

Essential Scripts

Five Lines of Code That Will Change Your Life
Macworld San Francisco 2005
Session M252

Arek Dreyer
Dreyer Network Consultants, Inc
arek@arekdreyer.com
Apple Certified Trainer ACSA 10.3

Apple Certifications for 10.3.x

Course Names	Certification Level	Length (days)
Mac OS X Help Desk Essentials	ACHDS	3
Mac OS X Server Essentials System	ACTC (with Help Desk Essentials)	4
Administration of Mac OS X Clients System Administration using Mac OS X Server	Apple Certified System Administrator	5 days each course

train.apple.com • 800-848-6398

Objectives

- After this session, you'll be able to:
 - Compose your own shell scripts
 - Use the cron system scheduler

Intended Audience

- Relatively new UNIX user

Rules of Thumb

- Invest time now, be rewarded later
- Use appropriate tool for the task
- Test scripts to avoid DISASTER

Topics

- Shell scripts:
 - Who, What, When, Why, and How
- Cron, Cronnix, and reporting
- Various scripts to illustrate

Shell Scripts

- Who
- What
- When
- Why
- How

Who writes scripts?

- Someone not afraid of Terminal
- You, after this session!

What is a shell script?

- Shell - interface to the OS
 - Finder is another interface to OS
- Script - collection of commands

When to write a shell script

- When you need to solve a problem
- Practice

Why write a shell script

- Depends on limiting resource
- If limited by execute time, use more complicated and faster language
- If limited by human time, use shell script

How to write a shell script

- Create a file with some text editor
- Enter commands just like you would at the Terminal
- Enter comments for goals, methodology
- Save file, ending with `.sh` or `.command`
- `chmod u+x MyScript.command`

Scripting techniques

- Decisions: if then
- Variables: \$UID, \$file, \$0
- Looping: for, while, until
- Wildcarding: *, ?
- External command: `ls` or \$(ls)
- Piping: ps aux | grep iTunes

Decisions: if/then/else/

fi

• `if [$UID -eq 0]`

- then

- echo "Running as Root"

- else

- echo "Not Root"

- fi

Variables

- `echo "My filename is ${0}"`
- `echo "Running script as ${USER}"`
- `Fontdir="${HOME}/Library/Fonts"`
- `Counter=$((${Counter} + 1)`
- `du -sh $HOME > /tmp/out.$$`

Looping

- for i in imac emacs xserve
- do
 - echo “Do something with $\{i\}$ ”
- done

Wildcarding

- * matches ZERO or ALL
- ? matches one character
- [a-z] matches a range

External Commands: simple

- Enter it like you would in the shell
- `/usr/sbin/screencapture -x \`
 - `MyScreenShot.pdf`

External Commands: a little more complicated

- Use `$()` or backticks ```
- `myuse=$(/usr/bin/du -sh $HOME)`
- *or*
- `myuse=`/usr/bin/du -sh $HOME``

Piping & Redirecting

- Pipe - output of one program goes to the input of another program
- Redirect - output/input to/from a file

Utilities for piping

- cut
 - cut specific characters in column: -c
 - cut specific fields in a line: -d
 - change field delimited from colon: -f
- grep
- sort

Piping and Redirecting

- `day=$(/bin/date | cut -d" " -f1)`
- `system_profiler | grep "Serial" \`
 - `| cut -c31-80`
- `echo "Good Stuff" >> $GoodNewsFile`

Scripting Templates

- `#!/bin/sh`
- `#` Info about the script
- Common looping - `dscl list Users`
- Common cleanup - `rm /tmp/log.$$`

Make script executable

- `chmod u+x somescript.sh`
- `chmod +x somescript.command`
- call the script with its path
 - `sh /path/to/somescript.sh`

Cron and Cronnix

- Cron: the built-in internal scheduler
 - Who - which user to run the script as
 - What - what script
 - When - when to run it
 - Where - where does the script live
 - Why - because you can!

Cron via Terminal:

`crontab -e`

- Uses default text editor `$EDITOR`
- Strict format
- Easy to make mistakes
- Avoid if you're a beginner

Cron via GUI: Cronnix

- Download and install
- Easy to learn and use
- Helps you learn `crontab -e`

Cron with Reporting

- Pipe the output to mail
- Redirect the output to a text file
- Deal with errors (or not)

Pipe output to mail

- `/bin/sh /Users/arek/script.sh \`
- `| mail -s "A Report" \`
 - `arekd@mac.com`

Redirect output to file

- `/bin/sh /Users/arek/script.sh >> \`
- `/Users/arek/logs/diskuse.log`

Deal with Errors

- `script.sh > /dev/null 2>&1`

Scripts to illustrate

- Really simple scripts- just commands
- Simple scripts to be run with cron
- Scripts with variables and decisions
- Scripts with loops
- Some complicated scripts

Really simple scripts

- Just a collection of commands
- Just like you would type in Terminal
- Saves you from typing over and over

Simple Script

- `#!/bin/sh`
- `# Open 3 URLs in default brwsr`
- `open http://www.mediamatters.org`
- `open http://www.foundmagazine.com`
- `open http://www.afp548.com`

Simple Script

- `#!/bin/sh`
- `# Mount two AFP Volumes`
- `/sbin/mount_afp afp://username:
pwd@server.local/Art /Volumes/Art`
- `/sbin/mount_afp afp://server.local/Ads /
Volume/Ads`

Simple Script

- `#!/bin/sh`
- `# Must be run as root`
- `# Repair perms then reboot`
- `/usr/sbin/diskutil \ repairPermissions /`
- `reboot`

Simple script with cron

- `#!/bin/sh`
- `# Show the disk use for 2 volumes`
- `# Use with cron to mail to admin`
- `echo "Latest Disk Space Report"`
- `df -kh /Volumes/Art /Volumes/Ads`

cron entry for previous script

- `0 * * * * /Users/arek/duse.sh | mail -s "Disk Use Report" arek`

Script with variables and decisions

- ```
#!/bin/sh
Take a screenshot if some is logged in
The -x flag for screencapture means "no click"
The [-n] is a test to see if variable is not empty
TheUser=$(/usr/bin/who | /usr/bin/grep console \
 | /usr/bin/cut -f1 -d" ")
if [-n "${TheUser}"];
then
 TheDate=$(/bin/date "+%Y%m%d-%H%M%S")
 TheFileName="${TheDate}-${TheUser}.pdf"
 /usr/sbin/screencapture -x /tmp/${TheFileName}
fi
```
-

# Decision

- `if [ -n "${TheUser}" ];`

# Script with loops: Process PDFs

```
• #!/bin/sh
Arek Dreyer
Fri Dec 10 19:35:30 MST 2004
ProcessPDF.command
Description:
If there are PDF files in folder, mail me information about each
then move it to another folder.
WatchedFolder=/Users/Shared
AnotherFolder=/Users/Shared/Notified
if [-f ${WatchedFolder}/*.pdf];
then
 for file in $(ls ${WatchedFolder}/*.pdf); do
 echo "There is a new file ${file}" | \
 mail -s ${file} arekdreyer
 mv ${file} ${AnotherFolder}
 done
fi
```

# Process PDF loop detail

- ```
for file in $( ls ${WatchedFolder}/*.pdf ); do
  echo "There is a new file ${file}" | \
  mail -s ${file} arekdreyer
  mv ${file} ${AnotherFolder}
done
```

Some complicated scripts

- Use these for inspiration
- Lots of testing
- I will explain and pull out highlights

report-diskuse.sh

```
# with cron: report-diskuse.sh | sort -r -n | mail -f from-  
address to-address  
# Find all user home Directories, report on space  
for user in $(/usr/bin/dscl localhost \  
    -list Search/Users) ; do  
    userdir=$(/usr/bin/dscl localhost \  
        -read /Search/Users/$user | \  
        grep NFSHomeDirectory | \  
        /usr/bin/cut -d" " -f2)  
    if [ -d $userdir ];  
        then  
            echo "$(du -s ${userdir}): ${user}"  
        fi  
done
```

report-diskuse.sh

- # Use with cron like this:
15 2 * * * report-diskuse.sh | sort -r -n |
mail -f from-address to-address

report-diskuse.sh detail

- for user in \
- `$(/usr/bin/dscl localhost \`
`-list Search/Users) ; do`

report-diskuse.sh

detail

- `userdir=$(/usr/bin/dscl localhost \
-read /Search/Users/$user | \
grep NFSHomeDirectory | \
/usr/bin/cut -d" " -f2)`

Mail-backup.sh

```
# Creates a disk image of the user's
~Library/Mail folder on the Desktop
#
if [ -d ${HOME}/Library/Mail -a -d
${HOME}/Desktop ]; then
    hdiutil create -fs HFS+ -volname
MailBackup.$$ -srcfolder ${HOME}/
Library/Mail
    ${HOME}/Desktop/MailBackup.$$ .dmg
    -attach
```

Mail-backup.sh

- `if [-d ${HOME}/Library/Mail -a -d ${HOME}/Desktop]; then`

Mail-backup.sh

- `hdiutil create -fs HFS+ -volname MailBackup.$$ -srcfolder ${HOME}/Library/Mail ${HOME}/Desktop/MailBackup.$$.dmg -attach`

Mail-backup.sh

- else
- echo "\$0: One or both of \${HOME}/Library/Mail \${HOME}/Desktop do not exist"
- fi

Mail-backup.sh

```
# Creates a disk image of the user's
~Library/Mail folder on the Desktop
#
if [ -d ${HOME}/Library/Mail -a -d
${HOME}/Desktop ]; then
    hdiutil create -fs HFS+ -volname
MailBackup.$$ -srcfolder ${HOME}/
Library/Mail
    ${HOME}/Desktop/MailBackup.$$ .dmg
    -attach
```

References

- Other People's scripts
- man pages
- UNIX Shell Programming (SAMS)
- UNIX Power Tools (O'Reilly)
- Wicked Cool Shell Scripts
(No Starch Press)
- Web sites out there

Essential Scripts: Summary

- Shell Script Defined
- Scripting techniques
- Using scripts with cron
- Go out there and start scripting!

Thank You!

Essential Scripts

Five Lines of Code That Will Change Your Life
Macworld San Francisco 2005
Session M252

Arek Dreyer
Dreyer Network Consultants, Inc
arekd@mac.com
Apple Certified Trainer ACSA 10.3