



**munki**

# Managed Software Updates

Application Deployment and Management with Munki

# Who is this guy?

Nate Walck  
Apple Image Coordinator  
Liberty University

Twitter: @natewalck  
irc: natewalck  
Email: nate@osxadm.in

# Client Management with Munki

- Install Methods
- Common Software Installs and Updates
- Bootstrap Deployments
- Advanced
- Q&A

# Install methods

- Typically set in the Manifest file on the repository
- Apple Update related settings are done in ManagedInstalls.plist on the client

# Apple Software Updates

- Downloads updates using softwareupdate.
- Installs using Munki's own mechanism
- Can pull from Apple's own servers or an Apple Software Update Server
- Can install Only Apple Updates - No munki Repo required.
- No admin rights required for end-users!

# Apple Software Updates

- Set in /Library/Preferences/ManagedInstalls.plist (Or via MCX)
- Turn on Apple Software Updates via Munki:

```
<key>InstallAppleSoftwareUpdates</key>  
<true/>
```

- To do **only** Apple Software Updates (No munki repo required)

```
<key>AppleSoftwareUpdatesOnly</key>  
<true/>
```

- Specify your internal Software Update Server (Optional)

```
<key>SoftwareUpdateServerURL</key>  
<string>http://YourSUS.company.com:8088/index.sucatalog</string>
```

# Managed\_Installs

- Installs software if not already installed
- If there are updates for a managed\_installs item, those will be installed as well
- Good for installing standard software sets
- Useful for Bootstrapping machines

# Managed\_Installs

```
<key>managed_installs</key>  
<array>  
  <string>CiscoAnyConnect</string>  
  <string>Firefox</string>  
  <string>FlashPlayer</string>  
  <string>Flip4Mac WMV</string>  
  <string>KeyAccess</string>  
  <string>munkitools</string>  
  <string>Office2011</string>  
</array>
```



# Managed\_Updates

- Used to update existing software
- Will only install if software is present
- Useful for updating software that not every client has
- Allows you to update software end-users might have installed themselves

# Managed\_Updates

```
<key>managed_updates</key>  
<array>  
  <string>Office2008_update</string>  
  <string>Office2011_update</string>  
  <string>AcrobatPro9Update</string>  
  <string>AcrobatXProUpdate</string>  
  <string>CommunicatorUpdate</string>  
  <string>Firefox</string>  
</array>
```

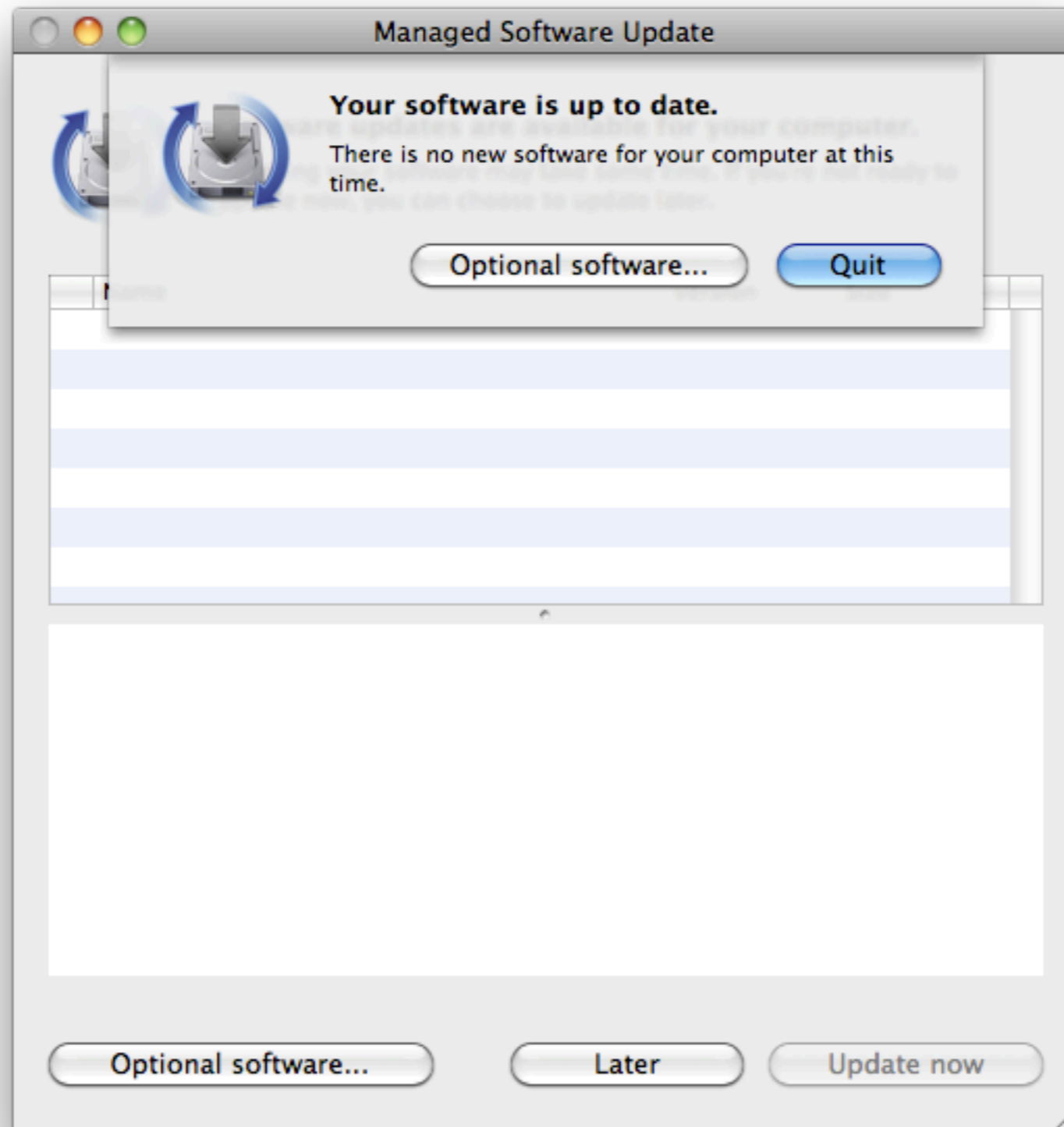
# Optional\_Installs

- Displays a list of available software to the end user
- Allows non-admin end users to install and remove specified software
- Set by the admin in the client's manifest
- Managed\_Installs and Managed\_Uninstalls take precedence

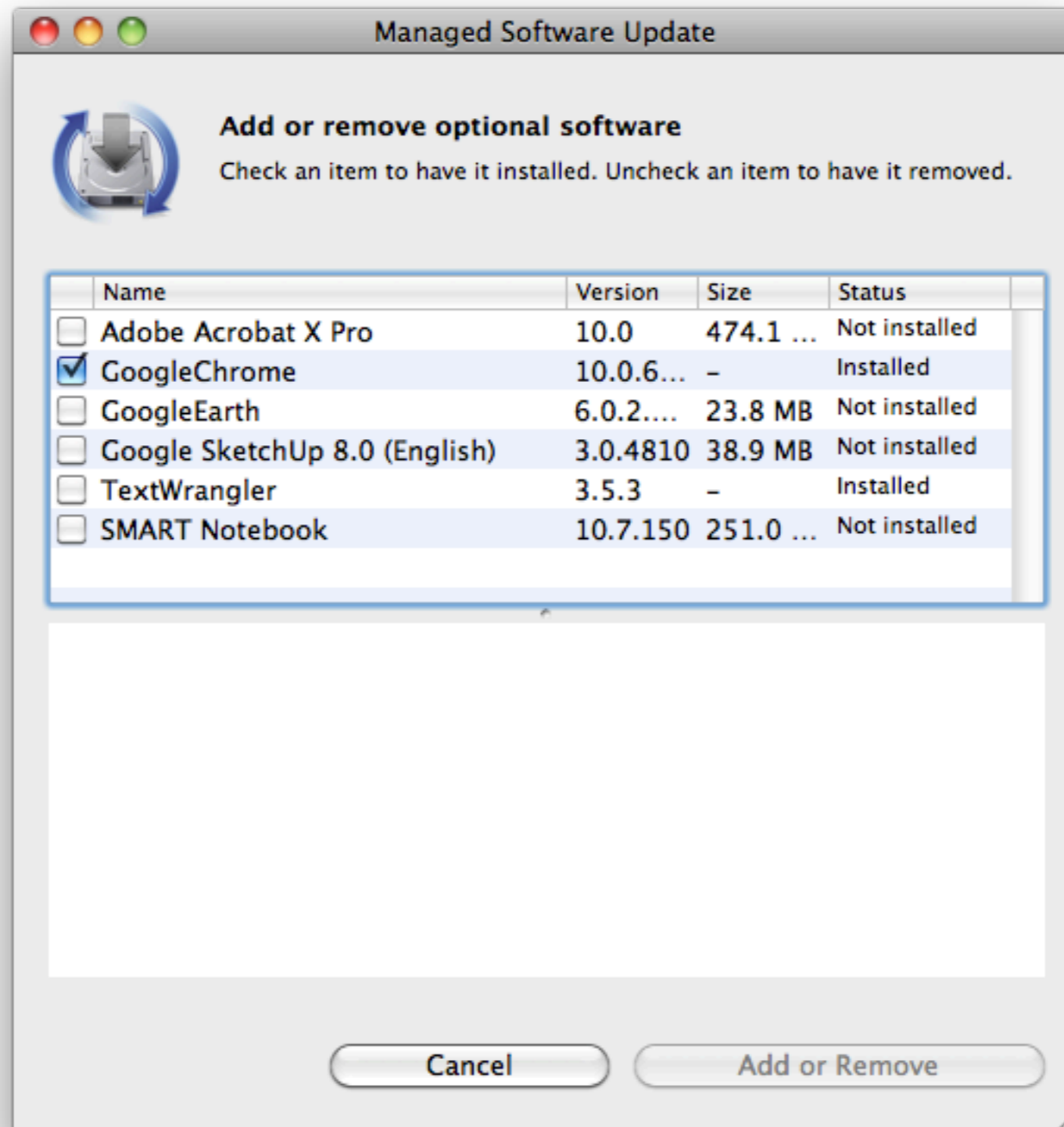
# Optional\_Installs

```
<key>optional_installs</key>  
<array>  
  <string>AcrobatXPro</string>  
  <string>GoogleChrome</string>  
  <string>GoogleEarth</string>  
  <string>GoogleSketchUp</string>  
  <string>TextWrangler</string>  
  <string>SMARTNotebook</string>  
</array>
```

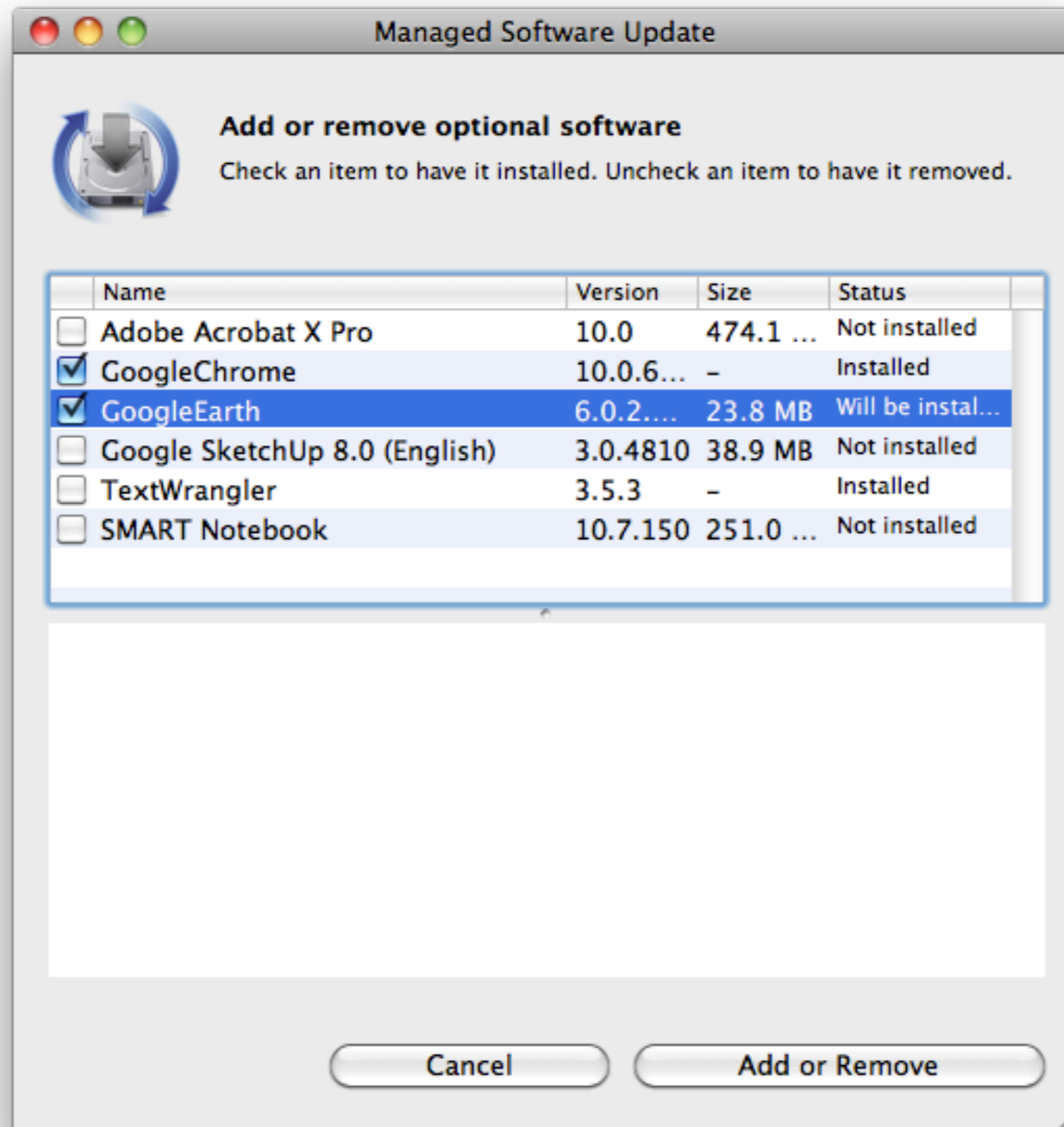
# Optional\_Installs



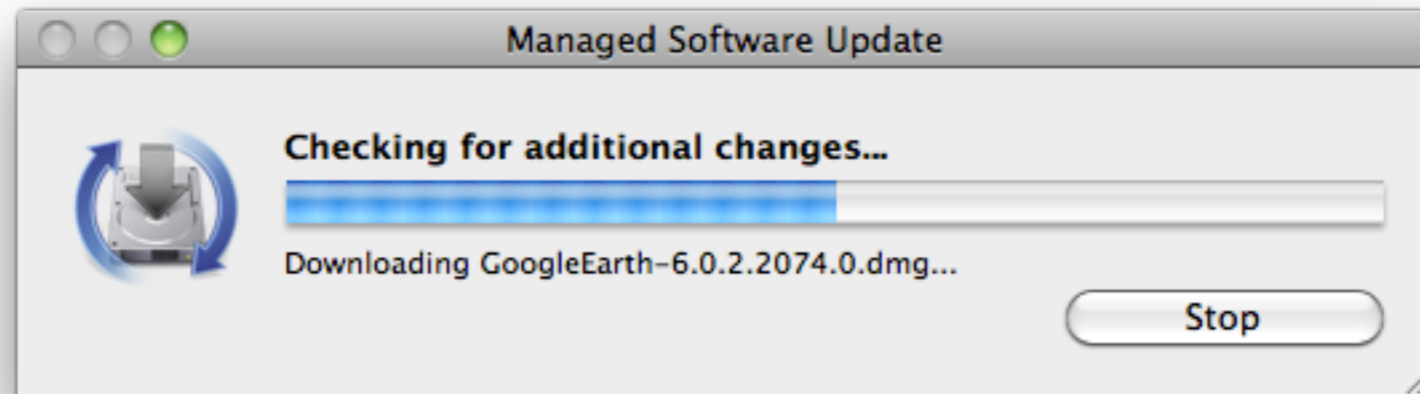
# Optional\_Installs



# Optional\_Installs

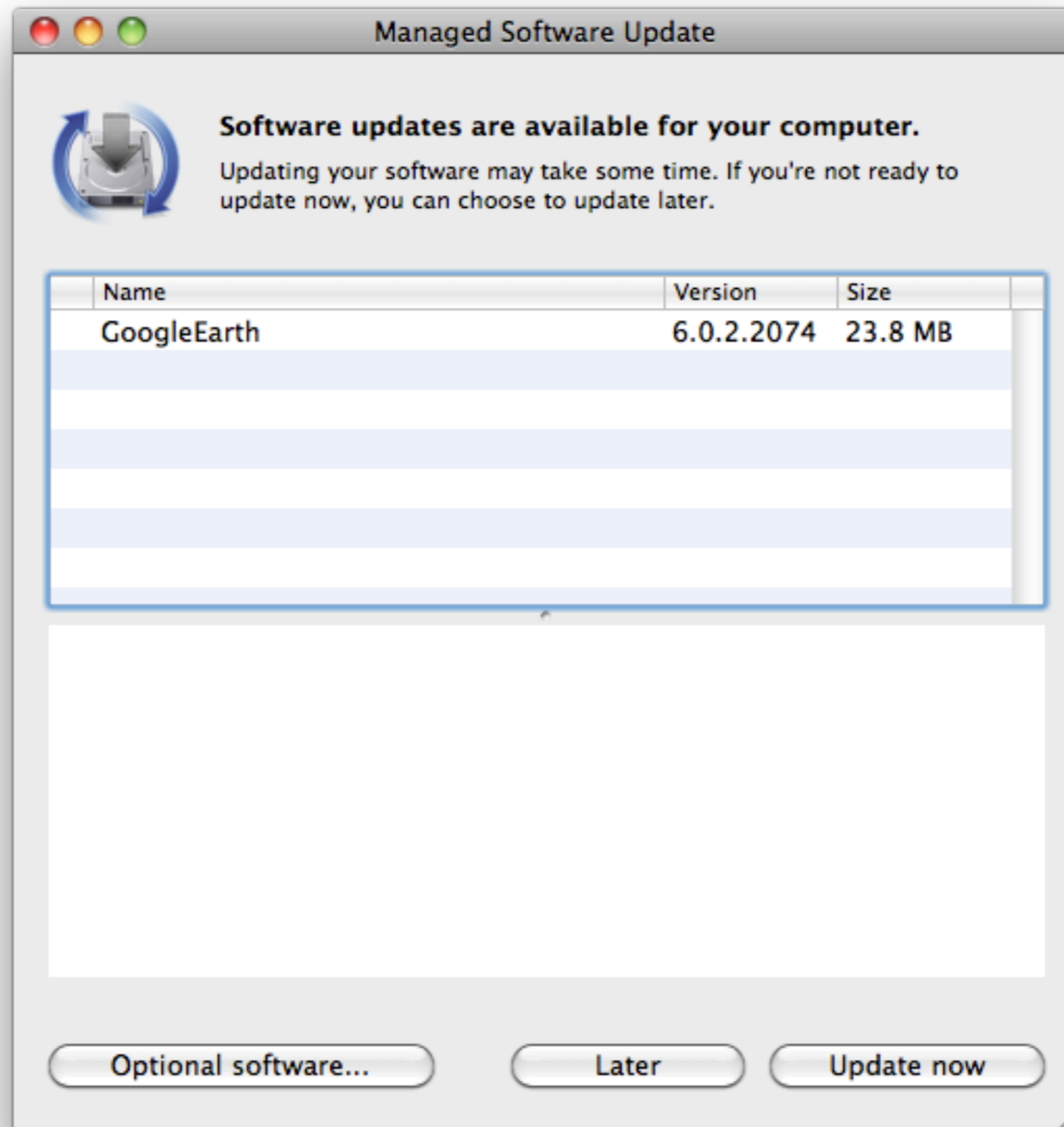


# Optional\_Installs

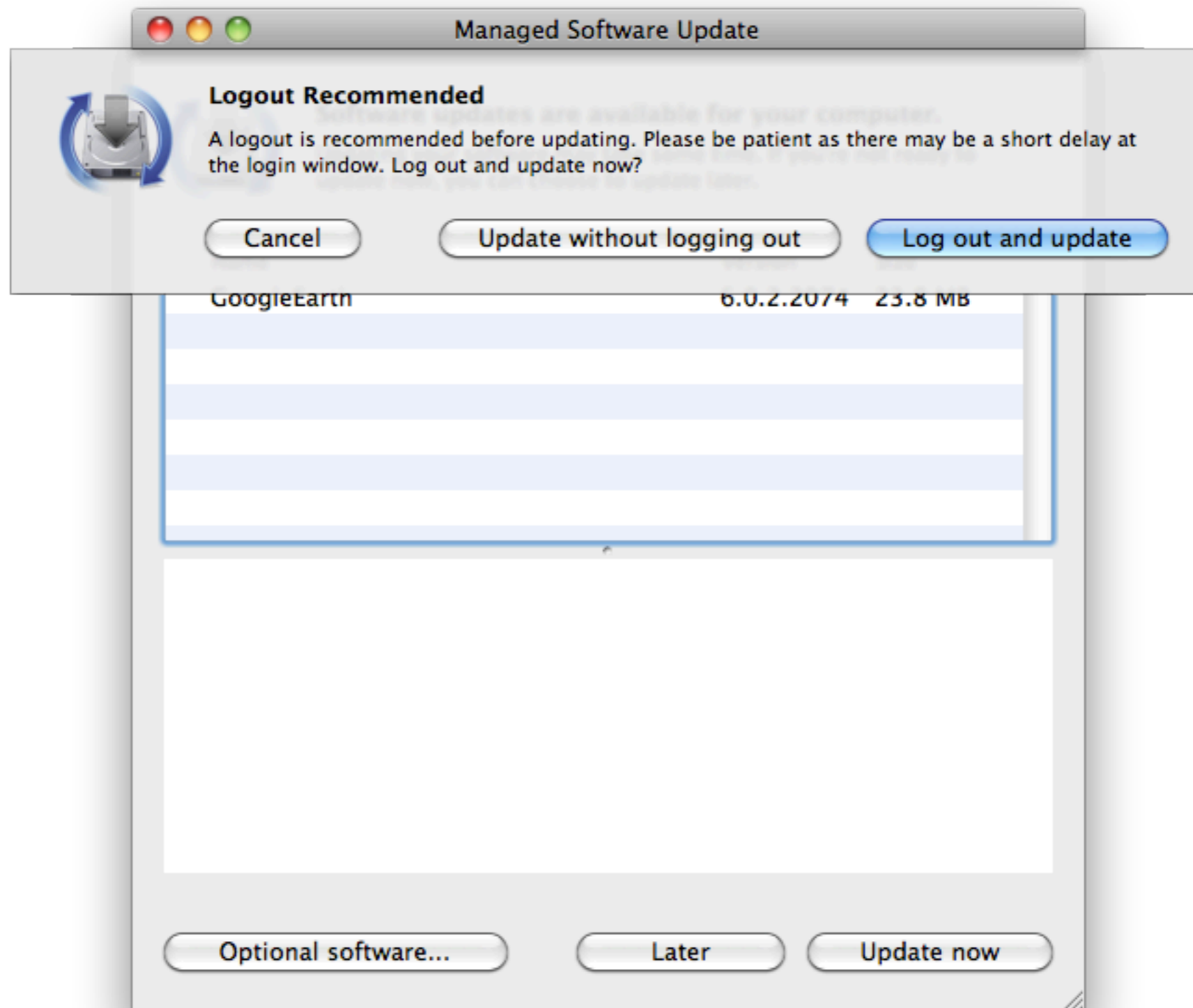




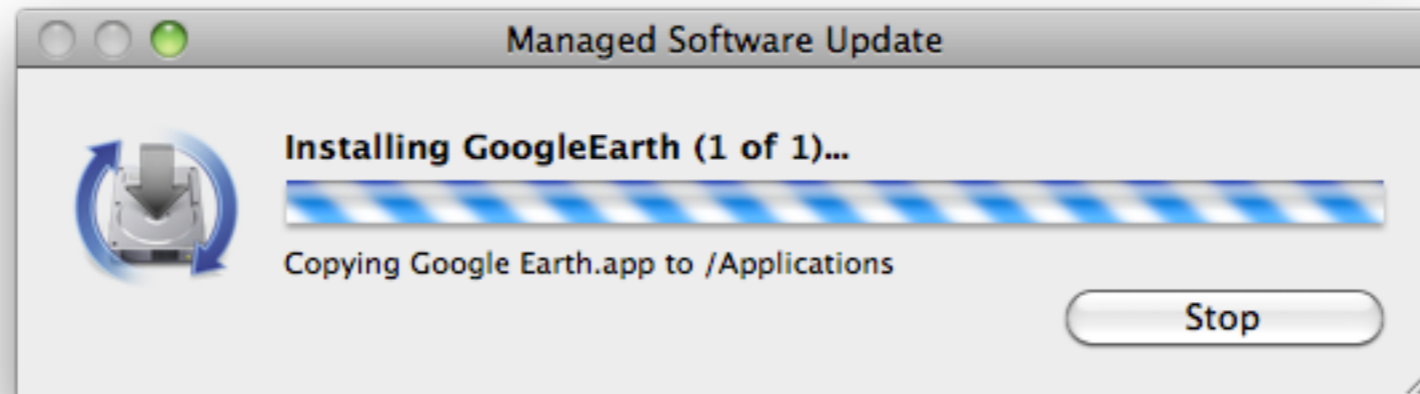
# Optional\_Installs



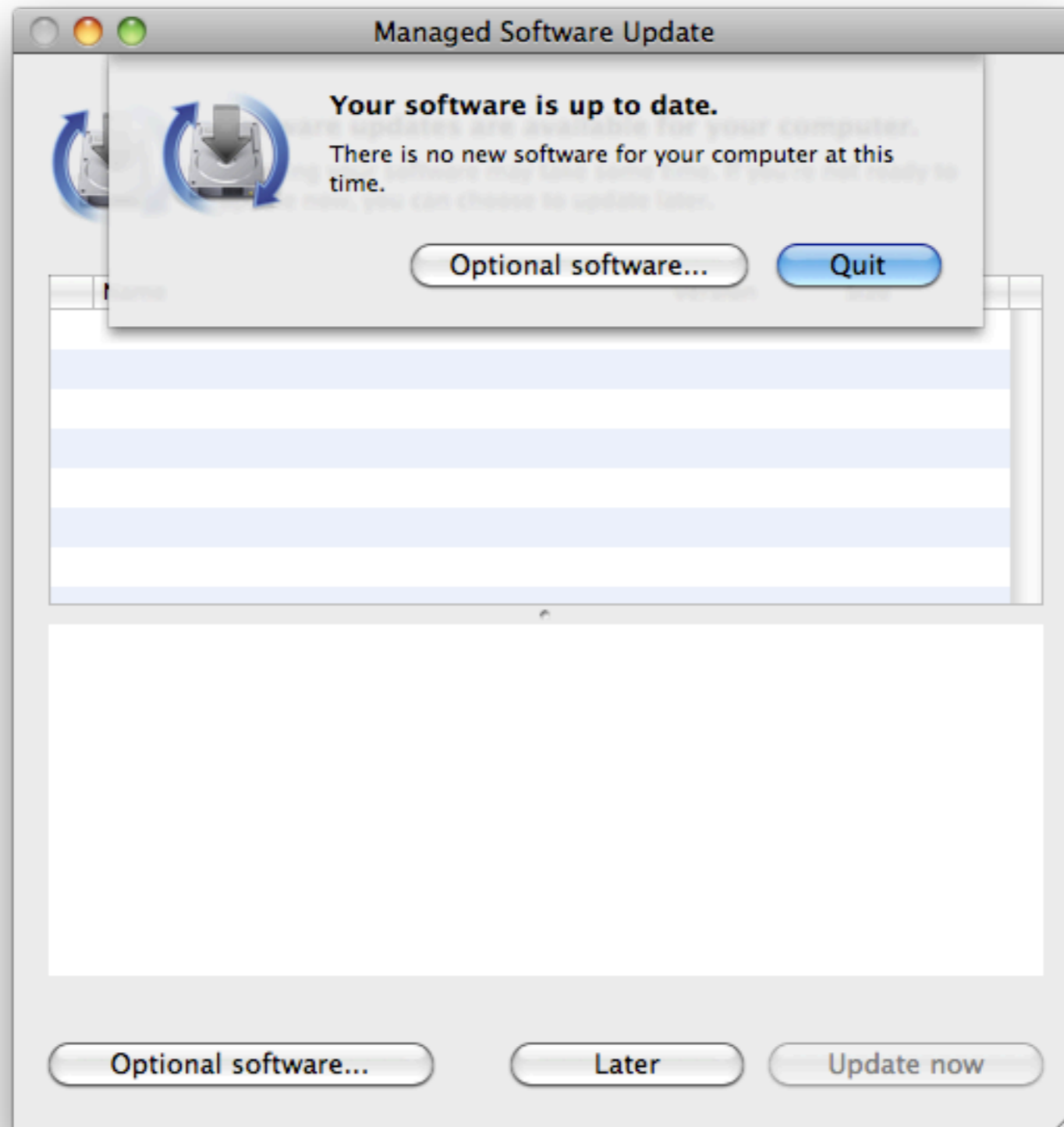
# Optional\_Installs



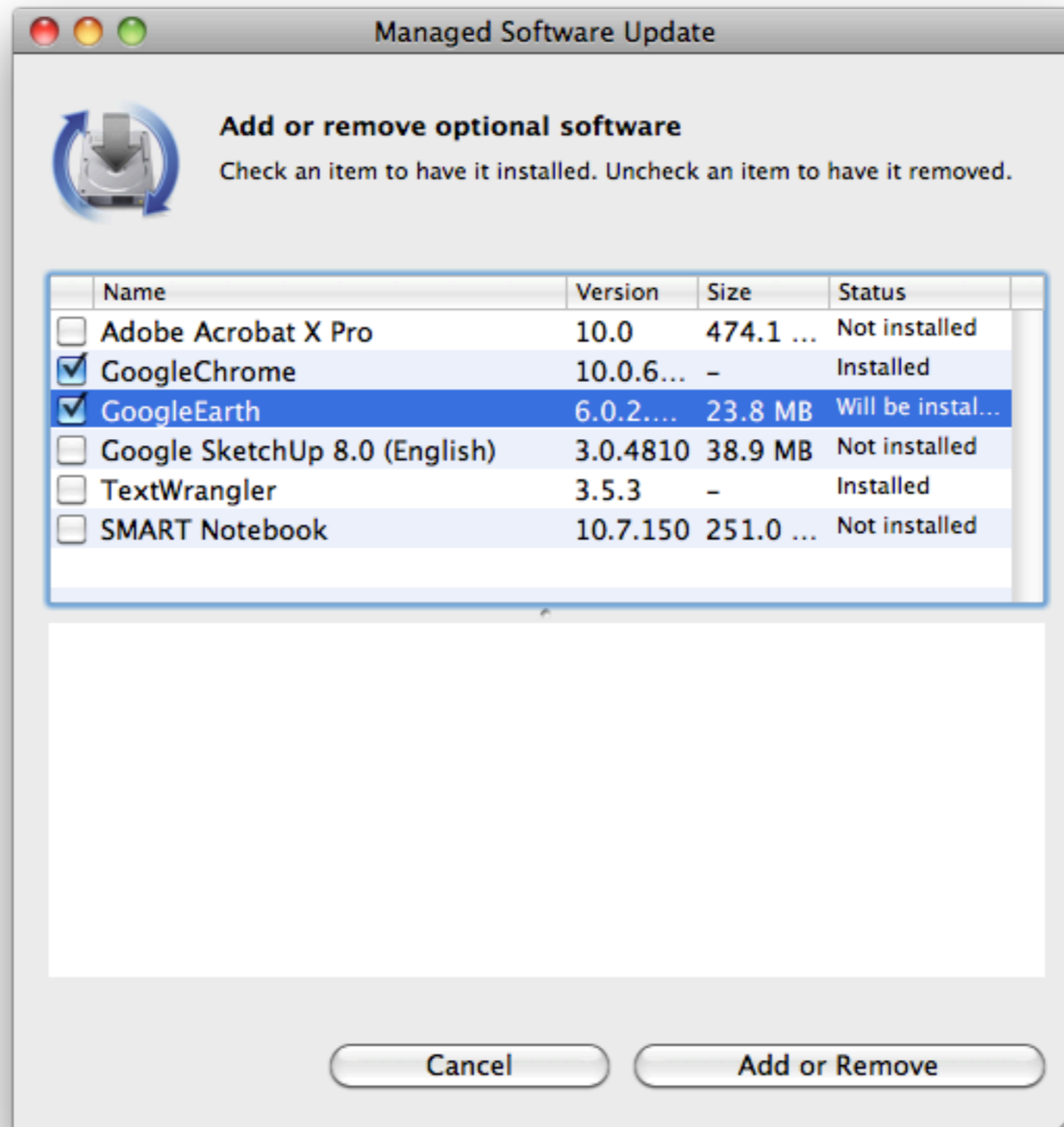
# Optional\_Installs



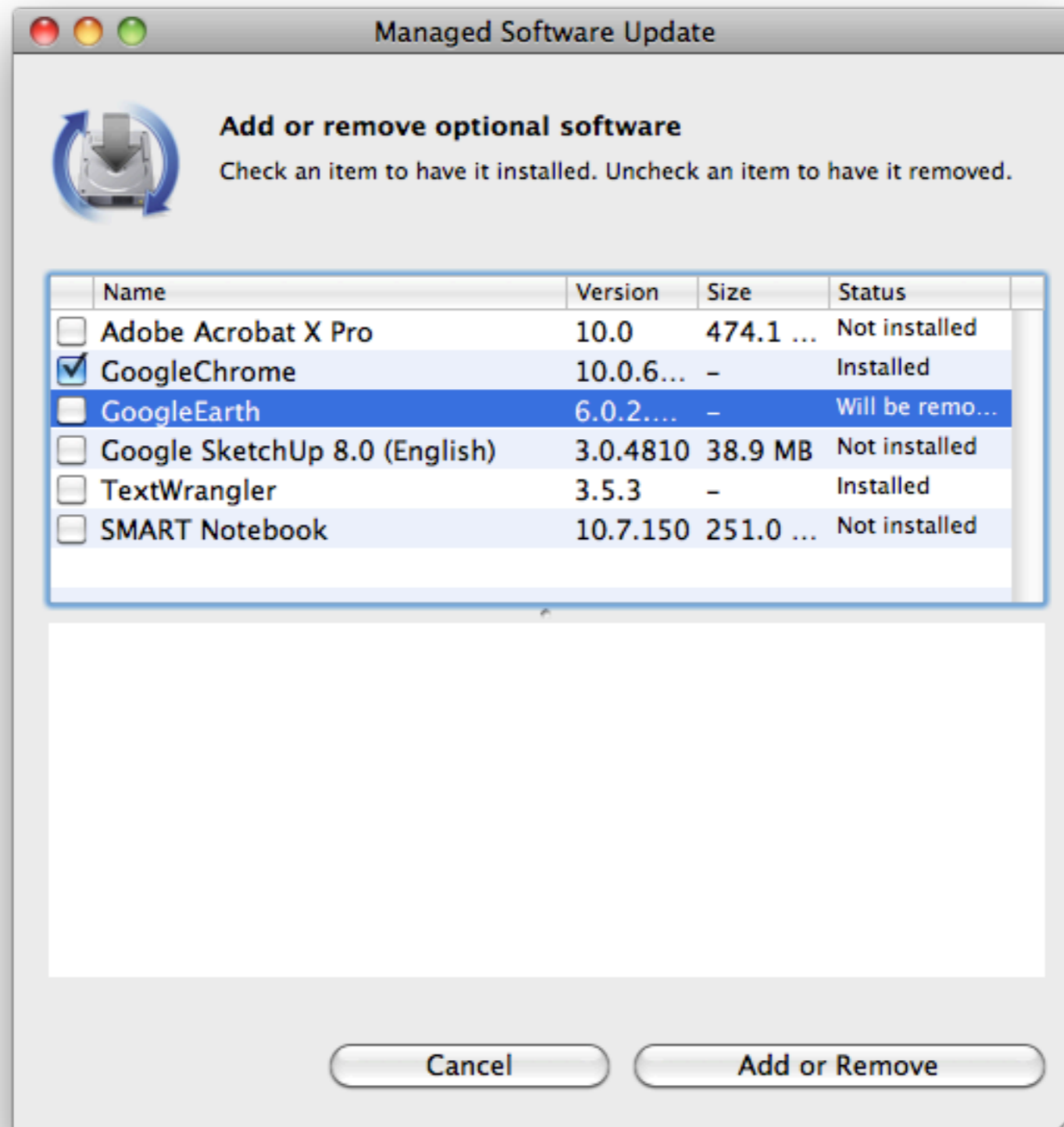
# Optional\_Installs



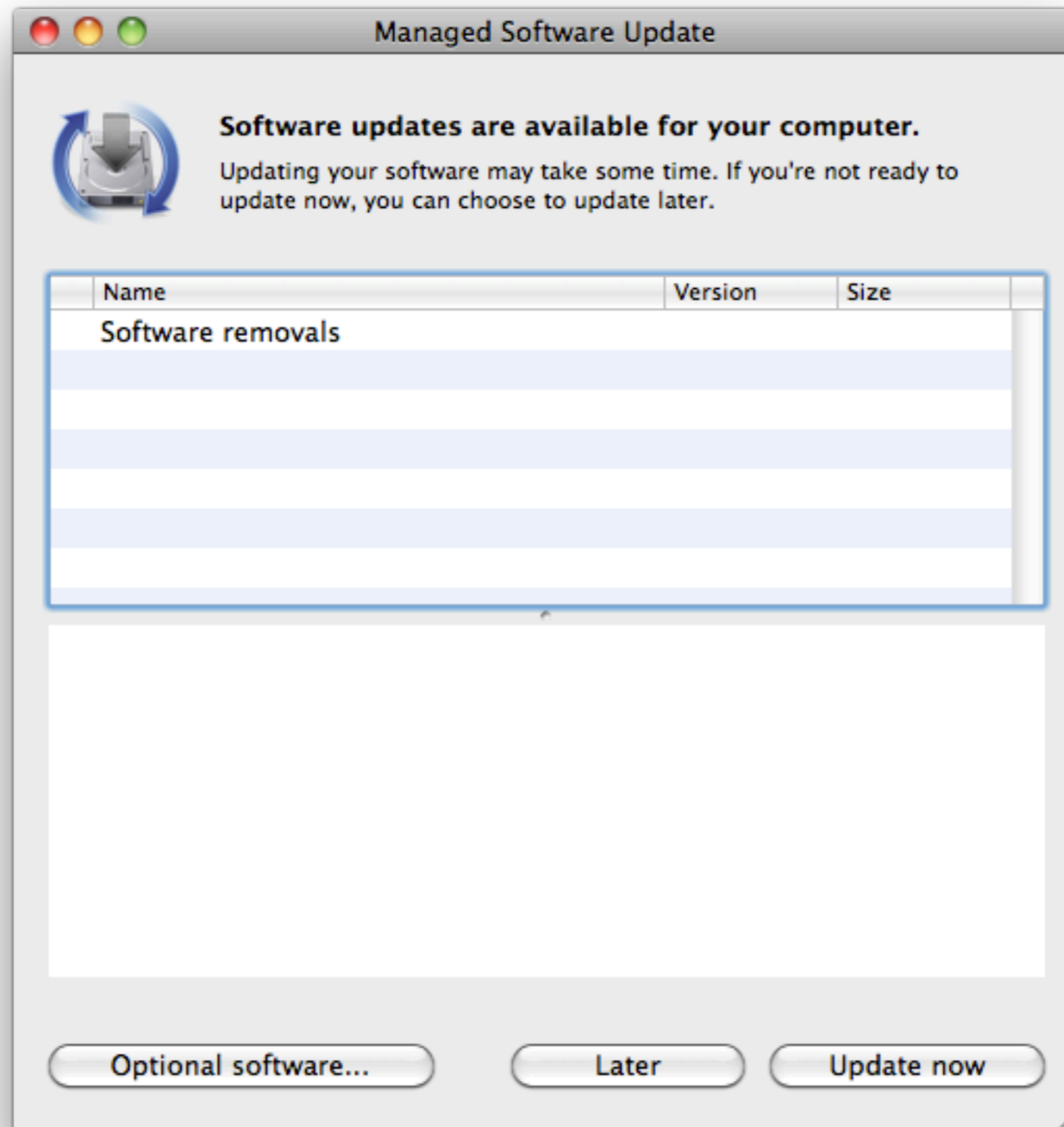
# Optional\_Installs



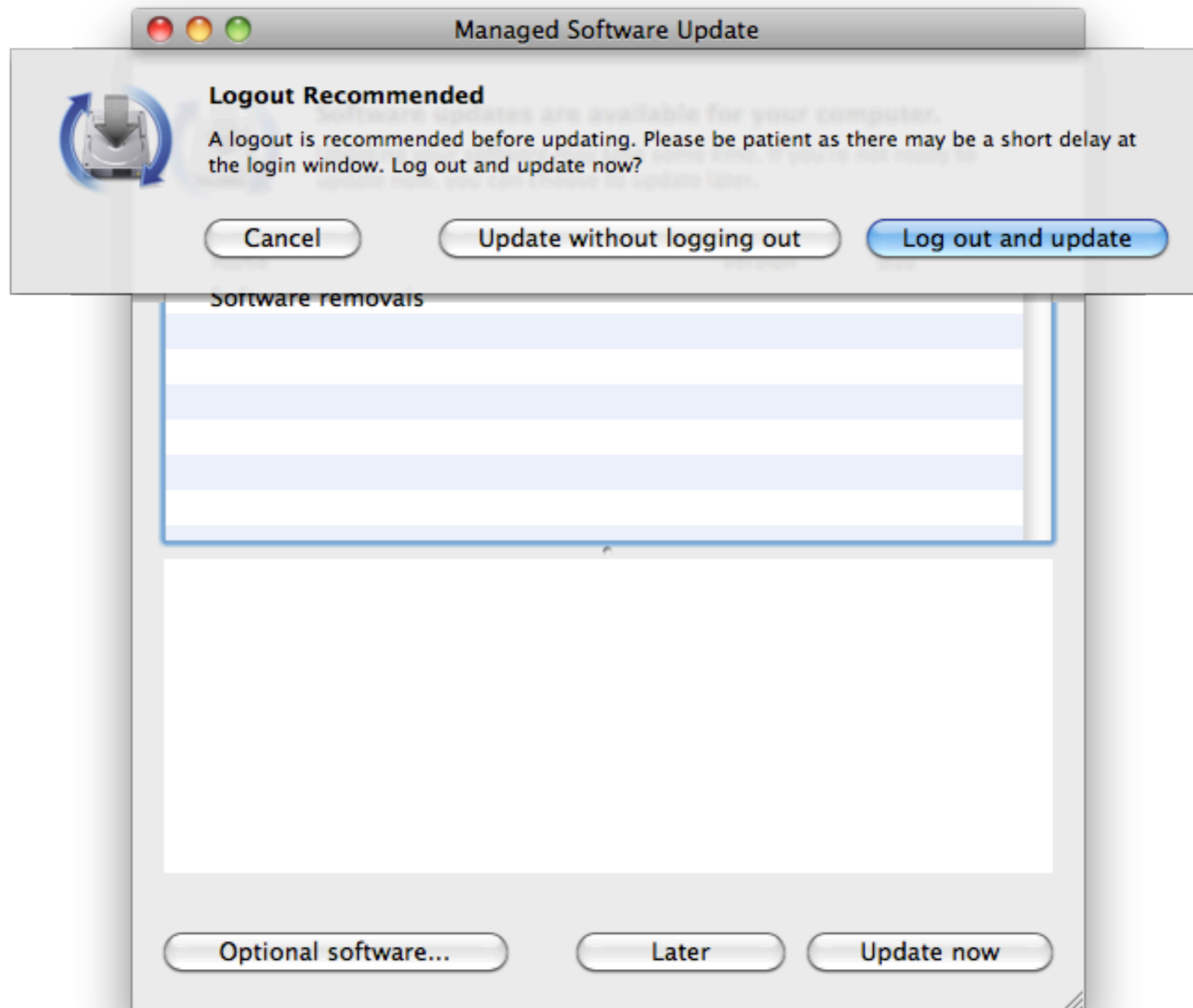
# Optional\_Installs



# Optional\_Installs



# Optional\_Installs



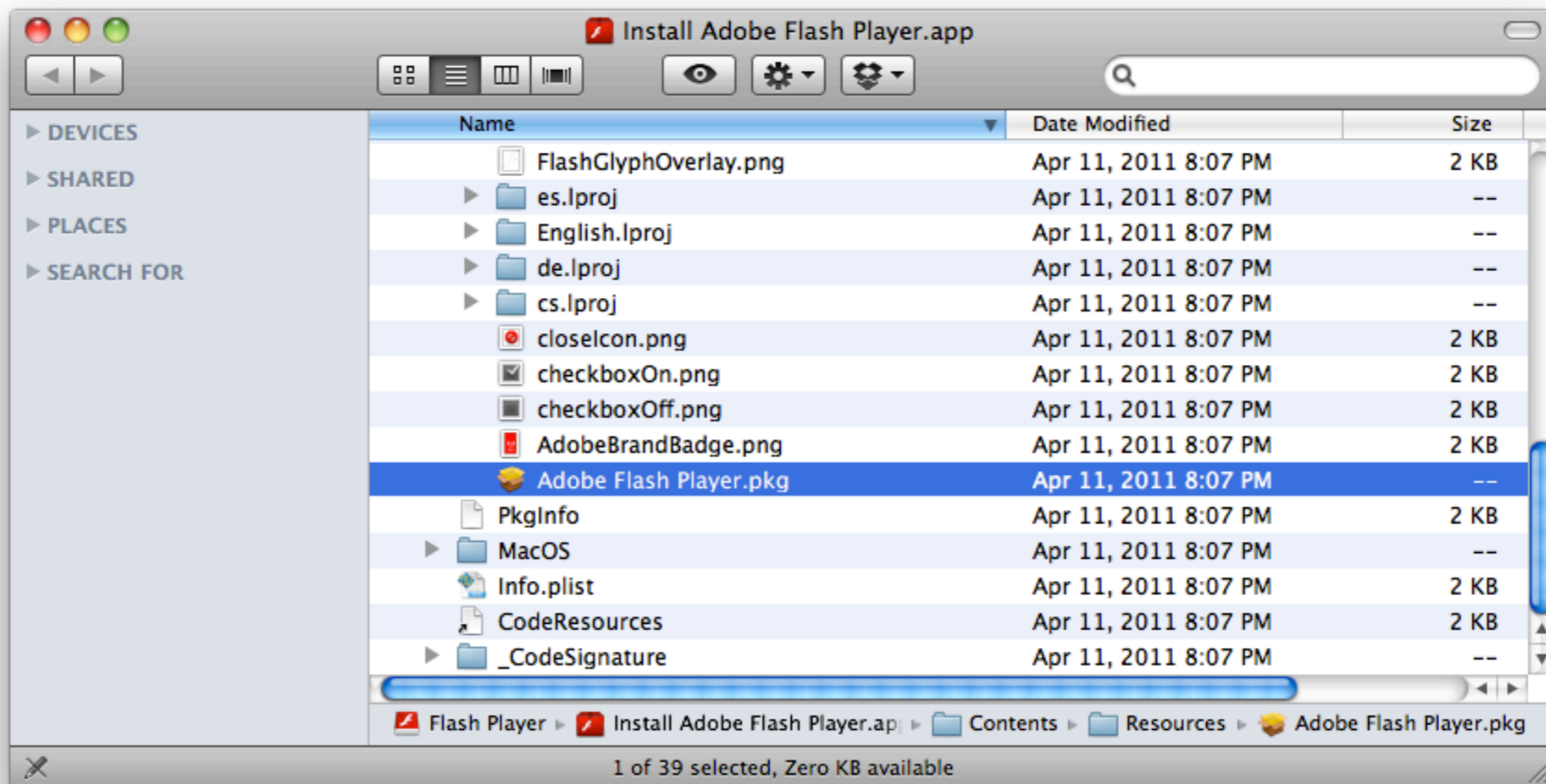


# Common Software Installs

- Adobe Flash Player
- Flip4Mac
- Third Party Web Browsers (Firefox, Chrome, etc)
- Office 2011
- Adobe Acrobat X Pro

# Adobe Flash Player

- Mount DMG, grab installer .pkg file from inside of 'Install Adobe Flash Player.app' and import with munkiiimport.



# Adobe Flash Player

- Make some pkginfo edits

```
<key>blocking_applications</key>
```

```
<array>
```

```
  <string>Firefox</string>
```

```
  <string>Safari</string>
```

```
  <string>Opera</string>
```

```
</array>
```

```
<key>supported_architectures</key>
```

```
<array>
```

```
  <string>i386</string>
```

```
  <string>x86_64</string>
```

```
</array>
```

# Flip4Mac, Firefox, Chrome, etc

- Download .dmg from their website
- Import using Munkiimport
- Test and Deploy
- Workflow for any drag and drop .app or .pkg sitting in a dmg.
- Typically, no pkginfo edits are *necessary*

# Office 2011

- Download from Microsoft (Or make disk image)
- Change .ISO to .DMG
- Import using munkiimport
- Edit the pkginfo file
- Test and Deploy

# Office 2011

- One Plist

```
<key>installs</key>
<array>
  <dict>
    <key>CFBundleShortVersionString</key>
    <string>14.0.0</string>
    <key>path</key>
    <string>/Applications/Microsoft Office 2011/Office/
MicrosoftComponentPlugin.framework/Resources/Info.plist</string>
    <key>type</key>
    <string>plist</string>
  </dict>
</array>
```

- Or each App within the installs key array

```
<dict>
  <key>CFBundleIdentifier</key>
  <string>com.microsoft.Word</string>
  <key>CFBundleName</key>
  <string>Word</string>
  <key>CFBundleShortVersionString</key>
  <string>14.0.0</string>
  <key>path</key>
  <string>/Applications/Microsoft Office 2011/Microsoft Word.app</string>
  <key>type</key>
  <string>application</string>
</dict>
```

# Acrobat X Pro

- Import 'Adobe Acrobat X Pro Installer.pkg' into Munki using munkiimport
- Make sure installs keys are acceptable.
- Create a licensing package to apply a serial number using Adobe's APT
- Test and make sure it installs properly.

# Acrobat X Pro

- Create a Licensing package using Licensing tool. (See [http://kb2.adobe.com/cps/837/cpsid\\_83709.html](http://kb2.adobe.com/cps/837/cpsid_83709.html) )
- This package should deliver the APT to /tmp and run a script to do the actual serializing of Acrobat X Pro.
- Add a `<key>update_for</key>` array that contains `<string>AcrobatXPro</string>` to the licensing package's pkginfo file.
- Test and make sure that Acrobat X installs, then the APT package runs and serializes Acrobat.



# Common Software Updates

- Adobe CS5 Products
- Adobe Acrobat Pro and Reader
- Office 2011

# Adobe CS5 Products

- Figure out dependancies (or refer to <http://code.google.com/p/munki/wiki/AdobeCS5UpdateDependencies>).
- Ensure update\_for keys are set correct
- Munki handles the logic

# Adobe CS5 Products

- Add update\_for key to the pkginfo for the applicable update

```
<key>update_for</key>  
<array>  
  <string>AdobePhotoshopCS5</string>  
</array>
```

- Set up an appropriate blocking\_applications for the update

```
<key>blocking_applications</key>  
<array>  
  <string>Adobe Photoshop CS5</string>  
</array>
```

# Acrobat 9 and X Pro

- Use `<key>requires</key>` and `<key>update_for keys</key>` to install updates in the correct order
- Use an `installs` key to make sure munki checks for the right version
- Munki will install only the updates needed

# Acrobat 9 and X Pro

- Add installs key to point to Acrobat 9 or X .app's info.plist (using /usr/local/munki/makepkginfo -f)

```
<key>installs</key>
<array>
  <dict>
    <key>CFBundleShortVersionString</key>
    <string>10.0.1</string>
    <key>path</key>
    <string>/Applications/Adobe Acrobat X Pro/Adobe Acrobat
Pro.app/Contents/Info.plist</string>
    <key>type</key>
    <string>plist</string>
  </dict>
</array>
```

# Acrobat 9 and X Pro

- Acrobat 10.0.1 Pro Update Keys
- Set `<key>name</key>` to `<string>AcrobatXProUpdate</string>`

```
<key>update_for</key>  
<array>  
  <string>AcrobatXPro</string>  
</array>
```

```
<key>requires</key>  
<array>  
  <string>AcrobatXPro-10.0.0.0.0</string>  
</array>
```

```
<key>blocking_applications</key>  
<array>  
  <string>Firefox</string>  
  <string>Safari</string>  
  <string>Opera</string>  
</array>
```

# Acrobat 9 and X Pro

- Acrobat 10.0.2 Pro Update Keys
- Set `<key>name</key>` to `<string>AcrobatXProUpdate</string>`

```
<key>update_for</key>  
<array>  
  <string>AcrobatXPro</string>  
</array>
```

```
<key>requires</key>  
<array>  
  <string>AcrobatXProUpdate-10.0.1.0.0</string>  
</array>
```

```
<key>blocking_applications</key>  
<array>  
  <string>Firefox</string>  
  <string>Safari</string>  
  <string>Opera</string>  
</array>
```

# Office 2008 and 2011

- Import using munkiimport
- Make sure requires, update\_for and installs keys are set properly
- Munki handles the logic



# Office 2008 and 2011

- Add requires and update\_for keys to the Update's pkginfo file.

```
<key>requires</key>  
<array>  
  <string>Office2011-14.0.0.0.0</string>  
</array>
```

```
<key>update_for</key>  
<array>  
  <string>Office2011</string>  
</array>
```

# Bootstrapping

## Image-less Deployments

- Pros
  - More agile than modular or golden master images
  - Software configs are already in place
  - Always fully updated at deployment time
- Cons
  - More time spent deploying
  - Not feasible for large images (20gb+)

# Bootstrapping

## Image-less Deployments

- Drop settings into place via a netboot environment
  - Install Munki Client and Apply Settings (ManagedInstalls.plist)
  - touch /Users/Shared/  
com.googlecode.munki.checkandinstallatstartup
  - Optional - Bind to OD (MCX-based settings)
- On first boot, munki installs all Managed\_Installs items

# Managed\_Installs

- Anything in managed\_installs gets installed at first boot

```
<key>managed_installs</key>
```

```
<array>
```

```
  <string>AdobePhotoshopCS5</string>
```

```
  <string>AdobeUpdaterOverrides</string>
```

```
  <string>CiscoAnyConnect</string>
```

```
  <string>Fetch</string>
```

```
  <string>Firefox</string>
```

```
  <string>FlashPlayer</string>
```

```
  <string>Flip4Mac WMV</string>
```

```
  <string>KeyAccess</string>
```

```
  <string>munkitools</string>
```

```
  <string>Office2011</string>
```

```
</array>
```

# Optional\_Installs

- Optional installs are made available to keep first boot installs to a minimum

```
<key>optional_installs</key>  
<array>  
  <string>AcrobatXPro</string>  
  <string>AdobeDreamweaverCS5</string>  
  <string>AdobeFlashProCS5</string>  
  <string>AdobeIllustratorCS5</string>  
  <string>AdobeInDesignCS5</string>  
  <string>GoogleChrome</string>  
  <string>GoogleEarth</string>  
  <string>GoogleSketchUp</string>  
  <string>TextWrangler</string>  
  <string>SMARTNotebook</string>  
</array>
```

# Advanced

- Unattended installs and uninstalls - Used to be named Forced Installs/uninstalls
- Preflights/Postflights
- Managing Admin rights
- CopyFromDMG installs (VERY flexible)

# Forced\_install

- To Force install software, add a managed\_installs item to your manifest

```
<key>managed_installs</key>  
<array>  
  <string>Firefox</key>  
</array>
```

- As well as a forced\_install key to the pkginfo file of that item

```
<key>forced_install</key>  
<true/>
```

# Forced\_uninstall

- To Force install software, add a managed\_uninstalls item to your manifest

```
<key>managed_uninstalls</key>  
<array>  
  <string>Firefox</key>  
</array>
```

- As well as a forced\_uninstall key to the pkginfo file of that item

```
<key>forced_uninstall</key>  
<true/>
```



# Preflight and Postflights

- Check for network status before running (wired or wireless, on or off site, etc)
- Dynamic Client settings (See <http://code.google.com/p/munki/wiki/DynamicClientConfigurationsViaPreflightScripting>)
- Send email on failures and warnings
- More info - <http://code.google.com/p/munki/wiki/PreflightAndPostflightScripts>

# Manage admin rights

- Leverages Payload-less package to run a postflight script
- Can add or remove admin rights for specific users.
- More info - <http://code.google.com/p/munki/wiki/ManagingAdminRightsWithMunki>

# Copy from DMG installs

- Can copy any file in place using copy from DMG.
- Allows you to set specific permissions on copied files, copy multiple files from one .dmg file to different locations.
- More Info - <http://code.google.com/p/munki/wiki/CopyFromDMG>

# Resources

- Munki Wiki - <http://code.google.com/p/munki/w/list>
- Munki-Dev google group - <http://groups.google.com/group/munki-dev>
- ##osx-server on [freenode.net](http://freenode.net)
- Community Wiki - [osxdeployment.com](http://osxdeployment.com)

Questions?