



# **Best Practices: Integrating Mac OS X with Active Directory**

Technical White Paper  
September 2007

# Contents

<b>Page 3</b>	<b>Apple's Built-In Solution</b>
<b>Page 4</b>	<b>How to Integrate Mac OS X with Active Directory</b>
	Getting Started
	dsconfigad
	Testing
	Troubleshooting
	Common Problems
	Passwords
	Single Sign-On
<b>Page 8</b>	<b>Deployment Strategies</b>
	Managed Preferences
	Home Directories
<b>Page 10</b>	<b>Conclusion</b>
<b>Page 11</b>	<b>Appendix</b>
	Extending the Active Directory Schema
<b>Page 13</b>	<b>Additional Resources</b>

# Apple's Built-In Solution

## Third-party add-on solutions

If your deployment requires DFS shares or GPOs, you can choose a third-party product to extend the capabilities of the Apple solution.

- **Group Logic's ExtremeZIP.** With this Apple Filing Protocol (AFP) server for Windows servers, Mac clients can access files on a Windows server using their native file-sharing protocol, AFP. See [www.grouplogic.com](http://www.grouplogic.com).
- **Centrify's DirectControl.** This AD plug-in enables Mac OS X to use AD GPOs without requiring any schema modification. See [www.centrify.com](http://www.centrify.com).
- **Thursby's ADmitMac.** This directory services AD plug-in and SMB client supports DFS shares. See [www.thursby.com](http://www.thursby.com).

A key component of any modern computing environment, directory services allow organizations to centralize information about users, groups, and computing resources. A network-based repository consolidates resources, simplifies system management, and reduces support and administration costs. At the same time, it benefits users by enabling them to access enterprise resources from anywhere on the network. Thus, a directory services infrastructure offers advantages for both administrators and end users.

Of course, the full benefits of directory services can only be realized when all of your desktop, laptop, and server systems are integrated into the same directory services infrastructure. This goal has been difficult to achieve in the past due to the proliferation of proprietary directory services solutions.

With the introduction of the Active Directory (AD) plug-in in Mac OS X v10.3 (Tiger), Apple made a concerted effort to enable IT administrators to integrate Mac OS X clients and servers easily into existing Active Directory infrastructures. While every Active Directory installation is different (especially in the enterprise space), Mac OS X integrates well with the vast majority of them—and with minimum effort. Whatever combination of Mac, Windows, and Linux systems your organization uses, you no longer need to maintain a separate directory or separate user records to support your OS X systems. Users can move effortlessly between different computers while still adhering to enterprise policies for strong authentication and password-protected access to network resources.

When fully integrated, Mac OS X offers a complete managed environment where:

- Users can be fully controlled and must abide by the AD password policies.
- Users have single sign-on access to the AD network through Kerberos.
- Users can have network-based home directories, local home directories, or a combination of the two called Portable Home Directories, which are similar to roaming profiles on Windows.

Apple's support for AD extends to Mac OS X Server as well. Integrating a server is just as easy as integrating a client system. This allows Windows-based departments to take advantage of file sharing and other services in Mac OS X Server, while using their existing AD infrastructure for identification and authentication. Secure network services (including network home directories) hosted on Mac OS X Server also support single sign-on for both Mac OS X and Windows clients.

Because every AD system is unique, don't hesitate to contact your Apple Account Executive regarding your needs. They will arrange for a System Engineer or Professional Services representative to talk with you in greater detail regarding the capabilities of Apple's AD plug-in.

# How to Integrate Mac OS X with Active Directory

## Computer accounts

Each Mac system has a unique computer account in Active Directory. If you clone a system or integrate NetBoot with AD, all of the cloned systems are assigned the same computer account. This means that it's important to take care when changing a computer account, as any change will break authentication from all systems using that account.

## Getting Started

Using these simple steps, you can configure a Mac client to use DNS and AD to determine the geometry of your AD domain, find the nearest domain controller, and create a new computer account in the domain—if there isn't an existing one with the computer ID that you have chosen.

On the Mac client, open the Directory Utility (or Directory Access in Mac OS X 10.4 or earlier) application in the Utilities folder: /Applications/Utilities. If it's not already unlocked, click the lock at the bottom left of the window, and authenticate as a local admin. Choose Active Directory and click Configure.

Leave the advanced options alone for now and enter the name of your Active Directory domain. The computer's account in AD will reflect the Computer ID in this window—make sure it reads correctly before proceeding. Click Bind.

Enter the user name and password of a user who has permission to bind clients to the Computer OU that you specify. This does not need to be an "admin" user—you may assign the privilege to any user. Click OK.

## dsconfigad

The functionality of the Directory Access graphical user interface (GUI) is also accessible from the CLI with the `dsconfigad` command. For example, the following command would bind a system to AD:

```
dsconfigad -f -a COMPUTERTNAME -forest example.com -domain  
example.com -u administrator -p 'password'
```

Once you've bound a system to the domain, you can use `dsconfigad` to set the administrative options that are available in Directory Access:

```
dsconfigad -preferred ads01.example.com -multidomain disable -  
groups domain admins@example.com, enterprise admins@example.com
```

When using `dsconfigad` in a script, you must include the cleartext password that was used to bind to the domain. Typically, an AD user with no other admin privileges is delegated the responsibility of binding clients to the domain. This user name and password pair is stored in the script.

Once dsconfigad has been used to bind the client to the AD domain, use the dscl command to add AD to the local system's authentication and contacts:

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

## Testing

You can quickly test for a successful bind from the command line.

Use the id command to check that the system can use AD to identify users and groups in the domain. The id command returns only the first 16 groups of which the user is a member, even though the client system is aware of all groups:

```
id <AD user shortname>
```

Next, test authentication using the su command to switch to an AD user:

```
su <AD user shortname>
```

You can also use dscl, the DirectoryService command-line tool, to check that the client can iterate the users in the domain. In a layout similar to a file system, dscl presents all directory services information of which the system is aware.

## Troubleshooting

Start by enabling directory services debug logging:

```
sudo killall -USR1 DirectoryService
```

Now when you attempt to bind, you can look at the log to see what is going on:

```
/Library/Logs/DirectoryService/DirectoryService.debug.log
```

When you have accomplished a successful bind, use the same command to disable the debug logging:

```
sudo killall -USR1 DirectoryService
```

For more verbose API-level debugging, you can use a -USR2 command, although typically this is not necessary. To conserve disk space, a -USR2 DirectoryService command stops executing after five minutes.

In some cases, you will need to have directory services debugging enabled at login. To do this, create an empty file that will enable debug logging at boot:

```
/Library/Preferences/DirectoryService/.DSLogDebugAtStart.
```

It may also be helpful to examine a packet trace of the client attempting to bind to the domain. Try capturing traffic for the following ports:

UDP 53	- DNS
TCP 88	- Kerberos
TCP 389	- LDAP
TCP/UDP 464	- Kerberos Password Changes (KPasswd)
TCP 3268	- Global Catalog (LDAP)

For example, to capture traffic over the built-in Ethernet connection to a file called “capture.out,” you could use the following syntax for tcpdump:

```
tcpdump -i en0 -s 0 -w capture.out port 88 or port 464  
or port 53 or port 389 or port 3268
```

## Common Problems

### DNS service

Since AD relies on DNS SRV service records, the Mac client must be using the same DNS servers as all of the Windows clients on the network. Use the dig command to test that the Mac can read the proper DNS records. In the following example, replace example.com with the DNS of your AD domain:

```
dig -t SRV _ldap._tcp.example.com
```

This should return the IP address of your domain controller. If it doesn't, your Mac systems are not using the same server for DNS as the AD clients, or your DNS server is misconfigured.

Beginning with Mac OS X 10.5, the Mac OS X client will attempt to dynamically update DNS records hosted by Active Directory.

### .local domains

Since Mac OS X uses the .local domain for Bonjour (link-local addressing), it will conflict with any .local AD domain. To get around this, add .local to the search domain settings in the Network preference pane. All .local DNS queries will be unicast to the DNS servers before being multicast to the network.

### Replication

In versions earlier than Mac OS X 10.5, when binding a Mac to a large AD domain, the computer account may be created on one domain controller while the Mac attempts to set the computer account's password on another. If the replication interval between the two systems is not quick enough, the password set request will fail, and the Mac will not be bound to the domain.

To address this problem, make sure that the sites are correctly set up and that the domain controllers within the sites are properly replicating to one another. Another alternative is to create the computer account in AD before binding the Mac. When the Mac binds, the Directory Access application will ask if you'd like to use the existing computer account. This eliminates the chance of a replication error during the bind process, since replication has already occurred.

This situation has been addressed in Mac OS X 10.5—the client will ensure that the same server is used for both Kerberos and LDAP connections.

## Passwords

Since Mac OS X leverages Kerberos, it inherently supports AD password policies and enforces restrictions on the length and complexity of passwords on client systems. Mac users can also change their passwords using the Accounts preference pane on the Mac OS X system.

In the days leading up to password expiration, users are notified during login and other authentication events that their password is about to expire. This gives them the opportunity to change their password in AD—which will reset the expiration timer—using the Accounts preference pane on the Mac client. When the password is within 24 hours of expiration, users cannot complete login until they have changed their password.

### Site awareness

Apple's AD plug-in is AD site aware. It queries the global catalog for site information and polls the site's domain controllers. From those that respond, the plug-in chooses two and uses them until a network change occurs or until one of the domain controllers stops responding.

Similar to a Windows system, the Apple plug-in creates a computer account in AD during the bind process. However, the Mac continues to use this account for all user look-ups—unlike a Windows system, which switches to using the user credentials once someone has logged in.

Beginning with Mac OS X 10.5, the computer account password is cycled. The default is every 14 days, but you can use the `dsconfigad` command-line tool to set any interval that your site's policy requires.

#### Windows Server 2003 R2

The AD plug-in has been successfully tested with Windows Server 2003 R2. The domain can be in either native or mixed mode without any change in the functionality of the Mac OS X clients.

### Single Sign-On

Microsoft and Apple both support MIT's Kerberos to provide a secure single sign-on environment. When configured to participate in the AD domain, a Mac uses Kerberos exclusively for all authentication activities. If desired, the use of NTLMv1 and NTLMv2 can be prohibited on the network with no impact on Apple-provided services integrated with Active Directory.

When an AD user logs in to a Mac, the AD domain controller automatically issues a Kerberos TGT. When the user attempts to use any service on the domain that supports Kerberos authentication, the TGT generates a ticket for that service, without requiring the user to authenticate again.

You can use the normal Kerberos administration tools to view tickets currently issued to a user. For example, `klist` from the command line shows the current tickets, and the Kerberos utility, `/System/Library/CoreServices/Kerberos.app`, displays the same information through a graphical interface.

### Namespace Support

Mac OS X 10.5 now offers the option of supporting multiple users with the same shortnames that exist in different domains within the AD forest.

By enabling namespace support, using the `dsconfigad` command line tool, users will have to log in using their domain followed by their shortname (`DOMAIN\shortname`), similar to logging in to a Windows machine.

### Signed Connections

The AD plug-in is able to both sign and encrypt the LDAP connections that are used to communicate with Active Directory. Along with the signed SMB support that is present in Mac OS X 10.5, you should have no need to downgrade your site's security policy to accommodate any Mac clients. Additionally, the signed and encrypted LDAP connections eliminate any need to use LDAP over SSL.

# Deployment Strategies

## Managed Client for OS X (MCX)

Since Windows and Mac OS X handle preferences differently, a Mac is unable to use GPOs in AD. Instead, Apple has a system called MCX that accomplishes the same task.

MCX can be stored locally on Mac clients that have been integrated into AD, but this makes updates difficult because it involves each individual computer. It's also possible to host the MCX objects in AD, which requires you to extend the schema. Another solution is to configure a secondary LDAP directory using Mac OS X Server and Apple's Open Directory. In this scenario, clients still use AD for user authentication, while OD supplies managed preferences only.

## Managed Preferences

When fully integrated, Mac OS X offers a complete managed environment where users can be fully controlled, and they are required to abide by AD password policies. Depending on the level of management your organization requires, there are several options for managing Mac client preferences.

**Do nothing.** Apple's plug-in automatically enables authentication to AD, including full support of password policies. It also allows you to set up network homes for Mac users on AD.

**Extend the AD schema to handle management.** By adding 38 attributes and 10 classes to the AD schema, your AD system can support all Mac OS X management policies. Just use the normal Mac OS X management tools and target the AD domain.

**Dual directory.** Sometimes known as the "magic triangle," this scenario adds Mac OS X Server to the solution. Mac clients integrate with AD and with an Open Directory (OD) domain on Mac OS X Server. AD users and groups are nested inside OD groups. Mac OS X 10.5 further enhances this scenario with "augmented records" that allow information from a secondary directory to be added to information directly from AD for the same record. This solution does not require any change to the AD schema, but it does require Mac OS X Server.

**Third-party solutions.** Products from Thursby and Centrify allow managed preferences to be stored in the AD domain without requiring you to extend the schema. With the Thursby solution, you use Mac OS X tools to create user preferences, while the Windows-based Centrify solution allows you to manage all of the preferences using native AD tools.

## Home Directories

Regardless of your strategy for managed preferences, you can set up users with local homes, network homes, or a combination of the two called Portable Home Directories, which are similar to roaming profiles on Windows.

**Local (do nothing).** With the default configuration of Apple's AD plug-in, the user's home stays on the local system, without any change to the user record in AD. If a network home is defined in the user record, that share will mount on the desktop when the user logs in.



#### **AFP network homes**

It's also possible to use an `afp://` URL for your homes. In AD, the URL remains in the standard Universal Naming Convention (UNC). On the Mac side, however, you can allow the client to translate the SMB path into an AFP path.

**Network.** To define a network home in the Mac user's AD record, use a URL in the form of `\\server\share\user`—just as you would for a Windows user. When interpreted by the AD plug-in on the Mac, the server name will be added to the AD domain, forming a URL: `smb://server.ad.domain/share/user`.

Note: If the user's domain is different from the domain of the user's home folder, it may be necessary to put the fully qualified name of the server in the URL. So, instead of `//server/share/user`, you would use `//server.userad.domain/share/home`. Using a Mac friendly naming convention does not affect the Windows systems on the network.

The Mac user's network home can be hosted on either a Mac OS X Server or a Windows server, using either AFP or SMB. You can even host home directories for both Mac OS X and Windows clients on Mac OS X Server, providing Mac services over AFP and Windows services over SMB.

**Portable Home Directory.** In this scenario, the AD user record and the network home are cached locally on the client system, making it ideal for managing laptop users when they are away from the network. According to a sync policy, the local system synchronizes with the remote home folder. For laptops, this typically happens when they reconnect to the network. Portable Home Directories can also be useful for managing desktop users. You decide how often the client syncs and what files are included in the sync—and allow them to operate offline the rest of the time.

# Conclusion

Apple's support for Active Directory within Mac OS X enables Mac clients and servers to integrate smoothly into existing AD environments, and provides the option of deploying a single directory services infrastructure that can support both Windows and Mac clients.

If you have any questions about the best practices discussed in this paper, or any other aspect of integrating Mac OS X systems with Active Directory, please contact your Apple Account Representative for assistance.

# Appendix

## Apple Professional Services

If you need assistance in extending the AD schema, contact Apple Professional Services, a team of expert consultants who provide a range of deployment and integration services. For more information about Apple Professional Services, see [www.apple.com/services/consulting](http://www.apple.com/services/consulting).

## Flexible Single Master Operations

The operations for some functions in Active Directory, such as extending the schema, can only be performed on one domain controller.

## Extending the Active Directory Schema

Use the following scripts as a starting point to extend the schema of your Active Directory installation.\*

### What's Included

A collection of schema to get full MCX functionality for Mac OS X clients.

### Usage

For your schema master, use the domain controller assigned to the Flexible Single Master Operations (FSMO) role. You can identify the FSMO by following the procedure on this website: <http://support.microsoft.com/kb/234790/en-us>.

1. Enable schema writing on the FSMO by double-clicking `writenableschema.reg`.
2. Get the AD domain name, if you don't already know it. From a Mac, you can use this CLI command:

```
ldapsearch -x -h <ipaddress of DC> -b "" -s base  
rootDomainNamingContext
```

Take the value from this line:

```
rootDomainNamingContext: DC=demotree,DC=com
```

3. Determine whether you want to run the "normal" or the "full" set of extensions. Run the `load_apple.bat` script on the FSMO and give it the AD domain that you found in step 2. Make sure that you keep the associated schema files in the same folder as the `.bat` script.

```
load_apple "DC=demotree,DC=com"
```

4. When you are done, disable schema writing on the FSMO by double-clicking `writedisableschema.reg`.
5. Ensure that the Mac computer accounts in AD can read the new schema. All computers in the domain must be able to read the attributes listed below. By default, the new attributes will be readable unless you have already changed directory permissions.

Note: A Visual Basic script that does the correct thing is available from your Apple representative.

### Attributes (38) and classes (10) from the “normal” set

#### Attributes:

dn: CN=apple-category,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-computer-list-groups,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-computeralias,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-computers,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-data-stamp,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-dns-domain,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-dns-nameserver,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-dnsname,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-group-homeowner,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-group-homeurl,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-imhandle,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-keyword,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-Managed Client Preferencesflags,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-Managed Client Preferencessettings,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-mountDirectory,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-mountDumpFrequency,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-mountOption,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-mountPassNo,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-mountType,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-neighborhoodalias,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-networkview,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-nodepathxml,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-service-location,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-service-port,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-service-url,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-user-authenticationhint,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-user-class,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-user-homequota,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-user-homesoftquota,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-user-mailattribute,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-user-picture,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-user-printattribute,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-webloguri,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-xmlplist,CN=Schema,CN=Configuration,DC=X  
dn: CN=ipHostNumber,CN=Schema,CN=Configuration,DC=X  
dn: CN=loginShell,CN=Schema,CN=Configuration,DC=X  
dn: CN=macAddress,CN=Schema,CN=Configuration,DC=X  
dn: CN=ttl,CN=Schema,CN=Configuration,DC=X

#### Classes:

dn: CN=apple-computer,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-computer-list,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-configuration,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-group,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-location,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-mount,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-neighborhood,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-serverassistant-config,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-service,CN=Schema,CN=Configuration,DC=X  
dn: CN=apple-user,CN=Schema,CN=Configuration,DC=X

\* These scripts are not provided with any warranty or supported by Apple Inc. support groups. They are not meant for end users and are provided as a courtesy and as a guide for extending the schema of Active Directory to include Apple attributes.

# Additional Resources

For more information about integrating Mac OS X clients into AD environments—including documentation, training, articles, scripts, and discussions—visit the following sites:

[www.apple.com/server/documentation](http://www.apple.com/server/documentation)

[www.apple.com/training](http://www.apple.com/training)

[www.apple.com/services/consulting](http://www.apple.com/services/consulting)

[www.afp548.com](http://www.afp548.com)

[www.bombich.com/mactips/activedir.html](http://www.bombich.com/mactips/activedir.html)

[www.macenterprise.org](http://www.macenterprise.org)

## For More Information

For more information, please contact your local Apple Sales Representative, or call Apple Enterprise Sales at 877-412-7753.

© 2007 Apple Inc. All rights reserved. Apple, the Apple logo, Bonjour, Mac, Mac OS, and Tiger are trademarks of Apple Inc., registered in the U.S. and other countries. Other product and company names mentioned herein may be trademarks of their respective companies. September 2007 L334436A