

Mac OS X System Administration

Leon Towns-von Stauber, Occam's Razor

LISA 2003

<http://www.occam.com/osx/>

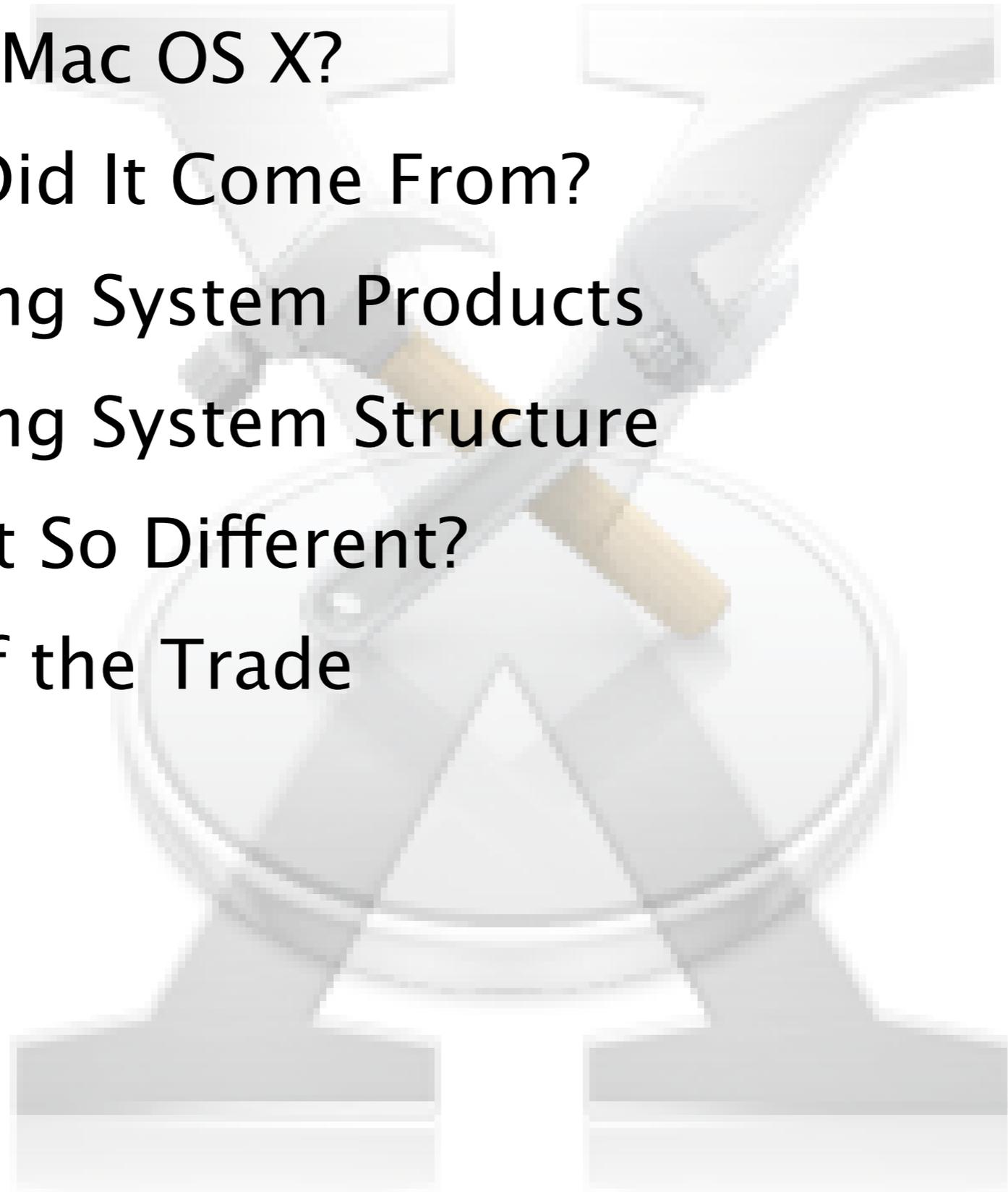
Opening Remarks.....	3
Orientation.....	6
Booting Up.....	26
Disk Volumes and Filesystems.....	41
Software Installation.....	56
Directory Services.....	62
Networking.....	89
Web Services.....	105
Mail Services.....	121
File Sharing.....	127
Printing.....	144
Resources.....	151
Closing Remarks.....	155

- I'm assuming familiarity with:
 - UNIX design: account management, filesystems, network services, etc.
 - Common UNIX software: Apache, BIND, OpenLDAP, Samba, etc.
 - Mac OS X user interface
- Where I'm coming from:
 - UNIX user and some-time admin since 1990
 - Full-time UNIX admin since 1995
 - NeXTstep/OS X user and admin since 1991
- An operating system is a big topic
 - Due to the amount of ground we need to cover, at many points I'll just be skimming the surface
 - Please ask questions as we go if you want to get more in-depth on a topic

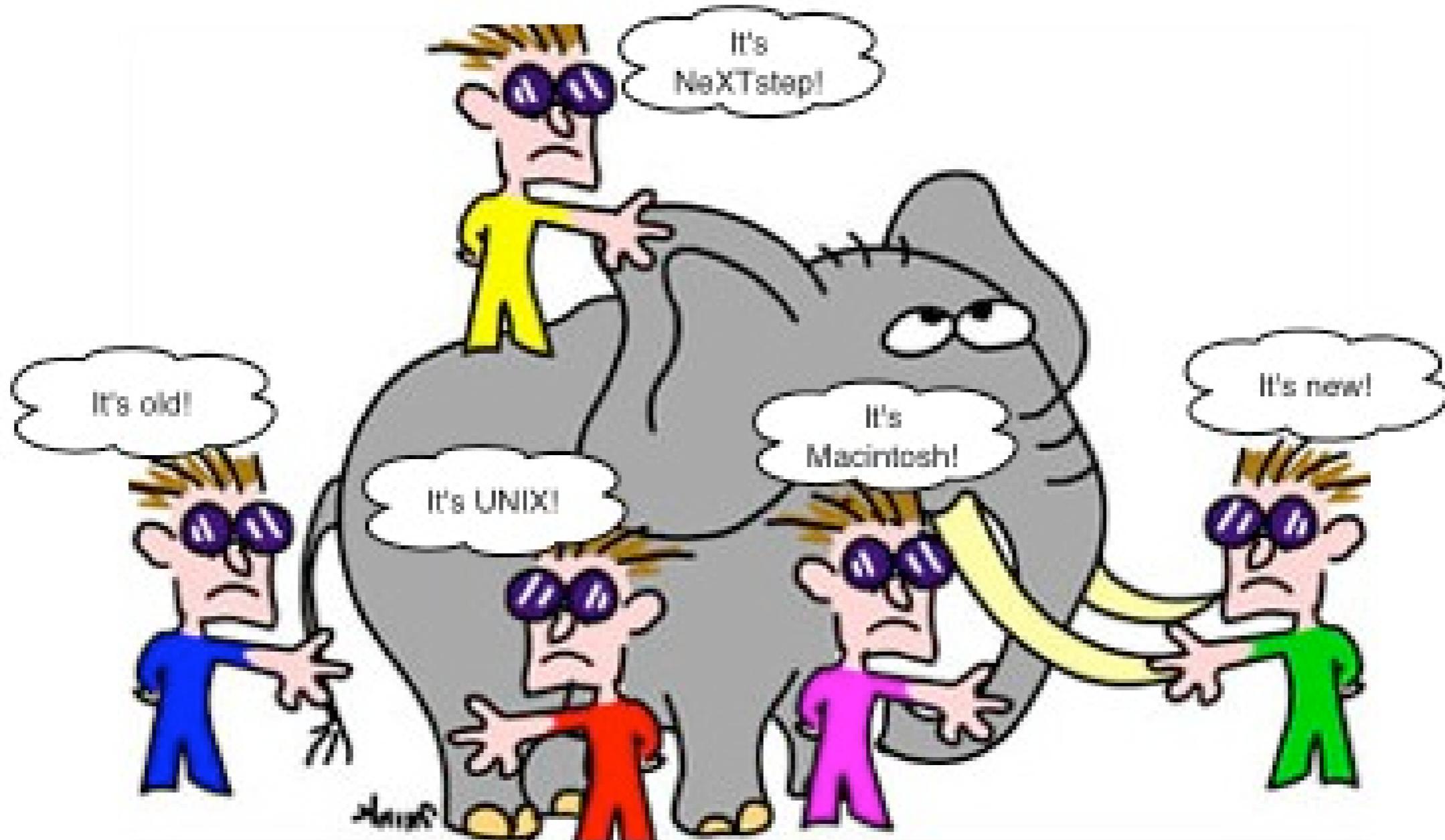
- This presentation primarily covers:
 - Mac OS X 10.2.6 (Darwin 6.6)
 - Mac OS X Server 10.2.6 (Darwin 6.6)
 - Includes some updates for Panther, Mac OS X 10.3 (Darwin 7.0)

- This presentation Copyright © 2003 Leon Towns–von Stauber. All rights reserved.
- Trademark notices
 - Apple®, Mac®, Macintosh®, Mac OS®, Aqua®, Finder™, Quartz™, Cocoa®, Carbon®, AppleScript®, Rendezvous™, Panther™, and other terms are trademarks of Apple Computer. See <http://www.apple.com/legal/appletmlist.html>.
 - NeXT®, NeXTstep®, OpenStep®, and NetInfo® are trademarks of NeXT Software. See <http://www.apple.com/legal/nextttmlist.html>.
 - PowerPC™ is a trademark of International Business Machines.
 - Java™ is a trademark of Sun Microsystems.
 - Other trademarks are the property of their respective owners.

- What Is Mac OS X?
- Where Did It Come From?
- Operating System Products
- Operating System Structure
- Why Is It So Different?
- Tools of the Trade



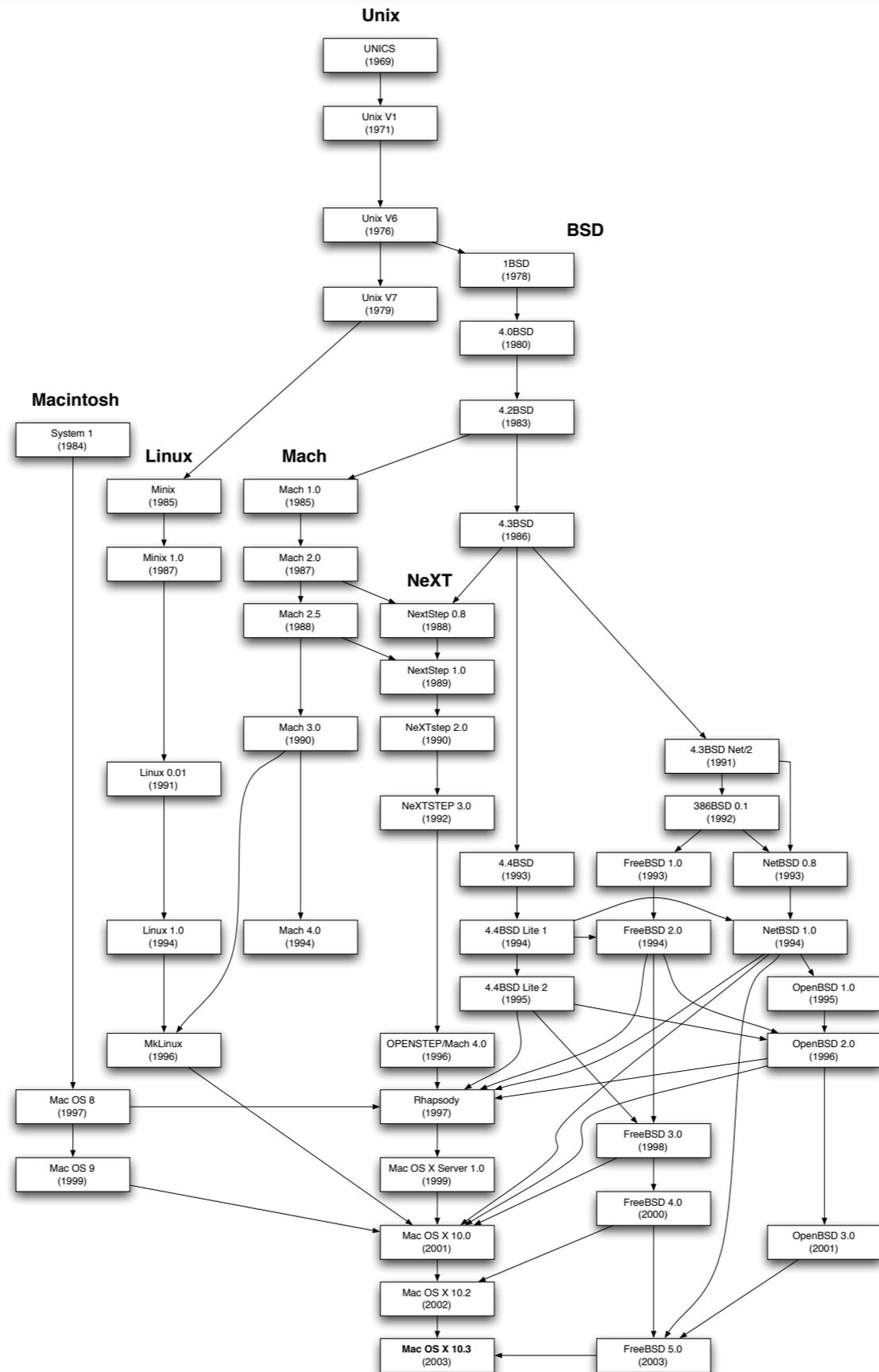
- It's an elephant



- I mean, it's like the elephant in the Chinese/Indian parable of the blind men, perceived as different things depending on the approach

- Inheritor of the Mac OS legacy
 - Evolved GUI, Carbon (from Mac Toolbox), AppleScript, QuickTime, etc.
- The latest version of NeXTstep
 - Mach, Quartz (from Display PostScript), Cocoa (from OpenStep), NetInfo, apps (Mail, Terminal, TextEdit, Preview, Interface Builder, Project Builder, etc.), bundles, faxing from Print panel, NetBoot, etc.
- A new flavor of UNIX
 - More specifically, a BSD UNIX variant
 - Full set of command-line utilities, libraries, server software, etc.
- All of the above

- A (Very) Brief History of Time, acc. to Steven P. Jobs
 - 1985: Jobs leaves Apple and founds NeXT Computer
 - 1988: NextStep 0.8 and the first NeXT Computer are released
 - 1996: NeXT purchased by Apple
 - 1997: Jobs returns to Apple (with some NeXT compatriots) and eventually becomes CEO
 - 2001: Mac OS X released for general use
- Progenitors
 - UNIX components primarily based on FreeBSD
 - Also NetBSD and OpenBSD, as well as NeXTstep's version of BSD
 - Kernel based on Mach 3.0, MkLinux, and NeXT Mach

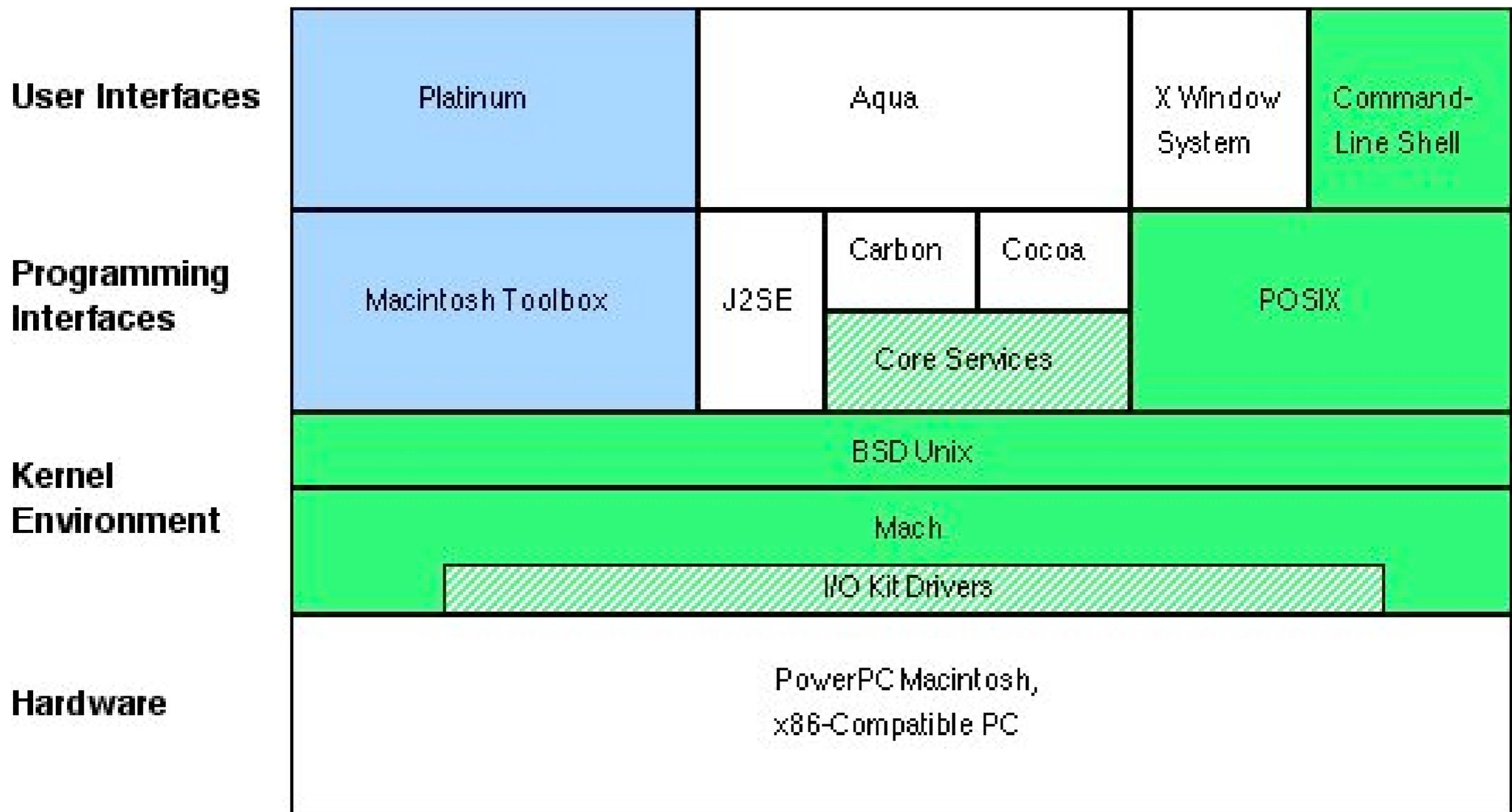


Operating System Ancestry of Mac OS X

- Mac OS X
 - Apple's flagship operating system
- Classic
 - An instance of Mac OS 9 running in a self-contained execution environment within Mac OS X
- Darwin
 - The open-source foundation of Mac OS X
- Mac OS X Server
 - Mac OS X with additional server and administrative software
 - Two license variations: 10-client and unlimited client
 - Number of clients is in reference to AFP, NetBoot, Macintosh Manager; traditional UNIX services unaffected



Hexley, the unofficial Darwin mascot



Classic



Darwin

The Structure of Mac OS X

- User Interfaces
 - Aqua
 - OS X is the only widely used UNIX with a native GUI not based on X11
 - X Window System (X11R6)
 - Implementations from Apple or third parties, based on XFree86
 - Included in Panther
 - Platinum (Classic environment)
 - BSD UNIX command line
 - Via Terminal application, SSH, single-user, `>console` login, Darwin

- Programming Interfaces
 - Macintosh Toolbox
 - Mac OS 9 executables run under Classic
 - POSIX(ish), for UNIX programs
 - Java 2 Platform, Standard Edition
 - Carbon
 - Overhaul of Macintosh Toolbox to support advanced features
 - Cocoa
 - Evolution of OpenStep

- Kernel Environment
 - BSD UNIX
 - Multiuser security (users, groups, file permissions), process model (forks, threads), network access (sockets)
 - Filesystems: HFS/HFS+, UFS, FAT, ISO 9660, UDF, AFP, NFS, SMB, ...
 - Mach
 - Developed at CMU as an experiment in microkernel design
 - Early versions integrated BSD, which NeXT used
 - Mac OS X kernel primarily derived from Mach 3.0 used in MkLinux, with NeXT enhancements
 - Still a monolithic kernel, for performance
 - Manages memory, processes, and hardware access
 - VM is file-based, with swapfiles created dynamically in `/var/vm/`

- Some important differences: Quartz vs. X11, HFS+ vs. UFS, Objective-C vs. C++, NetInfo vs. LDAP, AFP vs. NFS, file-based VM, etc.
- Many design decisions were made in the middle to late 1980s, during the development of NeXTstep
 - Many of today's ubiquitous technologies (X11, C++, YP/NIS, LDAP) were not yet well-established
 - NeXT was among the first to implement a UNIX GUI, a standard OO dev environment, directory services, etc., and happened to choose differently than the rest of the industry later did (in some cases, by developing proprietary technologies)
- Some changes were made to support Apple's existing user base
 - HFS+, AFP, secure default config

- But why does Apple stick with technologies that require special training?
- Because some are just better than the alternatives
 - Objective-C is a cleaner, more flexible language than C++
 - HFS+ is arguably more capable than UFS under certain circumstances
 - Quartz performs well and is self-consistent
 - NetInfo scales easily and has superior management tools (so far)
 - AFP offers security advantages over NFS
- Apple controls these technologies, and can drive their improvement

- Brief introduction to some applications of general utility
 - Others will be mentioned as we go along
- GUI apps
 - System Preferences (Accounts, Sharing, Network, etc.)
 - Apple System Profiler, Process Viewer, CPU Monitor
 - Terminal (for command-line access)
- CLI tools
 - Usual set of UNIX/BSD tools: `sysctl`, `pstat`, `fstat`, `top`, **etc.**
 - `vm_stat` is the best way to keep an eye on paging
 - Watch free pages and pageouts
 - `hostinfo`, `sw_vers`, `system_profiler`
 - `systemsetup`, `networksetup`

Apple System Profiler

System Profile | Devices and Volumes | Frameworks | Extensions | Applications | Logs

▼ Software Overview

- System version: Mac OS X 10.2.6 (6L60)
- Boot volume: Mac OS X Server
- Kernel version: Darwin Kernel Version 6.6: Thu May 1 21:48:54 PDT 2008; root:xnu-6.6.0/RELEASE_ARM_T8020
- User name: Leon Towns-von Stauber (leonvs)

▼ Hardware Overview

- Machine speed: 700 MHz
- Bus speed: 100 MHz
- Number of processors: 1
- L2 cache size: 512K
- Machine model: iBook (version = 1.12)
- Boot ROM info: 4.3.6f3
- Customer serial number: UV23107A-LQ5-ff10
- Sales order number: Not available

▼ Memory Overview

Location	Type	Size
DIMM0/BUILT-IN	SDRAM	128 MB
DIMM1/J12	SDRAM	256 MB

▼ Network Overview

▼ Built-in

Flags	0x8863<Up,Broadcast,b6,Running,Simplex,Multic...
-------	--------------------------------------------------

Kernel configured for up to 2 processors.
1 processor is physically available.
Processor type: ppc750 (PowerPC 750)
Processor active: 0
Primary memory available: 384.00 megabytes.
Default processor set: 40 tasks, 90 threads, 1 processors
Load average: 0.06, Mach factor: 0.94
{vamana}@leonvs[~]: vm_stat
Mach Virtual Memory Statistics: (page size of 4096 bytes)
Pages free: 51253.
Pages active: 15746.
Pages inactive: 20786.
Pages wired down: 10519.
"Translation faults": 921213.
Pages copy-on-write: 28929.
Pages zero filled: 445417.
Pages reactivated: 0.
Pageins: 5893.
Pageouts: 0.
Object cache: 9637 hits of 13571 lookups (71% hit rate)
{vamana}@leonvs[~]:

Process Listing

Find: Show: All Processes

Name	User	Status	% CPU	% Memory
Window Manager	leonvs	Running	3.40	5.50
Apple System Profiler	leonvs	Running	0.00	2.70
Finder	leonvs	Running	0.00	2.20
loginwindow	leonvs	Running	0.00	2.10
Dock	leonvs	Running	0.00	1.80
coreservicesd	root	Running	0.00	1.60
PTHClock	leonvs	Running	0.00	1.60
Terminal	leonvs	Running	0.00	1.40
Process Viewer	leonvs	Running	3.00	1.30
SystemUIServer	leonvs	Running	0.00	1.10
CPU Monitor	leonvs	Running	7.00	1.00
DirectoryService	root	Running	0.00	0.80
UniversalAccessApp	leonvs	Running	0.00	0.60

37 processes Sample every 3 seconds

▼ Less Info

Process ID	Statistics
Total CPU Time:	5:52PM
Virtual Memory Size:	57,188 kbytes
Resident Memory Size:	10,500 kbytes

```
lrwxrwxr-t 1 root admin 11 Sep 16 18:56 tmp -> private/tmp
drwxr-xr-x 10 root wheel 340 Jan 30 2003 usr
lrwxrwxr-t 1 root admin 11 Sep 16 18:56 var -> private/var
{vamana}@leonvs[~/]:
```

Administrative Applications

The screenshot displays two windows from the Mac OS X Panther administrative tools. The 'System Profile' window on the left shows hardware and software information for a machine named 'Vamana'. The 'Activity Monitor' window on the right shows a list of running processes and system resource usage.

System Profile - Hardware Overview

Machine Model:	iBook
CPU Type:	PowerPC
Number Of CPUs:	1
CPU Speed:	700 MHz
L2 Cache (per CPU):	512 KB
Memory:	384 MB
Bus Speed:	100 MHz
Boot ROM Version:	4.36f3
Serial Number:	UV231

Activity Monitor - My Processes

Process ID	Process Name	User	% CPU	# Threads	Real Memory	Virtual Memory
1260	WindowServer	leonvs	0.00	4	20.96 MB	94.43 MB
1230	System Prefer	leonvs	0.00	1	22.84 MB	94.26 MB
1056	tcsh	leonvs	0.00	1	1.65 MB	22.14 MB
1055	Terminal	leonvs	0.00	3	14.14 MB	89.77 MB
999	AppleSpell	leonvs	0.00	1	4.40 MB	36.30 MB
998	TextEdit	leonvs	0.00	2	17.82 MB	90.80 MB
724	Preview	leonvs	0.00	2	27.23 MB	102.68 MB
459	UniversalAcce	leonvs	0.00	1	9.67 MB	85.01 MB
458	Activity Monit	leonvs	6.50	3	19.75 MB	102.82 MB
446	Mirror Agent	leonvs	0.00	2	14.24 MB	85.73 MB
438	Finder	leonvs	0.00	1	27.77 MB	123.57 MB
436	SystemUIServe	leonvs	1.00	1	16.13 MB	89.05 MB
434	Dock	leonvs	0.00	2	9.16 MB	84.59 MB
430	pbs	leonvs	0.00	2	4.11 MB	43.78 MB
275	loginwindow	leonvs	0.00	4	16.20 MB	64.82 MB
269	ATSServer	leonvs	0.00	2	5.31 MB	65.30 MB
233	WindowServer	leonvs	1.00	2	34.58 MB	97.45 MB

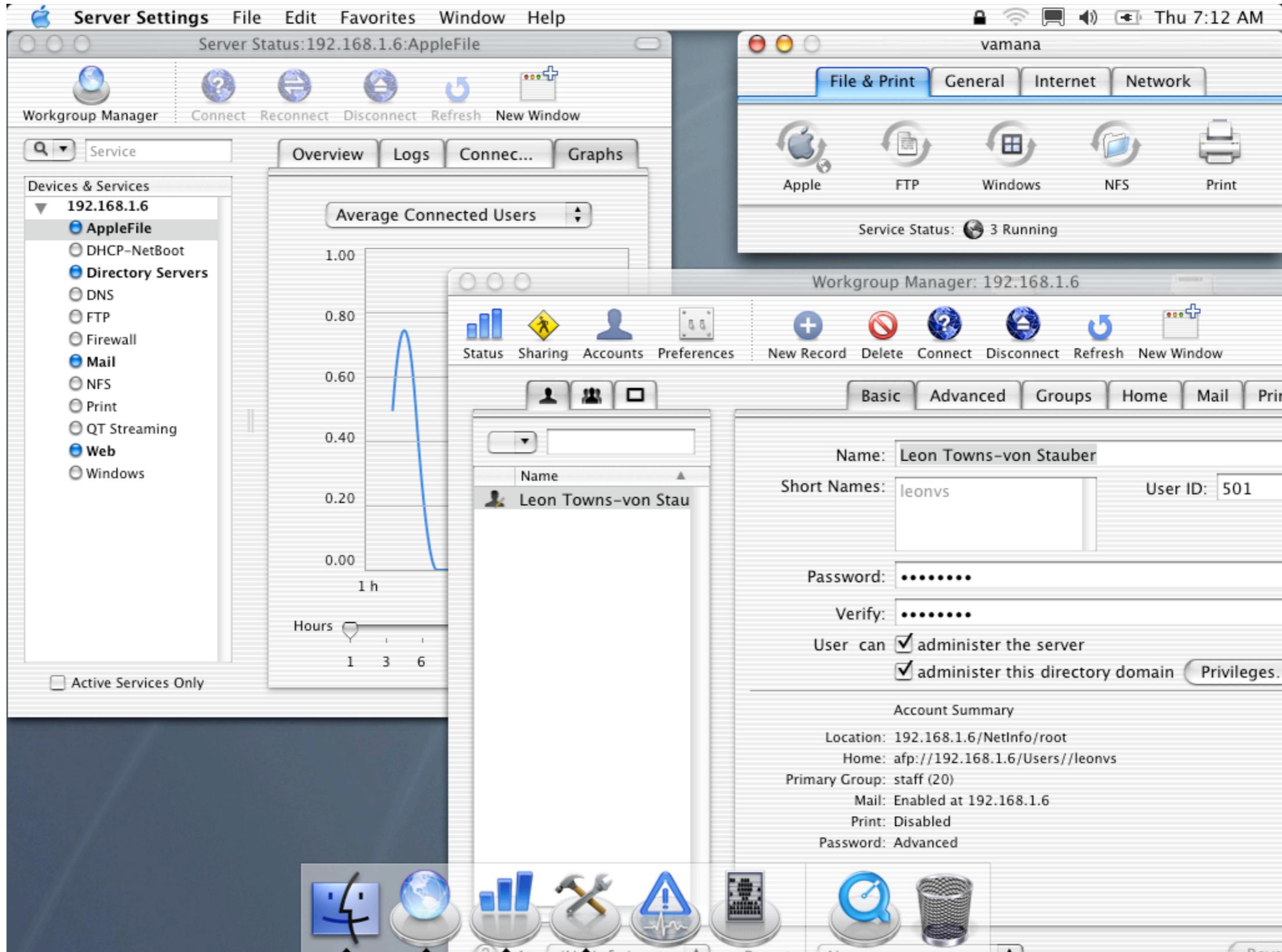
System Resource Usage

Wired:	48.64 MB	Free:	48.53 MB
Active:	125.42 MB	VM size:	2.93 GB
Inactive:	161.41 MB	Page ins/outs:	29403/0
Used:	335.47 MB		

384.00 MB

- Server apps
 - Mac OS X Server includes a set of applications that can be used to manage many of the system's capabilities either locally or remotely
 - Can be installed on any OS X system, not just Server
- Applications
 - Server Settings: Manage network services
 - Workgroup Manager: Manage users, groups, preferences, file shares
 - Server Status: View logs, usage, etc.
 - `serversetup`
 - Panther: Server Settings and Server Status are combined into Server Admin

- Server apps (cont'd.)
 - Daemons
 - `servermgrd` (TCP 687 and 311 (SSL)): Apache, used by Server Status
 - `serversettingsd` (TCP 660): Used by Server Settings and WM
 - **Panther**: `serversettingsd` gone; `servermgrd` now handles everything
 - `DirectoryService` (TCP 625): Used by Workgroup Manager

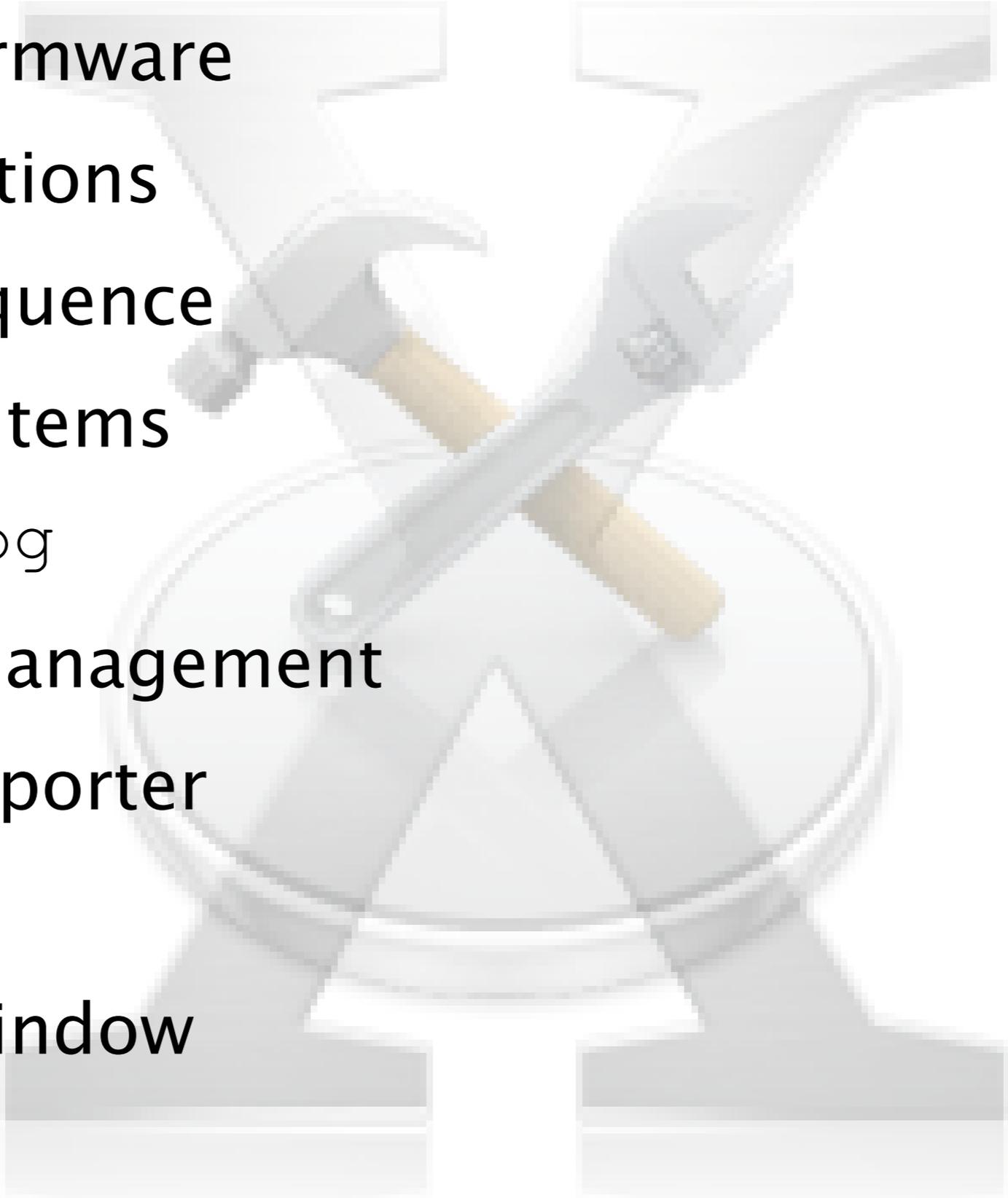


Server Applications

The screenshot displays the Workgroup Manager application on a Mac OS X system. The main window is titled "Server Admin:vamana.local:Server" and shows a "CPU Usage" graph with a peak at 83.33%. The left sidebar lists services for "vamana.local", with "Open Directory" selected. The right sidebar shows "LDAP Settings" and search parameters. A secondary window titled "Workgroup Manager: vamana.local" is open, showing a user list with "Leon Towns-von Stauber" selected and its "Basic" configuration page. The configuration includes fields for Name, User ID (501), Short Names (leonvs), Password, and Verify, along with checked options for "administer the server", "administer this directory domain", and "log in".

Panther Server Applications

- Developer Tools
 - Do install the Dev Tools package, even if you don't plan to develop software
 - Includes things like `make`, `m4`, RCS tools, `otool` (like `ldd`), and HFS-aware file utils

- Open Firmware
 - Boot Options
 - Boot Sequence
 - Startup Items
 - `watchdog`
 - Power Management
 - CrashReporter
 - NetBoot
 - Login Window
- 

- Responsible for initial bootstrapping
- Based on an open standard also used by Sun
- While OS is running, OF variables may be viewed and modified with `nvr`
- Can set an OF password that prevents booting from an alternate device
 - Download Open Firmware Password application from Apple
- Can use TELNET to remotely access another system's OF prompt, which could be useful for debugging if set up to occur automatically
 - 1) On target system, obtain an OF prompt (e.g., with `Cmd-Opt-O-F`)
 - 2) Enter `dev /packages ls`, and check for `/telnet`
 - 3) Enter `"enet:telnet,IP_address" io`
 - 4) On client, TELNET to the IP address
- <http://developer.apple.com/technotes/tn/tn1061.html>

- Open Firmware boot options enabled by holding down keys at startup
 - Verbose (textual startup): `Cmd-V`
 - Single-user: `Cmd-S`
 - Boot from CD-ROM: `C`
 - Boot from network: `N`
 - Boot device selection: `Option`
 - Open Firmware prompt: `Cmd-Opt-O-F`
 - Flash PRAM: `Cmd-Opt-P-R`
- May also be set using `nvr` to change `boot-args`
 - Verbose: `-v`
 - Single-user: `-s`
 - Safe (argument `passwd` to `kextd`): `-x`

- General pattern is the same as most UNIX systems: run bootstrap code from persistent memory, use that to find a kernel and load it into main memory, load hardware drivers, mount filesystems, and progress through a series of initialization programs that start up the services required on a multiuser operation system
- BootROM
 - Located in firmware
 - POST
 - Hardware initialized using drivers in Open Firmware
 - Boot device selected based on NVRAM settings
 - Affected by System Preferences → Startup Disk

- BootX
 - Located in `/System/Library/CoreServices/`
 - This directory is "blessed"; see the `bless` man page for more
 - The blessed directory ID is stored in the Master Directory Block, which is read by BootROM, and which then looks for a file with HFS type `tbxi`
 - Kernel (`/mach_kernel`), drivers, and boot-time kernel extensions loaded into memory
- Kernel initialization
 - Data structures initialized
 - I/O Kit initialized, drivers linked into kernel
 - Root filesystem mounted
 - Mach bootstrap port server (`mach_init`) started

- System initialization
 - `mach_init` starts BSD `init` (PID 1), takes on PID 2
 - `/etc/rc.boot` brings system to single-user
 - Runs `fsck` (unless `/fastboot` exists)
 - `/etc/rc` brings system to multi-user
 - Starts `kextd` to handle kernel extension requests
 - `kextd` also unloads unnecessary drivers
 - Starts virtual memory pager (`dynamic_pager`)
 - Starts System Configuration Server (`configd`) to monitor changes in network status
 - Runs `SystemStarter` to process startup items

- Contained in `StartupItems/` in `/System/Library/` and `/Library/`
- Each item is a directory, containing:
 - Executable named the same as the directory, run with `start` argument
 - `StartupParameters.plist`
 - Description
 - Services provided, required, and used
 - Preference: `First`, `Early`, `Late`, `Last`, `None`
- Startup items can execute in parallel, and the order is not deterministic
- Often enabled/disabled by settings in `/etc/hostconfig`
- `StartupItemManager` (<http://www.septicus.com/>) eases creation and management of custom startup items
- Startup items aren't executed on shutdown, which can cause problems for some things that require handholding, like databases

- Mac OS X Server includes the `watchdog` utility, which reads an `inittab`-like file (`/etc/watchdog.conf`) to handle starting and restarting certain daemons
- Started by `Watchdog` startup item, logs to `/Library/Logs/watchdog.event.log`
- Also resets the automatic reboot timer in the system's power management unit (PMU)
 - If the timer ever expires, the machine suffers a hard reboot
 - Meant to recover a hung system automatically
 - Don't issue `SIGKILL` to `watchdog`, because then it can't disable the timer!
 - Automatic reboot enabled in Energy Saver Preferences
 - Or by `servermode on`, executed by `Watchdog` startup item

- Much of the work in I/O Kit was to support advanced power management capabilities, such as sleep, that weren't traditional UNIX emphases
- Configured in Energy Saver Preferences
 - `pmset` permits manipulation from command line
- You should disable sleep on servers for anything but monitors
- Use Wake550 (<http://www.tc.umn.edu/~olive0003/wake550.html>) to wake sleeping machines set to "Wake for network administrator access"
- Uninterruptible power supplies
 - Jaguar supports some natively (via USB)
 - Default behavior to halt when UPS is under 20% capacity
 - Additional functionality available with third-party software, such as PowerGuardian (<http://www.powerguardian.com/>) and APC Tracker (<http://www.equinux.com/us/products/apctracker/>)

- CrashReporter captures data from system panics and application crashes for later analysis
- Startup item enabled from Console application preferences, or by setting `CRASHREPORTER=-YES-` in `/etc/hostconfig`
- `crashreporterd` **calls** `crashdump` **when an app crashes, logs to** `~/Library/Logs/CrashReporter/`
- **System panics**
 - When the system panics, the dump is saved to NVRAM
 - The startup item runs `panicdump`, which logs to `/Library/Logs/panic.log`
 - **See** <http://developer.apple.com/technotes/tn2002/tn2063.html> **for information on interpreting panic dumps**

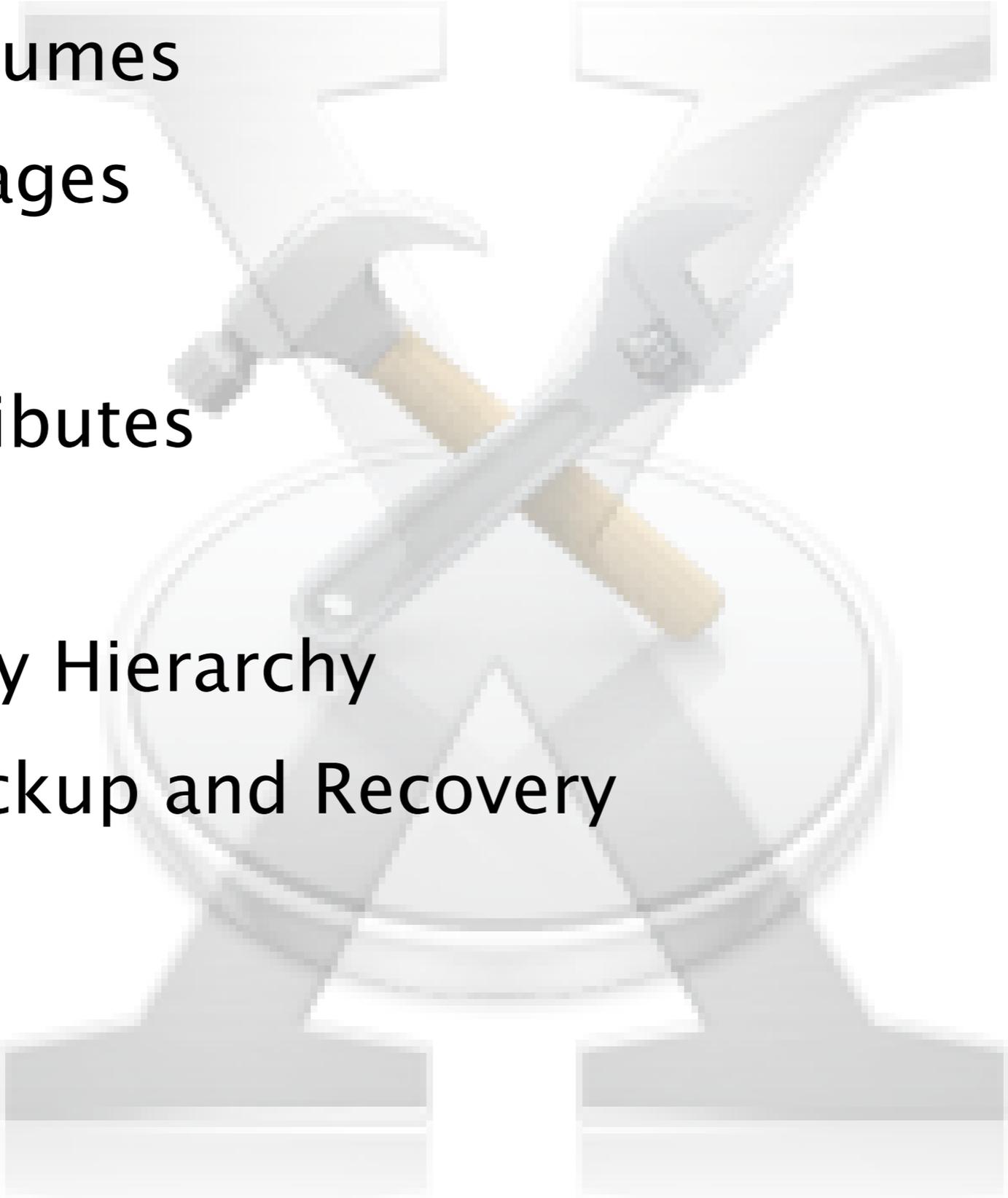
- The BootROM code in Open Firmware contains the ability to boot from a server on the network
- Mac OS X Server includes NetBoot server software
 - Network Image Utility creates NetBoot disk images
 - Server Settings used to manage service
- Based on several protocols: DHCP or BOOTP to assign address, BSDP (Boot Server Discovery Protocol, based on DHCP), TFTP to download files needed for booting, NFS to mount image
 - DHCP and NFS can be on other servers
 - Skip BSDP step by specifying boot server in OF (to boot across subnets)
 - `sudo nvram boot-device=enet:boot_server_IP`
- On client, start with N key held down, to use default image on server, or select network volume in Startup Disk Preferences

- NetBoot image is read-only; changes are written to and read from a "shadow" image for each client, which is recreated with each boot
- Shadow image is on server for OS 9 clients, local for OS X clients
- Clients should use local storage and/or other file shares to access and store changeable data
- Load balancing is implemented by having images on multiple volumes and/or servers, viewed as a single image on clients
- Images are stored on the server in `/Library/NetBoot/`
 - An image directory is automatically created on each disk volume, in order to provide load balancing of images across drive mechanisms
 - If volumes are instead partitions on a single disk device, nothing is gained by this, so unnecessary image locations (defined as share points in Workgroup Manager) should be removed

- `/var/db/bsdnpd_clients` keeps track of clients that have booted from the server in the past
- Also indicates that those clients should boot from the same server in the future, so if you add extra servers for load balancing, delete this file to make clients reattach to servers
- Some useful properties can be set in `/config/NetBootServer`, in the local OD database
 - `afp_users_max`: Maximum number of clients (default 50)
 - `age_time_seconds`: After this time, a client is aged out of consideration for the max
 - `shadow_size_meg`: Maximum size of shadow images
- See the Server Admin Guide for more details
 - Also see <http://homepage.mac.com/johnd/> for John DeTroye's Tip & Tricks

- `init` reads `/etc/ttys`, and starts Login Window
- After boot completes, Login Window requests `system.login.console` right via Security framework, which results in the launch of a Security Agent process to put up the login dialog and perform authentication
 - Login Window used to handle the login dialog itself, before 10.2
- Users can login with full names (i.e., GECOS data) as well as usernames
- Special login names
 - `>console`: Kills Window Server and Login Window, drops to textual console
 - `>exit`: Restarts Window Server and Login Window
 - `>restart`: Reboots computer
 - `>power`: Powers down computer

- After login, Login Window process continues running to handle things like Force Quit requests, dialogs confirming logout, etc.
- Login/Logout Hooks
 - Can provide argument to loginwindow to specify command to execute on every login or logout
 - In `/etc/ttys`, add `-LoginHook /path/to/program` and/or `-LogoutHook /path/to/program` to loginwindow command
 - Command receives username as argument (`$1`)
 - For example, see `updateByHostPrefs` tool at <http://www.occam.com/tools/>

- Disk Volumes
 - Disk Images
 - HFS+
 - File Attributes
 - Bundles
 - Directory Hierarchy
 - Data Backup and Recovery
- 

- By default, all connected disk devices are automatically mounted under `/Volumes/` **by** `autodiskmount`
- Starting with 10.2, can use `/etc/fstab` to statically mount volumes anywhere in the directory hierarchy
 - Can mount by device ID, but under OS X, that can change depending on how devices are connected
 - Preferably mount by disk label or UUID (Universally Unique ID)
- Disk Utility can be used to view volume information, run `fsck`, partition disks, and create software RAID sets
 - `diskutil` is command-line analog
 - **Common arguments:** `list`, `info`, `eject`, `verifyDisk`, `repairDisk`, `verifyPermissions`, `repairPermissions`
 - `disktool` is an older, less capable tool

- Disk images are disk volumes logically encapsulated within single files, usually with `.dmg` extensions
 - Double-clicking a disk image in the Finder mounts the volume
- Disk Copy can create images, mount them, burn them to CDs, etc.
 - Panther: Disk Copy rolled into Disk Utility
- `hdiutil` is command-line analog
 - **Common arguments:** `imageinfo`, `attach`, `detach`, `burn`, `create`
 - `hdid` is the same as `hdiutil attach`

- Two primary bootable filesystem formats on OS X
 - HFS+ (Mac OS Extended File System)
 - Development of Mac Hierarchical File System (HFS)
 - Default local filesystem
 - UFS (UNIX File System)
 - Standard UNIX filesystem, developed from Berkeley Fast File System
 - Can format boot volume, but performance problems and lack of support for multiple forks may create unforeseen problems
 - Panther to improve UFS performance
- Also support for HFS, FAT, ISO 9660, CDDA, UDF
 - Panther includes read-only support for NTFS
 - Implemented by plug-ins in `/System/Library/FileSystems/`

- From UNIX perspective, HFS+ exhibits behaviors that take getting used to
- Multiple forks per file
 - Data, auxiliary resources, and certain metadata are stored in separate filesystem objects
 - Data fork stores main file data (usually)
 - Resource fork used for file-specific icons, multimedia, whatever
 - Attribute fork stores HFS-specific metadata
 - For the most part, extra forks are invisible
 - Resource forks visible with `ls -l filename/..namedfork/rsrc`
 - Some CLI utils in `/Developer/Tools/` can deal with multiple forks

- Multiple forks per file (cont'd.)
 - Forks create huge problems for non-HFS-aware software, including standard UNIX tools
 - `cp` and `mv` only move data forks and leave resource forks orphaned, backups don't get all necessary data, etc.
 - Resource forks are discouraged in OS X
 - Developers should use bundles instead
 - Multi-forked files on UFS are stored in AppleDouble format
 - Content of resource and attribute forks kept in `._filename`

- Case-preserving, but case-insensitive
 - ReadMe is stored with mixed case retained for display, but it can also be accessed as `README`, `Readme`, or `readme`
 - ReadMe and `README` cannot exist in the same directory
 - Apple addresses this for Apache with `mod_hfs_apple`
 - Panther: Option to format HFS+ volumes as case-sensitive
 - Tip: `tcsh` command completion is still case-sensitive unless you set `complete = enhance` in `~/ .tcshrc`
- Path separator is a colon (:), not a slash (/)
 - Kernel converts pathnames on-the-fly, so colons look like slashes
 - Carbon apps convert slashes back to colons

- Application libraries access filesystem objects by numerical file IDs, not pathnames
 - File IDs are unique per disk volume
 - Lookups are faster than by pathname
 - Kind of like inode numbers; in fact, `ls -i` displays file IDs on HFS+
 - File IDs don't change when files are moved around on a disk volume
 - If you know a file's ID, and the the ID of the volume it's on, you can always access it as `/.vol/vol_ID/file_ID`
 - If you know the ID of the directory containing a file, you can access it as `/.vol/vol_ID/dir_ID/filename`

● Aliases

- An alias is a lightweight reference to a file or directory
 - Like a symbolic link, but uses both pathname and file ID
 - Before 10.2, file ID was primary, now pathname takes precedence
- An alias continues to refer to a file even if it's moved (on the same volume) or renamed
- Both aliases and symlinks are useful in different circumstances
 - If the actual pathname is important, or you need to use it from the CLI, use a symlink
- Both aliases and symlinks denoted by small arrows on icons in Finder
 - At CLI, an alias looks like a zero-length file, but with a resource fork
- No way to create symlinks from GUI, or aliases from CLI

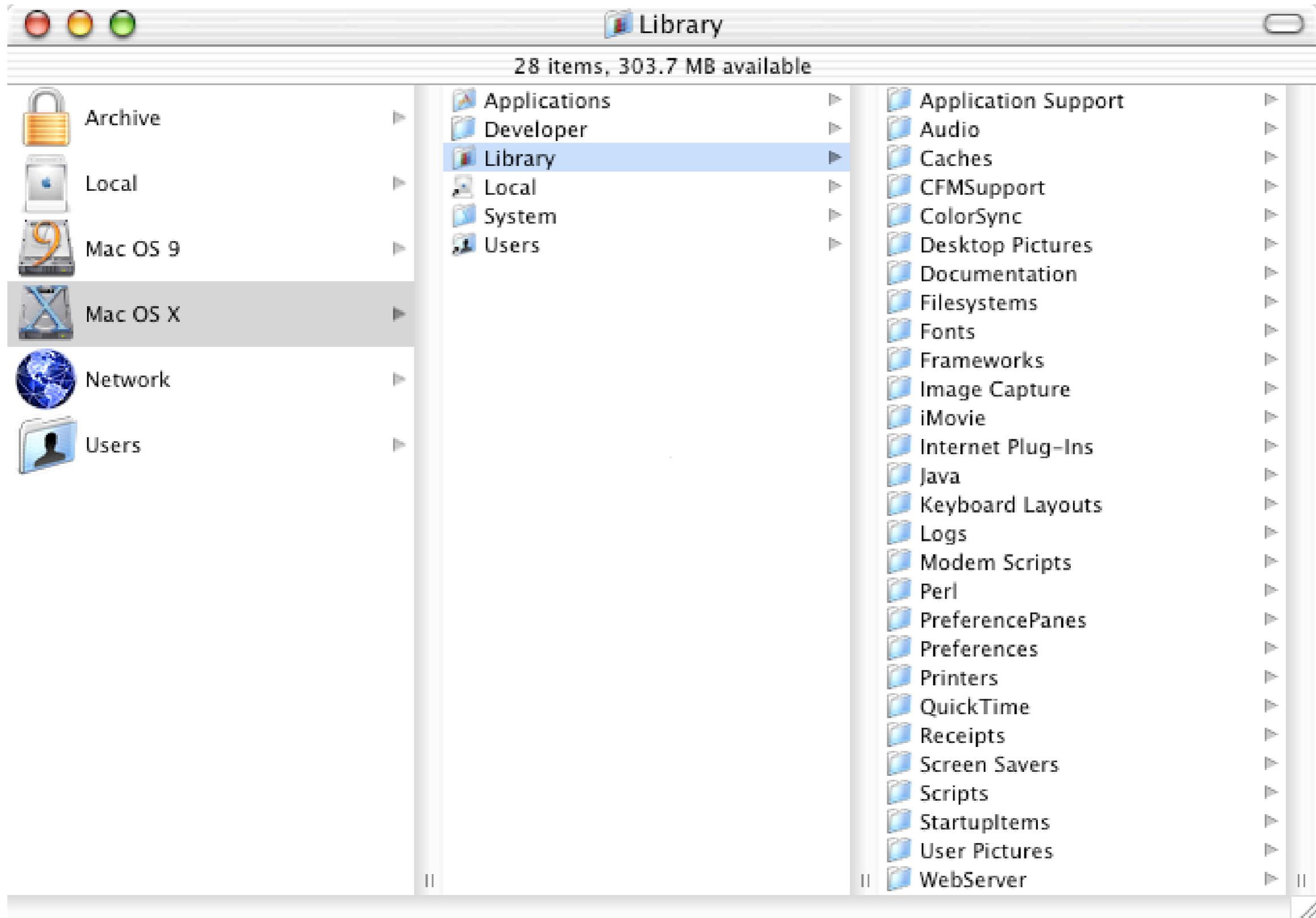


- Hard links
 - On UFS, a hard link is simply another reference to a file's inode
 - With no inodes, HFS+ lacks support for hard links
 - OS X supports hard links for backwards compatibility, but they're implemented in the kernel as symbolic links, faked out to look and act like hard links
 - Slower than real hard links
- Number of links shown for a directory in `ls -l` output counts all items within the directory, including files
- HFS+ lacks support for sparse files; void extents are zero-filled
- HFS+ supports journaling, for faster recovery after crash
- See http://www.mit.edu/people/wsanchez/papers/USENIX_2000/ for more on filesystem design decisions in OS X

- HFS+ supports extensive file metadata
- Typical UNIX metadata: owner, group, permissions, mod date, etc.
 - Files can exist without UNIX metadata (e.g., files created in Classic), in which case they show defaults based on the volume mount point
- BSD flags: immutable, append-only, etc. (`man chflags`)
- Macintosh file attributes: type, creator, creation date, alias, bundle, locked, invisible, etc.
 - Stored in attribute fork (or in `._filename` on UFS)
 - In `/Developer/Tools/`, `SetFile` lists available flags, `GetFileInfo filename` displays type, creator, and flags
- Filename extensions encouraged over type/creator attributes in OS X, for cross-platform compatibility

- Bundles are directories that appear as simple files in the GUI
 - Allows for an item (such as an application) and all its resources (icons, sounds, images, etc.) to be managed as a single file
- Either have bundle bit set, or an appropriate filename extension
- Some types of bundles:
 - `.app`: Application with resources
 - `.framework`: Dynamic shared library with resources
 - `.bundle`: Application-loadable bundle
 - `.kext`: Kernel extension
 - `.rtfd`: RTF document with resources
 - `.mbox`: Mail app mailbox
 - `.prefPane`: System Preferences plug-in

- Parts of the OS X directory hierarchy look pretty familiar when viewed from the command line: `/bin`, `/sbin`, `/dev`, `/usr`, ...
- `/etc`, `/var`, and `/tmp` are symlinks to subdirectories of `/private`
 - NeXTism related to NetBoot
- `/Applications`, `/Library`, `/System`, `/Users`, `/Network`, `/Developer`
- From the Finder (the graphical file manager), things look different
 - Top level contains list of volumes, including boot volume and those mounted under `/Volumes/`
 - UNIXy directories are usually invisible, as are "dot" files, and things listed in `.hidden`
 - Note: "Directories" are referred to as "folders" in the GUI



The View from the Finder

- HFS+ gives standard UNIX backup software fits
 - However, software situation is much better than it was
 - Built-in software includes `ditto -rsrc`, `asr`, and Disk Copy
 - Some Mac-specific third-party software:
 - **PsyncX** (<http://sourceforge.net/projects/psyncx/>)
 - **RsyncX** (<http://www.macoslabs.org/rsyncx/>)
 - **Carbon Copy Cloner** (<http://www.bombich.com/software/>)
 - **Retrospect** (<http://www.dantz.com/en/products/>)
 - **BRU CLI** (<http://www.tolisgroup.com/products/CLI/>)
 - **Tri-BACKUP** (<http://www.tri-edre.com/>)
 - **Impression** (<http://babelcompany.com/impression/>)

- Installation Methods
- Search Domains



- Drag-and-drop
 - Enabled by practice of packaging applications in bundles
 - Usually from a downloaded compressed disk image
- Installer
 - Application works from a `.pkg` bundle
 - Multiple packages included in `.mpkg` bundle
 - Package format developed from NeXT format
 - Unfortunately, `tar` replaced by `pax` in new format, which can lead to all sorts of problems: overwritten symlinks, changed permissions, etc.
 - As a result, Installer packages not used much, except by Apple
 - After installation, empty package moved to `/Library/Receipts/`
 - Command-line tools: `installer`, `lsbom`

- Software Update
 - Downloads and installs packages to update OS and other Apple s/w
 - Do it from command line with `softwareupdate`
 - When run with no arguments, lists uninstalled updates
- Network Install
 - Extension of NetBoot
 - Client boots from installer image, which leads to manual or automatic installation of software packages (including OS, if desired)
- Radmin (<http://rsug.itd.umich.edu/software/radmin/>)
 - Detects differences from a profile, can install files to match profile
 - Cross-platform, but often used to pick up where NetInstall leaves off
 - Mac OS X version includes GUI apps

- DarwinPorts (<http://www.opendarwin.org/projects/darwinports/>)
 - Similar to FreeBSD ports, being worked on by Apple
- Many methods ported from other UNIXen
 - Fink (<http://fink.sourceforge.net/>)
 - Port of Debian `apt-get` system
 - GNU-Darwin (<http://www.gnu-darwin.org/>)
 - Port of FreeBSD ports system
 - NetBSD Packages (<http://www.netbsd.org/Documentation/software/packages.html>)
 - RPM (<http://www.rpm.org/platforms/osx/>)
 - And the venerable tarball

- Cocoa and Carbon APIs specify search algorithms to find applications, frameworks, plug-ins, preferences, fonts, etc.
- Order of search domains
 - **User** (`~/Applications/`, `~/Library/`)
 - Managed by user, accessible only to user
 - **Local** (`/Applications/`, `/Library/`)
 - Managed by admins, accessible to local users
 - **Network** (`/Network/Applications/`, `/Network/Library/`)
 - Managed by admins, accessible to network users (via file sharing)
 - **System** (`/Applications/`, `/System/Library/`)
 - Managed by Apple, accessible to local users

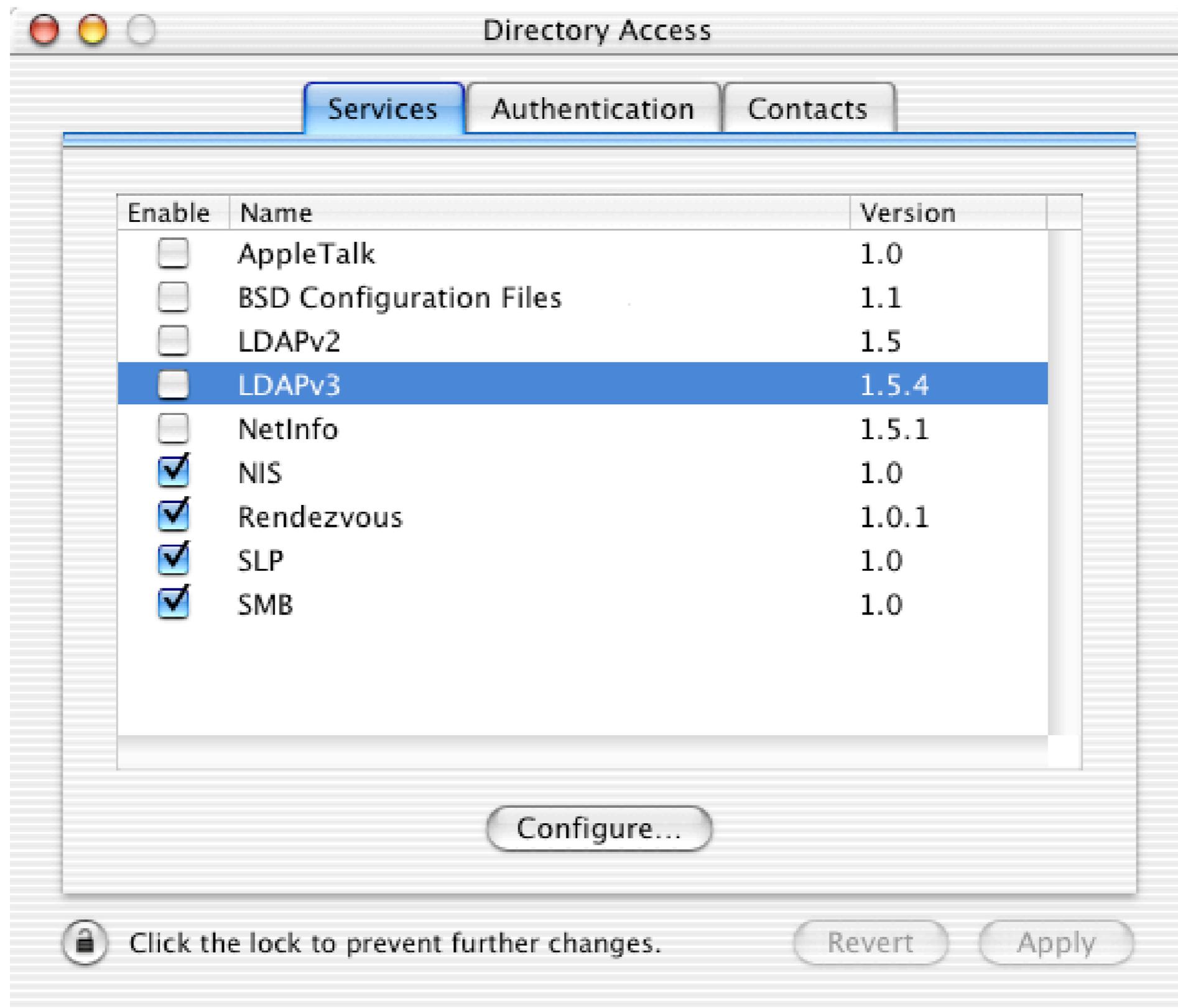
- Note the dual role of `/Applications/`, in both the Local and System domains
- Makes things very confusing, since it implies you have control over that area, when OS updates routinely make changes to it
 - Don't rearrange locations of apps in `/Applications/`
- For ease of administration, I leave `/Applications/` completely to Apple, and create `/Local/Applications/`, typically on a separate partition
 - Following NeXT convention
 - I also symlink `/usr/local` to `/Local`
- Why isn't there a `/System/Applications/?!`

- Introduction
- Open Directory
- Name Services
- Service Discovery

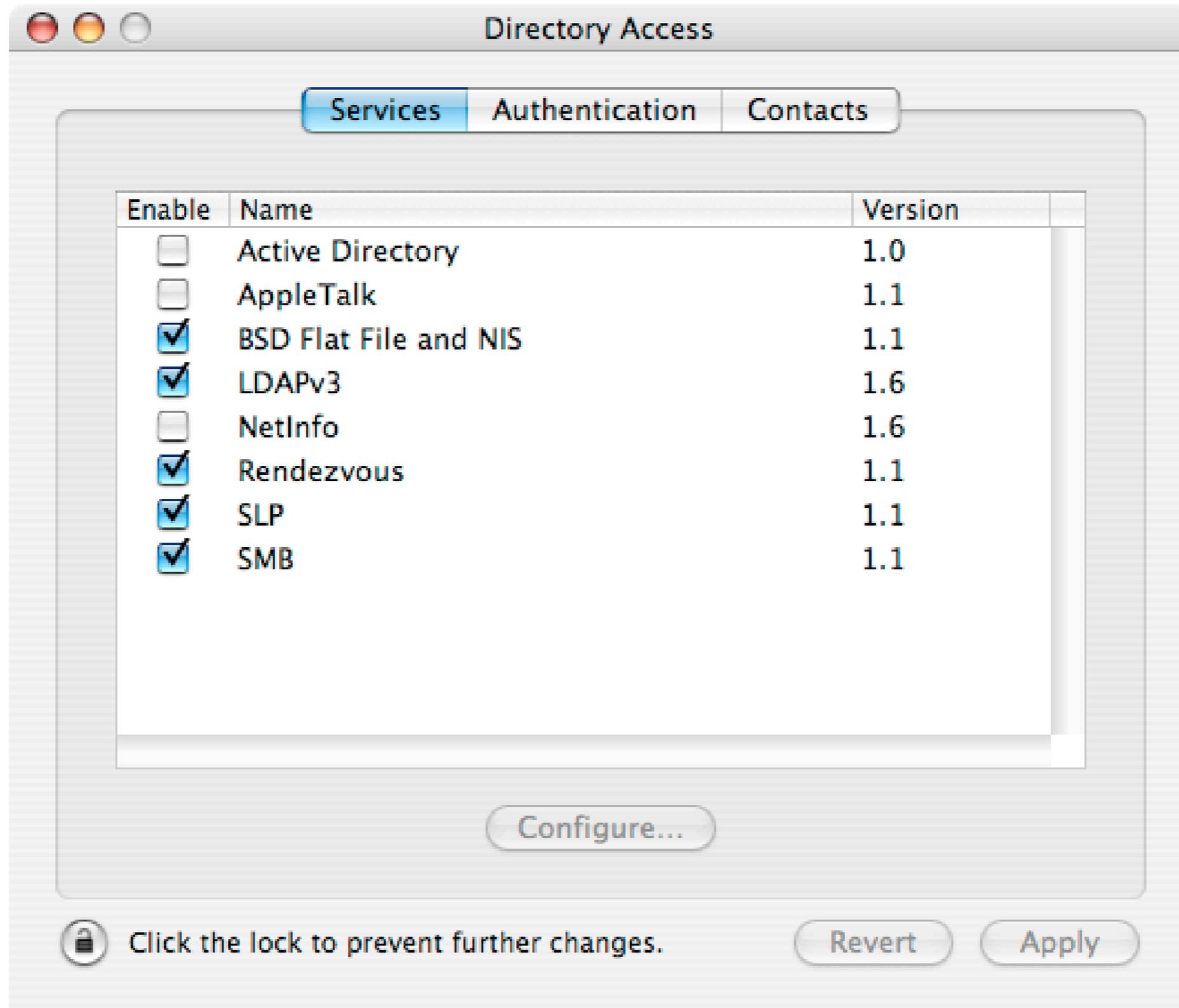


- Directory services are central to Mac OS X administration
- What is a directory service?
 - Loosely, it's a network service providing configuration data to clients
 - Information on users, groups, hosts, printers, etc.
 - Optimized for lots of quick lookups, infrequent changes
 - Examples: LDAP, YP (NIS), Active Directory, DNS, WINS, SLP
- Mac OS X is possibly the most flexible client and provider of directory services around
 - Deep history, owing to NeXT lineage
- The Directory Services framework uses a plug-in architecture to support many different directory service protocols
 - Plug-ins contained in `/System/Library/Frameworks/DirectoryService.framework/Resources/Plugins/`

- The `DirectoryService` daemon handles Directory Services framework requests
- For legacy UNIX programs, unaware of DS, the `getXbyY` system calls (`getpwnam`, `gethostbyaddr`, etc.) are rewritten to proxy lookups through `lookupd`
- The `lookupd` daemon can use the DS framework, or query some services directly (as a legacy of its pre-DS NeXT legacy)
- The search order of data sources consulted by DS is configured in the Directory Access application
- `lookupd`'s search order is configured as described in its man page, either with OD properties in `/locations/lookupd`, or with files in `/etc/lookupd/`
- Directory Access can contact `DirectoryService` on OS X Server systems (on TCP port 625) for remote configuration



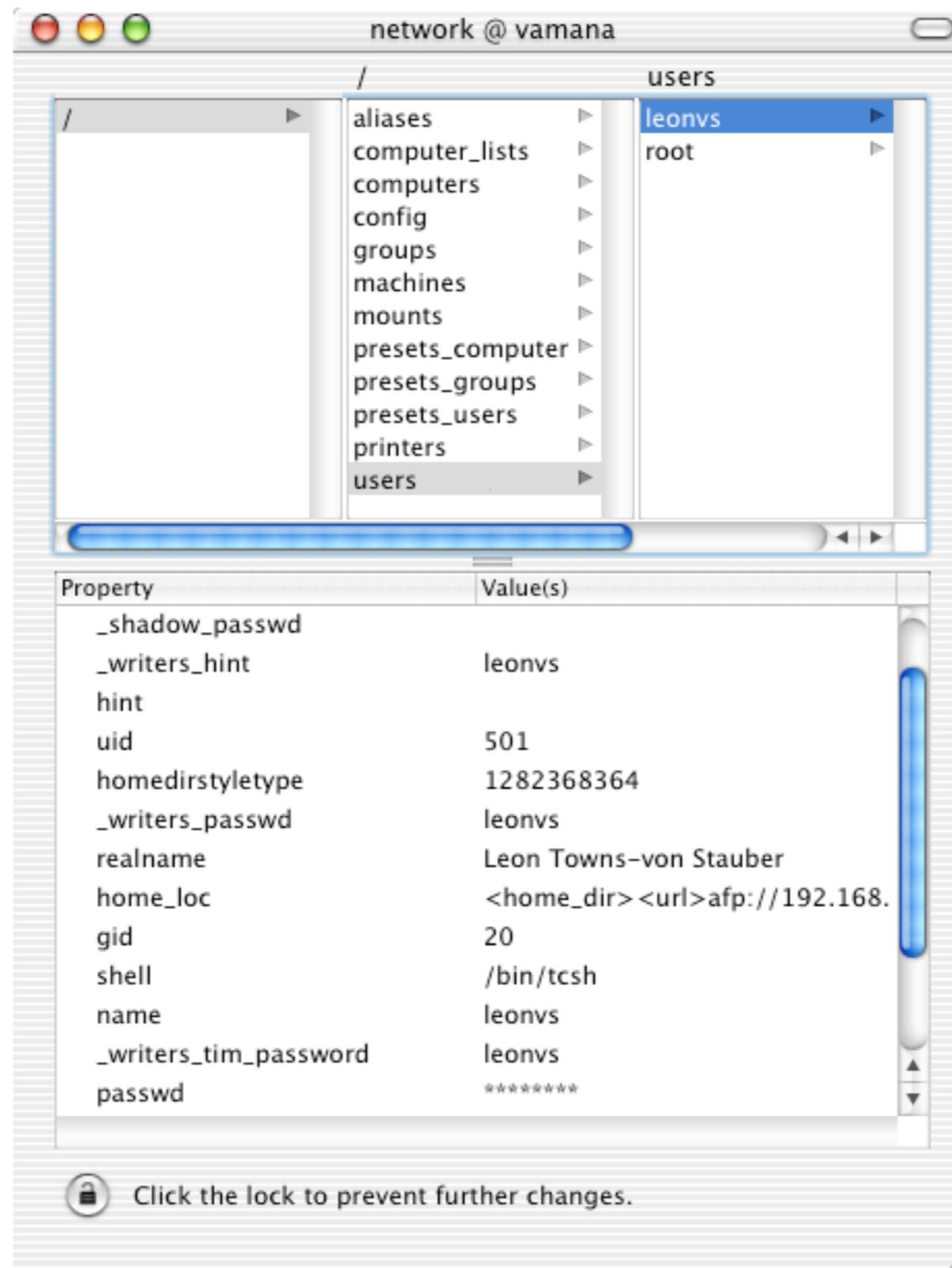
Directory Access



Panther Directory Access

- The Big Surprise
 - Many traditional UNIX flat files in `/etc` (`passwd`, `group`, etc.) aren't used by default (except in single-user mode)
 - This is less true in Jaguar
 - Open Directory is the primary source of configuration data for most Mac OS X machines

- "Open Directory" is a vague umbrella term referring to Apple's implementation of various directory services in Mac OS X
- I'm using the term to refer to the collection of software based on Open Directory domains, accessed by either NetInfo or LDAP, with on-disk data formatted as key/value pairs (e.g., NetInfo DB, Berkeley DB)
- Data in a NetInfo-formatted OD database is organized in a directory hierarchy, analogous to a filesystem directory hierarchy
 - Root is /, subdirectories include `/machines`, `/users/leonvs`, etc.
 - Nodes in the hierarchy have sets of properties, with each property being a key to a set of values
 - Properties include `name`, `uid`, `ip_address`, `passwd`, etc.



Contents of Open Directory database, shown in NetInfo Manager

- Each database is named with a **tag**, corresponding to the directory in which the database is stored (`/var/db/netinfo/tag.nidb/`)
- Panther: Open Directory databases may now be in Berkeley DB format (just like default OpenLDAP)
- Each database contains information for a single Open Directory **domain**
- Domains are organized in hierarchies of **parent** and **child** domains, of arbitrary depth
- Root domain is `/`, subdomains might be `/department`, `/department/hostname`, etc.
 - Domain names don't necessarily match database tags
- Current domain is `.`, parent domain is `..`

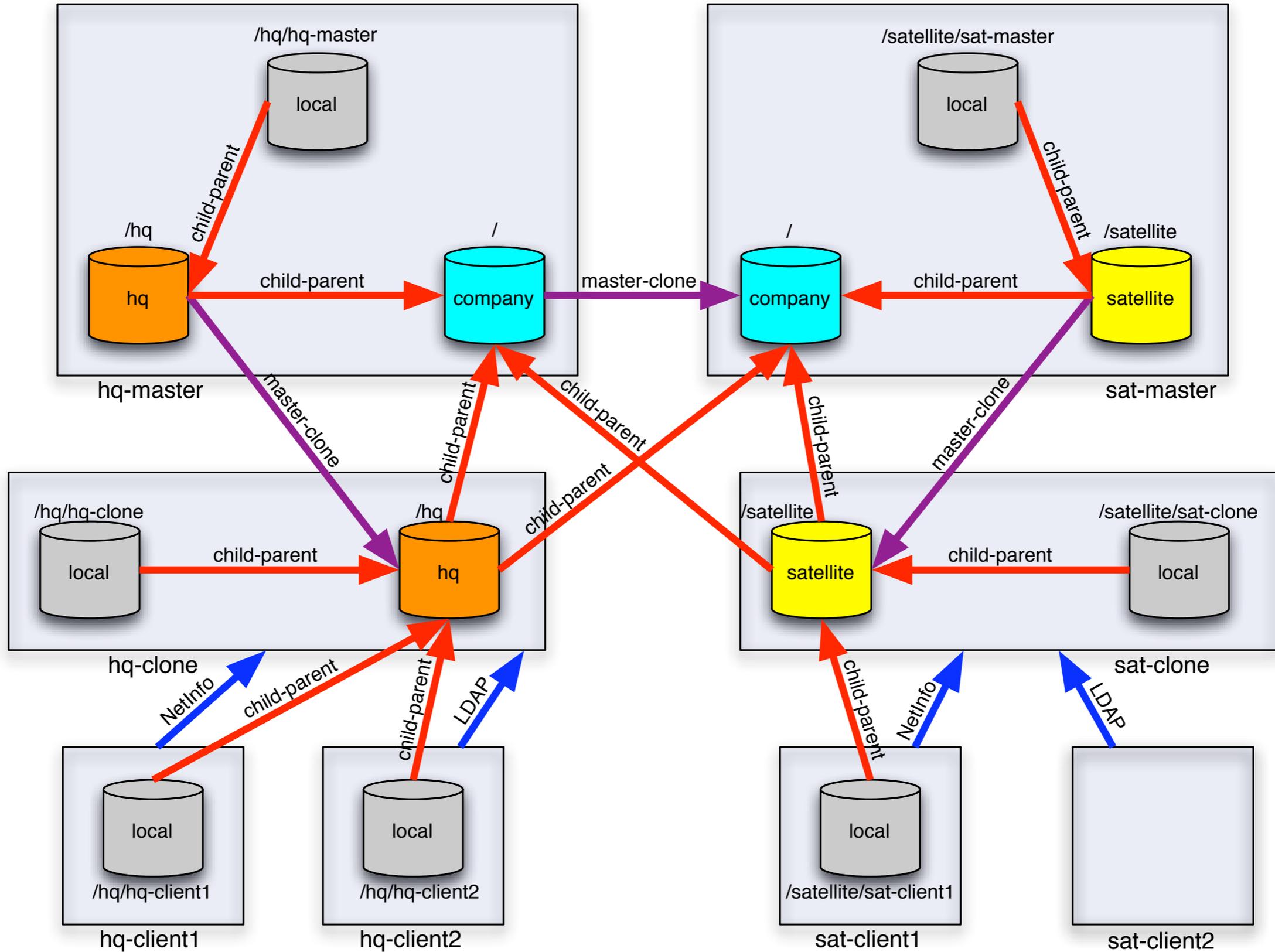
- Every OD member host has a domain with a tag of `local`
- A client may access information in its own `local` domain, and in all its parent domains up to the root, permitting flexible sharing of information among all systems in the hierarchy
- Common Open Directory hierarchy topologies
 - Single-tier: local domains only, no network sharing
 - Two-tier: root domain contains all network-wide information
 - Three-tier: middle-tier domains by department, location, or some other organization unit
 - Four-tier and deeper hierarchies are possible as well
- In two-tier and deeper hierarchies, each client "belongs" to a domain
 - That is, its `local` domain is the child of a designated parent domain

- Domain relationships determined by `serves` properties in machine records (i.e., entries for hosts under `/machines` in database)
- A `serves` property specifies the domain(s) served by the host, and the tag of the database serving the domain (as `domain/tag`)
 - Domain is usually relative (`..` or `.`)
- Each domain is served by one or more systems: one **master** server (hosting the read/write master database for the domain) and possibly several **clone** servers (hosting read-only copies of the database)
- Clones provide fault tolerance, and local service across WAN links from master
- Every OD host is master of its own `local` domain

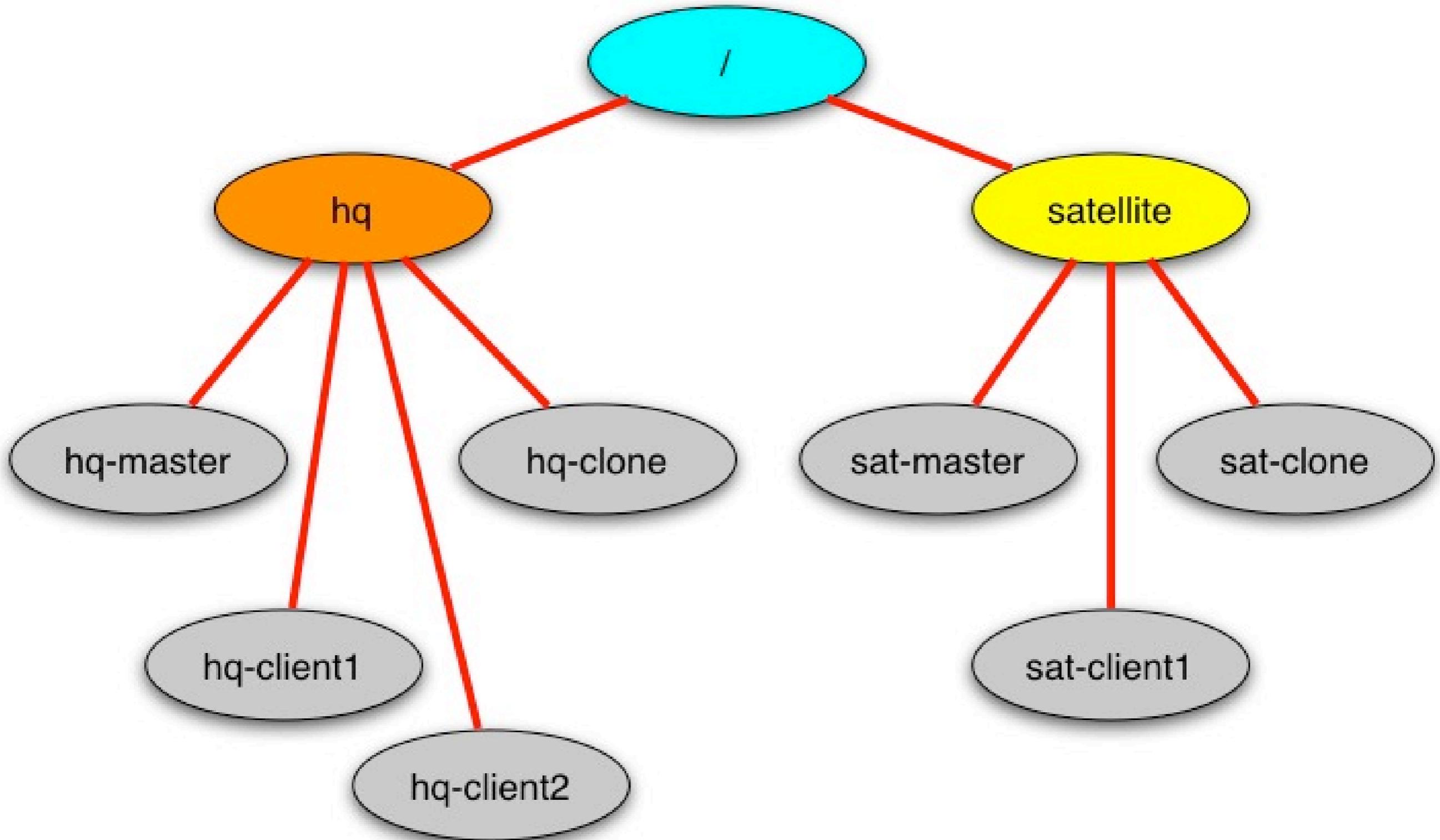
- Replication
 - Master notifies clones of changes, which then pull updates from master
 - Clones find master copies with the `master` property (in the database root directory), which specifies a hostname and tag
 - Synchronization usually occurs within seconds of a change
 - Checksum is used to guarantee uncorrupted transfer, before temporary database copy moved into production
 - Note: This describes NetInfo replication; the mechanism has probably changed in Panther, but I haven't yet studied it

- Domains may be accessed by clients using either NetInfo or LDAP
 - NetInfo: NeXT legacy protocol, deprecated in future revisions of OS X
 - LDAPv3 (Lightweight Directory Access Protocol): Standard directory access protocol, widely deployed in recent years
 - OpenLDAP with a custom back-end developed by Luke Howard of PADL Software (<http://www.padl.com/>)
- When part of a hierarchy, clients bind to servers in a number of ways, specific to the access protocol (and configured in Directory Access)
 - NetInfo: server hostname and DB tag explicitly configured on each host or obtained through DHCP, or client can broadcast to find a server
 - LDAP: simple LDAP bind, using server info configured on client or obtained from DHCP

- The diagrams on the following two slides illustrate a typical three-tier OD hierarchy
 - `hq-master` hosts master databases (tagged `company` and `hq`) for the `/company` and `/hq` domains
 - `sat-master` hosts master DB (tagged `satellite`) for `/satellite` domain, and cloned DB (tagged `company`) for `/company` domain
 - `hq-clone` hosts cloned DB (tagged `hq`) for `/hq` domain
 - `sat-clone` hosts cloned DB (tagged `satellite`) for `/satellite` domain
 - Each host (except for non-OS X host lower right) hosts own `local` DB, bound into domain hierarchy under `/hq` or `/satellite`
 - `hq-client1` and `sat-client1` talk to local servers using NetInfo
 - `hq-client2` and `sat-client2` talk to local servers using LDAP



Sample three-tier Open Directory hierarchy



Sample three-tier Open Directory hierarchy: logical domain structure

- Daemons

- `rpcbind`

- Standard RPC portmapper

- Used by clients to find NetInfo binder (`nibindd`)

- `nibindd`

- Parses `/var/db/netinfo/`, spawns `netinfod` for each database

- NI clients bind using RPC, and are then directed to `netinfod` port

- `netinfod`

- One per domain served by host

- Typically bind to arbitrary ports between 600 and 1023 (inclusive)

- `slapd`

- OpenLDAP server

- Tools
 - NetInfo Manager permits direct modification of database contents
 - Command-line tools
 - `nicl`
 - Full access to DB contents
 - In interactive mode , navigate DB like filesystem (`cd`, `ls`, `cat`, etc.)
 - Panther adds `dscl`
 - `nidump`, `noload`
 - Dump (or upload) domain contents in UNIX flat-file format
 - `nireport`, `nifind`, `nigrep`
 - `nidomain`
 - Interface with `nibindd` to list, create, destroy, and clone domains

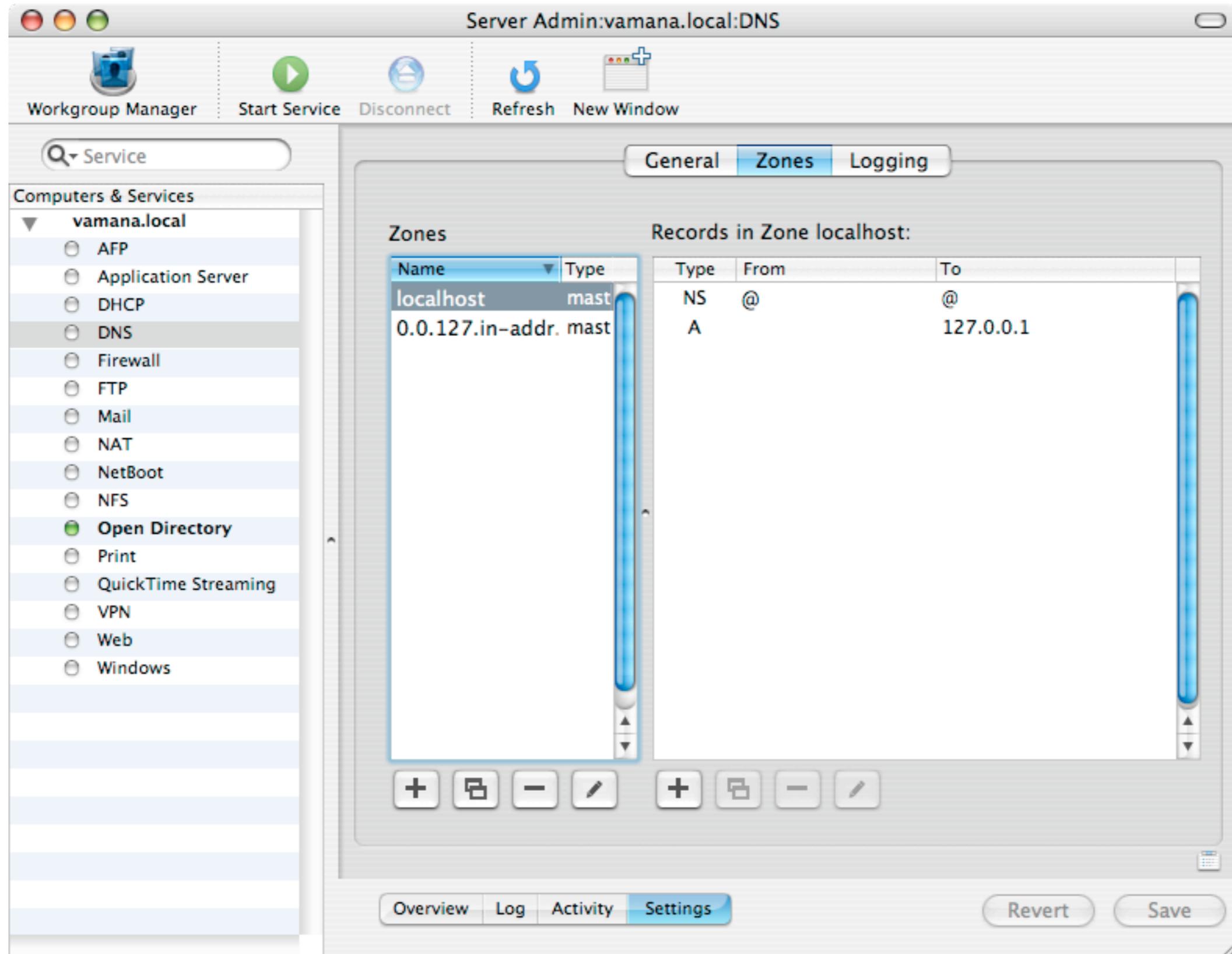
- Additional resources

- http://developer.apple.com/techpubs/macosx/Networking/Open_Directory/

- <http://www.padl.com/Articles/AdvancedOpenDirectoryConf.html>

- For the purposes of this talk, a "name service" is something that provides name and address resolution for network elements
- Most name service lookups go through `lookupd`
- Mac OS X supports offering various name services: DNS (including mDNS), WINS, AppleTalk, NIS
- WINS client and server support via Samba's `nmbd`
 - Mac OS X Server offers basic control under Server Settings->File & Print->Windows->Neighborhood
- AppleTalk service enabled from Network Preferences
- Standard NIS tools included; no GUI to set up service

- DNS resolver is configured in Network Preferences
 - `resolv.conf` created dynamically in `/var/run/`, symlinked from `/etc/resolv.conf`
- DNS server is BIND
 - Only built-in GUI is in Mac OS X Server, to start/stop it (Server Settings) and view basic statistics (Server Status)
 - Panther: More extensive GUI allows editing of zones and more
 - Third-party GUIs
 - QuickDNS and others (<http://www.menandmice.com/>)
 - Bindery (<http://www.afp548.com/software/Bindery/>)
 - iTools (<http://www.tenon.com/products/itools-osx/>)



Panther DNS management

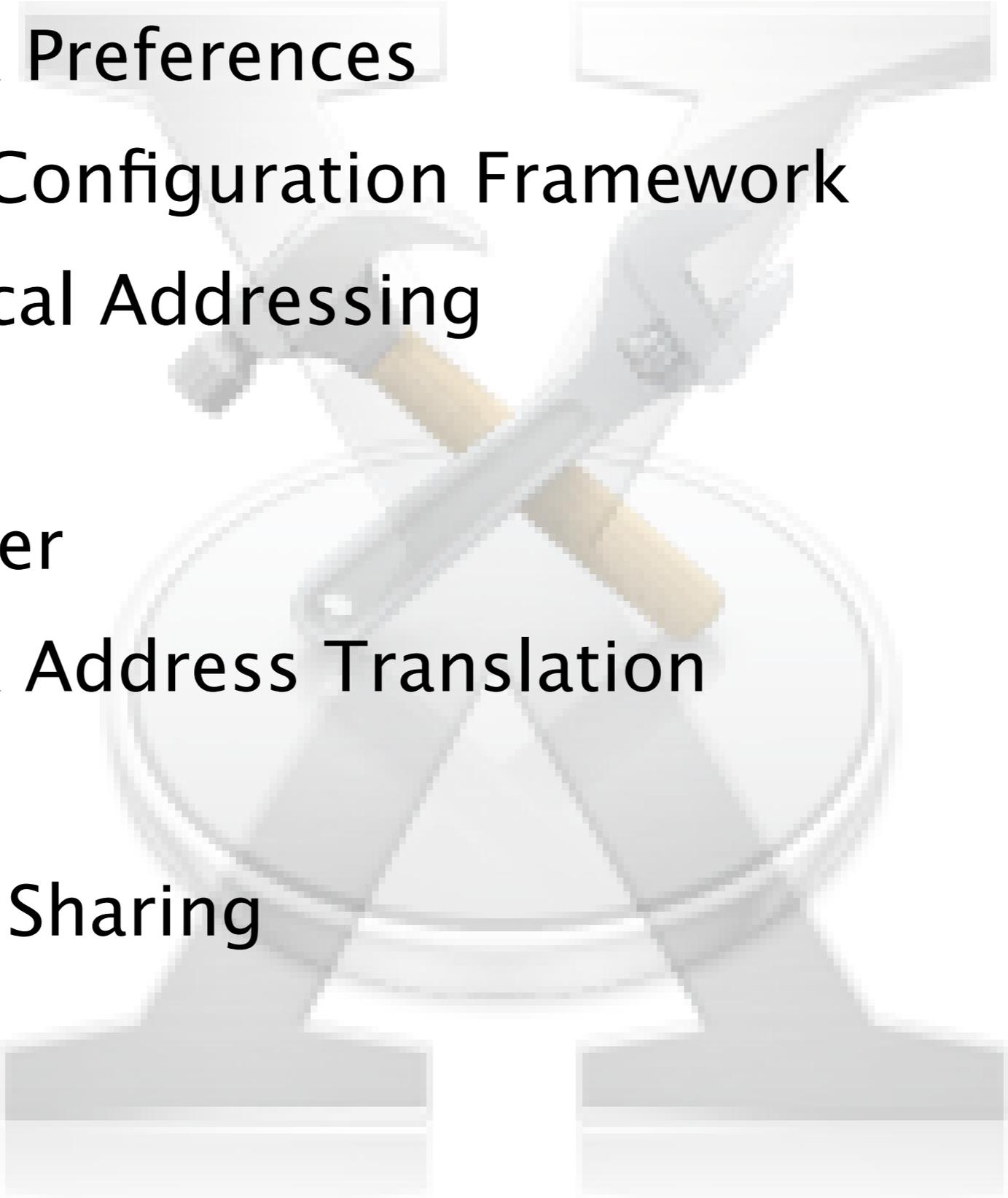
- Multicast DNS (mDNS)
 - One of three technologies under the umbrella of Zeroconf (what Apple calls Rendezvous)
 - `http://www.multicastdns.org/`
 - Distributed, instead of centralized, naming authority
 - Every host responsible for its own name to address mapping
 - Rendezvous hostname set in Sharing Preferences
 - mDNS names are in `.local` domain
 - The resolver knows to multicast for `.local` names
 - Configured with files in `/etc/resolver/`
 - Each host runs its own mini-DNS server, `mDNSResponder`
 - Listens on UDP port 5353 (and, on 10.2, 53 for legacy clients)

- Mac OS X supports several automated service discovery protocols
 - Used most prominently in Finder's Connect to Server...
- AppleTalk
 - Legacy of old Mac OS, now deprecated
 - CLI tools: `appletalk`, `atlookup`, `appleping`
- Service Location Protocol (SLP)
 - First attempt to replace AppleTalk with a TCP/IP-based alternative
 - RFC 2608
 - Services are identified with URLs
 - `nfs://`, `afp://`, `smb://`, `http://`, `ldap://`, **etc.**
 - Services are typically discovered and registered using multicast
 - SLP agents listen on TCP port 427

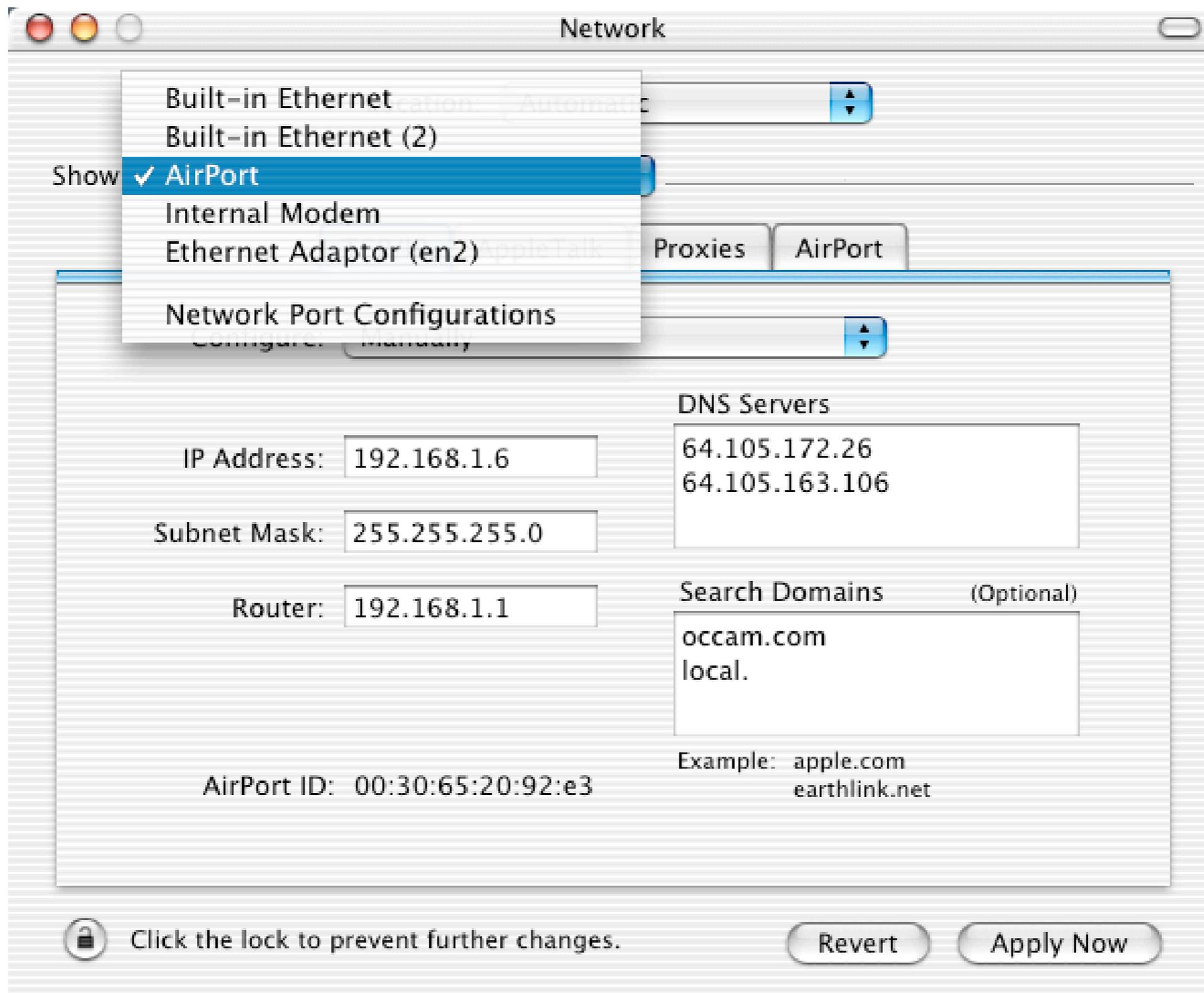
- SLP (cont'd.)
 - Services can be registered within named scopes, to limit visibility
 - SLP Service Agents (SAs) operate peer-to-peer, each responsible for registering its own services
 - Directory Agents (DAs) can centralize service listings to lessen network traffic
 - `slpd` is SLP daemon, can act as SA or DA
 - Configured by `/etc/slpsa.conf`, which is normally empty
 - You can find config parameters in Darwin source for `slpd`
 - Registered services listed in `/var/slp.regfile`
 - Can use `slp_reg` to manually register services
 - Mac OS X Server includes GUI tools to manage SLP DA service
 - Not in Panther anymore

- DNS Service Discovery (DNS-SD)
 - Another of three proposed standards comprising Zeroconf (Rendezvous), and the designated replacement for SLP
 - <http://www.dns-sd.org/>
 - Designed to work with Multicast DNS (but not dependent on it)
 - Uses a few different resource record types:
 - PTR: query for service type (like `_http._tcp.local.`), receive list of associated service instance names
 - SRV: query for service name, receive hostname and port of server
 - TXT: additional info as key/value pairs (e.g., queue name)
 - **iRoster** (<http://www.toxicsoftware.com/software/shareware/iRoster/>) provides GUI for generalized DNS-SD browsing

- Dynamic Host Configuration Protocol (DHCP)
 - DHCP can be used to configure service locations via options
 - NetInfo, LDAP, DNS, mail, etc.
- Server Message Block (SMB)
 - SMB clients do their own service discovery, browsing for file and print servers
 - Samba supports browse clients
- Common UNIX Printing System (CUPS)
 - CUPS supports both SLP and a CUPS-specific protocol for clients to discover CUPS print servers
 - **Controlled by** `Browsing` and `BrowseProtocols` directives in `/etc/cupsd.conf`

- Network Preferences
 - System Configuration Framework
 - Link-Local Addressing
 - DHCP
 - IP Failover
 - Network Address Translation
 - AirPort
 - Internet Sharing
- 

- Most basic network configuration is performed from System Preferences
->Network (or Apple Menu->Location->Network Preferences...)
- Network interfaces ("ports", as labeled in System Preferences) can be tried in a specified order, on the fly
 - For instance: If wireline Ethernet is unavailable, try wireless interface
- An interface can have multiple addresses
 - Simply copy the interface in System Preferences, and assign a different IP address to the duplicate
 - Mac OS X Server also has an alternate method: Configure extra IP addresses in `/etc/IPAliases.conf`, and set `IPALIASES=-YES-` in `/etc/hostconfig`
 - This mechanism is no longer needed
- Can set up "locations", with different network configurations



Network Preferences

- The settings from Network Preferences are saved in `/var/db/SystemConfiguration/preferences.xml`
- **Panther:** `/Library/Preferences/SystemConfiguration/preferences.plist`
- The System Configuration framework is responsible for monitoring changes in network status, providing notification to applications
- Applications communicate with through the API with the System Configuration Server, `configd` (started early, in `/etc/rc`)

- `scutil` offers interactivity with `configd` from the command line
- `ipconfig` offers some rudimentary network management, especially related to DHCP, that isn't too useful
 - `ipconfig getifaddr interface` lists interface's primary IP address
- `scselect` lets you view and change network location from the CLI
 - Bug: Changing location removes read permission to `preferences.xml` for all but `root`, which makes config unviewable in Network Prefs

- Link-local addressing is another leg of the Zeroconf (Rendezvous) stool
- <http://www.zeroconf.org/>
- If a system isn't configured with an IP address, it automatically acquires a random one in the 169.254.x.x range
 - Negotiated with other devices on the local network to prevent conflicts
- Combined with multicast DNS, you could potentially get by without any network configuration on client systems

- The Mac OS X DHCP server is `bootpd`
 - Typically managed by `xinetd`
 - Set `disable = no` in `/etc/xinetd.d./bootps` (or run `service bootps start`)
 - Configured in Open Directory, under `/config/dhcp/`
- You can assign specific IP addresses to DHCP clients
 - Create a machine record for the client (under `/machines/` in OD), setting the `ip_address` property, and set the `en_address` property to the client's MAC address
- Mac OS X Server provides a management GUI in Server Settings
 - Server Status shows clients with current leases
- AirPort Base Station also offers DHCP, configured in AirPort Admin Utility

- IP failover lets a standby Mac OS X Server system take over for another
- The active and standby servers must have access to two different subnets
 - A "public" network, over which the active machine offers services
 - A "private" network, used to verify the active server's failure
- The active server sends periodic broadcasts on both subnets
 - If broadcasts on **both** subnets stop, the standby takes on the active's public IP address, and runs a set of scripts
 - When broadcasts resume, the standby relinquishes the IP address
- `heartbeated` sends broadcasts, to UDP port 1694
- `failoverd` monitors active host's broadcasts
 - Calls `NotifyFailover` and `ProcessFailover` when status changes

- Procedure

- 1) On the active server, add to `/etc/hostconfig` (substituting appropriate broadcast addresses for the public and private subnets):

- `FAILOVER_BCAST_IPS="192.168.1.255 10.0.0.255"`

- 2) On the standby server, add to `/etc/hostconfig`:

- `FAILOVER_PEER_IP="10.0.0.1"` (active's private IP address)

- `FAILOVER_PEER_IP_PAIRS="en0:192.168.1.6"` (standby's public interface, and active's public IP address)

- `FAILOVER_EMAIL_RECIPIENT="admin@occam.com"` (to receive notification of status changes)

- 3) On the standby server, set up failover scripts

- 4) On the active server, then on the standby server, `SystemStarter`
`start IPFailover`

- Failover scripts
 - Scripts executed upon various state changes
 - Start or stop services, cleanup, extra notification, insert delays, etc.
 - In `/Library/IPFailover/active_server_public_IP/`
 - Test: Run first; if returns non-zero, standby only sends email notification, but doesn't acquire active's public IP address
 - If you only want notification, not failover, `cp /usr/bin/false /Library/IPFailover/IP_address/Test`
 - PreAcq*: Before acquiring IP address
 - PostAcq*: After acquiring IP address
 - PreRel*: Before relinquishing IP address to active
 - PostRel*: After relinquishing IP address

- Network address translation (NAT) performed by `natd`
 - See also `ipnat` man page
- Would probably need to create a startup item if you plan to use it
- No built-in GUI
 - Panther Server now has a GUI
- Third-party GUIs
 - **IPNetShareX (formerly gNAT)** (http://www.sustworks.com/site/prod_gnat_overview.html)
 - **BrickHouse** (http://personalpages.tds.net/~brian_hill/brickhouse.html)
 - **sunShield** (http://homepage.mac.com/opalliere/shield_us.html)
 - **Firewalk X** (<http://www.pliris-soft.com/products/firewalkx/>)

IPNetShareX

Subnet Mask: 255 . 255 . 255 . 0

Router Address: 192 . 168 . 0 . 1

NAT Settings

- Preserve Ports (improves compatibility)
- Use Sockets (improves compatibility)
- Create Aliasing Log (in /var/log/alias.log)
- Deny Incoming Requests (more secure)
- Create Denial Logs (via syslog)

External Network Interface: Built-in Ethernet (en0)

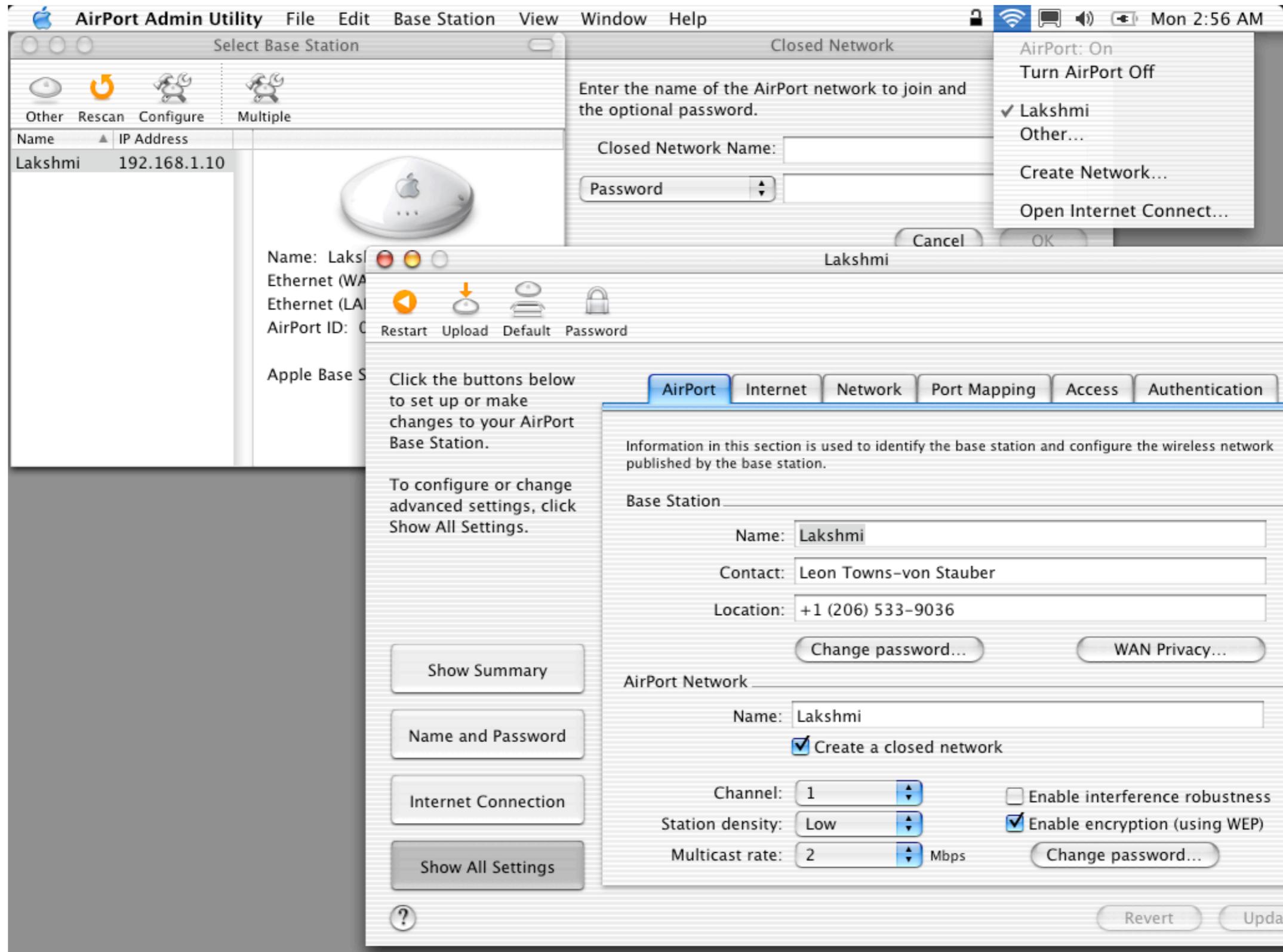
Internal Network Interface: Built-in Ethernet (en0)

- Dynamic IP Binding (for PPP, DHCP and PPPoE)
- Local Caching Name Server (for PPPoE)
- Load on Startup

Click to allow changes Reset Start Internet Sharing

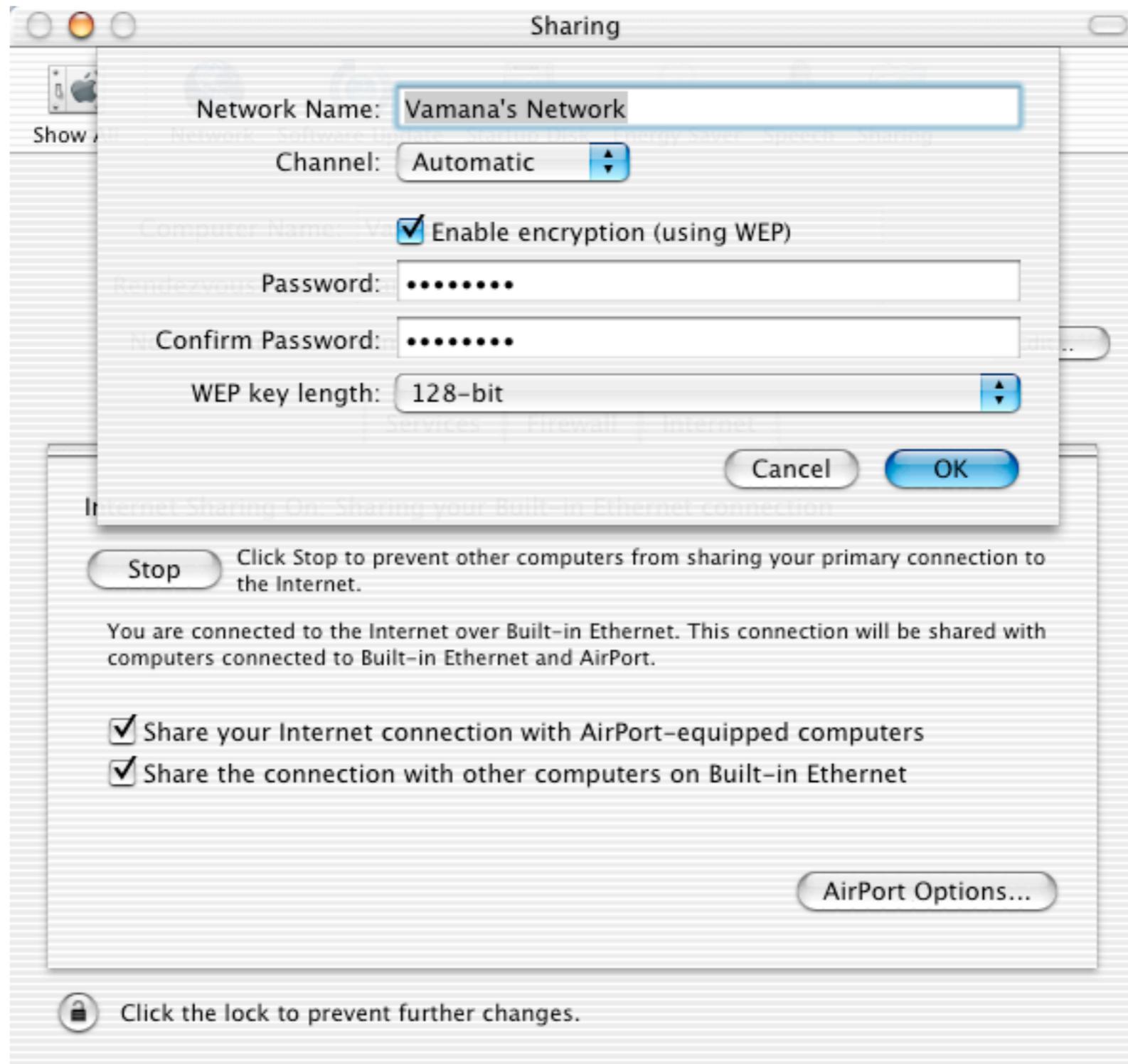
IPNetShareX

- AirPort is Apple's name for 802.11b (Wi-Fi) wireless networking
 - AirPort Extreme is 802.11g
- AirPort Base Station configured using AirPort Admin Utility
 - Supports DHCP service, port mapping, authentication to RADIUS, etc.
 - Uses Rendezvous to discover base stations

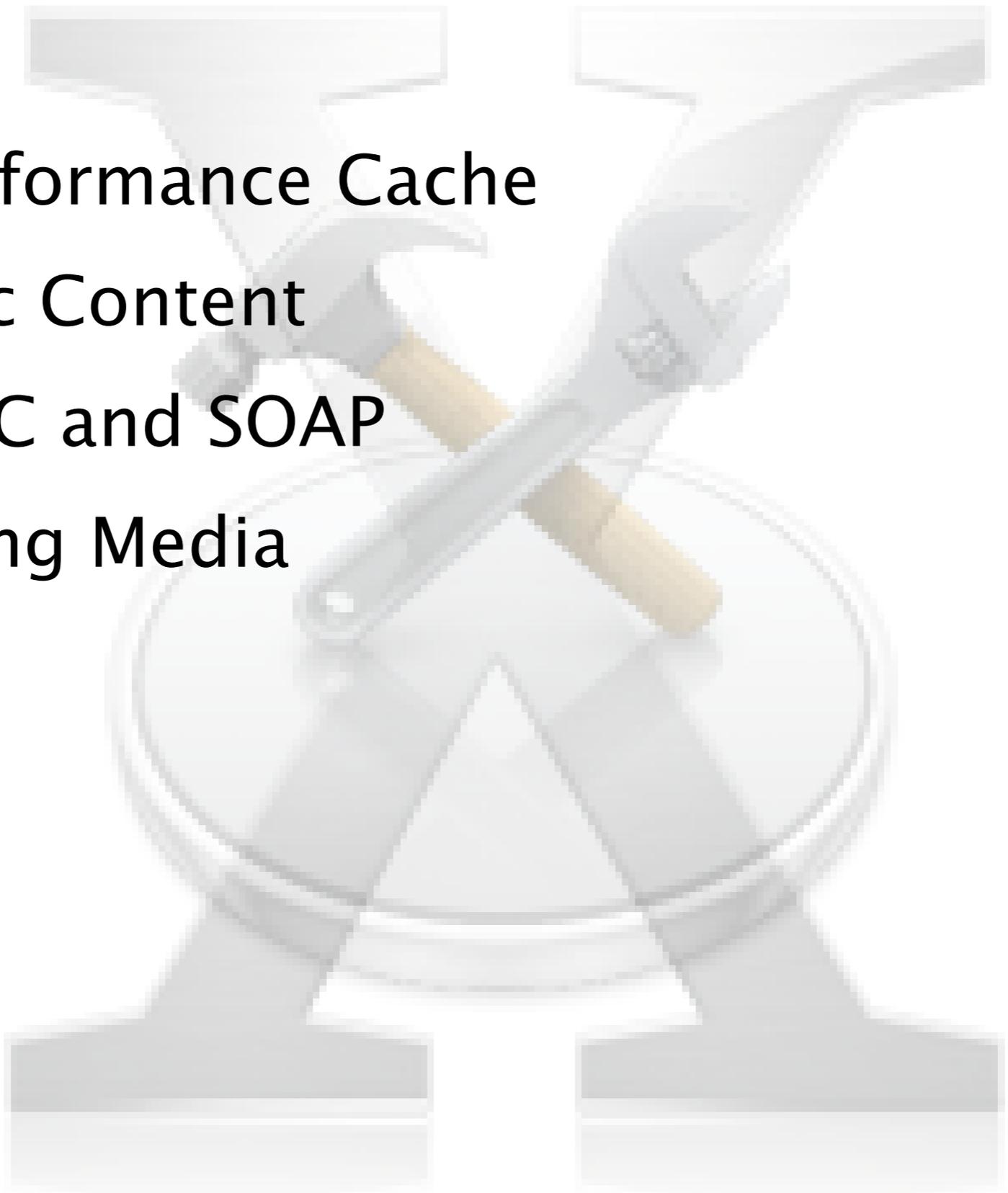


AirPort Admin Utility and AirPort menubar item

- Enabled in Sharing Preferences
- Sets up IP addresses on alternate subnets for each shared interface
 - View with `ifconfig`
- Establishes DHCP service by executing `bootpd` (directly, not via `xinetd`)
 - Look at `/config/dhcp/subnets/` in Open Directory
- Enables IP forwarding
 - `sysctl -w net.inet.ip.forwarding=1`
- Sets up NAT
 - `natd -alias_address IP_address -interface interface -use_sockets -same_ports -unregistered_only -dynamic -clamp_mss`
- Diverts traffic
 - `ipfw add divert natd ip from any to any via interface`

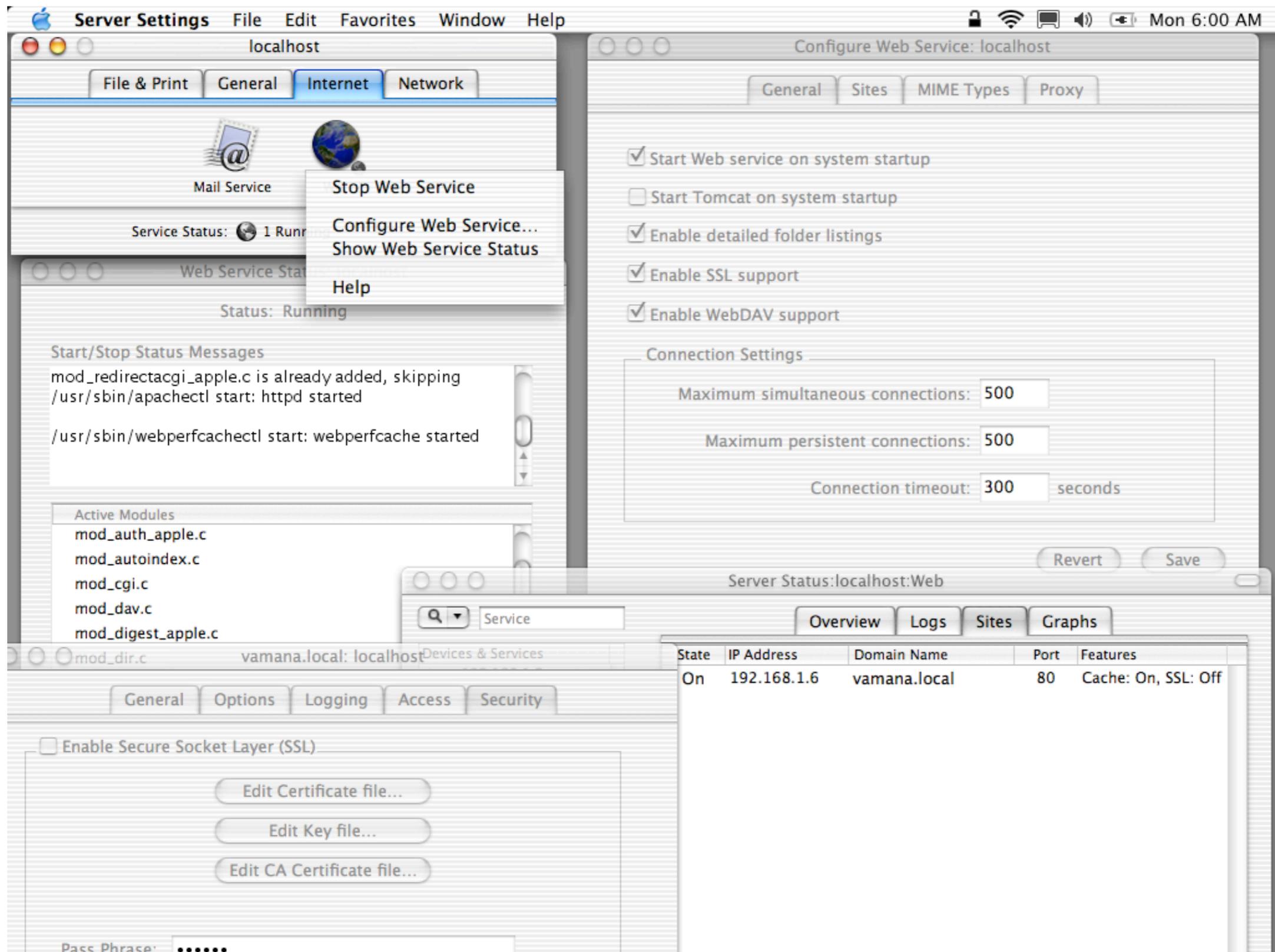


Internet Sharing Preferences

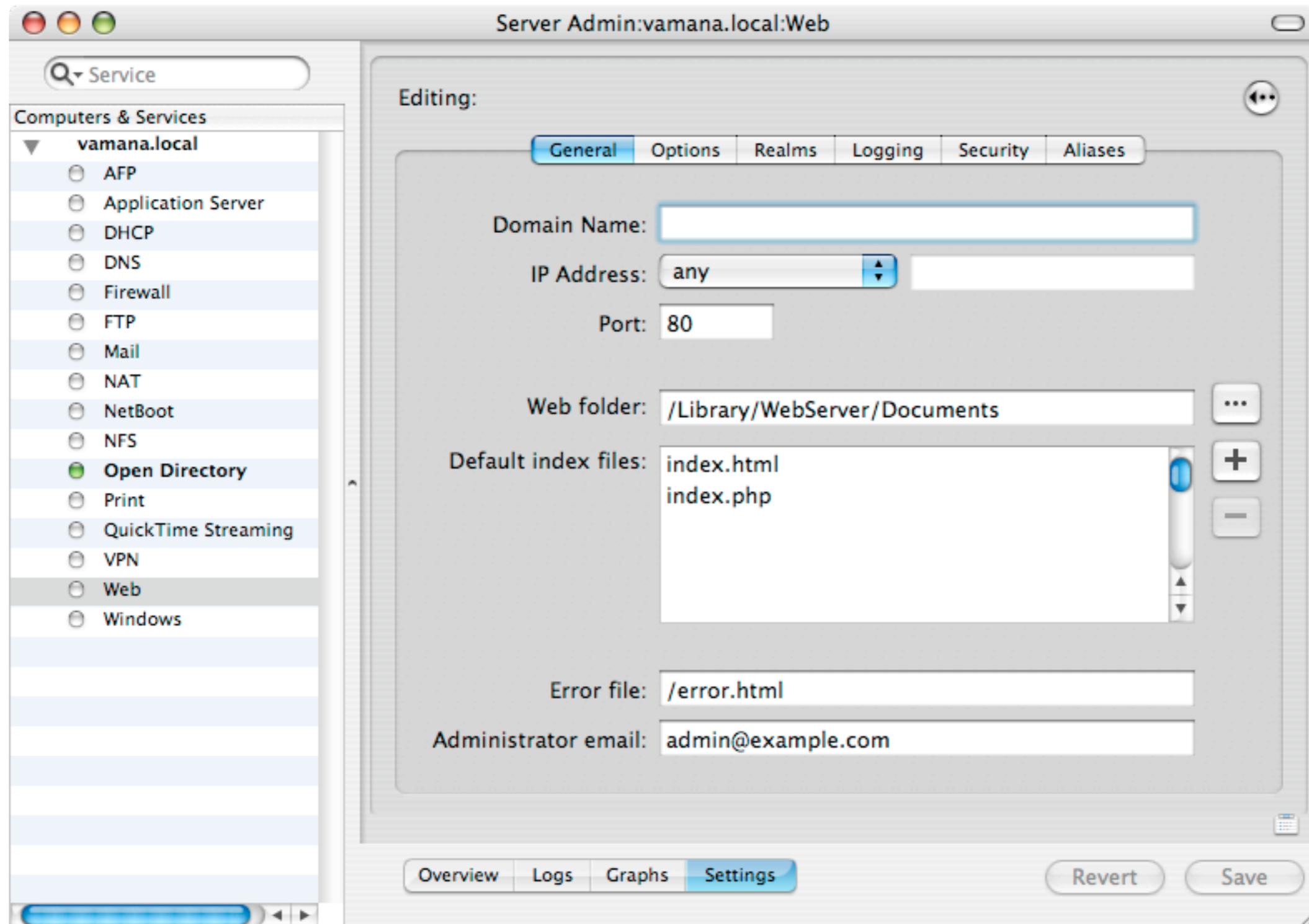
- Apache
 - Web Performance Cache
 - Dynamic Content
 - XML-RPC and SOAP
 - Streaming Media
- 

- Apache 1.3 bundled with OS X
 - Apache 2 on OS X Server, in `/opt/apache2/`
 - The latest iTools (<http://www.tenon.com/products/itools-osx/>) also includes Apache 2 (along with a lot of other server software and management utilities)
- File locations
 - Config files in `/etc/httpd/`
 - Document root and CGI directory in `/Library/WebServer/`
 - Log files in `/var/log/httpd/`
- Mac OS X features one-button activation, in Sharing Preferences
 - Enabling Personal Web Sharing sets `WEBSERVER=-YES-` in `/etc/hostconfig`
 - Documents also served from `~username/Sites/`

- Mac OS X Server replaces Personal Web Sharing with a fuller-featured front end in Server Settings and Server Status
- Changes in Server Settings are generally reflected in `/etc/httpd/httpd_macosxserver.conf`, both as standard Apache directives and as pseudo-directives specific to Server Settings
 - For example, when `#AutoStartServer` pseudo-directive is set to `On`, `serversettingsd` starts up Apache
- `servermgrd`, used by Server Status, is an instance of Apache
 - Config files in `/etc/servermgrd/`
 - Docs and CGI scripts in `/usr/share/servermgrd/`
 - Logs in `/var/log/servermgrd/`



Mac OS X Server Apache management tools



Panther Server Apache management tools

- Mac OS X includes some Apple-specific Apache modules
 - `mod_auth_apple.so`, `mod_digest_apple.so` (Server only): Enables Apache to use Directory Services for basic and digest authentication
 - `mod_hfs_apple.so`: Causes filenames on HFS+ volumes to be treated as case-sensitive, preventing bypasses of pathname-based security controls
 - `mod_macbinary_apple.so` (Server only): Automatically packages files in MacBinary format when `.bin` is appended to the URL
 - `mod_redirectcgi_apple.so` (Server only): Enables Apple CGIs
 - `mod_rendezvous_apple.so`: Causes Apache to broadcast its service using Rendezvous (DNS-SD), for automatic discovery by clients
 - `mod_sherlock_apple.so` (Server only): Provides searching through a site's documents at `http://site/.sherlock/`

- Mac OS X Server includes a web performance cache server
- Listens on TCP port 80, serving cached static documents, and relaying requests for dynamic or uncached content to Apache (which gets moved aside to port 16080)
- Daemon is `webperfcache`
 - Config files in `/etc/webperfcache/`
 - Logs in `/var/log/webperfcache/`
 - Controlled by `webperfcachectl` shell script
 - Arguments: `start, stop, restart, status, showlog`

- Can generate dynamic content with scripting, using `mod_perl` or PHP
 - Could also write CGI scripts in Python, Ruby, Tcl, etc.
- Mac OS X Server includes additional web development environments
 - Tomcat
 - Apache Software foundation implementation of Java servlets and Java Server Pages (JSP)
 - JBoss
 - Included in Panther Server to support J2EE
 - WebObjects
 - Deployment license included for Apple's web application server
 - Apple CGI (ACGI)

- Apple CGI (ACGI)
 - Write CGIs using AppleScript (or other Apple Event-capable scripting language, like...?)
 - ACGI daemon, `acgid`, listens for ACGI requests redirected by Apache
 - Enabled in `/etc/httpd/httpd.conf`
 - Configured by `/etc/acgid/acgid.conf`
 - By default, listens on TCP 9008
 - Normally started by running ACGI Enabler application
 - Can't use on a headless system?
 - Should be able to use `acgidctl` to start, stop, etc. `acgid`, but it depends on the existence of `/var/run/acgid.pid`, which isn't created when `acgid` is run

- Apple CGI (ACGI)
 - Could try running ACGIs on vanilla OS X with acgi dispatcher (<http://www.sentman.com/acgi/>)
 - No longer supported on Panther Server

- Services based on XML-RPC or SOAP transactions usually (and unfortunately) given the generic moniker of "Web services"
- Mac OS X 10.1 introduced integration with Apple Events and AppleScript
 - <http://developer.apple.com/documentation/AppleScript/Conceptual/soapXMLRPC/>
 - `aexml` translates SOAP and XML-RPC requests into Apple Events understood by applications
- Mac OS X 10.2 added the Web Services Core API
 - <http://developer.apple.com/documentation/Networking/Conceptual/WebServices/>
 - `/Developer/Tools/WSMakeStubs` does initial transformation of WSDL to AppleScript, Obj-C, and C++
- Other languages (Python, Perl, etc.) can be extended with XML-RPC or SOAP support

- Mac OS X Server includes Axis, the Apache Software Foundation's SOAP implementation, in `/System/Library/Axis/`
- `http://ws.apache.org/axis/`

- QuickTime Streaming Server (QTSS) bundled with Mac OS X Server
 - QTSS site, including link to Admin Guide: <http://www.apple.com/quicktime/products/qtss/>
 - Built on open-source Darwin Streaming Server (DSS)
 - <http://developer.apple.com/darwin/projects/streaming/>
- Config files, logs, web documents, media files, etc. all in `/Library/QuickTimeStreaming/`
- Server executable is `QuickTimeStreamingServer`
 - Two processes run: child provides service, parent handles housekeeping (like restarting child process if it dies)
 - Configuration similar to Apache

- QuickTimeStreamingServer listens on several ports
 - UDP 6970 and 6971 for RTP (Real Time Protocol) and RTCP (Real time Control Protocol)
 - TCP 554 and 7070 for RTSP (Real Time Streaming Protocol)
 - TCP 8001 and 8001 for MP3 broadcasting
 - TCP port 80 for tunneling over HTTP (optional)
- Administrative web interface (Streaming Server Admin, or SSA) provided by `streamingadminserver.pl`
 - Listens on TCP port 1220 (or 1240, if SSL enabled)
- QuicktimeStreamingServer startup item launches SSA if `QTSSERVER=--YES-` is set in `/etc/hostconfig`, then SSA starts QTSS if `qtssAutoStart=1` is set in `/Library/QuickTimeStreaming/Config/streamingadminserver.conf`

The screenshot displays the QuickTime Streaming Server administration web interface. The browser window title is "QuickTime Streaming Server: vamana" and the address bar shows "http://vamana:1220/parse_xml.cgi". The interface includes a "Start Server" button and a status indicator "Server is Idle". A sidebar on the left contains navigation links: Main, Broadcaster, Connected Users, Relay Status, General Settings (highlighted), Port Settings, Relay Settings, Log Settings, Playlists, Error Log, Access History, and Log Out. The main content area is titled "General Settings" and contains the following configuration options:

- Media Directory:** /Library/QuickTimeStreaming/Movies
- Secure Administration (SSL):** Enabled
- Max. Number of Connections:** 1000
- Max. Throughput:** 100 Mbps
- Default Authentication Scheme:** Digest (selected from a dropdown menu that also includes Basic)
- Start Server at System Startup:** Enabled

At the bottom of the settings area, there are three links with right-pointing arrows: "Change Admin Username/Password...", "Change Movie Broadcast Password...", and "Change MP3 Broadcast Password...". A "Save Changes" button is located in the bottom right corner.

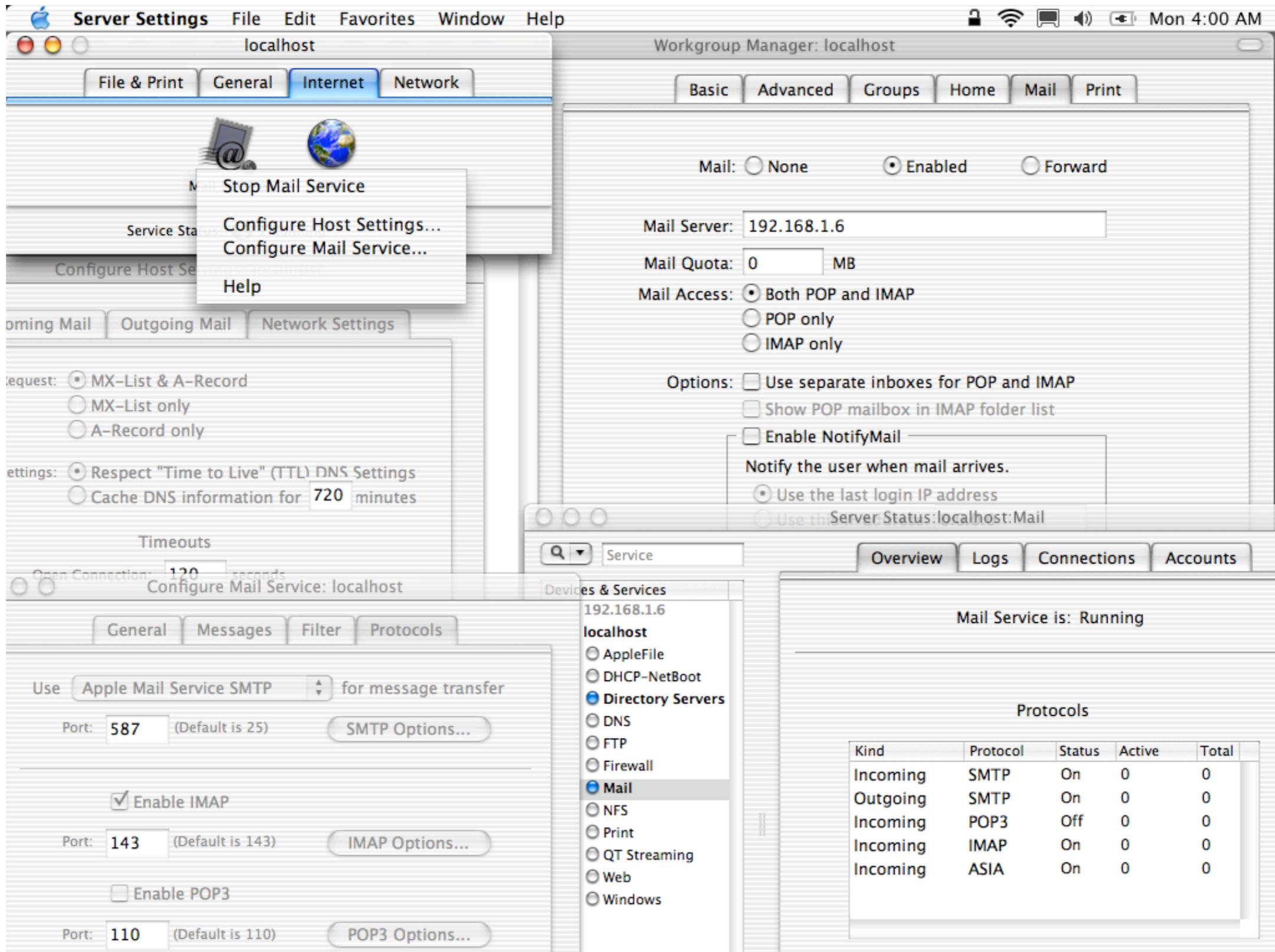
QuickTime Streaming Server admin web interface

- StreamingLoadTool
 - Emulates a client to measure server throughput
 - **Configured by** `/Library/QuicktimeStreaming/Config/streamingloadtool.conf`
- QuickTime Broadcaster (QTB) is bundled with Mac OS X
 - Performs capture and encoding of live multimedia content, and sends it to QTSS for streaming over a network
 - Can be managed through SSA if running on the same host as QTSS
 - **Configured by** `/Library/QuicktimeStreaming/Config/BroadcasterSettings.qtbr`

- Apple Mail Server
- Other Server Software

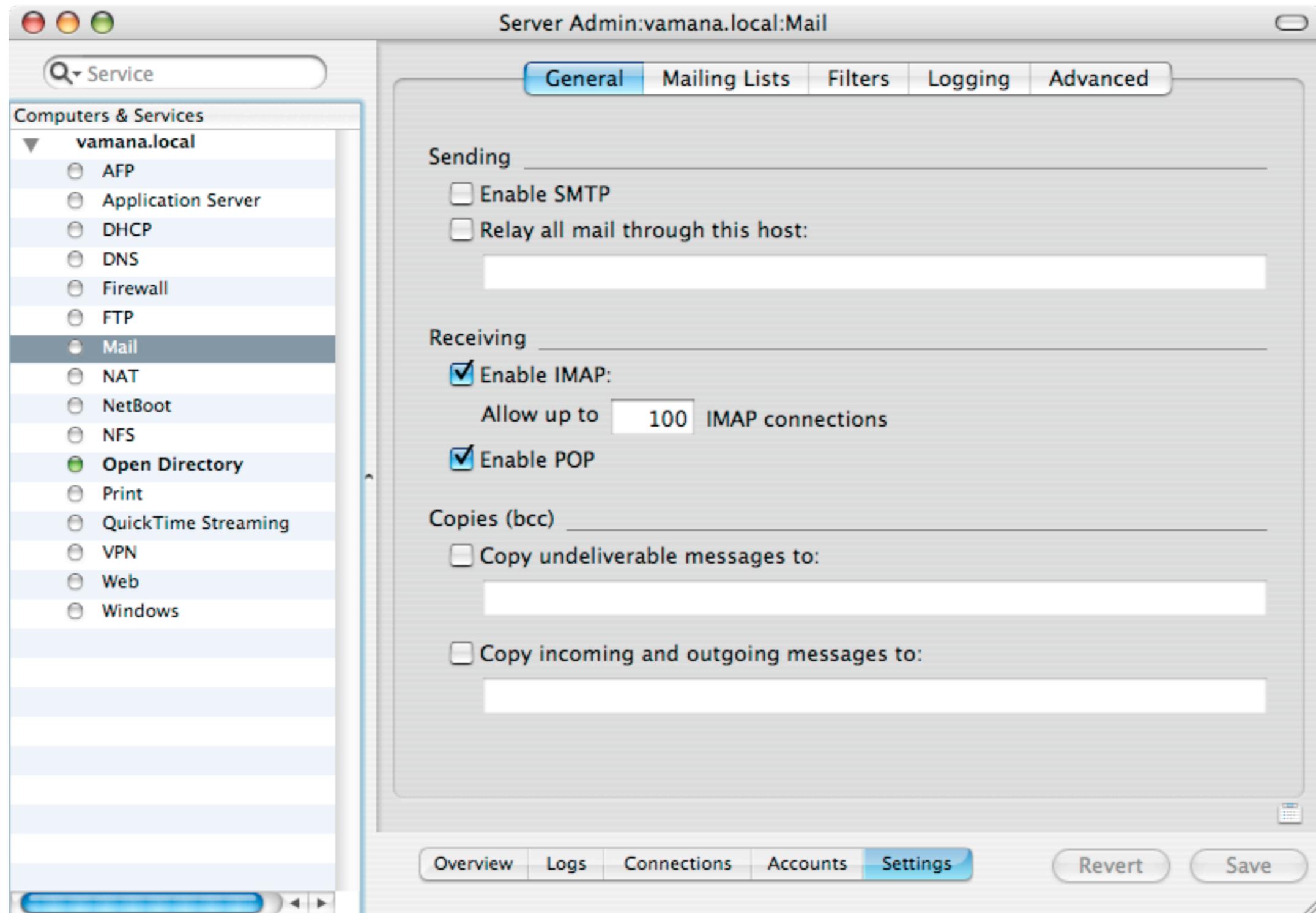


- Apple Mail Server (AMS) is derived from AppleShare IP (network software for the legacy Mac OS), and is bundled with Mac OS X Server
- In early versions of OS X Server, it was just a pain to use
 - Couldn't properly close open relays, buggy, etc.
 - Much better now, usable for simple environments (e.g., not hosting mail service for multiple distinct domains)
- Configured from Server Settings
 - Secure authentication options, SMTP relay filtering, message handling options, administrative access via IMAP, etc.
 - Config stored in `/config/AppleMailServer` in local OD domain
- Enable mail service for user accounts in Workgroup Manager
- View protocol status, current connections, list of mail accounts, and logs in Server Status



Apple Mail Server tools

- Panther Server has dropped AMS in favor of Postfix (for SMTP), Cyrus (IMAP, POP), and Mailman (lists)
- Sendmail bundled with Mac OS X
 - Useful feature: NetInfo supported as a database type
 - Can store aliases, mailertable, access DB, etc. in Open Directory
 - Can use it as SMTP MTA while AMS handles mailbox access protocols
- SquirrelMail bundled with Mac OS X Server for web access to email
 - Configured in `httpd_squirrelmail.conf` **and** `/etc/squirrelmail/`
 - Accessed as `http://www.example.com/WebMail/`
 - Uses AMS IMAP and SMTP service on the back end



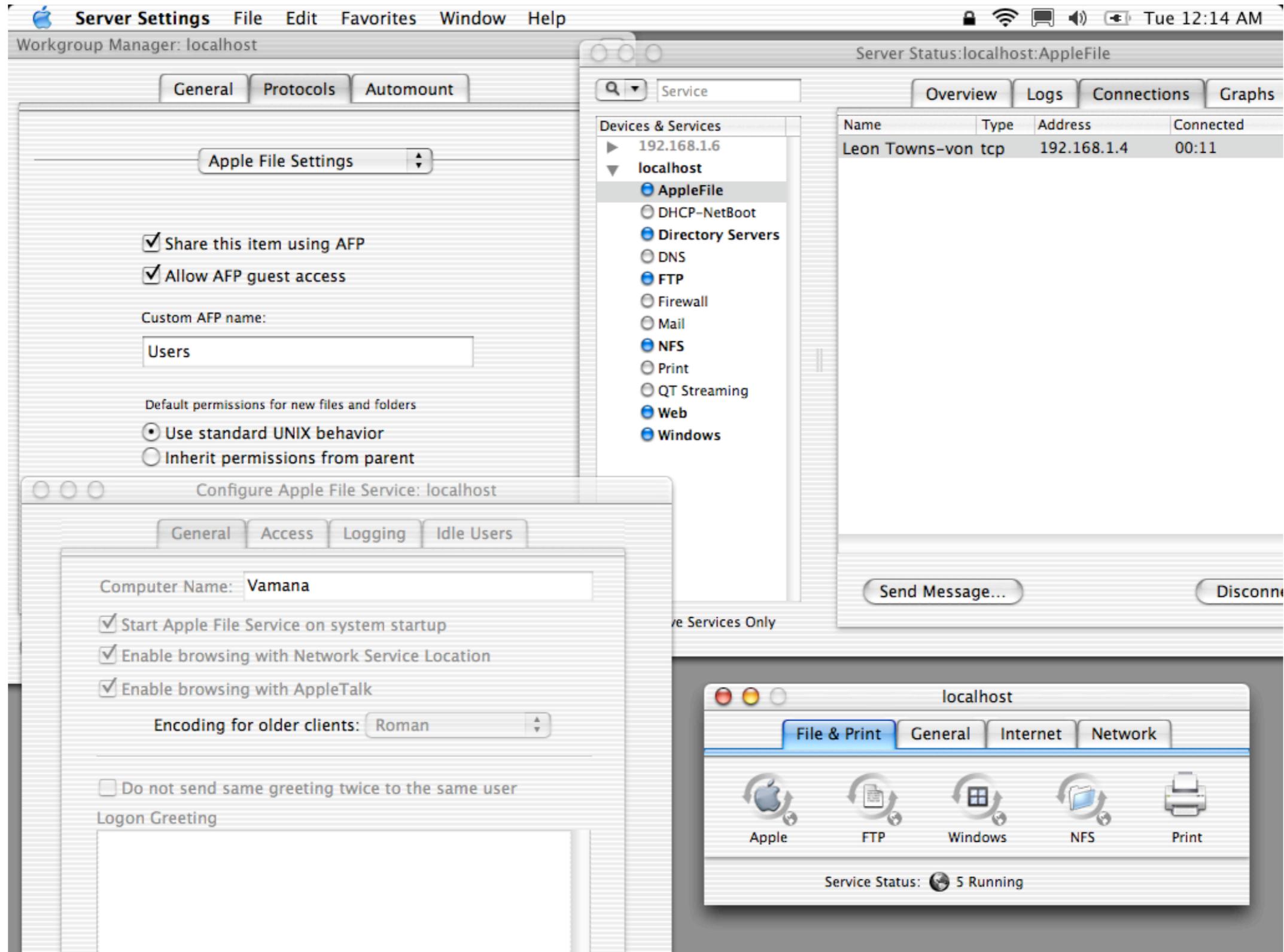
Panther Server mail management tools

- All major open-source servers ported: Postfix, Exim, Cyrus IMAP, etc.
- Cross-platform commercial packages as well
 - **Communigate Pro** (<http://www.stalker.com/Apple/>)
 - **Post.Office** (http://www.tenon.com/products/post_office/)

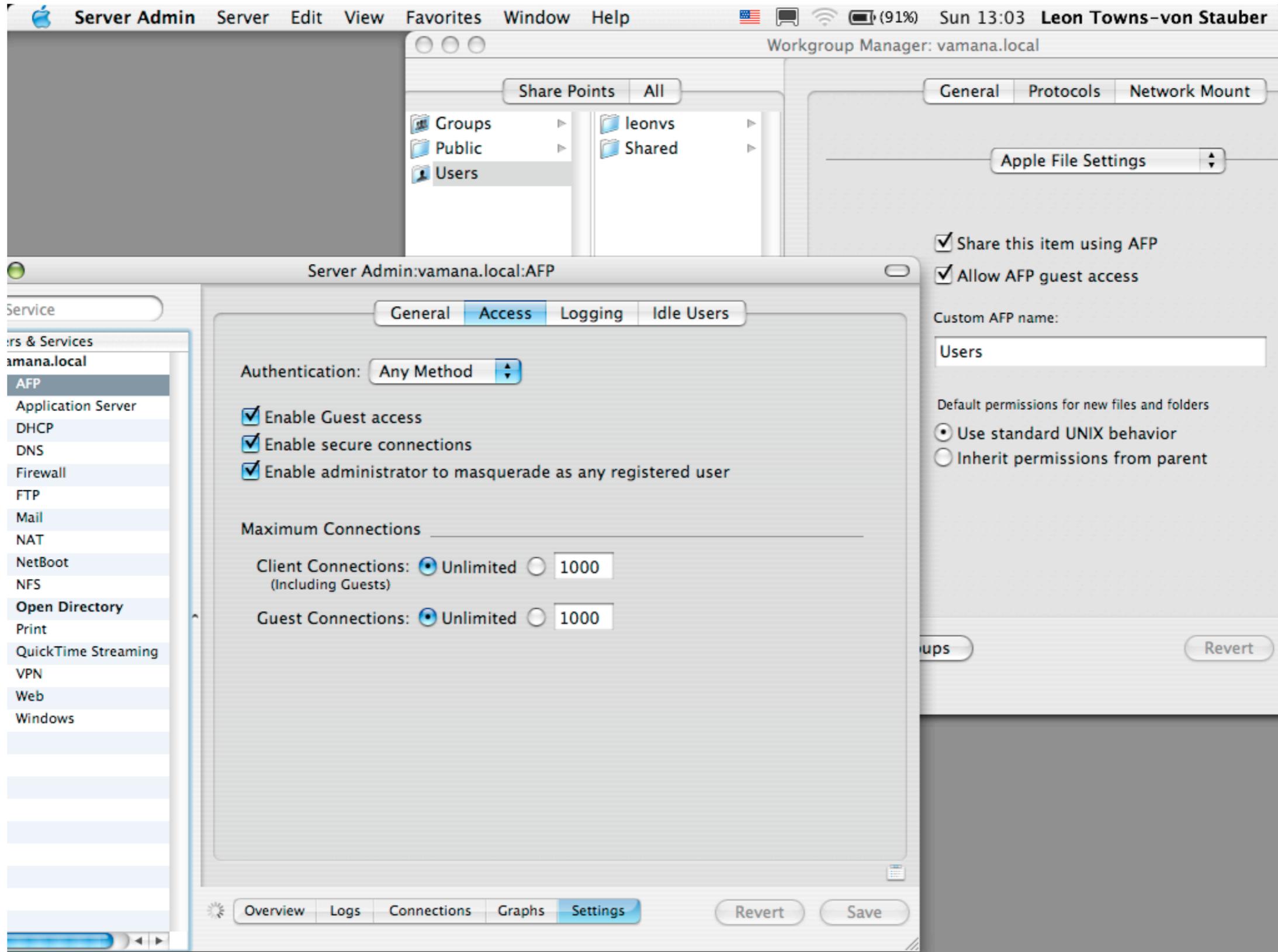
- Introduction
 - Apple Filesharing Protocol
 - Server Message Block
 - Network File System
 - File Transfer Protocol
 - Web-based Distributed Authoring & Versioning
- 

- Mac OS X offers one-click activation of AFP, SMB, and FTP service from Sharing Preferences
 - Sets values in `/etc/hostconfig`, which trigger startup items
 - No configuration GUIs
- Mac OS X Server
 - Use Workgroup Manager to define **share points**, folders that can be shared with AFP, SMB, NFS, or FTP
 - Use Server Settings to manage individual services
 - Use Server Status to view status, connections, and logs for services

- Apple Filesharing Protocol exhibits HFS+ semantics, intended for Mac clients
- Requires the least effort to use on OS X
- Features user-based authentication, SSH tunneling
- Client configuration in `~/Library/Preferences/.GlobalPreferences.plist`, under `com.apple.AppleShareClientCore` key
- Server configuration in OD, under `/config/AppleFileServer`
 - Sending HUP to `AppleFileServer` gets it to reread properties in OD

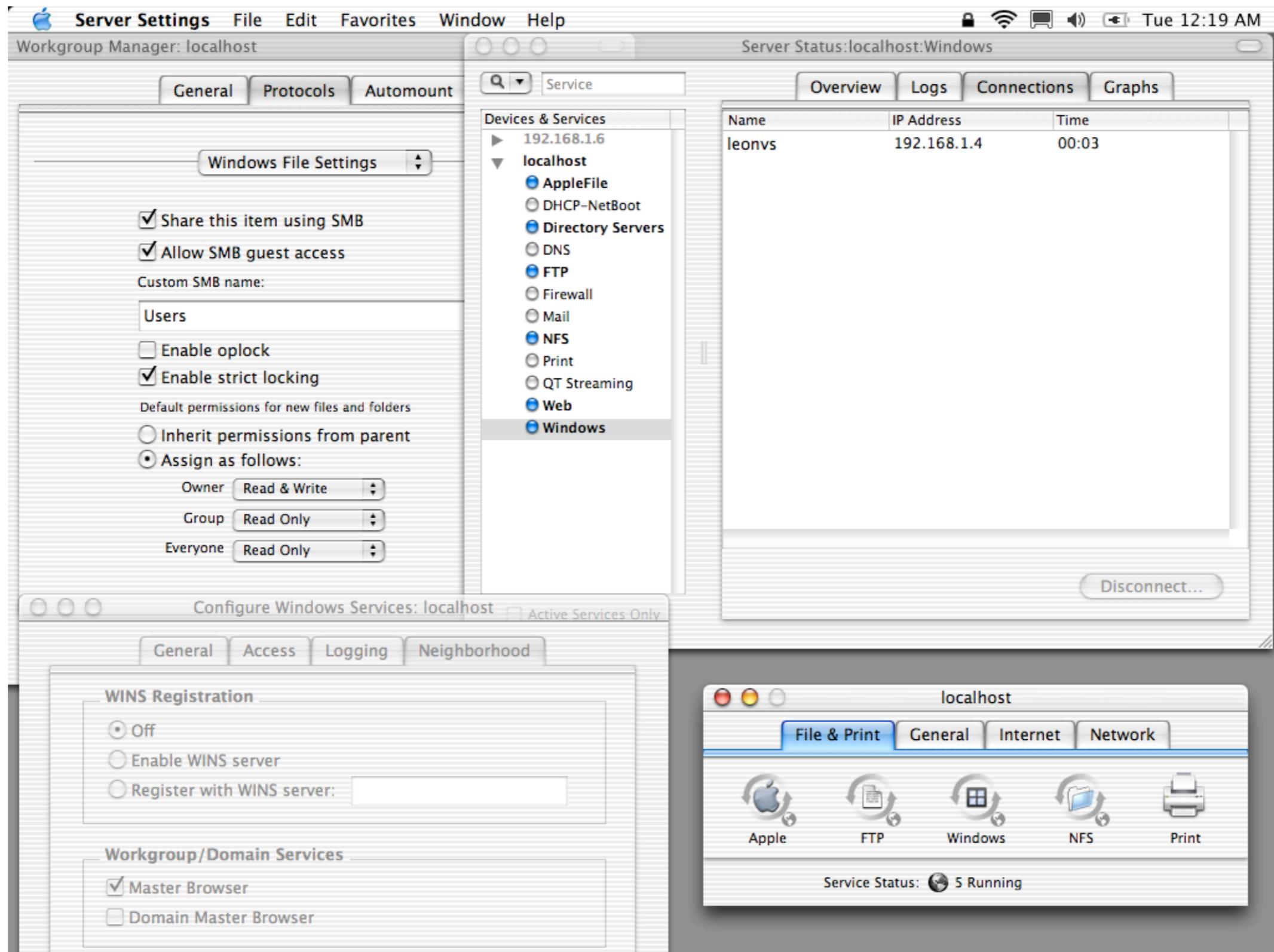


Mac OS X Server AFP tools

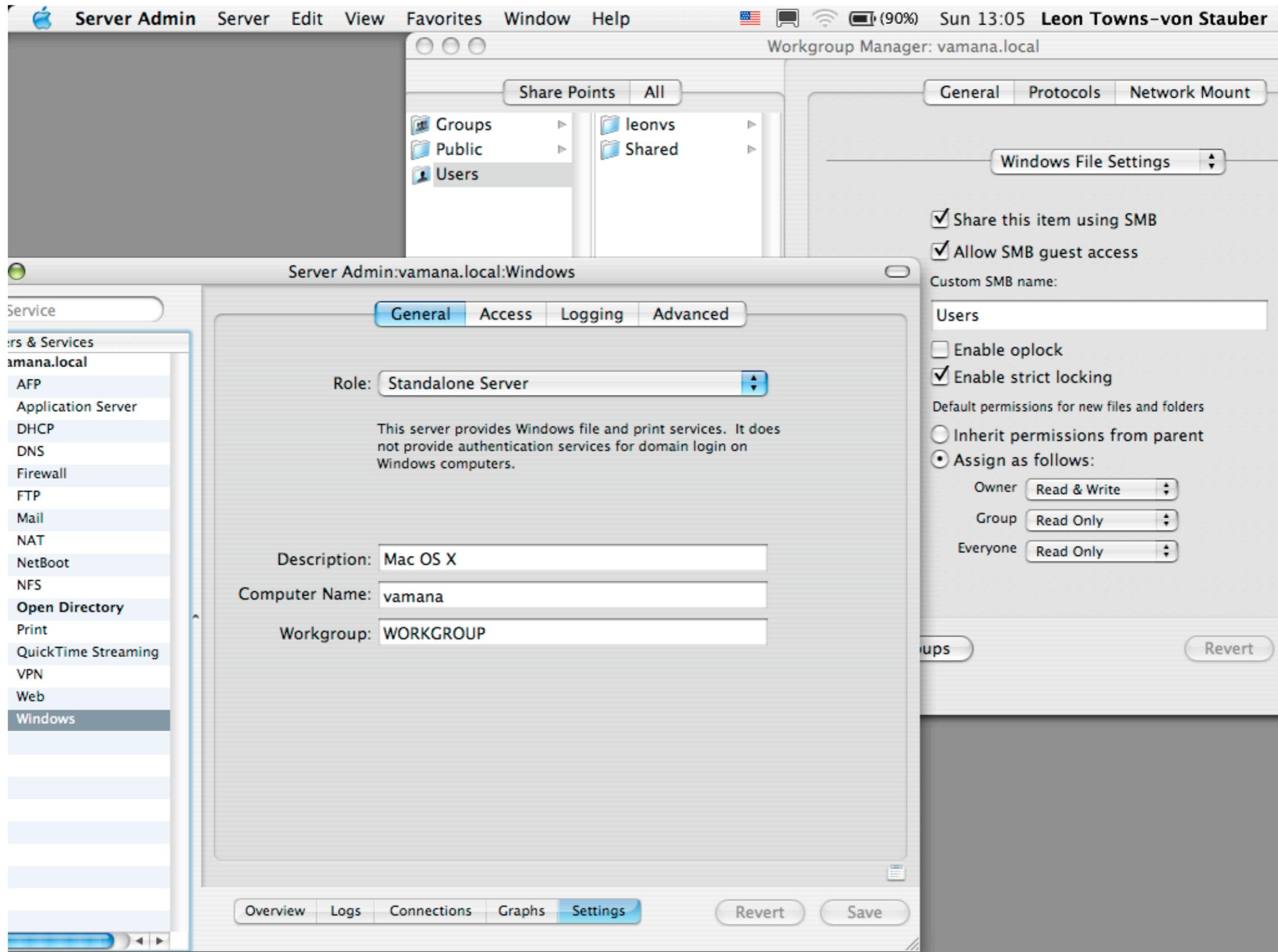


Panther Server AFP tools

- Server Message Block exhibits FAT semantics, intended for Windows clients
 - Also known as Common Internet File System (CIFS)
- Implemented by Samba on Mac OS X
- Features user-based authentication, excessive configuration flexibility
- Client configuration in `~/ .nsmbrc`
- Server configuration in `/etc/smb.conf`
 - Actually, on OS X Server, the GUI tools store config data in Open Directory (under `/config/SMBServer/` and `/config/SharePoints/`), then `smbadmind` (managed by `watchdog`) regenerates `smb.conf` and starts up the Samba daemons (`smbd` and `nmbd`)
 - If you want to make changes directly to `smb.conf`, keep the GUI from stepping on them with `chflags uchg /etc/smb.conf`

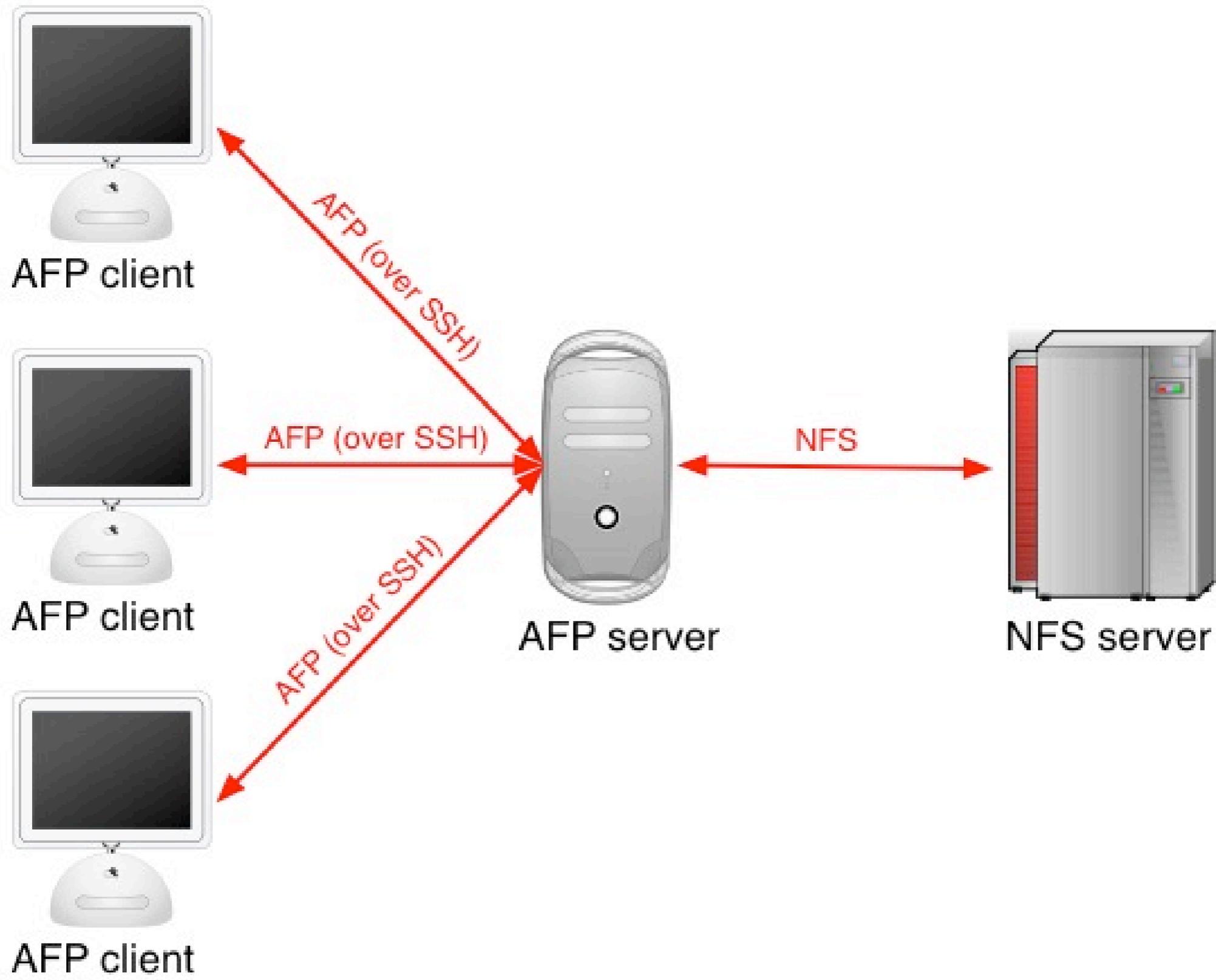


Mac OS X Server SMB tools

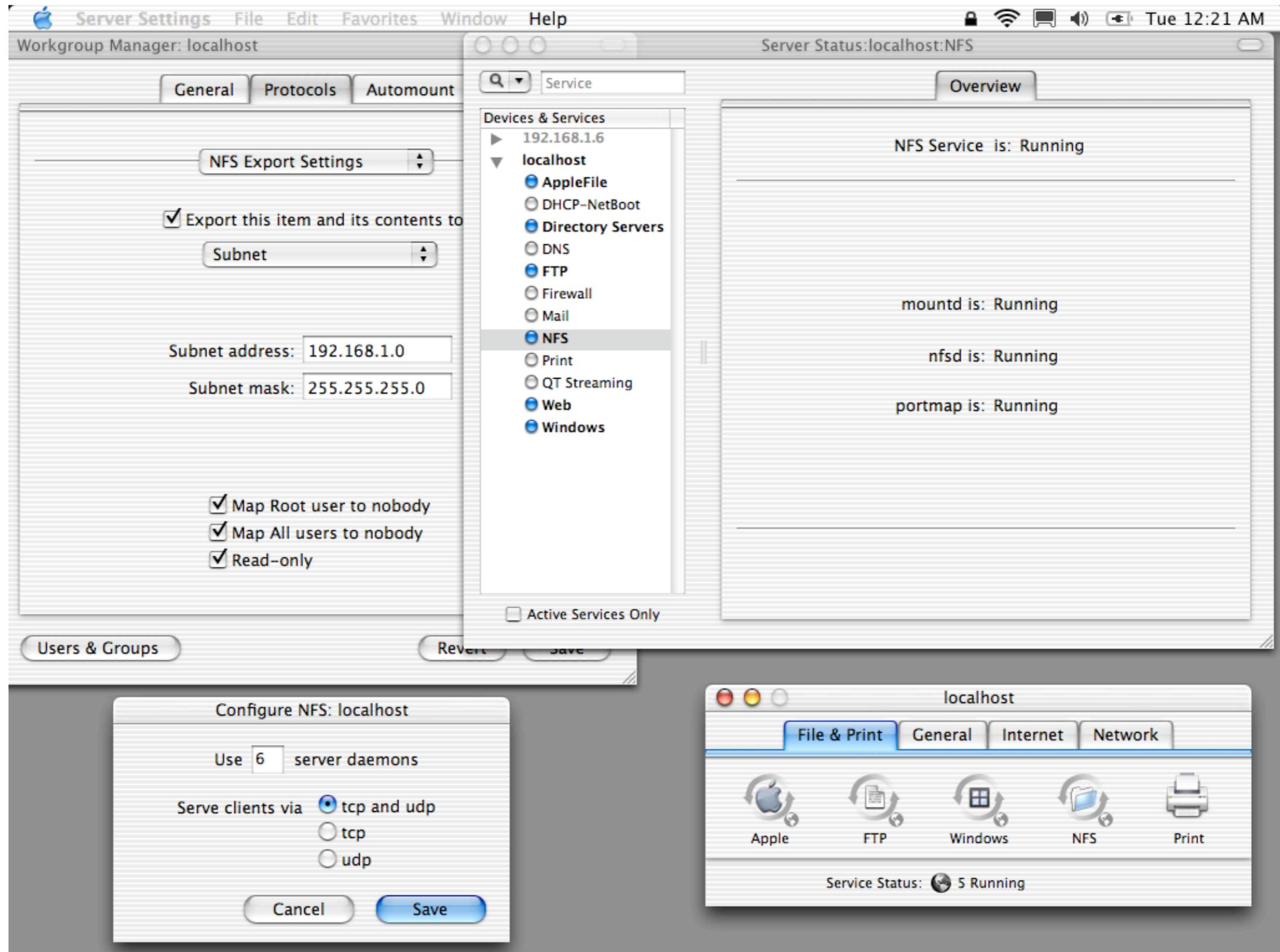


Panther Server SMB tools

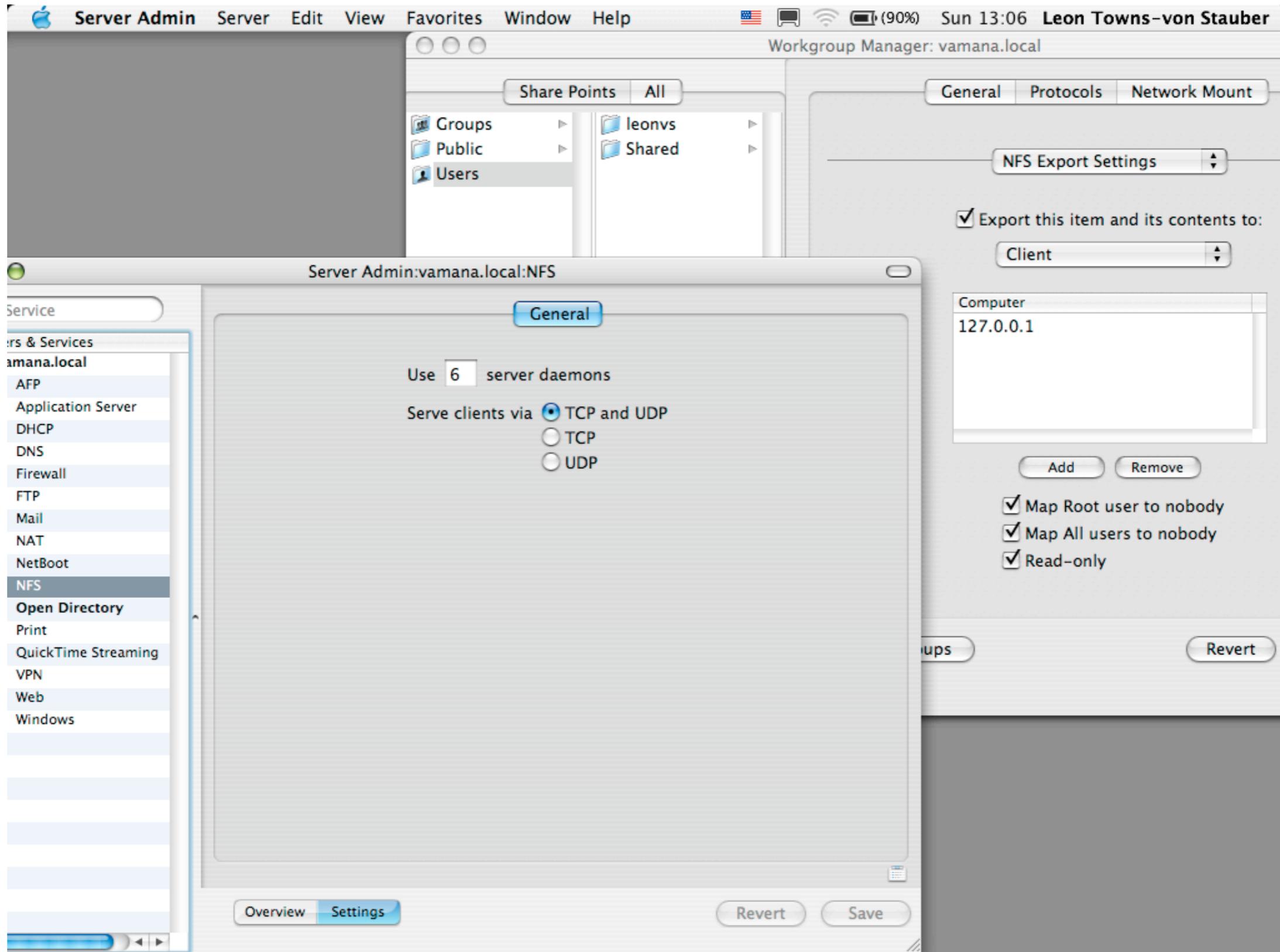
- Network File System exhibits UFS semantics, intended for UNIX clients
- Server configuration in Open Directory, in `/config/nfsd` and `/exports/`
- NFSManager (<http://www.bresink.de/osx/NFSManager.html>) provides graphical management
- Can reshare NFS mounts via AFP to gain user-based authentication, SSH encryption, and legacy Mac OS client support
 - AFP server's `root` must not be squashed on NFS server
 - 1) `sudo mkdir -m 0600 /nfs_reshares`
 - 2) Create mount points within `/nfs_reshares/`
 - 3) Mount NFS shares on mount points
 - 4) In Workgroup Manager, create share points from the NFS mounts, and set up AFP sharing



NFS resharing



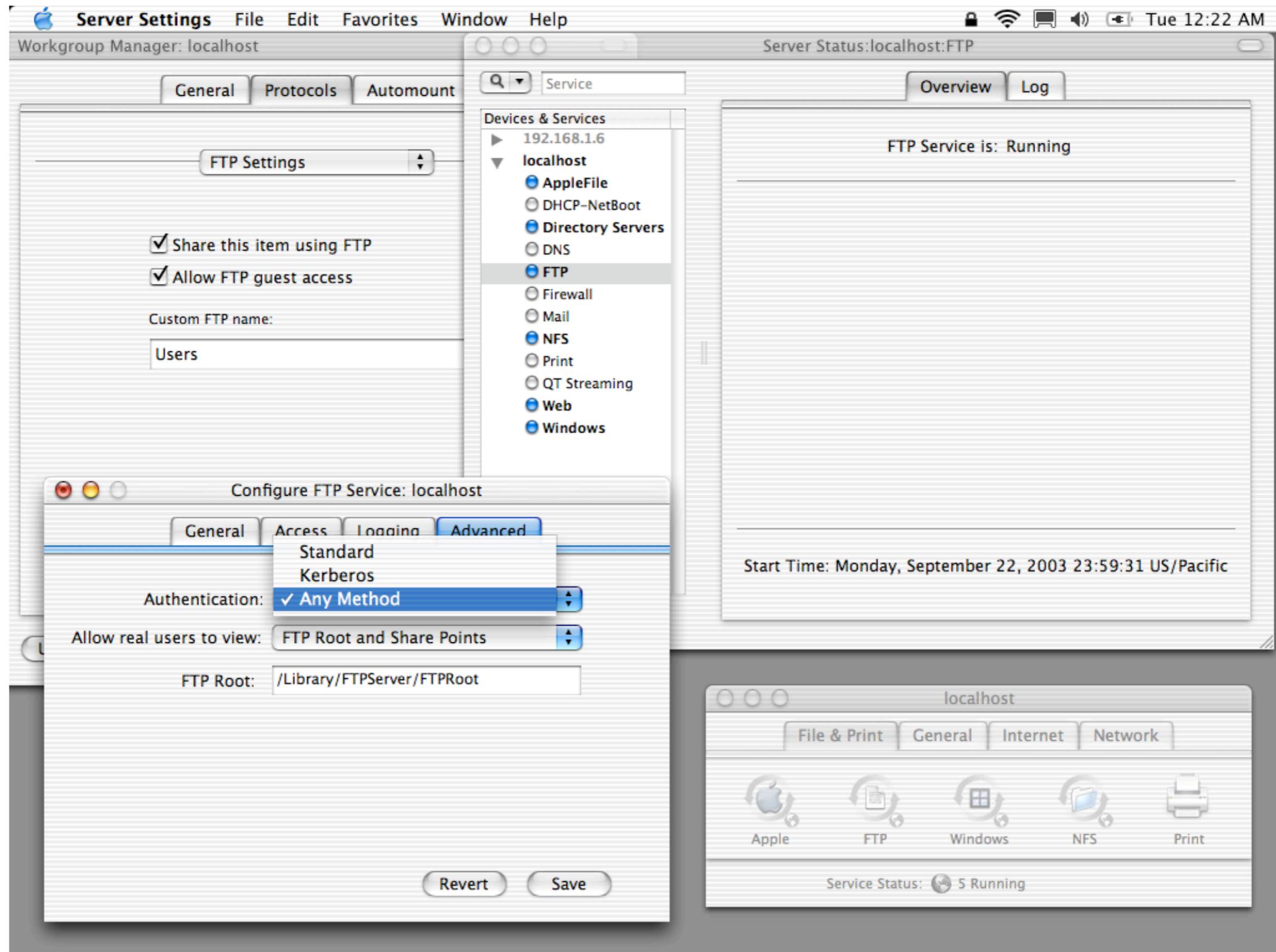
Mac OS X Server NFS tools



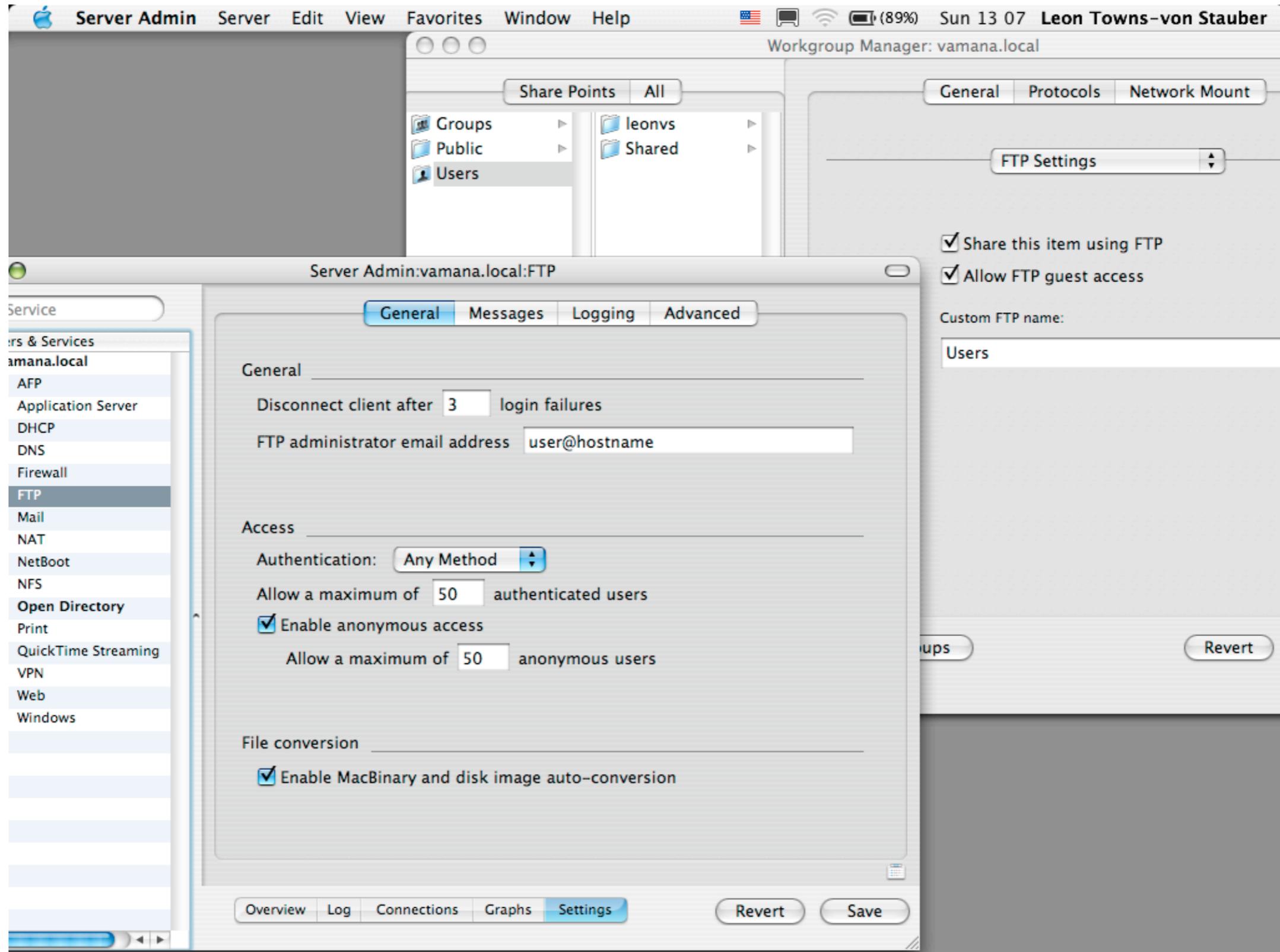
Panther Server NFS tools

- Vanilla OS X and OS X Server use different FTP server software
- Mac OS X: `lukemftpd` (ported from NetBSD)
 - Run by `xinetd` from `/etc/xinetd.d/ftp`
 - Server configuration in `/etc/ftpd.conf`, `/etc/ftpusers`
 - Logs to `/var/log/ftp.log`
 - As of 10.2, `ftpd` drops superuser privileges after authentication
 - Side effect: `chroot` fails, meaning anonymous FTP doesn't work

- Mac OS X Server: `xftpd` (modified `wu-ftpd`)
 - Run by `xinetd` from `/etc/xinetd.d/ftp`
 - Server configuration primarily in `/Library/FTPServer/Configuration/ftpaccess`
 - `influence_listings`: If set to `yes`, automatically converts files with resource forks into MacBinary (`.bin`) format using the `macbin` utility, and converts applications and other bundles into disk images (`.dmg`) with the `mkdmg` utility
- Logins automatically jailed, either in home directory or in `/Library/FTPServer/FTPRoot/` (for anonymous access)



Mac OS X Server FTP tools



Panther Server FTP tools

- Web-based Distributed Authoring and Versioning is an HTTP extension offering read/write access to files on a web site
- <http://www.webdav.org/>
- .Mac iDisk is a WebDAV share
- Implemented in Apache with the `mod_dav` module
- Can be enabled by adding following lines in `/etc/httpd/httpd.conf`:

```
LoadModule dav_module libexec/httpd/libdav.so
AddModule mod_dav.c
DAV On
DAVLockDB "/var/run/.davlock"
```

- Also need to set up authentication (preferably digest)

- Common UNIX Printing System
- Print Center
- Printer Sharing
- Command-Line Tools



- With Mac OS X 10.2, the open-source Common UNIX Printing System (CUPS) took over printing functionality
- `http://www.cups.org/`
- Supports a variety of printing protocols: BSD LPR, SysV `lp`, PAP (AppleTalk), direct USB, etc.
- Primary is the Internet Printing Protocol (IPP), an extension of HTTP
- Server executable is `cupsd`
 - Config files in `/etc/cups/`; primary is `cupsd.conf`
 - Configuration similar to Apache
 - Logs to `/var/log/cups/`
 - Started by `PrintingServices` startup item when `CUPS=-YES-` set in `/etc/hostconfig`

- CUPS includes an administrative web interface
 - `http://localhost:631/`
 - Add and configure printers, manage printer classes, manage jobs, view documentation
- Add a bunch of extra printer drivers with Gimp-Print (`http://gimp-print.sourceforge.net/MacOSX.php3`)
- Included with Panther

The screenshot shows a web browser window titled "Printer on localhost - CUPS v1.1.15-2" with the address bar at "http://localhost:631/printers". The interface features a navigation menu with "ESP", "Administration", "Classes", "Help", "Jobs", "Printers", and "Software". Under the "Printers" section, two printers are listed:

- Stylus_Photo_820** EPSON Stylus Photo 820
 - Description: Stylus Photo 820
 - Location: Buddha
 - Printer State: idle, accepting jobs.
 - Device URI: file:///dev/null
 - Buttons: Print Test Page, Stop Printer, Reject Jobs, Modify Printer, Configure Printer, Delete Printer
- Stylus_Scan_2500** EPSON Stylus Scan 2500
 - Description: Stylus Scan 2500
 - Location: Buddha
 - Printer State: idle, accepting jobs.
 - Device URI: usb:///foo/
 - Buttons: Print Test Page, Stop Printer, Reject Jobs, Modify Printer, Configure Printer, Delete Printer

At the bottom, there is an "Add Printer" button and a copyright notice: "Copyright 1993-2002 Easy Software Products, All Rights Reserved. The Common UNIX Printing System, CUPS, and the CUPS logo are the trademark property of Easy Software Products. All other trademarks are the property of their respective owners."

CUPS web interface

- Print client administration is typically performed with Print Center
 - Add/remove printers, manage jobs, etc.
- When adding a printer, select from the full list of types supported by CUPS by option-clicking the Add button in the toolbar, then selecting Advanced from the pop-up menu

- On vanilla Mac OS X, printer sharing is a simple matter of enabling it in Sharing Preferences
 - Causes CUPS to advertise locally configured printers using IPP
 - IPP clients browse for printers on UDP port 631, print to TCP port 631
- Mac OS X Server printer sharing is more flexible
 - Add queues, enable service, etc. in Server Settings->File & Print->Print
 - View queues and logs in Server Status
- Samba can be used to share printers to Windows clients

- **CUPS:** `cupsd`, `cups-polld`, `lpoptions`, `lppasswd`, `lpinfo`, `cups-config`
- **BSD:** `lpr`, `lpq`, `lprm`, `lpc`, `cups-lpd`
- **SysV:** `lp`, `lpstat`, `lpmove`, `accept`, `reject`, `cancel`, `disable`, `enable`, `lpadmin`
- **AppleTalk:** `at_cho_prn`, `atprint`, `atq`, `atrm`, `atstatus`, `atprintd`
- **SMB:** `smbutil`, `smbclient`
- **Mac OS X Server:** `PrintServiceAccess`, `PrintServiceMonitor`

- Web Sites
- Mailing Lists
- Books



- **Apple's Mac OS X site**

- <http://www.apple.com/macosx/>

- **Mac OS X Hints**

- <http://www.macosxhints.com/>

- **Occam's Razor Apple/NeXT page**

- <http://www.occam.com/links/apple.html>

- MacOSX-admin (Omni Group)

- <http://www.omnigroup.com/developer/maillinglists/macosx-admin/>

- macos-x-server (Apple)

- <http://lists.apple.com/mailman/listinfo/macos-x-server/>

- Mac OS X in a Nutshell
 - Jason McIntosh, Chuck Toporek, Chris Stone

- This talk has focused on issues mostly specific to Mac OS X
 - But remember that Mac OS X is UNIX, and similar considerations apply as to any other UNIX platform
- Evaluation forms
- Some considerations
 - Level of detail, pacing, slides
 - Content (things you'd have liked to see, or liked to see gone)
- BoF: Tuesday 9 PM
- Slides available at <http://www.occam.com/osx/>
- Q & A