purposes. You can set the log detail level as well. Server Admin doesn’t have the ability to modify this file. You can manage this file by removing it after a set time period. Deleting the client database file is very easy. You just have to navigate to /var/db/ and in it there is a file called bsdpd_clients. This is the client database file that stores known information about clients.
To configure the NetBoot service, click on the “Settings” tab. The first tab you will see is “General”. This will give you an option to select which port you want the NetBoot service turned on to. For our purpose we are creating a NetBoot server so we need the Ethernet port to be selected as shown in Figure 4.2 above.

You can also select where to put images and client data. This is very important as NetBoot service requires a place to store the Images and client data unless it won’t work. As you can see from the figure, it shows two volumes: Macintosh HD and Imaging. Macintosh HD is the local hard disk in this server and Imaging is the external hard drive. We have selected “Macintosh HD” to store our images and client data. Once you select these it will automatically create a “NetBoot” folder in the “Library” folder. Inside the NetBoot folder, it created two folders: NetBootSP0 and NetBootClients0 as shown in the figure 4.3 below.
In the NetBootSP0 folder, the diskless booting .nbi folders or NetInstall-Restore sets will be saved. In the next section you learn on how to create the NetInstall-Restore sets. The NetBootSP0 is like a sharepoint and it’s automatically shared when you start the NetBoot service.

Under “Images” tab you will see a list of bootable sets located in the NetBootSP0 folder. Each image has its own options. As shown in Figure 4.4 you can enable which image to use. As you can see there are two NetInstall Images, each one having a unique image index and a protocol. By setting an image to default, an image has the top most priority and the client will boot to it when a user holds the N key at startup. You can also change the image index number greater than >4096 for NetBoot server load balancing.
MULTIPLE ARCHITECTURES

Under Images it gives you an option to select architecture. NetBoot is compatible with the Intel architecture. You can create a Netinstall-restore set specifically for a PowerPC architecture. We don’t have any PowerPCs in our computer labs and we only have the Intel architecture. Therefore we have to set the architecture to “universal” and it works great.

MAC FILTERING

Sometimes there are fully automated images that install software. Unfortunately, if a client anywhere on the local subnet were to NetBoot by holding the N on startup, it could erase the client’s hard disk on accident. To prevent this type of accident we can set up Mac Filtering under “Filters” tab. In our NetBoot server we have selected the “Deny only clients listed below (allow others)” option. We have added the server Mac addresses and also our fellow staff members’ hardware addresses to prevent NetBooting on accident.

Now when we are familiar with the NetBoot service, we will look at different software we use for creating the NetInstall-restore sets and images.
5. NETRESTORE BY BOMBICH

Netrestore (http://www.bombich.com) is free software we use for imaging. This software is no longer supported by Bombich, but it still works great. There are two applications that come with NetRestore. One is the “NetRestore” and the other is “NetRestore Helper”. NetRestore is loaded through the NetBoot service using the server’s NetInstall-Restore set (.nbi file) which is created through NetRestore Helper. Confused? Don’t worry we will go over this in detail in this section.

**NETINSTALL-RESTORE SET**

This is very simple to build. First we need to launch the application called NetRestore Helper which is found in the same directory as the NetRestore. In the figure 5.1, it shows the NetRestore Helper application. As you can see there are two tabs at the top. We are not creating the Master Image yet; right now we are creating a “NetInstall Set” so we will select the tab called “Create NetInstall Set”.

![NetRestore Helper](image)

**Figure 5.1 - NetRestore Helper**

Then we will enter in a name of the NetInstall Set. As you can see from the figure we called our NetInstall-Restore set “wsunetrestore”. Then we will give it an Image ID of more than 4095 for load-balancing. Even though we do have load-balancing, putting an image >4096 will not have adverse effects. Then we will give it a description. In this case we will just say what date it was created and why. These descriptions can come handy when you have multiple NetInstall-Restore sets running on the same
NetBoot server. Then click on the “Save NetInstall-Restore Set” and choose the location /Library/NetBootSP0 folder to save as shown below in figure 5.2.

![NetBootSP0 Folder](image)

**Figure 5.2 – NetBootSP0 Folder**

### CONFIGURATION

Once the NetInstall-Restore set has been built, it will prompt you with an alert saying “would you like to modify NetRestore’s configurations and default preferences now?” You can always modify NetRestore’s configuration later but for right now we can configure it quickly and put a description as shown in the figure 5.3.
You can select the configuration you want depending on your network and click on Save. Once you have fully configured the NetRestore you can test out a client and press N on startup. If it loads up to the NetInstall-Restore set showing the following figure 5.4 of NetRestore, which means you have configured this correctly.

In our next section, we will look at how to create Master Images and then to deploy them using NetRestore.
6. MASTER IMAGE

We have to keep in mind that master images are architecture specific. If you build the image on PowerPC architecture then you cannot deploy it on the Intel architecture machine. This section will give you detailed guidance plus tips and tricks on how to create images on both Mac OS X and Windows XP using NetRestore Helper.

TIPS AND TRICKS

- Always use the best Mac model available
- Always install the operating system from scratch
- Use freshly formatted hard disks
- Always install all the updates and software needed
- Extensively check the software list and compare it with the master Image.

IN HOUSE SCRIPTS

There are number of custom resolutions we use and they are all listed in detail below:

- Ad-bind-script - This script binds the iMac to Active Directory and configures advanced options of the Active Directory plug-in. This script needs to be run only once after the iMac has been imaged. This script also deletes itself after it runs for security purposes. This script runs through Active Remote Desktop in Mac OS X Server. You can view the script in section “

- Rc.local script – This script is used to fix the issue of time synchronization between Windows and Mac. This script runs before the user logs into the Mac side. It refreshes the “networktime” service properly and enables the user to log in to the Mac side without any delay. You can view the script in the last section of this document in “Troubleshooting” in Figure 10.1.

- Mac-login-script - This script is used for recording lab statistics by talking to the SQL database. This script is also used for collecting information about all login activity for a specific user. It is run by the “mac-login.plist” file after the user logs in to the machine.

- Mac-logout-script - This script is used for recording lab statistics by talking to the SQL database. This script is also used for collecting information about all logout activity for a specific user. This script runs when the user logs out of the machine.

- Startup-script - This script takes the username as an argument, then removes any old default home directory, restores a fresh copy of the default home directory, then “chowns” it to logging in user. You can view what the script looks like in Figure 6.1.
THIRD-PARTY SOFTWARE

There is a number of third party software we include in our master image.

Faronics DeepFreeze – This is licensed software we use in both of our operating systems: Windows and Mac. It makes the operating system fully ‘bulletproof’, as it freezes the computer so whatever the student downloads (malicious software, viruses etc.), or if the student changes configuration the computer, it is wiped out after the computer reboots. It gives the students a trouble free computing environment. The DeepFreeze settings screen is shown in the Figure 6.2 below.
Refit – This is an open source program and we use it as a boot menu, to give students a choice to choose between Windows and Mac. We have fully customized it to fit our needs for example, the background image and the icons as shown in the Figure 6.3.

Figure 6.2 – DeepFreeze Settings

Figure 6.3 – Refit screen (comes up on start up)
CREATING IMAGES

We will use NetRestore Helper, the same program we used earlier to create the NetInstall Set. Therefore we will click on the tab for “Create Master Image”. We will select the master disk as “Macintosh HD” (for Mac OS X master Image) or “Windows XP” (for Windows XP master image). We will leave the format as “Read-only compressed” and we can leave the other options as default as shown in Figure 6.4.

![Figure 6.4 - NetRestore Helper](image)

We will then click on the padlock and enter the Admin password. This will prompt us with a location where to store the master image. This will create an ASR (Apple Software Restore) ready master image. It is a compressed .dmg disk image file.
7. IMAGING

Once we have the master image, we now deploy the master image across the network to other machines. There are other methods of imaging as well; one way is to image is using external hard drives. However that process is very tedious, and it’s still followed by some of the people today because they don’t want to spend time learning the other way. Imaging over the network isn’t some kind of rocket science, especially not with Intel Macs. It’s very easy. In this section you will learn imaging over the network using NetInstall-Restore.

At this point, you should have the Mac OS X Server using NetBoot service already running on the same subnet as the lab machines.

START NETINSTALL-RESTORE

First of you will need to NetBoot your lab machines using three following options:

- Go to System Preferences>Startup Disks. Once you are there, locate your network startup disk and click on Restart. This step sets the new startup disk as Network Volume. This is shown in Figure 7.1.

- Hold down the “N” key as the computer starts and release the “N” key when you see the globe flashing on the screen.

- Hold down the “Option” key as the computer starts, then select the globe icon and it will start Net Booting.
If the NetBooting is slow on your client machines, then consult this with your network administrator to increase network capacity. In Weber State University, we are running gigabit network in the labs so it’s really quick to image the machines.

NETBOOT IN ACTION

Once booted from the server’s Netinstall set, you will be in the Netrestore screen. You will see the same configuration you set up earlier in section “Netrestore by Bombich”. You can also have access to Disk Utility if you want to partition the hard disk, this is useful when pushing the Windows image.

We will select the target disk and the master image we created and we simultaneously push the image across the network as shown in the Figure 7.2.

![Figure 7.2 – Intel Macs being image over the network (Union Computer Classroom)](image)

Once the image has been pushed, the lab machines will reboot.

In summary these are the steps that we currently take to image the Intel Macs with a dual-boot environment are below.

- NetBoot all the computer lab machines.
- Once they are all Netbooted, choose the target disk and the master image created and simultaneously push it.
- Once the image has been pushed, the lab machines reboot.
- Then change the computer names and bind all the computers to Active Directory using the “Ad-bind-script” (We will talk about Active Directory in the next section)
- Then partition the hard drive by using disk utility. To automate the process we use licensed software called Apple Remote Desktop.
- Once the hard drive is partitioned, again NetBoot the lab machines and this time push the Windows image.
- Once the Windows image is pushed freeze the Windows side by the help of DeepFreeze Console.
- Then in the end, also freeze the Mac side by installing the DeepFreeze client using Apple Remote Desktop.
8. APPLE REMOTE DESKTOP 3.2

Once the computers are imaged, you need to rename each computer, bind each computer to Active Directory, install Deep Freeze for desktop security, maybe even install or update software in the future. Overwhelmed? Don’t worry Apple Remote Desktop 3.2 (ARD) makes this very easy.

It runs about $499 for unlimited managed systems which will make your life very easy. Installing ARD is very simple. To do this just follow the complete guide by Apple below and simply install the ARD onto the same Mac OS Server running NetBoot service.


Once it is installed you can easily add the computer labs for easier access as shown below in the figure 8.1.

![Figure 8.1 – Apple Remote Desktop 3.2 installed on the Mac OS X Server](image)

As you can see from the figure, there are currently four computer labs added onto ARD: Union Lab, Union Classroom, EH Lab and WC Lab.
As you can see from figure 8.1; on the left side pane, there are a number of custom scripts. We have created these scripts to easily manage the machines in computer labs. For example, to install DeepFreeze on one of the computer lab machines, we have just created a simple task as shown below:

![Install DeepFreezeMac for All Labs](Figure 8.2 - Install Package (Apple Remote Desktop 3.2))

This will install the DeepFreeze package file onto the target computer. We have selected the option “Force an immediate restart” so after the application install, it restarts the computer. Therefore we could install any application within a matter of minutes using ARD on 50+ machines.
9. LEVERAGING ACTIVE DIRECTORY

One of the major problems going live with iMacs in the computer labs was in implementing our existing Active Directory environment. We were the first ones on campus to implement Active Directory with Intel Macs. Therefore we faced a lot of initial issues in the beginning.

In summer of 2008, WSU Bookstore (Apple Authorized dealers) initiated a team of Apple representatives and WSU Network Management to facilitate working of Active Directory on iMacs. WSU Network Management purchased a new server and added it to the existing Active Directory servers. At this time, they were also able to synchronize Novell e-directory and Active Directory with each other. In fall 2008, we were given direct access to one of the Active Directory servers where we can make sure the Intel Macs are binding correctly using their hostnames.

This was a major obstacle as we wanted students to login to the iMacs using their WSU usernames and passwords.

BINDING TO ACTIVE DIRECTORY

Intel Macs come with a utility called “Directory Utility” located in Utilities. We use the script “ad-bind-script” mentioned earlier in section “Master Image”.

![Directory Utility](image)

Figure 9.1 – Directory Utility
This script binds the iMac to Active Directory and configures advanced options of the Active Directory plug-in. This script needs to be run only once after the iMac has been imaged. This script also deletes itself after it runs for security purposes.

Let’s view what the script looks like. Thanks to Bombich we didn’t have to write this script ourselves. The script is located on Bombich’s website (http://www.Bombich.com). Let’s take a look what the script looks like in Figure 9.2
# This script binds to AD and configures advanced options of the AD plugin
# As this script contains a password, be sure to take appropriate security
# precautions
#
# A good way to run this script is to set it as a login hook on your macOS
# because it only needs to be run once, the last time this script does is to delete
# itself. If you have another login script that you typically run, include the
# script on your macOS and indicate its path in the `newLoginhook` variable.
#
# If running this as a one-time login hook to bind to AD after imaging,
# be sure to enable auto-login (for any local user) before creating your macOS
#
# Host-specific parameters
# computerid should be set dynamically, this value must be machine-specific
# This value may be restricted to 16 characters! The only entry you will receive upon entering
# an invalid computer id is to the effect of not having appropriate privileges to perform the requested
# operation
#computerid="/Users/LocalHost | awk '{print $2}'; # MAC Address
#computerid="/Users/LocalHost | awk '{print $2}'; # MAC Address
#
# Standard parameters
domain="example.domain.com"  # fully qualified DNS name of Active Directory Domain
user="*"  # username of a privileged network user
password="********"  # password of a privileged network user
command="/usr/sbin/ad-bind.sh"  # command to run
command="/usr/sbin/ad-bind.sh"  # command to run
command="/usr/sbin/ad-bind.sh"  # command to run
command="/usr/sbin/ad-bind.sh"  # command to run
command="/usr/sbin/ad-bind.sh"  # command to run
command="/usr/sbin/ad-bind.sh"  # command to run
command="/usr/sbin/ad-bind.sh"  # command to run
command="/usr/sbin/ad-bind.sh"  # command to run

# Advanced options
allowdomains="enable"  # 'enable' or 'disable' automatic multi-domain authentication
localname="enable"  # 'enable' or 'disable' force hostname to local drive
protocol="smtp"  # 'smtp' or 'ssh' change how name is entered from server
mobile="enable"  # 'enable' or 'disable' mobile account support for offline login
mobileconfirm="disable"  # 'enable' or 'disable' warn the user that a mobile acct will be created
unmappedpaths="disable"  # 'enable' or 'disable' use AD ORVName attribute to determine the home dir
user_shell="/bin/bash"  # e.g., /bin/bash or 'none'
preferred="none"  # Use the specified server for all Directory lookups and authentication
# (e.g., "-nopreferred" or "-preferred"
server=server.edu"  # e.g., "-server.edu"
admingroups="APPLE\admin"  # These come-separated AD groups may administer
# the machine (e.g., "\" or "APPLE\admin")
packetsign="allow"  # 'allow' or 'disable' require
packetcrypto="allow"  # 'allow' or 'disable' require
packetsize="384"  # number of bytes
namegoes="domain"  # 'forest' or 'domain'

# Login hook setting -- specify the path to a login hook that you want to run instead of this script
newLoginhook="/Library/Management/login.sh"  # e.g., "/Library/Management/login.sh"

### End of configuration

# Activate the AD plugin
defaults write /Library/Preferences/DirectoryService/DirectoryService "Active Directory" "Active"
plutil -convert xml /Library/Preferences/DirectoryService/DirectoryService.plist

# Unbind from AD
### Uninstall the following line if the computer is already joined to the domain
# defaults write /Library/Preferences/com.apple.loginwindow "kDCInteractive" "false"
defaults write /Library/Preferences/com.apple.loginwindow "kDCInteractive" "false"

Figure 9.2 - Ad-bind-script
Figure 9.2 – Ad-bind script (Continued)
This script needs to run after imaging. Therefore it can be very tedious to run the script individually on each machine. ARD saves the day again and we can run the script through Active Remote Desktop as shown in figure 9.3 to as many machines as you want all at once.

Figure 9.3 – AD Bind Script sent through ARD

Once the script runs, the computer name is added to the campus Active Directory which is in sync with e-directory.
10. TROUBLESHOOTING

TIME SYNCHRONIZATION PROBLEM BETWEEN WINDOWS AND MAC

The main issue that arose when we went “live” was leveraging iMacs with Active Directory, which was very unreliable in logging in on the Mac side. This was caused mainly because of the time synchronization problem between Windows and Mac. We noticed that the time wasn’t in sync with Windows. After extensive troubleshooting, storing time information on the Windows side, custom registry fixes, custom scripts, ntpd-wrapper, plists on the Mac side - we finally implemented a “rc.local” script on the Mac side and that fixed the issue.

```
#!/bin/bash
killall ntpd
/usr/sbin/ntupdate
RETRY=$?
TRIES=1
sleep 1
until [[ "RETRY" =~ "0" || "TRIES" =~ "5" ]]; do
  /usr/sbin/ntupdate
  RETRY=$?
  sleep 1
 TRIES=$((TRIES + 1))
done;
SystemStarter -q start "Network Time"
./System/Library/StartupItems/NetworkTime/NetworkTime start
```

Figure 10.1 – rc.local script

This script runs before the user logs into the Mac side. It refreshes the “networktime” service properly and enables the user to log in to the Mac side without any delay. This script needs to be placed in /etc folder.

RANDOM BREAKAGE OF ACTIVE DIRECTORY

After overcoming this obstacle we faced another problem with the Active Directory. This time it was the 14 day breakage of Active Directory after the iMacs have been bound and imaged. This problem was caused by Active Directory binding script and third party software called Faronics DeepFreeze, which is
used for desktop security. Again after extensive troubleshooting we were able to modify the Active Directory binding script and make it to work and not break after the computer lab has been imaged.

OVERHEATING OF INTEL MACS ON WINDOWS XP SIDE

A third issue that arose was the heating up of the Intel Macs while booted into Windows operating system. We tried different remedies to solve the issue (power save mode) but they didn’t help. We found third party software called “smcFanControl” to solve this issue. This free software lets us change the speed of the built in fans so we can increase it a bit to keep the iMacs run cooler. The settings are shown in the Figure 10.2 below.

Figure 10.2 - smcFanControl
11. ADDITIONAL RESOURCES

APPLE’S HOME PAGE
➢ http://www.Apple.com

APPLE REMOTE DESKTOP
➢ http://www.apple.com/remotedesktop/

PROFILE FOR WEBER STATE UNIVERSITY ON APPLE’S WEBSITE

NETRESTORE BY BOMBICH
➢ http://www.Bombich.com

FARONICS DEEPFREEZE

DEPLOY STUDIO
➢ http://www.DeployStudio.com

REFIT
➢ http://refit.sourceforge.net/

SMCFANCONTROL
➢ http://www.eidac.de/