

**whatever you
were looking for,
you have come
to the right place**

feedback:

<http://j.mp/psumac65>

**Extending OS X
management systems
with scripting !!!**



NOT NECESSARILY A REAL iWATCH

SOURCE: http://fashion.163.com/photoview/43AJ0026/67218.html?from=tj_xgtj#p=9UI76UFN43AJ0026



AGENTS WITH SCRIPTING

Jeremy Reichman
@jaharmi

Rich Trouton
@rtrouton



Python Swift Bash

```
print("Chocolate")("/bin/echo "Peanut  
("Move over!") butter!"
```

```
#!/usr/bin/env xcrun  
#!/usr/bin/python swift -i #!/bin/bash
```



<https://twitter.com/wookiee/status/474625715023785985>

what we will cover

Scripts

- ▶ More about scripts, less about specific tools
- ▶ Where scripts fit
- ▶ Environment provided

Why

- ▶ New area or filling a gap
- ▶ Satisfy a request
- ▶ Fixing a problem

what we won't cover

- ▶ How to script
- ▶ What language(s) to use
- ▶ Text editors

Debug

- ▶ Scripts run
 - ▶ Across an entire fleet
 - ▶ Multiple times a day
- ▶ Avoid crashes, loops, and performance impacts

Debug

	Shell	Python
Exit status	<code>echo \$?</code>	
Logging	<code>logger "Message"</code>	<code>syslog.syslog("Message")</code>
Print progress	<code>echo "Message"</code>	<code>print("Message")</code>
	<code>bash -x</code>	

Exit status

Exit status	Meaning	Examples	
0	Success	<code>exit 0</code>	<code>sys.exit(0)</code>
Not 0	Some other condition	<code>exit 1</code>	<code>sys.exit(1)</code>

Python `exit()` vs. `sys.exit()`

<http://stackoverflow.com/questions/6501121/the-difference-between-exit-and-sys-exit-in-python>

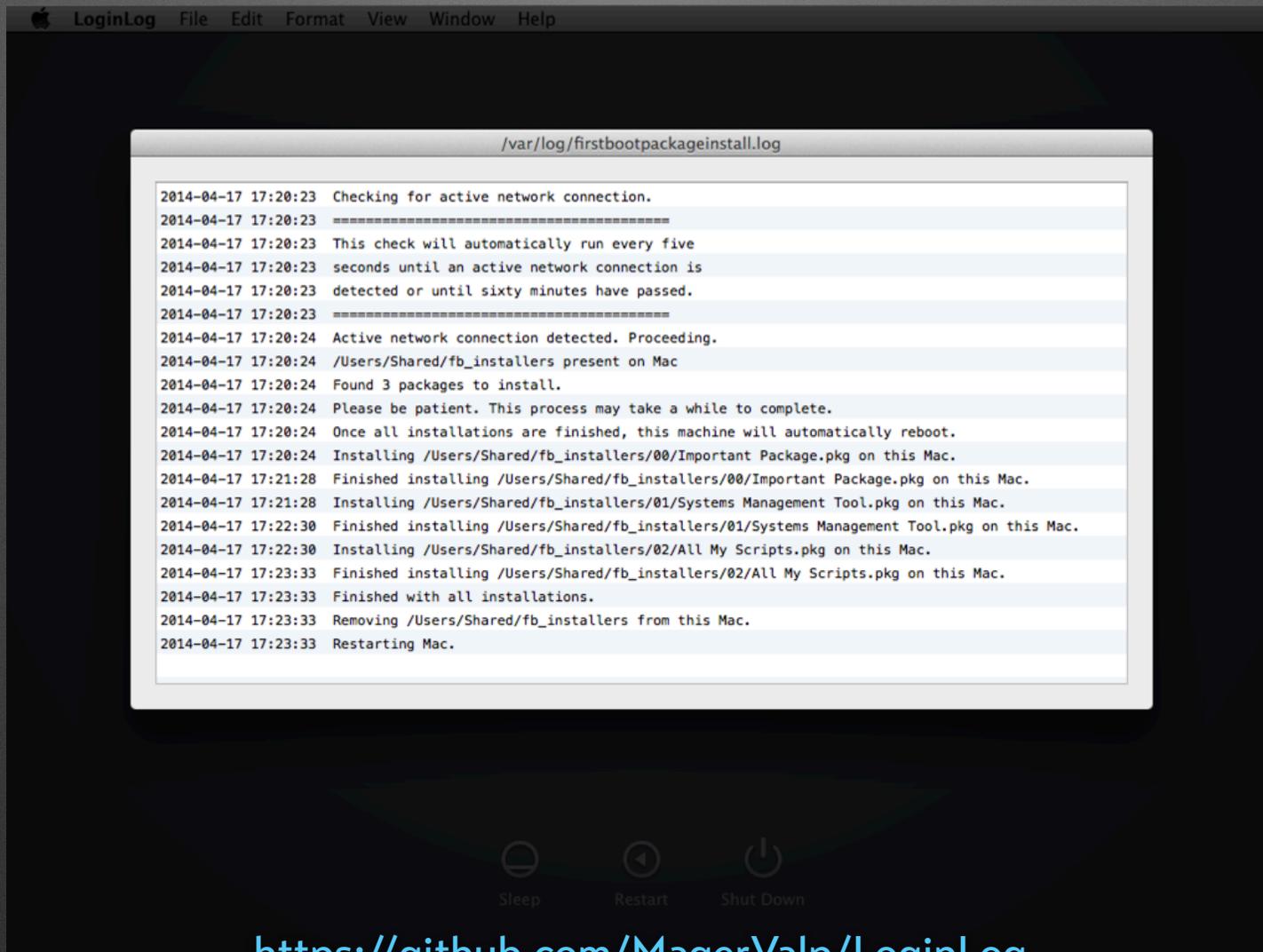
<https://docs.python.org/2/library/constants.html>

<https://docs.python.org/2/library/sys.html>

Code quality

- ▶ BBFlake / Flake8 / pyflakes
<https://flake8.readthedocs.org/>
- ▶ Run in VMs or test computer

LoginLog



The screenshot shows the LoginLog application window with a menu bar (Apple icon, LoginLog, File, Edit, Format, View, Window, Help) and a toolbar (Sleep, Restart, Shut Down). The main content is a log window titled "/var/log/firstbootpackageinstall.log" containing the following text:

```
2014-04-17 17:20:23 Checking for active network connection.
2014-04-17 17:20:23 =====
2014-04-17 17:20:23 This check will automatically run every five
2014-04-17 17:20:23 seconds until an active network connection is
2014-04-17 17:20:23 detected or until sixty minutes have passed.
2014-04-17 17:20:23 =====
2014-04-17 17:20:24 Active network connection detected. Proceeding.
2014-04-17 17:20:24 /Users/Shared/fb_installers present on Mac
2014-04-17 17:20:24 Found 3 packages to install.
2014-04-17 17:20:24 Please be patient. This process may take a while to complete.
2014-04-17 17:20:24 Once all installations are finished, this machine will automatically reboot.
2014-04-17 17:20:24 Installing /Users/Shared/fb_installers/00/Important Package.pkg on this Mac.
2014-04-17 17:21:28 Finished installing /Users/Shared/fb_installers/00/Important Package.pkg on this Mac.
2014-04-17 17:21:28 Installing /Users/Shared/fb_installers/01/Systems Management Tool.pkg on this Mac.
2014-04-17 17:22:30 Finished installing /Users/Shared/fb_installers/01/Systems Management Tool.pkg on this Mac.
2014-04-17 17:22:30 Installing /Users/Shared/fb_installers/02/All My Scripts.pkg on this Mac.
2014-04-17 17:23:33 Finished installing /Users/Shared/fb_installers/02/All My Scripts.pkg on this Mac.
2014-04-17 17:23:33 Finished with all installations.
2014-04-17 17:23:33 Removing /Users/Shared/fb_installers from this Mac.
2014-04-17 17:23:33 Restarting Mac.
```

<https://github.com/MagerValp/LoginLog>

be careful

- ▶ ``sudo`` in scripts already run with elevated privileges
- ▶ ``installer`` in package script
- ▶ ``launchctl bsexec``
- ▶ Version number comparisons

Change across OS X releases or not included

- ▶ curl
- ▶ Ruby
- ▶ diskutil
- ▶ Python "requests" module

Different environments

- ▶ Recovery HD or OS X installer

http://www.jaharmi.com/osx_mavericks_recoveryhd_executables

- ▶ DeployStudio's optional Python and Ruby

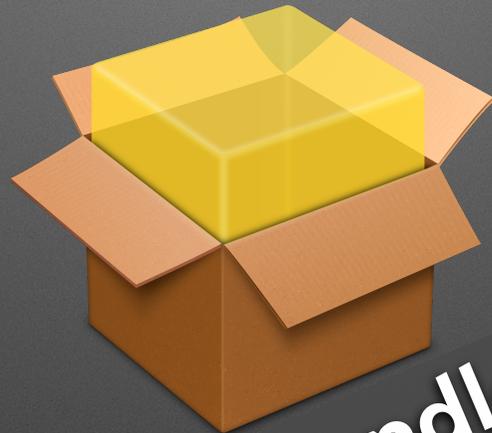
assumptions

where scripts fail

- ▶ Libraries or frameworks missing
- ▶ Loginwindow vs. logged-in user

- ▶ Network available
- ▶ DNS available
- ▶ Remote resources available
- ▶ Directory services available

Payload-free packages



Bundle

vs.

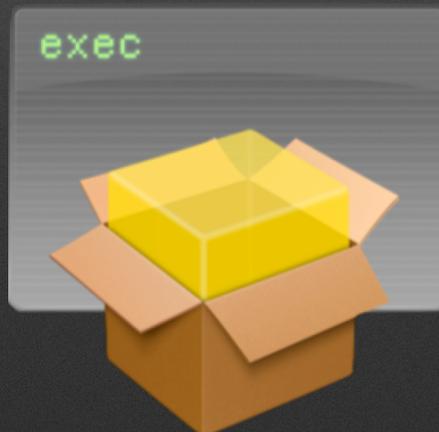
Flat

Bundle-style package scripts

Name	How used	Exit status effects
preflight	Run before files are being installed.	If the script does not return an exit status of 0, Installer will cancel the installation.
preinstall	Run before files are being installed and after the preflight script if one is defined. This script is run only if the component is being installed for the first time.	If the script does not return an exit status of 0, Installer will cancel the installation.
preupgrade	Run before files are being installed and after the preflight script if one is defined. This script is run only if the component has been previously installed.	If the script does not return 0, Installer will cancel the installation
postinstall	Run after files have been installed and before the postflight script if one is defined. This script is run only if the component is being installed for the first time.	If the script does not return 0, Installer will declare the installation failed.
postupgrade	Run after files have been installed and before the postflight script if one is defined. This script is run only if the component has been previously installed.	If the script does not return an exit status of 0, Installer will declare the installation failed.
postflight	Run after files have been installed.	If the script does not return an exit status of 0, Installer will declare the installation failed.

Bundle-style payload-free scripts

Name	How used	Exit status effects
preflight	Run before files are being installed.	If the script does not return an exit status of 0, Installer will cancel the installation.
postflight	Run after files have been installed.	If the script does not return an exit status of 0, Installer will declare the installation failed.



Script mapping

Bundle-style script

Equivalent flat script

preflight

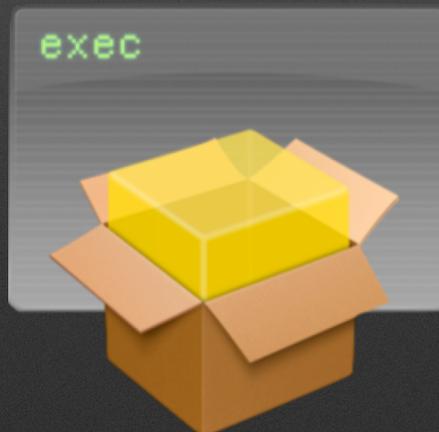
preinstall

postflight

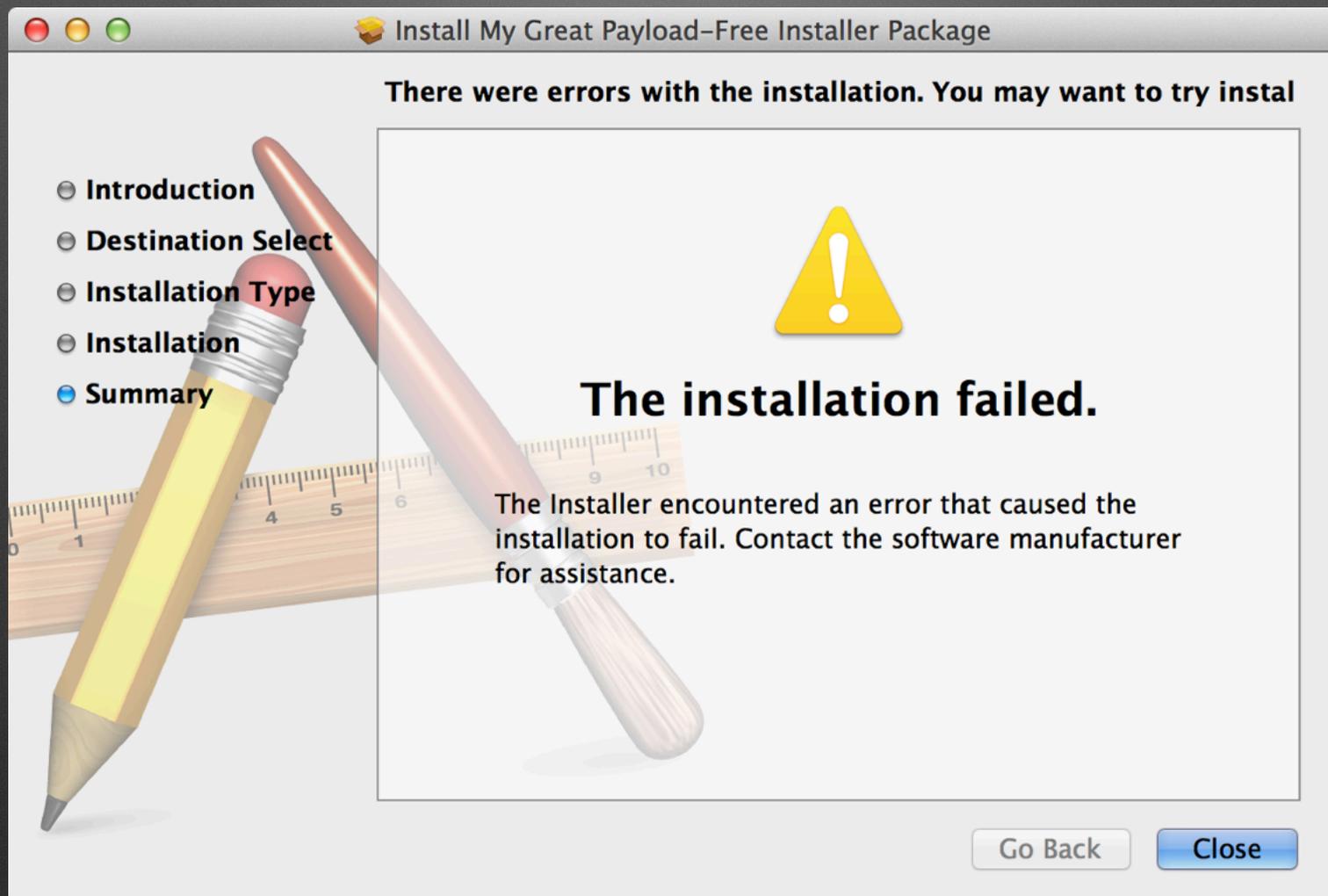
postinstall

Flat package payload-free scripts

Name	How used	Exit status effects
preinstall	Run before files are being installed.	If the script does not return an exit status of 0, Installer will cancel the installation.
postinstall	Run after files have been installed.	If the script does not return an exit status of 0, Installer will declare the installation failed.



Scripts must return an exit status of zero



Scripts must return an exit status of zero

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

```
if [[ -f "/path/to/file" ]]; then  
    /usr/sbin/do_something "/path/to/file"  
fi
```

```
exit 0
```

Installer Script Variables

Variable	What's referenced
\$0	returns the path to the script
\$1	returns the path to the package
\$2	returns the target location (for example: /Applications)
\$3	returns the target volume (for example: /Volumes/Macintosh HD)

Using Installer Script Variables

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

```
# Detects if /Users is present. If /Users is present,  
# the chflags command will unhide it
```

```
if [[ -d "$3/Users" ]]; then  
    chflags nohidden "$3/Users"  
fi
```

```
# Detects if /Users/Shared is present. If /Users/Shared is present,  
# the chflags command will unhide it
```

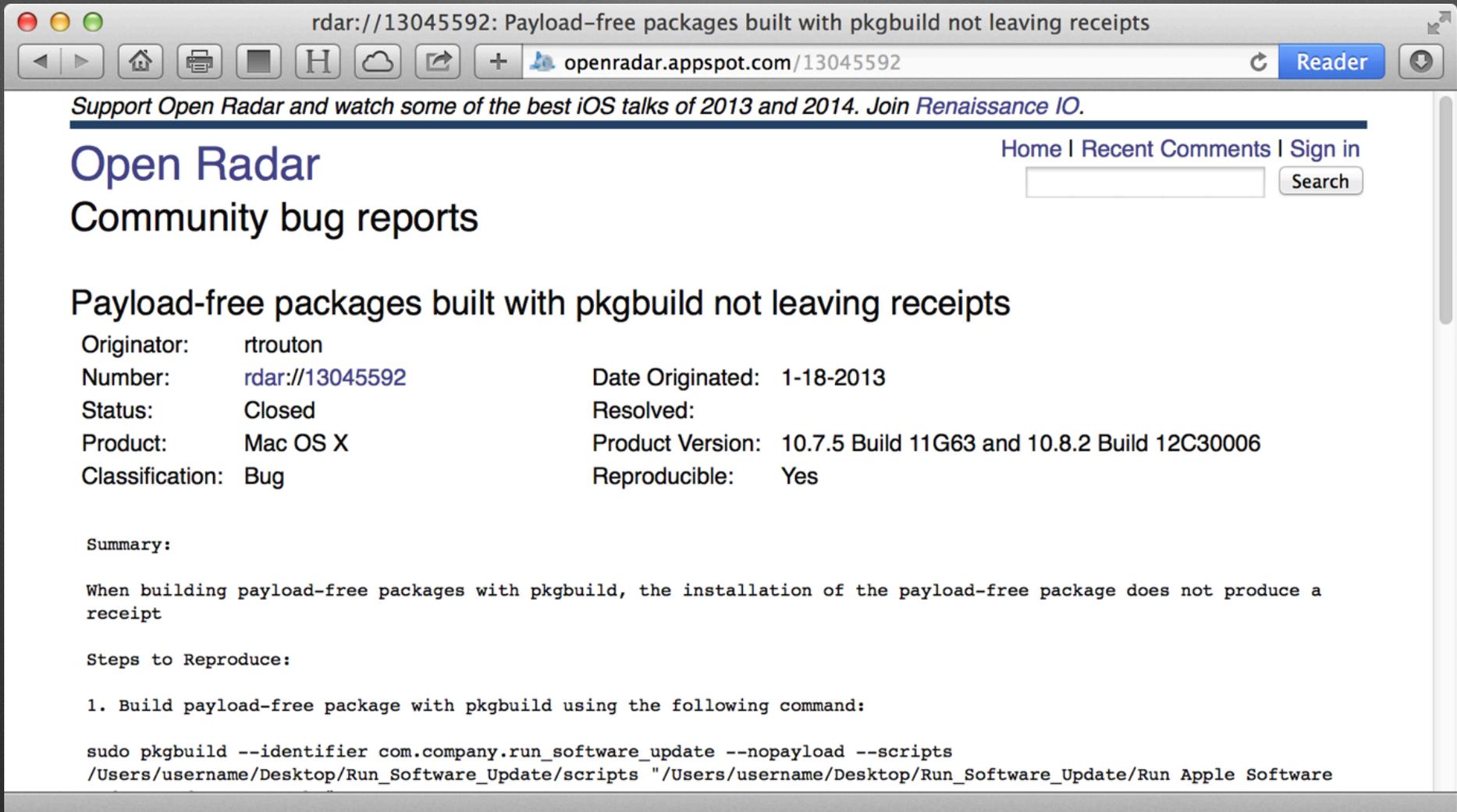
```
if [[ -d "$3/Users/Shared" ]]; then  
    chflags nohidden "$3/Users/Shared"  
fi
```

```
# Runs a permissions repair to fix the world-writable permission  
# on /Users
```

```
diskutil repairPermissions "$3"
```

```
exit 0
```

Payload-free flat packages may not leave installer receipts



The image shows a browser window displaying an Open Radar bug report. The browser's address bar shows the URL `openradar.appspot.com/13045592`. The page title is "Payload-free packages built with pkgbuild not leaving receipts". The report details include the originator (rtrouton), status (Closed), product (Mac OS X), and classification (Bug). The summary states that when building payload-free packages with pkgbuild, the installation does not produce a receipt. The steps to reproduce are listed, starting with building a payload-free package using pkgbuild with specific command-line options.

rdar://13045592: Payload-free packages built with pkgbuild not leaving receipts

openradar.appspot.com/13045592

Reader

Support Open Radar and watch some of the best iOS talks of 2013 and 2014. Join Renaissance IO.

Open Radar

Home | Recent Comments | Sign in

Community bug reports

Payload-free packages built with pkgbuild not leaving receipts

Originator: rtrouton

Number: rdar://13045592

Status: Closed

Product: Mac OS X

Classification: Bug

Date Originated: 1-18-2013

Resolved:

Product Version: 10.7.5 Build 11G63 and 10.8.2 Build 12C30006

Reproducible: Yes

Summary:

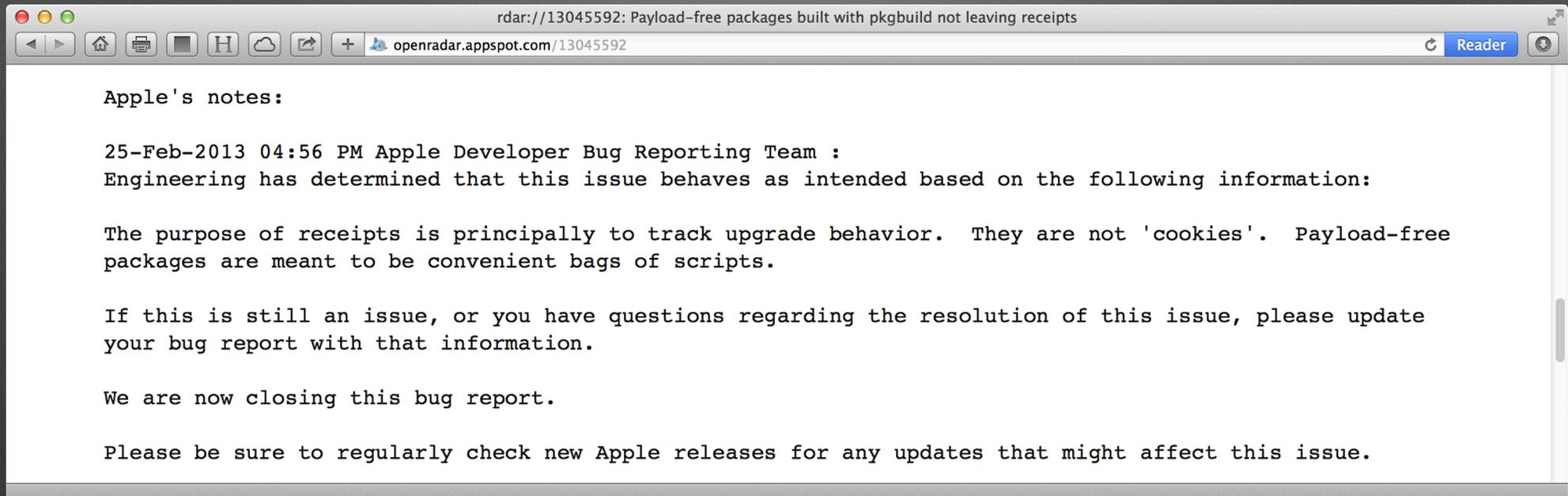
When building payload-free packages with pkgbuild, the installation of the payload-free package does not produce a receipt

Steps to Reproduce:

1. Build payload-free package with pkgbuild using the following command:

```
sudo pkgbuild --identifier com.company.run_software_update --nopayload --scripts /Users/username/Desktop/Run_Software_Update/scripts "/Users/username/Desktop/Run_Software_Update/Run Apple Software
```

Payload-free flat packages may not leave installer receipts

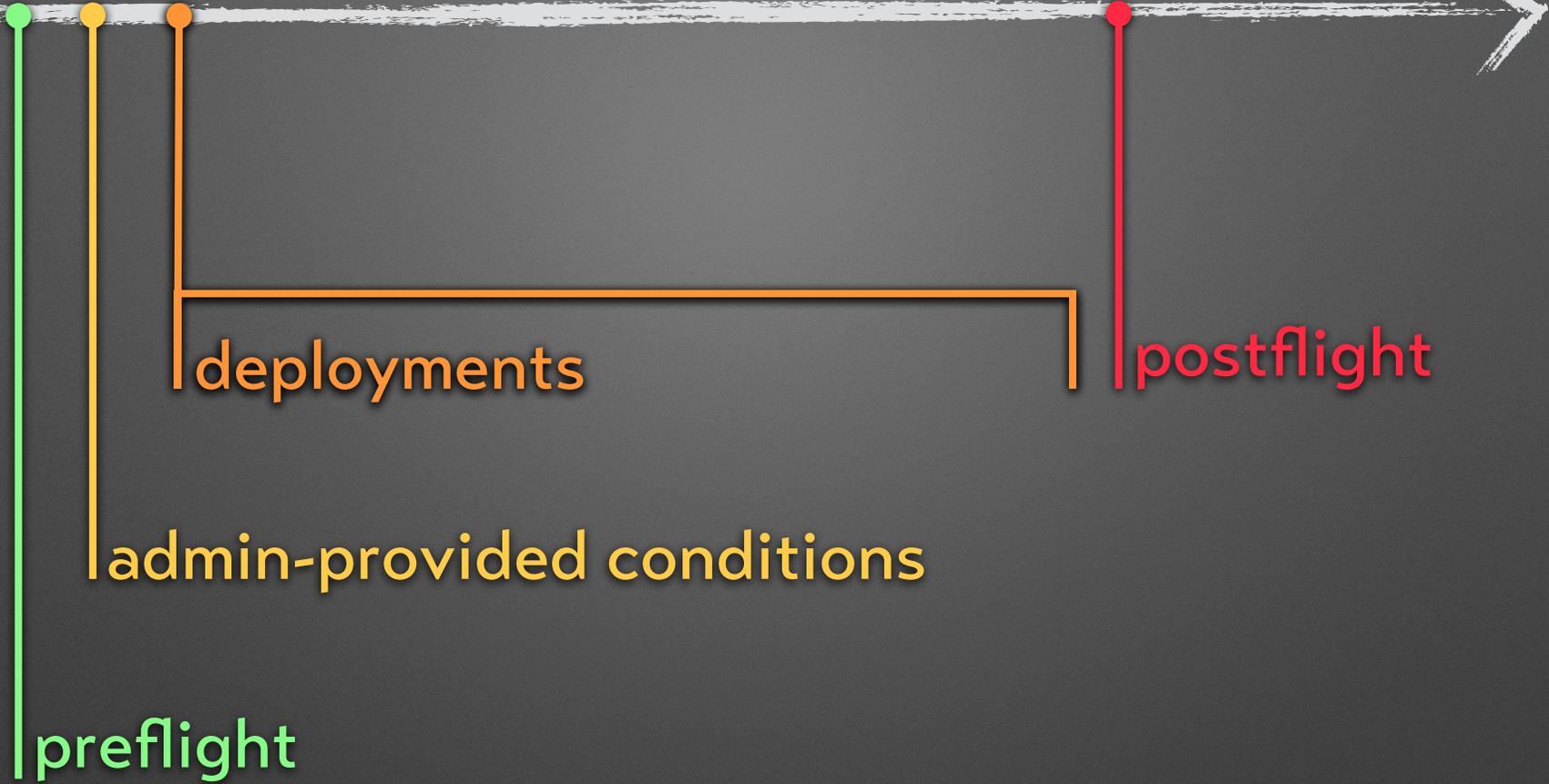


Extending Munki

Smart client

- ▶ Which client?
- ▶ Which server is checked?
- ▶ What deployments are assigned?
- ▶ What's involved with each deployment?
- ▶ Anything to do before or after checking?

Munki client check



preflight

admin-provided conditions

deployments

postflight

Add to client

- ▶ Deploy in advance of need, remove when done

Preflight

1

Postflight

1

Conditions

Multiple, as needed

- ▶ **Before** and **after** Munki run
- ▶ Exact name, no filename extension
- ▶ Same directory, ownership, permissions as `/usr/local/munki/managedsoftwareupdate`
- ▶ Passed "runtime" parameter
- ▶ Preflight status \neq 0, MSU exits

munki_version

catalogs

hostname

os_vers_patch

machine_model

machine_type

arch

Conditionals

os_vers_minor

serial number

date

os_vers_major

os_vers

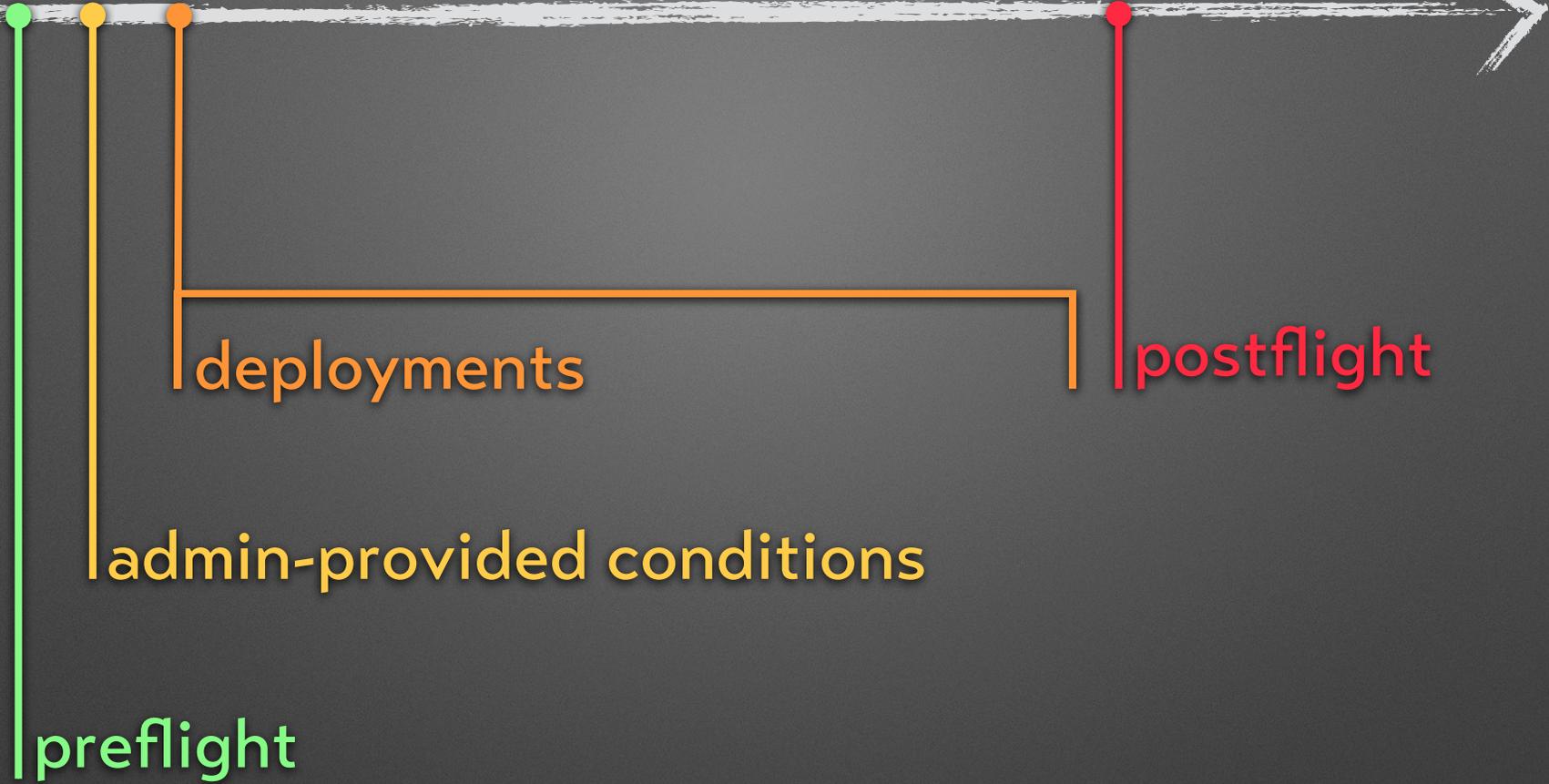
ipv4_address

Conditional scripts

- ▶ Each Munki client check
- ▶ Same ownership, permissions as pre/postflight
- ▶ Store in `/usr/local/munki/conditions`

- ▶ Scripts write key-value pairs
- ▶ *Temporary* output destination
/Library/Managed Installs/ConditionalItems.plist
- ▶ Troubleshoot via ManagedInstallReport.plist

Munki client check



preflight

admin-provided conditions

deployments

postflight

Munki client check

postflight

installcheck_script

preinstall_script

postinstall_script

uninstallcheck_script

preuninstall_script

postuninstall_script

uninstall_script

admin-provided conditions

preflight

Pkginfo file

- ▶ Minimum unit of **deployment**
- ▶ XML property list file
- ▶ Combined into *catalogs*
 - ▶ Read by smart client, which makes decisions
- ▶ **Installable conditions**

Embedded scripts

Install

Uninstall

Install check

Uninstall check
optional, but why not?

Preinstall

Preuninstall

Uninstall
(`uninstall_method: uninstall_script`)

Postinstall

Postuninstall

Type	Exit 0	Exit # 1
(Un)install check	Needed, proceed	Skip
Pre(un)install	Success	Complete but log
Post(un)install	Success	Complete but log
Package script	Success	Failure

Add to pkginfo

- ▶ Paste into existing pkginfo file in editor
- ▶ `munkiimport/makepkginfo` flags
 - ▶ Can separately version control

```
makepkginfo \  
  --installcheck_script=path/to/script1 \  
  --preinstall_script=path/to/script2 \  
  --postinstall_script=path/to/script3
```

Example

`nopkg`

- ▶ `installcheck_script`

- ▶ `postinstall`

- ▶ `uninstallcheck_script`

- ▶ `uninstall`

- ▶ `postuninstall`

installcheck_script

```
#!/bin/bash

/usr/bin/dscl -q . -read Groups/com.apple.local.ard_admin > /dev/null 2>&1
DSCL_EXIT_STATUS=$?

case ${DSCL_EXIT_STATUS} in
    0)
        # Group exists
        exit 1
        ;;
    56)
        # Group doesn't exist but needs to
        exit 0
        ;;
    *)
        # Something else but in Keynote no one can hear you scream
        exit 0
        ;;
esac
```

installcheck_script

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

```
/usr/bin/dscl -q . -read Groups/com.apple.local.ard_admin > /dev/null 2>&1
```

```
DSCL_EXIT_STATUS=$?
```

```
case ${DSCL_EXIT_STATUS} in
```

```
0)
```

```
    # Group exists
```

```
    exit 1
```

```
    ;;
```

```
56)
```

```
    # Group doesn't exist but needs to
```

```
    exit 0
```

```
    ;;
```

```
*)
```

```
    # Something else but in Keynote no one can hear you scream
```

```
    exit 0
```

```
    ;;
```

```
esac
```

installcheck_script

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

```
/usr/bin/dscl -q . -read Groups/com.apple.local.ard_admin > /dev/null 2>&1
```

```
DSCL_EXIT_STATUS=$?
```

```
case ${DSCL_EXIT_STATUS} in
```

```
0)
```

```
    # Group exists
```

```
    exit 1
```

```
    ;;
```

```
56)
```

```
    # Group doesn't exist but needs to
```

```
    exit 0
```

```
    ;;
```

```
*)
```

```
    # Something else but in Keynote no one can hear you scream
```

```
    exit 0
```

```
    ;;
```

```
esac
```

installcheck_script

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

```
/usr/bin/dscl -q . -read Groups/com.apple.local.ard_admin > /dev/null 2>&1  
DSCL_EXIT_STATUS=$?
```

```
case ${DSCL_EXIT_STATUS} in
```

```
0)
```

```
    # Group exists
```

```
    exit 1
```

```
    ;;
```

```
56)
```

```
    # Group doesn't exist but needs to
```

```
    exit 0
```

```
    ;;
```

```
*)
```

```
    # Something else but in Keynote no one can hear you scream
```

```
    exit 0
```

```
    ;;
```

```
esac
```

installcheck_script

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

```
/usr/bin/dscl -q . -read Groups/com.apple.local.ard_admin > /dev/null 2>&1  
DSCL_EXIT_STATUS=$?
```

```
case ${DSCL_EXIT_STATUS} in
```

```
0)
```

```
    # Group exists
```

```
    exit 1
```

```
    ;;
```

```
56)
```

```
    # Group doesn't exist but needs to
```

```
    exit 0
```

```
    ;;
```

```
*)
```

```
    # Something else but in Keynote no one can hear you scream
```

```
    exit 0
```

```
    ;;
```

```
esac
```

```
$ ./installcheck.sh
```

```
$ echo $?
```

```
0
```

postinstall

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

```
/usr/sbin/dseditgroup \  
-o create \  
-n /Local/Default \  
-r "Apple Remote Desktop admin" \  
-c "ARD administration group" \  
-i 808 \  
-t group \  
com.apple.local.ard_admin
```

```
$ ./installcheck.sh
```

```
$ echo $?
```

1

uninstall

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

```
/usr/sbin/dseditgroup \  
-o delete \  
-n /Local/Default \  
com.apple.local.ard_admin \  
> /dev/null 2>&1
```

uninstallcheck_script

```
#!/bin/bash

/usr/bin/dscl -q . -read Groups/com.apple.local.ard_admin > /dev/null 2>&1
DSCL_EXIT_STATUS=$?

case ${DSCL_EXIT_STATUS} in
    0)
        # Group exists, needs to be removed
        exit 0
        ;;
    56)
        # Group doesn't exist, skip
        exit 1
        ;;
    *)
        # Something else, draw four
        exit 4
        ;;
esac
```

uninstallcheck_script

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

```
/usr/bin/dscl -q . -read Groups/com.apple.local.ard_admin > /dev/null 2>&1  
DSCL_EXIT_STATUS=$?
```

```
case ${DSCL_EXIT_STATUS} in
```

```
0)
```

```
# Group exists, needs to be removed
```

```
exit 0
```

```
;;
```

Reverse exit codes from installcheck_script

```
56)
```

```
# Group doesn't exist, skip
```

```
exit 1
```

```
;;
```

```
*)
```

```
# Something else, draw four
```

```
exit 4
```

```
;;
```

```
esac
```

```
makepkginfo \  
  --nopkg \  
  --catalog=testing \  
  --installcheck_script=installcheck.sh \  
  --postinstall_script=postinstall.sh \  
  --uninstallcheck_script=uninstallcheck.sh \  
  --uninstall_script=uninstall.sh | bbedit
```

```
<?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>autoremove</key>
  <false/>
  <key>installcheck_script</key>
  <string>#!/bin/bash

/usr/bin/dscl -q . -read Groups/com.apple.local.ard_admin &gt; /dev/null 2&gt;&1
DSCL_EXIT_STATUS=$?

case ${DSCL_EXIT_STATUS} in
  0)
    # Group exists
    exit 1
    ;;
  56)
    # Group doesn't exist but needs to
    exit 0
    ;;
  *)
    # Something else but in Keynote no one can hear you scream
    exit 0
    ;;
esac</string>
  <key>installer_type</key>
  <string>nopkg</string>
  <key>minimum_os_version</key>
  <string>10.4.0</string>
  <key>postinstall_script</key>
  <string>#!/bin/bash
```

Extending JAMF's Casper Suite

Smart server

Extension Attributes

The screenshot shows the Casper Admin web interface. The top navigation bar includes 'Computers', 'Mobile Devices', and 'Users' tabs, with the user 'casperadmin' and a settings gear icon. The left sidebar contains a menu with categories: System Settings, Global Management, Computer Management, Mobile Device Management, Network Organization, and JSS Information. The main content area is titled 'Extension Attributes (7)' and features three buttons: '+ New', 'New From Template', and 'Upload'. Below these buttons is a table with the following entries:

Name
CasperCheck Installed
Checking for Java build M4508
Identify Local Admins
Identify Remote Management Accounts
JSS Certificate Validation
Verify Certificate Based Communication
Verify MDM Enrollment

Policies

The screenshot shows the 'Software Update' configuration page in Casper Admin. The top navigation bar is the same as the previous screenshot. The left sidebar menu is expanded to show 'Policies' and its sub-items: Configuration Profiles, Managed Preferences, Restricted Software, PreStage Imaging, Smart Computer Groups, Static Computer Groups, Enrollment Invitations, and Management Settings. The main content area is titled 'Software Update' and has tabs for 'Options', 'Scope', 'Self Service', and 'User Interaction'. The 'Options' tab is active, showing a list of categories on the left: General, Packages (0 Packages), Software Updates (Not Configured), Scripts (1 Script), Printers (0 Printers), Disk Encryption (Not Configured), Dock Items (0 Dock Items), and Local Accounts. The 'Scripts' category is selected, displaying the configuration for 'Selectable_SoftwareUpdate.sh'. The configuration includes a 'Priority' dropdown set to 'After', a 'Parameter Values' section with a description, and six empty input fields for parameters. 'Cancel' and 'Save' buttons are at the bottom right.



casperadmin

- System Settings >
- Global Management >
- Computer Management >**
- Mobile Device Management >
- Network Organization >

JSS Information >

Computer Management



Categories



Packages



Scripts



Printers



Directory Bindings



Disk Encryption Configurations



Dock Items



Configurations

Computer Management - Server Infrastructure



JDS Instances



Cloud Distribution Point



File Share Distribution Points



Software Update Servers



NetBoot Servers

Computer Management - Management Framework



User-Initiated Enrollment



Inventory Collection



Inventory Display



Check-In



Security



Extension Attributes



Autorun Imaging



Self Service



Self Service Plug-ins



Peripheral Types



Removable MAC Addresses



- System Settings >
- Global Management >
- Computer Management >
- Mobile Device Management >
- Network Organization >
- JSS Information >

Selectable_SoftwareUpdate.sh

General Script Options Limitations

Script Contents

```
1 #!/bin/bash
2
3 ## Script Name:      Selectable_SoftwareUpdate.sh
4 ## Script Author:    Mike Morales
5
6 ## Path to cocoaDialog (customize to your own location)
7 cdPath="/Library/Application Support/JAMF/bin/cocoaDialog.app/Contents/MacOS/cocoaDialog"
8
9 ## Get minor version of OS X
10 osVers=$( sw_vers -productVersion | cut -d. -f2 )
11
12 ## Set appropriate Software Update icon depending on OS version
13 if [[ "$osVers" -lt 8 ]]; then
14     swuIcon="/System/Library/CoreServices/Software Update.app/Contents/Resources/Software Update.icns"
15 else
16     swuIcon="/System/Library/CoreServices/Software Update.app/Contents/Resources/SoftwareUpdate.icns"
17 fi
18
19 ## Set appropriate Restart icon depending on OS version
20 if [[ "$osVers" == "9" ]]; then
21     restartIcon="/System/Library/CoreServices/loginwindow.app/Contents/Resources/Restart.tiff"
22 else
23     restartIcon="/System/Library/CoreServices/loginwindow.app/Contents/Resources/Restart.png"
24 fi
25
26
```

Cancel

Save

JavaScript

PHP

AppleScript



exec

Shell

Perl

Python

Expect

Ruby

Many More!

Reserved Scripting Variables

Variable	What's referenced
\$1	Mount point of the target drive
\$2	Computer name
\$3	Username

Non-Reserved Script Variables

The screenshot displays the JAMF Software console interface. On the left is a dark blue navigation sidebar with the following menu items: System Settings, Global Management, Computer Management (highlighted), Mobile Device Management, Network Organization, and JSS Information. The top navigation bar includes icons for Computers, Mobile Devices, and Users, along with the user name 'casperadmin' and a settings gear icon. The main content area is titled 'Selectable_SoftwareUpdate.sh' and features four tabs: General, Script, Options, and Limitations. The 'Options' tab is active, showing a 'Priority' dropdown set to 'After', a 'Parameter Labels' section with a descriptive note, and eleven input fields labeled 'Parameter 4' through 'Parameter 11'. At the bottom right of the configuration area are 'Cancel' and 'Save' buttons. The JAMF Software logo and copyright information are visible in the bottom left corner of the sidebar.

System Settings >
Global Management >
Computer Management >
Mobile Device Management >
Network Organization >
JSS Information >

Computers Mobile Devices Users casperadmin ⚙

Selectable_SoftwareUpdate.sh

General Script Options Limitations

Priority
Priority to use for running the script in relation to other actions during imaging
After ▾

Parameter Labels
Labels to use for script parameters. Parameters 1 through 3 are predefined as mount point, computer name, and username

Parameter 4

Parameter 5

Parameter 6

Parameter 7

Parameter 8

Parameter 9

Parameter 10

Parameter 11

Cancel Save

JAMF software
© 2002-2014 JAMF Software, LLC.

Parameter Values

Values for script parameters. Parameters 1-3 are predefined as mount point, computer name, and username

Username

Parameter 5

Parameter 6

Parameter 7

Parameter 8

Before and After Scripts

Casper Online

Options Scope Self Service User Interaction

General

Packages
0 Packages

Software Updates
Not Configured

Scripts
1 Script

Printers
0 Printers

Disk Encryption

Scripts

iscasperonline.sh - +

Priority
Priority to use for running the script in relation to other actions
Before ▾

Parameter Values
Values for script parameters. Parameters 1-3 are predefined as mount point, computer name, and username

Parameter 4

Cancel Save

Casper Online

Options Scope Self Service User Interaction

General

Packages
0 Packages

Software Updates
Not Configured

Scripts
1 Script

Printers
0 Printers

Disk Encryption

Scripts

iscasperonline.sh - +

Priority
Priority to use for running the script in relation to other actions
After ▾

Parameter Values
Values for script parameters. Parameters 1-3 are predefined as mount point, computer name, and username

Parameter 4

Cancel Save

Triggering Casper Policies

Trigger

Event(s) to use to initiate the policy

Startup

When a computer starts up. A startup script that checks for policies must be configured in the JSS for this to work

Login

When a user logs in to a computer. A login hook that checks for policies must be configured in the JSS for this to work

Logout

When a user logs out of a computer. A logout hook that checks for policies must be configured in the JSS for this to work

Network State Change

When a computer's network state changes (e.g. when the network connection changes, when the computer name changes, when the IP address changes)

Enrollment Complete

Immediately after a computer completes the enrollment process

Recurring Check-in

At the recurring check-in frequency configured in the JSS

Custom

At a custom event

Triggering Casper Policies

Trigger

Event(s) to use to initiate the policy

Startup

When a computer starts up. A startup script that checks for policies must be configured in the JSS for this to work

Login

When a user logs in to a computer. A login hook that checks for policies must be configured in the JSS for this to work

Logout

When a user logs out of a computer. A logout hook that checks for policies must be configured in the JSS for this to work

Network State Change

When a computer's network state changes (e.g. when the network connection changes, when the computer name changes, when the IP address changes)

Enrollment Complete

Immediately after a computer completes the enrollment process

Recurring Check-in

At the recurring check-in frequency configured in the JSS

Custom

At a custom event

Custom Event

Custom event to use to initiate the policy

Extension Attributes

The screenshot displays the JAMF System Settings application interface. On the left is a dark blue sidebar with a navigation menu containing: System Settings, Global Management, Computer Management (highlighted), Mobile Device Management, Network Organization, and JSS Information. The main content area is white and organized into sections. At the top, there are navigation tabs for 'Computers', 'Mobile Devices', and 'Users', along with a user profile 'casperadmin' and a settings gear icon. The 'Computer Management' section includes icons for Categories, Packages, Scripts, Printers, Directory Bindings, and Disk Encryption Configurations. The 'Computer Management - Server Infrastructure' section includes JDS Instances, Cloud Distribution Point, File Share Distribution Points, Software Update Servers, and NetBoot Servers. The 'Computer Management - Management Framework' section includes User-Initiated Enrollment, Inventory Collection, Inventory Display, Check-In, Security, and Extension Attributes (highlighted with a red border). Below this are Autorun Imaging, Self Service, Self Service Plug-ins, Peripheral Types, and Removable MAC Addresses. The JAMF Software logo and copyright information are visible in the bottom left corner.

Computers Mobile Devices Users casperadmin

Computer Management

Categories Packages Scripts Printers Directory Bindings Disk Encryption Configurations

Dock Items Configurations

Computer Management - Server Infrastructure

JDS Instances Cloud Distribution Point File Share Distribution Points Software Update Servers NetBoot Servers

Computer Management - Management Framework

User-Initiated Enrollment Inventory Collection Inventory Display Check-In Security **Extension Attributes**

Autorun Imaging Self Service Self Service Plug-ins Peripheral Types Removable MAC Addresses

JAMF software
© 2002-2014 JAMF Software, LLC.

Extension Attributes

Extension Attributes run on all devices

- Extension Attributes cannot be targeted to certain machines

Extension Attributes are run automatically

- Extension Attributes gather information from client machines when the Casper agent runs an inventory update

Extension Attributes

The screenshot displays the Casper Admin web interface. On the left is a dark blue sidebar with a logo at the top and a list of navigation items: System Settings, Global Management, Computer Management (highlighted), Mobile Device Management, Network Organization, and JSS Information. The top navigation bar includes icons for Computers, Mobile Devices, and Users, along with the user name 'casperadmin' and a settings gear icon. The main content area is titled 'Extension Attributes (7)' and features three action buttons: '+ New', 'New From Template', and 'Upload'. Below these buttons is a table listing seven extension attributes, each with a green up arrow icon on the right side.

Name	
<u>CasperCheck Installed</u>	▲
<u>Checking for Java build M4508</u>	
<u>Identify Local Admins</u>	
<u>Identify Remote Management Accounts</u>	
<u>JSS Certificate Validation</u>	
<u>Verify Certificate Based Communication</u>	
<u>Verify MDM Enrollment</u>	

Extension Attributes

Display Name

Display name for the extension attribute

My Great Extension Attribute

Description

Description to display for the extension attribute

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque tempor dolor eget purus semper tempor. Ut venenatis dolor elit, non blandit odio euismod quis

Data Type

Type of data being collected

String

Inventory Display

Category in which to display the extension attribute in the JSS

General

Input Type

Input type to use to populate the extension attribute

Text Field

Recon Display

Pane on which to display the extension attribute in Recon

Extension Attributes

Display Name

Display name for the extension attribute

My Great Extension Attribute

Description

Description to display for the extension attribute

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque tempor dolor eget purus semper tempor. Ut venenatis dolor elit, non blandit odio euismod quis

Data Type

Type of data being collected

Date (YYYY-MM-DD hh:mm:ss)

Inventory Display

Category in which to display the extension attribute in the JSS

General

Input Type

Input type to use to populate the extension attribute

Text Field

Recon Display

Pane on which to display the extension attribute in Recon

Extension Attributes

Data Type

Type of data being collected

Integer

Inventory Display

Category in which to display the extension attribute in the JSS

General

Input Type

Input type to use to populate the extension attribute

Text Field

Recon Display

Pane on which to display the extension attribute in Recon

Extension Attributes

Extension Attributes

Display Name

Display name for the extension attribute

Description

Description to display for the extension attribute

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque tempor dolor eget purus semper tempor. Ut venenatis dolor elit, non blandit odio euismod quis

Data Type

Type of data being collected

Inventory Display

Category in which to display the extension attribute in the JSS

Input Type

Input type to use to populate the extension attribute

Script

Script to use to collect data on OS X computers

1	
---	--

Extension Attributes

```
<result>
```

```
$info_for_Casper_goes_here
```

```
</result>
```

Extension Attributes

```
#!/bin/sh
```

```
ADDomainCheck=`dsconfigad -show | awk '/Active  
Directory Domain/{print $NF}`
```

```
if [ "$ADDomainCheck" = "" ]; then  
    result="Not Bound to Active Directory"  
elif [ "$ADDomainCheck" != "" ]; then  
    result=$ADDomainCheck  
fi
```

```
echo "<result>$result</result>"
```

Now for a story....

A rectangular terminal window with rounded corners and a dark gray background. The word "exec" is displayed in a bright green, monospaced font at the top left of the window. The rest of the window is empty, suggesting a command prompt or a simple execution environment.

```
exec
```



Per-OS Driver Installers

10.5.x – Xerox Print Driver **2.94.3**

10.6.x – Xerox Print Driver **2.112.0**

10.7.x and higher – Xerox Print Driver **2.113.0**

Desired Outcome

A. Casper will check the Mac to see if the needed Xerox drivers are installed.

B. If no Xerox drivers or older Xerox drivers are detected, the appropriate Xerox print drivers will be installed before the requested printer is set up.

C. If the installed drivers are the same version or higher as the print drivers available via Self Service, no drivers are installed and Casper will proceed to set up the printer using the installed drivers.

Mac OS X Finder window titled "PDEs". The window shows a hierarchical view of the file system. The sidebar on the left includes "Mac HD" and "Network". The main pane is divided into several columns: "Applications" (Library, opt, System, Users), "Preferences" (Printers, PrivilegedHelperTools, Python, QuickLook, QuickTime, Receipts, Ruby, Sandbox, Screen Savers, ScriptingAdditions, Scripts, Security, Sophos Anti-Virus, Speech, Spelling), "Canon" (Icons, InstalledPrinters.plist, PPDs, Xerox), "Filters" (PDEs, Profiles, Tools), and a selected file "XeroxFeatures.plugin". The right pane displays the file's icon and metadata.

Name	XeroxFeatures.plugin
Kind	Plug-in
Size	100.9 MB
Created	Thursday, October 3, 2013 at 6:11 AM
Modified	Thursday, October 3, 2013 at 6:11 AM
Last opened	Thursday, October 3, 2013 at 6:11 AM
Version	2.113.0

```

#!/bin/bash

# Determine OS version
OSVERS=$(sw_vers -productVersion | awk -F. '{print $2}')

# Check /Library/Printers/Xerox/PDEs/XeroxFeatures.plugin for the CFBundleShortVersionString key value. It should match the version of the installed drivers.
installed_driver=$(defaults read "/Library/Printers/Xerox/PDEs/XeroxFeatures.plugin/Contents/Info" CFBundleShortVersionString)
installed_version=$(echo "$installed_driver" | sed 's/[\._-]//g')
button1="OK"
jamfHelper="/Library/Application Support/JAMF/bin/jamfHelper.app/Contents/MacOS/jamfHelper"
icon="/System/Library/CoreServices/CoreTypes.bundle/Contents/Resources/AlertNoteIcon.icns"

if [[ ${OSVERS} -eq 5 ]]; then

# If the Mac is running 10.5.x, specify the current driver version by setting parameter 5 in the script on the JSS

jss_driver="$5"
driver_version=$(echo "$jss_driver" | sed 's/[\._-]//g')
dialog="The needed Xerox printer drivers have not been detected. Installing Xerox $jss_driver Print Drivers before adding the requested printer."
description="echo "$dialog"

if [[ ${installed_version} -ge ${driver_version} ]]; then
echo "Xerox $installed_driver Print Drivers installed"
fi

if [[ ${installed_version} -lt ${driver_version} ]]; then
echo "Xerox $jss_driver Print Drivers not installed. Installing Xerox $jss_driver Print Drivers"
"$jamfHelper" -windowType utility -description "$description" -button1 "$button1" -icon "$icon" -timeout 20
jamf policy -trigger companyxerodrivs
fi
fi

if [[ ${OSVERS} -eq 6 ]]; then

# If the Mac is running 10.6.x, specify the current driver version by setting parameter 6 in the script on the JSS

jss_driver="$6"
driver_version=$(echo "$jss_driver" | sed 's/[\._-]//g')
dialog="The needed Xerox printer drivers have not been detected. Installing Xerox $jss_driver Print Drivers before adding the requested printer."
description="echo "$dialog"

if [[ ${installed_version} -ge ${driver_version} ]]; then
echo "Xerox $installed_driver Print Drivers installed"
fi

if [[ ${installed_version} -lt ${driver_version} ]]; then
echo "Xerox $jss_driver Print Drivers not installed. Installing Xerox $jss_driver Print Drivers"
"$jamfHelper" -windowType utility -description "$description" -button1 "$button1" -icon "$icon" -timeout 20
jamf policy -trigger companyxerodrivs
fi
fi

if [[ ${OSVERS} -ge 7 ]]; then

# If the Mac is running 10.7.x or higher, specify the current driver version by setting parameter 7 in the script on the JSS

jss_driver="$7"
driver_version=$(echo "$jss_driver" | sed 's/[\._-]//g')
dialog="The needed Xerox printer drivers have not been detected. Installing Xerox $jss_driver Print Drivers before adding the requested printer."
description="echo "$dialog"

if [[ ${installed_version} -ge ${driver_version} ]]; then
echo "Xerox $installed_driver Print Drivers installed"
fi

if [[ ${installed_version} -lt ${driver_version} ]]; then
echo "Xerox $jss_driver Print Drivers not installed. Installing Xerox $jss_driver Print Drivers"
jamf displayMessage -message "$dialog"
jamf policy -trigger companyxerodrivs
fi
fi

exit 0

```

<https://gist.github.com/rtrouton/8917576/>

install_company_xerox_printer_drivers.sh

Printer Driver Install Scripts

Don't Run Run Before Run After

Parameter 4 Not Labeled:

Parameter 5 Not Labeled:

Parameter 6 Not Labeled:

Parameter 7 Not Labeled:

Parameter 8 Not Labeled:

Parameter 9 Not Labeled:

Parameter 10 Not Labeled:

Parameter 11 Not Labeled:

- Self Service
- How to Use Self Service
Using Self Service to Install Software
- X
- ?
- WOP

Featured

 <p>Reception Canon IR5255</p> <p>ADD PRINTER ⓘ</p>	 <p>Xerox Color</p> <p>ADD PRINTER ⓘ</p>	 <p>Canon IR5051</p> <p>ADD PRINTER ⓘ</p>
 <p>Canon IR5051</p> <p>ADD PRINTER ⓘ</p>	 <p>Canon IR5030</p> <p>ADD PRINTER ⓘ</p>	 <p>Canon IR5030</p> <p>ADD PRINTER ⓘ</p>
 <p>Canon IR5030</p> <p>ADD PRINTER ⓘ</p>	 <p>Canon IR3225</p> <p>ADD PRINTER ⓘ</p>	 <p>Canon IR5051</p> <p>ADD PRINTER ⓘ</p>
 <p>Canon IR5051</p> <p>ADD PRINTER ⓘ</p>	 <p>Canon IR5051</p> <p>ADD PRINTER ⓘ</p>	 <p>Canon IR5051</p> <p>ADD PRINTER ⓘ</p>
 <p>Canon IR5030</p> <p>ADD PRINTER ⓘ</p>	 <p>Canon IR5051</p> <p>ADD PRINTER ⓘ</p>	 <p>Canon IR5051</p> <p>ADD PRINTER ⓘ</p>

Categories

- Featured
- Printer Drivers
- Printers
- Software - Adobe
- Software - AntiVirus
- Software - Apple Quicktime
- Software - Browser Support
- Software - Communication
- Software - Development Tools
- Software - Dropbox
- Software - Internet Browsers
- Software - Java
- Software - Productivity
- Software - Scientific Software
- Software - X11

Self Service

In Progress 1
Installing Items

How to Use Self Service
Using Self Service to Install Software and

X

?

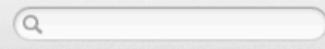
WOP

Featured

 Canon IR5255 ADD PRINTER	<div data-bbox="798 430 1533 755"><p>The needed Xerox printer drivers have not been detected. Installing Xerox 2.113.0 Print Drivers before adding the requested printer.</p><p>OK</p></div>	
 Canon IR5030 ADD PRINTER	 Canon IR3225 ADD PRINTER	 Canon IR5051 ADD PRINTER
 Canon IR5051 ADD PRINTER	 Canon IR5051 ADD PRINTER	 Canon IR5051 ADD PRINTER
 Canon IR5051 ADD PRINTER	 Canon IR5030 ADD PRINTER	 Canon IR5051 ADD PRINTER
 Canon IR5051 ADD PRINTER		

Categories

- Featured
- Printer Drivers
- Printers
- Software - Adobe
- Software - AntiVirus
- Software - Apple Quicktime
- Software - Browser Support
- Software - Communication
- Software - Development Tools
- Software - Dropbox
- Software - Internet Browsers
- Software - Java
- Software - Productivity
- Software - Scientific Software
- Software - X11



- Self Service
- Completed
1 succeeded
- How to Use Self Service
Using Self Service to Install Software and
- [X] [Redacted]
- [Lifebuoy] [Redacted]
- [Question Mark] [Redacted]
- [Printer] [Redacted]
- WOP [Redacted]

Featured

 Canon IR5255 ADD PRINTER	<div data-bbox="819 438 1512 738"></div>	
 Canon IR5030 ADD PRINTER	 Canon IR3225 ADD PRINTER	 Canon IR5051 ADD PRINTER
 Canon IR5051 ADD PRINTER	 Canon IR5051 ADD PRINTER	 Canon IR5051 ADD PRINTER
 Canon IR5051 ADD PRINTER	 Canon IR5030 ADD PRINTER	 Canon IR5051 ADD PRINTER
 Canon IR5051 ADD PRINTER		

- Categories**
- Featured
 - Printer Drivers
 - Printers
 - Software - Adobe
 - Software - AntiVirus
 - Software - Apple Quicktime
 - Software - Browser Support
 - Software - Communication
 - Software - Development Tools
 - Software - Dropbox
 - Software - Internet Browsers
 - Software - Java
 - Software - Productivity
 - Software - Scientific Software
 - Software - X11

**Deploying Xerox print drivers on a
per-OS basis via Casper's Self Service**

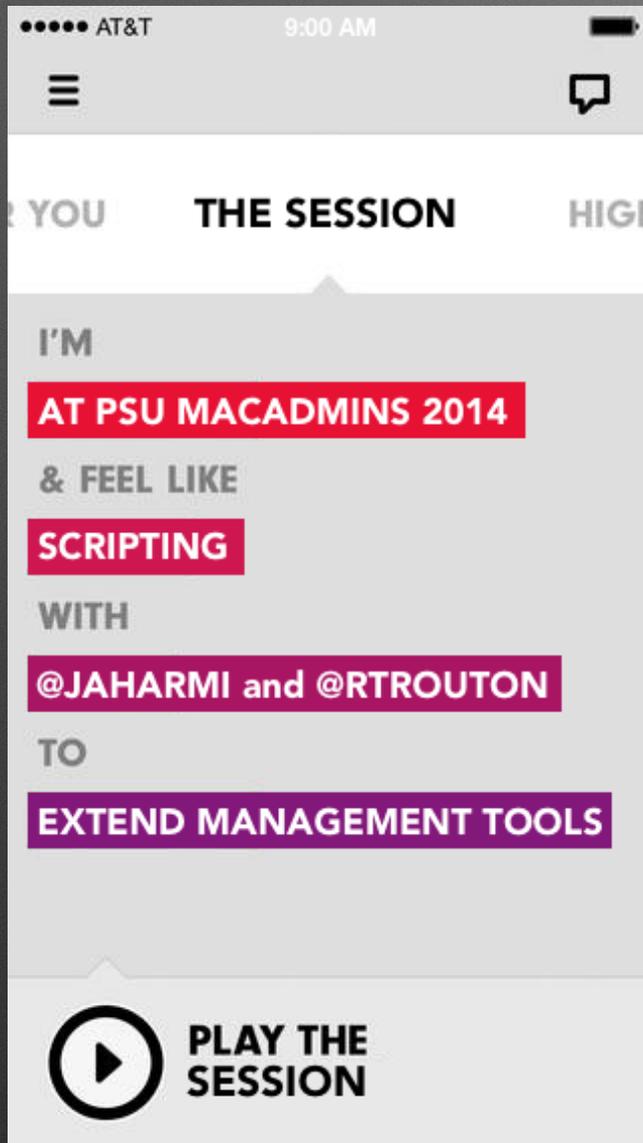
<http://wp.me/pjR60-17S>

“Just spent an hour scripting something and then found out it’s a simple checkbox in the JSS. Welcome to the unnecessary scripting club.”

@JohnKitzmilller

John Kitzmilller

06/13/2014



Q&A

feedback:

<http://j.mp/psumac65>