UNIX: Working the Command Line in OS X

Russell Poucher
russell@creativeresources.net
Apple Certified System Administrator
Using the Command Line

$ mkdir /Users/Shared/new_docs
$ mv hde*.txt /Users/Shared/new_docs
$ chown warren /Users/Shared/new_docs
Advantages of the Command Line

- Advanced configuration and features
- Running a command as a different user
- Efficiency
- Remote administration and troubleshooting
Command Line Interface

Getting to it

>console

SSH

Single-user mode

Terminal application
Available Shells in OS X

- **bash**: "Bourne-again" shell (default)
- **sh**: Bourne shell is present but physical copy of bash
- **csh**: C shell
- **tcsh**: Tenex C Shell (enhanced csh)
- **ksh**: Korn shell
- **zsh**: Close to ksh (includes features from csh, tcsh)

All shells are located in /bin except tclsh /usr/bin/tclsh
Shell Resource Files—/bin/bash

**Systemwide:**
- `/etc/profile`—Systemwide initialization for login shell

**User:**
- `~/.bash_profile`—Personal initialization file for login shells
- `~/.bashrc`—Individual per-interactive-shell startup file
- `~/.bash_logout`—Individual login shell cleanup file, executed when a login shell exits
- `~/.inputrc`—Individual readline initialization file
- `~/.bash_history`—Command history file

See man pages for the other shells
Change Default Shell

- dscl
- Workgroup Manager
- Server.app
- System Preferences
- Terminal.app

$ chsh -s shell
Automation with Shell Scripting

- Everything you can use in the shell you can use in a shell script.
- This enables building workflows with scripts.
- A lot is automated with shell scripts
  - /private/etc/periodic/
  - /System/Library/ServerSetup/
  - ...
- Anatomy of a script covered later
Example Commands

**Example Commands**

```bash
ls

ls -lA ~/Documents ...

Redirected output, or other commands, as needed

**BSD command:**
“ls” is the command to display a list of a folder’s contents.

**Option(s):**
Options add conditions, limits, or other modifiers to the BSD command. The “l” option requests a “long” listing, and the “A” option omits the . and .. entries from the list. Note the spaces between the different components of the command.

**Argument(s):**
This is the recipient of the action. In this case, the command will provide a list of the Documents folder within the current user’s home folder.

**man ls**

**mount -uw /**
Get Info About Commands

**Manual**
- All UNIX systems come with an extensive set of manuals
- Organized into different sections
- Use them to look up arguments and usage

- `man ssh`
- `man -k secure`
- `man 2 open`
Shell Filename Wildcard

- The * wildcard helps you to specify multiple files.
- List all of the user folders:
  
  `ls -ld /Users/*`

- List all of the pdf files in your Documents folder:

  `ls -l ~/Documents/* .pdf`
Command Line Interface

Using sudo to gain root permissions
- root disabled by default
- sudo -s
- sudo passwd root

Change in prompt

Be careful with root!
Playing with Files

- **touch**
  - Creates a new, empty file

- **.DS_Store**

- **.Trash**

- **mv**

- **rm**

- **cp**
  - Unix command do not recognize a resource fork
Directories

- push
  - remembers where you were
- popd - puts you back
- pwd - where you are
- mkdir
- cpdir
- rmdir
- Using wildcards - (?, [], *)
File/Folder Permissions

Type of file:
- d for directory
- - for file
- l for symbolic link

Owner permissions:
- rwx = read, write and execute

Group permissions:
- -wx = write and execute

Others' permissions:
- -wx = write and execute
Working at the Command Line

- **Command Prompt**: The text `client17:~ chris$` indicates the current directory and the user who is logged in.
- **Command Entry**: The text `ls -la` is an example of a command entry.
- **Command Results**: The output of the `ls -la` command shows various files and directories.

This image demonstrates how to use the command line interface to interact with a computer.
## Command Line Entry

<table>
<thead>
<tr>
<th>Shortcut</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tab</td>
<td>Completes word being typed</td>
</tr>
<tr>
<td>Drag folder to Terminal</td>
<td>Enters pathname</td>
</tr>
<tr>
<td>Up and Down Arrow keys</td>
<td>Accesses prior commands</td>
</tr>
<tr>
<td>Control-A</td>
<td>Moves cursor to the beginning of the line</td>
</tr>
<tr>
<td>Control-E</td>
<td>Moves cursor to the end of the line</td>
</tr>
<tr>
<td>Control-F</td>
<td>Moves forward one character</td>
</tr>
<tr>
<td>Control-B</td>
<td>Moves backward one character</td>
</tr>
<tr>
<td>Esc F</td>
<td>Moves forward one word</td>
</tr>
<tr>
<td>Esc B</td>
<td>Moves backward one word</td>
</tr>
<tr>
<td>Control-C</td>
<td>Terminates command in progress</td>
</tr>
<tr>
<td>Control-L or “clear”</td>
<td>Clears screen</td>
</tr>
</tbody>
</table>
Command Line Issues

- Easy to make serious mistakes and there is no trash or undo

- Exact syntax required and some characters are easy to confuse:
  - O and 0 (uppercase letter O and the numeral zero)
  - - and ~
  - l and 1 (lowercase letter l and the numeral one)

- Exact pathnames are important—spaces in filenames and paths can have unexpected results
Changing File/Folder Permissions

chown apple:staff /Users/Shared/ReadMe.rtf
chown owner[:group] file ...

chmod g+w /Users/Shared/ReadMe.rtf
chmod mode file ...

The options for chmod specify three things: who is affected, the change, and the permissions being applied to the folder or file.

<table>
<thead>
<tr>
<th>who</th>
<th>action</th>
<th>permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>u (user/owner)</td>
<td>+, -, =</td>
<td>r (read)</td>
</tr>
<tr>
<td>g (group)</td>
<td></td>
<td>w (write)</td>
</tr>
<tr>
<td>o (others/everyone else)</td>
<td></td>
<td>x (execute)</td>
</tr>
<tr>
<td>a (all=ugo)</td>
<td></td>
<td>t (sticky bit)</td>
</tr>
</tbody>
</table>
Text Editors on OS X

- Text Edit (GUI)
- TextWrangler
- vi
- nano (pico)
- emacs
Processes & Daemons

- Activity Monitor (GUI)
- top Command
- kill
  - for processes
  - for daemons
- sudo kill
  - processes or daemons you don’t own
Managing Processes from the Command Line

- `ps`
- `top`
- `kill`
- `killall`
Using Apple Remote Desktop to Send Commands
The life of an administrator is about automating repetitive tasks.

**automation** |ˌöteɪˈmɑː ʃən|  

The use of largely automatic equipment in a system of manufacturing or other production process.
Why automate?

- Eliminate repetitive tasks
- Help with time-consuming tasks
- Quality by avoiding mistakes
- Ensure task execution
- More efficient
Scripting

Mac OS X’s UNIX heritage provides a huge range of open source scripting languages.

Out-of-the-box support for:
- Perl
- Python
- PHP
- Tcl
- Ruby
- Shells
  - bash, ksh, zsh, and csh
Apple Technologies

**AppleScript**
- System and applications scripting
- Can be recorded
- English-like syntax

**Automator**
- Script and workflow creation
- Drag-and-drop creation process
- The action itself is an Xcode project
  - Written in AppleScript, Cocoa, Shell Script, ...
Schedule/Automate Tasks

Part of automation is scheduling your tasks.

UNIX has a variety of tools dedicated to scheduling.

- `cron(8)`
- `at(1)`
- `periodic(8)`

Apple-specific

- `Startup Items`
- `launchd(8)`
Scheduling

- Run always
- Run on demand
- Run at(1) a specific time
- Run periodic(8) at a specific time
cron

- Executes periodic scheduled commands
- Only starts when it sees files in:
  - `/etc/crontab`
  - `/usr/lib/cron/tabs`
- Crontab folder changed location
  - `/var/cron/tabs (10.4)`
  - `/usr/lib/cron/tabs (10.5)`
- Functionality replaced by `launchd(8)`
at—Queue for Later Execution

- `at(1)`
  - Executes commands at a specified time
- `atq(1)`
  - Lists the pending jobs
- `atrm(1)`
  - Deletes jobs
- `batch(1)`
  - Executes commands when system load levels permit

**Disabled by default (as of 10.5.2)**

```
$ launchctl load -w /System/Library/LaunchDaemons/com.apple.atrun.plist
```
periodic—Run Periodic Commands

- **daily**
  - Runs at 3:15 a.m.
  - Executes scripts in `/etc/periodic/daily/`

- **weekly**
  - Runs every Saturday at 3:15 a.m.
  - Executes scripts in `/etc/periodic/weekly/`

- **monthly**
  - Runs every first day of the month at 5:30 a.m.
  - Executes scripts in `/etc/periodic/monthly/`

```bash
$ ls -cl /etc/periodic/
  daily/
  100.clean-logs
  110.clean-tmps
  130.clean-msgs
  430.status-rwho
  500.daily
  600.daily.server
  601.daily.server.krb5kdc
  700.daily.server.cyrus
```
Add Your Own Periodic Scripts

- Take advantage of the existing process
  - Move your script into the right folder
    - /etc/periodic/daily/, /etc/periodic/weekly/, or /etc/periodic/monthly/
  - Add a number in front to set the execution order
  - Make executable and readable for owner/group
  - Change owner to root and the group to wheel

$ sudo mv pretendCoBAK.sh /etc/periodic/weekly/850.pretendCoBAK.sh
$ sudo chmod 550 /etc/periodic/weekly/850.pretendCoBAK.sh
$ sudo chown root:wheel /etc/periodic/weekly/850.pretendCoBAK.sh
Startup Items

- Deprecated in favor of `launchd(8)`
- Located in
  - `/System/Library/StartupItems`
  - `/Library/StartupItems`
- Perform basic system initialization tasks
- Prepare the system for normal operation
- Used via `SystemStarter`
launchd (8)

- launchd debuted in Mac OS X v10.4 as a replacement for a long list of different UNIX components.
- In a nutshell, launchd has two main areas:
  - Interprocess communication
  - Starting and monitoring processes
Purpose

launchd is designed to replace:

- init
- SystemStarter
- mach_init
- /etc/rc
- at
- cron
- watchdog
- xinetd
Key Features

- Fully asynchronous bootstrap
- Fault isolation and error recovery
- Pay as you go (start on demand)
- Security (privilege separation)
launchctl

Talking to launchd from the shell

Some useful subcommands

- list
- load <plist>
- unload <plist>
- log level debug
- start <job label>
- stop <job label>
- help

<table>
<thead>
<tr>
<th>PID</th>
<th>Status</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>edu.mit.Kerberos.KerberosAgent</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>com.apple.CoreMIDIServer</td>
</tr>
<tr>
<td>571</td>
<td></td>
<td>com.apple.UserEventAgent-Aqua</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>com.apple.syncservices.uihandler</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>com.apple.syncservices.SyncServer</td>
</tr>
<tr>
<td>570</td>
<td></td>
<td>com.apple.Spotlight</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>com.apple.ServerScanner</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>com.apple.scrod</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>com.apple.ScreenSharing.server</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>com.apple.ReportPanic</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>com.apple.ReportCrash.SafetyNet</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>com.apple.ReportCrash</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>com.apple.RemoteUIT</td>
</tr>
<tr>
<td>71329</td>
<td></td>
<td>com.apple.RemoteDesktop.agent</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>com.apple.quicklook</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>com.apple.qtkitserver</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>com.apple.PubSub.Agent</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>com.apple.ProblemReporter</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>com.apple.PreferenceSyncAgent</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>com.apple.pictd</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>com.apple.PCIESlotCheck</td>
</tr>
<tr>
<td>578</td>
<td></td>
<td>com.apple.pboard</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
launchd Instances

- One launchd for each user

$ launchctl list | wc -l
  79

$ sudo launchctl list | wc -l
  161

$ ps -o user -o command -ax | grep /sbin/launchd
  root /sbin/launchd
  localadmin /sbin/launchd
**CPU**

- **cpuwalk.d**
  - Measure which CPUs a process runs on

- **top -u**
  - List processes by CPU usage
Memory

vm_stat(1)

- Processes: 104 total, 3 running, 5 stuck, 96 sleeping... 444 threads
- Load Avg: 0.45, 0.41, 0.38
- CPU usage: 11.21% user, 12.07% sys, 76.72% idle
- SharedLibs: num = 3, resident = 53M code, 3980K data, 3944K linkedit.
- MemRegions: num = 27350, resident = 658M + 21M private, 595M shared.
- PhysMem: 276M wired, 1112M active, 512M inactive, 1900M used, 14M free.
- VM: 15G + 373M

$ vm_stat
Mach Virtual Memory Statistics: (page size of 4096 bytes)
Pages free: 17409.
Pages active: 284415.
Pages inactive: 146856.
Pages wired down: 73583.
"Translation faults": 432342007.
Pages copy-on-write: 5581205.
Pages zero filled: 247414912.
Pages reactivated: 1059381.
Pageins: 918500.
Pageouts: 611593.
Object cache: 503460 hits of 3926331 lookups (12% hit rate)

top(1)

- Object cache: 503460 hits of 3926331 lookups (12% hit rate)
Memory
Disk

- `df(1)`
  - Display free disk space

- `iostat(1)`
  - Report I/O statistics

- `lsof(8)`
  - List open files

- `fs_usage(1)`
  - Report system calls

```
$ df -h -t hfs
Filesystem     Size   Used  Avail Capacity Mounted on
/dev/disk0s3   74Gi   51Gi   23Gi    69%    /Volumes/MediaFWStore
/dev/disk1s2  465Gi  347Gi  118Gi    75%    /Volumes/MediaFWStore

$ iostat -w 1 -d disk0 disk1
disk0           disk1

KB/t tps MB/s     KB/t tps MB/s

4.00   1  0.00     0.00   0  0.00
420.00  10  4.09    41.71   7  0.28
1024.00   1  1.00     0.00   0  0.00
768.00   3  2.24     0.00   0  0.00
1024.00   4  3.99     0.00   0  0.00

# lsof /Volumes/MediaFWStore/
COMMAND   PID  USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
bash    63817 media  cwd    DIR   14,5      680    2 /Volumes/MediaFWStore/

# lsof -u _teamsserver
COMMAND PID         USER   FD   TYPE    DEVICE  SIZE/OFF     NODE NAME
Python   55 _teamsserver  cwd    DIR      14,2      1258        2 /
Python   55 _teamsserver  txt    REG      14,2     38144   365614 /System/Library/Frameworks/Python.framework/Versions/2.5/Resources/Python.app/Contents/MacOS/Python
Python   55 _teamsserver  txt    REG      14,2     92900   364097 /System/Library/Frameworks/Python.framework/Versions/2.5/lib/python2.5/lib-dynload/time.so
...

# fs_usage -w -f filesystem Python
10:54:40.359 getdirentries F=14 B=0x0 0.000047 Python
```
Disk

Disk Activity

CPU: System Memory

Disk Activity

Disk Usage

Network

Volumes:

Name | Name | Name
--- | --- | ---
Ga | Server_2 | Server_1
25.2 | 30.5 | 100.0

All Disks

Drive: Drive 2

Kind: disk4

Manufacturer: Seagate

Model: ST380101AS PN

Bytes Read: 28.10 GB

Bytes Written: 23.31 GB

Capacity: 76.33 GB

Pre-Failure Warning: No Warnings

File Sharing: 1 connected users

iCal: 0 events

iChat: 0 connected users

Drive Bay: Internal (Bay 1)

Space utilized: 14.71 GB

Space Free: 15.80 GB

16,967,471,104 bytes

30.52 GB

File Sharing: 1 connected users

VPN: Offline

CPU: System Memory

Disk Activity

Disk Usage

Network

Disk Usage: Server_2

File Sharing: 1 connected users

iCal: 0 events

iChat: 0 connected users

Drive Bay: Internal (Bay 1)

Space utilized: 14.71 GB

Space Free: 15.80 GB

16,967,471,104 bytes

30.52 GB
# Network

- netstat(1)
- tcpdump(1)

<table>
<thead>
<tr>
<th>Name</th>
<th>Mtu</th>
<th>Network</th>
<th>Ipks</th>
<th>Ierrs</th>
<th>Ibytes</th>
<th>Opks</th>
<th>Oerrs</th>
<th>Obytes</th>
<th>Coll</th>
</tr>
</thead>
<tbody>
<tr>
<td>en0</td>
<td>1500</td>
<td>&lt;Link#4&gt;</td>
<td>1810076</td>
<td>0</td>
<td>2027413615</td>
<td>1276472</td>
<td>0</td>
<td>245011138</td>
<td>0</td>
</tr>
<tr>
<td>en0</td>
<td>1500</td>
<td>10.0.1/24</td>
<td>miniserver.extr</td>
<td>1810076</td>
<td>-</td>
<td>2027413615</td>
<td>1276472</td>
<td>-</td>
<td>245011138</td>
</tr>
<tr>
<td>en0</td>
<td>1500</td>
<td>10.1/16</td>
<td>mainserver.pret</td>
<td>1810076</td>
<td>-</td>
<td>2027413615</td>
<td>1276472</td>
<td>-</td>
<td>245011138</td>
</tr>
</tbody>
</table>

# tcpdump -s0 -v -i en0

tcpdump: listening on en0, link-type EN10MB (Ethernet), capture size 65535 bytes
13:13:08.164723 IP (tos 0x10, ttl 64, id 29534, offset 0, flags [DF], proto TCP (6), length 180) mainserver.pretendco.com.ssh > 10.1.17.5.58276: P, cksum 0xa459 (correct), 1929289080:1929289208(128) ack 4201395484 win 65535 <nop,nop,timestamp 88298191 952884239>
13:13:08.165815 IP (tos 0x10, ttl 64, id 17992, offset 0, flags [DF], proto TCP (6), length 52) 10.1.17.5.58276 > mainserver.pretendco.com.ssh: ., cksum 0xc03c (correct), ack 128 win 65535 <nop,nop,timestamp 952884320 88298191>
Network
SSH

Secure replacement for `telnet(1)` and more

- `ssh(1)`
- `scp(1)`
- `sftp(1)`
- port forwarding (tunneling)

OpenSSH implementation
**SSH Authentication**

### Password
- `ssh(1)` client sends user name
- `sshd(8)` requests password

### Key Pair
- Generate key pairs, public part needs to be installed in the user’s `~/.ssh` directory on the server
- SSH client sends user name
- `sshd` sends challenge based on public part of key pair
- SSH client creates response using private part of key pair
Helpful Built-In GUI Tools

GUI Tools for Assessment and Utilization

- Server Admin
- Server Status widget
- Disk Utility
- Activity Monitor
- System Profiler
- Web browser
- Network browser
Helpful Built-In CLI Tools

- uptime(1)
- top(1)
- ps(1)
- diskutil(8)
- df(1)
- du(1)
- iostat(8)
- networksetup(8)
- netstat(1)
- systemsetup(8)
- scutil(8)
- serveradmin(8)
- serversetup(8)
- system_profiler(8)
- launchctl(1)
- sysctl(8)
- ...

>`-`
Assess Service/Process Utilization

- **Activity Monitor**
- **ps(1)**
- **top(1)**

Processes: 113 total, 2 running, 4 stuck, 107 sleeping... 425 threads

```
PID COMMAND   %CPU  TIME   TH  PRTS  MREGS  RPRVT  RSHRD  RSIZE  VSIZE
296 LaunchCFMA 16.1% 53:48.99 3  124   699   80M    33M   117M   862M
```
Assess Hardware Utilization

Activity Monitor

- vm_stat(1)
- sysctl(8)
- top(1)

```
server17:~ admin$ vm_stat
Mach Virtual Memory Statistics: (page size of 4096 bytes)
Pages free: 8725.
Pages active: 293357.
Pages inactive: 152717.
...
Pageins: 138306.
Pageouts: 3603.
Object cache: 53134 hits of 191862 lookups (27% hit rate)
```

```
server17:~ admin$ sysctl vm.loadavg
vm.loadavg: { 0.28 0.33 0.38 }
```

```
server17:~ admin$ sysctl vm.swapusage
vm.swapusage: total = 128.00M used = 76.41M free = 51.59M (encrypted)
```
Assess Storage Utilization

- Server.app
- Disk Utility
- System Profiler
Assess Storage Utilization

- `system_profiler(8)`
- `diskutil(8)`
- `df(1)`
- `du(1)`
- `iostat(8)`

```
server17:-- localadmin$ system_profiler SPSerialATADataType
... 
Server_1:
  Capacity: 30.52 GB
  Available: 15.61 GB
  Writable: Yes
  File System: Journaled HFS+
  BSD Name: disk0s2
  Mount Point: /

Untitled 2:
  Capacity: 100.84 GB
  Available: 84.21 GB
  Writable: Yes
  File System: Journaled HFS+
  BSD Name: disk0s3
  Mount Point: /Volumes/Untitled 2
...
```

```
server17:-- localadmin$ df -h
Filesystem      Size   Used  Avail Capacity Mounted on
/dev/disk0s2    31Gi   15Gi   16Gi    49%    /
devfs          108Ki  108Ki    0Bi   100%    /dev
fdesc          1.0Ki  1.0Ki    0Bi   100%    /dev
map -hosts      0Bi    0Bi    0Bi   100%    /net
map auto_home   0Bi    0Bi    0Bi   100%    /home
map -fstab      0Bi    0Bi    0Bi   100%    /Network/Servers
/dev/disk0s3   101Gi   17Gi   84Gi    17%    /Volumes/Untitled 2
...
```

```
server17:-- localadmin$ sudo du -sh *
  0B  Shared
  6.2M  diradmin
```
Finishing up

- `logout`
  - Terminates an SSH session
- `exit`
  - Gets you out of a root shell
- `reboot`
- `halt`
  - Properly shuts down all processes
Backup Tools

- rsync(1)
- ditto(1)
- asr(8)
- Other UNIX tools
- server.app(8)
- Time Machine
- Third-Party Solutions
rsync(1)

```
/usr/bin/rsync

- Keeps backup data in sync with the original
- One-way sync
- Only copies files or blocks of files that have changed
- Can copy extended attributes, including resource forks
- Can back up via network using ssh

$ rsync -auvE ~/Documents/ServerConfig/
  localadmin@server17.local:/Backup
```
ditto(1)

/usr/bin/ditto

- Performs a complete copy of specified folders
- Copies extended attributes, ACLs, and resource forks
- Can compress to PKZip or cpio format

$ ditto /var/spool/postfix ./var/spool/postfix
asr(8)

/usr/sbin/asr—Apple Software Restore

Back up and restore an entire volume

Source needs to be unmountable

# asr verb [options]

# asr restore --source source --target target [options]
Other Useful UNIX Tools

- `scp (1)`—Secure copy data to a remote location
- `hdiutil(1)`—Disk image manipulation
- `tar(1)`—Tape archiving utility to create, extract tar archives
- `pax(1)`—Read/write archives, copy directory hierarchies
- `zip(1)`—Package and compress (archive) files
- `kdb5_util(8)`—Kerberos database maintainance utility
- `mkpassdb`—Create/modify/back up password server
- `dd(1)`—Low-level copying and conversion utility
Resources

Books - O’Reilly
- Learning Unix for Mac OS X Leopard
- Mac OS X Snow Leopard for Unix Geeks

Websites
- www.osxfaq.com
- www.macdevcenter.com

Certification Courses (ACSA)
- Directory Services, Deployment and Mobility & Security
ACL’s in the Command Line
POSIX vs. ACLs Review

drwxrwxr-x+ 33 root        admin      1122 Apr 21 13:28 Applications
0: group:everyone deny delete
1122 Apr 21 13:28 Applications

User/Group: Everyone
Permission Type: Deny

Inheritance
- Apply to this folder
- Apply to child folders
- Apply to child files
- Apply to all descendants

Cancel  OK
ACL Command-Line Tools

fsaclctl(1)—Controls access control list support

$ fsaclctl -a -e

chmod(1)—Sets access control list

$ chmod +a "w2sdesigner allow list,add_file..." Destination

ls(1)—Lists access control list

$ ls -le

   total 0
   drwxrwx---+ 2 localadmin admin  68 Apr 19 11:05 Design
   0: group:w2sdesigner allow
       list,add_file,search,delete,add_subdirectory,delete_child,re
       adat
Verifying File System ACLs

```
server14:~ ladmin$ ls -edl " /Shared Items/Public"
-.drwxrwxr-x+ 2 root admin 68 Apr 25 14:00 /Shared Items/Public
0: 94FFAC59-F743-402C-95E9-0AF75A7220D2 allow
list,add_file,search,delete,add_subdirectory,delete_child,readattr,writeattr,

server14:~ ladmin$ ls -edln " /Shared Items/Public"
-.drwxrwxr-x+ 2 0  80  68 Apr 25 14:00 /Shared Items/Public
0: 94FFAC59-F743-402C-95E9-0AF75A7220D2 allow
list,add_file,search,delete,add_subdirectory,delete_child,readattr,writeattr,
readextattr,writeextattr,readsecurity,writesecurity,chown,file_inherit,
```
Binding AD/OD from the CLI
Binding with dsconfigldap

```
sudo dsconfigldap -fv -a mainserver.pretendco.com -n mainserver.pretendco.com -c `hostname -s` -u diradmin -p <diradminpw> -l cadmin -q <localpw>
```

- `-f` force authenticated binding/unbinding
- `-v` verbose logging
- `-a` servername
- `-n` configuration name displayed in Directory Utility
- `-c` client computer id
- `-u` directory user with permission to add computers
- `-p` password
- `-l` local administrator
- `-q` local administrator password
Verifying Binding via Command Line

```
id diradmin
uid=1000(diradmin) gid=20(staff) groups=20(staff),80(admin)

id -u diradmin
1000

dscl
Entering interactive mode... (type "help" for commands)
> cd /LDAPv3/mainserver.pretendco.com/Users
/LDAPv3/mainserver.pretendco.com/Users > ls
delegate
diradmin```
Verifying Login at Command Line

- `id` and `dscl` to verify the existence of the account in the directory
- `dirt` to verify user authentication against Open Directory
- `kinit` to test Kerberos authentication
- `klist` to verify what tickets a user is granted
The Binding Process

Server Admin and Directory Utility hide a host of steps involved in binding

From the command line, binding manually involves dscl, defaults, dsconfigldap, and DirectoryService

Server Admin and Directory Utility reduce those steps to a few clicks and text fields in a GUI

1) sudo dscl /Search create /SearchPolicy CSPSearchPath
2) sudo defaults write /Library/Preferences/DirectoryService/DirectoryService LDAPv3 Active
3) sudo killall DirectoryService
4) sudo dsconfigldap -v -a server17.pretendco.com -n server17 -u diradmin -p apple -l admin -q apple
5) sudo dscl /Search append /CSPSearchPath /LDAPv3/server17.pretendco.com
Joining Kerberos at the CLI

```
sudo /usr/sbin/sso_util configure
  -r SERVER17.PRETENDCO.COM -a diradmin -p apple
  -f /LDAPv3/server17.pretendco.com -v 1 all
```
dscl

directory services command-line utility

Apple-supported command-line tool for managing DS data

Directory structure appears like file-system structure

Single-user mode

Local and external nodes
Binding with dsconfigad

1. Enable AD plug-in with defaults command

2. Bind with dsconfigad

   dsconfigad -a computername \\ndomain pretendco.com \\n-u Administrator -p ADadminpw \\n-lu cadmin -lp cadmin

3. Confirm binding with dsconfigad -show

4. Add Active Directory to search path with dscl
dsconfigad Specific Options

- `packetsign` <disable | allow | require>

- `packetencrpyt` <disable | allow | require>

- `namespace` <forest | domain>
  - For username conflicts across domains

- `passinterval` <days>
  - How often to automatically change the computer password
Questions

russell@creativeresources.net