

# Command-line for the Uninitiated

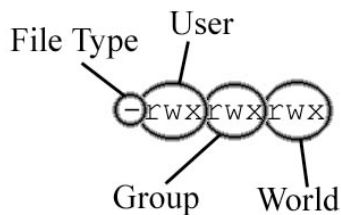
## Getting around

### Special Characters

For those who have always used the Mac, the concept of special characters in filenames is different, meaning you never had to pay attention to what was or wasn't a special character. The DOS/Windows world had strict rules, which are getting lighter, and the UNIX world has always accepted them, albeit in a special way. So here's a little discussion surrounding the use of special characters.

- \ (backslash) is the most special character of all. When placed in front of another special character, it means "treat this literally, not as a special character." This is frequently referred to as escaping a character.
- \* and ? are wildcards, meaning they can be used in place of other characters. \* means "0 or more characters," while ? means "one character."
- A space is special in that it needs to be escaped in order to be recognized properly when used as part of a command. Some commands, have trouble with spaces in filenames and require quotes instead of the backslash. And still others may allow either.

### UNIX Permissions



(from [http://www.acm.uiuc.edu/webmonkeys/html\\_workshop/unix.html](http://www.acm.uiuc.edu/webmonkeys/html_workshop/unix.html))

UNIX file permissions are centered around three groups of people, the user, group, and world (often referred to as other). This is not unlike the traditional Mac permissions.

Each file contains three pieces of permission information, whether the item is readable, writable, and executable by each of those sets of users. Directories must be executable in most situations to work properly. Other than that, only executable files need be executable.

The file type is also useful because it can tell you which items are files, directories, or special cases.

## Important Files and Directories

Hidden files are any filename that starts with a “.” (dot). They may be displayed with the `-a` flag in the `ls` command. The system GUI also hides other files when the “hidden” flag is marked in the resource fork. This applies to some fairly significant historical UNIX directories such as `/etc` and `/var`. Following is a list of directories and files you might find useful.

<code>.</code>	the current location
<code>~</code>	your home directory
<code>~username</code>	username’s home directory
<code>/etc</code>	location of UNIX configuration files
<code>/var</code>	location of UNIX “variable space” information—databases, logs, mail and printing spools, etc.
<code>/var/db</code>	location of open directory and netinfo databases, among others
<code>/var/log</code> <code>/Library/Logs</code> <code>~/Library/Logs</code>	typical locations of logs on OS X systems
<code>/Volumes</code>	All mounted drives will show up here
<code>/Network/Servers</code>	All mounted servers will show up here

## Command-Line Examples

ls -l	long list (details) of files
ls -a	list all files, including hidden (dot) files
ls -ltr	long list, sorted by time, in reverse
shutdown -r 23:59	shutdown and reboot at 11:59 p.m. today
cat file	display the contents of the file on the screen
less file	display the contents of the file on the screen one page at a time, with search capabilities (see the man page)
head file	display the top 10 lines of file
tail -20 file	display the bottom 20 lines of file
tail -f file	display the bottom 10 lines of file, and display new lines as they are added
zcat file   less	display the text contents of a gzipped file (e.g., old logs), then send the output to less so they will display one page at a time
grep searchstring file	search for “searchstring” in the text of file
last   grep username	run last and send it to grep (instead of standard output—the screen), then grep for occurrences of “username” and send only that to the screen
mv dirname /new/directory	move dirname to /new/directory (10.4 only recommended)
cp -pR dirname /new/directory	copy, keeping current permissions, all of the contents of dirname to /new/directory (10.4 only recommended)
rm filename	remove filename
rm -r dirname	remove dirname and all files and subdirectories inside
find / -name filename	find, beginning at the root level, all files with the name “filename”
find . -name *.app	find all files ending in “.app” from the current directory and all subdirectories
sudo chmod -R ugo+rw dirname	as the superuser, add read and write privileges to all files inside dirname for the user (owner), group, and other (everyone)
ditto -rsrc dirname /new/dirname	copy, with HFS formatting, from dirname to /new/dirname (useful for backup)
ditto -c -k -rsrc dirname archive.zip	copy, with HFS formatting, from dirname to the PKZIP formatted archive.zip archive file. Suitable for storing backup on PC servers.

## **Useful Server Commands/Tools**

<code>du -sh *   sort -nr &gt; outputfile</code>	Get disk usage on all files and subdirectories, sort numerically in reverse (biggest results on top), and save it to outputfile
<code>df -h</code>	display disk usage of all disks on the system
<code>repquota -a</code>	report quota information
<code>dscl localhost</code>	enter directory service command line mode
<code>ifconfig -a</code>	display interface information
<code>netstat -rn</code>	display the server routing table
<code>chown -R user:groupname dirname</code>	change the owner of dirname and all enclosed files to user and the group to groupname
<code>sudo chmod -R ugo+rw dirname</code>	as the superuser, add read and write privileges to all files inside dirname for the user (owner), group, and other (everyone)
<code>find 20* -name *.app -or \ -name *.dmg</code>	find, in all subdirectories of the current location starting with 20, files with the name ending in .app or ending in .dmg
<code>chmod -R o-rwx *</code>	remove “other” permissions from all files and subdirectories from the current location. Useful to remove permissions from student subdirectories in the absence of Passenger.

## Scripts and Scripting

First, if you want to write your own, buy a good book or do some on-line reading. Second, if you use someone else's, take care to understand the commands before using. Here's some examples that should be straightforward to people with a little bit of programming background.

Even if these scripts don't do what you want, you may be able to find a really useful line or two inside.

### Backup

```
#!/bin/bash
# A script to do backups and take a parameter for the destination
directory
# Written 2005-03-23 gdn
if [ -z $1 ]
then
    echo "Usage: `basename $0` backup-destination-directory"
    exit 1
fi

# optionally remove the destination directory files prior to new backup
# ditto does not erase old files--removal ensures a clean backup
# comment out the rm command if not desired
rm -rf $1/*

# edit the subdirectory names in the DIRNAME list for your server
# also edit the volume path in the ditto statement below
for DIRNAME in Apps 2009 2010 2011 staff
do
    /usr/bin/ditto -rsrc /Volumes/Data/$DIRNAME $1/$DIRNAME
done
```

### Cleanup

```
#!/bin/sh
# a script to do some maintenance on homes, then run a usage report
# created 2005-03-23 gdn

USERBASE=/Volumes/Data
DESTDIR=/Users/district/scripts/diskuse

for DIRNAME in 2009 2010 2011 staff
do
    find $USERBASE/$DIRNAME -name Caches -exec rm -r {}/Safari \
        {}/Firefox {}/MS\ Internet\ Cache \;
    find $USERBASE/$DIRNAME -name *.dmg -or -name REGED*---- -exec \
        rm {} \;
done

# Sum and sort disk use of students and staff, then put each in a file
du -sk $USERBASE/20*/ * | sort -nr > \
    $DESTDIR/diskuse-`date +%Y%m%d`.txt
du -sk $USERBASE/staff/* | sort -nr > \
    $DESTDIR/diskuse-staff-`date +%Y%m%d`.txt
```

## Automating with Cron

Cron is a great little utility that allows you as the root user to schedule events regularly. This can be anything from a weekly or monthly reboot to a fancy disk maintenance or other script. Here is an example crontab:

```
0 2 * * 0 /sbin/shutdown -r now
0 22 * * 1-5 /Users/adminuser/scripts/backup.sh /Volumes/Backup/Daily
0 22 * * 6 /Users/adminuser/scripts/backup.sh /Volumes/Backup/Weekly
0 20 * * 0 /Users/adminuser/scripts/diskmaint.sh
```

There are many good web sites, and also the man page, to explain how to read crontabs and create your own, so I won't try to do it here, other than to list what the columns mean. From left to right, they are minutes, hours, days of the month, month of the year, and day of the week, followed by the command to execute.

If you are looking for something more on the GUI level to help you with cron, look into Cronnix at <http://www.abstrature.de/cronnix>.

## Digging Deeper

If you want more on the command line, scripting, etc., there are tons of good resources out there. Some are listed in the ARD piece, but here are some of my other favorites:

**Apple lists:** Sign up at <http://lists.apple.com/client-management@lists.apple.com>  
[macos-x-server@lists.apple.com](mailto:macos-x-server@lists.apple.com)

### *Unix Power Tools, 3rd Edition*

By Shelley Powers, Jerry Peek, Tim O'Reilly, Mike Loukides  
3rd Edition October 2002  
ISBN: 0-596-00330-7  
1156 pages, \$69.95 US  
<http://www.oreilly.com/catalog/upt3/>

(I don't have this one, but expect it would be useful.)

### *Learning the bash Shell, 3rd Edition*

By Cameron Newham  
3rd Edition March 2005  
ISBN: 0-596-00965-8  
352 pages, \$34.95 US  
<http://www.oreilly.com/catalog/bash3/>

## **Apple Remote Desktop**

ARD is extraordinarily useful anyway, but ARD 2 includes the “Send UNIX command” option. In so doing, it unleashed an entirely new set of people on the lab management environment. This session has no time to discuss ARD, but here are some useful things you can do with or to your labs and some fancy UNIX commands.

There are some really good resources out there for Apple Remote Desktop, so I don’t want to duplicate all of that effort. However, I’ll put some of the fun and useful ones here.

## **MacEnterprise**

MacEnterprise.org is the higher ed user group for the really techie Mac stuff. They have good resources on a lot of things, including Apple Remote Desktop. There is at least one webcast on ARD archived there. The Scripts & Tools page (<http://macenterprise.org/content/blogsection/10/129/>) has some good stuff, and they even have a page dedicated to types of things you can do with ARD and the Send UNIX Command button (<http://macenterprise.org/content/blogcategory/117/130/>).

## **Kickstart**

Kickstart is a utility built into the ARD client to start the system from the command line. It’s not overly useful on clients with ssh disabled, but allows you to restart the agent in the case there is a problem, and do some remote stuff on the server.

Kickstart for ARD 1.2: <http://docs.info.apple.com/article.html?artnum=107837>

Kickstart for ARD 2: <http://docs.info.apple.com/article.html?artnum=108030>

## **Automated Lab Login**

Thanks to John Detroye and the [remote-desktop@lists.apple.com](mailto:remote-desktop@lists.apple.com) list for this one. If you really want to get into ARD, joining this list is a must.

```
osascript <<EOF
set logInUser to "localuser" -- or whoever you want to log in as
set logInPassword to "whatever"

tell application "System Events"
    tell application process "loginwindow"
        keystroke logInUser
        delay 1
        keystroke tab
        delay 1
        repeat with aChar in characters of logInPassword
            keystroke aChar
            delay 0.2
        end repeat
        keystroke return
    end tell
end tell
EOF
```