Think Different about Windows on your Mac (or off)

Rob Roy
Macintosh Technologist
Sheridan College
July 11, 2018

Sheridan College

Faculty of Animation, Arts and Design
Largest Arts School in Canada
Around 1300 Macs

Agenda

- Bootcamp via VM
- HP RGS - remote Windows computing
- Other tools and tips

Bootcamp via VM

How to standardize your Windows image

By the numbers

- How many machines running Bootcamp?
- Bachelor of Interaction Design - 173
- Other lab based programs - 45
- Laptop based programs - 300

The Problem:

6 Windows Images

- 2012 Mac Pro
- 2013 Mac Pro
- 2011 27” iMac
- 2012 27” iMac
- 2017 21.5” iMac 4K
- 2017 27” iMac 5K

Issues:

- Custom Mac drivers for each hardware type
- Frequent backups needed in case of issues with setup
- Deployment was inconsistent between Macs
  - Sometimes Winclone (& munki), sometimes Deploy Studio
- 4K block sizes on new post 2015 Macs

There has to be a better way

Brainstorm…

- How to make it generic for multiple Macs
- How to get back to a known good version if setup goes bad
- How not to have one of each machine cluttering up my desk
- How to make deployment super reliable

Create a Virtual Machine

- One base image for all machine types
- The stock drivers work on boot
- Snapshots allow easy roll back if there are setup issues
- Roll back from Syspreps

VMware Fusion

- This has been tested with VMWare fusion
- Not sure about other options
- HappyMac has a write up about how to setup the physical disk for VM

VM Setup

2012 Mac Pro with 2 drives
1 for OS and VMware
1 for VM creation drive

VIRTUAL MACHINES

Windows 10 x64

Windows 10 x64

Windows 10 x64

Add a note here

4 Processor Cores

4096 MB Memory

Hard Disks 931.5 GB Snapshots 4.1 GB Reclaimable

```
sudo /Applications/VMware\ Fusion.app/
Contents/Library/vmware-rawdiskCreator
create /dev/disk0 fullDevice ~/Documents/
Virtual\ Machines.localized/Windows\ 10\
x64.vmwarevm/internal-disk ide
```

```
sudo /Applications/VMware\ Fusion.app/
Contents/Library/vmware-rawdiskCreator
create /dev/disk0 fullDevice ~/Documents/
Virtual\ Machines.localized/Windows\ 10\
x64.vmwarevm/internal-disk ide
```

```
sudo /Applications/VMware\ Fusion.app/
Contents/Library/vmware-rawdiskCreator
create /dev/disk0 fullDevice ~/Documents/
Virtual\ Machines.localized/Windows\ 10\
x64.vmwarevm/internal-disk ide
```

```
sudo /Applications/VMware\ Fusion.app/
Contents/Library/vmware-rawdiskCreator
create /dev/disk0 fullDevice ~/Documents/
Virtual\ Machines.localized/Windows\ 10\
x64.vmwarevm/internal-disk ide
```

```
sudo /Applications/VMware\ Fusion.app/
Contents/Library/vmware-rawdiskCreator
create /dev/disk0 fullDevice ~/Documents/
Virtual\ Machines.localized/Windows\ 10\
x64.vmwarevm/internal-disk ide
```

```
sudo /Applications/VMware\ Fusion.app/
Contents/Library/vmware-rawdiskCreator
create /dev/disk0 fullDevice ~/Documents/
Virtual\ Machines.localized/Windows\ 10\
x64.vmwarevm/internal-disk ide
```

```
sudo /Applications/VMware\ Fusion.app/
Contents/Library/vmware-rawdiskCreator
create /dev/disk0 fullDevice ~/Documents/
Virtual\ Machines.localized/Windows\ 10\
x64.vmwarevm/internal-disk ide
```

```
sudo /Applications/VMware\ Fusion.app/
Contents/Library/vmware-rawdiskCreator
create /dev/disk0 fullDevice ~/Documents/
Virtual\ Machines.localized/Windows\ 10\
x64.vmwarevm/internal-disk ide
```

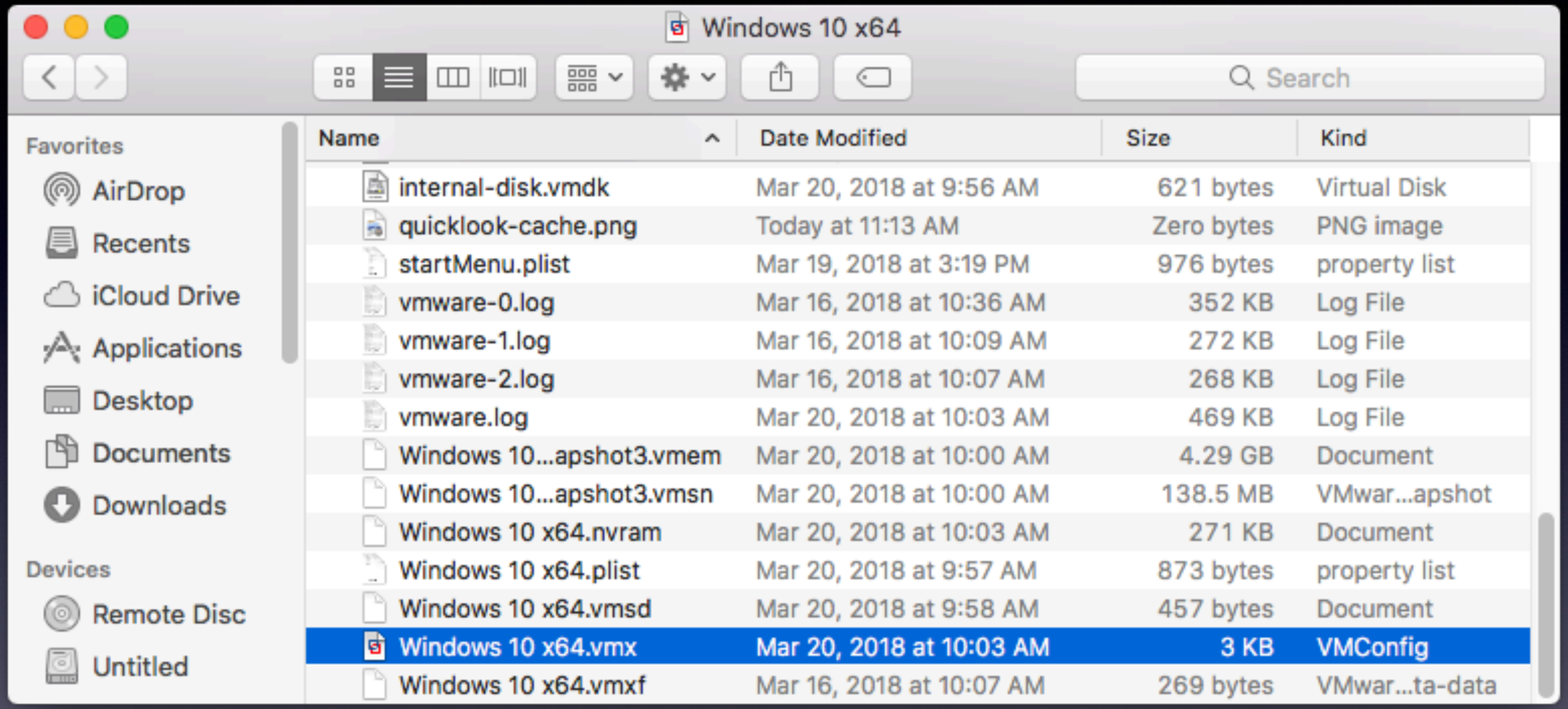
```
sudo /Applications/VMware\ Fusion.app/
Contents/Library/vmware-rawdiskCreator
create /dev/disk0 fullDevice ~/Documents/
Virtual\ Machines.localized/Windows\ 10\
x64.vmwarevm/internal-disk ide
```

```
sudo chown -R admin:staff ~/Documents/
Virtual\ Machines.localized/Windows\ 10\
x64.vmwarevm/internal-disk.vmdk
```

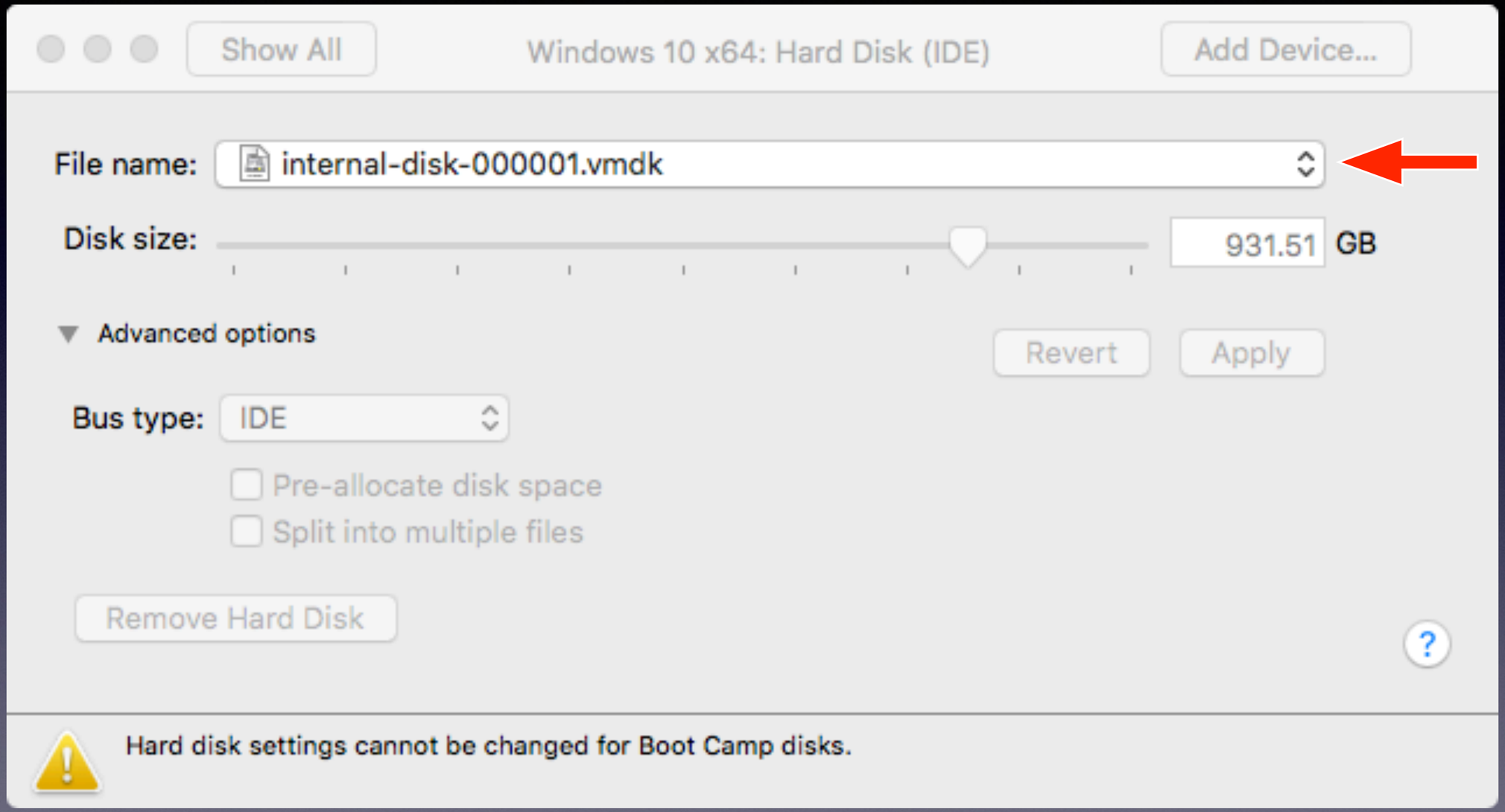
```
sudo chown -R admin:staff ~/Documents/
Virtual\ Machines.localized/Windows\ 10\
x64.vmwarevm/internal-disk.vmdk
```

```
sudo chmod -R 755 ~/Documents/Virtual\
Machines.localized/Windows\ 10\
x64.vmwarevm/internal-disk.vmdk
```

```
sudo chmod -R 755 ~/Documents/Virtual\
Machines.localized/Windows\ 10\
x64.vmwarevm/internal-disk.vmdk
```


Windows 10 x64
Search
Favorites
AirDrop
Recents
iCloud Drive
Applications
Desktop
Documents
Downloads
Devices
Remote Disc
Untitled
Name
Date Modified
Size
Kind
internal-disk.vmdk
Mar 20, 2018 at 9:56 AM
621 bytes
Virtual Disk
quicklook-cache.png
Today at 11:13 AM
Zero bytes
PNG image
startMenu.plist
Mar 19, 2018 at 3:19 PM
976 bytes
property list
vmware-0.log
Mar 16, 2018 at 10:36 AM
352 KB
Log File
vmware-1.log
Mar 16, 2018 at 10:09 AM
272 KB
Log File
vmware-2.log
Mar 16, 2018 at 10:07 AM
268 KB
Log File
vmware.log
Mar 20, 2018 at 10:03 AM
469 KB
Log File
Windows 10...apshot3.vmem
Mar 20, 2018 at 10:00 AM
4.29 GB
Document
Windows 10...apshot3.vmsn
Mar 20, 2018 at 10:00 AM
138.5 MB
VMwar...apshot
Windows 10 x64.nvram
Mar 20, 2018 at 10:03 AM
271 KB
Document
Windows 10 x64.plist
Mar 20, 2018 at 9:57 AM
873 bytes
property list
Windows 10 x64.vmsd
Mar 20, 2018 at 9:58 AM
457 bytes
Document
Windows 10 x64.vmx
Mar 20, 2018 at 10:03 AM
3 KB
VMConfig
Windows 10 x64.vmxf
Mar 16, 2018 at 10:07 AM
269 bytes
VMwar...ta-data

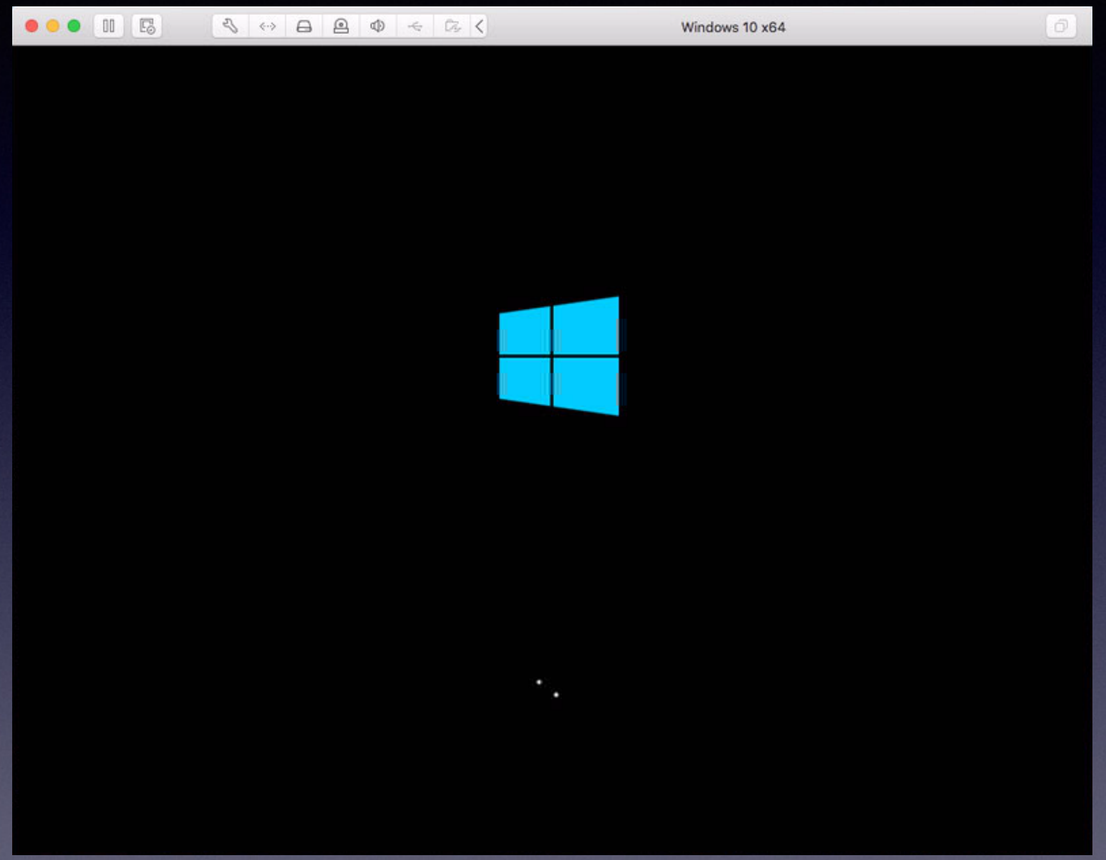
```
ide0:0.present = "TRUE"
ide0:0.filename = "internal-disk.vmdk"
ide0:0.redo = ""
```


Show All
Windows 10 x64: Hard Disk (IDE)
Add Device...
File name:
internal-disk-000001.vmdk
Disk size:
931.51
GB
Advanced options
Revert
Apply
Bus type:
IDE
Pre-allocate disk space
Split into multiple files
Remove Hard Disk
?
Hard disk settings cannot be changed for Boot Camp disks.

Install Windows

Do not install the VM drivers

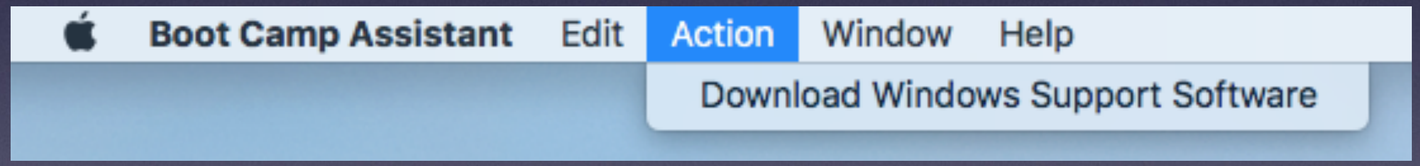
Install applications and snapshot often



Mac Drivers

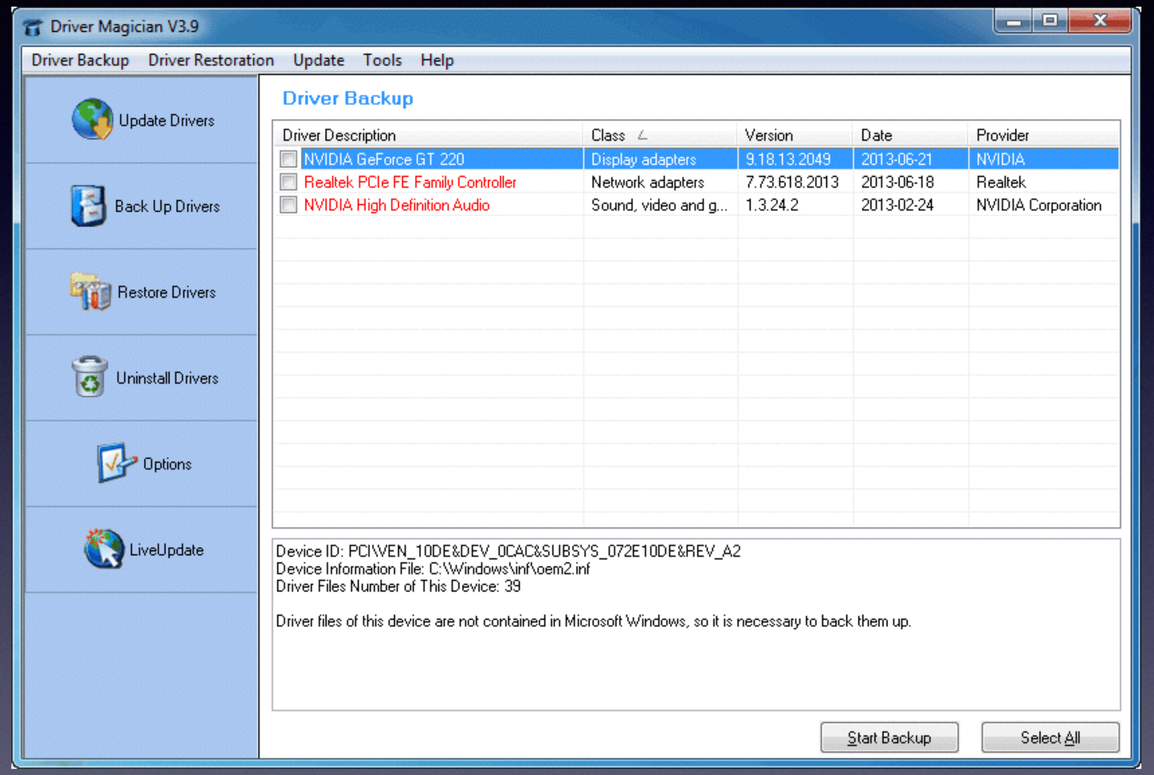
- Create an EXFAT partition for local storage

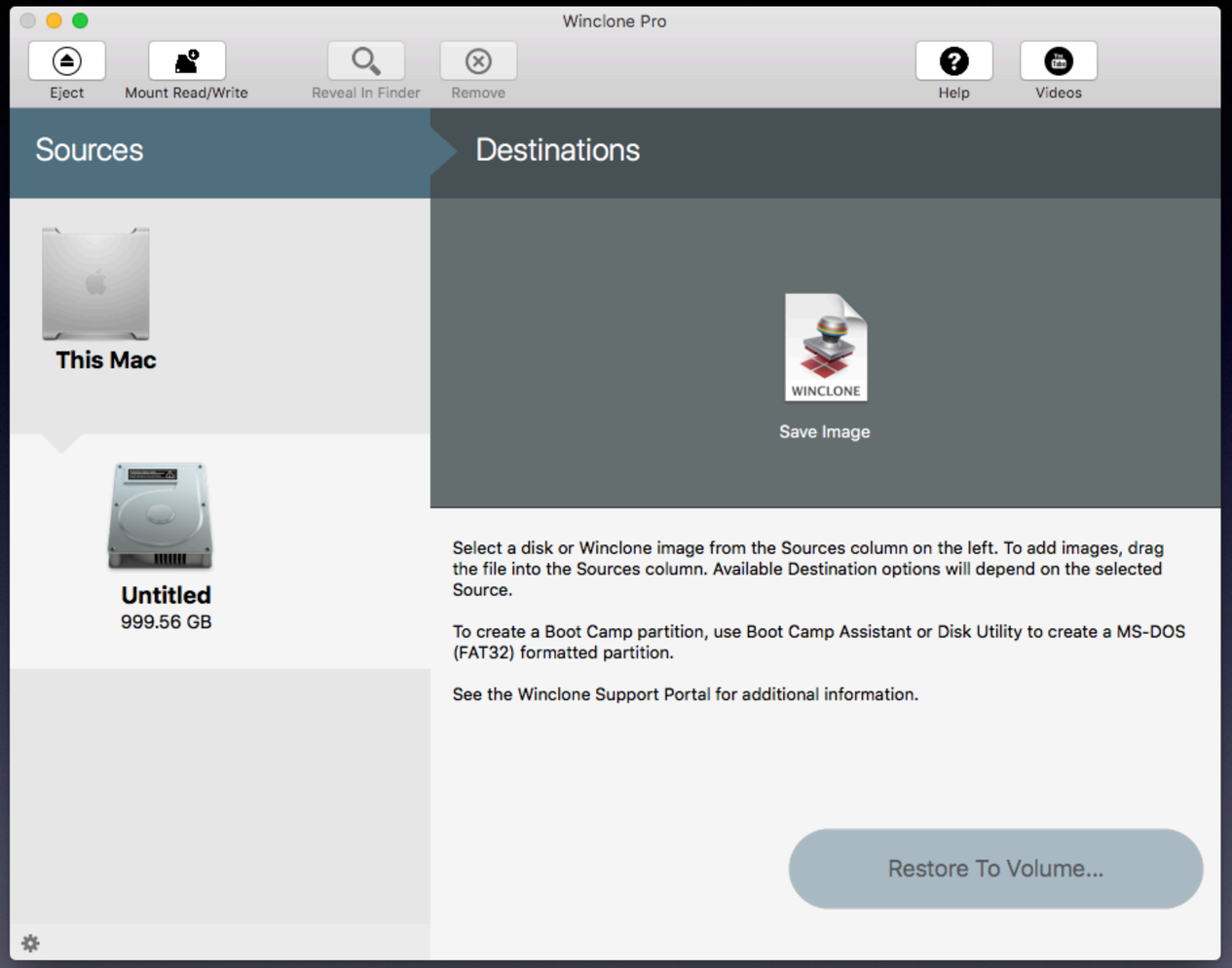
- Download the drivers for that machine

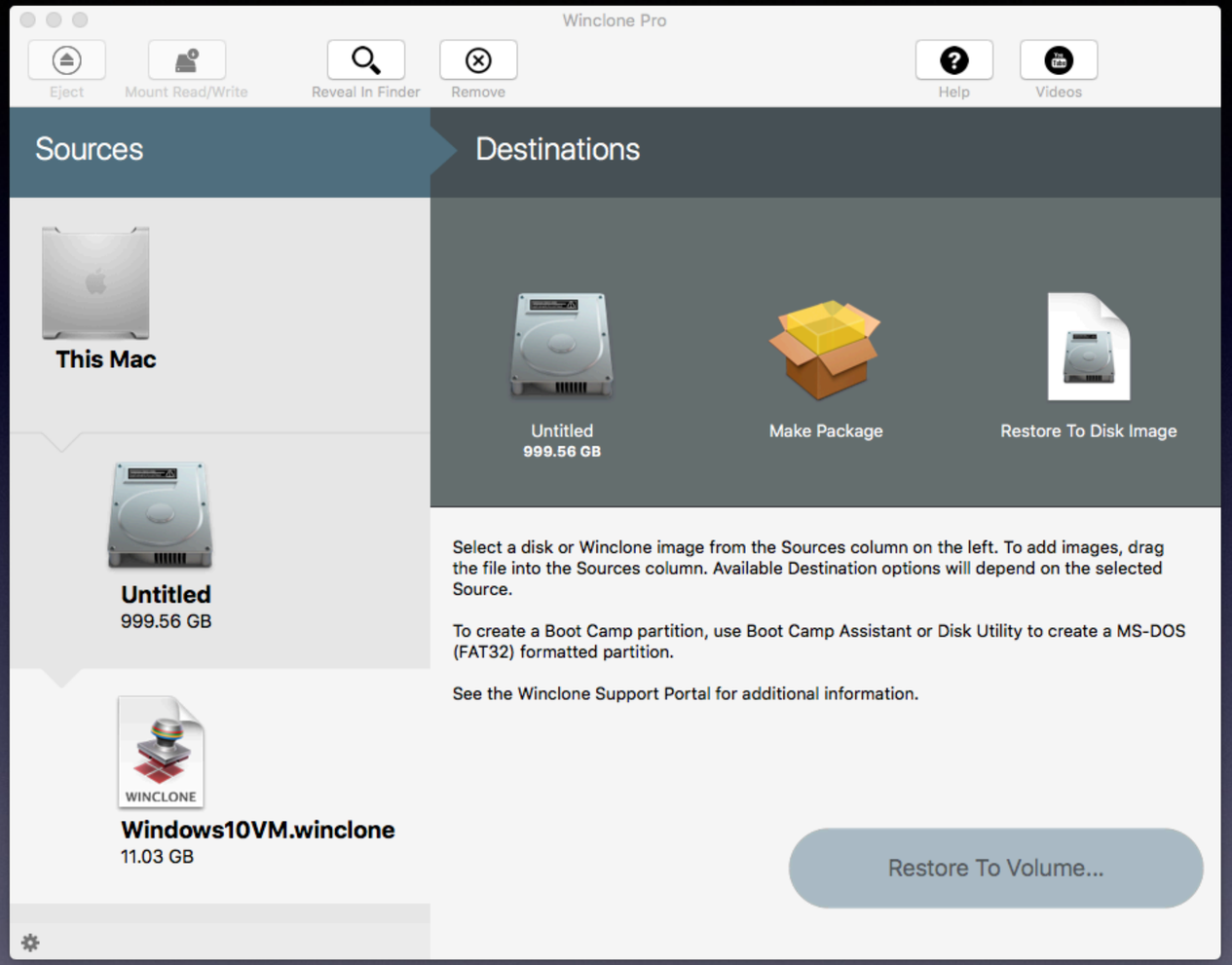


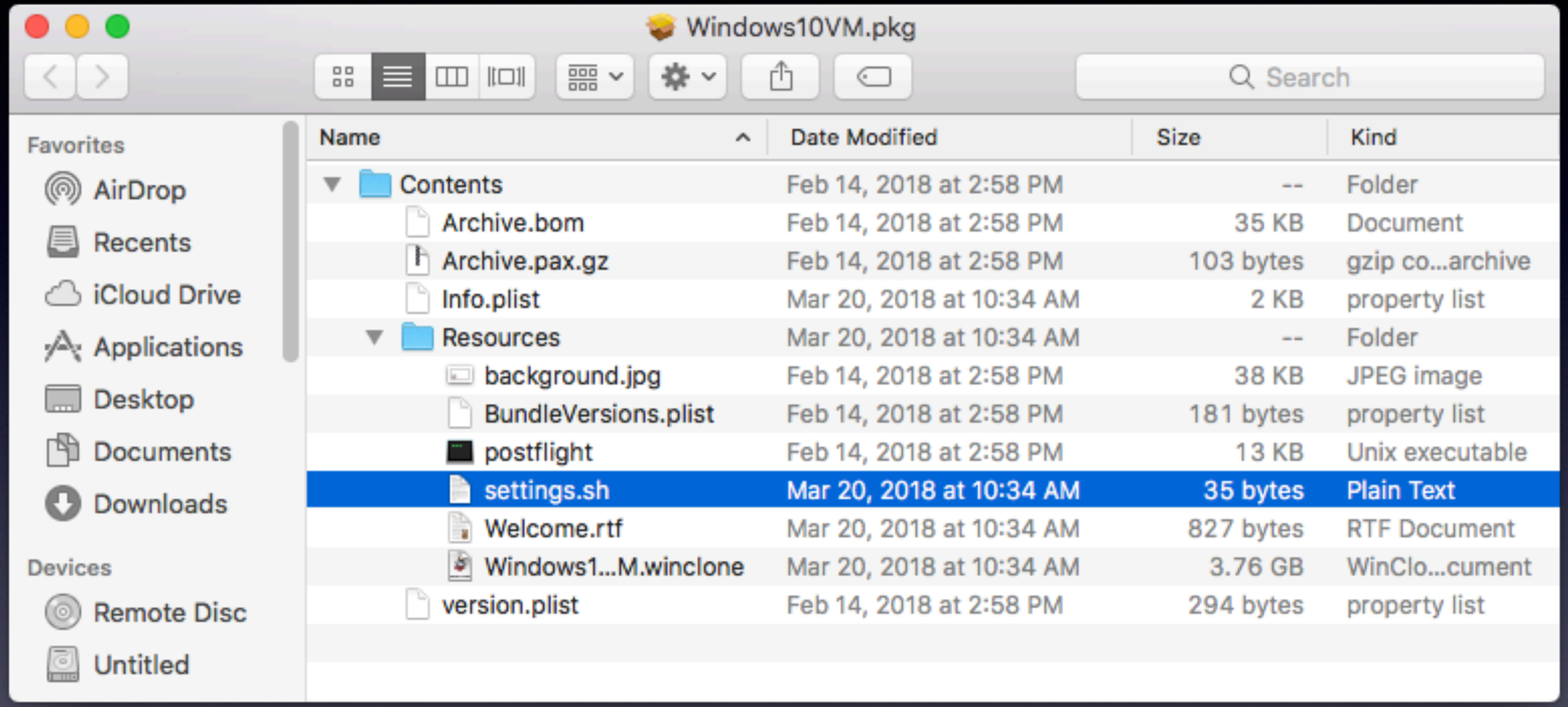
- Run the package during Windows setup scripting

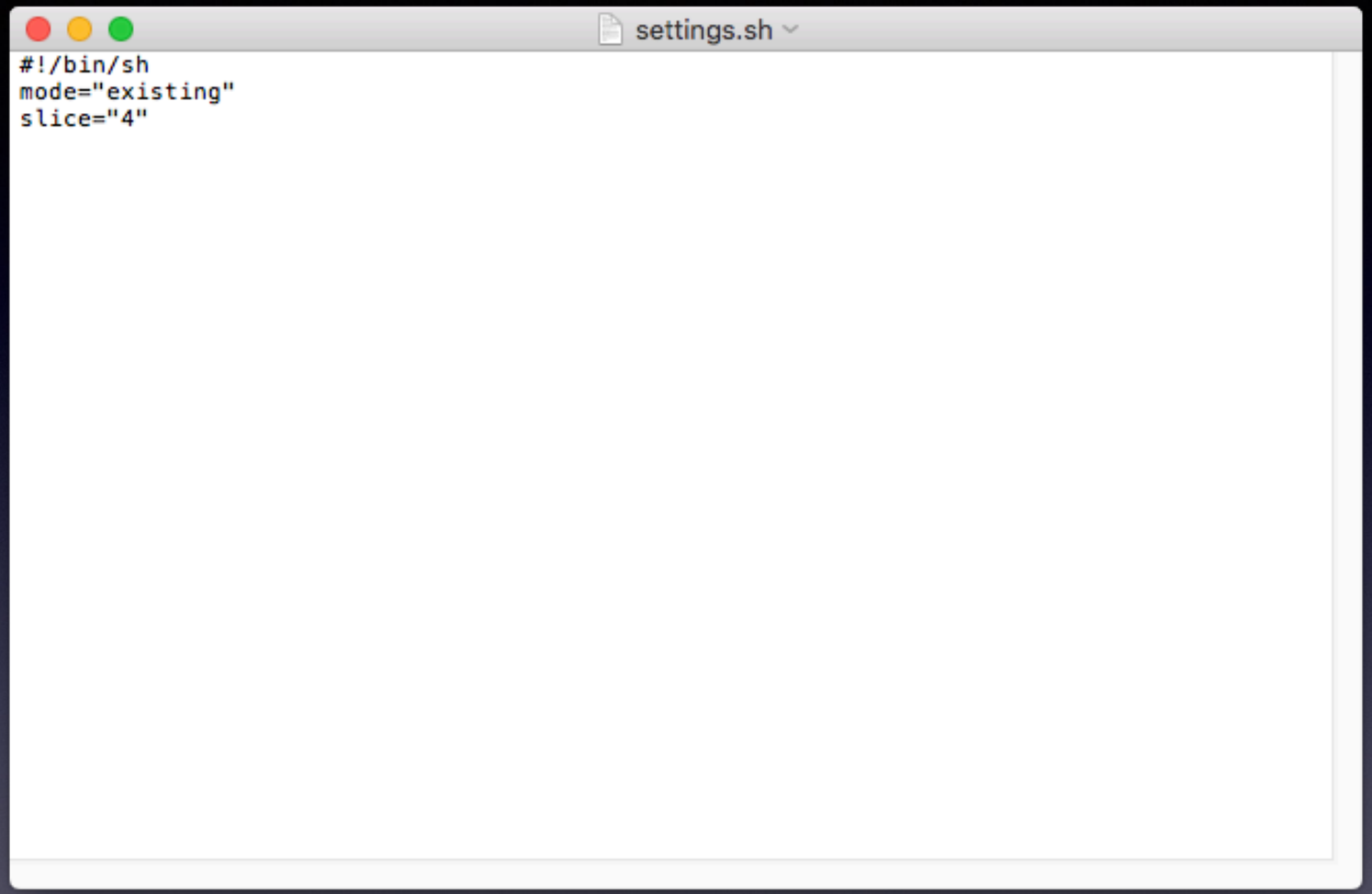
- Driver Magician
- Backup all the drivers
- Create a common folder
- Put it in the VM




Winclone Pro
Eject
Mount Read/Write
Reveal In Finder
Remove
Help
Videos
Sources
Destinations
This Mac
Untitled
999.56 GB
WINCLONE
Save Image
Select a disk or Winclone image from the Sources column on the left. To add images, drag the file into the Sources column. Available Destination options will depend on the selected Source.
To create a Boot Camp partition, use Boot Camp Assistant or Disk Utility to create a MS-DOS (FAT32) formatted partition.
See the Winclone Support Portal for additional information.
Restore To Volume...


Winclone Pro
Eject
Mount Read/Write
Reveal In Finder
Remove
Help
Videos
Sources
Destinations
This Mac
Untitled
999.56 GB
Untitled
999.56 GB
Make Package
Restore To Disk Image
WINCLONE
Windows10VM.winclone
11.03 GB
Select a disk or Winclone image from the Sources column on the left. To add images, drag the file into the Sources column. Available Destination options will depend on the selected Source.
To create a Boot Camp partition, use Boot Camp Assistant or Disk Utility to create a MS-DOS (FAT32) formatted partition.
See the Winclone Support Portal for additional information.
Restore To Volume...


Windows10VM.pkg
Search
Favorites
AirDrop
Recents
iCloud Drive
Applications
Desktop
Documents
Downloads
Devices
Remote Disc
Untitled
Name
Date Modified
Size
Kind
Contents
Feb 14, 2018 at 2:58 PM
--
Folder
Archive.bom
Feb 14, 2018 at 2:58 PM
35 KB
Document
Archive.pax.gz
Feb 14, 2018 at 2:58 PM
103 bytes
gzip co...archive
Info.plist
Mar 20, 2018 at 10:34 AM
2 KB
property list
Resources
Mar 20, 2018 at 10:34 AM
--
Folder
background.jpg
Feb 14, 2018 at 2:58 PM
38 KB
JPEG image
BundleVersions.plist
Feb 14, 2018 at 2:58 PM
181 bytes
property list
postflight
Feb 14, 2018 at 2:58 PM
13 KB
Unix executable
settings.sh
Mar 20, 2018 at 10:34 AM
35 bytes
Plain Text
Welcome.rtf
Mar 20, 2018 at 10:34 AM
827 bytes
RTF Document
Windows1...M.winclone
Mar 20, 2018 at 10:34 AM
3.76 GB
WinClo...cument
version.plist
Feb 14, 2018 at 2:58 PM
294 bytes
property list


settings.sh
#!/bin/sh
mode="existing"
slice="4"

HP RGS software

How to run your Windows remotely

Is Bootcamp needed?

- A lot of software has no need to talk to direct hardware
- One Windows technologist being tasked to support other programs
- Not enough time to schedule classes into Bootcamp rooms

HP Z620

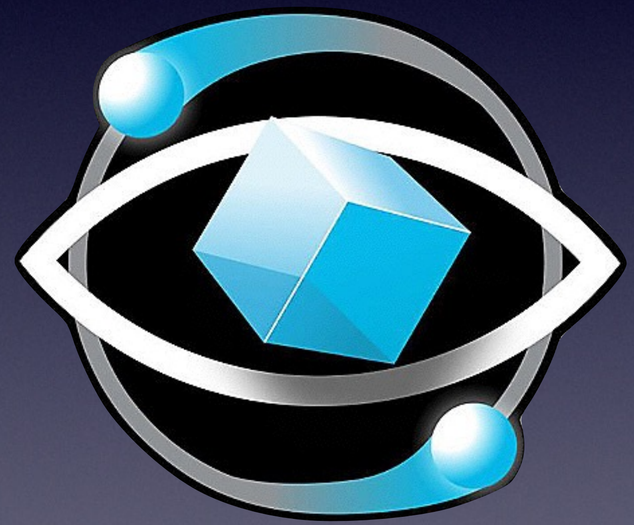
Workstation with Quadro Graphics

Used by Animation and Gaming programs on campus

Coming off lease in large numbers

Remote Graphics Software

- Allows high performance remote graphics
- Can be run on Mac, Windows, & Linux
- Free for Z class workstations



RGS installation

80 workstations in racks
Internal LAN only

Which machine?

- Needs a round robin solution for picking workstation
- What happens if a machine is down?
- What if someone doesn’t log out / shuts down?

Applescript!

- User launches the script
- The script checks a folder of available workstations
- Then remotely logins into the RGS client and presents full screen
- The user can log into the machine with AD credentials. And get a remote volume auto-mounted for assets

rgs01
rgs02
rgs03
rgs04

Boot

rgs01

Logout

rgs01

Shutdown

rgs01

Mount the remote server with the folder full of machine files

Sort by the oldest machine to ensure equal use time

Copy the URL of the machine

Delete the text file to indicate it is in use

On error, ask the user to notify support

Unmount the remote server

Launch the RGS app

Log the user into the RGS client

Set the client to show the remote workstation at full size

Show the dialog box asking the user to logout on completion of use

Quit Applescript app

QuickTime Player File Edit View Window Help
Fri 2:48 PM PSU
Macintosh HD
Backup
DEMO
Launch RGS

Challenges

- Applescript authenticates the RGS login so users can’t directly attach to a machine and take control
- If users forget to logout, we will ask the workstation to self logout after a specified idle time
- Applescript can handle Mac clients, however we are investigating ideas for Windows clients

Custom Tools and Tips

Some things to help out along the way

Option Booting?

- Can take some time. Large RAM = long POST waits
- Security issue booting other disks
- USB extension cables can sometimes fail

11:58 AM

Sheridan | Get Creative

oa-c153-80-M | Please login with your Access Sheridan credentials

Restart

Get The Boot

- Simple Swift app
- Clicking the macOS button quits the app and drops to loginwindow
- Clicking the Windows side does a 1 time boot to Windows
- Clicking on the app window to get focus and pressing ⌘+ n will Netboot the machine

```
#!/bin/bash

mactype=`system_profiler SPHardwareDataType | grep -m1 Model\ Identifier | awk '{print $3}'`
oldmacpro="MacPro5,1"
oldiMac="iMac12,2"

if  [ $mactype = $oldmacpro ]
then
     efidisk=disk1s1
else
     efidisk=disk0s1
fi

if [ $mactype = $oldiMac ]
then
    /usr/sbin/bless --device /dev/disk0s4 --setBoot --legacy --nextonly
else
    uuid=`diskutil info $efidisk | grep -m1 Disk | awk '{print $5}'`
    args=\<array\>\<dict\>\<key\>IOMatch\</key\>\<dict\>\<key\>IOProviderClass\</key\>\<string\>IOMedia\</string\>
\<key\>IOPropertyMatch\</key\>\<dict\>\<key\>UUID\</key\>\<string\>$uuid\</string\>\</dict\>\</dict\>\<key
\>BLLastBSDName\</key\>\<string\>$efidisk\</string\>\</dict\>\</array\>
    nvram efi-boot-next=$args
fi
```

```
#!/bin/bash

mactype=`system_profiler SPHardwareDataType | grep -m1 Model\ Identifier | awk '{print $3}'`
oldmacpro="MacPro5,1"
oldiMac="iMac12,2"

if  [ $mactype = $oldmacpro ]
then
     efidisk=disk1s1
else
     efidisk=disk0s1
fi

if [ $mactype = $oldiMac ]
then
    /usr/sbin/bless --device /dev/disk0s4 --setBoot --legacy --nextonly
else
    uuid=`diskutil info $efidisk | grep -m1 Disk | awk '{print $5}'`
    args=\<array\>\<dict\>\<key\>IOMatch\</key\>\<dict\>\<key\>IOProviderClass\</key\>\<string\>IOMedia\</string\>
\<key\>IOPropertyMatch\</key\>\<dict\>\<key\>UUID\</key\>\<string\>$uuid\</string\>\</dict\>\</dict\>\<key
\>BLLastBSDName\</key\>\<string\>$efidisk\</string\>\</dict\>\</array\>
    nvram efi-boot-next=$args
fi
```

```
#!/bin/bash

mactype=`system_profiler SPHardwareDataType | grep -m1 Model\ Identifier | awk '{print $3}'`
oldmacpro="MacPro5,1"
oldiMac="iMac12,2"

if  [ $mactype = $oldmacpro ]
then
     efidisk=disk1s1
else
     efidisk=disk0s1
fi

if [ $mactype = $oldiMac ]
then
    /usr/sbin/bless --device /dev/disk0s4 --setBoot --legacy --nextonly
else
    uuid=`diskutil info $efidisk | grep -m1 Disk | awk '{print $5}'`
    args=\<array\>\<dict\>\<key\>IOMatch\</key\>\<dict\>\<key\>IOProviderClass\</key\>\<string\>IOMedia\</string\>
\<key\>IOPropertyMatch\</key\>\<dict\>\<key\>UUID\</key\>\<string\>$uuid\</string\>\</dict\>\</dict\>\<key
\>BLLastBSDName\</key\>\<string\>$efidisk\</string\>\</dict\>\</array\>
    nvram efi-boot-next=$args
fi
```

```
#!/bin/bash

mactype=`system_profiler SPHardwareDataType | grep -m1 Model\ Identifier | awk '{print $3}'`
oldmacpro="MacPro5,1"
oldiMac="iMac12,2"

if  [ $mactype = $oldmacpro ]
then
     efidisk=disk1s1
else
     efidisk=disk0s1
fi

if [ $mactype = $oldiMac ]
then
    /usr/sbin/bless --device /dev/disk0s4 --setBoot --legacy --nextonly
else
    uuid=`diskutil info $efidisk | grep -m1 Disk | awk '{print $5}'`
    args=\<array\>\<dict\>\<key\>IOMatch\</key\>\<dict\>\<key\>IOProviderClass\</key\>\<string\>IOMedia\</string\>
\<key\>IOPropertyMatch\</key\>\<dict\>\<key\>UUID\</key\>\<string\>$uuid\</string\>\</dict\>\</dict\>\<key
\>BLLastBSDName\</key\>\<string\>$efidisk\</string\>\</dict\>\</array\>
    nvram efi-boot-next=$args
fi
```

```
#!/bin/bash

mactype=`system_profiler SPHardwareDataType | grep -m1 Model\ Identifier | awk '{print $3}'`
oldmacpro="MacPro5,1"
oldiMac="iMac12,2"

if  [ $mactype = $oldmacpro ]
then
     efidisk=disk1s1
else
     efidisk=disk0s1
fi

if [ $mactype = $oldiMac ]
then
    /usr/sbin/bless --device /dev/disk0s4 --setBoot --legacy --nextonly
else
    uuid=`diskutil info $efidisk | grep -m1 Disk | awk '{print $5}'`
    args=\<array\>\<dict\>\<key\>IOMatch\</key\>\<dict\>\<key\>IOProviderClass\</key\>\<string\>IOMedia\</string\>
\<key\>IOPropertyMatch\</key\>\<dict\>\<key\>UUID\</key\>\<string\>$uuid\</string\>\</dict\>\</dict\>\<key
\>BLLastBSDName\</key\>\<string\>$efidisk\</string\>\</dict\>\</array\>
    nvram efi-boot-next=$args
fi
```

```
#!/bin/bash

mactype=`system_profiler SPHardwareDataType | grep -m1 Model\ Identifier | awk '{print $3}'`
oldmacpro="MacPro5,1"
oldiMac="iMac12,2"

if  [ $mactype = $oldmacpro ]
then
     efidisk=disk1s1
else
     efidisk=disk0s1
fi

if [ $mactype = $oldiMac ]
then
    /usr/sbin/bless --device /dev/disk0s4 --setBoot --legacy --nextonly
else
    uuid=`diskutil info $efidisk | grep -m1 Disk | awk '{print $5}'`
    args=\<array\>\<dict\>\<key\>IOMatch\</key\>\<dict\>\<key\>IOProviderClass\</key\>\<string\>IOMedia\</string\>
\<key\>IOPropertyMatch\</key\>\<dict\>\<key\>UUID\</key\>\<string\>$uuid\</string\>\</dict\>\</dict\>\<key
\>BLLastBSDName\</key\>\<string\>$efidisk\</string\>\</dict\>\</array\>
    nvram efi-boot-next=$args
fi
```

```
#!/bin/bash

mactype=`system_profiler SPHardwareDataType | grep -m1 Model\ Identifier | awk '{print $3}'`
oldmacpro="MacPro5,1"
oldiMac="iMac12,2"

if  [ $mactype = $oldmacpro ]
then
     efidisk=disk1s1
else
     efidisk=disk0s1
fi

if [ $mactype = $oldiMac ]
then
    /usr/sbin/bless --device /dev/disk0s4 --setBoot --legacy --nextonly
else
    uuid=`diskutil info $efidisk | grep -m1 Disk | awk '{print $5}'`
    args=\<array\>\<dict\>\<key\>IOMatch\</key\>\<dict\>\<key\>IOProviderClass\</key\>\<string\>IOMedia\</string\>
\<key\>IOPropertyMatch\</key\>\<dict\>\<key\>UUID\</key\>\<string\>$uuid\</string\>\</dict\>\</dict\>\<key
\>BLLastBSDName\</key\>\<string\>$efidisk\</string\>\</dict\>\</array\>
    nvram efi-boot-next=$args
fi
```

```
#!/bin/bash

/usr/sbin/bless --netboot --booter tftp://192.168.1.1/NetBoot/NetBootSP0/deploy.nbi/i386/booter --kernelcache tftp://
192.168.1.1/NetBoot/NetBootSP0/deploy.nbi/i386/x86_64/kernelcache --options ‘rp=http://192.168.1.1/NetBoot/NetBootSP0/
deploy.nbi/netinstall.dmg'
```

```
#!/bin/bash

/usr/sbin/bless --netboot --booter tftp://192.168.1.1/NetBoot/NetBootSP0/deploy.nbi/i386/booter --kernelcache tftp://
192.168.1.1/NetBoot/NetBootSP0/deploy.nbi/i386/x86_64/kernelcache --options ‘rp=http://192.168.1.1/NetBoot/NetBootSP0/
deploy.nbi/netinstall.dmg'
```

```
#!/bin/bash

/usr/sbin/bless --netboot --booter tftp://192.168.1.1/NetBoot/NetBootSP0/deploy.nbi/i386/booter --kernelcache tftp://
192.168.1.1/NetBoot/NetBootSP0/deploy.nbi/i386/x86_64/kernelcache --options ‘rp=http://192.168.1.1/NetBoot/NetBootSP0/
deploy.nbi/netinstall.dmg'
```

```
#!/bin/bash

/usr/sbin/bless --netboot --booter tftp://192.168.1.1/NetBoot/NetBootSP0/deploy.nbi/i386/booter --kernelcache tftp://
192.168.1.1/NetBoot/NetBootSP0/deploy.nbi/i386/x86_64/kernelcache --options ‘rp=http://192.168.1.1/NetBoot/NetBootSP0/
deploy.nbi/netinstall.dmg'
```

```
#!/bin/bash

/usr/sbin/bless --netboot --booter tftp://192.168.1.1/NetBoot/NetBootSP0/deploy.nbi/i386/booter --kernelcache tftp://
192.168.1.1/NetBoot/NetBootSP0/deploy.nbi/i386/x86_64/kernelcache --options ‘rp=http://192.168.1.1/NetBoot/NetBootSP0/
deploy.nbi/netinstall.dmg'
```

Bless command

- systemsetup tool supposed to replace bless command
- sudo systemsetup -setstartupdisk /Volumes/BOOTCAMP
- Doesn’t seem to work for switching to BootCamp.

What about SIP?

- Bless requires turning off part of SIP
- Might as well set a firmware password too
- Might as well add some Netboot servers into SIP
- Might as well make sure it boots to Mac every time

```
csrutil enable --without nvram

csrutil netboot add 192.168.1.1

csrutil netboot add 192.168.1.2

csrutil netboot add 192.168.1.3

nvram -d boot-args

bless --device /dev/disk0s2 –setBoot

/Applications/Utilities/\"Firmware Password Utility.app
\"/Contents/Resources/setregproptool
-p myfirmwarepswd -m command
```

How do I do this efficiently on all my Macs?!?

Barcode!

Convert typing to blasting

csrutil enable --without nvram

csrutil netboot add 192.168.1.1

csrutil netboot add 192.168.1.2

nvram -d boot-args

bless --device /dev/disk0s2 --setBoot

/Ap/Ut/Fi/Co/Re/setregproptool -p myfirmwarepswd -m command

DEMO

Arduino Micro!

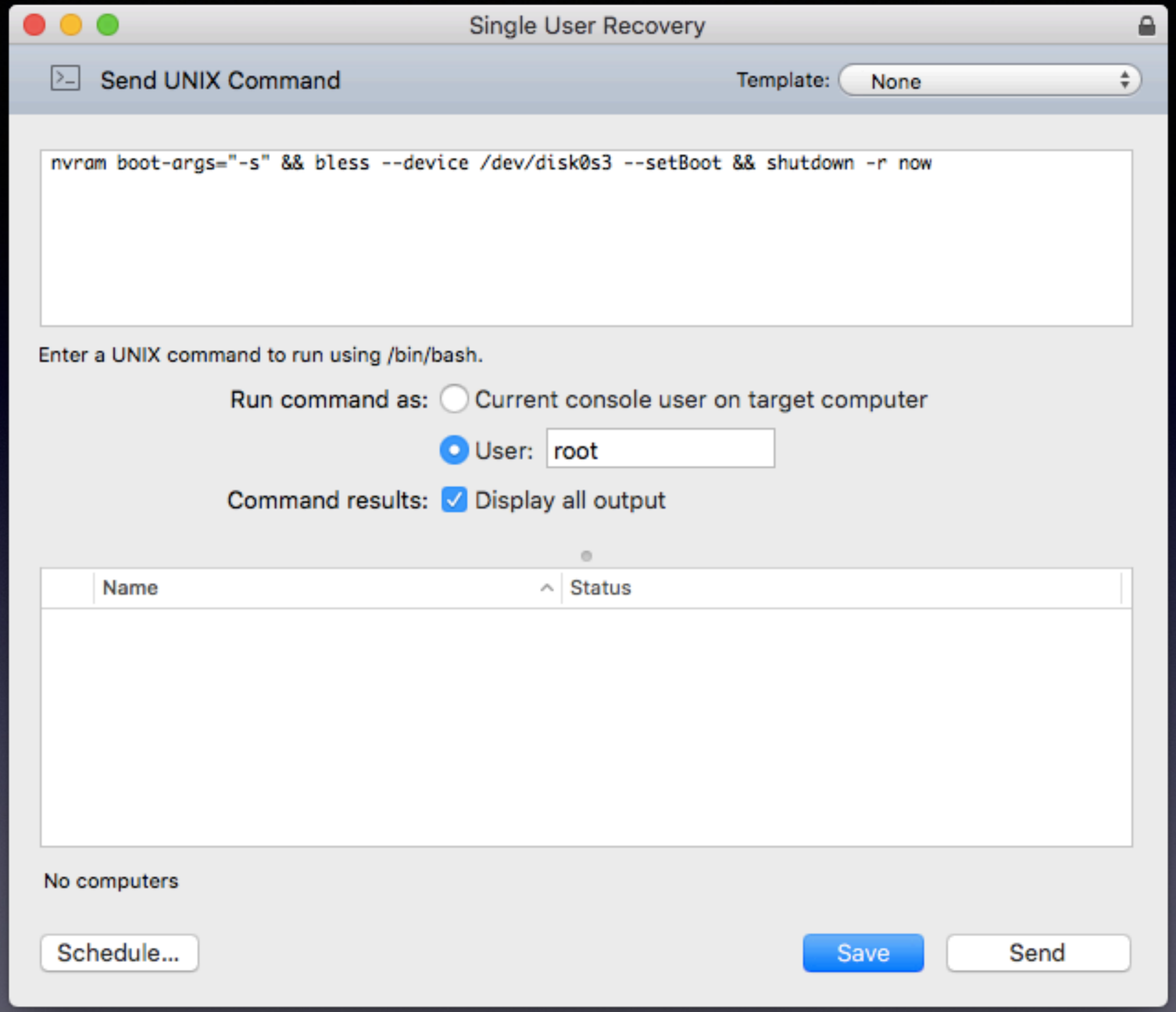
Add the library:
HID-Project by Nico Hood

```
#include <HID-Project.h>
#include <HID-Settings.h>

void setup() {
  // put your setup code here, to run once:
  BootKeyboard.begin();
  delay(500);
}

void loop() {
  // put your main code here, to run repeatedly:
  BootKeyboard.println("csrutil enable --without nvram\n");
  BootKeyboard.println("csrutil netboot add 192.168.1.1\n");
  BootKeyboard.println("csrutil netboot add 192.168.1.2\n");
  BootKeyboard.println("csrutil netboot add 192.168.1.3\n");
  BootKeyboard.println("nvram -d boot-args\n");
  BootKeyboard.println("bless --device /dev/disk0s2 --setBoot\n");
  BootKeyboard.println("/Applications/Utilities/\"Firmware Password Utility.app\"/Contents/Resources/setregproptool -p
myfirmwarepswd -m command\n");
  BootKeyboard.println("reboot\n");
  while (1);
}
```

DEMO

Super Tip!

- Booting into the Recovery partition takes a long time 1:33
- Holding ⌘ + s while booting gets you single user mode
- Holding ⌘ + s + r while booting gets you single user recovery mode, where SIP can be manipulated in 13 seconds!



* With 10.13 the recovery is /dev/disk1s3

Code available at:

robroy.firebird.sheridanc.on.ca

Thank You for your time!

Q&A

Code available at:

robroy.firebird.sheridanc.on.ca

rob.roy@sheridancollege.ca