

Deploying 1400+  
computers in 3 weeks? Are  
you nuts?!?

Sean Kaiser  
Northmont City Schools

# Who am I?

- systems integration specialist
- server, network, Mac, Google apps domain administrator
- level 3 tech support

# What this session is not

- A how-to on:
  - DeployStudio
  - the luggage
  - munki
  - puppet
  - reposado

# What this session is

- How I configured, used, and extended:
  - DeployStudio
  - the luggage
  - munki
  - puppet
  - reposado

to complete a major deployment of machines in a short amount of time by a small group of staff

# Brief Project Background

- needs
  - replace various old platforms (iBooks, eMacs, iMacs)
  - prepare for mandated online testing
- why during school year
  - financial deals from Apple during Oct-Dec quarter
  - warehouse space limited for huge shipment of computers
- timeline ~2 months from initial discussion until go time

# History of deployment methods

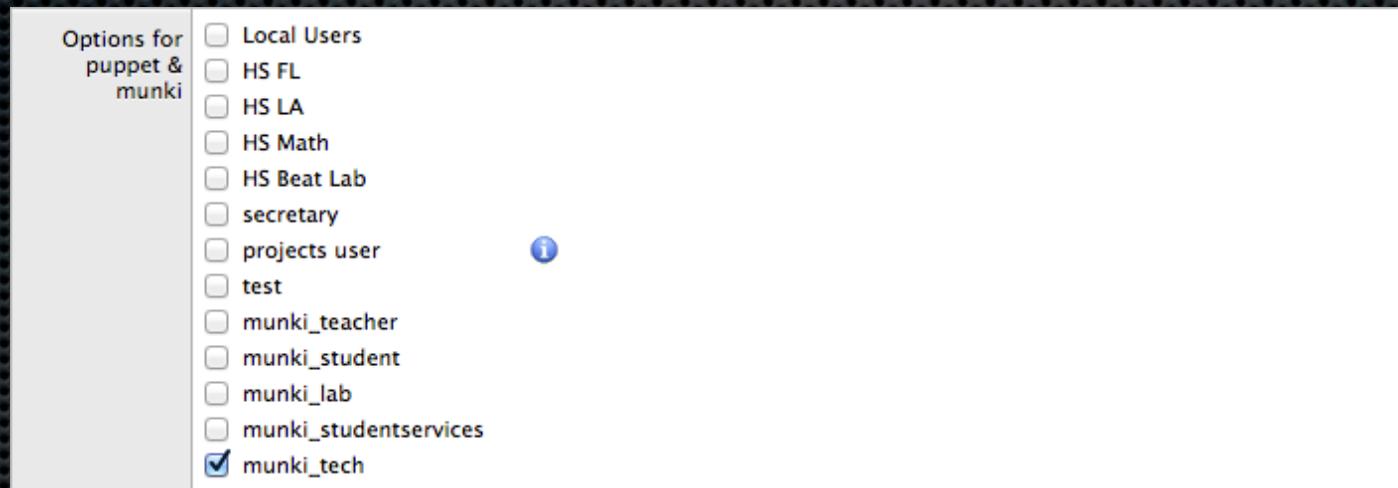
- manual
- firewire target disk mode
- netboot
  - netrestore
  - deploystudio
- monolithic/golden master vs. modular deployment

# Getting the pieces in place

- Web Help Desk
- DeployStudio
- the luggage
- munki
- puppet
- reposado
- take delivery of machines

# Getting the pieces in place - Web Help Desk

- custom field for puppet classes, add options for munki



- performance issues due to frontbase database and tie-in with puppet
- refactor existing lookup scripts to query mysql database instead of frontbase

# Getting the pieces in place - DeployStudio

- updated existing workflow for 10.7.5 image
- created new workflow for 10.8.2 image
- created new NBI for new hardware
- created new workflow for modular deployment (“Northmont-ize”)
- full imaging (wipe/reinstall) workflows include same steps as the Northmont-ize workflow

# Getting the pieces in place - DeployStudio

- Northmont-ize workflow
  - set time server; computer name; anonymous bind to OD; install facter, puppet, and config files/service; create local user accounts; install munki; enable ARD; join appropriate WLAN; split hard drive; set munki bootstrap file
  - all postponed installs (so runs at first boot)
  - workflow takes ~2 minutes

# Getting the pieces in place - DeployStudio

- first boot process takes <5 minutes
- longest part is the split of the hard drive due to diskutil verifying drive before doing the split
- split hard drive into 2 partitions
  - 200 GB for boot partition
  - rest for userHD which is mounted via /etc/fstab at /Users (by UUID)
- allows reimaging of boot drive without affecting user's data

# Getting pieces in place - the luggage

- (re)package installers
  - Flash player
  - other applications that need customization beyond provided installer
- build printer installers

# Getting pieces in place - puppet vs munki

## puppet

Adobe Reader

Chrome

FileMaker Pro

Firefox

Flash player

Google Earth

Google SketchUp

iLife

local users

Office

puppet/facter installers

services (scripts/plists)

VLC

# Getting pieces in place - puppet vs munki

## puppet

local users

services (scripts/plists)

## munki

Adobe Reader

Chrome

FileMaker Pro

Firefox

Flash player

Google Earth

Google SketchUp

iLife

Office

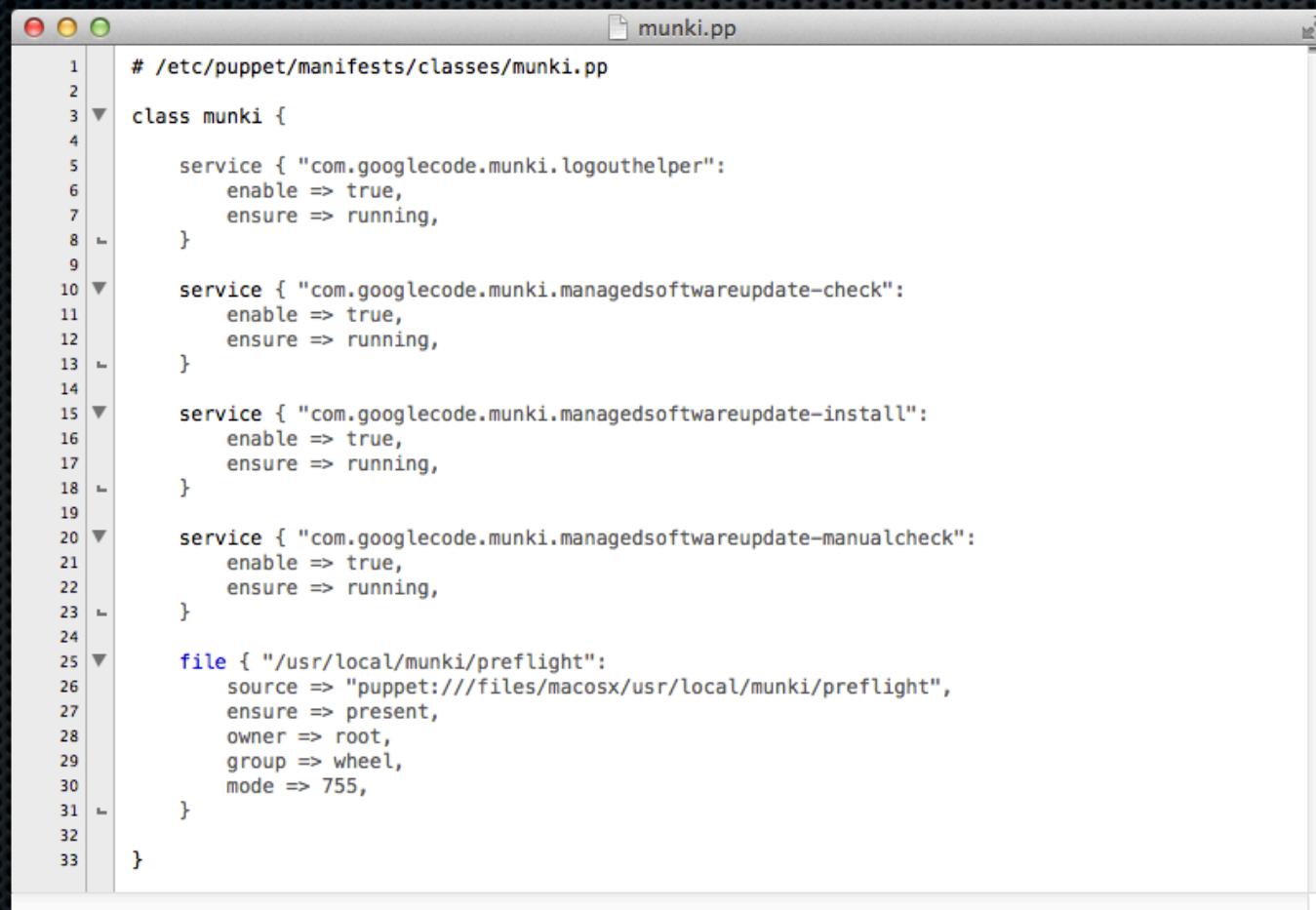
puppet/facter installers

VLC

# Getting pieces in place - puppet

- disable installs that we were moving to munki
- manage munki services and preflight script

# Getting pieces in place - puppet



A screenshot of a Mac OS X application window titled "munki.pp". The window contains a code editor displaying a Puppet manifest. The manifest defines a class named "munki" with four service resources and one file resource. The code is as follows:

```
# /etc/puppet/manifests/classes/munki.pp

class munki {

    service { "com.googlecode.munki.logouthelper":
        enable => true,
        ensure => running,
    }

    service { "com.googlecode.munki.managedsoftwareupdate-check":
        enable => true,
        ensure => running,
    }

    service { "com.googlecode.munki.managedsoftwareupdate-install":
        enable => true,
        ensure => running,
    }

    service { "com.googlecode.munki.managedsoftwareupdate-manualcheck":
        enable => true,
        ensure => running,
    }

    file { "/usr/local/munki/preflight":
        source => "puppet:///files/macosx/usr/local/munki/preflight",
        ensure => present,
        owner => root,
        group => wheel,
        mode => 755,
    }
}
```

# Getting pieces in place - puppet

- what was left for puppet to manage?
  - users (2 admins and 1 non-admin/no password student account)
  - services
    - remote logon (sshd)
    - puppet (launchd plist, service)
    - munki (launchd plists, services)
  - scripts
    - munki's preflight and restart puppet launchdaemon

# Getting pieces in place - munki

- learn... via MacTech articles, munki wiki
- what software to install and how to install it
  - managed or optional install
  - attended or unattended install

# Getting pieces in place - munki

- design manifest “groups”
  - use web help desk options for puppet & munki custom field



- build munki configuration
  - MunkiAdmin

# Getting pieces in place - munki (via MunkiAdmin)

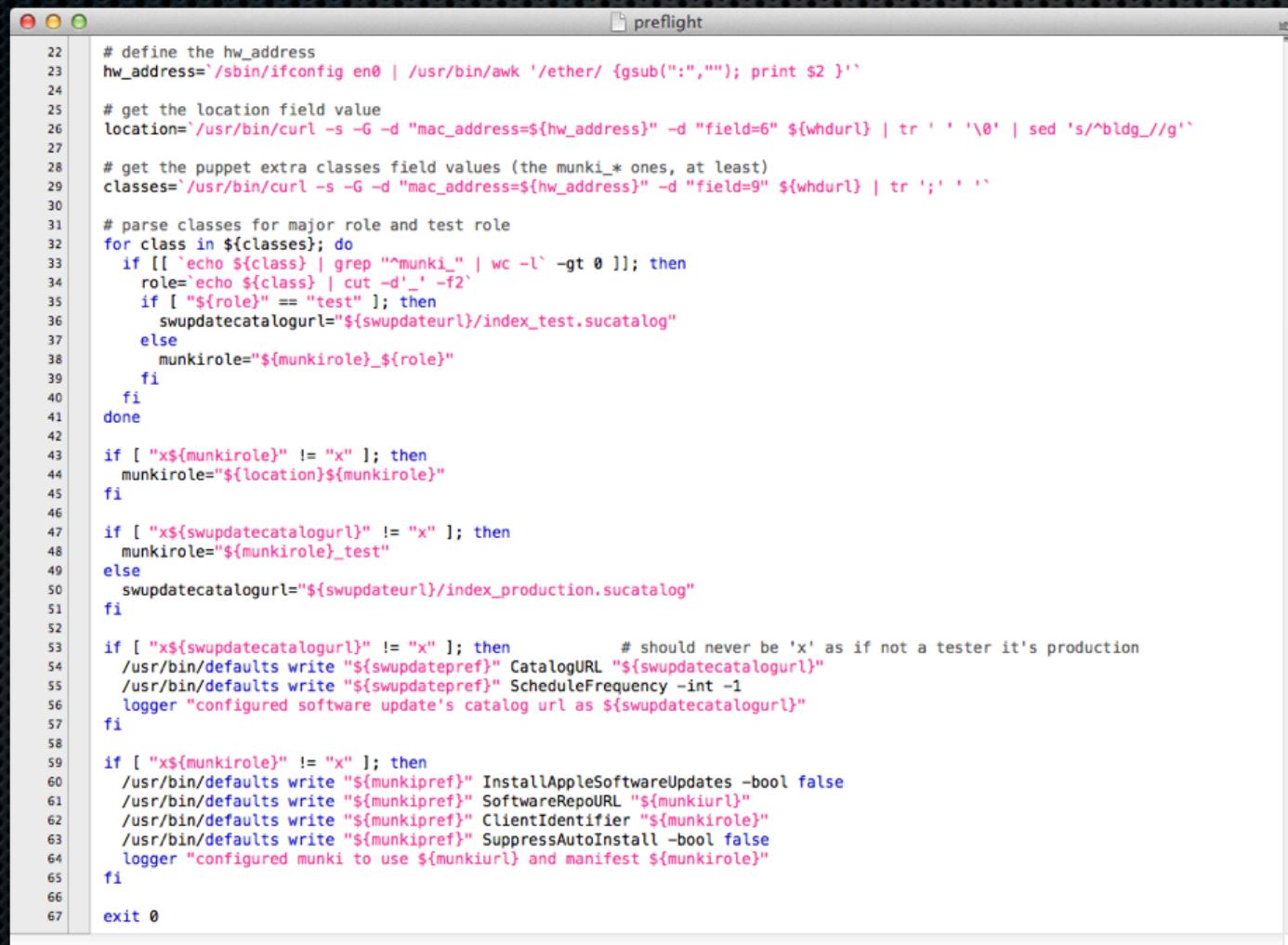
The screenshot shows the MunkiAdmin application interface. The title bar reads "MunkiAdmin – Packages". The main window has a toolbar with "View", "Open", "Save", "Reload", and "Make" buttons. On the left is a "REPOSITORY" sidebar titled "All Packages" with a tree view of installed software categories: Apps, Adobe, Apple, Epson, FileMaker, Firefox, Google, GraphicConverter, IPEVO, Java, and Lexia. The central area is a table titled "All Packages" with columns: Name, Display Name, Version, and Modified. The table lists numerous packages, mostly from the Adobe and Apple categories, such as Adobe Flash Player, Adobe Reader XI Installer, BoardMaker Plus, Chrome, and various printer drivers. To the right of the table are three summary sections: "Summary", "Pkginfos", and "Installer".

Name	Display Name	Version	Modified
Adobe Flash Player	Adobe Flash Player	11.5.502.110	4/3/13 4:39 PM
Adobe Flash Player	Adobe Flash Player	11.5.502.149	4/3/13 4:39 PM
Adobe Flash Player	Adobe Flash Player	11.6.602.167	4/3/13 4:39 PM
Adobe Flash Player	Adobe Flash Player	11.6.602.171.01	4/3/13 4:39 PM
Adobe Flash Player	Adobe Flash Player	11.6.602.180	4/10/13 8:41 AM
Adobe Flash Player	Adobe Flash Player	11.7.700.169	4/9/13 2:16 PM
Adobe Reader XI Installer	Adobe Reader XI	11.0.0	4/3/13 4:39 PM
Adobe Reader XI Installer	Adobe Reader XI	11.0.02	4/10/13 8:52 AM
BoardMaker Plus	BoardMaker Plus	6.1.6	2/4/13 4:46 PM
Chrome	Chrome	23.0.1271.64	4/3/13 4:39 PM
Chrome	Chrome	26.0.1410.43	4/10/13 8:52 AM
Education Software Inst...	SMART Notebook	11.1.7.0	3/5/13 11:36 AM
ee-hp-4e-printer	ee-hp-4e-printer	1.0	12/14/12 1:30 PM
ee-hp-2430-printer	ee-hp-2430-printer	1.0	12/14/12 1:30 PM
ee-hp-3500-printer	ee-hp-3500-printer	1.0	12/14/12 1:30 PM
ee-hp-4600-printer	ee-hp-4600-printer	1.0	12/14/12 1:30 PM
ee-hp-module-printer	ee-hp-module-printer	1.0	12/14/12 1:30 PM
ee-hp-rm9-printer	ee-hp-rm9-printer	1.0	12/14/12 1:30 PM
ee-hp-rm10-printer	ee-hp-rm10-printer	1.0	1/8/13 1:41 PM
ee-hp-rm10-printer	ee-hp-rm10-printer	1.01	1/8/13 1:41 PM
ee-hp-rm12-printer	ee-hp-rm12-printer	1.0	12/14/12 1:30 PM
ee-ipad-printer	ee-ipad-printer	1.0	12/14/12 1:30 PM
eh-library-2430-printer	eh-library-2430-printer	1.0	1/2/13 11:04 PM
eh-library-color-3500...	eh-library-color-350...	1.0	1/18/13 5:30 PM
eh-office-ricoh-copier	eh-office-ricoh-copier	1.0	1/18/13 5:30 PM
eh-progress-book-printer	eh-progress-book-pr...	1.0	1/18/13 5:30 PM
eh-ricoh-copier	eh-ricoh-copier	1.0	1/18/13 5:30 PM
eh-rm108-3500-printer	eh-rm108-3500-printer	1.0	1/30/13 10:46...
eh-rm109-1320-printer	eh-rm109-1320-printer	1.0	1/30/13 12:53 PM
eh-rm111-1300-printer	eh-rm111-1300-printer	1.0	1/30/13 12:53 PM
eh-rm114-1320-printer	eh-rm114-1320-printer	1.0	1/30/13 12:53 PM

# Getting pieces in place - munki (preflight script)

- dynamically configures manifest option (ClientIdentifier) from data in web help desk
  - combination of location and munki role fields
  - sets CatalogURL in com.apple.softwareupdate.plist and disables scheduled checks
- installed as part of DeployStudio workflow

# Getting pieces in place - munki (preflight script)



```
# define the hw_address
hw_address=`sbin/ifconfig en0 | /usr/bin/awk '/ether/ {gsub(":", ""); print $2 }'` 

# get the location field value
location=`/usr/bin/curl -s -G -d "mac_address=${hw_address}" -d "field=6" ${whdurl} | tr ' ' '\0' | sed 's/^bldg_//g'` 

# get the puppet extra classes field values (the munki_* ones, at least)
classes=`/usr/bin/curl -s -G -d "mac_address=${hw_address}" -d "field=9" ${whdurl} | tr ';' ' '` 

# parse classes for major role and test role
for class in ${classes}; do
    if [[ `echo ${class} | grep '^munki_' | wc -l` -gt 0 ]]; then
        role=`echo ${class} | cut -d'_' -f2`
        if [ "${role}" == "test" ]; then
            swupdatecatalogurl="${swupdateurl}/index_test.sucatalog"
        else
            munkirole="${munkirole}_${role}"
        fi
    fi
done

if [ "x${munkirole}" != "x" ]; then
    munkirole="${location}${munkirole}"
fi

if [ "x${swupdatecatalogurl}" != "x" ]; then
    munkirole="${munkirole}_test"
else
    swupdatecatalogurl="${swupdateurl}/index_production.sucatalog"
fi

if [ "x${swupdatecatalogurl}" != "x" ]; then      # should never be 'x' as if not a tester it's production
    /usr/bin/defaults write "${swupdatepref}" CatalogURL "${swupdatecatalogurl}"
    /usr/bin/defaults write "${swupdatepref}" ScheduleFrequency -int -1
    logger "configured software update's catalog url as ${swupdatecatalogurl}"
fi

if [ "x${munkirole}" != "x" ]; then
    /usr/bin/defaults write "${munkipref}" InstallAppleSoftwareUpdates -bool false
    /usr/bin/defaults write "${munkipref}" SoftwareRepoURL "${munkiurl}"
    /usr/bin/defaults write "${munkipref}" ClientIdentifier "${munkirole}"
    /usr/bin/defaults write "${munkipref}" SuppressAutoInstall -bool false
    logger "configured munki to use ${munkiurl} and manifest ${munkirole}"
fi

exit 0
```

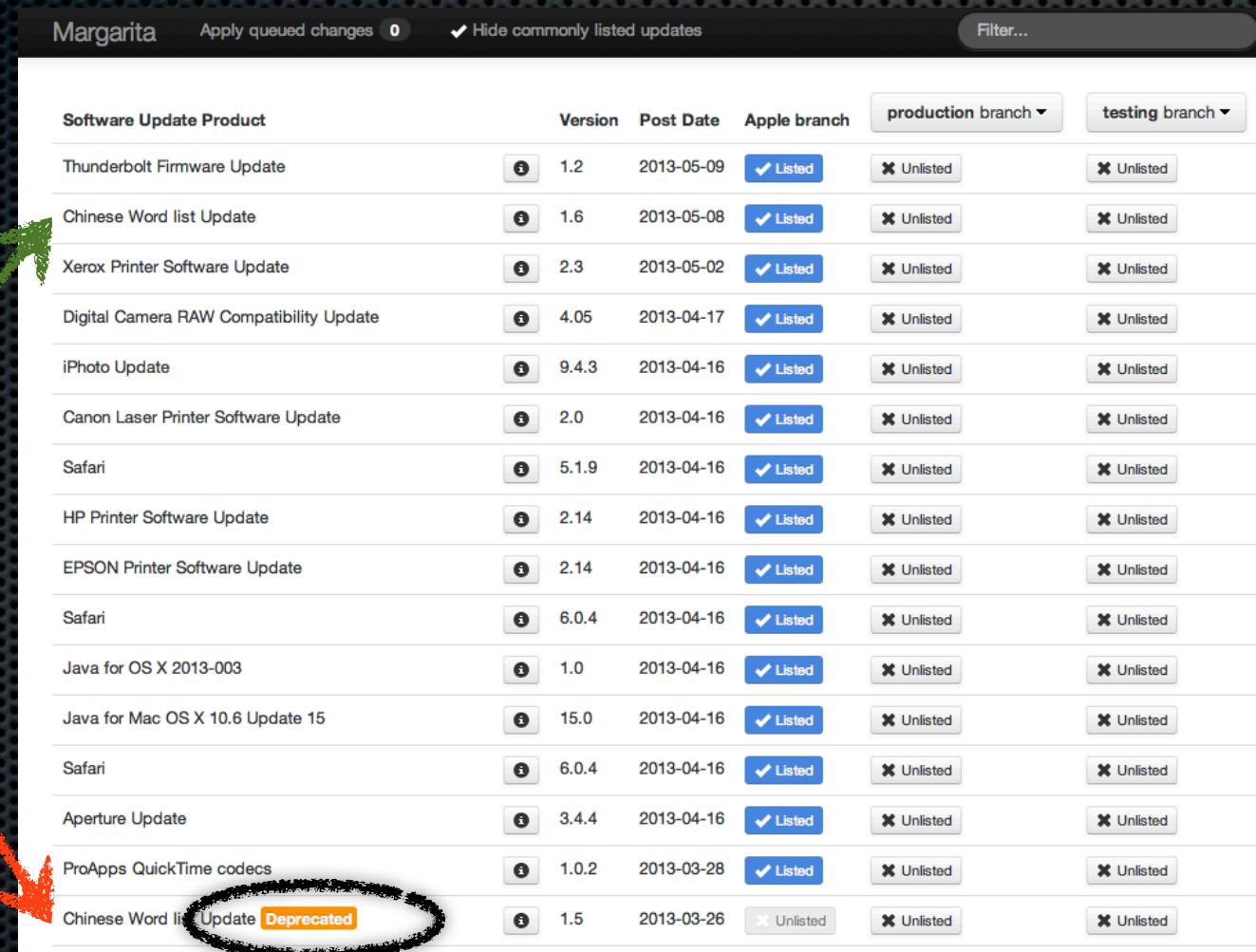
# Getting pieces in place - reposado

- want to mirror Apple software update service internally due to internet bandwidth, and have the ability to control what updates are available
- but how to manage?
  - regular command line tool (repoutil)
  - web interface

# Getting pieces in place - reposado (& margarita)

- margarita (web interface)
- no authentication though, so need to secure otherwise
  - anyone who can get to web page can modify catalogs
  - run on a non-standard port
  - work being done to get authentication working, though

# Getting pieces in place - reposado (& margarita)



The screenshot shows a list of software updates in the Margarita interface. A green arrow points upwards from the bottom left towards the top of the table, while a red arrow points downwards from the top right towards the bottom of the table. A specific row is highlighted with a black oval around the 'Update' column, which contains the word 'Deprecated'.

Software Update Product	Version	Post Date	Apple branch	production branch	testing branch
Thunderbolt Firmware Update	1.2	2013-05-09	<input checked="" type="checkbox"/> Listed	<input type="checkbox"/> Unlisted	<input type="checkbox"/> Unlisted
Chinese Word list Update	1.6	2013-05-08	<input checked="" type="checkbox"/> Listed	<input type="checkbox"/> Unlisted	<input type="checkbox"/> Unlisted
Xerox Printer Software Update	2.3	2013-05-02	<input checked="" type="checkbox"/> Listed	<input type="checkbox"/> Unlisted	<input type="checkbox"/> Unlisted
Digital Camera RAW Compatibility Update	4.05	2013-04-17	<input checked="" type="checkbox"/> Listed	<input type="checkbox"/> Unlisted	<input type="checkbox"/> Unlisted
iPhoto Update	9.4.3	2013-04-16	<input checked="" type="checkbox"/> Listed	<input type="checkbox"/> Unlisted	<input type="checkbox"/> Unlisted
Canon Laser Printer Software Update	2.0	2013-04-16	<input checked="" type="checkbox"/> Listed	<input type="checkbox"/> Unlisted	<input type="checkbox"/> Unlisted
Safari	5.1.9	2013-04-16	<input checked="" type="checkbox"/> Listed	<input type="checkbox"/> Unlisted	<input type="checkbox"/> Unlisted
HP Printer Software Update	2.14	2013-04-16	<input checked="" type="checkbox"/> Listed	<input type="checkbox"/> Unlisted	<input type="checkbox"/> Unlisted
EPSON Printer Software Update	2.14	2013-04-16	<input checked="" type="checkbox"/> Listed	<input type="checkbox"/> Unlisted	<input type="checkbox"/> Unlisted
Safari	6.0.4	2013-04-16	<input checked="" type="checkbox"/> Listed	<input type="checkbox"/> Unlisted	<input type="checkbox"/> Unlisted
Java for OS X 2013-003	1.0	2013-04-16	<input checked="" type="checkbox"/> Listed	<input type="checkbox"/> Unlisted	<input type="checkbox"/> Unlisted
Java for Mac OS X 10.6 Update 15	15.0	2013-04-16	<input checked="" type="checkbox"/> Listed	<input type="checkbox"/> Unlisted	<input type="checkbox"/> Unlisted
Safari	6.0.4	2013-04-16	<input checked="" type="checkbox"/> Listed	<input type="checkbox"/> Unlisted	<input type="checkbox"/> Unlisted
Aperture Update	3.4.4	2013-04-16	<input checked="" type="checkbox"/> Listed	<input type="checkbox"/> Unlisted	<input type="checkbox"/> Unlisted
ProApps QuickTime codecs	1.0.2	2013-03-28	<input checked="" type="checkbox"/> Listed	<input type="checkbox"/> Unlisted	<input type="checkbox"/> Unlisted
Chinese Word li	Update	Deprecated			
	1.5	2013-03-26	<input type="checkbox"/> Unlisted	<input type="checkbox"/> Unlisted	<input type="checkbox"/> Unlisted

# Getting pieces in place - taking delivery

- 1250 13" MacBook Pros (5 packs)
- 60 21.5" last model iMacs
- occupied a lot of our warehouse



# Getting pieces in place - inventory

- serial number, MAC addresses (ethernet and wifi) scanned from shipping boxes into a spreadsheet
- asset number, client name, location, wlan, munki role added
- imported spreadsheet into web help desk



# Ready to go!



DeployStudio first boot  
(less than 5 minutes)



munki bootstrap  
(hours... uh oh...)

# Back to the drawing board

- what went wrong?
  - munki and reposado repos at district NOC
  - 20 mbps WAN link
  - 10 simultaneous machines
  - ~400 MB of installation dmgs
  - 10.8.2 combo update and iTunes 10.7 (800+ MB)

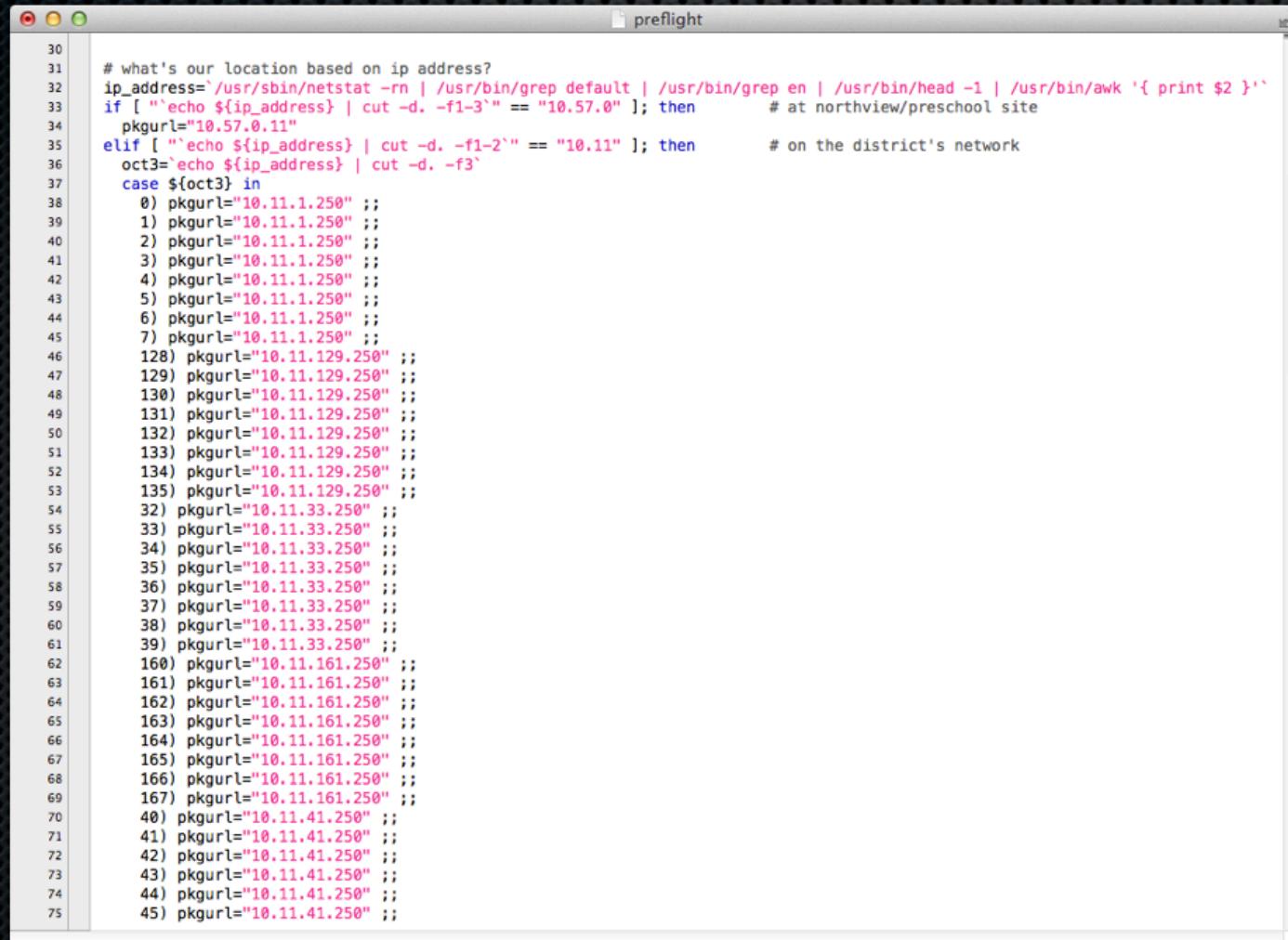
# Options

- move setup operation to high school (NOC location)
  - pro: bandwidth not an issue (local LAN)
  - con: no space for all of the new machines; have to move them twice; no time
- change configuration/mirror repo
  - pro: fairly quick fix; future-proof
  - con: half a day of imaging “wasted” getting local repository configured

# The fix

- replicate munki repo to local server
- change munki preflight script and deploy via puppet
  - set PackageURL to local server (based on subnet)
  - set InstallAppleSoftwareUpdates to false
- result: munki bootstrap = about 8-10 minutes (even with 10 machines at a time)

# The fix



```
preflight
30 # what's our location based on ip address?
31 ip_address=/usr/sbin/netstat -rn | /usr/bin/grep default | /usr/bin/grep en | /usr/bin/head -1 | /usr/bin/awk '{ print $2 }'
32 if [ "`echo ${ip_address} | cut -d. -f1-3`" == "10.57.0" ]; then      # at northview/preschool site
33   pkgurl="10.57.0.11"
34 elif [ "`echo ${ip_address} | cut -d. -f1-2`" == "10.11" ]; then      # on the district's network
35   oct3=`echo ${ip_address} | cut -d. -f3`
36   case ${oct3} in
37     0) pkgurl="10.11.1.250" ;;
38     1) pkgurl="10.11.1.250" ;;
39     2) pkgurl="10.11.1.250" ;;
40     3) pkgurl="10.11.1.250" ;;
41     4) pkgurl="10.11.1.250" ;;
42     5) pkgurl="10.11.1.250" ;;
43     6) pkgurl="10.11.1.250" ;;
44     7) pkgurl="10.11.1.250" ;;
45     128) pkgurl="10.11.129.250" ;;
46     129) pkgurl="10.11.129.250" ;;
47     130) pkgurl="10.11.129.250" ;;
48     131) pkgurl="10.11.129.250" ;;
49     132) pkgurl="10.11.129.250" ;;
50     133) pkgurl="10.11.129.250" ;;
51     134) pkgurl="10.11.129.250" ;;
52     135) pkgurl="10.11.129.250" ;;
53     32) pkgurl="10.11.33.250" ;;
54     33) pkgurl="10.11.33.250" ;;
55     34) pkgurl="10.11.33.250" ;;
56     35) pkgurl="10.11.33.250" ;;
57     36) pkgurl="10.11.33.250" ;;
58     37) pkgurl="10.11.33.250" ;;
59     38) pkgurl="10.11.33.250" ;;
60     39) pkgurl="10.11.33.250" ;;
61     160) pkgurl="10.11.161.250" ;;
62     161) pkgurl="10.11.161.250" ;;
63     162) pkgurl="10.11.161.250" ;;
64     163) pkgurl="10.11.161.250" ;;
65     164) pkgurl="10.11.161.250" ;;
66     165) pkgurl="10.11.161.250" ;;
67     166) pkgurl="10.11.161.250" ;;
68     167) pkgurl="10.11.161.250" ;;
69     40) pkgurl="10.11.41.250" ;;
70     41) pkgurl="10.11.41.250" ;;
71     42) pkgurl="10.11.41.250" ;;
72     43) pkgurl="10.11.41.250" ;;
73     44) pkgurl="10.11.41.250" ;;
74     45) pkgurl="10.11.41.250" ;;
```

# The fix

```
152     *) pkgurl="munki.northmontschools.com" ;;
153     esac
154   fi
155
156 # parse classes for major role and test role
157 for class in ${classes}; do
158   if [[ `echo ${class} | grep "munki_" | wc -l` -gt 0 ]]; then
159     role=`echo ${class} | cut -d'_' -f2`
160     if [ "${role}" == "test" ]; then
161       swupdatecatalogurl="${swupdateurl}/index_test.sucatalog"
162     else
163       munkirole="${munkirole}_${role}"
164     fi
165   fi
166 done
167
168 if [ "x${munkirole}" != "x" ]; then
169   munkirole="${location}${munkirole}"
170 fi
171
172 if [ "x${swupdatecatalogurl}" != "x" ]; then
173   munkirole="${munkirole}_test"
174 else
175   swupdatecatalogurl="${swupdateurl}/index_production.sucatalog"
176 fi
177
178 if [ "x${swupdatecatalogurl}" != "x" ]; then      # should never be 'x' as if not a tester it's production
179   /usr/bin/defaults write "${swupdatepref}" CatalogURL "${swupdatecatalogurl}"
180   /usr/bin/defaults write "${swupdatepref}" ScheduleFrequency -int -1
181   logger "configured software update's catalog url as ${swupdatecatalogurl}"
182 fi
183
184 if [ "x${munkirole}" != "x" ]; then
185   /usr/bin/defaults write "${munkipref}" SoftwareRepoURL "${munkiurl}"
186   /usr/bin/defaults write "${munkipref}" SkipInstall -bool true
187   /usr/bin/defaults write "${munkipref}" SuppressAutoInstall -bool false
188   /usr/bin/defaults write "${munkipref}" InstallAppleSoftwareUpdates -bool false
189   if [ "x${pkgurl}" != "x" ]; then
190     /usr/bin/defaults write "${munkipref}" PackageURL "http://${pkgurl}/munki/repo/pkgs"
191   else
192     /usr/bin/defaults delete "${munkipref}" PackageURL
193   fi
194   logger "configured munki to use ${munkiurl} and manifest ${munkirole}, skip: ${skip}, pkgs: ${pkgurl}"
195 fi
196
197 exit 0
```

# But... what about Apple updates?

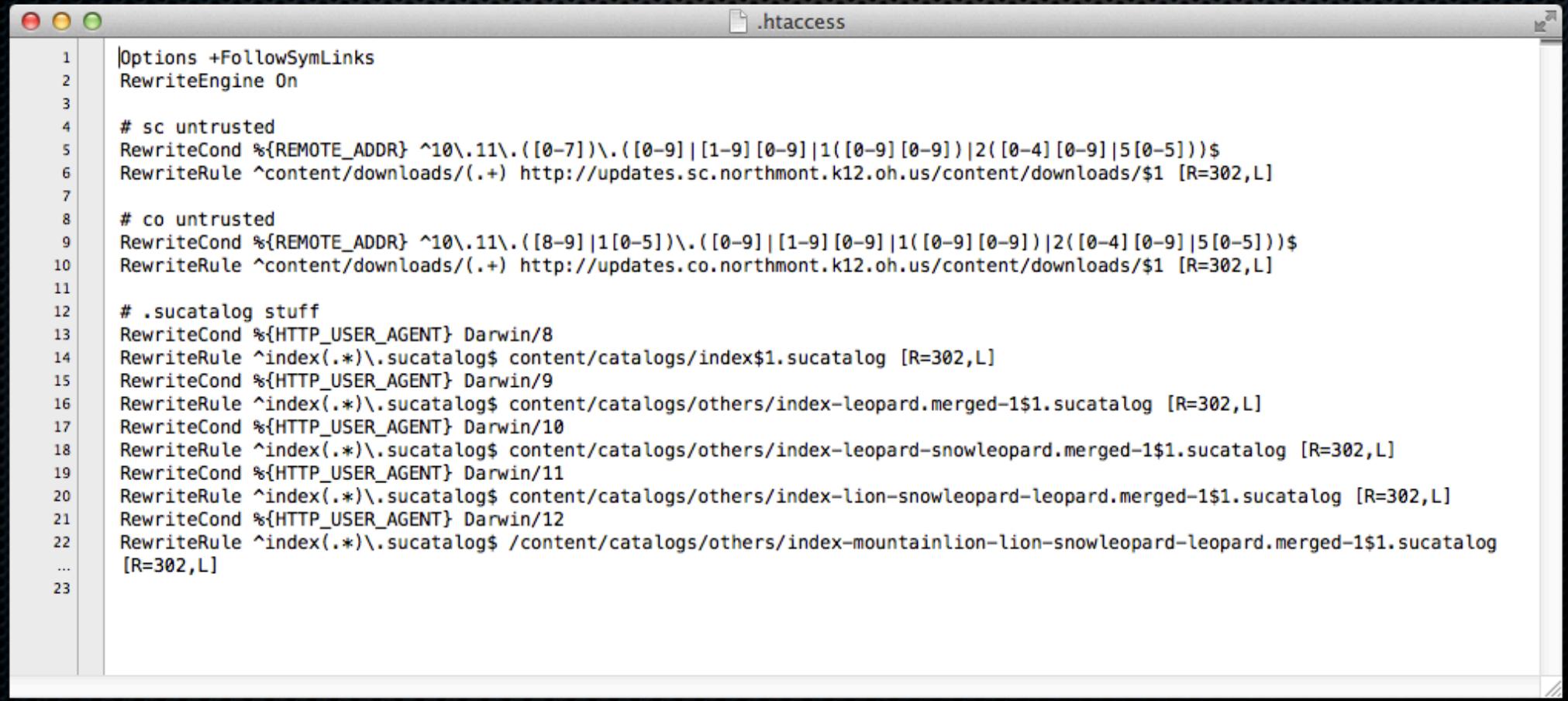
- didn't fix the Apple swupdate issue, just disabled them
- no way to specify different locations for CatalogURL and actual packages (like you can with munki's SoftwareRepoURL versus PackageURL, CatalogURL, and ManifestURL)
- solution: use apache's mod\_rewrite to redirect content/downloads (but not catalog) requests to local repo host based on ip address
- supplements OS version agnostic CatalogURL setting on client

# Multi-site reposado

- synchronize master reposado repository to local servers' web directory
  - only the reposado/html/content/downloads folder is necessary
  - use synchronization tool of choice
- why not set CatalogURL to local reposado repo URL?
  - portable machines not always awake to run puppet and munki
  - some staff move between buildings, so reconfiguration necessary prior to running softwareupdate

# Multi-site reposado

- .htaccess on master reposado server (not on clones)



A screenshot of a Mac OS X TextEdit window titled ".htaccess". The window contains a code listing for an .htaccess file. The code includes various RewriteCond and RewriteRule directives for handling different user agents and remote addresses, particularly for Darwin-based systems. The window has the standard OS X title bar with red, yellow, and green buttons.

```
1 Options +FollowSymLinks
2 RewriteEngine On
3
4 # sc untrusted
5 RewriteCond %{REMOTE_ADDR} ^10\.11\.(0-7)\.(0-9|1-9)[0-9]|1([0-9][0-9])|2([0-4][0-9]|5[0-5]))$ 
6 RewriteRule ^content/downloads/(.+)$ http://updates.sc.northmont.k12.oh.us/content/downloads/$1 [R=302,L]
7
8 # co untrusted
9 RewriteCond %{REMOTE_ADDR} ^10\.11\.(8-9|1[0-5])\.(0-9|1-9)[0-9]|1([0-9][0-9])|2([0-4][0-9]|5[0-5]))$ 
10 RewriteRule ^content/downloads/(.+)$ http://updates.co.northmont.k12.oh.us/content/downloads/$1 [R=302,L]
11
12 # .sucatalog stuff
13 RewriteCond %{HTTP_USER_AGENT} Darwin/8
14 RewriteRule ^index\.(.*).sucatalog$ content/catalogs/index$1.sucatalog [R=302,L]
15 RewriteCond %{HTTP_USER_AGENT} Darwin/9
16 RewriteRule ^index\.(.*).sucatalog$ content/catalogs/others/index-leopard.merged-1$1.sucatalog [R=302,L]
17 RewriteCond %{HTTP_USER_AGENT} Darwin/10
18 RewriteRule ^index\.(.*).sucatalog$ content/catalogs/others/index-leopard-snowleopard.merged-1$1.sucatalog [R=302,L]
19 RewriteCond %{HTTP_USER_AGENT} Darwin/11
20 RewriteRule ^index\.(.*).sucatalog$ content/catalogs/others/index-lion-snowleopard-leopard.merged-1$1.sucatalog [R=302,L]
21 RewriteCond %{HTTP_USER_AGENT} Darwin/12
22 RewriteRule ^index\.(.*).sucatalog$ /content/catalogs/others/index-mountainlion-lion-snowleopard-leopard.merged-1$1.sucatalog [R=302,L]
...
23
```



# Monitoring client machines' state

- need to know if software is updated, what version(s)
- munkiwebadmin
  - very easy to install (< 30 minutes)
  - munki preflight/postflight scripts
  - shows installed applications' version
  - add-on script includes internet plug-ins, too!
  - use puppet to deploy preflight/postflight scripts
    - didn't include directly in pre/postflight scripts, but rather as standalone scripts that get called

# Munkiwebadmin

The screenshot shows a Mac OS X desktop environment with a dark grey textured background. A web browser window titled "MunkiWebAdmin" is open, displaying the "report/overview" page. The address bar shows the URL "munki.northmontschools.com/report/overview". The top menu bar includes standard OS X icons for file, edit, and search, along with links to "Apple", "iCloud", "Facebook", "Twitter", "Wikipedia", "Yahoo!", "News", "Popular", and "dvr". On the right side of the window, there are "Reader" and "Download" buttons, and a "+" button for new tabs.

The main content area is titled "MunkiWebAdmin". Below it, there are three sections with counts: "Errors 46", "Warnings 1271", and "Activity 518".

Machine	User	IP	Latest Run
<a href="#">nwprincipal-23829</a>	mrichardson	10.11.43.141	2013-05-14 08:24 2 pending installs
<a href="#">or28-23244</a>	acozad	10.11.211.38	2013-05-14 08:24 1 pending install
<a href="#">nm104-22506</a>	mschreiber	10.11.195.42	2013-05-14 08:24 2 pending installs
<a href="#">nm216-22695</a>	awhite	10.11.195.56	2013-05-14 08:24 4 pending installs
<a href="#">ms117-23567</a>	sryan	10.11.148.233	2013-05-14 08:24 2 pending installs
<a href="#">studentservices-23863</a>	afolkening	10.11.170.78	2013-05-14 08:23 3 pending installs
<a href="#">studentservices-23891</a>	tforest	10.11.154.88	2013-05-14 08:23 5 pending installs
<a href="#">hs-rm103-18195</a>	ktröhler	10.11.27.194	2013-05-14 08:23 1 package installed
<a href="#">ms219-23624</a>	amcguinness	10.11.149.12	2013-05-14 08:22 3 pending installs
<a href="#">studentservices-23864</a>	scolucci	10.11.211.70	2013-05-14 08:22 1 pending install
<a href="#">studentservices-23906</a>	tdisabatino	10.11.154.34	2013-05-14 08:20 3 pending installs

At the bottom of the window, there is a status bar with the text "Open \"http://munki.northmontschools.com/lookup/detail/a8:20:66:1a:72:55\" in a new tab".

# Munkiwebadmin

The screenshot shows a web browser window titled "MunkiWebAdmin" with the URL "munki.northmontschools.com/inventory/items?name=Flash%20Player&version=11.6.602.171". The browser's toolbar includes standard icons for back, forward, search, and refresh, along with links to Apple, iCloud, Facebook, Twitter, Wikipedia, Yahoo!, News, Popular, and dvr.

The main content area is titled "Flash Player" and displays a table of inventory items. The table has columns for "Machines" (4), "Show 20 entries", "Search: flash player", and a list of versions with their counts:

Version	Count
9.0.45.0	1
11.7.700.169	1288
11.6.602.180	26
11.6.602.171	4
11.5.502.149	1
11.5.502.110	1
10.1.102.64	2

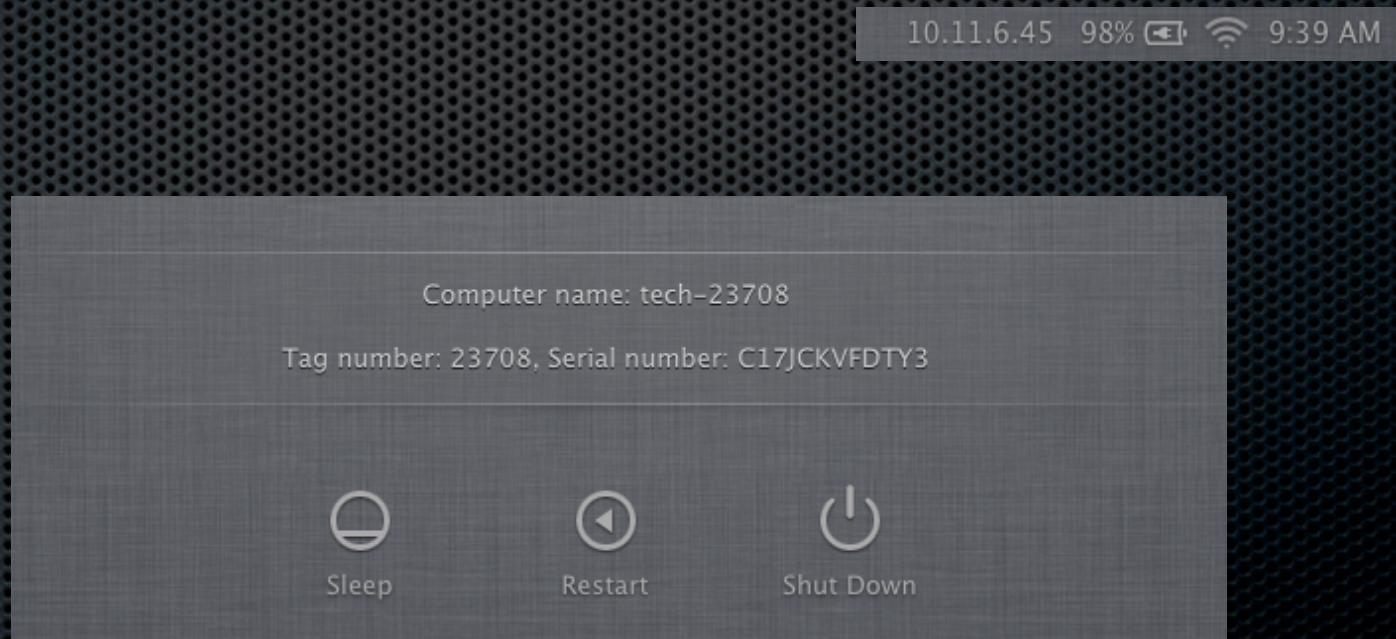
Below the table, there is a detailed view of two entries:

Machine	Version	Type	Path
23107	11.6.602.171	Player.plugin	/Library/Internet Plug-Ins/Flash Player.plugin
orcarr3-23357	com.macromedia.Flash Player.plugin	Flash Player	/Library/Internet Plug-Ins/Flash Player.plugin

At the bottom, it says "Showing 1 to 4 of 4 entries" and includes navigation buttons for "← Previous", "1", and "Next →".

# Changes since initial deployment

- login window info (replaces cycling info area pre-10.7), deployed via puppet in updated machine naming script



# Changes since initial deployment

- ARD privileges rollback
  - DeployStudio's enable ARD step enables for all local users... and we have a non-admin, no password account
- printer administration - lpadmin group
  - staff can add printers at home on their assigned machine
  - any staff member can resume printing on any machine

# Changes since initial deployment

- lots of application updates
  - flash player, adobe reader, java mainly
- vpn configuration profile
  - launchagent to ensure vpn menu item is loaded
- we're a school district and schools love to make end of the year videos on DVD

# Changes since initial deployment

- ❖ “critical” mode for munki updates (for attended installs)
  - ❖ turn “bug the heck out of the users” mode on/off
  - ❖ file on munki server is the controller

# Changes since initial deployment

```
munkirole="${munkirole}_${role}"
fi
fi
done

if [ "x${munkirole}" != "x" ]; then
  munkirole="${location}${munkirole}"
fi

if [ "x${swupdatecatalogurl}" != "x" ]; then
  munkirole="${munkirole}_test"
else
  swupdatecatalogurl="${swupdateurl}/index_production.sucatalog"
fi

if [ "x${swupdatecatalogurl}" != "x" ]; then      # should never be 'x' as if not a tester it's production
  /usr/bin/defaults write "${swupdatepref}" CatalogURL "${swupdatecatalogurl}"
  /usr/bin/defaults write "${swupdatepref}" ScheduleFrequency -int -1
  logger "configured software update's catalog url as ${swupdatecatalogurl}"
fi

if [ "x${munkirole}" != "x" ]; then
  /usr/bin/defaults write "${munkipref}" SoftwareRepoURL "${munkiurl}"
  /usr/bin/defaults write "${munkipref}" ClientIdentifier "${munkirole}"
  /usr/bin/defaults write "${munkipref}" SuppressAutoInstall -bool false
  /usr/bin/defaults write "${munkipref}" InstallAppleSoftwareUpdates -bool false
  if [ "x${pkgurl}" != "x" ]; then
    /usr/bin/defaults write "${munkipref}" PackageURL "http://${pkgurl}/munki/repo/pkgs"
  else
    /usr/bin/defaults delete "${munkipref}" PackageURL
  fi
  if [ "curl ${munkiurl}/critical" == "xYES" ]; then
    logger "critical flag is set... changing preferences to bug user"
    /usr/bin/defaults write "${munkipref}" SuppressUserNotification -bool false
    /usr/bin/defaults write "${munkipref}" DaysBetweenNotifications 0
    /usr/bin/defaults write "${munkipref}" SuppressStopButtonOnInstall -bool true
  else
    logger "critical flag is not set... changing preferences to nice mode"
    /usr/bin/defaults write "${munkipref}" SuppressUserNotification -bool true
    /usr/bin/defaults write "${munkipref}" DaysBetweenNotifications 1
    /usr/bin/defaults write "${munkipref}" SuppressStopButtonOnInstall -bool false
  fi
  logger "configured munki to use ${munkiurl} and manifest ${munkirole}, pkgs: ${pkgurl}"
fi

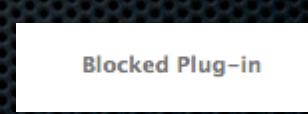
exit 0
```

# What's left

- nested manifests in munki
  - most manifests contain same applications
  - modify one manifest instead of 30+
- update munki preflight script to reenable installing Apple software updates
- set some users up as “test” users and do early app deployments to that group
  - set via checkbox in web help desk

# What's left

- put munki and reposado in the dmz
  - school is almost out
  - users won't be "on campus" but will get xprotect updates
- uh oh...



So you probably want to see the physical result of replacing 1500+ iBooks, eMacs, Mac Minis, and iMacs with these new machines, right?

# The results



13" MacBook Rungs 10 miling.50.8



when eMacs go out to pasture



ready to go over the rainbow

# Questions?

# Supplemental info

- blog article with links to projects, code, etc.
  - <http://seankaiser.com/redirects/psumacadmins2013/>
- contact me
  - @seankaiser
  - seankaiser.com