



JAMFNATION user conference

Get Your Users to do Your Job (Without Them Knowing It)

Get Your Users To Do Your Job

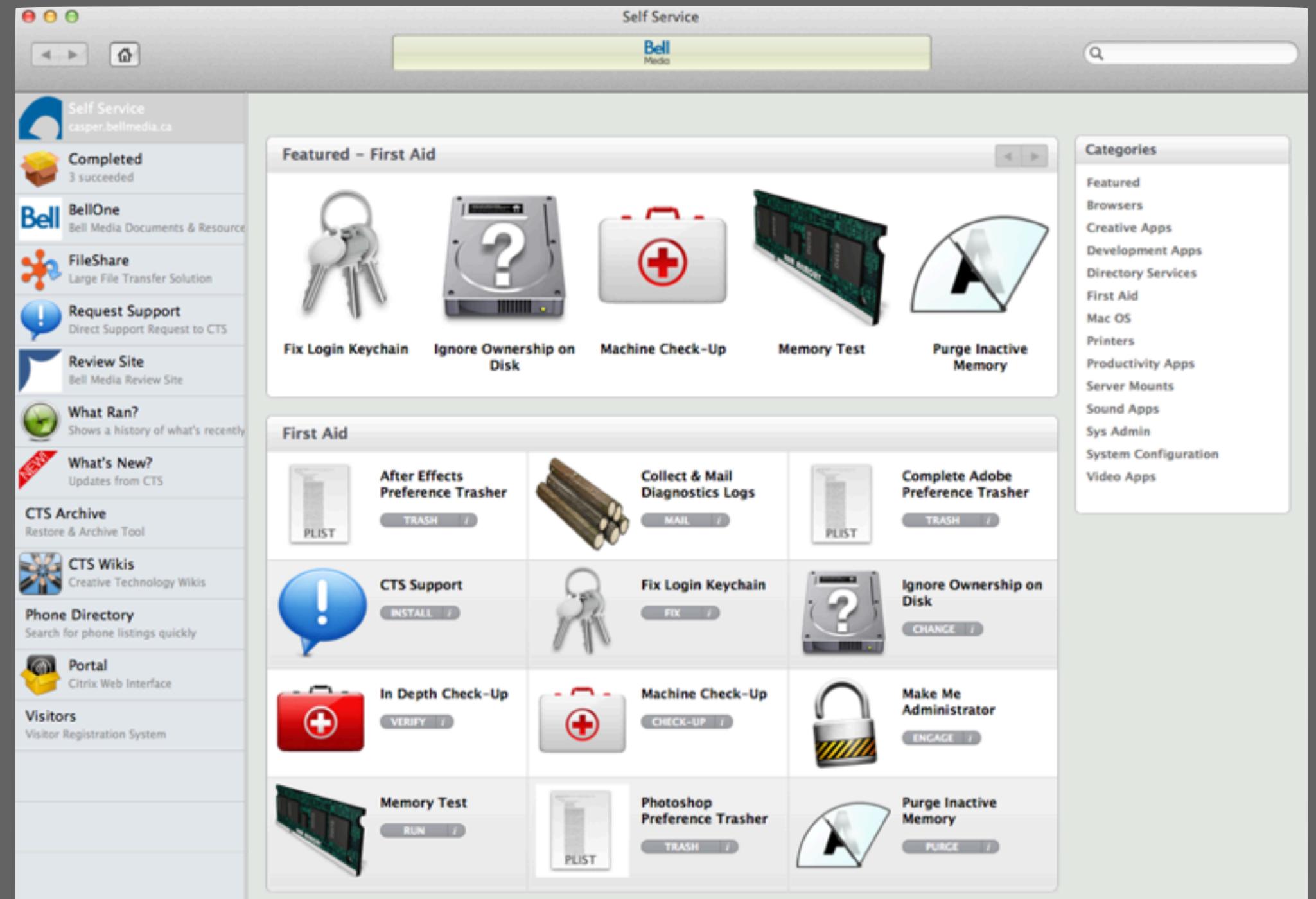
Presentation agenda:

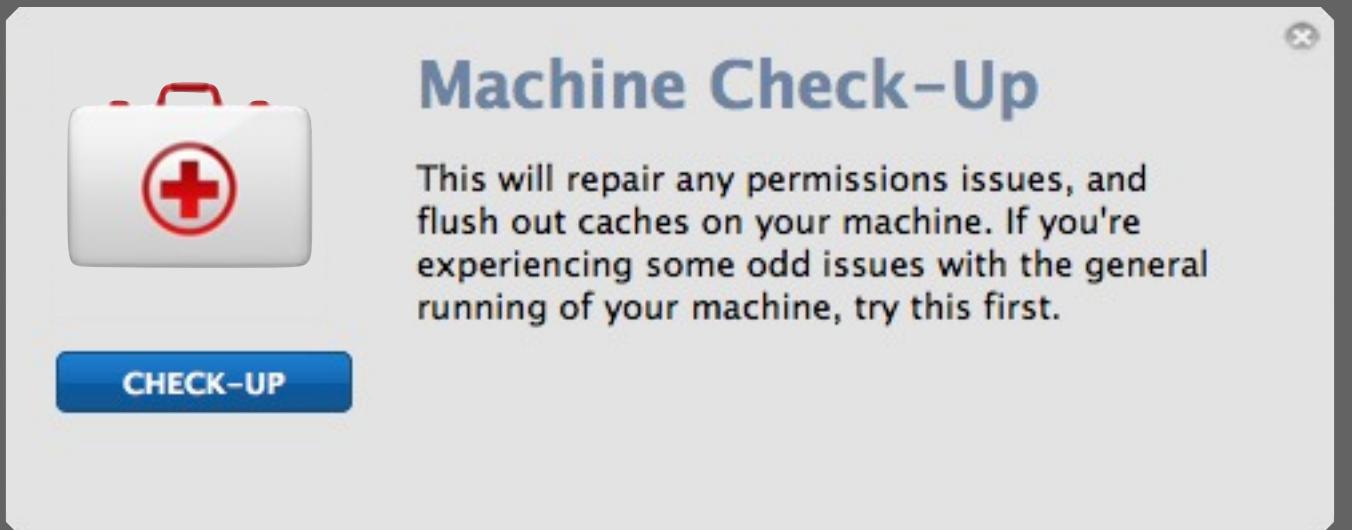
<http://is.gd/jnucUsers>

Empower your Users

Stop getting email about the same thing over and over

Stop the phone from ringing





Edit Policy: Machine Check-Up

General Scope Self Service Packages Scripts Printers Dock Accounts Reboot Advanced

Maintenance

Update Inventory Update Prebindings Flush System Caches
 Reset Computer Names Fix Permissions Flush User Caches
 Self Heal Packages Fix ByHost Files Verify Startup Disk

Files & Processes

Search for file by path: Delete if found
Search for file by name: Update Locate DB
Spotlight Search:
Search for Process: Kill if found
Run Command:

Disk Encryption Configurations

Enabled	Display Name	Recovery Key Type	Encryption Type
<input type="checkbox"/>	Bell Media FileVault	Individual And Institutional	FileVault 2

► Management Framework Options – Not required for clients running 7.3 or later

Cancel Save

Edit Policy: Machine Check-Up

General Scope Self Service Packages Scripts Printers Dock Accounts Reboot Advanced

Self Service Options

Allow this Policy to be used for Self Service

Install Button Text: Check-Up

Force users to view description

Description: This will repair any permissions issues, and flush out caches on your machine. If you're experiencing some odd issues with the general running of your machine, try this first.

Icon: checkup.png Change...

Feature This Policy on the Main Page

Categories:

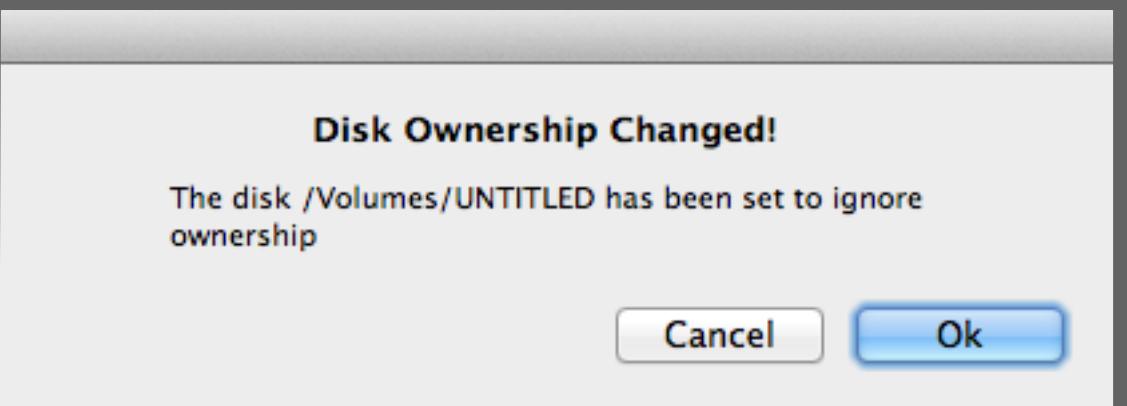
Category	Display	Featured
Browsers	<input type="checkbox"/>	<input type="checkbox"/>
Creative Apps	<input type="checkbox"/>	<input type="checkbox"/>
Development Apps	<input type="checkbox"/>	<input type="checkbox"/>
Directory Services	<input type="checkbox"/>	<input type="checkbox"/>
First Aid	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Imaging	<input type="checkbox"/>	<input type="checkbox"/>
Mac OS	<input type="checkbox"/>	<input type="checkbox"/>
New Machine	<input type="checkbox"/>	<input type="checkbox"/>
Plugins	<input type="checkbox"/>	<input type="checkbox"/>
Printers	<input type="checkbox"/>	<input type="checkbox"/>
Productivity Apps	<input type="checkbox"/>	<input type="checkbox"/>
Server Mounts	<input type="checkbox"/>	<input type="checkbox"/>
Shared Apps	<input type="checkbox"/>	<input type="checkbox"/>

Machine Checkup

- Easy to Set up
- Tools already in your JSS
- Instant Superstar Status







Edit Policy: Ignore Ownership on Disk

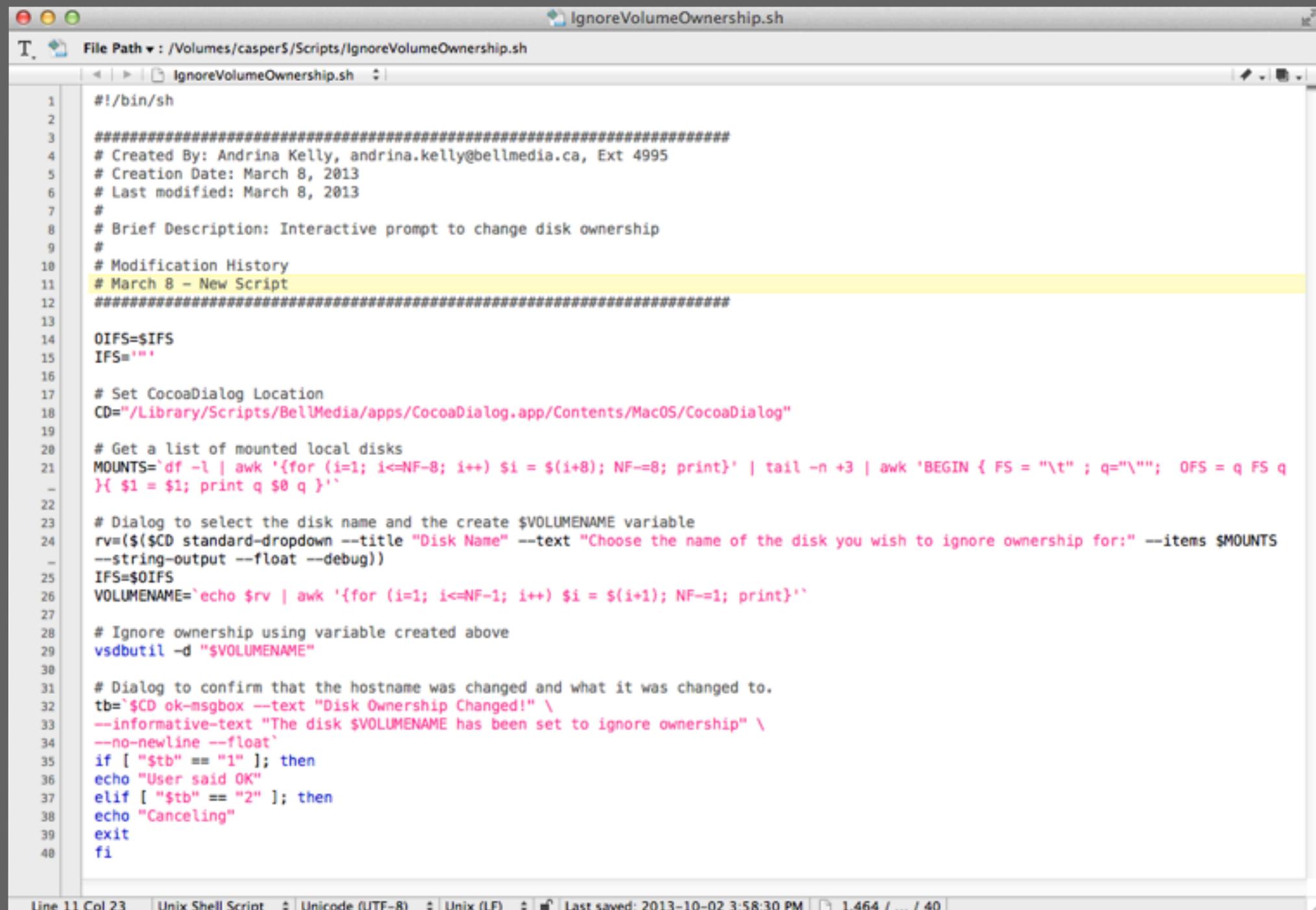
General Scope Self Service Packages Scripts Printers Dock Accounts Reboot Advanced

Scripts

Script Name	Priority	Remove
IgnoreVolumeOwnership.sh	Before	Remove

IgnoreVolumeOwnership.sh / \$computerName \$userName

Add Script Cancel Save



```
IgnoreVolumeOwnership.sh
File Path : /Volumes/casper$/Scripts/IgnoreVolumeOwnership.sh
IgnoreVolumeOwnership.sh

1 #!/bin/sh
2 #####
3 # Created By: Andrina Kelly, andrina.kelly@bellmedia.ca, Ext 4995
4 # Creation Date: March 8, 2013
5 # Last modified: March 8, 2013
6 #
7 # Brief Description: Interactive prompt to change disk ownership
8 #
9 # Modification History
10 # March 8 - New Script
11 #####
12
13
14 OIFS=$IFS
15 IFS="`"
16
17 # Set CocoaDialog Location
18 CD="/Library/Scripts/BellMedia/apps/CocoaDialog.app/Contents/MacOS/CocoaDialog"
19
20 # Get a list of mounted local disks
21 MOUNTS='df -l | awk '{for (i=1; i<NF-8; i++) $i = $(i+8); NF-=8; print}' | tail -n +3 | awk 'BEGIN { FS = "\t" ; q="""; OFS = q FS q }{ $1 = $1; print q $0 q }'
22
23 # Dialog to select the disk name and the create $VOLUMENAME variable
24 rv=$(($CD standard-dropdown --title "Disk Name" --text "Choose the name of the disk you wish to ignore ownership for:" --items $MOUNTS
--string-output --float --debug)
25 IFS=$OIFS
26 VOLUMENAME=`echo $rv | awk '{for (i=1; i<NF-1; i++) $i = $(i+1); NF-=1; print}'`
27
28 # Ignore ownership using variable created above
29 vsdbutil -d "$VOLUMENAME"
30
31 # Dialog to confirm that the hostname was changed and what it was changed to.
32 tb="$CD ok-msgbox --text "Disk Ownership Changed!" \
--informative-text "The disk $VOLUMENAME has been set to ignore ownership" \
--no-newline --float"
33 if [ "$tb" == "1" ]; then
34 echo "User said OK"
35 elif [ "$tb" == "2" ]; then
36 echo "Canceling"
37 exit
38 fi

Line 11 Col 23 Unix Shell Script Unicode (UTF-8) Unix (LF) Last saved: 2013-10-02 3:58:30 PM 1,464 / ... / 40
```

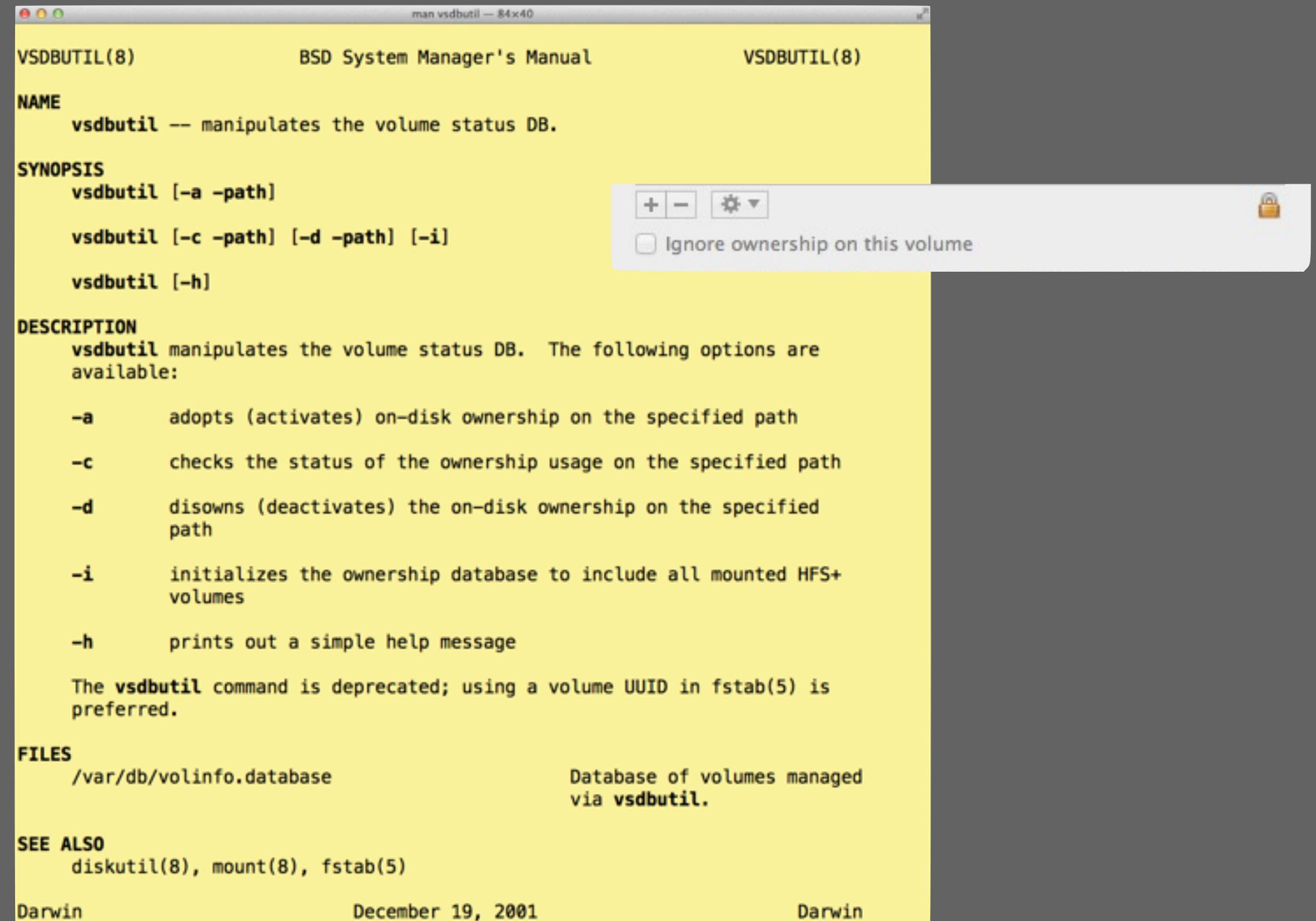
```
OIFS=$IFS  
IFS=' ''  
  
# Set CocoaDialog Location  
CD="/Library/Scripts/BellMedia/apps/CocoaDialog.app/Contents/  
MacOS/CocoaDialog"
```

```
# Get a list of mounted local disks
MOUNTS=`df -l | awk '{for (i=1; i<=NF-8; i++) $i = $(i+8);
NF-=8; print}' | tail -n +3 | awk 'BEGIN { FS = "\t" ; q="\\""; 
OFS = q FS q }{ $1 = $1; print q $0 q }'`
```

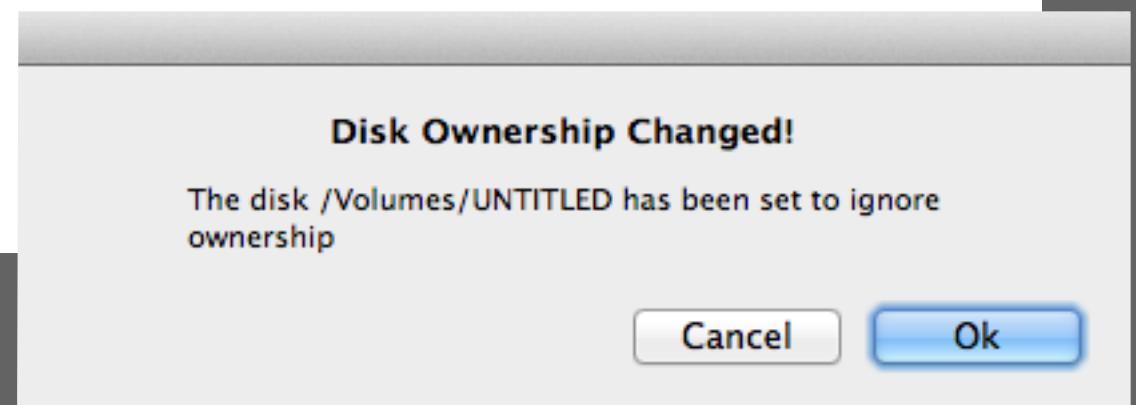
```
# Dialog to select the disk name and the create $VOLUMENAME  
variable  
rv=($($CD standard-dropdown --title "Disk Name" --text "Choose  
the name of the disk you wish to ignore ownership for:"  
--items $MOUNTS --string-output --float --debug))  
  
IFS=$0IFS  
VOLUMENAME=`echo $rv | awk '{for (i=1; i<=NF-1; i++) $i =  
$(i+1); NF-=1; print}'`
```



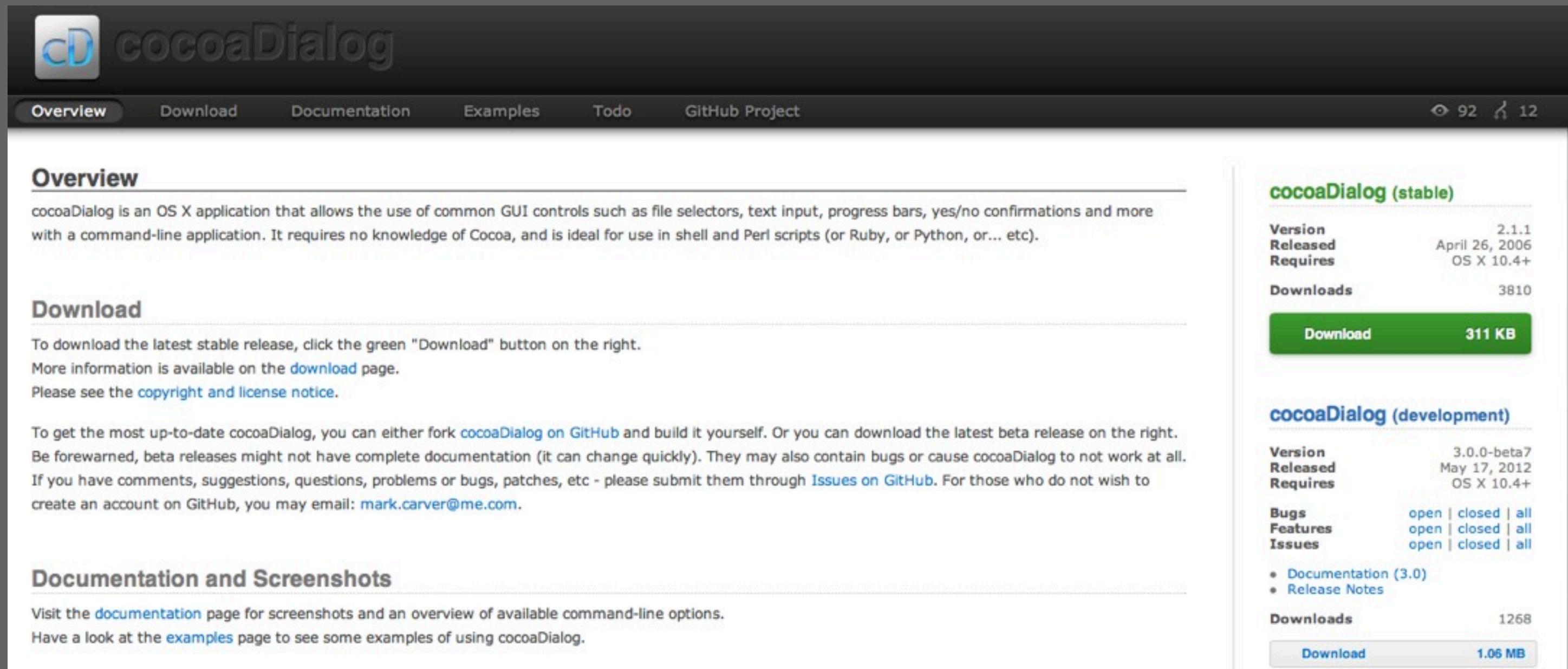
```
# Ignore ownership using variable created above  
vsdbutil -d "$VOLUMENAME"
```



```
# Dialog to confirm that the hostname was changed and what it  
# was changed to.  
tb=`$CD ok-msgbox --text "Disk Ownership Changed!" \  
--informative-text "The disk $VOLUMENAME has been set to  
ignore ownership" \  
--no-newline --float`  
if [ "$tb" == "1" ]; then  
echo "User said OK"  
elif [ "$tb" == "2" ]; then  
echo "Canceling"  
exit  
fi
```



Wait, what the CocoaDialog?



cocoaDialog

Overview Download Documentation Examples Todo GitHub Project 92 12

Overview

cocoaDialog is an OS X application that allows the use of common GUI controls such as file selectors, text input, progress bars, yes/no confirmations and more with a command-line application. It requires no knowledge of Cocoa, and is ideal for use in shell and Perl scripts (or Ruby, or Python, or... etc).

Download

To download the latest stable release, click the green "Download" button on the right.

More information is available on the [download](#) page.

Please see the [copyright and license notice](#).

To get the most up-to-date cocoaDialog, you can either fork [cocoaDialog on GitHub](#) and build it yourself. Or you can download the latest beta release on the right. Be forewarned, beta releases might not have complete documentation (it can change quickly). They may also contain bugs or cause cocoaDialog to not work at all. If you have comments, suggestions, questions, problems or bugs, patches, etc - please submit them through [Issues on GitHub](#). For those who do not wish to create an account on GitHub, you may email: mark.carver@me.com.

Documentation and Screenshots

Visit the [documentation](#) page for screenshots and an overview of available command-line options.

Have a look at the [examples](#) page to see some examples of using cocoaDialog.

cocoaDialog (stable)

Version	2.1.1
Released	April 26, 2006
Requires	OS X 10.4+
Downloads	3810
Download	311 KB

cocoaDialog (development)

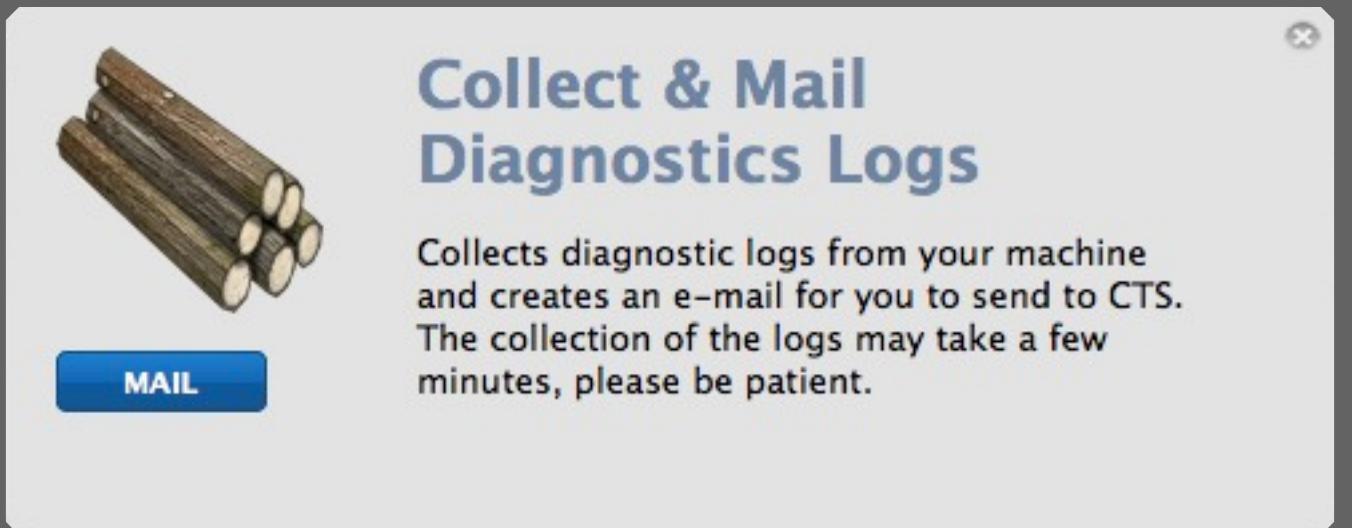
Version	3.0.0-beta7
Released	May 17, 2012
Requires	OS X 10.4+
Bugs	open closed all
Features	open closed all
Issues	open closed all
• Documentation (3.0)	
• Release Notes	
Downloads	1268
Download	1.06 MB

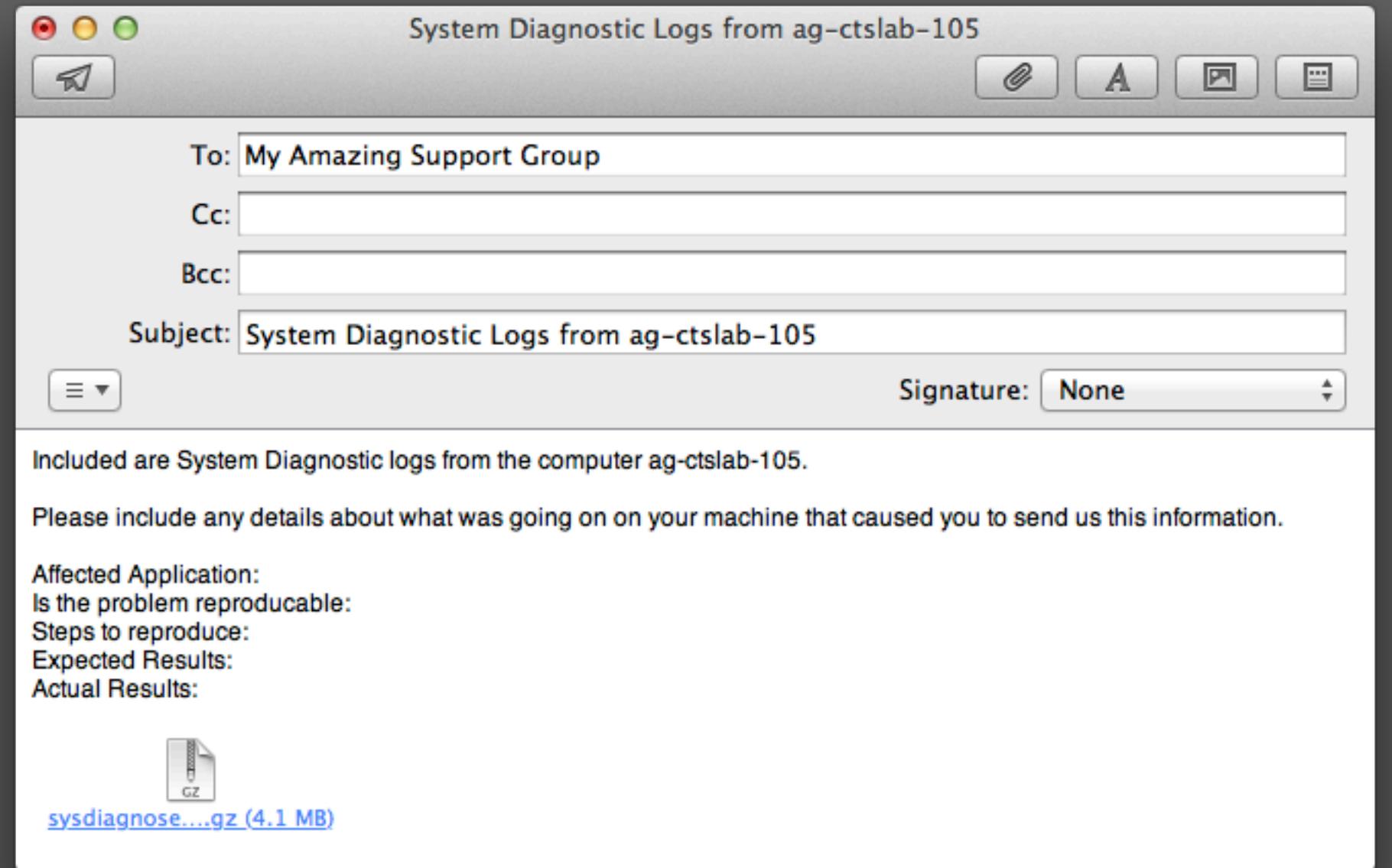
Wait, what the CocoaDialog?

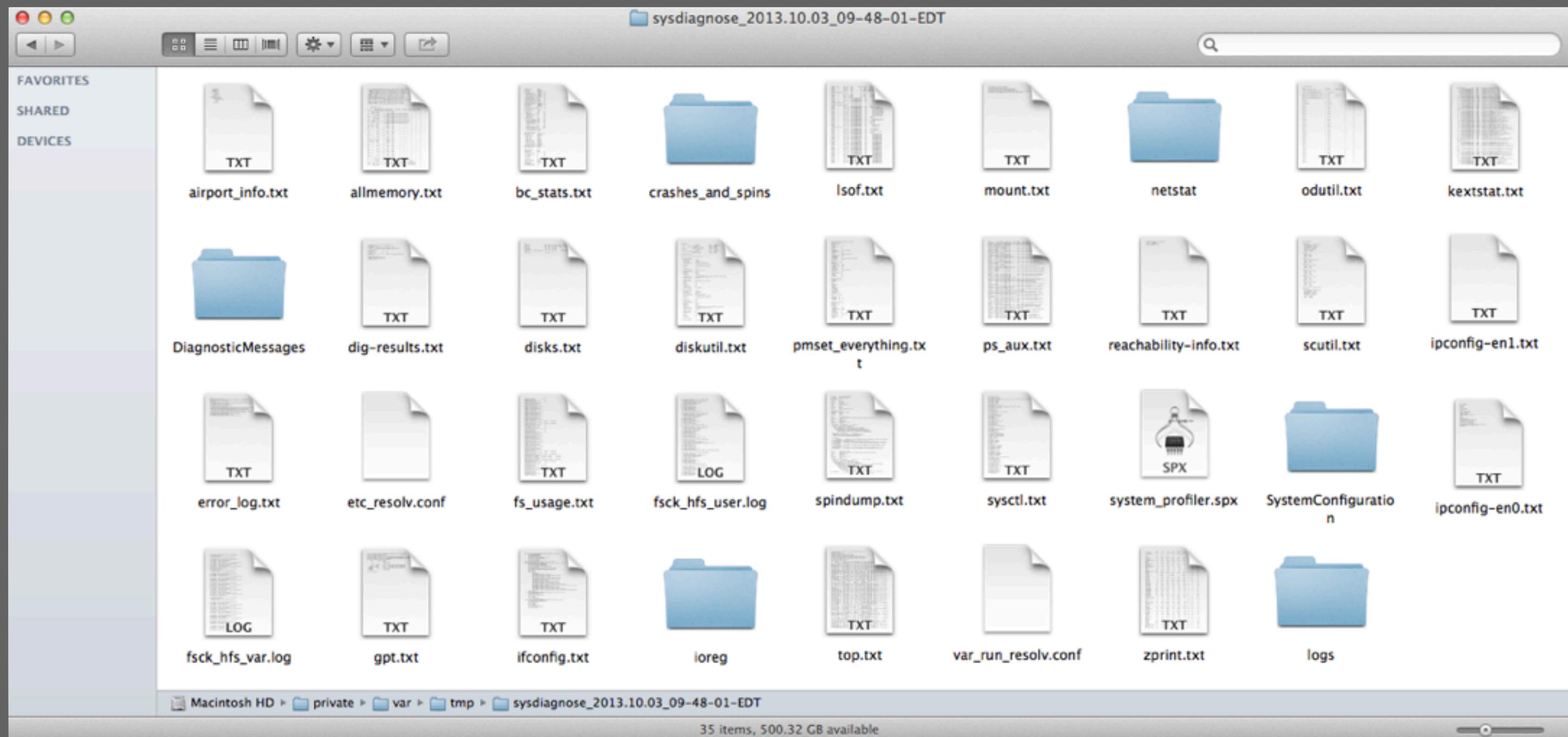
<http://is.gd/cocoadialog>

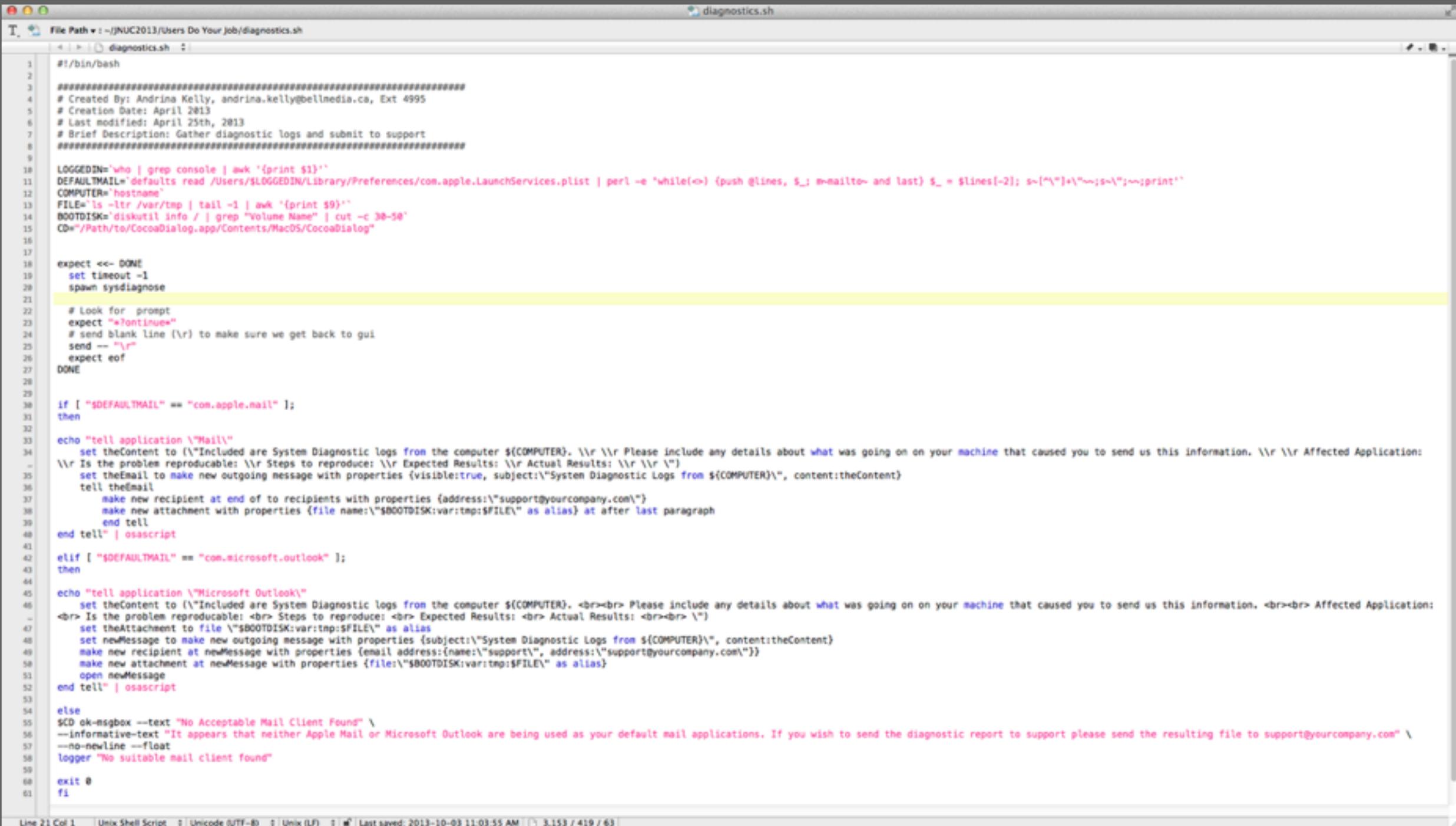
Ignore Ownership on Disk

- Solves a problem normally only an admin could deal with
- Looks like a GUI tool to your users
- Major Scripting Cred!







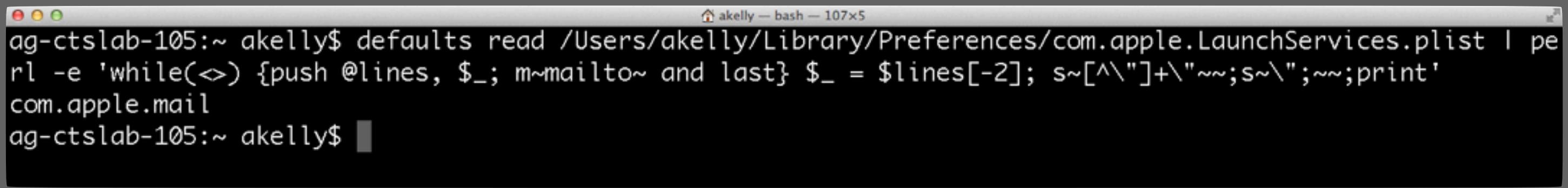


```
File Path : ~/JNUC2013/Users Do Your Job/diagnostics.sh
diagnostics.sh

1 #!/bin/bash
2
3 ######
4 # Created By: Andrina Kelly, andrina.kelly@bellmedia.ca, Ext 4995
5 # Creation Date: April 2013
6 # Last modified: April 25th, 2013
7 # Brief Description: Gather diagnostic logs and submit to support
8 #####
9
10 LOGGEDIN=`who | grep console | awk '{print $1}'`
11 DEFAULTMAIL=`defaults read /Users/$LOGGEDIN/Library/Preferences/com.apple.LaunchServices.plist | perl -e "while(<>) {push @lines, $_; if($lines[-1] =~ m~mailto~ and last) {$_ = $lines[-2]; s~/^~~/~~;s~/~~/~~;print"}}
12 COMPUTER=`hostname`
13 FILE=`ls -ltr /var/tmp | tail -1 | awk '{print $9}'`
14 BOOTDISK=`diskutil info / | grep "Volume Name" | cut -c 38-54`
15 CD="/Path/to/CocoaDialog.app/Contents/MacOS/CocoaDialog"
16
17 expect <<- DONE
18     set timeout -1
19     spawn sysdiagnose
20
21     # Look for prompt
22     expect "*?continue?"
23     # send blank line (\r) to make sure we get back to gui
24     send -- "\r"
25     expect eof
26
27 DONE
28
29
30 if [ "$DEFAULTMAIL" == "com.apple.mail" ];
31 then
32
33 echo "tell application \"Mail\""
34     set theContent to ("Included are System Diagnostic logs from the computer ${COMPUTER}. \\\r \\\r Please include any details about what was going on on your machine that caused you to send us this information. \\\r \\\r Affected Application:\\r Is the problem reproducible: \\\r Steps to reproduce: \\\r Expected Results: \\\r Actual Results: \\\r \\\r ")
35     set theEmail to make new outgoing message with properties {visible:true, subject:"System Diagnostic Logs from ${COMPUTER}", content:theContent}
36     tell theEmail
37         make new recipient at end of to recipients with properties {address:"support@yourcompany.com"}
38         make new attachment with properties {file name:"$BOOTDISK:var/tmp:$FILE" as alias} at after last paragraph
39         end tell
40     end tell" | osascript
41
42 elif [ "$DEFAULTMAIL" == "com.microsoft.outlook" ];
43 then
44
45 echo "tell application \"Microsoft Outlook\""
46     set theContent to ("Included are System Diagnostic logs from the computer ${COMPUTER}. <br><br> Please include any details about what was going on on your machine that caused you to send us this information. <br><br> Affected Application:<br> Is the problem reproducible: <br> Steps to reproduce: <br> Expected Results: <br> Actual Results: <br> <br> ")
47     set theAttachment to file "$BOOTDISK:var/tmp:$FILE" as alias
48     set newMessage to make new outgoing message with properties {subject:"System Diagnostic Logs from ${COMPUTER}", content:theContent}
49     make new recipient at newMessage with properties {email address:(name:"support", address:"support@yourcompany.com")}
50     make new attachment at newMessage with properties {file:"$BOOTDISK:var/tmp:$FILE" as alias}
51     open newMessage
52     end tell" | osascript
53
54 else
55     $CD ok-msgbox --text "No Acceptable Mail Client Found" \
56     --informative-text "It appears that neither Apple Mail or Microsoft Outlook are being used as your default mail applications. If you wish to send the diagnostic report to support please send the resulting file to support@yourcompany.com" \
57     --no-newline --float
58     logger "No suitable mail client found"
59
60     exit 0
61 fi
```

```
LOGGEDIN=`who | grep console | awk '{print $1}'`  
DEFAULTMAIL=`defaults read  
/Users/$LOGGEDIN/Library/Preferences/com.apple.LaunchServices.plist | perl -e  
'while(<>) {push @lines, $_; m~mailto~ and last} $_ = $lines[-2];  
s~[^"]+"~~;s~\";~~;print'  
COMPUTER=`hostname`  
FILE=`ls -ltr /var/tmp | tail -1 | awk '{print $10}'`  
BOOTDISK=`diskutil info / | grep "Volume Name" | cut -c 30-50`  
CD="/Path/to/CocoaDialog.app/Contents/MacOS/CocoaDialog"
```

```
ag-ctslab-105:~ akelly$ defaults read /Users/akelly/Library/Preferences/com.apple.LaunchServices.plist
{
    LSHandlers = (
        {
            LSHandlerContentTag = cr;
            LSHandlerContentTagClass = "public.filename-extension";
            LSHandlerRoleAll = "citrix.servicerecords";
        },
        {
            LSHandlerRoleAll = "com.jamfsoftware.selfservice";
            LSHandlerURLScheme = selfservice;
        },
        {
            LSHandlerRoleAll = "com.apple.mail";
            LSHandlerURLScheme = mailto;
        }
    );
}
ag-ctslab-105:~ akelly$
```



```
ag-ctslab-105:~ akelly$ defaults read /Users/akelly/Library/Preferences/com.apple.LaunchServices.plist | perl -e 'while(<>) {push @lines, $_; m~mailto~ and last} $_ = $lines[-2]; s~[^"]+\"~~;s~\";~~;print' com.apple.mail
ag-ctslab-105:~ akelly$
```

```
ag-ctslab-105:~ akelly$ ls -ltr /var/tmp
total 16296
-rw-r--r--  1 root    wheel
drwx-----  3 root    wheel
srw-rw-rw-  1 root    wheel
drwxrwxrwx  7 root    wheel
drwxr-xr-x  4 akelly  wheel
drwxr-xr-x 13 root    wheel
-rw-r--r--  1 root    wheel
drwxr-xr-x 38 akelly  BBIDM\Domain Users

akelly — bash — 107x16
4093923  2 Oct 17:08 sysdiagnose_2013.10.02_17-04-23-EDT.tar.gz
102     2 Oct 17:34 launchd
        0  2 Oct 17:34 com.barebones.authd.socket
238     3 Oct 09:17 com.symantec.symipc
136     3 Oct 09:17 com.citrix.sf_ipc
442     3 Oct 09:47 sysdiagnose_2013.10.03_09-47-01-EDT
4244415  3 Oct 09:51 sysdiagnose_2013.10.03_09-48-01-EDT.tar.gz
1292    3 Oct 10:06 sysdiagnose_2013.10.03_09-48-01-EDT
```

```
ag-ctslab-105:~ akelly$ ls -ltr /var/tmp
total 16296
-rw-r--r--  1 root    wheel
drwx-----  3 root    wheel
srw-rw-rw-  1 root    wheel
drwxrwxrwx  7 root    wheel
drwxr-xr-x  4 akelly   wheel
drwxr-xr-x  13 root   wheel
-rw-r--r--  1 root    wheel
drwxr-xr-x  38 akelly  BBIDM\Domain Users
ag-ctslab-105:~ akelly$ ls -ltr /var/tmp | tail -1
drwxr-xr-x  38 akelly  BBIDM\Domain Users
akelly — bash — 107x16
```

4093923 2 Oct 17:08 sysdiagnose_2013.10.02_17-04-23-EDT.tar.gz
102 2 Oct 17:34 launchd
0 2 Oct 17:34 com.barebones.authd.socket
238 3 Oct 09:17 com.symantec.symipc
136 3 Oct 09:17 com.citrix.sf_ipc
442 3 Oct 09:47 sysdiagnose_2013.10.03_09-47-01-EDT
4244415 3 Oct 09:51 sysdiagnose_2013.10.03_09-48-01-EDT.tar.gz
1292 3 Oct 10:06 sysdiagnose_2013.10.03_09-48-01-EDT
1292 3 Oct 10:06 sysdiagnose_2013.10.03_09-48-01-EDT

```
ag-ctslab-105:~ akelly$ ls -ltr /var/tmp
total 16296
-rw-r--r-- 1 root wheel 4093923 2 Oct 17:08 sysdiagnose_2013.10.02_17-04-23-EDT.tar.gz
drwx----- 3 root wheel 102 2 Oct 17:34 launchd
srw-rw-rw- 1 root wheel 0 2 Oct 17:34 com.barebones.authd.socket
drwxrwxrwx 7 root wheel 238 3 Oct 09:17 com.symantec.symipc
drwxr-xr-x 4 akelly wheel 136 3 Oct 09:17 com.citrix.sf_ipc
drwxr-xr-x 13 root wheel 442 3 Oct 09:47 sysdiagnose_2013.10.03_09-47-01-EDT
-rw-r--r-- 1 root wheel 4244415 3 Oct 09:51 sysdiagnose_2013.10.03_09-48-01-EDT.tar.gz
drwxr-xr-x 38 akelly BBIDM\Domain Users 1292 3 Oct 10:06 sysdiagnose_2013.10.03_09-48-01-EDT
ag-ctslab-105:~ akelly$ ls -ltr /var/tmp | tail -1
drwxr-xr-x 38 akelly BBIDM\Domain Users 1292 3 Oct 10:06 sysdiagnose_2013.10.03_09-48-01-EDT
ag-ctslab-105:~ akelly$ ls -ltr /var/tmp | tail -1 | awk '{print $10}'
sysdiagnose_2013.10.03_09-48-01-EDT
ag-ctslab-105:~ akelly$
```

```
expect <<- DONE
set timeout -1
spawn sysdiagnose

# Look for prompt
expect "*?ontinue*"
# send blank line (\r) to make sure we get back to gui
send -- "\r"
expect eof
DONE
```

```
if [ "$DEFAULTMAIL" == "com.apple.mail" ];  
then  
  
echo "tell application \"Mail\"\n    set theContent to (\"Included are System Diagnostic logs from the computer  
${COMPUTER}. \\r \\r Please include any details about what was going on on your  
machine that caused you to send us this information. \\r \\r Affected  
Application: \\r Is the problem reproducible: \\r Steps to reproduce: \\r  
Expected Results: \\r Actual Results: \\r \\r \")  
    set theEmail to make new outgoing message with properties {visible:true,  
subject:\"System Diagnostic Logs from ${COMPUTER}\", content:theContent}  
    tell theEmail  
        make new recipient at end of to recipients with properties  
{address:\"support@yourcompany.com\"}  
        make new attachment with properties {file  
name:\"$BOOTDISK:var:tmp:$FILE\" as alias} at after last paragraph  
    end tell  
end tell" | osascript
```

```
elif [ "$DEFAULTMAIL" == "com.microsoft.outlook" ];  
then  
  
echo "tell application \"Microsoft Outlook\"  
    set theContent to (\"Included are System Diagnostic logs from the computer  
${COMPUTER}. <br><br> Please include any details about what was going on on your  
machine that caused you to send us this information. <br><br> Affected  
Application: <br> Is the problem reproducible: <br> Steps to reproduce: <br>  
Expected Results: <br> Actual Results: <br><br> \")  
    set theAttachment to file \"${B00TDISK}:var:tmp:$FILE\" as alias  
    set newMessage to make new outgoing message with properties {subject:\"System  
Diagnostic Logs from ${COMPUTER}\", content:theContent}  
        make new recipient at newMessage with properties {email  
address:{name:\"support\", address:\"support@yourcompany.com\"}}  
        make new attachment at newMessage with properties  
{file:\"${B00TDISK}:var:tmp:$FILE\" as alias}  
        open newMessage  
end tell" | osascript
```

```
else
$CD ok-msgbox --text "No Acceptable Mail Client Found" \
--informative-text "It appears that neither Apple Mail or Microsoft Outlook are
being used as your default mail applications. If you wish to send the diagnostic
report to support please send the resulting file to support@yourcompany.com" \
--no-newline --float
logger "No suitable mail client found"

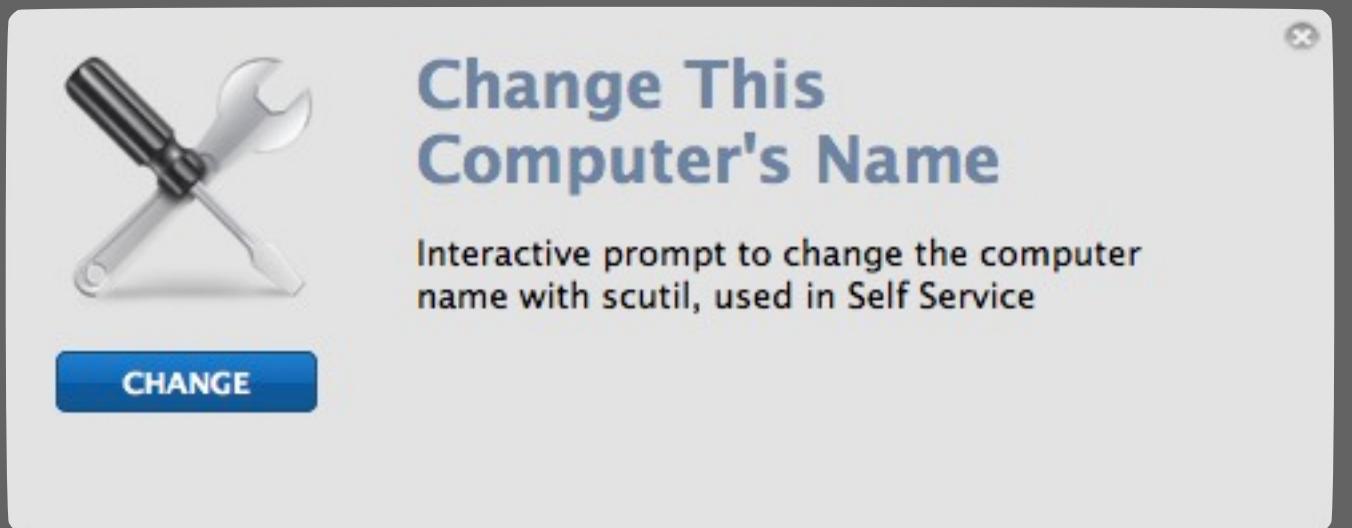
exit 0
fi
```

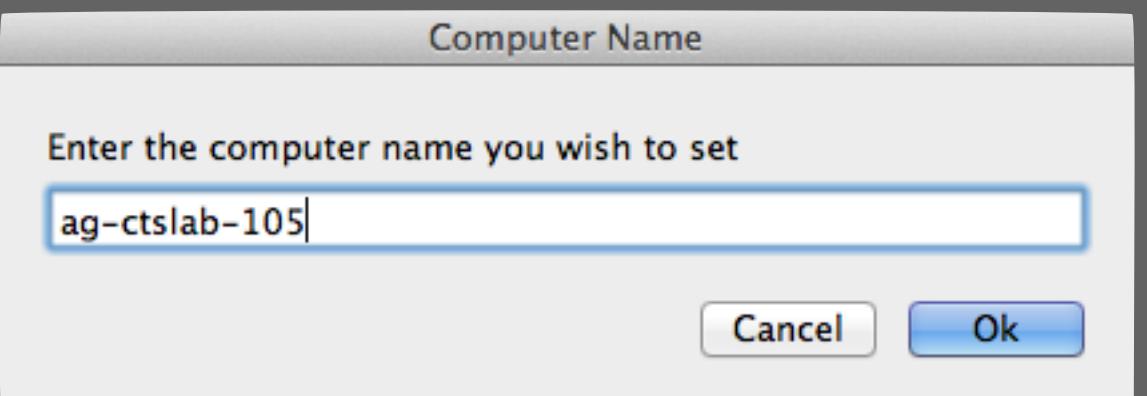
Collect & Mail Diagnostic Logs

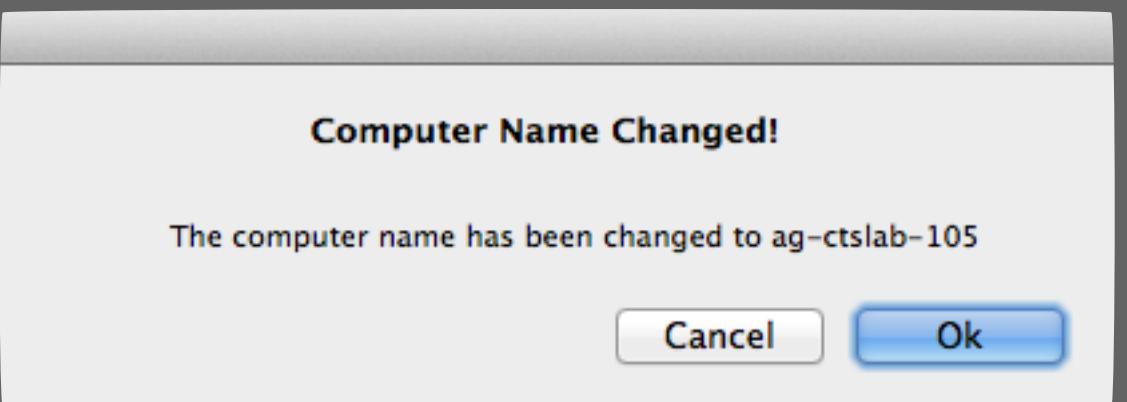
- Encourages useful information from users
- Wow, that's 4 different scripting languages in 1 script!
- Makes it easy to answer the question “What do the logs say?”

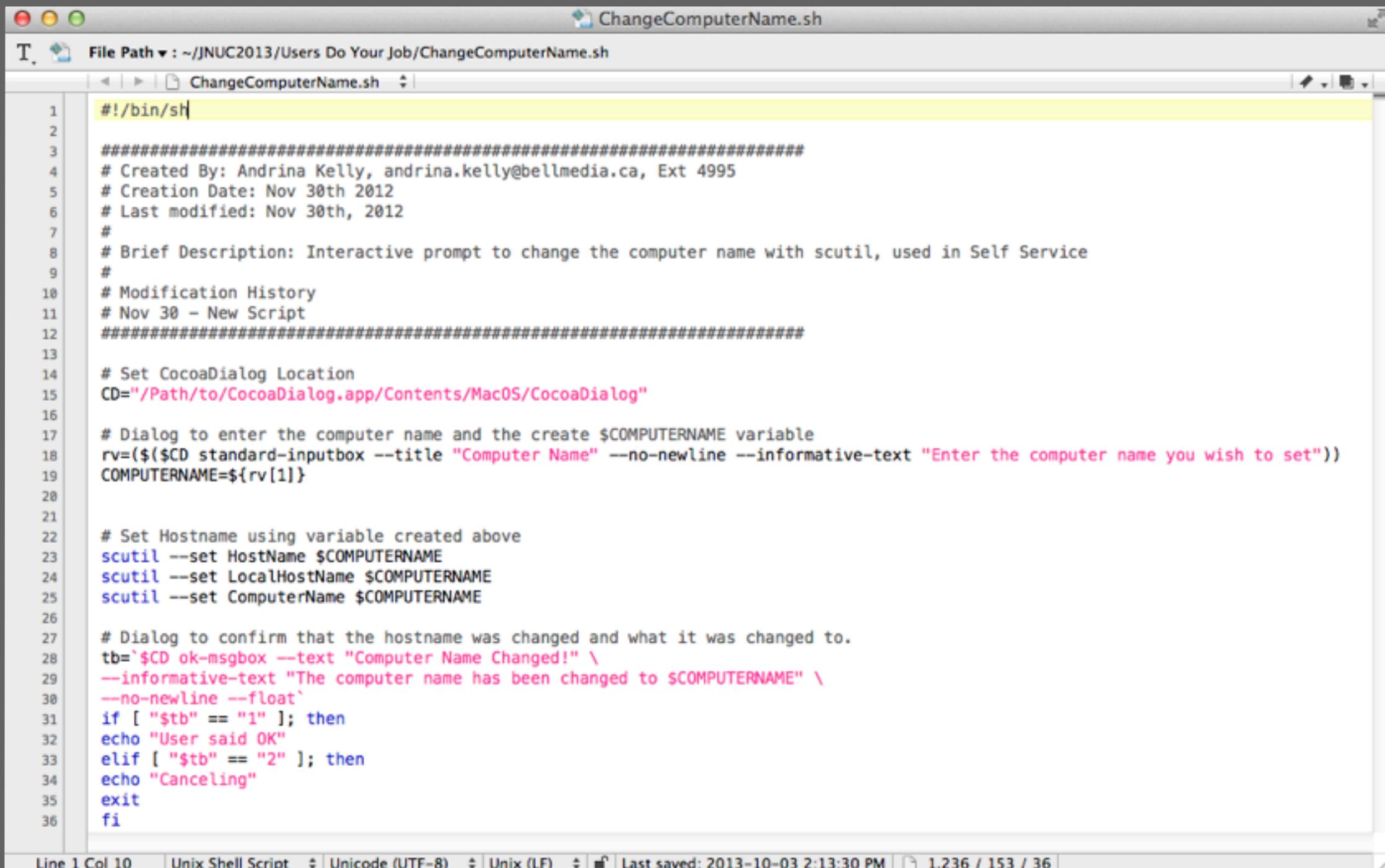
Change This Computer's Name

- No more CLI for support technicians
- Easy tool for the support toolbox









```
#!/bin/sh
#####
# Created By: Andrina Kelly, andrina.kelly@bellmedia.ca, Ext 4995
# Creation Date: Nov 30th 2012
# Last modified: Nov 30th, 2012
#
# Brief Description: Interactive prompt to change the computer name with scutil, used in Self Service
#
# Modification History
# Nov 30 - New Script
#####

# Set CocoaDialog Location
CD="/Path/to/CocoaDialog.app/Contents/MacOS/CocoaDialog"

# Dialog to enter the computer name and the create $COMPUTERNAME variable
rv=(${CD} standard-inputbox --title "Computer Name" --no-newline --informative-text "Enter the computer name you wish to set")
COMPUTERNAME=${rv[1]}

# Set Hostname using variable created above
scutil --set HostName $COMPUTERNAME
scutil --set LocalHostName $COMPUTERNAME
scutil --set ComputerName $COMPUTERNAME

# Dialog to confirm that the hostname was changed and what it was changed to.
tb='${CD} ok-msgbox --text "Computer Name Changed!" \
--informative-text "The computer name has been changed to $COMPUTERNAME" \
--no-newline --float'
if [ "$tb" == "1" ]; then
echo "User said OK"
elif [ "$tb" == "2" ]; then
echo "Canceling"
exit
fi
```

Line 1 Col 10 Unix Shell Script # Unicode (UTF-8) # Unix (LF) # Last saved: 2013-10-03 2:13:30 PM 1,236 / 153 / 36

```
# Dialog to enter the computer name and the create  
$COMPUTERNAME variable  
rv=($($CD standard-inputbox --title "Computer Name"  
--no-newline --informative-text "Enter the computer name you  
wish to set"))  
COMPUTERNAME=${rv[1]}
```

```
# Set Hostname using variable created above
scutil --set HostName $COMPUTERNAME
scutil --set LocalHostName $COMPUTERNAME
scutil --set ComputerName $COMPUTERNAME
```

```
# Dialog to confirm that the hostname was changed and what it  
# was changed to.  
tb=`$CD ok-msgbox --text "Computer Name Changed!" \  
--informative-text "The computer name has been changed to  
$COMPUTERNAME" \  
--no-newline --float`  
if [ "$tb" == "1" ]; then  
echo "User said OK"  
elif [ "$tb" == "2" ]; then  
echo "Canceling"  
exit  
fi
```

```
# Dialog to confirm that the hostname was changed and what it  
# was changed to.  
tb=`$CD ok-msgbox --text "Computer Name Changed!" \  
--informative-text "The computer name has been changed to  
$COMPUTERNAME" \  
--no-newline --float`  
if [ "$tb" == "1" ]; then  
echo "User said OK"  
elif [ "$tb" == "2" ]; then  
echo "Canceling"  
exit  
fi
```

```
# Dialog to confirm that the hostname was changed and what it  
# was changed to.  
tb=`$CD ok-msgbox --text "Computer Name Changed!" \  
--informative-text "The computer name has been changed to  
$COMPUTERNAME" \  
--no-newline --float`  
if [ "$tb" == "1" ]; then  
echo "User said OK"  
elif [ "$tb" == "2" ]; then  
echo "Canceling"  
exit  
fi
```

```
# Dialog to confirm that the hostname was changed and what it  
# was changed to.  
tb=`$CD ok-msgbox --text "Computer Name Changed!" \  
--informative-text "The computer name has been changed to  
$COMPUTERNAME" \  
--no-newline --float`  
if [ "$tb" == "1" ]; then  
echo "User said OK"  
elif [ "$tb" == "2" ]; then  
echo "Canceling"  
exit  
fi
```

```
# Dialog to confirm that the hostname was changed and what it  
# was changed to.  
tb=`$CD ok-msgbox --text "Computer Name Changed!" \  
--informative-text "The computer name has been changed to  
$COMPUTERNAME" \  
--no-newline --float`  
if [ "$tb" == "1" ]; then  
echo "User said OK"  
elif [ "$tb" == "2" ]; then  
echo "Canceling"  
exit  
fi
```

```
# Dialog to confirm that the hostname was changed and what it  
# was changed to.  
tb=`$CD ok-msgbox --text "Computer Name Changed!" \  
--informative-text "The computer name has been changed to  
$COMPUTERNAME" \  
--no-newline --float`  
if [ "$tb" == "1" ]; then  
echo "User said OK"  
elif [ "$tb" == "2" ]; then  
echo "Canceling"  
exit  
fi
```

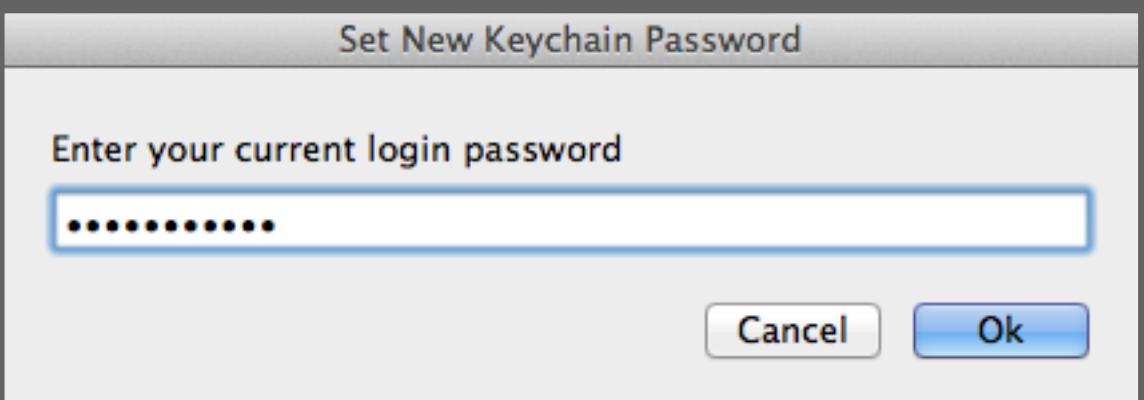
```
# Dialog to confirm that the hostname was changed and what it  
# was changed to.  
tb=`$CD ok-msgbox --text "Computer Name Changed!" \  
--informative-text "The computer name has been changed to  
$COMPUTERNAME" \  
--no-newline --float`  
if [ "$tb" == "1" ]; then  
echo "User said OK"  
elif [ "$tb" == "2" ]; then  
echo "Canceling"  
exit  
fi
```

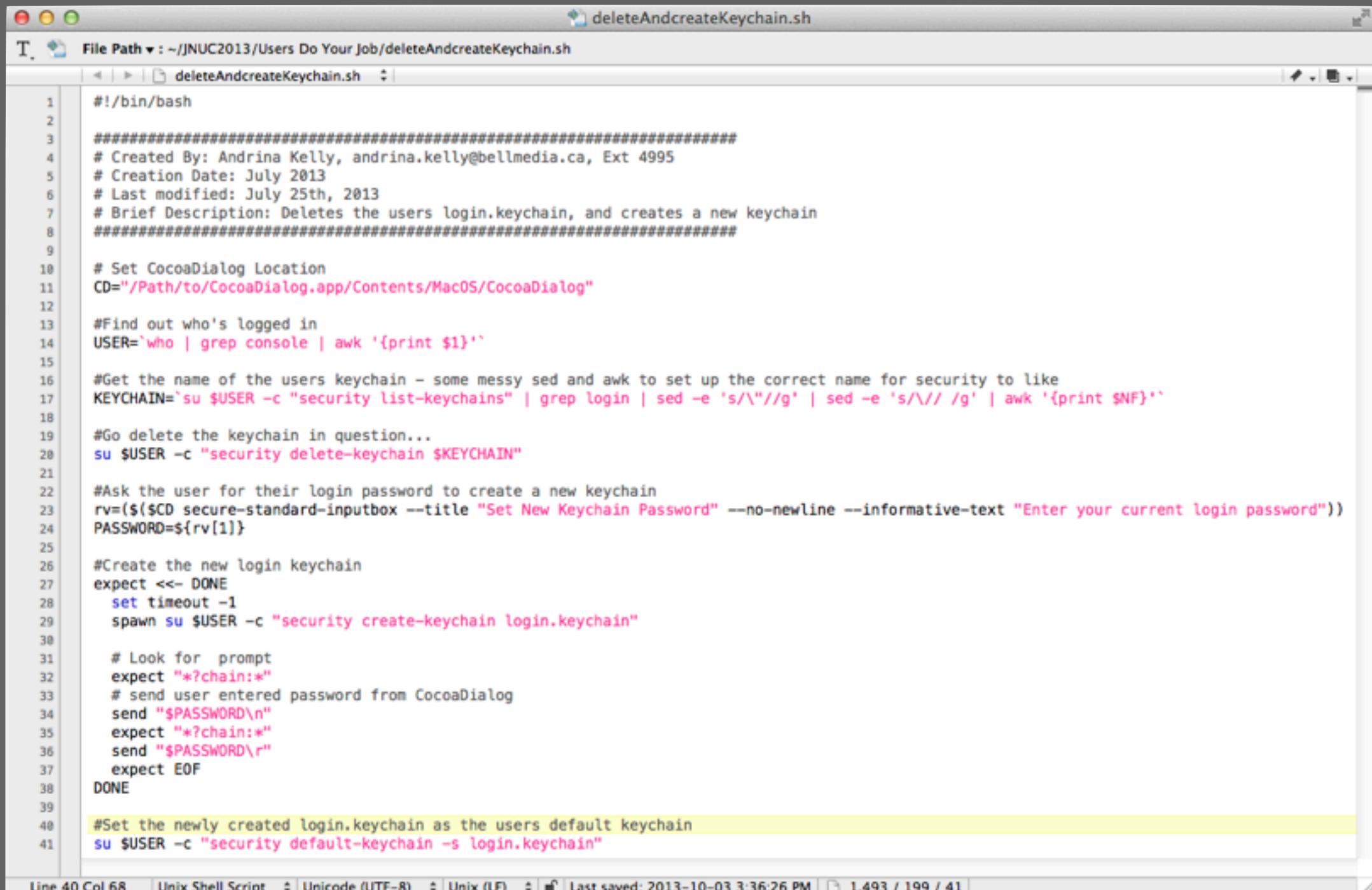
```
# Dialog to confirm that the hostname was changed and what it  
# was changed to.  
tb=`$CD ok-msgbox --text "Computer Name Changed!" \  
--informative-text "The computer name has been changed to  
$COMPUTERNAME" \  
--no-newline --float`  
if [ "$tb" == "1" ]; then  
echo "User said OK"  
elif [ "$tb" == "2" ]; then  
echo "Canceling"  
exit  
fi
```

```
# Dialog to confirm that the hostname was changed and what it  
# was changed to.  
tb=`$CD ok-msgbox --text "Computer Name Changed!" \  
--informative-text "The computer name has been changed to  
$COMPUTERNAME" \  
--no-newline --float`  
if [ "$tb" == "1" ]; then  
echo "User said OK"  
elif [ "$tb" == "2" ]; then  
echo "Canceling"  
exit  
fi
```









```
deleteAndcreateKeychain.sh
File Path : ~/JNUC2013/Users Do Your Job/deleteAndcreateKeychain.sh
deleteAndcreateKeychain.sh

1 #!/bin/bash
2
3 ######
4 # Created By: Andrina Kelly, andrina.kelly@bellmedia.ca, Ext 4995
5 # Creation Date: July 2013
6 # Last modified: July 25th, 2013
7 # Brief Description: Deletes the users login.keychain, and creates a new keychain
8 #####
9
10 # Set CocoaDialog Location
11 CD="/Path/to/CocoaDialog.app/Contents/MacOS/CocoaDialog"
12
13 #Find out who's logged in
14 USER=`who | grep console | awk '{print $1}'`
15
16 #Get the name of the users keychain - some messy sed and awk to set up the correct name for security to like
17 KEYCHAIN=`su $USER -c "security list-keychains" | grep login | sed -e 's/\//g' | sed -e 's/\// /g' | awk '{print $NF}'`
18
19 #Go delete the keychain in question...
20 su $USER -c "security delete-keychain $KEYCHAIN"
21
22 #Ask the user for their login password to create a new keychain
23 rv=$(($CD secure-standard-inputbox --title "Set New Keychain Password" --no-newline --informative-text "Enter your current login password"))
24 PASSWORD=${rv[1]}
25
26 #Create the new login keychain
27 expect <<- DONE
28   set timeout -1
29   spawn su $USER -c "security create-keychain login.keychain"
30
31   # Look for prompt
32   expect "*?chain:?"
33   # send user entered password from CocoaDialog
34   send "$PASSWORD\n"
35   expect "*?chain:?"
36   send "$PASSWORD\r"
37   expect EOF
38 DONE
39
40 #Set the newly created login.keychain as the users default keychain
41 su $USER -c "security default-keychain -s login.keychain"
```

```
# Set CocoaDialog Location
CD="/Path/to/CocoaDialog.app/Contents/MacOS/CocoaDialog"

#Find out who's logged in
USER=`who | grep console | awk '{print $1}'`
```

```
#Get the name of the users keychain – some messy sed and awk to set up the correct  
name for security to like  
KEYCHAIN='su $USER -c "security list-keychains" | grep login | sed -e 's/\//g' |  
sed -e 's/\// /g' | awk '{print $NF}''
```

```
ag-ctslab-105:~ akelly$ security list-keychains
"/Users/akelly/Library/Keychains/0C_KeyContainer__andrina.kelly@bellmedia.ca"
"/private/var/folders/xn/5p68knvj6q5dxdclhg2_cmc1pmprs2/T/MDM_Connect-04ECA6BD-04CD-4CDB-8247-148C0B92CFC6.keychain"
"/private/var/folders/zz/zxyvpxvq6csfxvn_n000000000000/T/323066DF-671F-4590-BC52-EA31B51A7B6B-fdesetup/fdesetup.keychain"
"/private/var/folders/zz/zxyvpxvq6csfxvn_n000000000000/T/2E97984A-40B5-460A-9A5C-F52E7356EA4F-fdesetup/fdesetup.keychain"
"/private/var/folders/zz/zxyvpxvq6csfxvn_n000000000000/T/F346C744-C016-4127-B47D-2B0DA85BF3C8-fdesetup/fdesetup.keychain"
"/private/var/folders/zz/zxyvpxvq6csfxvn_n000000000000/T/F53B39BE-0DC5-4AA0-883A-D0F73D58F06B-fdesetup/fdesetup.keychain"
"/Users/akelly/Library/Keychains/login.keychain"
"/Library/Keychains/System.keychain"
```

```
ag-ctslab-105:~ akelly$ security list-keychains
"/Users/akelly/Library/Keychains/0C_KeyContainer__andrina.kelly@bellmedia.ca"
"/private/var/folders/xn/5p68knvj6q5dxdclhg2_cmc1pmprs2/T/MDM_Connect-04ECA6BD-04CD-4CDB-8247-148C0B92CFC6.keychain"
"/private/var/folders/zz/zxyvpxvq6csfxvn_n000000000000/T/323066DF-671F-4590-BC52-EA31B51A7B6B-fdesetup/fdesetup.keychain"
"/private/var/folders/zz/zxyvpxvq6csfxvn_n000000000000/T/2E97984A-40B5-460A-9A5C-F52E7356EA4F-fdesetup/fdesetup.keychain"
"/private/var/folders/zz/zxyvpxvq6csfxvn_n000000000000/T/F346C744-C016-4127-B47D-2B0DA85BF3C8-fdesetup/fdesetup.keychain"
"/private/var/folders/zz/zxyvpxvq6csfxvn_n000000000000/T/F53B39BE-0DC5-4AA0-883A-D0F73D58F06B-fdesetup/fdesetup.keychain"
"/Users/akelly/Library/Keychains/login.keychain"
"/Library/Keychains/System.keychain"
ag-ctslab-105:~ akelly$ security list-keychains | grep login
"/Users/akelly/Library/Keychains/login.keychain"
```

```
ag-ctslab-105:~ akelly$ security list-keychains
"/Users/akelly/Library/Keychains/0C_KeyContainer__andrina.kelly@bellmedia.ca"
"/private/var/folders/xn/5p68knvj6q5dxdclhg2_cmc1pmprs2/T/MDM_Connect-04ECA6BD-04CD-4CDB-8247-148C0B92CFC6.keychain"
"/private/var/folders/zz/zxyvpxvq6csfxvn_n000000000000/T/323066DF-671F-4590-BC52-EA31B51A7B6B-fdesetup/fdesetup.keychain"
"/private/var/folders/zz/zxyvpxvq6csfxvn_n000000000000/T/2E97984A-40B5-460A-9A5C-F52E7356EA4F-fdesetup/fdesetup.keychain"
"/private/var/folders/zz/zxyvpxvq6csfxvn_n000000000000/T/F346C744-C016-4127-B47D-2B0DA85BF3C8-fdesetup/fdesetup.keychain"
"/private/var/folders/zz/zxyvpxvq6csfxvn_n000000000000/T/F53B39BE-0DC5-4AA0-883A-D0F73D58F06B-fdesetup/fdesetup.keychain"
"/Users/akelly/Library/Keychains/login.keychain"
"/Library/Keychains/System.keychain"
ag-ctslab-105:~ akelly$ security list-keychains | grep login
"/Users/akelly/Library/Keychains/login.keychain"
ag-ctslab-105:~ akelly$ security list-keychains | grep login | sed -e 's/\"//g'
/Users/akelly/Library/Keychains/login.keychain
```

```
ag-ctslab-105:~ akelly$ security list-keychains
"/Users/akelly/Library/Keychains/0C_KeyContainer__andrina.kelly@bellmedia.ca"
"/private/var/folders/xn/5p68knvj6q5dxdclhg2_cmc1pmprs2/T/MDM_Connect-04ECA6BD-04CD-4CDB-8247-148C0B92CFC6.keychain"
"/private/var/folders/zz/zxyvpxvq6csfxvn_n000000000000/T/323066DF-671F-4590-BC52-EA31B51A7B6B-fdesetup/fdesetup.keychain"
"/private/var/folders/zz/zxyvpxvq6csfxvn_n000000000000/T/2E97984A-40B5-460A-9A5C-F52E7356EA4F-fdesetup/fdesetup.keychain"
"/private/var/folders/zz/zxyvpxvq6csfxvn_n000000000000/T/F346C744-C016-4127-B47D-2B0DA85BF3C8-fdesetup/fdesetup.keychain"
"/private/var/folders/zz/zxyvpxvq6csfxvn_n000000000000/T/F53B39BE-0DC5-4AA0-883A-D0F73D58F06B-fdesetup/fdesetup.keychain"
"/Users/akelly/Library/Keychains/login.keychain"
"/Library/Keychains/System.keychain"
ag-ctslab-105:~ akelly$ security list-keychains | grep login
"/Users/akelly/Library/Keychains/login.keychain"
ag-ctslab-105:~ akelly$ security list-keychains | grep login | sed -e 's/\"//g'
/Users/akelly/Library/Keychains/login.keychain
ag-ctslab-105:~ akelly$ security list-keychains | grep login | sed -e 's/\"//g' | sed -e 's/\// /g'
Users akelly Library Keychains login.keychain
```

```
ag-ctslab-105:~ akelly$ security list-keychains
"/Users/akelly/Library/Keychains/0C_KeyContainer__andrina.kelly@bellmedia.ca"
"/private/var/folders/xn/5p68knvj6q5dxdclhg2_cmc1pmprs2/T/MDM_Connect-04ECA6BD-04CD-4CDB-8247-148C0B92CFC6.keychain"
"/private/var/folders/zz/zxyvpxvq6csfxvn_n000000000000/T/323066DF-671F-4590-BC52-EA31B51A7B6B-fdesetup/fdesetup.keychain"
"/private/var/folders/zz/zxyvpxvq6csfxvn_n000000000000/T/2E97984A-40B5-460A-9A5C-F52E7356EA4F-fdesetup/fdesetup.keychain"
"/private/var/folders/zz/zxyvpxvq6csfxvn_n000000000000/T/F346C744-C016-4127-B47D-2B0DA85BF3C8-fdesetup/fdesetup.keychain"
"/private/var/folders/zz/zxyvpxvq6csfxvn_n000000000000/T/F53B39BE-0DC5-4AA0-883A-D0F73D58F06B-fdesetup/fdesetup.keychain"
"/Users/akelly/Library/Keychains/login.keychain"
"/Library/Keychains/System.keychain"
ag-ctslab-105:~ akelly$ security list-keychains | grep login
"/Users/akelly/Library/Keychains/login.keychain"
ag-ctslab-105:~ akelly$ security list-keychains | grep login | sed -e 's/"//g'
/Users/akelly/Library/Keychains/login.keychain
ag-ctslab-105:~ akelly$ security list-keychains | grep login | sed -e 's/"//g' | sed -e 's/\// /g'
    Users akelly Library Keychains login.keychain
ag-ctslab-105:~ akelly$ security list-keychains | grep login | sed -e 's/"//g' | sed -e 's/\// /g' | awk '{print $NF}'
login.keychain
ag-ctslab-105:~ akelly$
```

```
#Go delete the keychain in question...
su $USER -c "security delete-keychain $KEYCHAIN"
```

```
#Ask the user for their login password to create a new keychain
rv=($($CD secure-standard-inputbox --title "Set New Keychain Password"
--no-newline --informative-text "Enter your current login password"))
PASSWORD=${rv[1]}
```

```
#Create the new login keychain
expect <<- DONE
    set timeout -1
    spawn su $USER -c "security create-keychain login.keychain"

# Look for prompt
expect "*?chain:*
# send user entered password from CocoaDialog
send "$PASSWORD\n"
expect "*?chain:*
send "$PASSWORD\r"
expect EOF
DONE
```

```
ag-ctslab-105:~ akelly$ security create-keychain jnuc.keychain
password for new keychain:
retype password for new keychain:
```

```
#Set the newly created login.keychain as the users default keychain  
su $USER -c "security default-keychain -s login.keychain"
```

Fix Login Keychain

- Users will change their password in the most inconvenient places...
- Faster than doing the same thing in the GUI

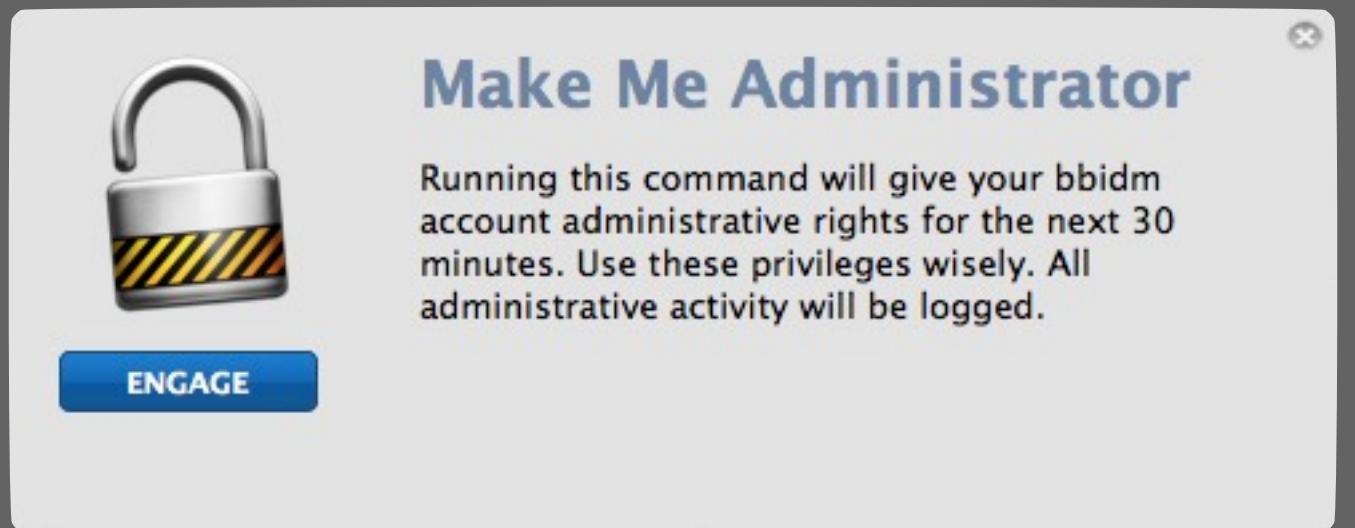






Make Me Admin

- Why would you ever do such a thing!?!?
- Need to know why it was used
- Revert back to regular user after a timeout



Administrative Access Request | Creative Technology Systems

Bell Media

Step 1: Your details

Name First and last name

Email your.name@bellmedia.ca

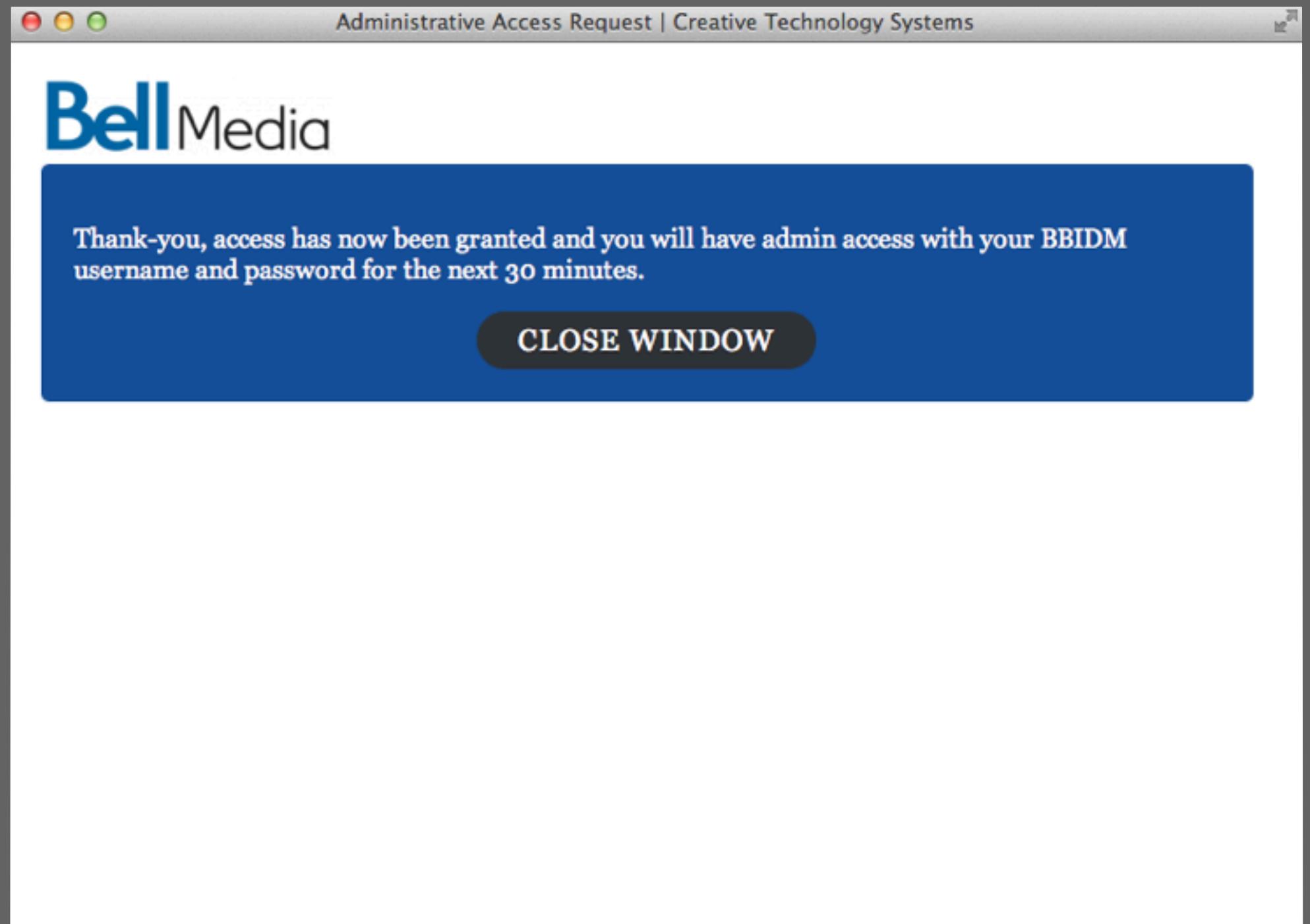
Extension Eg. x6000

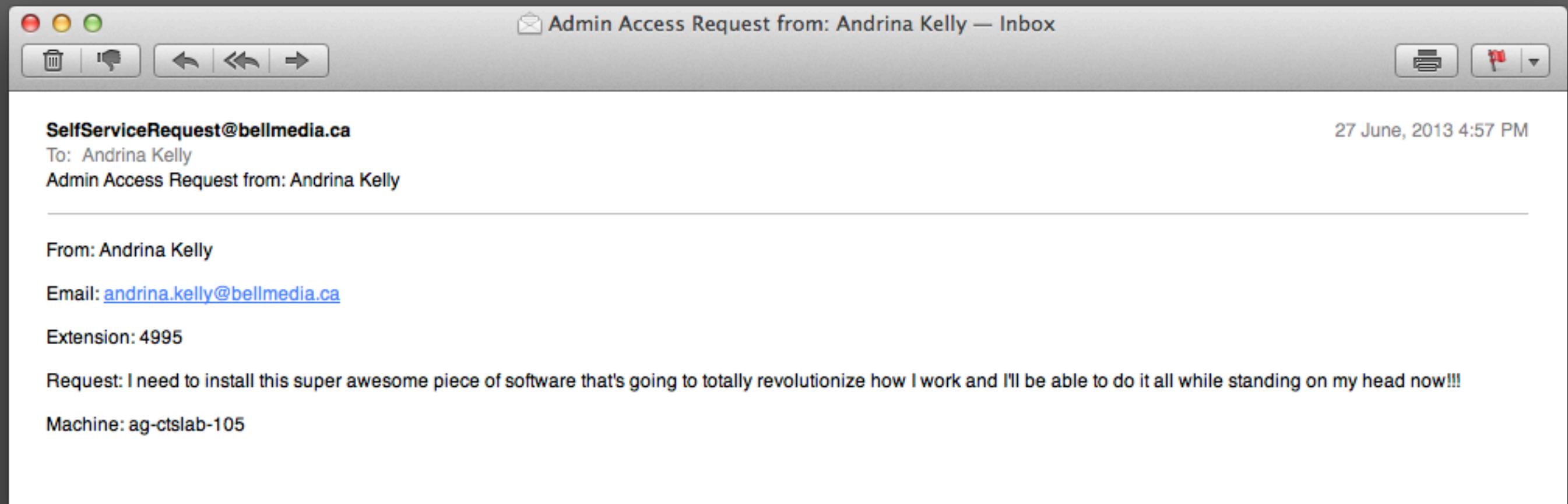
Step 2: What is Admin Access Required for?

Describe what Eg. I needed admin to install this super awesome software at 3am and didn't want to wake you
admin access is guys up!
required for:

Machine Name: Eg. ag-camcws-1234

REQUEST ACCESS





Make Me Admin

- 1st Policy = Dummy Receipt & 2 scripts
- 2nd Policy = Custom trigger, clean up receipt & 1 script

Edit Policy: Make Me Admin

General Scope Self Service Packages Scripts Printers Dock Accounts Reboot Advanced

Maintenance

Update Inventory Update Prebindings Flush System Caches
 Reset Computer Names Fix Permissions Flush User Caches
 Self Heal Packages Fix ByHost Files Verify Startup Disk

Files & Processes

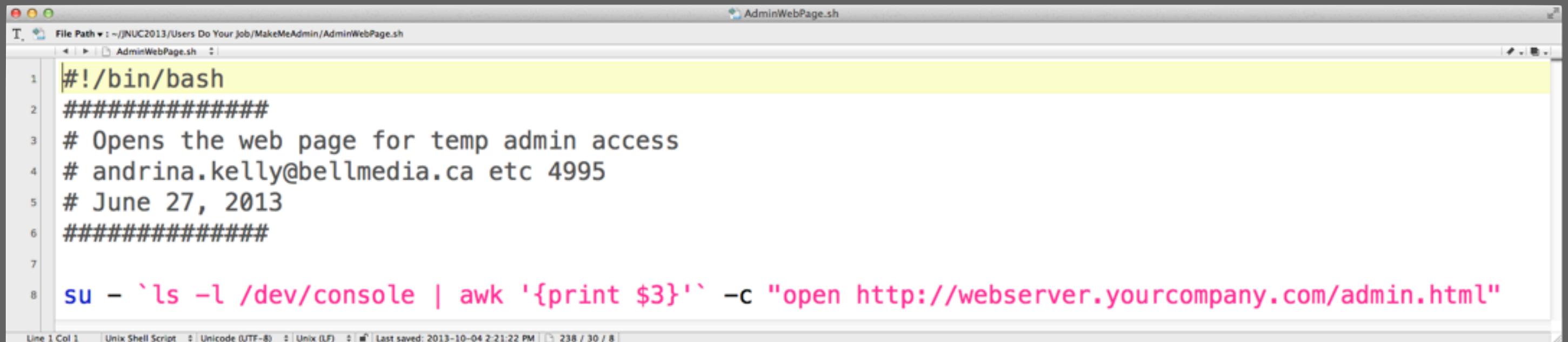
Search for file by path: Delete if found
Search for file by name: Update Locate DB
Spotlight Search:
Search for Process: Kill if found
Run Command: `touch /Library/Application\ Support/JAMF/Receipts/MakeA`

Disk Encryption Configurations

Enabled	Display Name	Recovery Key Type	Encryption Type
<input type="checkbox"/>	Bell Media FileVault	Individual And Institutional	FileVault 2

► Management Framework Options – Not required for clients running 7.3 or later

Cancel Save



```
#!/bin/bash
#####
# Opens the web page for temp admin access
# andrina.kelly@bellmedia.ca etc 4995
# June 27, 2013
#####

su - `ls -l /dev/console | awk '{print $3}'` -c "open http://webserver.yourcompany.com/admin.html"
```

```
#!/bin/bash
#####
# This script will give a user 30 minutes of Admin level access, from Jamf's self
# service.
# At the end of the 30 minutes it will then call a jamf policy with a manual
# trigger.
# Remove the users admin rights and disable the plist file this creates and
# activites.
# The removal script is removetempAdmin.sh
#####

USERNAME=`who |grep console| awk '{print $1}'`
```

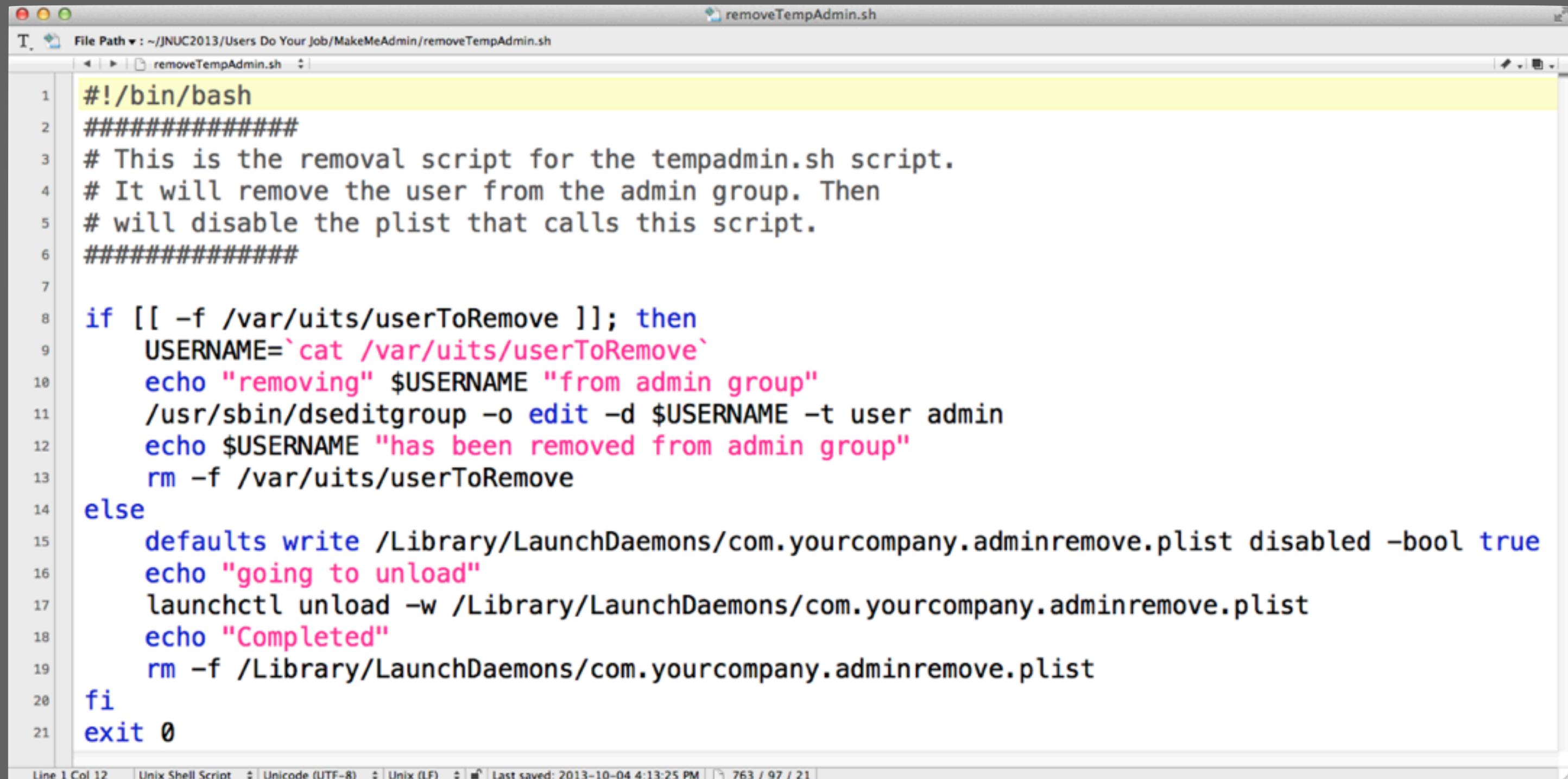
```
# Place launchd plist to call JSS policy to remove admin rights.  
#####  
echo "<?xml version='1.0' encoding='UTF-8'?>  
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"  
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">  
<plist version='1.0'>  
<dict>  
    <key>Disabled</key>  
    <true/>  
    <key>Label</key>  
    <string>com.yourcompany.adminremove</string>  
    <key>ProgramArguments</key>  
    <array>  
        <string>/usr/sbin/jamf</string>  
        <string>policy</string>  
        <string>-trigger</string>  
        <string>adminremove</string>  
    </array>  
    <key>StartInterval</key>  
    <integer>1800</integer>  
</dict>  
</plist>" > /Library/LaunchDaemons/com.yourcompany.adminremove.plist  
#####
```

```
#set the permission on the file just made.  
chown root:wheel /Library/LaunchDaemons/com.yourcompany.adminremove.plist  
chmod 644 /Library/LaunchDaemons/com.yourcompany.adminremove.plist  
defaults write /Library/LaunchDaemons/com.yourcompany.adminremove.plist disabled  
-bool false  
  
# load the removal plist timer.  
launchctl load -w /Library/LaunchDaemons/com.yourcompany.adminremove.plist
```

```
# build log files in var/uits
mkdir /var/uits
TIME=`date "+Date:%m-%d-%Y TIME:%H:%M:%S"`
echo $TIME " by " $USERNAME >> /var/uits/30minAdmin.txt

echo $USERNAME >> /var/uits/userToRemove
```

```
# give current logged user admin rights  
/usr/sbin/dseditgroup -o edit -a $USERNAME -t user admin  
exit 0
```



```
removeTempAdmin.sh
File Path : ~/JNUC2013/Users Do Your Job/MakeMeAdmin/removeTempAdmin.sh
removeTempAdmin.sh

1 #!/bin/bash
2 #####
3 # This is the removal script for the tempadmin.sh script.
4 # It will remove the user from the admin group. Then
5 # will disable the plist that calls this script.
6 #####
7
8 if [[ -f /var/uits/userToRemove ]]; then
9     USERNAME=`cat /var/uits/userToRemove`
10    echo "removing" $USERNAME "from admin group"
11    /usr/sbin/dseditgroup -o edit -d $USERNAME -t user admin
12    echo $USERNAME "has been removed from admin group"
13    rm -f /var/uits/userToRemove
14 else
15     defaults write /Library/LaunchDaemons/com.yourcompany.adminremove.plist disabled -bool true
16     echo "going to unload"
17     launchctl unload -w /Library/LaunchDaemons/com.yourcompany.adminremove.plist
18     echo "Completed"
19     rm -f /Library/LaunchDaemons/com.yourcompany.adminremove.plist
20 fi
21 exit 0
```

Line 1 Col 12 Unix Shell Script Unicode (UTF-8) Unix (LF) Last saved: 2013-10-04 4:13:25 PM 763 / 97 / 21

Policy log: ag-ctslab-105 at Thursday, June 20 2013 at 10:06 AM

```
/usr/sbin/jamf is version 8.7
Executing Policy Remove Temporary Admin...
Creating directory structure for /Library/Application Support/JAMF/Downloads/
Downloading http://casper.bellmedia.ca:80/casper-downloads/Scripts/removetempAdmin.sh...
Running script removetempAdmin.sh...
Script exit code: 0
Script result: removing akelly from admin group
akelly has been removed from admin group
Running command rm -rf /Library/Application\ Support/JAMF/Receipts/MakeAdminRan.pkg...
Result of command:
Running Recon...
```

[Flush History for this Policy/Computer](#)

Note: Clicking this button will make it appear that
this policy has never run on this computer.

[Cancel](#)

Edit Policy: Remove Temporary Admin

General Scope Self Service Packages Scripts Printers Dock Accounts Reboot Advanced

Maintenance

Update Inventory Update Prebindings Flush System Caches
 Reset Computer Names Fix Permissions Flush User Caches
 Self Heal Packages Fix ByHost Files Verify Startup Disk

Files & Processes

Search for file by path: Delete if found
 Search for file by name: Update Locate DB
 Spotlight Search:
 Search for Process: Kill if found
 Run Command: `rm -rf /Library/Application\ Support/JAMF/Receipts/MakeA`

Disk Encryption Configurations

Enabled	Display Name	Recovery Key Type	Encryption Type
<input type="checkbox"/>	Bell Media FileVault	Individual And Institutional	FileVault 2

► Management Framework Options – Not required for clients running 7.3 or later

Cancel Save

ag-ctslab-105 removed from Self Service Make Me Admin Ran — Inbox

JAMF Software Server <cts-casper@bellmedia.ca>

To: Andrina Kelly , [REDACTED] , [REDACTED] , [REDACTED] , [REDACTED] , [REDACTED]
ag-ctslab-105 removed from Self Service Make Me Admin Ran

28 June, 2013 11:44 AM Hide Details

The computer "ag-ctslab-105" was removed from the Smart Computer Group "Self Service Make Me Admin Ran".

Computer Name: ag-ctslab-105
IP Address: 10.201.4.186 <ssh://ctsadmin@10.201.4.186>

Realname: Andrina Kelly
Username: akelly
Department: CTS
Room: Agincourt 9 Channel Nine Court
Phone Number: 416-384-4995
Email Address: Andrina.Kelly@bellmedia.ca

Model: iMac13,2
Serial Number: [REDACTED]
Total RAM: 8 GB
Processor: 3.4 GHz x4 (Intel Core i7 i386)
Operating System: Mac OS X 10.8.4 (12E55)

You can access the full details of this computer at:

https://casper.bellmedia.ca/viewComputer.html?computer_id=1311

The criteria for this group is:

"Packages Installed By Casper" has "MakeAdminRan.pkg"

Make Me Admin

- Props to Kyle Brockman, University of Wisconsin–Milwaukee
 - <https://jamfnation.jamfsoftware.com/discussion.html?id=6990>
- Solves the out of hours admin requests
- Solves the “urgent” requests from remote users

Get Your Users To Do Your Job

What Next:

Scripting for Mortals - Tues. 2:15 McGuire Proscenium

Scripting Beyond Mortals - Thurs. 10:15 Dowling Studio

Cultivating a Self Service Culture - Wed. 3:30 Dowling Studio

Grazie!

Danke!

Merci!

Gracias!

Obrigado!

谢谢

Thank you!

спасибо!