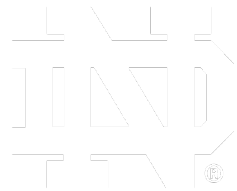


Case Study: Thin OS X Deployment with Munki

Clarence Helm & Matt Willmore
College of Arts & Letters
University of Notre Dame



Agenda

1. Background Information
2. Problem Set
3. Desired State
4. Solution & Implementation
5. Recommendations
6. Resources
7. Q&A





Who We Are

Clarence Helm

- ④ IT Support Consultant in the Office of Information Technologies at the University of Notre Dame. (1 year)
- ④ M.S. in Management Information Technologies from Indiana University
- ④ 2009 to 2012 - data acquisitions and data management while working on a grant-funded biometrics research project at Notre Dame (<http://www3.nd.edu/~ccl/research/pubs/bxgrid-mdqcs10.pdf>)



Who We Are

Matt Willmore

- IT Support Engineer in Office of Information Technology, supporting staff & faculty in College of Arts and Letters (3 years)
- B.S. and M.S. from Purdue University in Organizational Leadership and Supervision



Background Information



Background Information

Customer

- Arts & Letters Computing supports approx. 800 Macs within the College (1,800 systems total)
- 6 techs total; most are Windows-centric
- 99% are individual users, roughly half are laptops
- No bulk buys; tied to hiring and multiple refresh cycles
- Little hardware standardization



Background Information

Technology

- ④ A&L is part of central IT group, but each College has very different requirements and user bases
- ④ Using a blend of centralized and decentralized technology
 - ④ Centralized: AD, LDAP, McAfee ePO; Quest QAS/QMX for authentication, group policies, inventory, compliance, some software distro
 - ④ Central IT makes an OS X image available
 - ④ Missing College-specific resources
 - ④ Slow to update for interstitial builds



Background Information

Where We Were

- Entirely based on monolithic images
- Waited for "reference release" of OS
 - Interstitial releases were very uncomfortable
- Used InstaDMG to build factory-fresh OS
- Used existing Munki repository for major software installers, manual installation/configuration for everything else
- Able to manage post-installation with Munki
- Much better place than in 2010



Problem Set



Problem Set

- ⑤ "You can't use Munki anymore."
- ⑤ Interstitial builds still an issue
- ⑤ Replacement for Munki (Quest/SCCM)
not meant for thin imaging
- ⑤ Only one team member with know-how
- ⑤ Long time to distribute new images



Desired State



Desired State

1. Stop using Munki for managed desktop
2. Start using Quest/SCCM/GPOs for managed desktop
3. Eliminate pain associated with interstitial builds
4. Distribute build knowledge and ability
5. Shorten/eliminate time to distribute new images (critical for interstitials)
6. Construct a build framework that will outlive our time in ALCO



Solution & Implementation



Solution & Implementation

Stop Using Munki for Managed Desktop

Answer: Migrate to Quest/SCCM/GPOs for desktop management (inventory, compliance, security-related software patches (Java, Flash Player, etc.)

- ⑤ Notre Dame rolled out Quest QAS and QMX in July 2010
- ⑤ QAS for authentication and policy application
- ⑤ QMX (SCCM snap-in) for compliance, inventory, and software distribution



Solution & Implementation

Eliminate Interstitial Build Pain

Answer: Build a framework that allows anyone to quickly build any Mac out of the box, and use that for a monolithic image if desired



Solution & Implementation

Distribute Knowledge

Answer: Document everything on our internal wiki

- ④ `munkiimport` output for each item in munki
- ④ Dependencies
- ④ `pkginfo` data for each item in munki
- ④ Any additional helpful information



Solution & Implementation

Shorten Distribution Time

Answer: combination of new thin build framework and changing workflows around the process



Solution & Implementation

Create a New Framework

Answer: Munki, of course!

So...how did we do this?

(Why you're sitting in this session)



Solution & Implementation

Thin/Thick Build Workflow

1. Identify everything to be installed on the Mac
 - a. Applications and registration
 - b. Setting changes
 - c. Drivers
 - d. System utilities



Solution & Implementation

Thin/Thick Build Workflow

2. Get everything into Munki
 - a. Drag and drop packages
 - b. Standard packages – little/no extra work
 - i. iLife '11, Adobe CS6, iWork '09, Office 2011, etc.
 - c. Items needing packaging or repackaging
 - d. Payload-free packages, such as...
 - i. Unhook preinstalled iLife apps from App Store
 - ii. Delete quarantine xattr from everything in /Applications
 - iii. Allow all admins to unlock from screensaver or sleep
 - e. Around 70 packages total



Solution & Implementation

Thin/Thick Build Workflow

3. Add everything to a "10.8-bootstrap" manifest
4. Create a .pkg installer with PackageMaker that installs:
 - a. munki
 - b. munki preferences (URL, manifest, etc.)
 - c. Desktop shortcut for Managed Software Update (we called it "Install ND Software")
 - d. .command file on desktop that erases all traces of installation including itself



Solution & Implementation

RemoveMunki.command

```
#!/bin/bash
echo "This script will remove all traces of Managed Software Update and Munki from the computer."

# Unloads LaunchDaemon
echo "Unloading LaunchDaemon..."
sudo launchctl unload /Library/LaunchDaemons/com.googlecode.munki.*

# Main application
echo "Deleting Managed Software Update.app..."
sudo rm -rf "/Applications/Utilities/Managed Software Update.app"

# Supporting files
echo "Deleting support files..."
sudo rm -f /Library/LaunchDaemons/com.googlecode.munki.*
sudo rm -f /Library/LaunchAgents/com.googlecode.munki.*
sudo rm -rf "/Library/Managed Installs"
sudo rm -rf /usr/local/munki

# Installer receipts
echo "Deleting package receipts..."
sudo pkgutil --forget com.googlecode.munki.admin
sudo pkgutil --forget com.googlecode.munki.app
sudo pkgutil --forget com.googlecode.munki.core
sudo pkgutil --forget com.googlecode.munki.launchd

# Alias to Managed Software Update
echo "Deleting desktop alias..."
sudo rm /Users/ndadmin/Desktop/Install\ ND\ Software

# This script
echo "Deleting myself..."
sudo rm /Users/ndadmin/Desktop/RemoveMunki.command

echo "Done!"

exit 0
```



Solution & Implementation

10.8-bootstrap manifest

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>catalogs</key>
  <array>
    <string>10.8-bootstrap</string>
    <string>production-lion</string>
    <string>production</string>
  </array>
  <key>included_manifests</key>
  <array>
    <string>adobe_cs6-10.8-bootstrap</string>
    <string>10.8-printers</string>
    <string>10.8-ilifel1</string>
  </array>
  <key>managed_installs</key>
  <array>
    <!-- APPLICATION INSTALLERS -->
    <string>iWork09</string>
    <string>Adobe Flash Player</string>
    <string>AdobeReader</string>
    <string>Chrome</string>
    <string>CiscoCAA</string>
    <string>CiscoCAAgentStartup</string>
    <string>CiscoVPN</string>
    <string>CiscoJabber</string>
    <string>ESCMenueExtra</string>
    <string>ESCSecTool</string>
    <string>Firefox</string>
    <string>Flip4Mac</string>
    <string>IdentityFinder</string>
    <string>Java</string>
    <string>KeyAccess</string>
    <string>McAfeeVS</string>
    <string>MSOffice2011SP2</string>
    <string>MSOffice2011_Prefs</string>
  </array>
</dict>
</plist>
```



Solution & Implementation

10.8-bootstrap manifest

(continued...)

```
<string>PlistBuddy</string>
<string>QuestQAS</string>
<string>Silverlight</string>
<string>Skype</string>
<string>TextWrangler</string>
<string>VLC</string>

<!-- SYSTEM SETTING CHANGES -->
<string>10.8-ALCOBuild</string>
<string>10.8-AutomountCorpFS</string>
<string>10.8-AutomountNetfile</string>
<string>10.8-ALCOBoot</string>
<string>10.8-UnlockScreensaver</string>
<string>10.8-RemoveQuarantineXattr</string>
<string>10.8-NDUserDefaults</string>
<string>10.8-NDAdminDefaults</string>
<string>10.8-iLifeRemoveMAS</string>
<string>10.8-HideDirectories</string>
<string>10.8-CUPS</string>
<string>TrustedCerts</string>
</array>
</dict>
</plist>
```



Solution & Implementation

User Workflow

1. Unbox new Mac
2. Create admin account; username and password are all that matters
3. Install bootstrap package and reboot
4. Run Managed Software Update, reboot as necessary
5. Run Software Update, reboot as necessary



Solution & Implementation

User Workflow

6. If this is a thin build, run our build script
 - a. Gets info on end user
 - b. Binds to AD
 - c. Installs appropriate printers
(based on LDAP query)
 - d. Saves build report to network share
 - e. Triggers FileVault and escrows recovery key
(if laptop)
7. ...if not thin build, tech reboots from boot drive,
captures with DeployStudio and updates DS workflow



Solution & Implementation

No new hardware?

1. Boot from external drive running intended OS (eg. if you want a 10.8 image, boot volume should be on 10.8 as well)
2. Use InstaDMG to create never-booted image from InstallESD.dmg (within Install OS X Mountain Lion.app)
3. Restore new image to Mac, run Software Update to make sure it's at the latest version, then install bootstrap package and proceed
4. Also check out `createOSXinstallPkg` (link at end of slides)



Solution & Implementation

User Workflow – Munki

- ⑤ Used nested manifests to group common elements and simplify main manifest (Adobe CS6, iLife11, printer drivers, etc.)
- ⑤ Installed pre-configured ManagedInstalls.plist because we know ahead of time which manifest we'll be using
- ⑤ Hint: also change energy settings to never sleep; Munki does not preempt sleep on download or install
 - ⑤ `sudo pmset -a sleep 0`



Solution & Implementation

User Workflow

- ④ Before capturing, do a walkthrough to make sure all is working as expected
- ④ Make any last-minute changes, clear out recent items, etc.
- ④ Boot from boot volume with DeployStudio and capture image; DS will auto-capture recovery partition
- ④ Test by deploying onto 2nd Mac, make any necessary tweaks
- ④ Always use the newest hardware you have, but beware of interstitial builds that won't work on other hardware
 - ④ Monolithic builds will not help here, but that's where the thin builds come in!



Solution & Implementation

Deployment Drives

- ⑤ Not using central DeployStudio server; each boot drive has DeployStudio server running at localhost
- ⑤ Faster to use Carbon Copy Cloner to clone old to new drive with changes
- ⑤ Also allows you to update entire drive with OS updates, updated standalone installers, etc.
- ⑤ Have enough drives that you can trade an "old" for "new" on the spot



Recommendations



Recommendations

- Take a close look at resources
 - Capital, time, people, infrastructure, training and depth/breadth of knowledge
- Who benefits from this?
- Look at both user and admin needs - what are people's pressure points?
 - "I can't give up my computer for that long."
 - "My Mac is here on campus, why can't I have it yet?"
 - "I want to upgrade to <insert OS version>, is it easy?"

 - "Rebuilds take my whole day."
 - "If Apple releases another one-off OS build, I'll take this bat..."
 - "I feel like I am constantly making new images."



Recommendations

- ⑤ Document, document, document
 - ⑤ What happens if you're gone tomorrow?
- ⑤ Use the built-in community and add to it
- ⑤ Stress-test your tools
 - ⑤ Timed exercises, fault analysis, ask honest questions about limitations
- ⑤ Find a user known for breaking things and have them be your guinea pig
 - ⑤ Good opportunity to build relationship with your users



Resources



Resources

- MacEnterprise – <http://www.macenterprise.org>
 - Mailing List – <http://www.macenterprise.org/ mailing-list>
- AFP548 – <http://afp548.com/>
- IRC – <irc.freenode.net> – ##osx-server
- createOSXInstallPkg – <http://managingosx.wordpress.com/2012/07/25/son-of-installion-pkg/>
- InstaDMG – <http://code.google.com/p/instadmg/>
 - Forums – <http://afp548.com/forum/index.php?forum=45>
- Munki – <http://code.google.com/p/munki/>
 - munki-dev – <http://groups.google.com/group/munki-dev>
- DeployStudio – <http://www.deploystudio.com/>
 - <http://deploystudio.com/Forums/>



Resources

- MacIT 2010 – talks on Munki and Simian, DeployStudio and OS X administration are all fantastic – <http://www.macenterprise.org/macworld-2010-slides>
- All the talks this week



Q&A



Thanks

