

# Munki & osascript Notes

Sometimes it is still easier to use AppleScript to get things done and it is possible to have Munki's Self-Service On Demand feature use AppleScripts when a Bash Script utilizes the "osascript" command (Open Scripting Architecture).

But it seems to get harder to do all the time as Apple tightens up security in each successive macOS.

As a illustration, lets look at what I had to do to get my Self-Service On Demand script that triggers a Malwarebytes scan up and running.

Note that we first installed Malwarebytes 3.

If you are deploying via Munki then you may have to also deploy the LaunchAgents.

A good article by Clint McIntosh that discusses this is available on Jamf Nation:

<https://www.jamf.com/jamf-nation/discussions/17532/deploying-malwarebytes-for-mac-1-1-3-72>

## 1) An AppleScript for Malwarebytes 3

There were various scripts available online that worked with Malwarebytes AntiMalware 1.2 but these do not work with the revised version 3. We needed an AppleScript and while Malwarebytes received a complete overhaul going from v. 1.2.6 to v 3.0.3 it did not get an AppleScript Dictionary.

Worse, it doesn't even have classic file menus. Just GUI buttons.

Scripts posted online for version 1.2 did not work in 3.

So I used an AppleScript that passes the button click to the correct element using "System Events".

```
-- AppleScript to initiate a scan in Malwarebytes 3
-- Works when the Window is at the "Dashboard" tab but not when the index has changed to the "Scan" tab.
-- The "Dashboard" tab is active when the app is first launched.
tell application "Malwarebytes" to activate
delay 5
tell application "System Events"
    tell process "Malwarebytes"
        click button "Scan Now" of group 1 of group 1 of window "Malwarebytes" of application process "Malwarebytes" of application "System Events"
    end tell
end tell
```

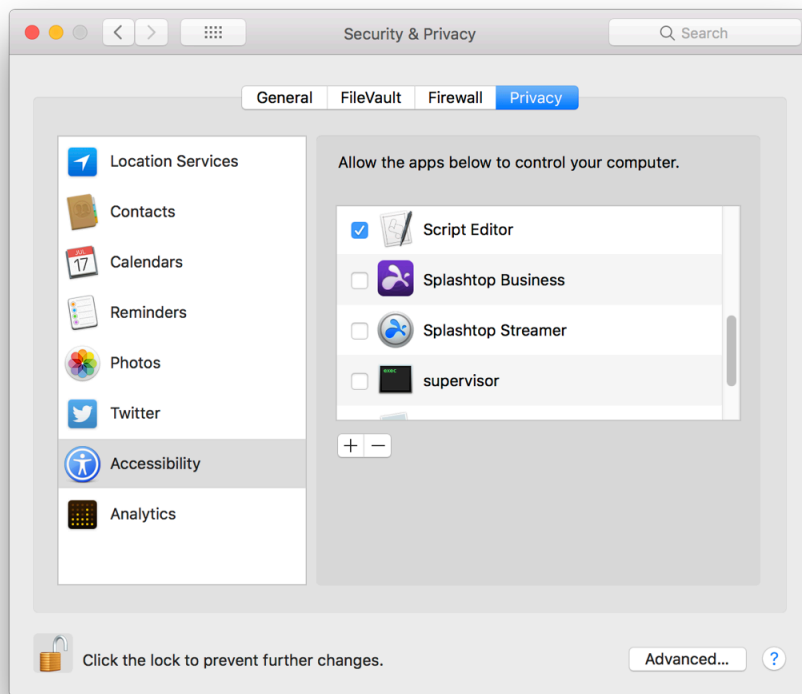
(And I discovered the syntax for clicking the button menu item using a script I modified from Jacques Rioux.

You can get the list of elements for any app that has no AppleScript Dictionary by replacing "theProcess" with your app.

```
-- adapted from a script suggested by Jacques Rioux
-- https://discussions.apple.com/thread/4390028?tstart=0
-- remember to specify "theProcess" (in this example it is the MalwareBytes 3 application)
set theProcess to "Malwarebytes"
set allButtons to {}
tell application "System Events"
    tell process theProcess
        with timeout of 0 seconds
            set tElements to entire contents of window 1
        end timeout
        repeat with i in tElements
            if class of i is button then set end of allButtons to contents of i
        end repeat
    end tell
end tell
allButtons
```

## 2) System Events enablement

For this script to run using Script Editor you must allow Script Editor in System Preferences:Security & Privacy:Privacy:Accessibility. It is easy to add using the "+" button and navigating to "Script Editor" in your Applications:Utilities folder.



### 3) A Bash Script for Malwarebytes 3

So now we take that script and bash it up with "osascript" and get:

```
#!/bin/bash
/usr/bin/osascript -e 'tell application "Malwarebytes" to activate'
/usr/bin/osascript -e 'tell application "System Events" to click button "Scan Now" of group 1 of group 1 of window "Malwarebytes" of application process "Malwarebytes" of application "System Events"'
exit 0
```

When we run this bash script on a local computer using Terminal:

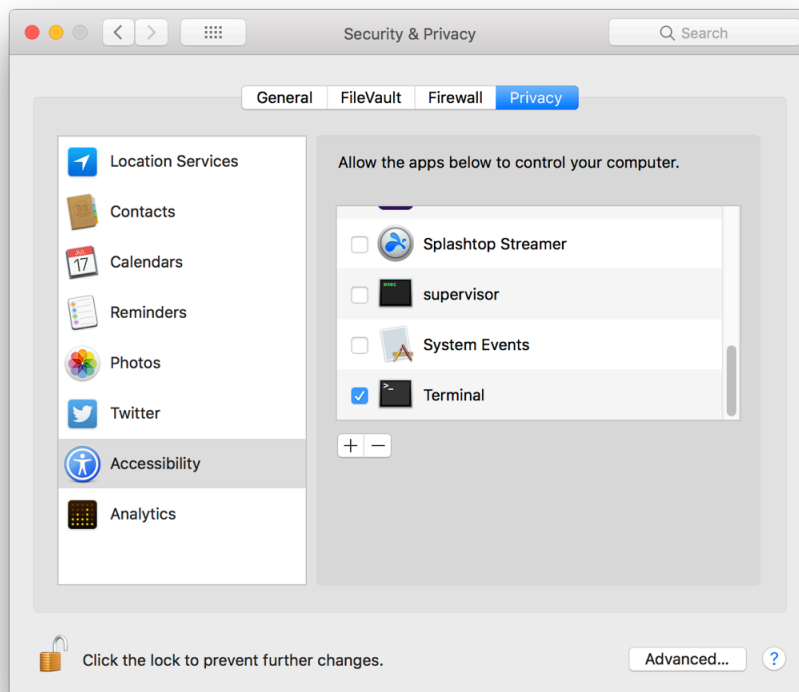
```
sh ~/desktop/RunMalwareScan.sh
```

We get an error:

```
execution error: System Events got an error: osascript is not allowed assistive access. (-1728)
```

So we have to enable Terminal in System Preferences.

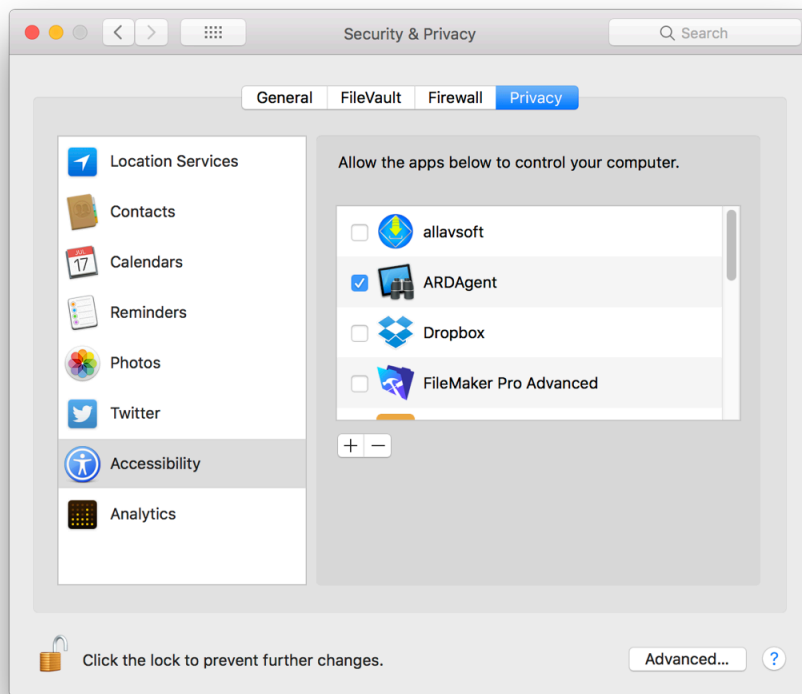
Again this is easy to do using the "+" button.



#### 4) Enabling for ARD

You could also push this bash script out using the Send Unix command of Apple Remote Desktop but then you would have to enable the client for ARD by adding the Remote Desktop Agent to Accessibility.

```
/System/Library/CoreServices/RemoteManagement/ARDAgent
```



## 5) Munki Enablement

When we put that bash script into our Munki Self-Service On Demand item and it runs as a "Post Install" script we find that although the script seems to complete without error nothing happens! There is nothing in the Error log!

So we check the Managed Install log at:

```
/LibraryManaged Installs/Logs/ManagedSoftwareUpdate.log
```

And we will see this:

```
Installing PS Malware Scan (1 of 1)
Running postinstall_script for ps-malware_scan
35:43: execution error: An error of type -10810 has occurred. (-10810)
36:175: execution error: An error of type -10810 has occurred. (-10810)
```

What we now need to enable is "Supervisor" which seems to be what Munki is using to run this osascript.

```
/usr/local/munki/supervisor
```

But Supervisor is a python script and not an Application so we can't just drag and drop it into the System Preferences or use the "+" button.

Now I think I somehow managed to get this in there when I was still on 10.11.6 so whatever I used no longer works.

And there used to be a command line to add to the Accessibility database but Apple seems to have shut that down as of 10.12.2.

It seems the database is now SIP protected and is now Read Only apparently because DropBox was modifying it to add DropBox.app without letting the user know about the security implications.

Therefore, in order to edit the database you would have to disable SIP. Run the commands and then reenble SIP.

Not all that elegant.

The commands I found online were for editing the database before the SIP protection and were for ARDAgent but I have adapted for 10.12.2, SIP, and Supervisor.

Jacob Salmela has a good description:

<https://jacobalsalmela.com/2014/06/12/bash-script-enable-access-assistive-devices-programmatically-os-x-mavericks-10-9-x-simulate-keystrokes/>

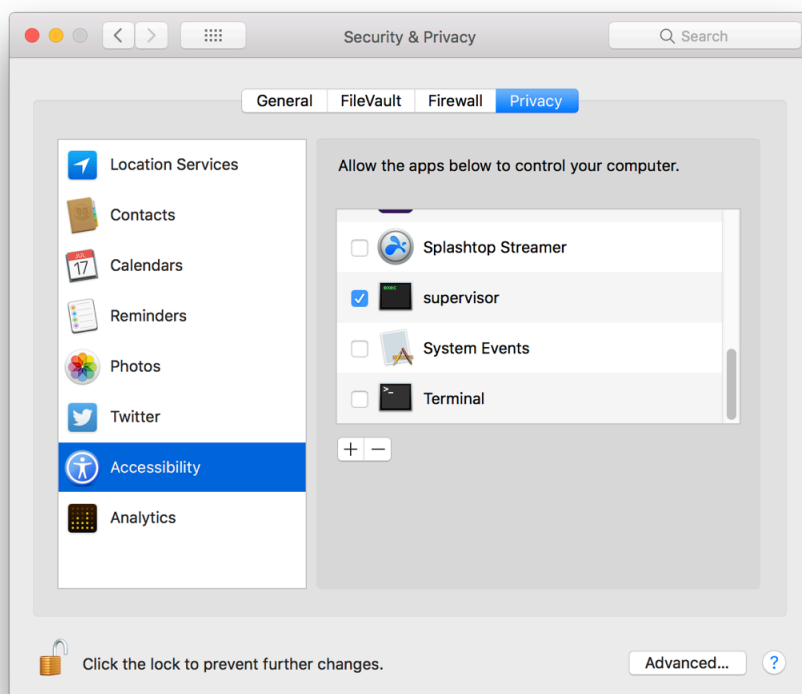
Since that time with OS 10.12 and 10.13 there appear to now be 7 fields instead of 6 so I have made adjustments accordingly.

I believe the correct command to add Supervisor might look like this:

```
sudo sqlite3 /Library/Application\ Support/com.apple.TCC/TCC.db "INSERT or REPLACE INTO access VALUES('kTCCServiceAccessibility','/usr/local/munki/supervisor',1,1,1,'(BLOB)',NULL)"
```

And to remove it, this:

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



So now with all that enabled we can initiate our Malwarebytes scan using the PS-Malware\_Scan On Demand item.  
<https://github.com/precursorca/Munki-SelfService-On-Demand>

And voila, a Malwarebytes 3 scan is properly triggered.

October 16, 2017

---

Alex Narvey  
Precursor.ca