

# More Munki Tricks



Nate Walck  
@natewalck  
[nate@osxadm.in](mailto:nate@osxadm.in)

What's New?

# MunkiWebAdmin

First Commit August 21, 2012

- Django-based (Python) and OS-Agnostic
- Useful for Admins and Support Staff alike
- Provides Reports and Manifest editing
- “Official” Web App for Munki

# nopkg installs

added in 0.8.3.1634

- You can “install” something without providing a package
- Great for printers, settings and more

# Printer Management

nopkg

- Install drivers first via 'requires' key
- See <http://goo.gl/HhzGp> for complete pkginfo
- See <http://goo.gl/4eu6S> for printer script and alt method

# Apple Updates

Now with more options

- All Updates install in one run (as of 0.8.4.1770)
- Override `blocking_applications` and `RestartAction`
- Apply as `unattended_install` and/or `force_install_after_date`

# Creating the pkginfo

## Apple Updates

```
makepkginfo --apple_update=zzzz041-9792  
--unattended_install --force_install_after_date  
2013-05-22T13:45:00Z
```

# Example pkginfo

## Apple Updates

```
<dict>
  <key>catalogs</key>
  <array>
    <string>testing</string>
  </array>
  <key>force_install_after_date</key>
  <string>2013-05-22T13:45:00Z</string>
  <key>installer_type</key>
  <string>apple_update_metadata</string>
  <key>name</key>
  <string>zxxx041-9792</string>
  <key>unattended_install</key>
  <true/>
  <key>version</key>
  <string>1.0</string>
</dict>
```



Conditionals, how do  
they work?

# OS Version

```
<key>conditional_items</key>
<array>
  <dict>
    <key>condition</key>
    <string>os_vers BEGINSWITH "10.7"</string>
    <key>optional_installs</key>
    <array>
      <string>OS X Mountain Lion</string>
    </array>
  </dict>
</array>
```

# Catalog Filtering

```
<key>conditional_items</key>
<array>
  <dict>
    <key>condition</key>
    <string>catalogs CONTAINS "testing"</string>
    <key>managed_installs</key>
    <array>
      <string>NewSoftware</string>
    </array>
  </dict>
</array>
```

# Date Conditional

added in 0.8.4.1770

```
<key>conditional_items</key>
<array>
  <dict>
    <key>condition</key>
    <string>date > CAST("2013-05-22T13:25:00Z", "NSDate")</string>
    <key>managed_installs</key>
    <array>
      <string>Opera</string>
    </array>
  </dict>
</array>
```

# Compound

```
<key>conditional_items</key>
<array>
  <dict>
    <key>condition</key>
    <string>machine_type == "laptop" AND os_vers
BEGINSWITH "10.8"</string>
    <key>managed_installs</key>
    <array>
      <string>CiscoAnyConnectVPN</string>
    </array>
  </dict>
</array>
```

# Nested

```
<key>conditional_items</key>
<array>
  <dict>
    <key>condition</key>
    <string>machine_type == "laptop"</string>
    <key>conditional_items</key>
    <array>
      <dict>
        <key>condition</key>
        <string>os_vers BEGINSWITH "10.8"</string>
        <key>managed_installs</key>
        <array>
          <string>CiscoAnyConnectVPN</string>
        </array>
      </dict>
      <dict>
        <key>condition</key>
        <string>os_vers BEGINSWITH "10.6"</string>
        <key>managed_installs</key>
        <array>
          <string>CiscoVPNClient</string>
        </array>
      </dict>
    </array>
  </dict>
</array>
```

# Installable Condition

added in 0.8.3.1679

```
<key>installable_condition</key>  
<string>machine_model MATCHES "MacBookPro10,[12]"</string>
```

Or maybe...

```
<key>installable_condition</key>  
<string>os_vers MATCHES '10\.(7\.[5-9]|8\.[2-9])'</string>
```

# Roll your own Condition

- Create a script that writes out a key/value pair of the desired information
- Write the values to `/Library/Managed Installs/ConditionalItems.plist`
- Place script in `/usr/local/munki/conditions`
- See <http://goo.gl/oTuRY> for examples



# Managing Settings

- Use a real settings management system instead (seriously)
- But if you must, here are some guidelines...

# Local MCX

- Preferred Method
- See Enterprise Mac Managed Preferences ([goo.gl/8wPjb](http://goo.gl/8wPjb)) for details on using MCX Locally

# Profile

- Another Preferred Method
- Package Template - <http://goo.gl/B7BPP>  
(Luggage)
- Import package into munki

# Profile

- Add this installcheck\_script:

```
#!/bin/bash
```

```
profile_identfier='org.macenterprise.DontSaveNewDocumentsToiCloud'  
profile_check=$(profiles -P | grep $profile_identfier)
```

```
if [ "${profile_check}" == "" ] ; then
```

```
    exit 0
```

```
else
```

```
    exit 1
```

```
fi
```

# Defaults

- Don't do this
- No really, don't do it
- In fact, forget I even mentioned it...

# systemsetup

- NTP Server
- Time Zone
- Power Settings

# Time Zone

```
<key>installcheck_script</key>  
<string>#!/bin/bash  
timeserver='time.company.com'  
systemsetup -getnetworktimeserver | awk '{print $4}' | grep -vq  
"${timeserver}"</string>
```

```
<key>postinstall_script</key>  
<string>#!/bin/bash  
timeserver='time.company.com'  
systemsetup -setnetworktimeserver "${timeserver}"  
</string>
```

Full pkginfo: [goo.gl/RrUiM](http://goo.gl/RrUiM)

# Community-made Tools

One big happy family



# MunkiServer

- Complete Server Solution for Munki
- All-in-one Web Interface
- Can't import existing Munki repo into MunkiServer
- On Github: [goo.gl/rCvYE](https://github.com/voxpuppets/munkiserver)

# MunkiAdmin

- Not fond of the command line?
- Cocoa App for managing a Munki Repo
- GUI-based view of your repo
- On Github: [goo.gl/aLCHe](https://github.com/leeches/munkiadmin)

# MunkiReport-php

- PHP-based (kind of obvious)
- Focused on Reporting
- Easier to Setup than MunkiWebAdmin
- If you know PHP, easier to add custom inventory items than on MunkiWebAdmin
- On Google Code: [goo.gl/WRRFb](http://goo.gl/WRRFb)

# automunki

- Can be used with a Continuous Integration system (Jenkins, etc)
- Written in Bash and Ruby
- On Github: [goo.gl/uwk0u](https://goo.gl/uwk0u)

# autoMunkilImporter

- Can run standalone via launchd
- Written in Perl
- On Github: [goo.gl/ZHeiw](https://github.com/ZHeiw)

# Helpful Resources

- [munki.googlecode.com](https://munki.googlecode.com)
- munki-dev mailing list
- ##osx-server on irc.freenode.net
- [github.com](https://github.com)