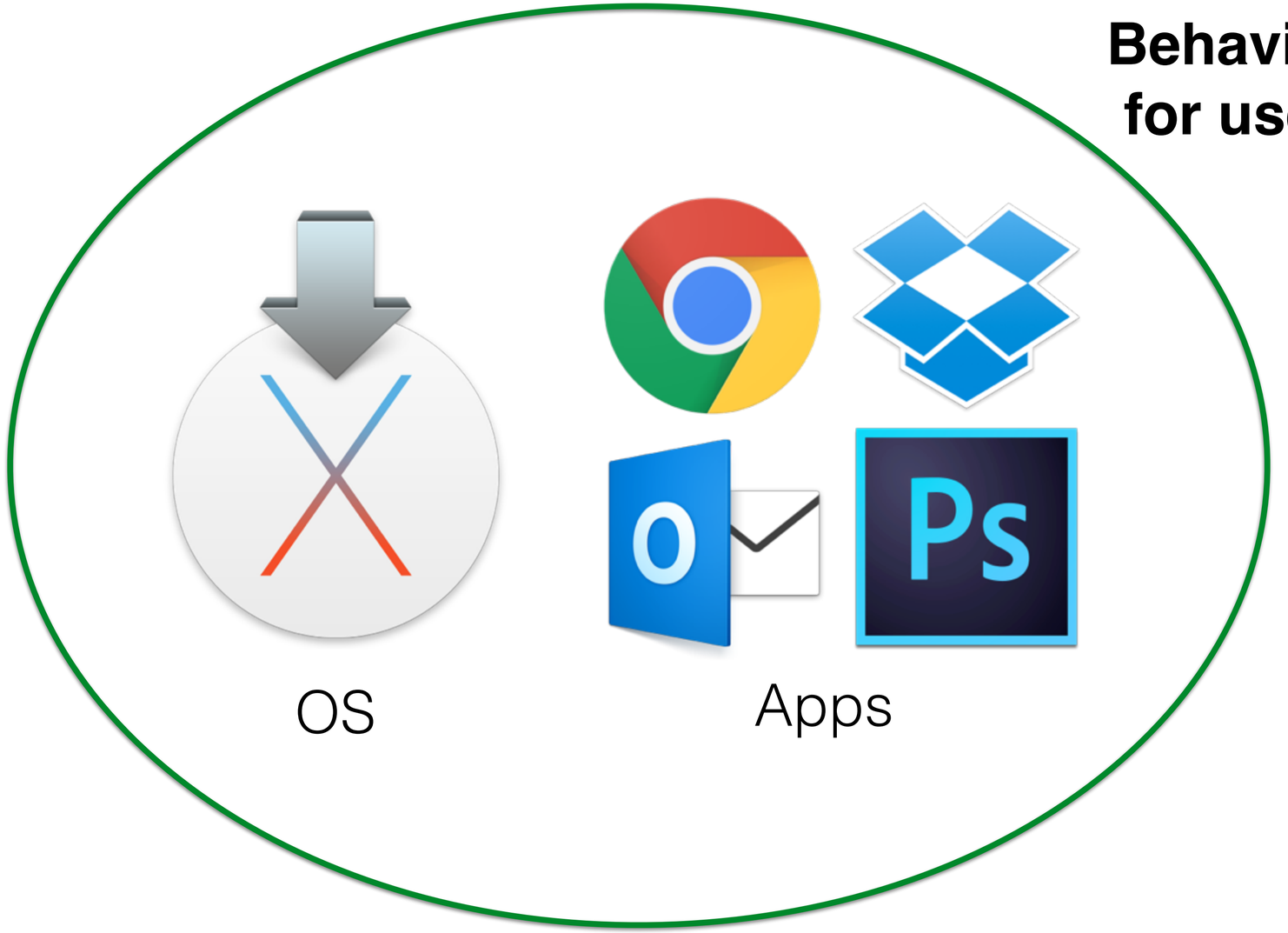
Managing the User Experience

Tim Sutton
Concordia University, Faculty of Fine Arts
Montreal

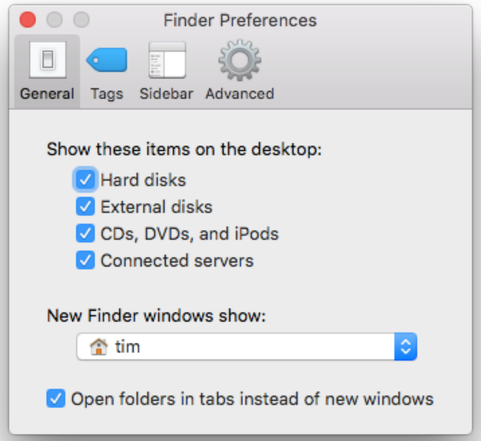
Behaviour for users

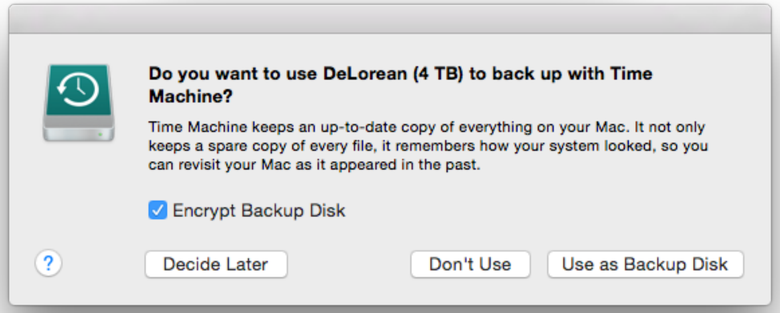


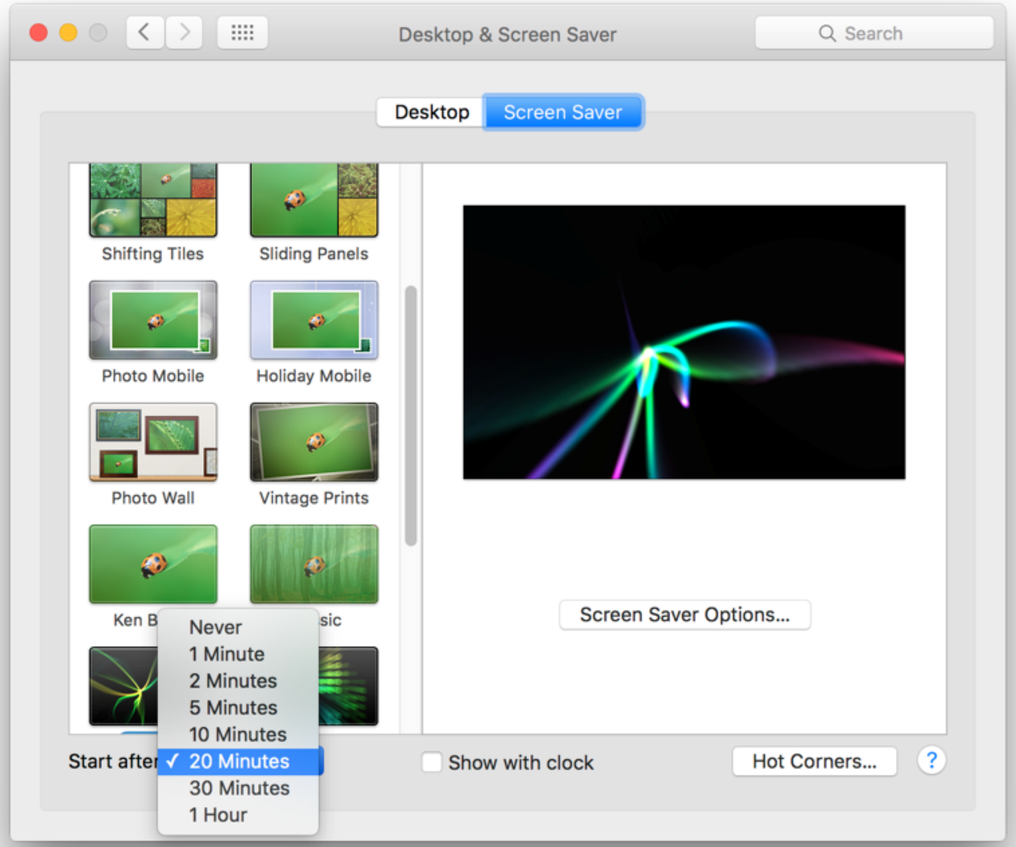
Why?

- Modify specific features causing issues in your environment
  - Auto-updaters
  - Installation issues (e.g. Installer scripts trying to set user Preferences)
- Licensing
- Particulars of your environment (shares, printers, network settings)
- Settings for specific software workflows

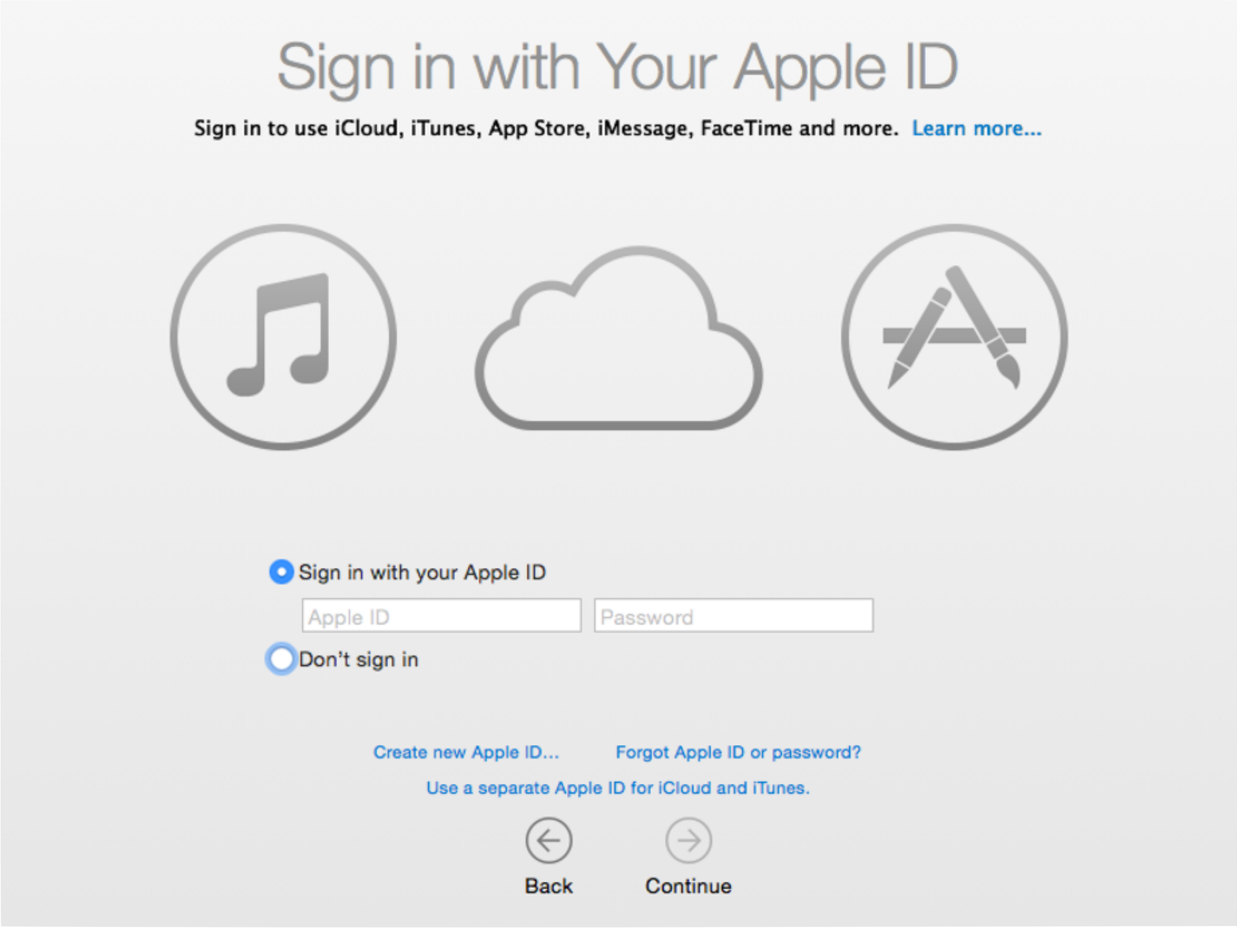
OS X


Finder Preferences
General
Tags
Sidebar
Advanced
Show these items on the desktop:
Hard disks
External disks
CDs, DVDs, and iPods
Connected servers
New Finder windows show:
tim
Open folders in tabs instead of new windows

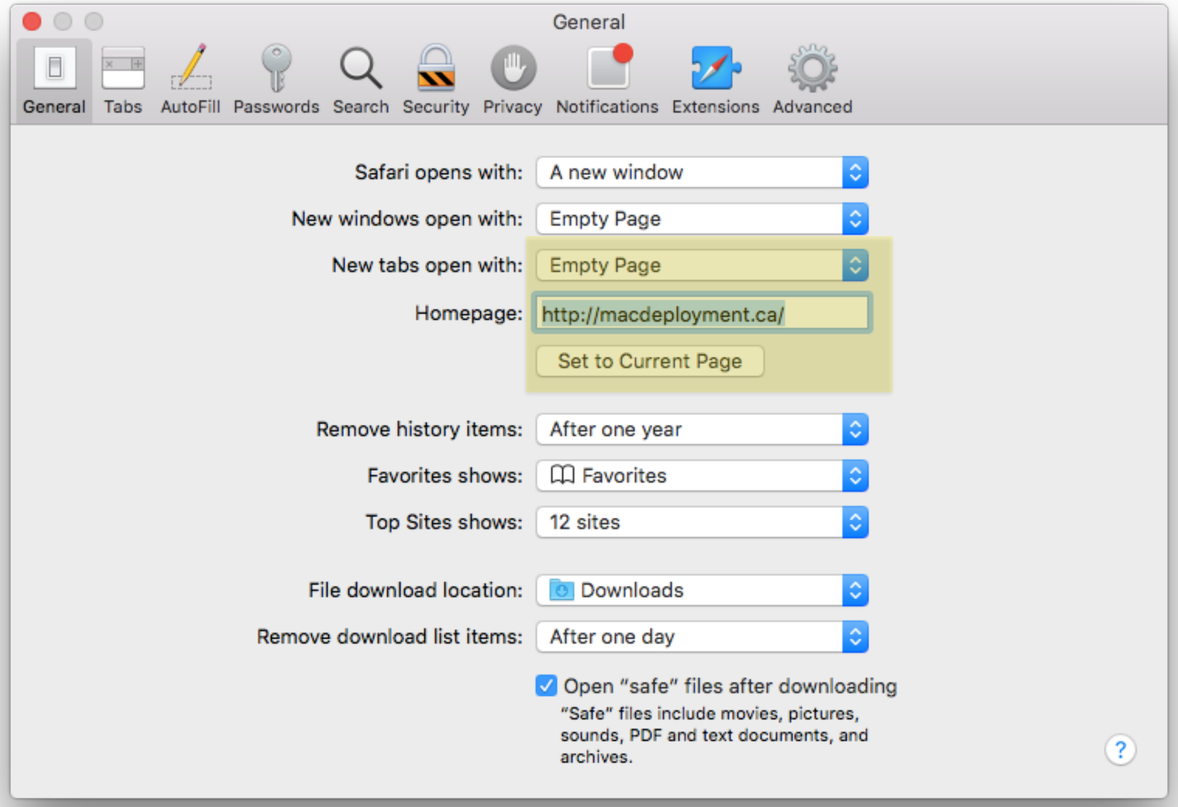

Do you want to use DeLorean (4 TB) to back up with Time Machine?
Time Machine keeps an up-to-date copy of everything on your Mac. It not only keeps a spare copy of every file, it remembers how your system looked, so you can revisit your Mac as it appeared in the past.
Encrypt Backup Disk
Decide Later
Don't Use
Use as Backup Disk


Desktop & Screen Saver
Search
Desktop
Screen Saver
Shifting Tiles
Sliding Panels
Photo Mobile
Holiday Mobile
Photo Wall
Vintage Prints
Never
1 Minute
2 Minutes
5 Minutes
10 Minutes
20 Minutes
30 Minutes
1 Hour
Start after
Screen Saver Options...
Show with clock
Hot Corners...

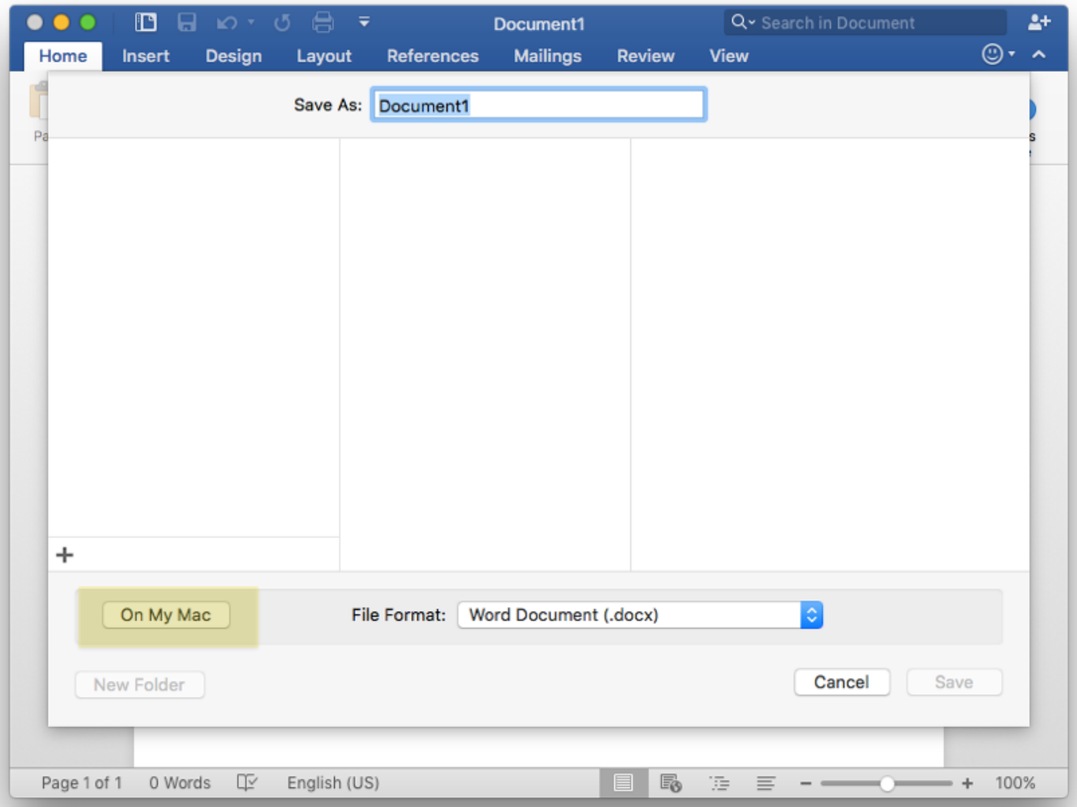
OS X


Sign in with Your Apple ID
Sign in to use iCloud, iTunes, App Store, iMessage, FaceTime and more. Learn more...
Sign in with your Apple ID
Apple ID
Password
Don't sign in
Create new Apple ID...
Forgot Apple ID or password?
Use a separate Apple ID for iCloud and iTunes.
Back
Continue

Apps using OS X Preferences

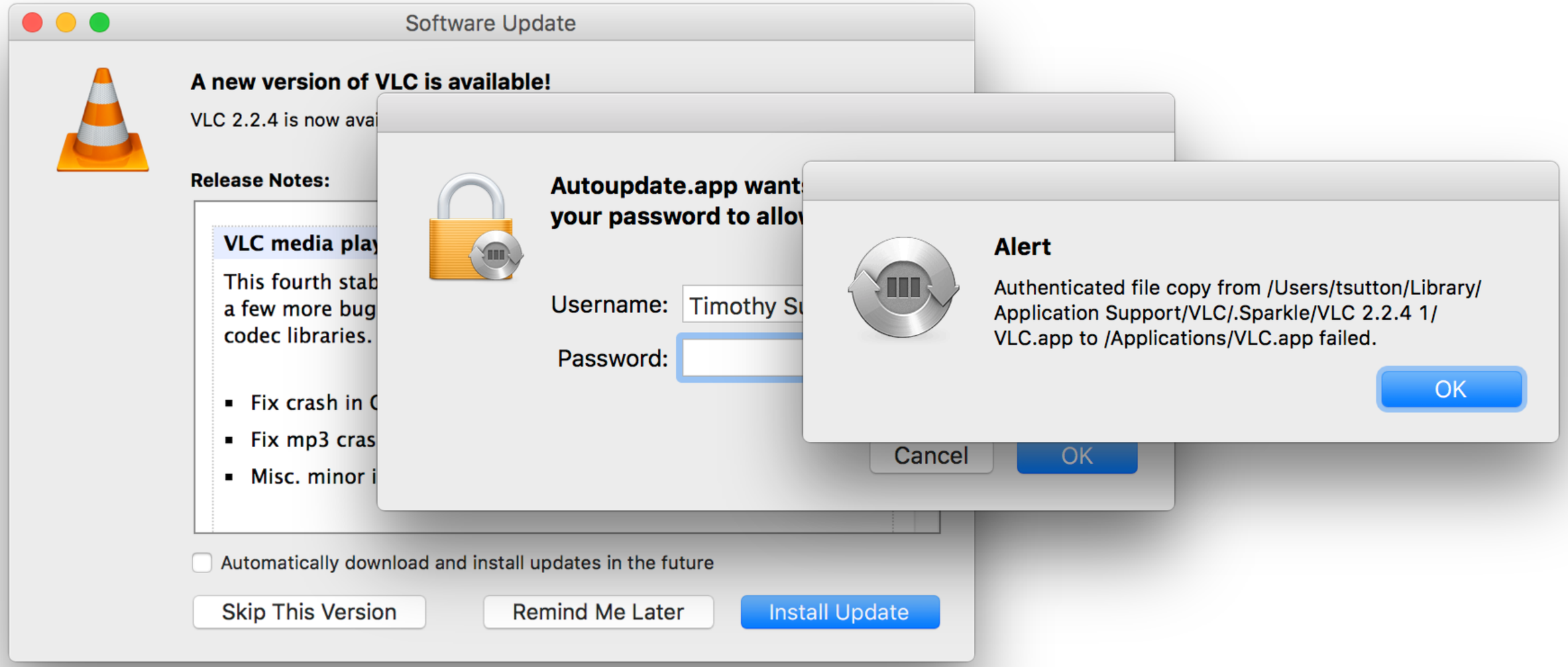


Safari

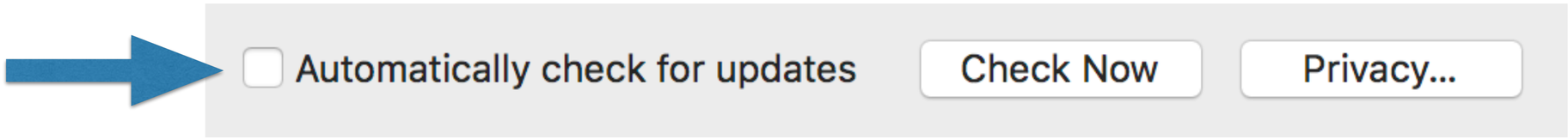


Office 2016 (sort of)

Apps using OS X Preferences



Apps using OS X Preferences



`SUEnableAutomaticChecks`

Apps using proprietary preferences systems

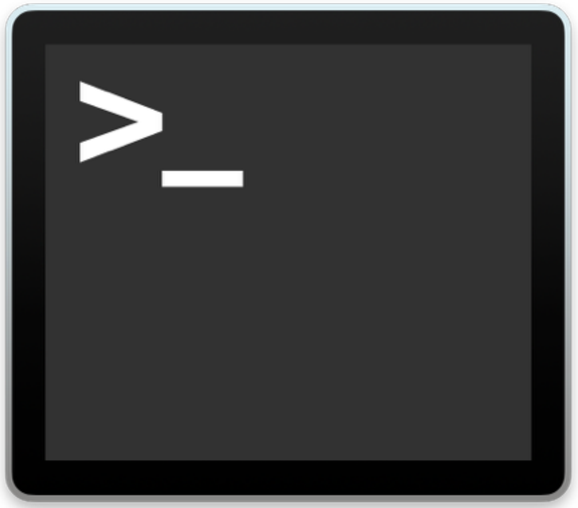
- Chrome, Firefox
- Most Adobe apps
- Many of Office 2016’s user settings
- Many cross-platform applications

Know your environment

- Are users admins?
- Shared (lab / kiosk / classroom / studio) use, or individual user?
- Network users or local users? Persistent home folders, or brand new on every login?

Tools

Managing OS X Preferences



`defaults`

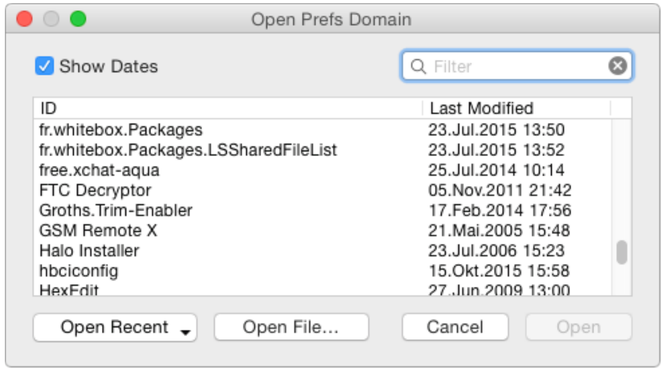
Configuration Profiles

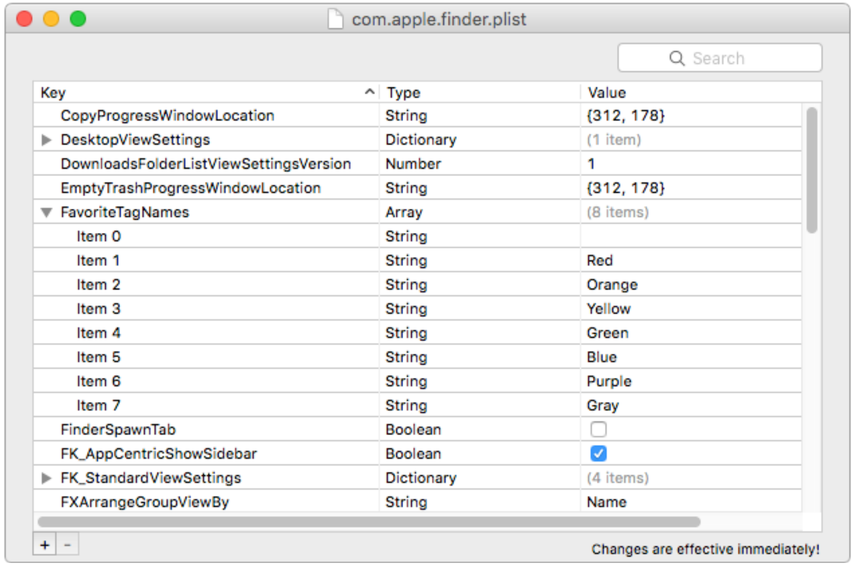
NSUserDefaults
CFPreferences

Managing OS X Preferences

Prefs Editor
(Thomas Tempelmann)

http://apps.tempel.org/



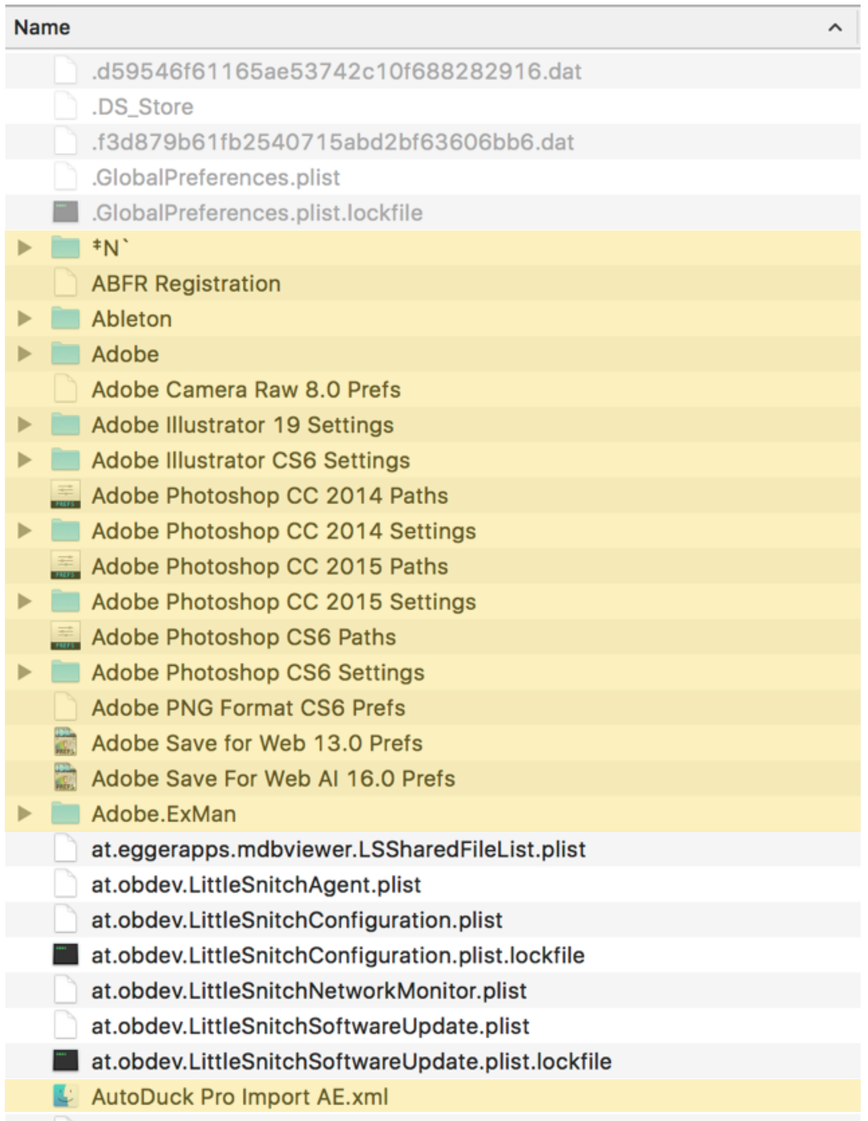


Managing proprietary preferences

- Depends on the application, but won't be using defaults or config profiles!
  - e.g. `/Library/Application Support/Macromedia/mms.cfg`
- Scripts and LaunchAgents running at login time
  - e.g. https://github.com/chilcote/outset
- https://github.com/MagerValp/LoginScriptPlugin (powerful, but install and test carefully)
- LoginHooks (deprecated by Apple, not recommended)

Managing proprietary preferences

- Common locations:
  - ~/Library/Application Support
  - ~/Library/Preferences
  - ~/Library/Preferences/App
  - ~/.app, ~/Documents/App
- Experimentation and testing necessary!



Managing proprietary preferences

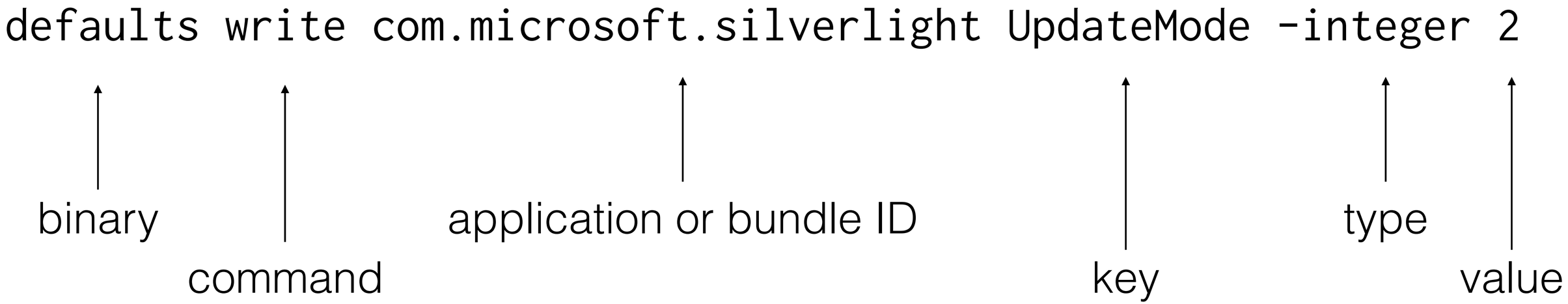
```
{
  "browser": {
    "check_default_browser": false,
    "show_update_promotion_info_bar": false
  },
  "distribution": {
    "make_chrome_default": false,
    "show_welcome_page": false,
    "skip_first_run_ui": true
  },
  "first_run_tabs":["http://macdeployment.ca"],
  "homepage": "http://macdeployment.ca/",
  "sync_promo": {
    "user_skipped": true
  }
}
```

Managing proprietary preferences

```
#!/bin/sh

prefs_src="/Library/MyOrg/Files/google_chrome_preferences"
prefs_tgt_dir="$HOME/Library/Application Support/Google/Chrome/Default"
prefs_tgt_file="$prefs_tgt_dir/Preferences"

# make our user's chrome profile dir if one doesn't already exist
[ -d "$prefs_tgt_dir" ] || mkdir -p "$prefs_tgt_dir"

# if prefs file doesn't already exist, copy it
[ -e "$prefs_tgt_file" ] || cp "$PREFS_SRC" "$prefs_tgt_file"

# create the special 'first run' file
touch "$prefs_tgt_dir/../First Run"
```

defaults

defaults


defaults write com.microsoft.silverlight UpdateMode -integer 2
binary
command
application or bundle ID
key
type
value

defaults

```
defaults write com.microsoft.silverlight UpdateMode -integer 2
```

current-user domain

defaults

```
defaults write /Users/tim/Library/Preferences/com.microsoft.silver
```

current-user domain

defaults

```
defaults write /Library/Preferences/com.microsoft.silverlight Upda
```

any-user domain

defaults

```
defaults -currentHost write com.apple.screensaver idleTime -integer 0
```

current-host domain

defaults

```
/Users/tsutton/Library/Preferences/ByHost/com.apple.Safari.23C1B133-A1D4-5CA6-A526-C1F88B1C47CC.plist
/Users/tsutton/Library/Preferences/ByHost/com.apple.Safari.9B5C8635-D289-5EA3-B598-7277C0175D56.plist
/Users/tsutton/Library/Preferences/ByHost/com.apple.Safari.C1BB22BC-64CE-55E6-B871-F110399060EA.plist
/Users/tsutton/Library/Preferences/ByHost/com.apple.Safari.EEE79E13-BDD2-51A6-B602-DB056032BF88.plist
```

```
➜ defaults -currentHost read com.apple.Safari

{
    SyncedTabsDeviceUUID = "14691C26-0723-4E67-B206-18D14AE98075";
}
```

User and host domains

Search Order	User scope	Host scope	Mapping to `defaults`
1	Current user	Current host	`defaults -currentHost read <id>`
2	Current user	Any host	`defaults read <id>`
3	Any user	Current host	`defaults read /Library/Preferences/<id>`

https://developer.apple.com/library/mac/documentation/MacOSX/Conceptual/BPRuntimeConfig/Articles/UserPreferences.html#//apple_ref/doc/uid/20002092-120915-TPXREF104

Defaults is weird

- `defaults write /tmp/my_new.plist SomeKey “the value”`

From the defaults manpage: ”WARNING: The defaults command will be changed in an upcoming major release to only operate on preferences domains. General plist manipulation utilities will be folded into a different command-line program."

Property Lists

com.apple.SoftwareUpdate.plist

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
   <key>AutomaticCheckEnabled</key>
   <false/>
   <key>CatalogURL</key>
   <string>http://my.reposado.server/index_seed_testing.sucatalog</string>
   <key>LastFullSuccessfulDate</key>
   <date>2016-06-09T15:56:20Z</date>
   <key>LastRecommendedUpdatesAvailable</key>
   <integer>0</integer>
   <key>LastResultCode</key>
   <integer>2</integer>
   <key>PrimaryLanguages</key>
   <array>
      <string>en</string>
      <string>en-CA</string>
   </array>
   <key>RecommendedUpdates</key>
   <array/>
</dict>
</plist>
```

Interacting with plists

- /usr/libexec/PlistBuddy
- TextWrangler (free) handles both XML and binary, displays always as XML
- Text editor of choice (Sublime, Atom, TextMate)
- QuickLook (using the spacebar)
- Python:
  - plistlib (standard library, can't open binary plists as shipped in OS X)
  - FoundationPlist (by Greg Neagle, supports XML and Binary) https://github.com/munki/munki/blob/master/code/client/munkilib/FoundationPlist.py

Property Lists

- Binary formats:
  - `plutil -convert <xml1 | binary1>`
- `man plutil`
  - (linting, merging, and more!)

Managing OS X Preferences

~/Library/Preferences
~/Library/Preferences/ByHost
/Library/Preferences
/Library/Managed Preferences

Managing OS X Preferences

Prefs APIs

cfprefsd

defaults server

Cache to disk

Applications

Plists on disk

Managing OS X Preferences

/System/Library/User Template/<lang>.lproj

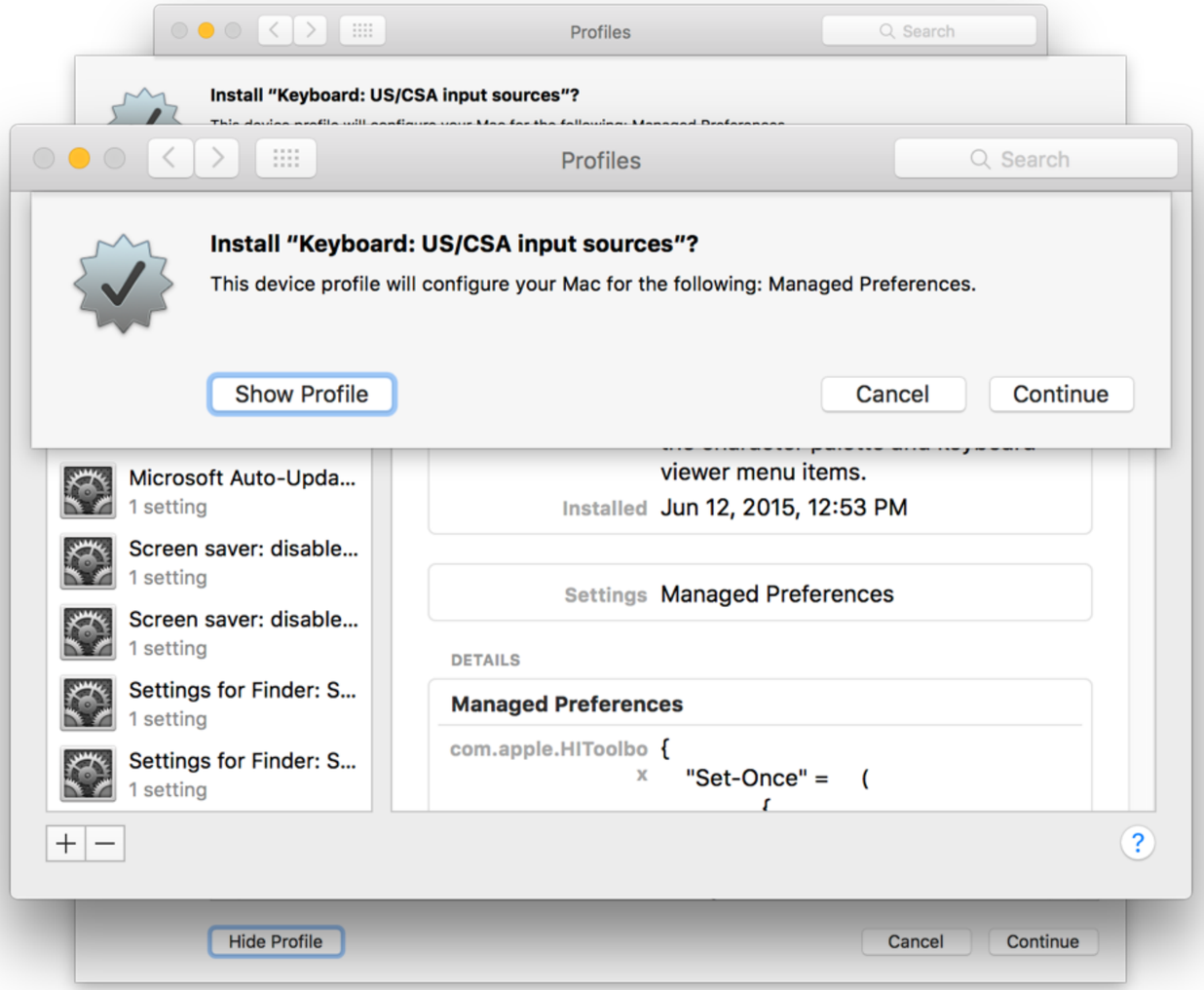
/System/Library/User Template/Non_localized

Configuration Profiles

- Originated on iOS, brought to OS X in Lion
- Defines multiple settings in one “blob”, which can be added or removed, installable using:
  - Mobile Device Management (MDM)
  - Downloaded to OS X system as a .mobileconfig file, installed by double-clicking or Terminal using the `profiles` command
    - .. and/or wrapped in an installer package https://github.com/timsutton/make-profile-pkg

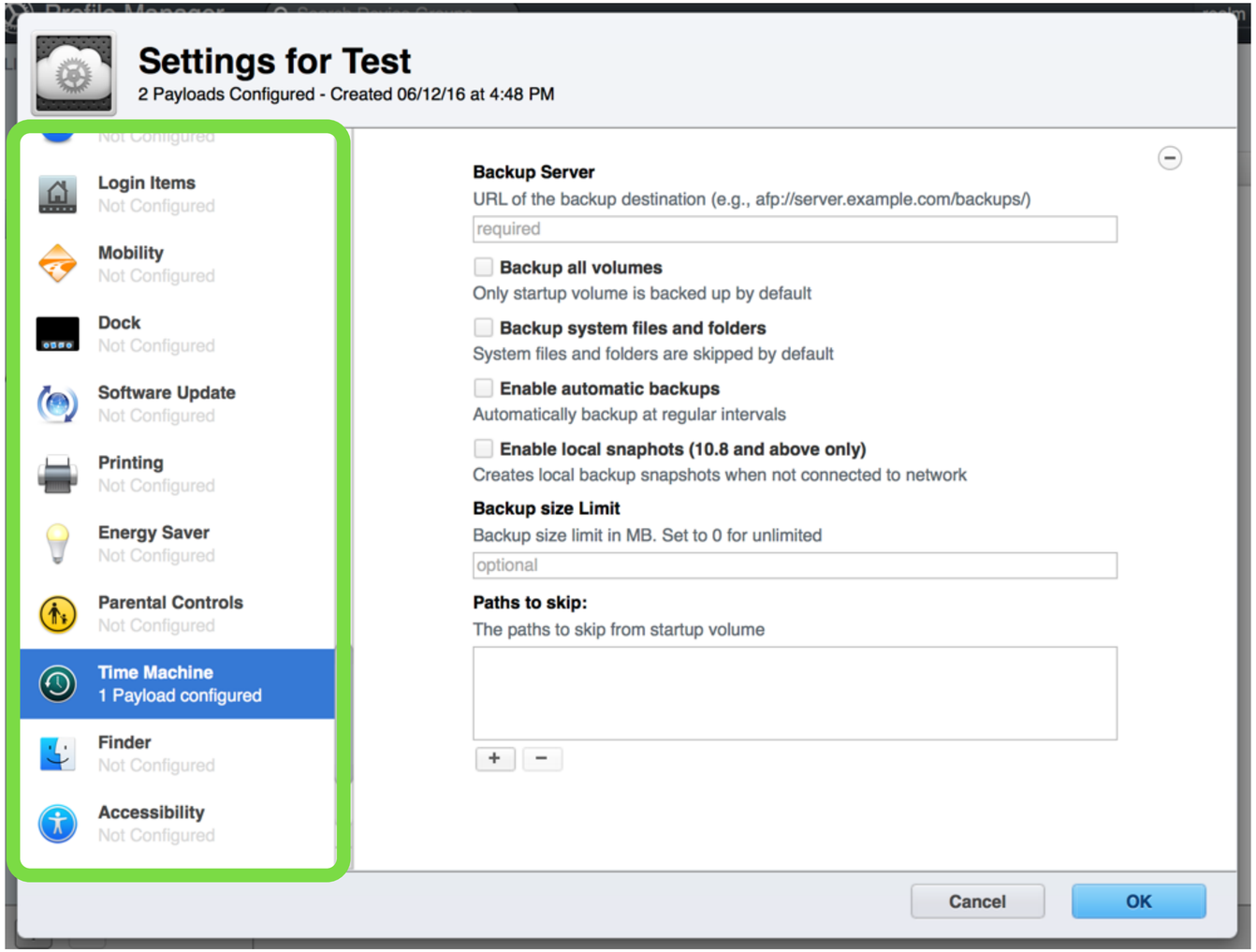
Configuration Profiles

KeyboardInputSources_USAndCSA.mobileconfig

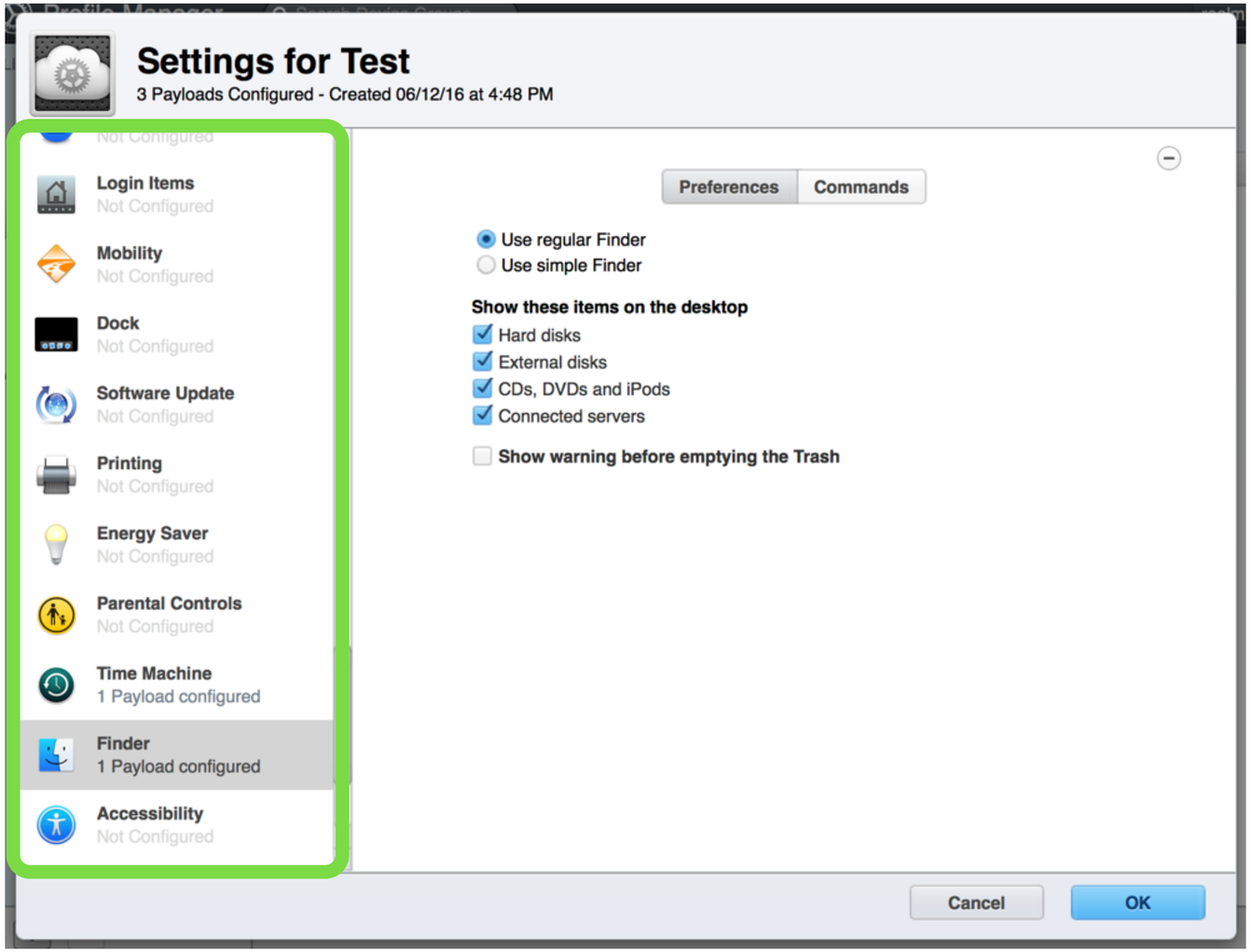


```
<key>PayloadContent</key>
<dict>
  <key>com.apple.Safari</key>
  <dict>
    <key>Set-Once</key>
    <array>
      <dict>
        <key>mcx_data_timestamp</key>
        <date>2014-12-04T19:03:28Z</date>
        <key>mcx_preference_settings</key>
        <dict>
          <key>DomainsToNeverSetUp</key>
          <array>
            <string>google.com</string>
            <string>yahoo.com</string>
          </array>
          <key>HomePage</key>
          <string>http://myorg.org</string>
          <key>LastDisplayedWelcomePageVersionString</key>
          <string>4.0</string>
          <key>LastSafariVersionWithWelcomePage</key>
          <string>8.0</string>
          <key>NewWindowBehavior</key>
          <integer>0</integer>
          <key>SendDoNotTrackHTTPHeader</key>
          <true/>
        </dict>
      </dict>
```

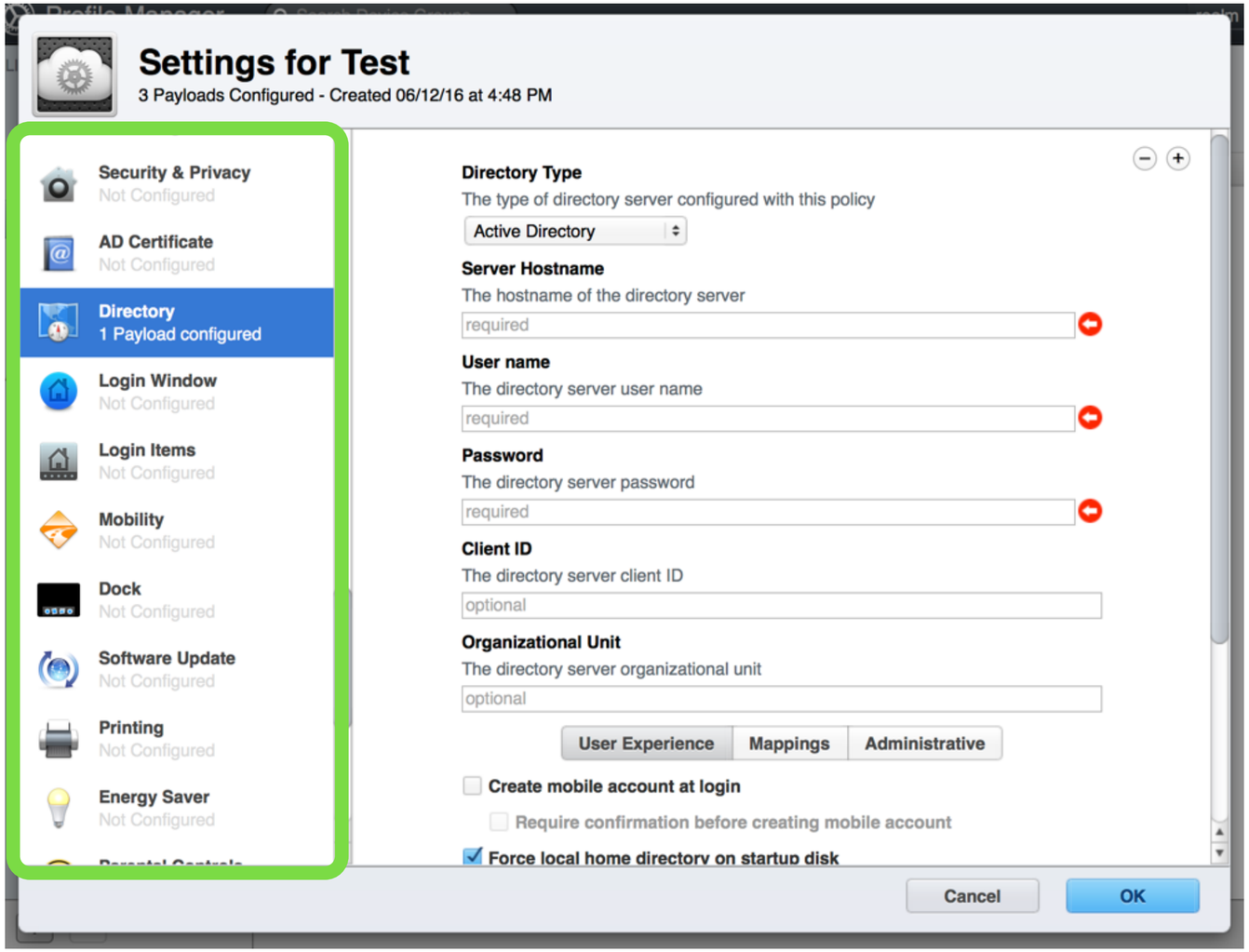
Profile Manager (OS X Server)



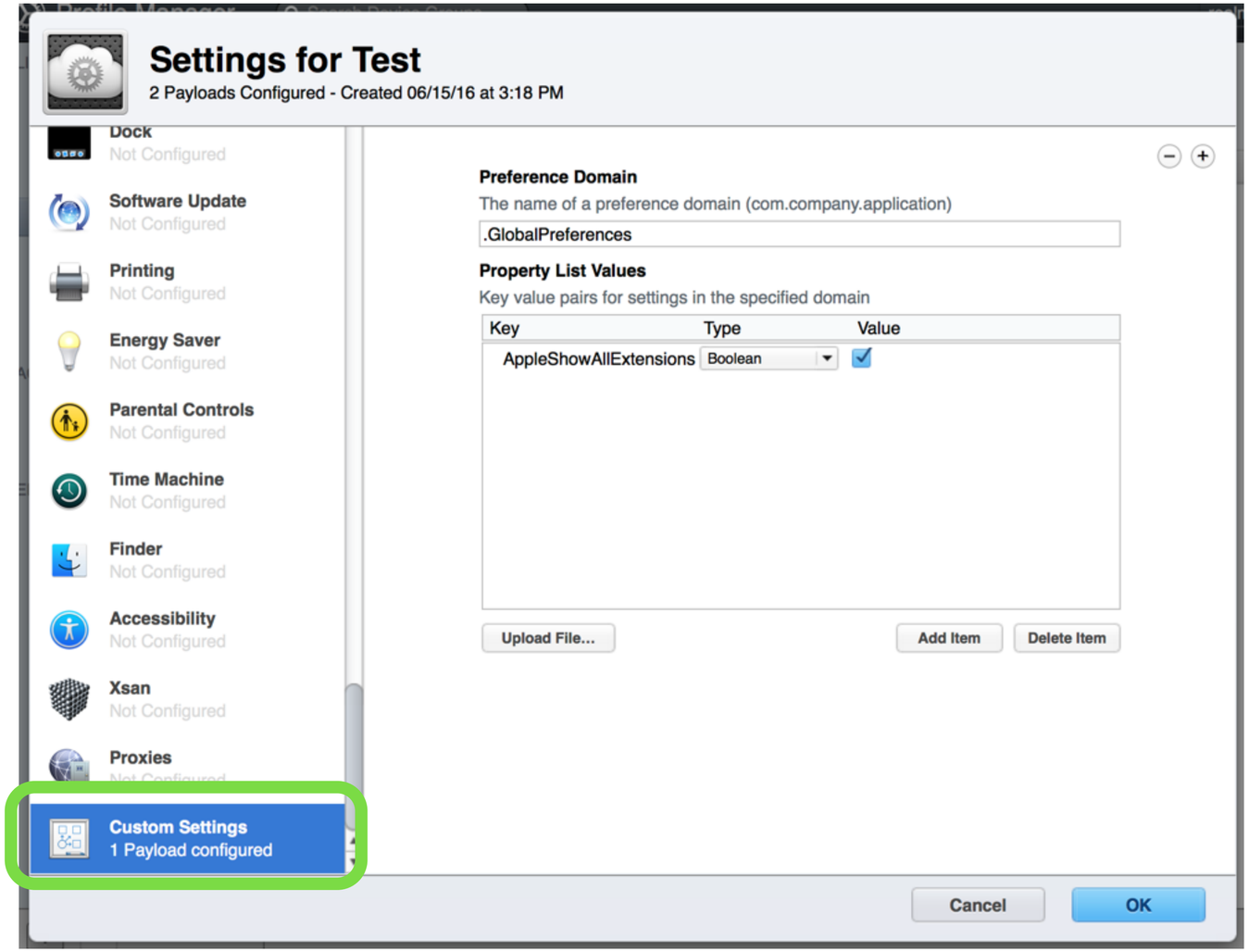
Profile Manager (OS X Server)



Profile Manager (OS X Server)



Profile Manager (OS X Server)



Configuration Profiles

- Pros
  - Portable and easy to install
  - Put lots of configuration into a single logical “blob”
  - Overall simplification of management
- Cons
  - Profile Manager sets settings as “Forced”, which often doesn’t allow user to make changes
  - “Forced” can have unexpected results with some applications
  - Not easy to combine multiple settings for the same app id across multiple profiles

Configuration Profiles

- https://github.com/timsutton/mcxToProfile
  - Lightweight way to generate profiles from defaults, plists and existing MCX (Open Directory, Local, LDAP)
  - Only handles the Custom Settings types
  - Can also set the ‘once’ or ‘often’ management state like Workgroup Manager, and ByHost prefs – not supported by Profile Manager

Additional resources

ProfileCreator (Erik Berglund):
https://github.com/ProfileCreator/ProfileCreator

Dock Master (Michael Page):
https://github.com/Error-freeIT/Dock-Master

Preference Management with Profiles
(Greg Neagle, MacSysAdmin 2015):
http://docs.macsysadmin.se/2015/video/Day2Session2.mp4

When to use what mechanism?

- Configuration Profiles are the simplest to manage: try that first
  - Especially if you know the thing you're managing runs early at login - scripting a defaults write won't help if the application is already launched, like the Finder
- For proprietary preferences, see if vendor-supported options are available for managing a setting with defaults or Profiles
- If prefs are all nested within a single key (Adobe Acrobat/Reader), can't manage these easily with a profile/defaults
  - Possible to do with more granularity via a script that can manage nested preference data, but is non-trivial to do it safely

Consider the problem, not the solution

http://xyproblem.info

Thank you!

@timsutton

@tvsutton

https://macops.ca