

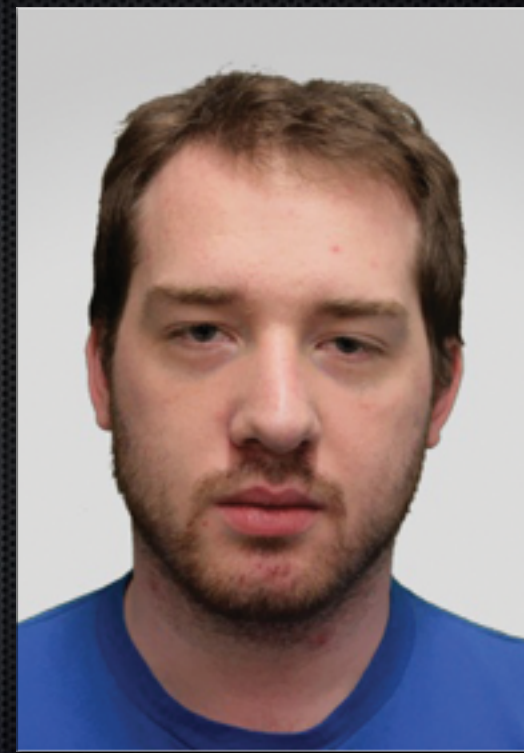
# UW-Milwaukee JAMF Implementation

Kyle Brockman  
Mac Support Specialist

<http://tinyurl.com/odpestv>

# Kyle Brockman

- Worked at UWM for 6 years and still counting
- E-mail: [brockma9@uwm.edu](mailto:brockma9@uwm.edu)
- Twitter: @brockma9



Talk about who I am, what I do at UWM.

My e-mail and twitter account

# What is JAMF's Casper Suite

- Servers
- Manage Mac OS X
- Manage iOS Devices



So what is JAMF.

What servers you need, that is manages macs and iOS Devices. This system wait for clients to talk to it.



# What are the Pieces to JAMF

- JAMF Software Server(s) (JSS)
- Distribution server(s)
- JAMF makes a NetSUS, VM appliance (Optional)

There is a central server called the JSS

There is a distribution server called distribution servers

And a third server that is option but use full, the NetSUS

# JSS Server

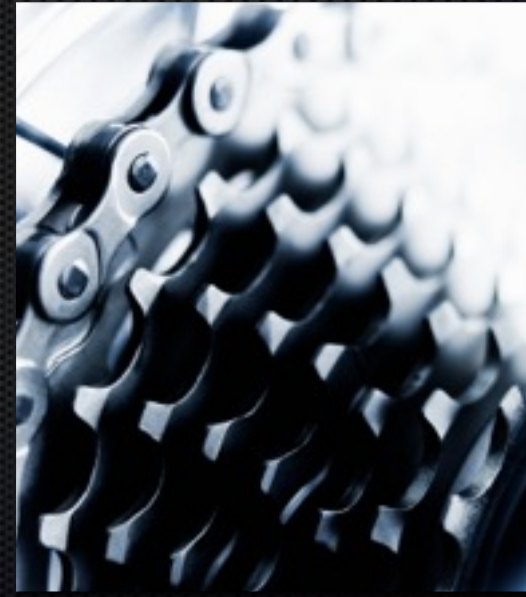
- Tomcat
- MySQL



This is the brains of the system, it keep the information in a MySQL database. The front end is a Tomcat web server

# Distribution server

- SMB
- AFP
- HTTPS
- JDS



This is where scripts and packages are stored, unless you have a JDS.

SMB - Server message block (Windows file share)

- downsides to this

AFP - Apple File share

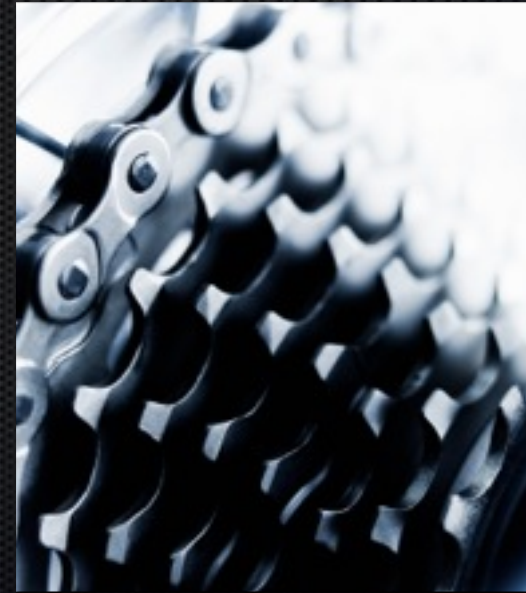
HTTPS - web access ssl

JDS - Jamf Distribution server



# NetSUS

- NETboot
- Software Updates Server(SUS)



This is used to net boot and have a local software updates repos. The software updates server is based on reposado.

reposado is a open source project. <https://github.com/wdas/reposado>

<https://jamfnation.jamfsoftware.com/viewProduct.html?id=180&view=info>

<https://github.com/jamf/NetSUS>

# UW-Milwaukee Setup

- We have a JSS server on RHEL in VM
- Then two Mac mini's with 250GB SSDs





UW-Milwaukee Server Layout



Inside the website

# Grouping Computers

- Static Groups
- Smart Groups

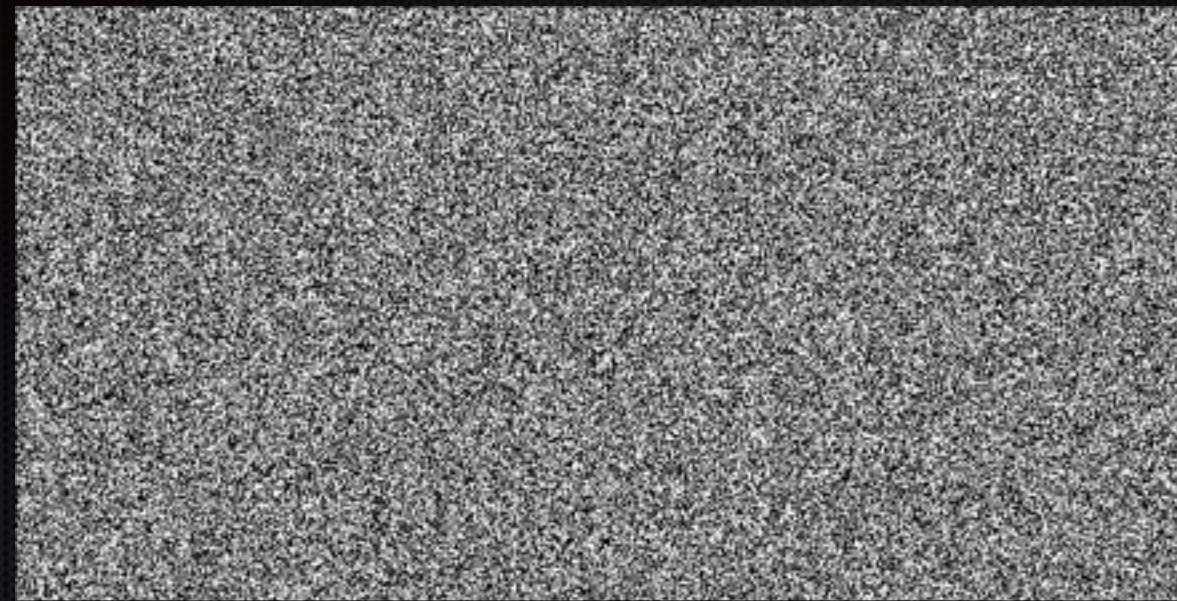


Grouping the Computers

Static Groups

Smart Groups





# Static Groups

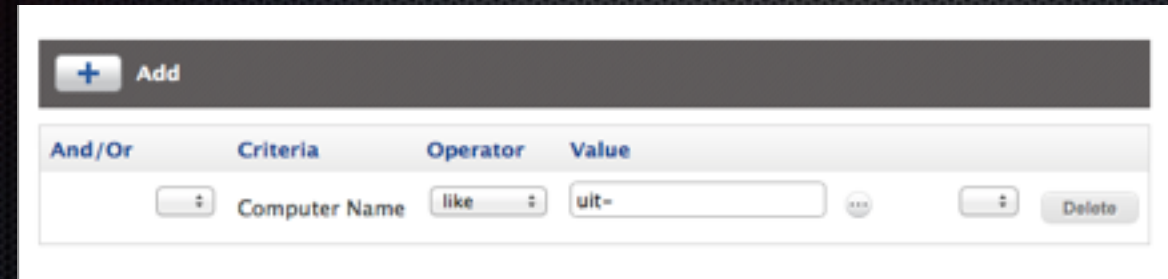
## Static Groups

Groups like you have know them. Add a machine to a group and keep them up to date.



# Smart Groups

Smart Groups



The screenshot shows a search interface with a dark background. At the top, there is a grey bar with a blue plus icon and the text "Add". Below this is a table with four columns: "And/Or", "Criteria", "Operator", and "Value". The table has one row with the following values: a dropdown menu showing "and", the text "Computer Name", a dropdown menu showing "like", and a text input field containing "uit=". To the right of the input field is a small circular button with a plus icon. At the bottom right of the table is a "Delete" button.

And/Or	Criteria	Operator	Value
and	Computer Name	like	uit=

Smart groups are Searches

Search for what you want to be in this group.





## Criteria for Searches

Extension Attributes

FileVault 2 Status

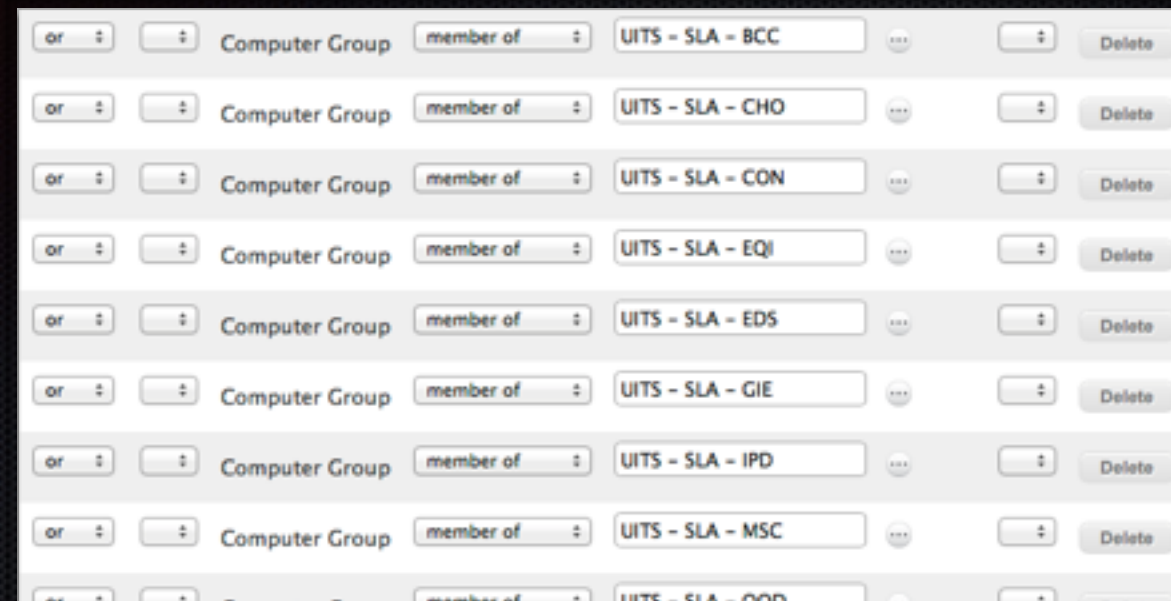
Mapped Printers

Recovery HD Present

Room

Serial Number - Apple recalls

Available RAM Slots



## Groups in a Group

You can have your groups inside of smart groups.

Build all labs, all office machines from departments

The screenshot shows a search criteria interface with a table structure. At the top is a dark grey bar with a blue plus icon and the text 'Add'. Below this is a table with four columns: 'And/Or', 'Criteria', 'Operator', and 'Value'. The first row has an empty 'And/Or' cell, 'Computer Group' in 'Criteria', 'member of' in 'Operator', and 'UIT - SLA - ALL Office Mac' in 'Value'. The second row has 'and' in 'And/Or', 'FileVault 2 Status' in 'Criteria', 'is' in 'Operator', and 'No Partitions Encrypted' in 'Value'. Each row has a 'Delete' button to its right. There are also small circular icons with vertical lines next to the criteria and operator fields.

And/Or	Criteria	Operator	Value
	Computer Group	member of	UIT - SLA - ALL Office Mac
and	FileVault 2 Status	is	No Partitions Encrypted

## A Group in a Group in a Group

Then the big groups that have groups can be used for other smart searches. Is all the office laptops encrypted?



UIT - All Managed Machines	262
UIT - All Managed Machines - Microsoft Office Update	17
UIT - All Managed Machines - Need Java Runtime	0
UIT - SLA - ALL - Mavericks Cached	3
UIT - SLA - ALL - Mavericks Upgrade Clean up	1
UIT - SLA - ALL Non-UIT	46
UIT - SLA - ALL Office Machines	79
UIT - SLA - ALL Office Machines - Chrome Update	4
UIT - SLA - ALL Office Machines - Firefox Update	37
UIT - SLA - ALL Office Machines - iMovie update	1
UIT - SLA - ALL Office Machines - iPhoto Update	0
UIT - SLA - ALL Office Machines - Laptops	37
UIT - SLA - ALL Office Machines - Laptops - No Encrypted	6
UIT - SLA - ALL Office Machines - OpenOffice Update	1
UIT - SLA - ALL Office Machines - TM Non-encrypted	0
UIT - SLA - ALL Office Machines - VLC Update	0
UIT - SLA - ALL Office Machines - WUSMU	9
UITS - Lab - UIT - CCL - All Machines	57
UITS - Lab - UIT - CCL - Bolton	6

# Naming is important

Keeping this clean is important, name and name to the standards you have in your department. Then the windows guy can kind of understand what is going on.

# Casper Extension Attributes

- These are scripts that client machines run and return values

```
1 #!/bin/bash
2
3 #####
4 #
5 # Extension Attribute checks to display Java Version number.
6 #
7 #
8 #####
9
10 java_version=`java -version 2>&1 | head -n 1 | cut -d\" | -f 2`
11 echo "<result> $java_version </result>"
12
13 exit 0
```

This is a script for checking the Java Runtime Installed.

These are scripts that report information to you. They need to be returned in a special echo command.

You can have if statements and so on.

<https://jamfnation.jamfsoftware.com/article.html?id=92>



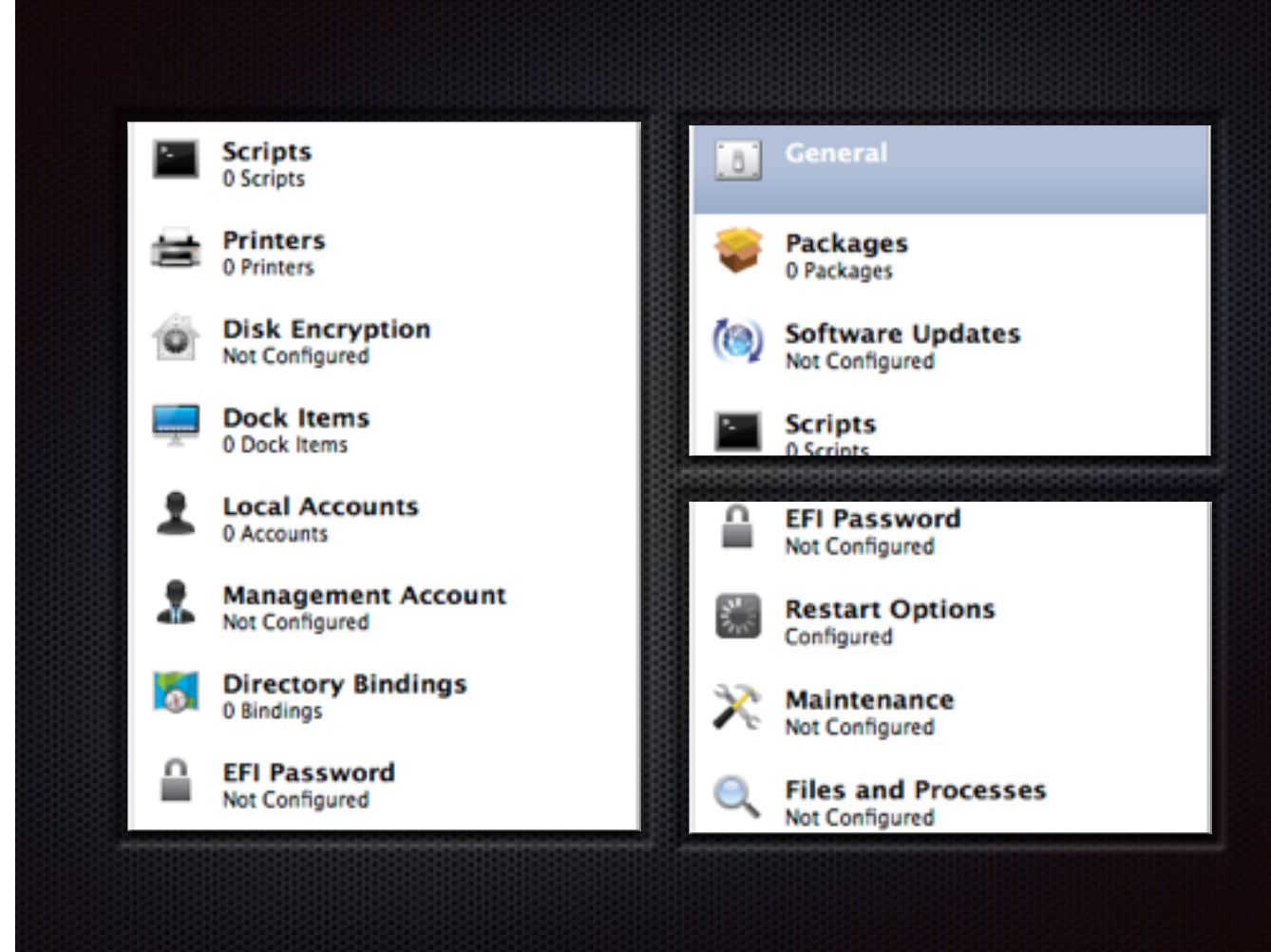
Managing the Macs





# Policies

- This is what does the actions

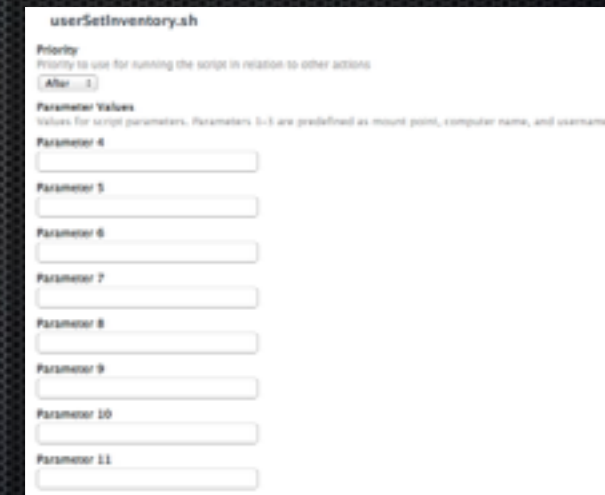


These are all the Options on the Policies.

I'm going to focus on three. Scripts, Packages, and Maintenance.

# Scripts

- You can have it run a script
- You can pass it Parameters



The screenshot shows a configuration window for a script named 'userSetInventory.sh'. It includes a 'Priority' section with a dropdown menu set to 'After...3'. Below this is a 'Parameter Values' section with a note: 'Values for script parameters. Parameters 1-3 are predefined as mount point, computer name, and username'. There are eleven input fields labeled 'Parameter 4' through 'Parameter 14', each with a corresponding text box for user input.

You can just build a stock script that does stuff.

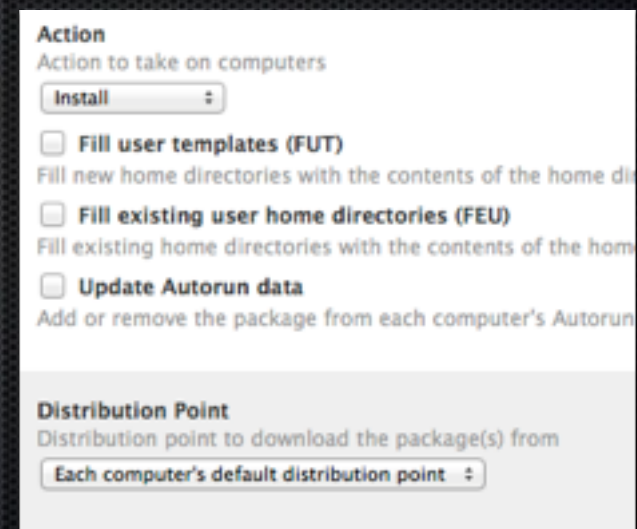
You can use scripts that have Parameters that need to be filled in from the policy.

\$4 \$5



# Packages

- Action
- Fill templates and home dirs



The screenshot shows a configuration window with two main sections. The first section, titled 'Action', has a subtitle 'Action to take on computers'. It contains a dropdown menu set to 'Install' and three checkboxes: 'Fill user templates (FUT)', 'Fill existing user home directories (FEU)', and 'Update Autorun data'. Each checkbox has a descriptive line of text below it. The second section, titled 'Distribution Point', has a subtitle 'Distribution point to download the package(s) from' and a dropdown menu set to 'Each computer's default distribution point'.

**Action**  
Action to take on computers

Install

☐ **Fill user templates (FUT)**  
Fill new home directories with the contents of the home di

☐ **Fill existing user home directories (FEU)**  
Fill existing home directories with the contents of the hom

☐ **Update Autorun data**  
Add or remove the package from each computer's Autorun

**Distribution Point**  
Distribution point to download the package(s) from

Each computer's default distribution point

Action: Install, Cache, and Installed Cached.

Install - does as it says, Cache - puts the app in the waiting room, Install Cached - install packages that have been cached to the clients device.

Fill explain

**Maintenance**

- ☒ **Update Inventory**  
Force computers to submit updated
- ☐ **Reset Computer Names**  
Change the computer name on comp
- ☐ **Install Cached Packages**  
Install packages cached by the Casp
- ☐ **Fix Disk Permissions**
- ☐ **Fix ByHost Files**
- ☐ **Flush System Caches**  
Flush caches from /Library/Caches/
- ☐ **Flush User Caches**  
Flush caches from ~/Library/Caches
- ☐ **Verify Startup Disk**

# Maintenance

Inventory - update machine info in JSS for smart groups

Rename computer based on name in JSS

Fix Disk Permissions

### Prevent Installing OS X Mavericks

Options

Scope

**Display Name**  
Display name for the restricted software record

Prevent Installing OS X Mavericks

**Process Name**  
Name of the process to restrict (e.g. "Chess", or "Chess.app"). The asterisk (\*) can be used as a wildcard character

Install OS X Mavericks

☒ **Restrict exact process name**  
Only restrict processes that match the exact process name. The match is case-sensitive and recognizes the asterisk (\*) as a literal character

☐ **Send email notification on violation**  
When the process is found, send an email notification to JSS users with email notifications enabled. An SMTP server must be set up in the JSS for this to work

☒ **Kill process**  
Terminate the process when found

☐ **Delete application**  
Delete the application running the restricted process

**Message**  
Message to display to users when the process is found

Cancel

Save

# Restricted Software



## Prevent Installing OS X Mavericks

Options	Scope
<p><b>Display Name</b> Display name for the restricted software record</p> <p>Prevent Installing OS X Mavericks</p> <p><b>Process Name</b> Name of the process to restrict (e.g. "Chess", or "Chess.app"). The asterisk (*) can be used as a wildcard character</p> <p>Install OS X Mavericks</p> <p><input checked="" type="checkbox"/> <b>Restrict exact process name</b> Only restrict processes that match the exact process name. The match is case-sensitive and recognizes the asterisk (*) as a literal character</p> <p><input type="checkbox"/> <b>Send email notification on violation</b> When the process is found, send an email notification to JSS users with email notifications enabled. An SMTP server must be set up in the JSS for this to work</p> <p><input checked="" type="checkbox"/> <b>Kill process</b> Terminate the process when found</p> <p><input type="checkbox"/> <b>Delete application</b> Delete the application running the restricted process</p> <p><b>Message</b> Message to display to users when the process is found</p> <p></p>	
<p>Cancel Save</p>	

Name the restricted software, then you need to know the process name.

Restrict the exact process name, e-mail when it's run, and then kill the process, and delete the app if it tries to run.



## Managing iOS Devices

I'm going to be talking about three things with this. Configuration Profiles, Apps, and VPP management.



## Configuration Profiles

Everything that you can do in iPhone config utility can be done here. This is nice for setting up e-mail, wifi, and so on.





# Apps

You can make an app catalog of the apps that all the users should have. You can control the codes for get the apps for your clients.

# Self Service



It's an in house App Store

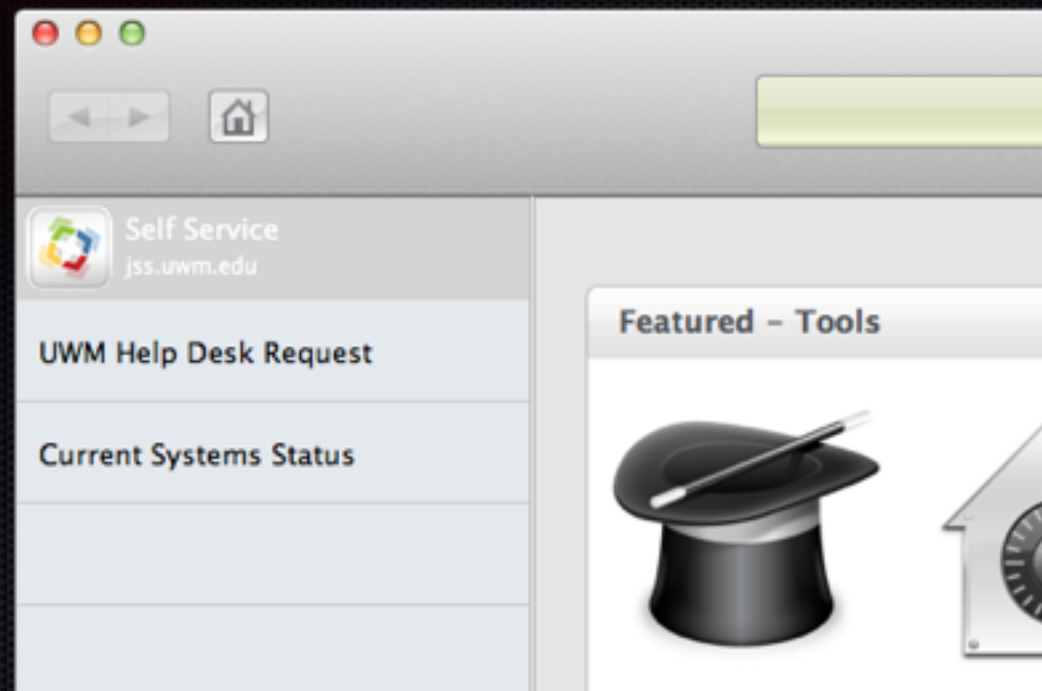
Also you can give clients access to more



# The Interface

You setup everything that is in this.

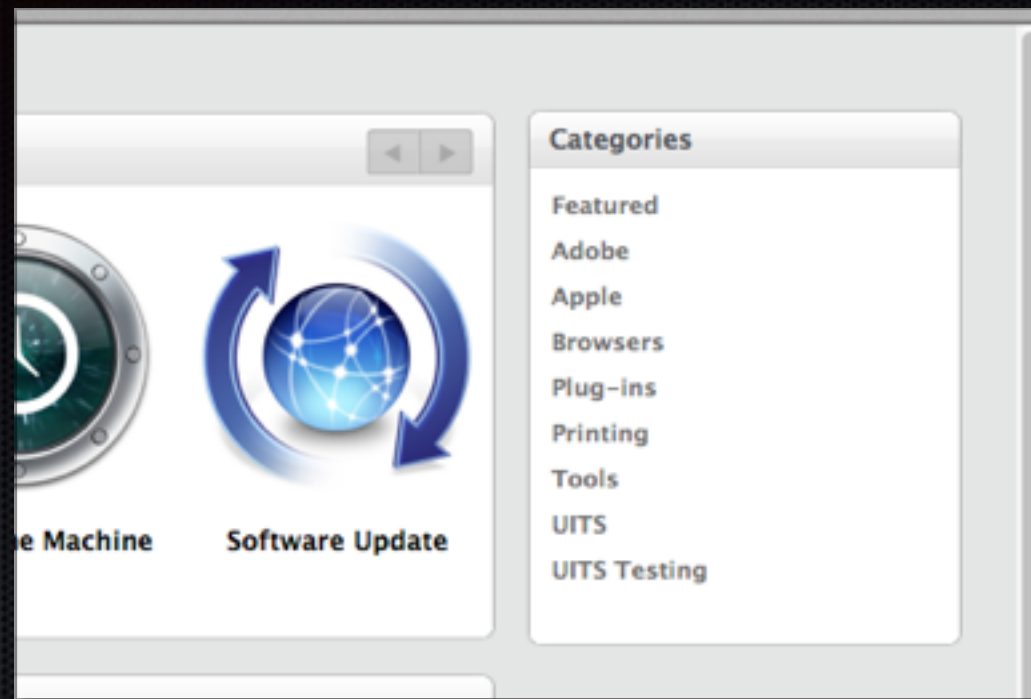




## Add-ons

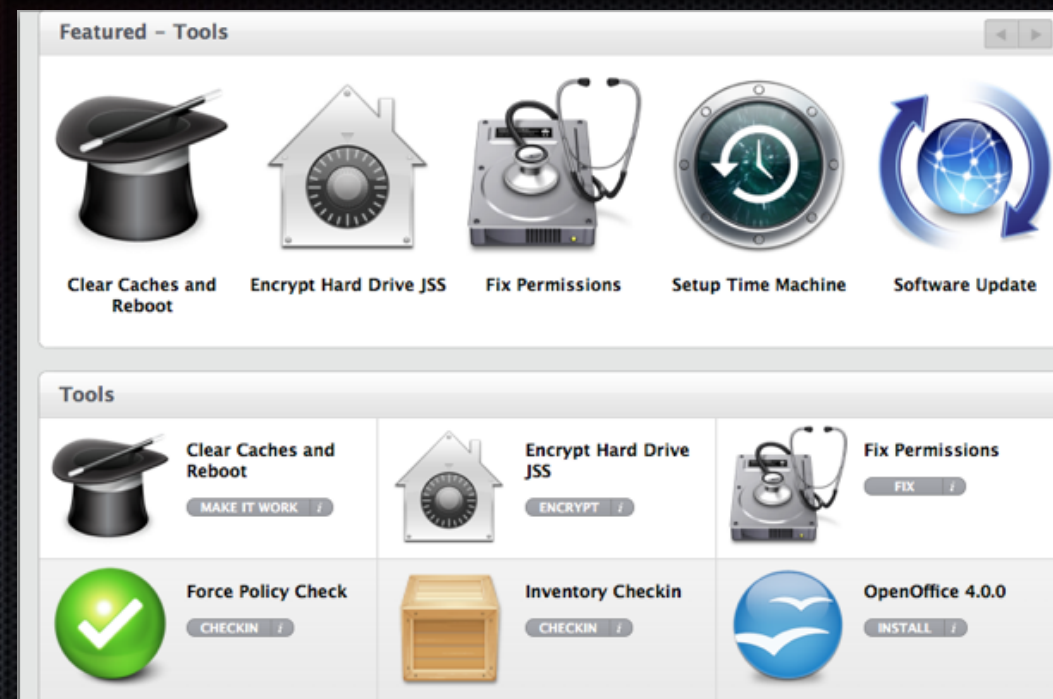
You setup everything that is in this.

We just have web sites, but this gets clients to the web form for Help desk and the Status page for our systems.



## Categories

This are a way to group apps  
Keep is simple, this is for the clients.



# First AID Tools

Polices for the users to do things with out having to call IT.



# Cool tricks with Self Service



Neat tricks with self service

# Mavericks upgrade

- Having the clients updating to Mavericks



Having the clients updating their computers to Mavericks on there time frame.



# Cache the update pkg

PreBuilt

This package was built with createOSXinstallpkg from the munki tools.

<http://managingosx.wordpress.com/2012/07/25/son-of-installdion-pkg/>

Runs an installer package from

<http://derflounder.wordpress.com/2013/05/13/first-boot-package-install-pkg/>

it runs two scripts and the java runtime for Mavericks





## Button in Self Service

The button will show up after the installer package is cached on the clients machine. This will allow them to update off network, nights and weekends.



Installer runs

Then they see the standard install screen. This looks like the standard process.



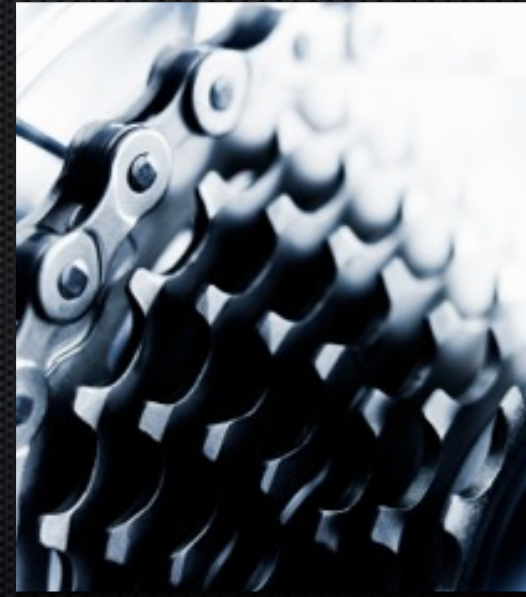
## Clean up

Then after to update, in the back group I have scripts in jamf that will remove the cached package freeing up the 5.5GB of data it is using up.



# Having the Client backup their machine before re-image

- Have the client see part of the rebuild of a machine process



This will be a button in self service that the client clicks when they are ready to give up their machine to be rebuilt or have their home dir transfered to a new machine.

```

1 #!/bin/bash
2
3 ###
4 #Script Designed by Ian Gunther and Kyle Brockman
5 #While working at the University of Wisconsin-Milwaukee
6 ###
7
8 export TIME="date +%Y.%m.%d"
9
10 export FULLTIME="date +%Y.%m.%d %H:%M:%S"
11
12 export M="MacBackups"
13
14 export bkVolume="/tmp/VM"
15
16 export serial=$(system_profiler | grep "Serial Number (system)" | awk '{print $4}')
17
18 export BACKUP=TIME-serial-'hostname'
19
20 # create a named pipe
21 mkfifo /tmp/hpipe
22
23 # create a background job which takes its input from the named pipe
24 /Applications/Utilities/CocoaDialog.app/Contents/MacOS/CocoaDialog progressbar \
25 --indeterminate --title "Machine Backup For UITS Desktop Support" \
26 --text "Backing Up..." > /tmp/hpipe &
27
28 # associate file descriptor 3 with that pipe and send a character through the pipe
29 exec 3<>/tmp/hpipe
30 echo -n . >&3
31
32 #Stop the machine from sleeping
33caffeinate &
34
35 #Wake the mount point if it does not already present.
36 if [[ ! -d "$bkVolume" ]] then
37     mkdir $bkVolume
38 fi
39
40 #Unmount anything that might be mounted to the backup point
41 umount $bkVolume
42
43 sleep 10
44
45 #mount the backup share
46 mount_smbfs //server.local/VM $bkVolume
47
48

```

This is the code

Puts up a process bar and prevents the machine from going to sleep. Then mounts the server share, the machine will be backing up to.

```

58
59
60 if mount | grep $bkVolume; then
61     echo "SMB is Mounted"
62 else
63     echo "Samba is NOT MOUNTED"
64     # now turn off the progress bar by closing file descriptor 3
65     exec 3&&-
66     # wait for all background jobs to exit
67     wait
68     rm -f /tmp/hpipe
69     exit 1
70 fi
71
72 #make the backup of the drive
73 hdiutil create $bkVolume/AutoDelete/BACKUP.dmg --verbose --format UDBK --noeraseover
74 --srcfolder /
75
76 #scan the image for restore
77 #tar imagescan --source $bkVolume/BACKUP --verbose
78
79 #check the backup
80 hdiutil verify $bkVolume/AutoDelete/BACKUP.dmg
81
82 if [[ $? == 0 ]]; then
83     echo "Backup is good"
84 else
85     mail -s "Backup Failed for 'hostname' at `date +%Y.%m.%d %H:%M:%S`"
86     ds-tachs@uam.edu ds-ftach@uam.edu <<TOP
87
88     Hello,
89
90     The backup for "hostname" failed.
91
92     Thank you,
93     Backup Process.
94
95     TOP
96 fi
97 # now turn off the progress bar by closing file descriptor 3
98 exec 3&&-
99
100 # wait for all background jobs to exit
101 rm -f /tmp/hpipe
102
103 #Mail Tachs the backup is completed
104
105

```

This is the code

This will then verify the share is mounted, then start dumping the root disk to the backup location. After the backup completes it will verify the disk image, then it sends an e-mail to IT, to let them know the backup completed, and computers is ready for pickup.



```
95  
96 mail -s "Backup Complete for `hostname` at `date "+%Y.%m.%d %H:%M:%S""  
97 ds-techs@uwm.edu ds-fte@uwm.edu <<EOF  
98 Hello,  
99  
100 The backup for `hostname` is complete and ready to be picked up to be re-imaged.  
101  
102 It started at `echo $FULLTIME` and ended at `date "+%Y.%m.%d %H:%M:%S"`.  
103  
104 Total backup size is `ls -lah /tmp/MacBackups/AutoDelete/$BACKUP.dmg | awk '{ print $5  
105 }'`.  
106  
107 Thank you,  
108 Backup Process.  
109 EOF  
110  
111 killall caffeinate  
112  
113 exit 0
```

This is the code

Then kill the process preventing the computer from sleeping.

Then have the policy shut the computer down.



## Temporary Admin Rights

This allows the user of a system to get admin right but only temporary.

Why do this ...

This way it's easy for a client to get admin install what they need and for the rights to stop.



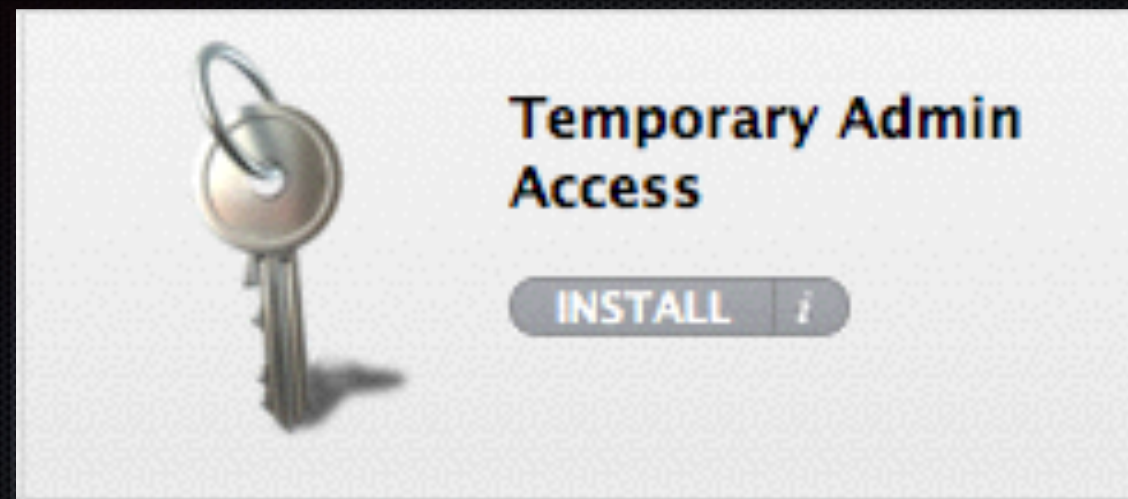
**INITECH**

# T.P.S. REPORT

Have client fill out a form

This form is so they know with great power comes great responsibility.





Client clicks the button

Then the client clicks the button and magic happens for them.  
They get admin rights.



# The rights go bye bye

Launch Daeman

30 min later a launchD agent runs and takes the rights a way.

We lock down other things in our clients, with the power now with security command and the auth db flag. They can not unlock accounts and other system preference panes.

*DEMO*



*Questions?*

<http://tinyurl.com/odpestv>