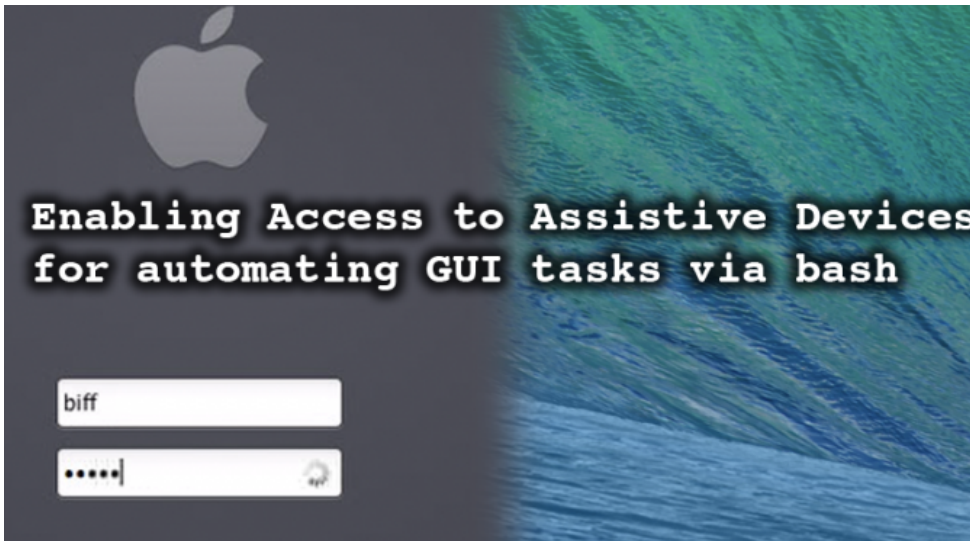




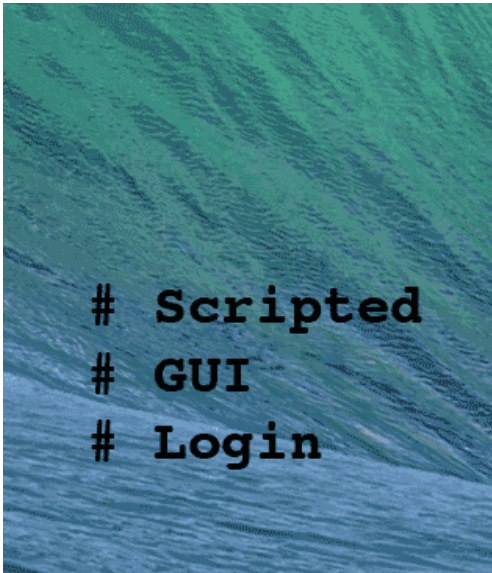
## Jacob Salmela

#!/bin/bash



## Bash Script: Enable Access to Assistive Devices Programmatically in OS X Mavericks 10.9.x – Simulate Keystrokes

Using a **bash script** to **enable access to assistive devices** is possible in Mavericks (and also Yosemite) despite the move to a per-app database. This is useful for entering keystrokes or clicking GUI buttons via a bash script.



Prior to OS X Mavericks, **enabling access to assistive devices** was relatively easy. Running the command

```
sudo touch /private/var/db/.AccessibilityAPIEnabled
```

would enable it. But now each app now needs to be allowed or disallowed access.

I often used Applescript/**osascript to simulate keystrokes or click GUI buttons**, which was great for automating “un-scriptable” tasks on multiple computers. One particular feature I always found incredibly useful was logging a user in via the loginwindow by sending it a script from a remote machine. This proved useful in a lab environment when I wanted to open a specialized piece of software to make sure it worked on all of the machines.

The script would simulate keystrokes and fill in the username and password fields and then press enter or click login. Unfortunately, Mavericks does not allow Applescript (or more accurately, `/usr/bin/osascript`) by default, which is what I used to accomplish this from my script. Trying to run this script returned an error:

```
62:115: execution error: System Events got an error: osascript is not allowed assistive access. (-10006)
```

The first thing I tried was to drag `/usr/bin/osascript` into the Accessibility window of the Security & Privacy Preferences pane, which is how you can add other apps. This did not work as the command-line tool was not an `.app`.

Needless to say, after struggling to figure out how to enable this through a script, I finally **found a way**.

## Enable Access To Assistive Devices via bash

Each entry in the Accessibility window is actually part of a database file. Using `sqlite3`, this file can be manipulated to add, remove, or modify items. To get `osascript` to work at the login window and be **allowed access to assistive devices**, run this command:

If you are using `tccutil.py`:

```
sudo tccutil.py -i com.apple.RemoteDesktopAgent
```

Or you can use `sqlite3`

```
sudo sqlite3 /Library/Application\ Support/com.apple.TCC/TCC.db "INSERT or REPLACE INTO access VALUES('kTCCServiceAccessibility',
```

To remove it:

```
sudo tccutil.py -r com.apple.RemoteDesktopAgent
```

```
sudo sqlite3 /Library/Application\ Support/com.apple.TCC/TCC.db "DELETE from access where client='com.apple.RemoteDesktopAgent'"
```

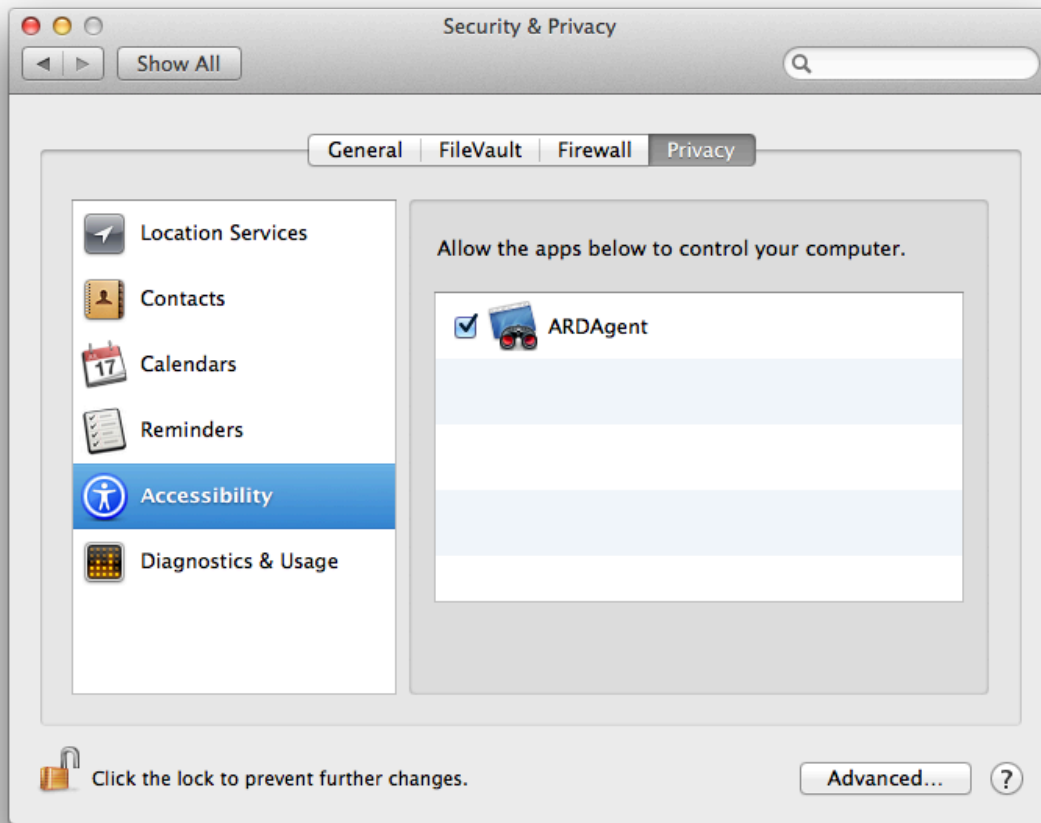
or completely wipe out every entry with:

```
tccutil reset Accessibility
```

`osascript` must be some part of `/System/Library/CoreServices/RemoteManagement/ARDAgent.app`, because another script I was working on triggered the prompt to allow it access.



After enabling it in System Preferences and rebooting,



I was able to run the script I have used for years that **enters a users credentials into the login window fields via a script** sent from Apple Remote Desktop. The script is below, but I also have [multiple versions of it on my Github page](#). Even one written in Python.

```
1 osascript -e <<EOF 'tell application "System Events"
2     tell process "SecurityAgent"
3         set value of text field 2 of window "Login" to "username"
4         set value of text field 1 of window "Login" to "password"
5     end tell
6     tell application "system events" to keystroke return
7     tell application "system events" to keystroke return
8     end tell'
9 EOF
```

os-x-login-user-to-gui-via-script.sh hosted with ❤ by GitHub

[view raw](#)

When enabling this, I also add a few other things that may come in useful:

```
sqlite3 /Library/Application\ Support/com.apple.TCC/TCC.db "INSERT or REPLACE INTO access VALUES('kTCCServiceAccessibility','com.
sqlite3 /Library/Application\ Support/com.apple.TCC/TCC.db "INSERT or REPLACE INTO access VALUES('kTCCServiceAccessibility','com.
sqlite3 /Library/Application\ Support/com.apple.TCC/TCC.db "INSERT or REPLACE INTO access VALUES('kTCCServiceAccessibility','/usr.
sqlite3 /Library/Application\ Support/com.apple.TCC/TCC.db "INSERT or REPLACE INTO access VALUES('kTCCServiceAccessibility','/usr.
```

This allows me to use things like the lock screen or the `say` command from the login window. In my scripts, I use it as a visual indicator that the job is done without having to log in. I can either listen for the command to complete or the lock screen to disappear.

## Another Solution

Apple's utility, tccutil only supports one command in Mavericks. I wrote my own version of it called **tccutil.py**. You can get it below, or look at the [source code](#).

This is just a little Python command line utility that can be used in scripts. I'm sure Apple will expand their utility, but for now, this is something you can use.

## Scripting Your Own GUI Actions

If you want to automate mouse clicks or keystrokes, I would suggest using [UIElementInspector](#), or [Accessibility Inspector](#), as it is called now (part of Xcode). This works great to find out what window, menu, or button names are called, so you can use them in your AppleScripts or bash scripts ( [osascript](#) ).



If you are using Mavericks, you will have to, ironically, allow it access to accessibility by either dropping it in the list, or following the prompts when you run the app for the first time.



## Troubleshooting

Adding osascript to the accessibility database doesn't always mean your app or script will run without producing the error. Often, you need to add the app that is being manipulated. For example, if you had an AppleScript that clicked a menu in TextEdit, you would need to add TextEdit to the accessibility database.

---

### Share this:



---

### Like this:



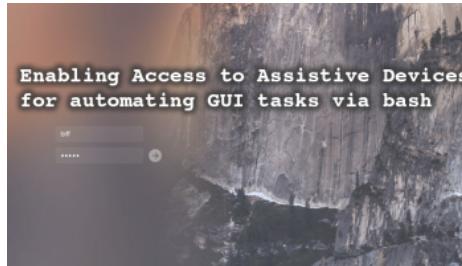
Be the first to like this.

---

### Related



[tccutil.py: Command Line Utility for Modifying the Accessibility Database in OS X](#)  
In "System Administration"



[OS X Yosemite and osascript: Enabling Access to Assistive Devices](#)  
In "System Administration"

### Test Center Workstation: One-Time Registration

To use this machine as a test center workstation, a Site Admin, Proctor, or Campus Admin must authorize it by assigning the machine a name and entering his or her credentials below.

The Machine Name appears in the proctor dashboard to identify this workstation. Please specify a unique name.

Machine Name	<input type="text"/>
Username	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="Register"/>	

[More Minnesota Testing Woes: Scripting MN ACT Compass Test](#)  
In "System Administration"

This entry was posted in System Administration and tagged administration, automation, scripting on 2014-06-12 [<http://jacobsalmela.com/bash-script-enable-access-assistive-devices-programmatically-os-x-mavericks-10-9-x-simulate-keystrokes/>].

4 Comments

Jacob Salmela

1

Login


Recommend

Share

Sort by Best



Join the discussion...

**Robert Pelletier** · 14 days ago



Thanks for the python script. However, for some reason the logon script works to enter the credentials, but does not work to enter the return key at the end. Therefore, the machines just sit there with the credentials. Any ideas as to why this would happen?

^

|

▼

· Reply · Share

**jacobsalmela** Mod  **Robert Pelletier** · 14 days ago


Try changing line 6 to `tab` instead of `return`. That seems to work for me on the newer OS's.

^

|

▼

· Reply · Share

**Rob Cole** · 10 months ago



Jacob - thank you SOOOO much for this!  
PS - my son's name is Jacob :-)  
Rob

^

|

▼

· Reply · Share

**Jacob Salmela**  **Rob Cole** · 10 months ago

Glad you found it useful!!

^

|


▼

· Reply · Share

ALSO ON JACOB SALMELA


A Web Interface For The Ad-blocking Pi-hole

5 comments • 4 months ago

**jacobsalmela** — Reports from users say that ads get blocked in apps. It still has trouble with Hulu/YouTube video ads though. But I am working on that (development is slow because I don't get ...

Pi-hole Improvements: More Ad Domains And A Better Script


10 comments • 5 months ago

**jacobsalmela** — There isn't a way to revert the list, as it just creates a brand new one each time you run the script, which won't be as much of an issue in the next release. Since the list ...

WHAT'S THIS?


Find Out What Sites Paid To Be Whitelisted From AdBlock Plus

2 comments • 6 months ago

**jacobsalmela** — Interesting. So what is the X-Adblock-Key used for?

Huge Update To Ad List That The Pi-hole Uses

8 comments • 3 months ago

**jacobsalmela** — The link in the post is fixed now and points to 127.0.0.1 instead of 192.168.1.68. Thanks for catching that! I also received a donation, which inspired me to do some more ...