

OS X 10.6 Snow Leopard and eDirectory

Joe Jenkins / Davis Tool Inc / Rev .02 / Nov 2009

This document is a general outline of our attempts at getting Mac OS X clients running Snow Leopard to authenticate to eDirectory and mount their network home directories over AFP. This has been documented in pieces in several places, and because of that it took us a bit of time to get all the pieces working properly together.

Our environment is primarily a Windows / Linux client network that uses Novell Netware 6.5SP8 for file and print services, and Edirectory 8.8SP5 for Directory Services. We have not tested this with OES2 running on Linux, but I am sure most of it applies.

Recommendations:

- Get your Netware server fully patched and up to date, including NMAS and Edirectory.
- As of this writing, we needed to use a Novell FTF for AFP to enable it to work properly with 10.6.x Snow Leopard. Check the Novell download section under Netware for more information.
- Enable AFP on your volumes, we have it set to case-insensitive and AFP logging on (useful when setting up to spot problems) The Novell NFAP documentation covers this.
- Make sure your server and volume names make sense and are named well in the SYS:\etc\AFPVOL.CFG – one thing we discovered is that what you name your servers and volumes in this file is critical when you create mount objects for these volumes later in Edirectory. An example entry in ours looks like SERVER.SYS “SYS” (we only have one server, so we map SERVERNAME.VOLUME to “VOLUME” - You can also do SERVERNAME.VOLUME “SERVERNAME.VOLUME” if you have more than one server. Don't use spaces in the names in quotes, and remember that whatever the name is here, it will have to match on mount objects you create in Edirectory.
- Make sure your SYS:\etc\ctxs.cfg lists all OU's where you want the system to look for user objects when mounting AFP volumes. They are listed in this file as ou.ou.treename for example.
- Set up Universal Password, create a policy and make sure your OS X users that will auth against Edirectory are assigned to this policy. You can use simple passwords but they are far less secure and are sent in cleartext over your network.
- From a Mac client logged in as a local user, make sure you can mount your Novell volumes via AFP with Finder. In Finder, you can use Command-K or “Connect to Server” to connect to AFP enabled volumes. Use an afp://server.domain.com/ style URL. You should be asked for your Novell login and password when you go to mount, if the mount fails, you can use DSTRACE to monitor NMAS or enable AFP logging on when you load the afptcp.nlm on your server.

Extending the Edirectory Schema

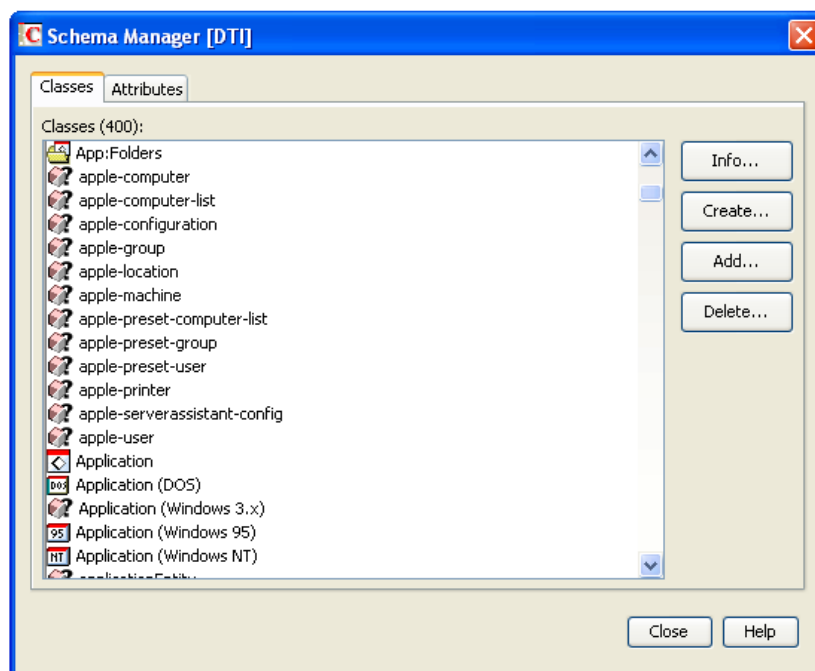
In order for Mac OS X computers to act as clients on a Novell network, they need to be able to get useful information from the directory. Some of this information already exists in Edirectory but to make an OS X machine really work as a proper client, it's best to extend the schema to include Apple-specific schemas. Extending the schema is a much safer option to reusing existing Edirectory attributes to store apple specific values. The Apple schema can be found on any Mac OS X machine in /etc/openldap/schema/apple.schema

In order for this to be useful to Edirectory, it has to be converted to an LDIF file before it can be imported into Edirectory. A working LDIF file will be included with this document as I do not know how to convert a schema file into a functional LDIF at this time.

There are several ways to extend the schema, we used the ConsoleONE utility on a Windows PC. In ConsoleONE, if you have the proper NDS snap-ins installed, you can log into your tree and load the NDS Import / Export wizard and import an LDIF schema into your directory. For information on doing this, consult the Novell Edirectory documentation here:

<http://www.novell.com/documentation/edir88/>

Once properly imported without any errors, you can go into the ConsoleONE schema manager and view all the newly created Apple-specific classes and attributes. Make sure they are all there (side by side comparison to the apple.schema is useful) (**NOTE:** It is a good idea to use the ConsoleONE Schema manager to check your existing classes and attributes before an import, make sure none of the new ones in the the Apple schema already exist in your Edirectory schema, if they do, they may already be in use for something else, but chances are, they won't be there. I don't recommend deleting or modifying classes and attributes in the schema unless you absolutely know what you're doing.)



Setting up the Home Volume and Mount Object

In order for Mac clients to log in via a network, they need someplace to store the user's home directory, preferences, etc. Typically this will be on a network volume, and this is what I will be describing. When a user account is created in Edirectory, you also specify which volume and what directory their home directory will exist in.

For Mac OS X, we created a new volume for their home directories called it MACHOME. Next, in the same OU as the server on which this volume resides, we created a new 'mount' object (mount is a class that was created in your schema during the extension outlined above)

When you create a new mount object, name it as SERVER:/VOLUME (we use IP's in the mount objects, we found that hostnames didn't work here for us during login even though they resolved properly via DNS.) This is also where the names in AFPVOL.CFG are very important. In our case, we have the following line in our AFPVOL.CFG:

```
SERVERNAME.MACHOME "MACHOME"
```

Had we used SERVERNAME.MACHOME "SERVERNAME.MACHOME" the name of our mount object would have had to be 192.168.3.6:/SERVERNAME.MACHOME

You can check your AFP mount names in Mac OS X with a Command-K in Finder and connect via AFP url to your server. Once authenticated, it will list out what AFP mount names are available. If they are wrong, or have spaces in them, you will need to do a rename and reload of the AFPTCP.NLM – the instructions for this are in the AFPVOL.CFG file or in the NFAP documentation on Novell's website. (Other useful AFP console commands can also be found here!) From my experience, all AFP mounts on a Novell file server can be mounted in OS X without creating this mount object, but in order to use a volume as a home directory upon network login, this is absolutely critical. Once the object is created, you will need to go into the properties and add several attributes, shown below. These attributes are part of the schema extensions done above.

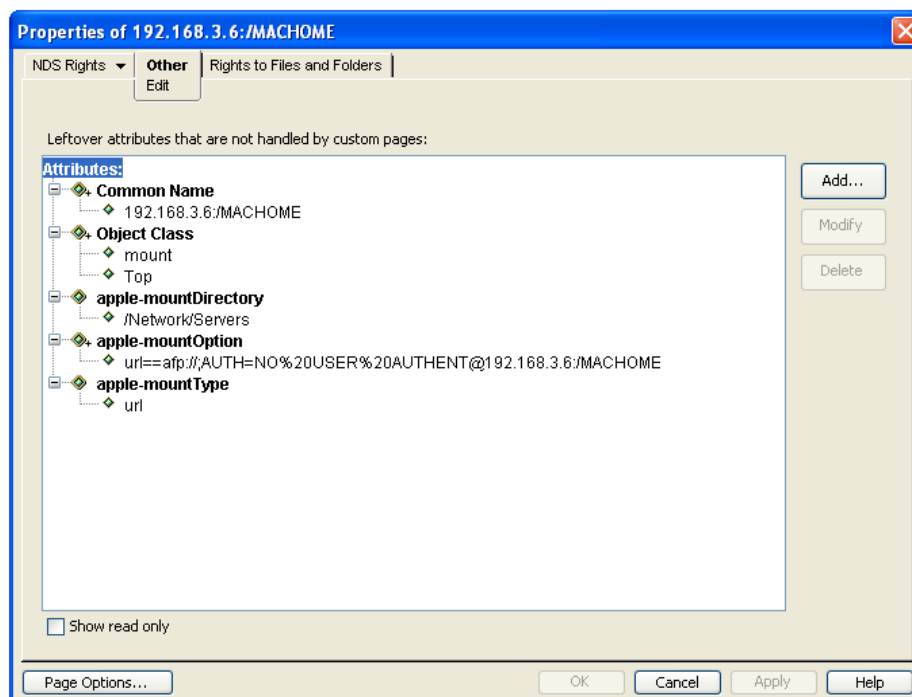
Apple-mountDirectory: /Network/Servers

This is where on a local Mac OS X machine network home directories will be mounted. For example, if my username is Joe, and my server is 192.168.3.6, and my home directory is Joe in MACHOME, it would map it as /Network/Servers/192.168.3.6/MACHOME/Joe

apple-mountOption:

url==afp://;AUTH=NO%20USER%20AUTHENT@SERVERIP:/VOLUME

apple-mountType: url

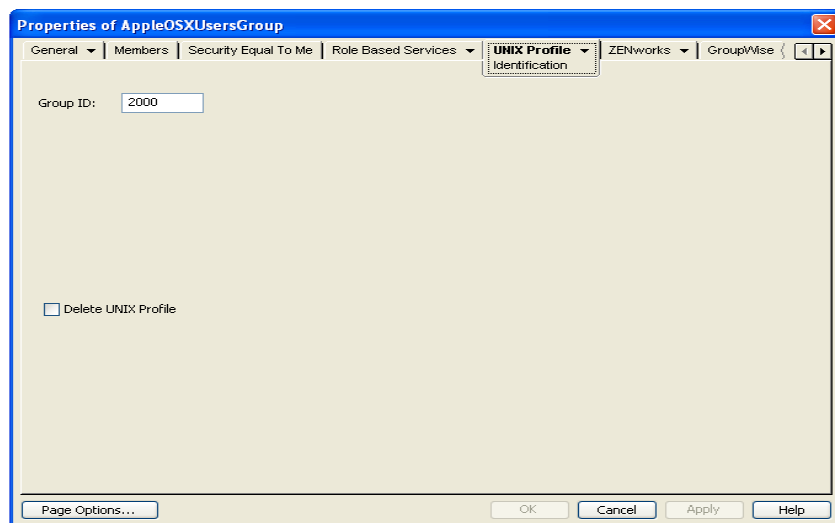


Users and Groups

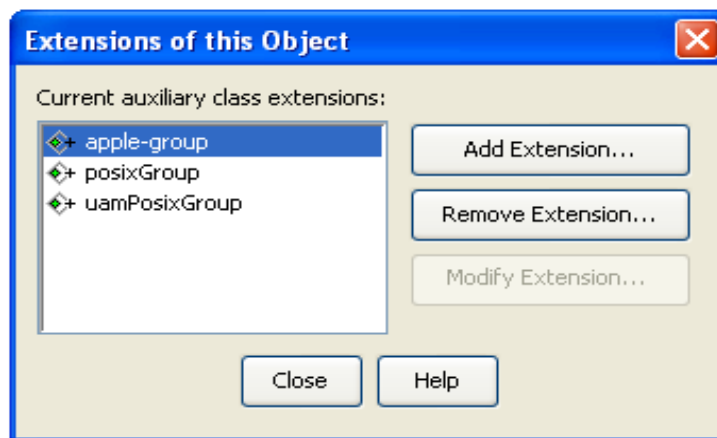
When you authenticate on a Mac OS X client to Edirectory / LDAP certain mappings take place, the most important of these are Users, Groups and Mounts. The values found in Edirectory via LDAP are then mapped to Mac OS X friendly versions (via the Open Directory utility setup, discussed later.) A user object in Edirectory has to be properly set up and added to a Universal password policy before it can be used to log in to a Mac OS X client.

We create our own test OU in Edirectory called Apple and then created a new group object. We gave it a group ID (under Unix Profile) of 2000. This group ID will be added to new users that will be using Macs. You can give it any Group ID you want but be wary, Mac OS X is BSD Unix based, and several groups are already in use (take a look at /etc/group for ones that are already used on any Mac)

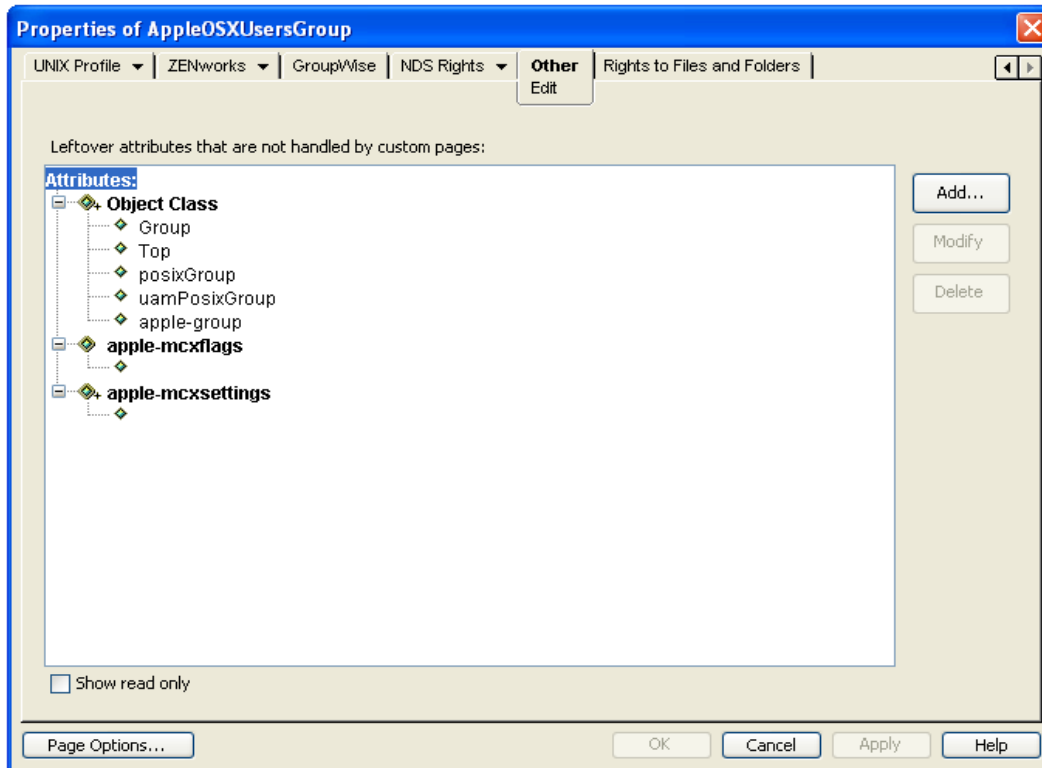
Below are the ConsoleONE views of a group object in Edirectory that we have set up for use:



Right click on the group object and select "Extensions of this Object" and add "apple-group" - your group object's extensions should look like this:



We also use MCXSETTINGS and MCXFLAGS (used for managing system preferences on Macs on a per user, per group, per machine or per machine list basis, discussed later in this document) To do this we add a couple of attributes to the group properties under the Other tab:

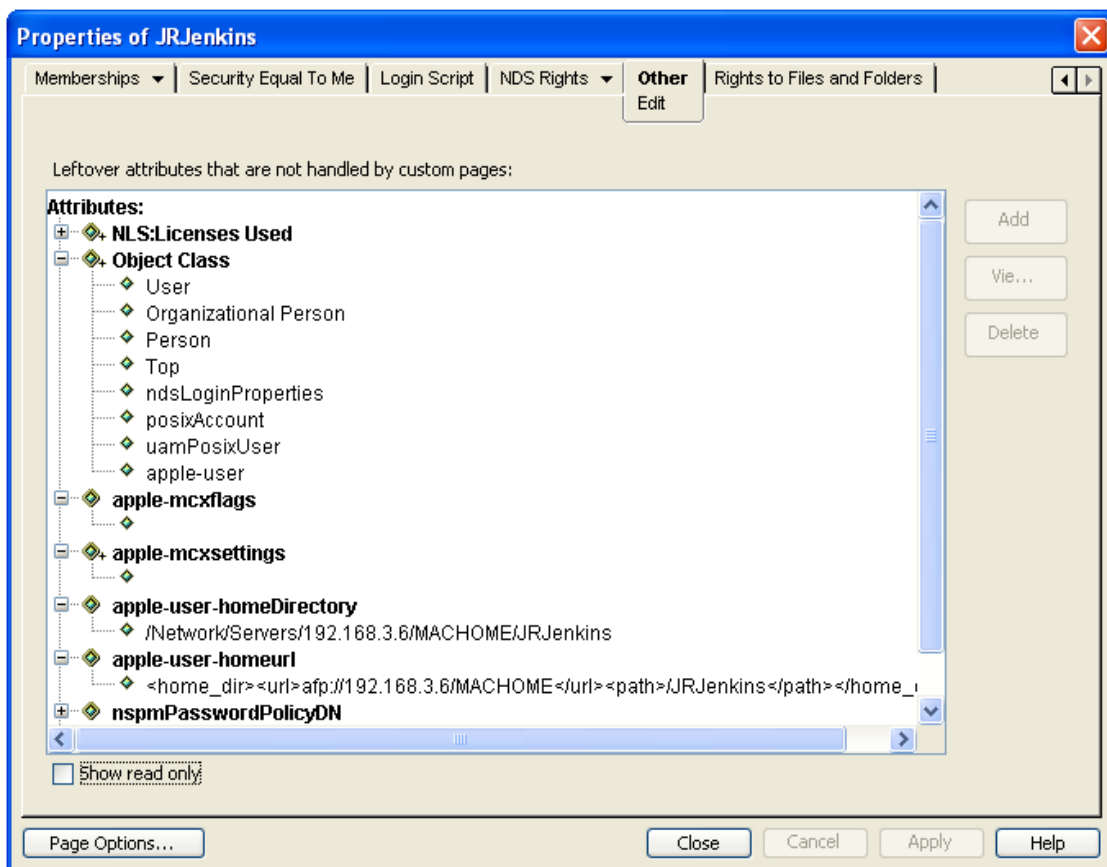


These entries are blank for now because they will eventually be managed via the Workgroup Manager in OS X.

Now, to create a new user in Edirectory that is able to use a Mac on our network, we follow these steps:

- Create a new user object in the appropriate OU.
 - “Name” is their login name, also fill in their “Surname”
 - Primary Login Sequence: NDS (depending on your environment)
 - Create Home Directory: Yes – specify server and volume of user's home directories that correlate to the mount object.
 - Assign NDS Password: Yes – Prompt During Creation (set it now, it cannot be set during initial login as with a Novell client)
 - Define additional properties: Yes
 - Once in the properties page, fill out the General tab with as much information about the user as possible, we find this is helpful in keeping these values available for Mac OS X and the directory utilities.
 - Unix Profile tab – assign a unique user ID to the user – do not reuse ID's, especially if several users share a machine!
 - Assign them the appropriate apple user group already created.
 - Login shell: /bin/sh or /bin/bash
 - Home Directory: /home/Username (not used for Macs with our LDAP mapping)

- Close the properties page of the user and right click on the new user's object, select “Extensions of this Object” and then Add “apple-user”
- On the “Other” tab, select attributes and then “Add” - add the following:
 - apple-user-homeDirectory – set it to /Network/Servers/SERVER/VOLUME/HomeDir where SERVER is the server's IP, the VOLUME is the volume name of the mount object, and HomeDir is this user's home directory on that volume.
 - apple-user-homeurl – set it to
`<home_dir><url>afp://SERVER/VOLUME</url><path>/HomeDir</path></home_dir>`
 this will obviously vary based on your situation.
 - apple-mcxflags and apple-mcxsettings – leave these blank, they will be managed by Workgroup Manager as outlined later in this document.
- Add the new user to your Universal Password Policy in iManager – check the documentation at Novell's site for how to set all this up. You may also want to reset their Universal Password in iManager, this seems to work better than trying to do it in ConsoleONE sometimes.



If everything is correct, this user can now login to a properly configured Mac OS X machine.

Setting up a Mac OS X Client Machine for authenticating to Edirectory

Now that the schema is extended, the mount object is created, and the user object is configured, the user should now be able to use a Mac with Edirectory. The Mac will have to be configured to use Edirectory first, as outlined below. Log into the Mac with a local administrator account, or make sure you have the login for an administrator account on that machine available.

In OS X, this is managed in the directory utility. In 10.5 Leopard, this can be found in Applications → Utilities, Directory Utility. In 10.6 Snow Leopard, this has been moved to System Preferences → Accounts → Login Options. Click “Edit” next to Network Account Server and click “Open Directory Utility”

Before setting up a connection to LDAP via the Directory Utility, it is necessary to copy a working template file into ~/Library/Application Support/Directory Access/LDAPv3/Templates/ so that Directory Utility will have a copy of the LDAP mappings available for this new type of connection. I have created EdirectoryMapping88SP5.plist for this purpose. I will not go over the individual mappings here, but you can open the file with a PLIST editor and see the structure and what maps to what.

Once the template file is in place, go into Directory Utility and under Services, go into LDAPv3. Click on “New” and enter the information relating to your server (IP or DNS name, SSL) Make sure “Use for Authentication” and “Use for Contacts” is checked. Click “Continue” and once it finds your server, Directory Utility will ask for a template, select the EdirectoryMapping88 from the drop-down and then fill in the search base for your organization in the form of ou=ou, o=org etc. Since we want our entire tree searched for users etc during login, we use o=treename (where treename is the tree in Edirectory.) Click Add and allow it to be added to your LDAP services list. Give it a unique configuration name, highlight the entry and then click edit. Make sure the security settings / SSL etc are all set according to your environment. You can also browse all the LDAP mappings while in the edit mode. In some cases it is necessary to use an LDAP proxy user to authenticate for directory access, this can be set up under the Security tab.

Reboot the Mac and once it comes back up, log in again with a local user account, and pull up a Terminal (under applications, Utilities)

Use dscl to test your LDAP settings: (here I am reading the info on the mount object)

```
mac:~ joe$ dscl
Entering interactive mode... (type "help" for commands)
>
> ls
BSD
LDAPv3
Local
Search
Contact
> cd LDAPv3/
/LDAPv3 > ls
192.168.3.6
/LDAPv3 > cd 192.168.3.6/
```

```

/LDAPv3/192.168.3.6 > ls
ComputerLists
Computers
Config
Groups
Mounts
People
PresetGroups
PresetUsers
Users
/LDAPv3/192.168.3.6 > cd Mounts
/LDAPv3/192.168.3.6/Mounts > ls
192.168.3.6:/MACHOME
/LDAPv3/192.168.6/Mounts > read 192.168.3.6:VMACHOME/
dsAttrTypeNative:apple-mountDirectory: /Network/Servers
dsAttrTypeNative:apple-mountOption: url==afp://;AUTH=NO%20USER
%20AUTHENT@192.168.3.6:/MACHOME
dsAttrTypeNative:apple-mountType: url
dsAttrTypeNative:cn: 192.168.3.6:/MACHOME
dsAttrTypeNative:objectClass: mount Top
AppleMetaNodeLocation: /LDAPv3/192.168.3.6
RealName: 192.168.3.6:/MACHOME
RecordName: 192.168.3.6:/MACHOME
RecordType: dsRecTypeStandard:Mounts
VFSLinkDir: /Network/Servers
VFSOpts: url==afp://;AUTH=NO%20USER%20AUTHENT@192.168.3.6:/MACHOME
VFSType: url
/LDAPv3/192.168.3.6/Mounts >

```

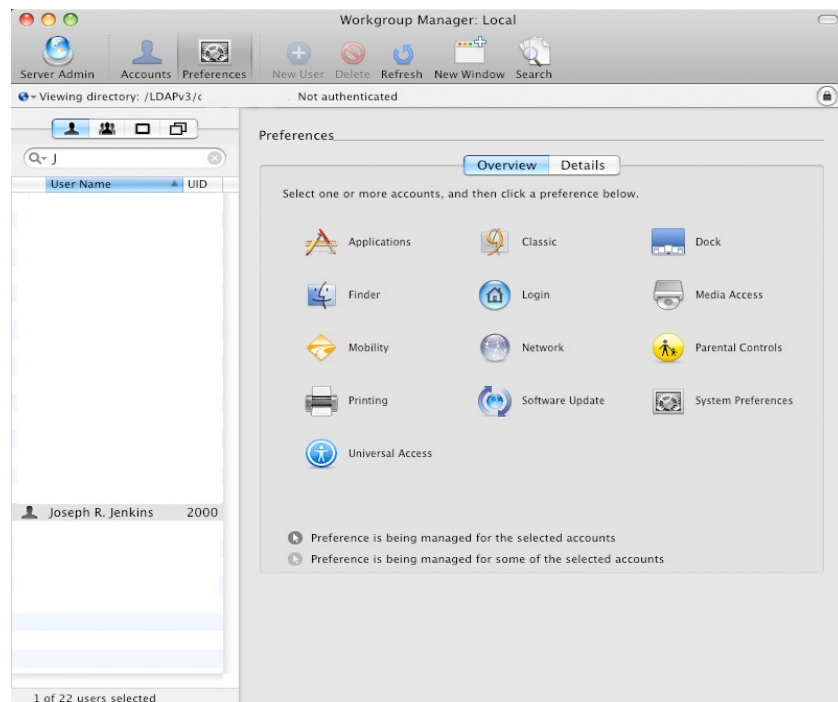
You can browse the directory and also read in user objects. This is a good way to ensure you're getting the data back that you expect from the directory, especially values like the username, uid, gid, apple-home-directory, apple-home-url, etc.

If everything looks good, you should be able to log out, and then log in with the network account from the same Mac. If login isn't working or only working properly, troubleshoot the following:

- Make sure the user exists in a universal password policy. Set up a universal password for the user manually in iManager.
- If the home directory will not mount on login, double check your AFPVOL.CFG, make sure afptcp.nlm is loaded and enable logging (see the NFAP documentation for info on this) Check the mount object naming and also check the user object's apple-home-url, this is critical. The apple-user-homeDirectory is also critical. They must be entered correctly and match the configuration of the mount object in Edirectory,
- Use DSTRACE to monitor LDAP and NMAP during a login. Check the AFPTCP.LOG for errors. Also, log in locally to the Mac client and use the Console utility to check the system logs.

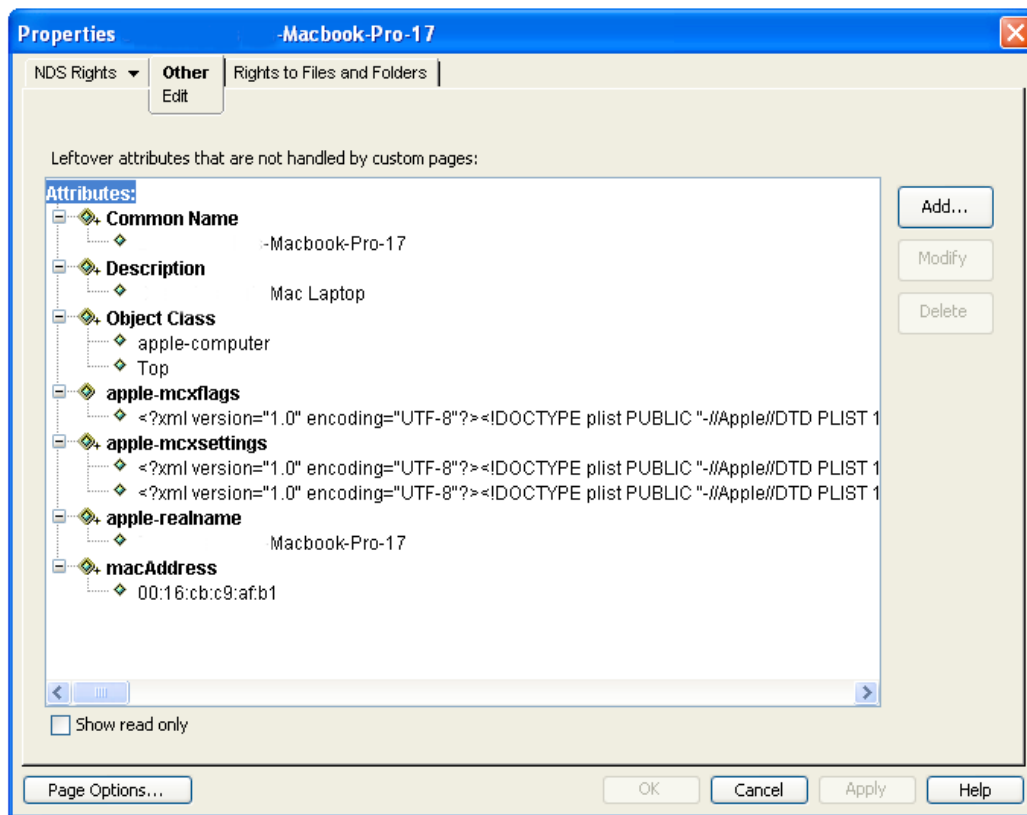
MCXSETTINGS and MCXFLAGS and Management

Many of the objects you extend with the Apple schema in Edirectory can have the apple-mcxflags and apple-mcxsettings attributes added to them under the "Other" tab in their object properties in ConsoleONE. This is particularly useful with Users, Groups and Apple-Computer objects. This allows you to connect to your Edirectory with Apple's Workgroup Manager and manage per-user, per group and per-machine preferences, security, mounts, etc and it's very powerful. Apple stores these configuration values in plists and they are in an xml format. Since it would be very difficult to manage these with ConsoleONE, it's easier to connect to Edirectory via LDAP with Workgroup Manager, authenticate as an administrator, and manage these details. For information on what you can manage with Workgroup Manager, please see the OS X Server documentation on Apple's website.



In the above screenshot, you can see Workgroup Manager connected to my Edirectory and I'm displaying my own record. Assuming I've added apple-mcxflags and apple-mcxsettings as attributes to my user object in Edirectory, I can now go into most of these preference panes and set them up for my individual user. This is also possible with group objects you've extended with apple-group and then added the appropriate mcx attributes to. Also, you can create computer records in Edirectory that correspond to the workstations on your network and manage system preferences for the machine itself, no matter who logs in. This is very useful for adding login network share mappings, scripts, system security settings for user media, etc. **NOTE:** I do not recommend using WM to manage user attributes you typically manage with ConsoleONE or iManager, I am not sure if this could create problems in your directories. For now, we use it solely for managing apple-specific preference settings which are best managed in Workgroup Manager.

To do this, in ConsoleONE, go to your preferred OU and right click the OU, Create Object and select apple-computer. Give it a name that matches the machine name of the computer you want to manage. You'll then want to edit it's properties and add some additional attributes in the Other tab:



I've added Description, apple-mcxflags, apple-mcxsettings (both mcx attributes are left empty, this record was later edited with WM), apple-realname (same as the object's name), and macAddress. Once the new workstation's object has been created, it can then be managed with Workgroup Manager.

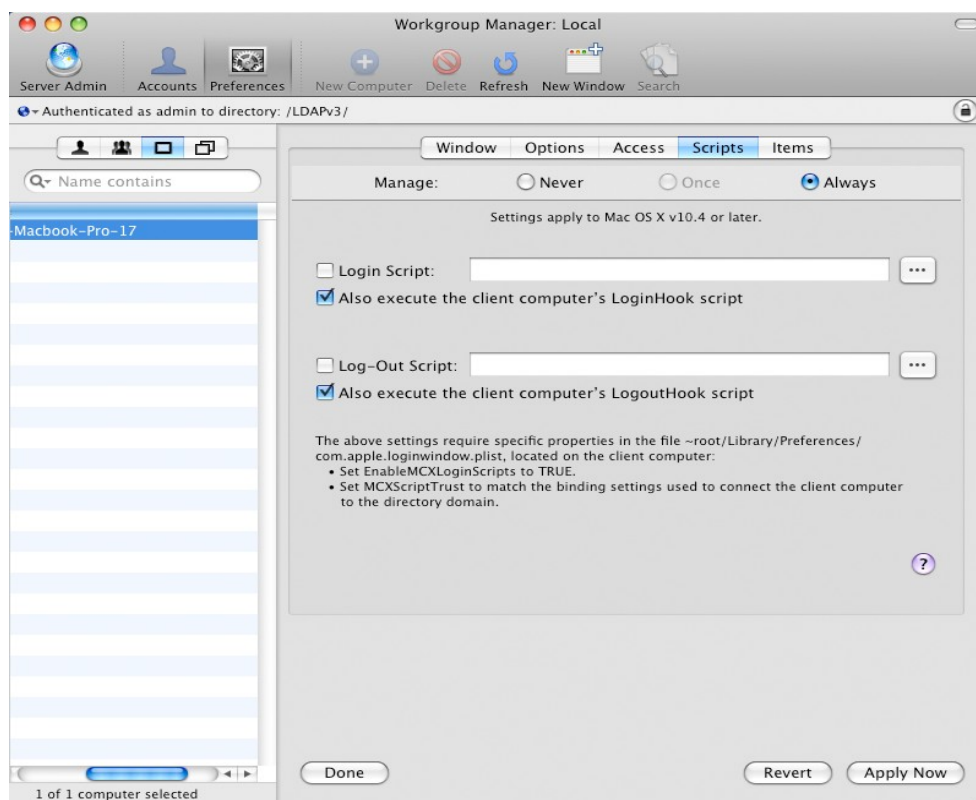
Automount / Login Scripts

There are a number of ways to achieve automount in OS X. Unfortunately, they are not very obvious. We are experimenting with implementing these in a number of different ways. On Windows, and even in Linux, with the Novell Client, when you login, the Novell Client can process a login script that can map drives for the user and perform other actions at login. With OS X + Edirectory, we can mimick this behavior in several ways:

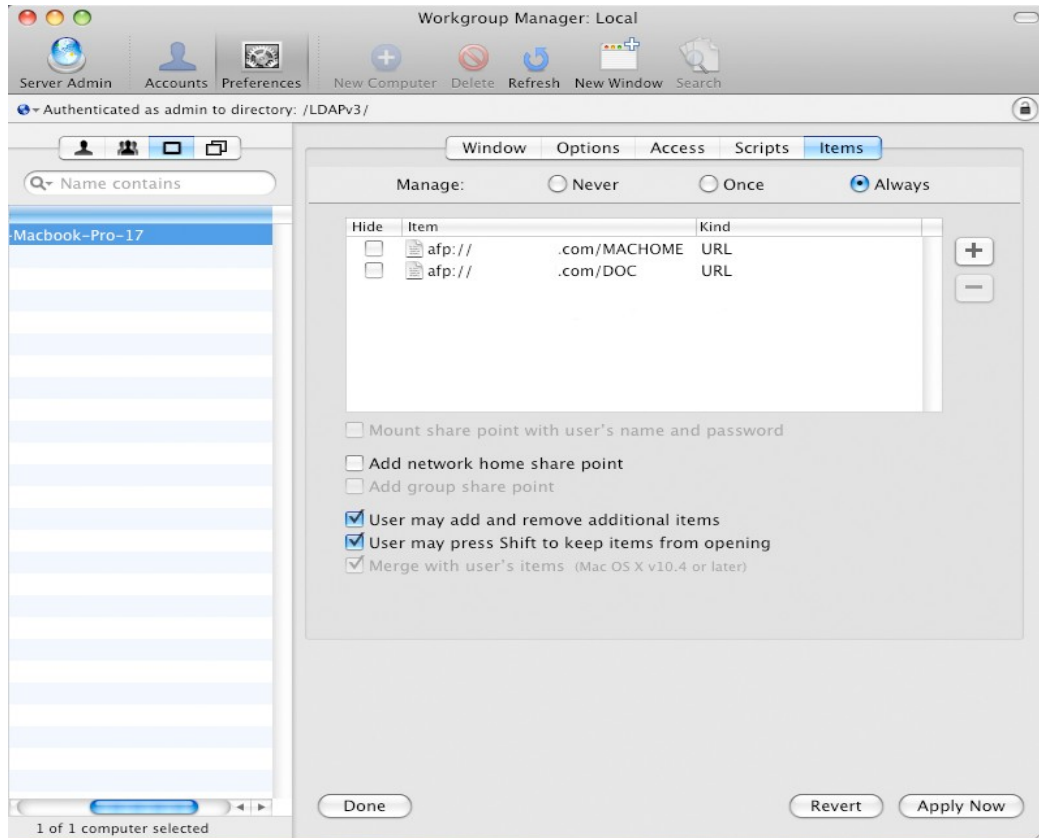
- Per-user login items – it's possible for individual users to login, go into system preferences and add AFP volumes to their login items with simple drag and drop, this places an afp:// url in their login items, and the next time they login, it will attempt to mount the volume automatically. It will ask for a login and password every time unless the user adds their Netware login and password to the keychain upon mounting the volume. Once this has been done, the next time they log in the volumes will mount automatically and are immediately accessible.
- Per-user / per-group / per-computer login items managed with Workgroup Manager – if you have set up the MCXFLAGS and MCXSETTINGS attributes on Apple-enabled users, groups or

workstations in Edirectory, it becomes very easy to set up automounted AFP volumes in the Preference management portion of Workgroup Manager. When the user logs in, it will initially ask them for their Netware login and password to mount the volume, and again, if the user adds this login to their OS X keychain, future logins will not require it (unless the user changes their password)

- Applescript or other language-based login scripts – you can also create specific scripts that are executed upon login via login items or loginhook in OS X (there is also a logout hook) With Applescript, it is relatively simple to create a script that has a few lines like:
 - mount volume “afp://SERVER/VOLUME” - usernames and passwords are not required here. The system will prompt the user for them, and if they use their keychain, it makes it really user-friendly for future logins.
- Applescripts can be set as login items and can be executed at login. As with other preferences, they can also be controlled in Edirectory with Workgroup Manager and MCXSETTINGS and MCXFLAGS on a per-user, per-group or per-workstation basis. Also, the Workgroup Manager also gives you the ability to explicitly set scripts that are bound as login or logout hooks. The screenshot below shows Workgroup Manager accessing the preferences for a single Apple workstation I've created in Edirectory. **NOTE:** Some preferences for “Login” can only be managed on a per-workstation basis, particularly login scripts and loginhook and logouthook. Because of this, we use our Apple-enabled groups to manage login volume automounting. It is possible to create an apple-computer-list which can manage preferences for an entire list of workstations in Edirectory, we have not yet experimented with this.

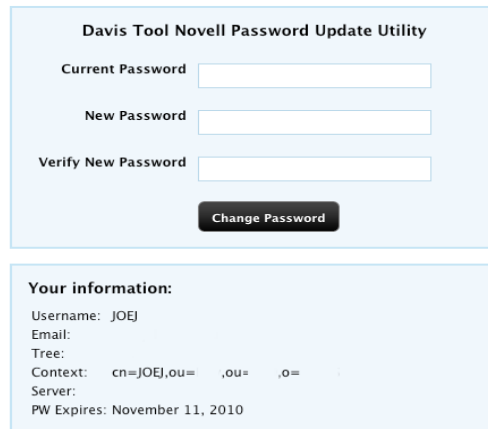


This screenshot shows the same workstation in Workgroup Manager with login items added. Here, a couple of AFP urls have been set up to allow automounting of these volumes for any user that uses this Mac workstation. We find it's better to manage these types of things on a group basis, as our users are in distinct groups and not all of them need the same volume mappings.



Applescript Login Scripts

I am now in the process of developing a login script system that somewhat mimics the login scripts stored in Edirectory and used by other Novell clients. One of the problems we encountered is that OS X will obviously ignore the passwordExpirationTime that is stored per user in Novell's Edirectory. We enforce periodic password changes, and so we have to have a method for checking this on login. I found an Applescript that checks this via LDAP and redirects the user to a predefined website. I am in the process of rewriting this to work properly in our environment. Currently, it checks the passwordExpirationTime via LDAP and redirects them to a password change utility I wrote in PHP that is hosted on our Netware server:



Davis Tool Novell Password Update Utility

Current Password

New Password

Verify New Password

Your information:

Username: JOEJ
Email:
Tree:
Context: cn=JOEJ,ou= ,ou= ,o= .
Server:
PW Expires: November 11, 2010

When I get this code working properly, I will post it with this document for others to try out. In the future, we hope to be able to read the login script directly out of Edirectory and process it on the individual workstations at login. This may turn out to be overkill, however, as we can achieve most of what we require without it currently.

Example LDAP Password Expiration Applescript

This is a script that can be used as a login item that will check a user's passwordExpirationTime and redirect the user to an internal website to change it if need be. I used some of Randy Saek's code from CoolSolutions (http://www.novell.com/coolsolutions/tools/downloads/password_expire.applescript) and modified it to suit our needs:

```
set shellScriptReturn to do shell script "ldapsearch -x -h YOUR_LDAP -LLL cn=`whoami` passwordExpirationTime" as text
set n to (text 1 thru 14 of word -1 of shellScriptReturn) as text
set expiryDate to date "Monday, December 1, 1000 12:00:00 PM"
set pwExpireYear to (characters 1 thru 4 of n)
set expiryDate's year to pwExpireYear as string
set pwExpireMonth to (characters 5 thru 6 of n)
set expiryDate's month to pwExpireMonth as string
set pwExpireDay to (characters 7 thru 8 of n)
set expiryDate's day to pwExpireDay as string

set expiryDays to ((expiryDate - (current date)) div days)
log expiryDays
if expiryDays < 7 and expiryDays > 0 then
    set theReply to display dialog "Your password will expire in " & expiryDays & " day(s). Do you want to change it now?" buttons {"Yes", "No"}
        default button 1 with icon caution
    set theUser to do shell script "whoami" as text
    if button returned of theReply is "Yes" then
        tell application "Safari"
            activate
            make new document at the beginning of documents
```

```

        set the URL of the front document to "http://NWSERV/phpMyNovellPassword/index.php?userName=" & theUser
        set the bounds of window 1 to {0, 22, 900, 644}
        close (every window whose name is "Untitled 1")
    end tell
end if
else if expiryDays < 1 then
    set theReply to display dialog "Your password will expire today.
You are required to change it now." buttons {"OK"} default button 1 with icon stop
    set theUser to do shell script "whoami" as text
    if button returned of theReply is "OK" then
        tell application "Safari"
            activate
            make new document at the beginning of documents
            set the URL of the front document to "http://NWSERV/phpMyNovellPassword/index.php?userName=" & theUser
            set the bounds of window 1 to {0, 22, 900, 644}
            close (every window whose name is "Untitled 1")
        end tell
    end if
end if
end if

```

I've removed some code concerning grace logins out for the time being but will add it back later. It's in Randy Saek's example in the link above.

This applescript is executed as a login item and if the password is about to expire, it directs them to a web-based password change utility I created in PHP. Since Netware has Apache2 and PHP support, the utility lives right on our Netware file server under SYS:/etc/apache2/htdocs/phpMyNovellPassword.

Attached in the ZIP file that contains this document is the PHP code and javascript for our utility.

Acknowledgements:

This document was created because I felt like there wasn't a very clear step-by-step method of putting this all together. I've found a lot of information on OS X and Edirectory online but there were some pieces missing. I hope this helps clear up some of the confusion and makes it easier for others trying this out for the first time.

Resources:

<http://www.afp548.com/>

<http://www.novell.com/coolsolutions/feature/11740.html>

http://andromeda.rutgers.edu/~sysmail/mac/Integrating_MacOS_X_and_edir.pdf

<http://www.novell.com/documentation/>

<http://www.apple.com/server/macosx/resources/documentation.html>

Thanks to everyone who has taken the time to assist with my questions in various Novell and Apple forums. This is an evolving document. If there are mistakes, additions, or changes that you would like to see included, please email joe@nerdnet.com

Written By:

Joe Jenkins

Network Engineer

Davis Tool, Inc.

Hillsboro, Oregon

Changelog:

Rev 01 – Initial Release

Rev 02 – Added automount / login script sections

Rev 03 – Added LDAP passwordExpirationTime Script info and PHP based pw change code