

AD/OD Integration

Active Directory users integrated with managed client preferences from an OS X Server Open Directory System

Joel Rennich
mactroll@afp548.com

v. 2.1

Overview....4

A Note about OS X 10.4....4

Active Directory....4

Open Directory....5

Directory Host....6

OS X Client....6

Methodology....7

Configure the DirHost system....7

Configuring DirHost under 10.4....7

Configuring DirHost under 10.3....7

Create Home Sharepoint....9

Configure the AD Users....9

Configure Open Directory....10

Configuring OD under 10.4....10

Configuring OD under 10.3....10

Kerberos Hamstring....11

Managing Preferences....12

Configure OS X Client....12

Additional Configuration....14

DirHost Backup....14

OD Replica....15

Home Folder Creation....16

Kerberizing the Other Services on DirHost.....16
Synchronizing AD Groups with OD Groups - 10.4.....16
Synchronizing AD Groups with OD Groups - 10.3.....16
Troubleshooting.....18
DNS.....18
Strange Password Errors when joining.....18
Debugging.....18
Other Notes.....20
Windows clients.....20
File Quotas on the DirHost.....20
.local namespace - 10.4.....20
.local namespace - 10.3.....21
A word about AD computer accounts.....21
Automatically joining clients to the AD domain.....22
Providing mail for AD users.....22
Revisions.....24
Version 1.4.....24
Version 2.0.....24
Version 2.1.....24
User Management Script to load in user shortnames and create home folders.....26

Overview

A fault-tolerant directory services system to allow easy integration of OS X clients into a predominately Active Directory environment. The system comprises these major components.

AD – Active Directory is the primary authentication service for the network. All users will be created and managed from within AD using the normal AD tools for such purposes.

OD – An Open Directory database hosted on OS X Server will provide client management at the machine and group level. An Open Directory replica can optionally be set up on the network to provide management information redundancy.

DirHost – An OS X Server connected to a large volume, preferably a SAN system, to house home folders for the OS X client machines.

OSX – For the scope of this project, OS X will be the only client type that concerns us. All of this hinges on using 10.3.3 or later for the client OS.

In addition, when referring to the action of "making a computer part of the AD domain" this document may use both the term "join" and "bind". The term "join" should be familiar to Windows administrators, but the term "bind" is what the action is typically referred to as in OS X and is the same term used when describing making a computer part of a standard LDAP domain.

A Note about OS X 10.4

Tiger brings a number of enhancements to this setup. It streamlines a lot of the tedious and confusing hand editing of config files that you used to have to engage in. However, much of the basic premise of this configuration is identical between 10.3 and 10.4.

Version 2.0 of this document, which you are reading, adds information about Tiger. To hopefully keep things rather simple bits of this whitepaper that pertain only to Tiger (10.4) are shown in red. Information only pertaining to Panther (10.3) is shown in blue.

Active Directory

The system will integrate with AD as if the OS X machines were native Windows clients. The only change necessary to AD is to specify a home folder location for the users. This location will reference a share point and home folder stored on the DirHost system.

When a user logs onto an OS X machine, that user's entire home folder will be stored on DirHost. This is the primary location for all user documents and preferences which means that they will all be stored on the server in an easily accessible place.

When a user logs onto a Windows machine the user's Z drive will be mapped to their home folder stored on the DirHost system. If you would like to have the users "My Documents" folder and the desktop to be stored in this network folder you'll need to implement a GPO in AD to redirect those folders, or do this on a user by user basis.

All password polices specified in AD should be honored by the OS X client machines. For example, if you specify that the user needs to change his or her password at the next logon, the user will be prompted to do so when they log on to the OS X machine.

The primary means of authentication to the AD server will be over Kerberos. Since the OS X client acts like a native Windows AD client, the user will receive a TGT from the AD KDC which can be used for single-sign on to all Windows file servers in the AD domain.

Open Directory

This is a fairly straightforward Open Directory (OD) setup. One server is installed as an Open Directory master. Under 10.3 this server was best if it was not joined to the Active Directory domain. However, starting with OS X Server 10.4 you should have no issues joining the server to AD after making it an OD master, as long as you keep a few things in mind. Mac OS X clients will be joined to both OD and AD (discussed later in the OS X Client section).

Workgroup Manager (WGM) can be used on a client system to add AD users into groups hosted in the OD database. Since your OD Master won't be bound to AD, and thus can't see AD users itself, you will need to designate a client workstation as the "admin" workstation and install the Server tools on it so that you will have WGM on it. Because this "admin client" will be bound to AD and OD, you will be able to see information in your AD and OD at the same time. You can then run WGM on the admin station, connect to your OD Master, and add users from AD to groups you have set up on your OD master.

Once the AD users are in an OD groups, client management policies can be applied to them. This ensures that client management policies can be enforced without having to make any non-standard changes to the AD system.

Additionally the OS X client machines are added to machine lists on the OD system. This allows for machine-level management policies to be enforced. Also you can specify what groups will be able to log on to what lists of machines.

Ideally you would want to setup a replica OD system to replicate the OD database from the master. However, this is not essential and can be omitted if your organization does not have the resources available to spare a secondary server for this task.

If you are only using the OD system to supply management information and have no intention of running file services from this machine, which is the preferred situation, you

can use the 10-user version of OS X Server on both the OD Master and Replica systems. Also, as these systems will be used primarily for LDAP connections the processor load should be quite light, so you don't need to put your most advanced hardware into this system, save that for hosting your home folders.

Directory Host

The DirHost system will share all user network home folders from the Xserve RAID, or other large volume. The home folders will be shared over AFP but will also be available over SMB for Windows client access. This is especially important if you would like to use this Xserve shared home as a Windows Network Home folder.

The DirHost SMB server will be configured to use the AD domain for SMB service authentication. This means that Windows clients will get single-sign-on support for SMB services. OS X clients, who have authenticated through AD, will also get single-sign-on support for all services hosted on the DirHost server.

You should dedicate your most capable equipment to this task since it will have the highest processor utilization.

It is possible to create a failover solution with this system and another Xserve to protect yourself in case of server failure. Successfully implementing the solution takes some time and effort, plus tends to complicate the entire integration issue with AD, so make sure you have the AD/OD integration working before attempting to provide for AFP failover.

OS X Client

The OS X client machine will be configured to get authentication from both the AD domain and the OD domain. The AD plug-in will be used to configure the AD domain and the LDAP plug-in will be used to connect the client to the OD system.

This configuration allows the OS X client to authenticate users from either AD or OD and pull client management information for either type of user from the OD domain.

Typically every Mac OS X computer on your network is joined to the AD domain using a unique computer ID. This will allow you to easily identify machine accounts using the AD management tools and trouble-shoot and dictate domain access to individual machine accounts. This is the recommended setup, in almost all cases. (See the section entitled "A word about AD computer accounts" for a discussion of why this is the case).

Methodology

Configure the DirHost system.

On the first Xserve in the DirHost group, configure the AD plug-in in Directory Access to allow the server to join the AD domain. You can do this by launching the Directory Access application directly from the Utilities folder, or by changing your server to be in "Connected to a Directory System" mode via the Open Directory settings in Server Admin (which then gives you a button to launch Directory Access directly from within Server Admin). It is advisable to use a unique computer account to bind all servers to the AD domain. After the binding use WGM or dscl from the command line to ensure that the binding is working and that AD users can be seen from the server.

The process of binding will create a Kerberos config file at /Library/Preferences/edu.mit.Kerberos. You can test the validity of this file by using kinit and the domain name of an AD user. If the command takes your password without response then the Kerberos config file is valid. For example:

```
kinit joel@addomain.com
```

Now the configuration of 10.4 and 10.3 diverge.

Configuring DirHost under 10.4

After joining the server into AD, you can easily enable Kerberos for all services that support it, including SMB, by using the "Join Kerberos" button that's exposed in the Open Directory Module of Server Admin after you have successfully bound to AD.

This can also be done by using the dsconfigad command.

```
dsconfigad -enableSSO
```

At this point AFP, SMB, HTTP and any other service that can be Kerberized in 10.4 will have been Kerberized. That's really it.

Configuring DirHost under 10.3

A brief note about AD groups being seen by the OS X Server: It takes a while. Members of AD groups are listed by their full LDAP name, which is in contrast to OS X which lists group members by their short name. As such the OS X Server has to iterate through every AD group member and convert the LDAP name to a more-Unix like short name.

This list takes a while to generate the first time out. On a large AD system this can be as much as 24 hours or more. It's all done behind the scenes so just leave the server on and give it some time.

Until 10.3.4 OS X client systems followed the same behavior. Now, a client machine will dynamically create a group list of what groups the currently logged in user belongs to and not the entire AD domain.

Next begin the integration of the SMB server on the DirHost with the AD domain. In Server Admin supply a server name for your OS X Server. I've had best luck if this name is an all caps. Also enter in the NT Workgroup name of the AD domain in the Workgroup field. Leave the server role in the Windows settings panel at Standalone, even though there is a Domain Member setting. The Domain Member setting is not functional for our purposes here, as it refers to being an NT domain member not an AD domain member (and is different than the Open Directory role).

You'll want to make sure that there is an entry in the DNS server being used by the AD domain for the DirHost server. There should be both forward and reverse records that map the server name to its IP address. In this example we are using DIRHOST as the NetBIOS name of the OS X Server, so there should be a forward and reverse entry for DIRHOST.addomain.com.

Before turning on the SMB server edit the server's /etc/smb.conf file. We'll need to hand edit this to fully integrate the OS X Server into the AD domain. Begin by changing the security directive from "user" to "ads". Then add a realm directive with the AD domain's Kerberos realm name, which should be the same as the AD domain but in all caps. For example:

```
realm = ADDOMAIN.COM
```

Finish up by changing the "spnego" directive from "no" to "yes".

Starting with 10.3.6 I've also found it helpful to add

```
winbind separator = +
```

into this file.

Once you have made these changes you will be able to manage the smb server from Server Admin as long as you do not change the Windows role of the server, i.e. leave it as a stand-alone smb system. Changing the Windows role will overwrite your changes to /etc/smb.conf. When you are finished with your edits it would be a good idea to back this file up. Resist the urge to lock it, since that will prevent you from adding SMB sharepoints through Workgroup Manager. Just don't change the Windows role and things will be fine.

Note: As mentioned above, the settings in the Windows services in Server Admin will not necessarily reflect the fact that your server has joined a domain, and will still state that your server is a stand-alone SMB system. Ignore Server Admin in this case. This setting

only refers to an NT style domain member, and applies only to SMB shares, whereas binding via the AD plugin (as you did above) provides AD style domain joining and access to domain information for all services on the server.

Create Home Sharepoint

Finally you'll need to create a new sharepoint to be used to host the AD users home directories. In this example we'll use "OSXHome" as the sharepoint name. You can set the group ownership and permissions on this share to whatever you would like. However it is best to keep the privileges for everyone to at least read-only on this folder.

Before moving on you should pick a test user out of the AD domain that you will be using to test your configuration and create a home folder for them on this server. You can do this by copying over the default user template which is located at /System/Library/UserTemplate/English.lproj, to the new OSXHOME sharepoint, which in our example resides at /Volumes/XRAID/OSXHome.

```
sudo cp –r /System/Library/User\ Template/English.lproj
     /Volumes/XRAID/OSXHome/joel

sudo chown –R joel /Volumes/XRAID/OSXHome/joel
```

Configure the AD Users

First test to make sure that the DirHost server has been successfully integrated into the AD domain.

Log into a Windows client as an AD user. From the "run" menu fileshare to the OSXHOME sharepoint by using the server's name being careful to include the sharepoint in the UNC. For Example:

```
\\DIRHOST\OSXHome
```

You should not be prompted for a username or password. If you are you'll need to double-check both the Kerberos config file and the /etc/smb.conf file on the DirHost server.

Next you'll need to assign a home folder to a user that you would like to test out the authentication system with. Using the AD Users and Groups application select a domain user, get their properties and supply the UNC for a home folder that resides on the DirHost server. You'll need to specify a drive to map this to for Windows clients. Pick anyone that your organization has free. Here's a sample UNC for the home folder location.

```
\\DIRHOST\OSXHome\joel
```

When applying this change you will get a notification that the home folder could not be created because it already exists. Furthermore you will be asked if you would like to change the permissions so that the selected user will own the folder. If you don't get notified that the folder already exists, you need to check the UNC that you supplied for the home location. When asked if you want to change the permissions, select "no" just for fun. I have found no difference in permissions regardless of how you answer this question.

You're now done with all of the configuration that you'll need to do with the domain.

Configure Open Directory

You should now focus on the OS X Server that you'll be using to supply client management information to the OS X clients. This is going to make up the OD domain (and should be a separate system from the DirHost system if you're using 10.3). Since it will be primarily used only for LDAP lookups, you can use less powerful hardware to fill this role.

First promote this server's role to be an Open Directory Master server using Server Admin. Then use Workgroup Manager to ensure that the promotion took. You should see an /LDAPv3/127.0.0.1/ entry in your selection of databases which can be reached by using the small blue globe at the upper right of the Server Admin window.

Configuring OD under 10.4

Tiger Server changes some of the basic methodology in this situation. Under 10.3 I encouraged all listeners to keep the OD server away from AD. Under Tiger, though, you can easily have your OD and your AD too. This means that you should be able to host both the LDAP MCX settings for your OS X clients in addition to being a home directory server for the SMB homes and eliminate the need for these services to be hosted on different servers. However, if you are expecting to have more than 30-50 users, I'd suggest a server for MCX and a server for the homes.

Once you have established yourself as an OD master, go to Directory Access and join AD. Do not enable Kerberos yet. Go to the Authentication tab in Directory Access and drag the LDAP entry below the AD entry. With AD above the server's own OD entry your server will only use the AD Kerberos realm, and not your OD realm.

Configuring OD under 10.3

As mentioned above, you should not join a 10.3 OD server to the AD domain. This just keeps the directory service configuration on the OD Master simpler. Plus it makes it easier if you would like to set up cross-realm authentication between the OD realm and the AD realm.

Not joining the AD domain means you will not have direct access to your AD users from your OD Master. This is really of no consequence since you can use an OSX client and scripts provided at the end of this documentation to help populate your OD groups with AD users.

Now that your OD master is up and running, you can create a new group in the LDAP domain on the OD server and add your AD users to it. You can then add users from AD to your OD groups by hand using the dscl command. To add the AD user "joel" to the OD group "adusers" you'd use this command on the OD server.

```
dscl -u <adminuser> -p <adminpass> /LDAPv3/127.0.0.1
merge /Groups/adusers GroupMembership joel
```

Go back into Workgroup Manager and ensure that the user has been added to the group in question.

If you are uncomfortable with the command line, wait until you have bound a client to both AD and OD, then you will be able to use Workgroup Manager on that system to add AD users to OD groups.

Now, regardless of how you added the user, you'll need to enable managed preferences for this user. Make enough changes to the default preferences that you'll know that management is taking place. (Obvious changes that are good for testing are things such as Dock size and position, as well as mandating "Simple Finder").

You can create non-AD users and groups in your OD domain if you would like. These users will be able to login to your OS X client machines and authenticate from the OD domain. No additional client configuration, other than what we are already doing, is necessary for this.

Once you have configured an OS X client to join both domains, as I discuss in the next section you'll be able to use WGM locally on a client machine to add AD users to your OD groups. The procedure for this is quite simple. When WGM comes up do not login to a server. Instead use the "View Directories" menu item to display a WGM window. Do this a second time to get two windows open. At which point you can view your AD domain with one and your OD domain with the other. Authenticate to the OD domain by clicking the lock icon at the upper right of the window and then drag users from the AD domain into OD groups.

Kerberos Hamstring

Important Note: When using either 10.3 or 10.4 in addition to adding AD users to your OD groups so you can assign them managed preferences, you will need to follow the instructions in the Apple tech-info article that explains how to disable Kerberos auto-configuration on the Open Directory Master. This is necessary because your client Macs

will be bound to both Active Directory and Open Directory. Both of these directory systems use Kerberos for authentication. Both will try to auto-configure the OSX clients to use their Kerberos authentication realm at the same time. Because you are authenticating users against Active Directory, you need to disable the record in Open Directory that tells OSX clients to use it for Kerberos. If you don't, the OS X clients will get confused as to which directory system they should be using for their authentication source and your users will get locked out.

While this is supposed to work better with 10.4, I'd make sure you didn't run the risk of conflict and go ahead and remove the OD Kerberos config.

See: http://docs.info.apple.com/article.html?artnum=300765

On 10.3 it's possible to leave the Kerberos realm field blank in the Open Directory module of Server Admin when creating the server, which will prevent Kerberos from being initialized. However, this does not work on 10.4, as the GUI requires a value to be entered into the field.

Managing Preferences

You can now use Workgroup Manager to log into the LDAP server hosted on the OD system and create groups in OD. Then add users from AD, using the user selection tools in WGM, to the OD group. You can then manage this group as if it was any other OD group.

For computer lists, you'll need to actually enter each machine's MAC address into the list. There is no way to just drag the computer account from AD into an OD computer list.

Configure OS X Client

Now on to the client configuration. On a fresh install of OS X Client configure the AD plug-in in Directory Access to join the AD domain. Make sure that you add the AD domain to the authentication tab in Directory Access.

For more in-depth information on the AD binding process read Michael Bartosh's articles on it at the O'Reilly network site, http://www.oreillynet.com, or read my article on troubleshooting the plug-in, http://www.afp548.com/article.php?story=20040722074608881.

You should now be able to logout and login into the client machine as an AD user. You should find a folder that has been added to your dock, which references the AD user's network home folder host on DirHost.

By default you'll also have a new user folder created for this AD user. This is a local home folder that has no relation to the network home folder that has been created for this user.

Since you have authenticated to the AD domain, you have received a TGT for the AD domain controller, which will allow you single-sign-on access to all Windows servers in the domain, including the DirHost server. Note that this only works for SMB connections.

Next log back into the client machine as an admin user. We'll now configure the client to mount the network home folder over afp instead of using a local home folder. Do this by using the dsconfigad command.

```
sudo dsconfigad –localhome disable
sudo dsconfigad –mountstyle afp
```

For more info consult the man page for dsconfigad.

At this point you can reboot just for fun, although the changes should take effect immediately. Log back in as the AD user. If everything is correct the user will be put into a network home folder host on the OS X Server DirHost and being mounted over afp to the client machine. If you navigate to /Network/Servers/DIRHOST/ you should see an entry for OSXHome

If the whole process is working you can now join the OS X client to the OD domain. So log back into the client as a local admin user. Go to Directory Access and configure a new LDAP entry that references the OD domain. If you have more than one OD server in the OD domain, for example a master and replicas, you only need to point the client to any one of these servers.

Now add the OD domain to the authentication path on the client machine. Make sure it appears second in the list, under the AD domain.

Finally log out of the machine and log back in as the AD user that belongs to a workgroup hosted on the OD domain. You should get a network home folder and a managed environment.

Additional Configuration

DirHost Backup

If you intend on providing an AFP fail over solution for your DirHost server a few changes need to made to set up the IP failover between the two servers hosting the DirHost RAID. Again, keep in mind that there will soon be much better, and Apple-supported mind you, methods of doing this. However, if you must know the following information should be enough to make you dangerous.

To implement this you will need two servers and some sort of shared storage system. Both servers need to be able to see the storage at the same time, so you must have some sort of SAN solution for this to work.

The basic premise of this method has the secondary server not mount the shared storage. In fact the secondary server does absolutely nothing until a failover situation occurs. At which point the secondary will mount the storage volume and assume the IP of the primary server. Users WILL notice an interruption in services, but are normally able to log back into their home folders within 30 seconds of the primary server going down.

On the primary server, DirHost1, create a new firewire interface. Choose any IP scheme that you want as long as it doesn't interfere with your existing configuration. In our example we'll set it to 192.168.199.1 with a subnet mask of 255.255.255.0. The Ethernet interface is set to 10.99.99.10 with a subnet mask of 255.255.0.0.

On the backup server, DirHost2, also create a firewire interface in the same subnet as DirHost1. For the example, we'll set this to 192.168.199.2. Additionally the Ethernet interface is set to 10.99.99.11.

Edit the /etc/hostconfig file of DirHost1 to set up the failover heartbeat process on it by adding this directive.

```
FAILOVER_BCAST_IPS="10.99.255.255 192.168.199.255"
```

On DirHost2 you'll also edit /etc/hostconfig to receive the heartbeat from DirHost1.

```
FAILOVER_PEER_IP="192.168.199.1"
FAILOVER_PEER_IP_PAIRS="en0:10.99.99.10"
FAILOVER_EMAIL_RECIPIENT="admin@addomain.com"
```

You can now configure the scripts on DirHost2 that will augment the failover process. We'll be referencing the RAID system by using /dev/disk2 in this example. You'll need to check on your system to see what disk the RAID mounts as.

Create a directory on DirHost2 for the scripts at /Library/IPFailover/10.99.99.10. Then create a PostAcq script in that folder.

```
#!/bin/bash

serveradmin start afp
serveradmin start smb
diskutil mount /dev/disk2
```

Also create a script to shut down the fileservers and unmount the RAID. This script goes in the same folder and is called PreRel.

```
#!/bin/bash

serveradmin stop afp
serveradmin stop smb
diskutil eject /dev/disk2
```

Finally finish up the configuration on DirHost2 by editing /etc/fstab so that the RAID is not mounted on the second server. Add this line to the file:

```
LABEL="XRAID"       none hfs  ro,noauto
```

This will cause DirHost2 to ignore the XRAID volume when automounting at bootup.

Now reboot both systems. And you should have a failover network.

We have a much more detailed article on this at http://www.afp548.com

OD Replica

You are more than welcome to create an OD replica to better protect your OD domain from failures. You are able to do this at anytime during the configuration after the OD master has been setup. Use Server Admin to configure the secondary server as the replica.

You will not need to join the replica to the AD domain. Although under 10.4 you certainly could. All of your changes will be made to the LDAP database on the OD master.

The replication process will ensure that all changes made to the master will be replicated to the replica server, so once you set up the replica you should not have to do any directory service management on it.

Home Folder Creation

While AFP home folders are supposed to be auto-created when a user logs in for a first, it has not necessary been my experience that that functionality works with AD users. As such you will most likely need to create these by hand. I'm including a script at the end of this documentation that will allow you to feed it a list of user's short names and it will generate home folders.

Alternatively you can find another good script to do this at Michael Bartosh's site in the downloads section, http://www.4am-media.com.

Kerberizing the Other Services on DirHost

Editing /etc/smb.conf will allow your AD users single-sign-on access to the SMB server on DirHost, however it does not automatically configure the other services. Again this is another area where OS X Server 10.4 should improve on.

However, it's a rather simple process to do this on a 10.3 system. The documentation currently existing on how to do this is complete enough that there is little point in replicating it here. So check out Aaron Rosenblum's information, http://www.aaronrosenblum.com/SSOStuff/ODAuthAuthzAD.html, or refer again to Mr. Bartosh's site, http://www.4am-media.com/sso.

Synchronizing AD Groups with OD Groups - 10.4

Starting with 10.4.3, which I suggest you upgrade your system to if you haven't already, you can nest AD groups within OD groups. For example, on the OD Master you can create a group in LDAP, ManagedUsers, and then drag, from the group picker or another WGM window, and AD group, or groups, into the membership pane of the OD group.

Now you will be able to establish managed preferences on the OD group which will apply to all of the AD users in the nested AD group. You can also nest OD groups inside OD groups. Please note that while group memberships are resolved, managed preferences are not combined. For example if you were to nest a previously managed OD group, OtherManagedUsers, into the ManagedUsers group and logged into a client system as a member of the OtherManagedUsers you would see two entries in the managed group pick list in the login window. If you choose to be a managed by the ManagedUsers group, none of the managed settings from

Synchronizing AD Groups with OD Groups - 10.3

Another common hang-up with this solution is that there is no synchronization between the membership of an AD group with the membership of an OD group.

Nicole Jacque has written a wonderful utility to do this for you. You can find reference to it here, http://www.afp548.com/article.php?story=20050809123828646

Troubleshooting

This is not a simple process to get all of this working together. Although for many people this process is pretty painless if taken slow, there are a number of places for issues to occur.

DNS

This is probably where most issues occur. You MUST be using the AD DNS system or else you won't have any chance at joining the AD domain with the AD plugin.

This accounts for easily 90% of the issues encountered when using the AD plugin.

Strange Password Errors when joining

In some, usually larger AD environments, you can run into issues with joining OS X clients into AD due to the fact that OS X may use different servers for the LDAP connection that creates the machine account and the Kerberos connection that sets the machine account's password.

When the AD plugin does it's think, it asks AD which domain controllers it should be dealing with. It will then take two from the list and cache them for use with Kerberos and LDAP connections. The problem is you don't really have any control over which one you use for each connection. So, if the machine account is created by LDAP on one domain controller, and then the plugin uses Kerberos to set the password on another, before the machine account has replicated from one controller to the other you'll get this problem.

The solution is to pre-create the machine accounts and allow them to be fully replicated out before you attempt to join the OS X machines. By doing this the plugin will automatically reuse the existing machine account and since the account is already on all controllers you won't have any problems with Kerberos trying to change the password for something that doesn't exist.

Debugging

When you are running into the really strange things with the AD plugin in general, the first place to start is to put the plugin into debug mode.

```
sudo killall -USR1 DirectoryService
```

This will cause DirectoryService to log to /Library/Logs/DirectoryService/DirectoryService.debug.log. Open this up in Console.app while you are trying to join to AD. There will be a lot of information displayed here, but you may find something very useful. At a minimum this is what Apple or most other consultants are going to want to see, so might as well have it ready if you are going somewhere else for help.

To disable debugging, just rerun that command.

You can also go to the next level and increase the verbosity of the logs.

```
sudo killall -USR2 DirectoryService
```

Which will log at the API level to the debug log. Sometimes this is helpful, but most times it's just to much information. Keep in mind that this level of debugging will turn itself off after five minutes so that it won't fill up your harddrive.

Other Notes

Windows clients

Windows client machines will mount the network home folder specified for the user in AD, just as they would any network home folder. The OSXHome share will be mapped to the drive letter specified in the user record.

If the /etc/smb.conf file on the DirHost server is misconfigured or overwritten by Server Admin, the mapping on the Windows client side will break.

If you would like to redirect the Windows user's Desktop and My Documents folder to the network home, you can either do this manually for each user or use a GPO. Please check the Microsoft Knowledge Base website for more information on this.

File Quotas on the DirHost

It's perfectly feasible to impose file quotas on the AD users that have their homes, or any document storage, on the DirHost system. For many admins this seems to be a bit of a leap of logic as to why this would work. Most of the reason for that is as OS X Server admins we are used to configuring quotas in Workgroup Manager, which we can't do for AD users since we are unable to write those changes back to the AD database.

However, Workgroup Manager, like many things in OS X, is just a GUI to the underlying Unix quota system. The quota system has no concern about where the user comes from. Instead it just assigns a quota value to a user's shortname which the filesystem then keeps track of.

Knowing this, you are perfectly welcome to use the `edquota` command to setup quotas for the shortnames of AD users. Obviously this is a lot of work to do by hand. Luckily you can set up the quota for one user and then feed a list of users to `edquota` and have it make them all like the original.

Finally, you'll need to stop the AFP server on DirHost from stomping on the quotas when users login over AFP. So do this on DirHost.

```
sudo serveradmin settings afp:updateHomeDirQuota = no
```

.local namespace - 10.4

Apple has made the use of .local domains in 10.4 incredibly simple. Just add your AD domain, example.local for example, as the top entry in the search domains field of every OS X client and server that will be using AD. This eliminates the need to do any

manipulation of the local DNS configuration that was necessary for 10.3 as described next.

.local namespace - 10.3

It's not uncommon to find an AD domain that utilizes the .local naming convention for the domain name. While this makes integration a bit trickier it is not hard to overcome.

For a good overview of this problem and an easy solution, please check out an article on my site entitled "Using Alternative .local Domains and Other DNS Tricks", http://www.afp548.com/article.php?story=20041228092123788

A word about AD computer accounts

The AD plug-in in OS X will attempt to create a computer account in the AD domain when joining the OS X client to the AD domain. This account will be visible in the Active Directory Users and Computers tool just as all of your Windows computers are.

If the client was a Windows machine, the computer account would be relevant since it would control various access rights and GPOs that might be assigned to that specific machine. However, since OS X will not respond to GPOs or other AD rights, the computer account is rather irrelevant.

Just to keep things less confusing, and to give you some flexibility, I would strongly suggest that all OS X Servers and clients be given their own computer account in AD. Keep in mind that you will have to find a method to join each machine to the domain, which may be quite a task if you have installed many machines using a network install image. One good way to do this is to use the dsconfigad command in a post install script.

If you would rather just have one account for all clients, this can work. This approach may seem easier on the surface, however it has major pitfalls.

You will need to be *very careful* with joining the domain using that common account. Create a master client image and join this system to the domain. Now image all other machines from that master client image. This will cause all imaged clients to be bound to the AD domain using the same account when you boot them up.

As long as you never attempt to join or unjoin the AD domain using the master client's computer account everything will work since the computer account password will be cached in the master client image.

Let me reiterate the last part. If you ever, ever unjoin an imaged machine from the domain, all OS X client machines will have to be rejoined to the domain since they all

share the same generic computer account. Ideally, after joining the master image change the permissions on the computer account in the AD database to prevent it's modification.

Automatically joining clients to the AD domain

Now that you know you might have to individually join each machine to the AD domain, At first this sounds like a royal pain in the behind. Luckily, for you at least, others have gone before you to do this.

On our site you'll find an article on using a login hook to do this automatically, complete with the code to do it. Tweak it up some if you want and start joining! Read the article here: http://www.afp548.com/article.php?story=20040914111624679.

This is somewhat easier under Tiger, as Apple is now keeping an eye on making dsconfigad play nice in scripts. To that end you can now supply a local admin user name and password in the script instead of having to make sure that it runs as root.

Then once you've used dsconfigad to join the machine to AD you can then use dscl to add the AD domain to your authentication path.

```
dscl /Search -append / CSPSearchPath /Active\ Directory/All\ Domains
dscl /Search -create / SearchPolicy CSPSearchPath
```

Obviously putting any user credentials, especially admin ones, into a script is something that you need to approach with great caution. You can mitigate this risk in 10.4 by making changes to `/etc/authorization` to use a non-admin group for the `system.services.directory.configure` key.

Most smart admins will have the script used to join AD automatically delete itself when it's done to prevent prying eyes from getting any passwords.

Providing mail for AD users

A number of admins have asked about providing mail services, hosted on an OS X Server, for AD users. In 10.3 this was rather hard to accomplish and it was many times easier to create an LDAP connection between the OS X Server and AD then to use the AD plugin.

However, in 10.4 things change a bit. While you're still more than able to add the apple-user-mailattribute attribute to AD you can also use static mapping to map that attribute to a static value, more on this in the man page for `dsconfigad`, or you can use Service Access Control Lists (SACLs).

SACLs are enabled in Server Admin and can apply to a number of different services provided by OS X Server. SACLs are designed to enable or disable groups of users from

using a particular service. In the case of mail, you're able to drag a list of groups that can either be allowed or denied from using the mail services. A side effect of enabling the SACL is the automatic enabling of the mail attribute for all users allowed to use the mail service.

In other words, to enable mail for all of your AD uses, without editing anything within AD, just create a SACL for mail in Server Admin and drag in the groups you require. Keep in mind that this method overrides any and all existing mail attributes already associated with the users. Also, since existing values are overridden there is no way to assign mail quotas or mail forwarding addresses within the GUI.

Revisions

Version 1.4

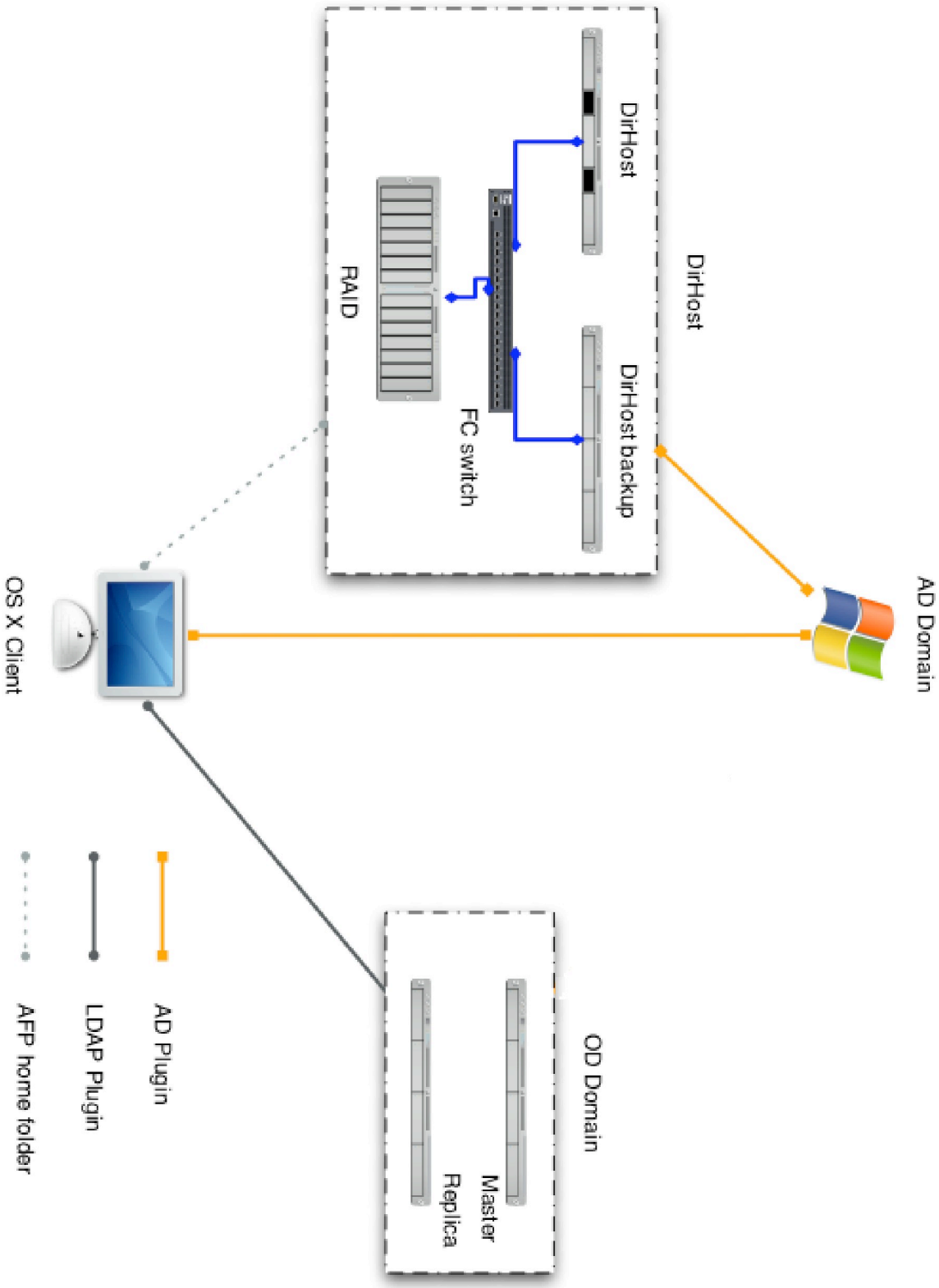
- Updated doc to reflect current practice of not joining OD Master to AD domain.
- Updated doc to clarify warnings about using the same computer account for all OS X clients
- Updated doc to include link to kbase to explain how to disable Kerberos file autoconfig on OD Master.
- Added comment to clarify the terms “joining” and “binding”.
- Updated diagram to reflect OD master not bound to AD.
- Updated URL to Aaron Rosenblum’s SSO doc.
- Added troubleshooting section.
- Added information on setting up quotas for the AD users.
- Added information about pre-creating machine accounts.
- Added information about using loginhook to automatically join to the AD domain.
- Clarified the difference between an OD “role” in Server Admin and a role as configured in the Windows settings in Server admin in the context of AD integration vs. domain integration.

Version 2.0

- Updated information to include OS X 10.4 Server and Client.

Version 2.1

- Updated information to include changes in OS X 10.4.3.
- Added information about setting up mail accounts for AD users.
- Clarified use of a single OS X Server for both OD and home directories.


AD Domain
DirHost
DirHost
DirHost backup
FC switch
RAID
OS X Client
OD Domain
Master
Replica
AD Plugin
LDAP Plugin
AFP home folder

User Management Script to load in user shortnames and create home folders.

```
#!/bin/bash

#
# Script to add users, either singularly or by a list
# to the OD domain
# and create a home folder for them if necessary
# This should be run on your OD master server.
#

#
# Original script - Joel Rennich 4/04
#

#
# An LDAP Admin user and password
# We need this because we have to auth to the LDAP
# domain. This is a HUGE security risk to leave this hanging
# around, so do be careful where you leave this script.
#

LDAP_ADMIN =

#
# Some variables to start things off
# Set these if you want, or override with CLI switches
#

HOME_DIR_LOCATION = /Volumes/HDS/HomeDir/
USER_TEMPLATE_LOCATION = /System/Library/User\ Template/English.lproj

#
# Now for some proceedures
#

#
# check for root privs
#

check_root() {
    if [ `whoami` != "root" ]
    then
        echo "you need to be root to do this"
        exit 0;
    fi
}

#
```

```
# Check for an admin passsword
# and quit if there isn't one
#

check_admin(){
    if [ ! -z "LDAP_ADMIN_PASS ]
    then
          echo "How very smart of you to not put your password in an
option"
        echo "Please type it now: "
        stty_orig=`stty -g`
        stty -echo
        read LDAP_ADMIN_PASS
        stty $stty_orig
    fi
{
# This creates a home folder if one isn't there already

create_home(){
    if [ ! -d ${HOME_DIR_LOCATION} ]
    then
        echo "Doh! The home folder location doesn't exist!"
        echo "You'll need to run this on another server"
        exit 0;
    else
        if [ ! -d ${HOME_DIR_LOCATION}${CURRENT_USER} ]
        then
                  cp  -r  ${USER_TEMPLATE_LOCATION}  ${HOME_DIR_LOCATION}$
{CURRENT_USER}
                       chown  -R  ${CURRENT_USER}  ${HOME_DIR_LOCATION}$
{CURRENT_USER}
        fi
    fi
}

# This adds the user to the specified group

add_to_group(){
     dscl -u $LDAP_ADMIN -p $LDAP_ADMIN_PASS /LDAPv3/127.0.0.1 merge /
Groups/${CURRENT_GROUP} GroupMembership ${CURRENT_USER}
}

# This reads in the groups and makes them homes and adds them in if
requested

read_file(){
    while read CURRENT_USER
    do
        [ ! -z "MAKE_HOME" ] && create_home
        [ ! -z "CURRENT_GROUP" ] && add_to_group
    done < $1
}

#
```

```
# this checks for options when the script is called
#

while getopts f:ughlUP SWITCH
do
        case $SWITCH in
                f) USER_LIST_FILE=$OPTARG;;
                u) CURRENT_USER=$OPTARG;;
                g) CURRENT_GROUP=$OPTARG;;
                h) MAKE_HOME=YES;;
                l) HOME_DIR_LOCATION=$OPTARG;;
                t) USER_TEMPLATE_LOCATION=$OPTARG;;
                U) LDAP_ADMIN=$OPTARG;;
                P) LDAP_ADMIN_PASS=$OPTARG;;
                  *)  echo "script to create home dirs and add AD users
to OD groups for 10.3 server"
                       echo useage: $PROGRAM [ -f file location ] [ -u
user] [ -g group ] [ -h ] [ -l home folder location ] [ -t home folder
template location] [ -U LDAP Admin user ] [ -P LDAP Admin user's
password ]
                  exit 1;;
        esac
done

check_root

[ ! -z "CURRENT_GROUP" ] && check_admin

if [ ! -z "$USER_LIST_FILE" ]
then
    read_file $USER_LIST_FILE
else
        [ ! -z "MAKE_HOME" ] && create_home
        [ ! -z "CURRENT_GROUP" ] && add_to_group
fi
```