

Nettes kleines Shell-Tool für Sleep, Wakeup und eine Menge anderer Funktionen

Beitrag von „Sascha_77“ vom 20. Dezember 2015, 11:05

Morgen Leute.

Ich habe mir gestern auf meinem T61p ein kleines Tool installiert. Damit kann man Aktionen ausführen wenn man den Rechner in den Sleep schickt bzw. er wieder aufwacht.

Sicherlich gibts hier bereits schon Programme für. Das sind aber GUI-Lösungen und zudem noch kostenpflichtig. Das Tool was ich verwende arbeitet in der Shell und ist somit äußerst schlank und an Ressourcen benötigt es so gut wie gar nichts.

<http://www.bernhard-baehr.de/>

http://www.bernhard-baehr.de/sleepwatcher_2.2.tgz

Auf meinem Rechner nutze ich es dafür um z.B. eine VPN-Verbindung vor dem Sleep zu beenden. Macht man das nicht hat man jedesmal nach dem Aufwachen eine Fehlermeldung, dass die VPN-Verbindung unterbrochen wurde. Nervig. Und so wird sie unterbrochen vor dem Sleep und beim Aufwachen wieder hergestellt. Das gleiche mache ich mit dem HWMonitor welcher auf unseren T61(p) sehr wichtig ist. Dieser scheint manchmal nicht mehr richtig zu klappen nach dem Aufwachen. Ihn jedesmal von Hand zu beenden und neu zu starten war mir irgendwann aber zu blöd. Und wenn man es vergisst läuft man Gefahr, dass der Rechner irgendwann zu heiss wird da die Sensoren nicht mehr ausgelesen werden.

Lange Rede kurzer Sinn hier nun meine Scripte als Beispiel:

```
~/Dokumente/shell/sleepwatcher/sleep.sh
```

Code

1. `osascript -e 'tell application "HWMonitor" to quit'`
2. `scutil --nc stop "VPN (PPTP)"`

~/Dokumente/shell/sleepwatcher/wakeup.sh

Code

1. `sleep 5`
2. `osascript -e 'tell application "HWMonitor" to launch'`
- 3.
- 4.
5. `while [true]`
6. `do`
7. `if ping -c 1 -W 2 192.168.1.254 &> /dev/null`
8. `then`
9. `scutil --nc start "VPN (PPTP)"`
10. `exit`
11. `else`
12. `echo 0`
13. `fi`
14. `done`

Alles anzeigen

Damit das Ganze auch beim Booten automatisch geladen wird benötigt man noch eine .plist für den LaunchAgent.

~/Library/LaunchAgents/de.bernhard-baehr.sleepwatcher.plist

XML

1. `<?xml version="1.0" encoding="UTF-8"?>`
2. `<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">`
3. `<plist version="1.0">`
4. `<dict>`
5. `<key>Label</key>`

6. <string>de.bernhard-baehr.sleepwatcher</string>
7. <key>ProgramArguments</key>
8. <array>
9. <string>~/Documents/shell/sleepwatcher/sleepwatcher</string>
10. <string>-s ~/Documents/shell/sleepwatcher/sleep.sh</string>
11. <string>-w ~/Documents/shell/sleepwatcher/wakeup.sh</string>
12. </array>
13. <key>RunAtLoad</key>
14. <true/>
15. <key>KeepAlive</key>
16. <true/>
17. </dict>
18. </plist>

Alles anzeigen

Wenn man jetzt nicht Neustarten möchte um den Agent erstmalig in Betrieb zu nehmen, so kann man das mit folgender Befehlszeile tun:

Code

1. `sudo launchctl load ~/Library/LaunchAgents/de.bernhard-baehr.sleepwatcher.plist`

Wenn man die Beispiele von den Pfaden her so übernimmt muss das Ganze am Ende wie auf dem Bild zu sehen ausschauen. Kann jeder natürlich anders machen.

PS: Habe auch mal das Programm selber mit angehängen falls die offizielle Webseite irgendwann mal weg sein sollte.

Beitrag von „fundave3“ vom 20. Dezember 2015, 11:56

Geil Danke dir . Dann kann ich ja VPN Verbindungen trennen lassen und wieder herstellen lassen.

Beitrag von „thommel“ vom 20. Dezember 2015, 12:15

Tolle Sache, Danke 👍

Beitrag von „Sascha_77“ vom 20. Dezember 2015, 12:18

[Zitat von fundave3](#)

Geil Danke dir . Dann kann ich ja VPN Verbindungen trennen lassen und wieder herstellen lassen.

Dann solltest Du die IP deines Routers aber in meinem Script ändern. Aber vllt. hast Du ja die selbe.

Beitrag von „fundave3“ vom 20. Dezember 2015, 12:24

Sicher nicht. Im Lokalen NETz benutze ich 10er IPs. 10.120.21.1
Und die interne im VPN Server netz 10.211.120.2

Also von daher ich habe schon daran gedacht ja aber Vielen Dank für den Hinweis

Beitrag von „Sascha_77“ vom 20. Dezember 2015, 12:29

Ich habe mein wakeup script gerade kurz angepasst. So pollt er jetzt nicht wie blöd hintereinander sondern wartet immer 2 Sekunden bis er die Pingbarkeit erneut prüft.

Beitrag von „fundave3“ vom 20. Dezember 2015, 12:30

Das ist aber toll. Danke

Beitrag von „griven“ vom 21. Dezember 2015, 22:21

Gute Sache wäre auch was für die WIKI 😄
Wenn ich darf würde ich das gerne übernehmen.

Beitrag von „Sascha_77“ vom 22. Dezember 2015, 08:22

Na aber sicher doch. Immer rein damit. 😊

Beitrag von „griven“ vom 23. Dezember 2015, 22:47

Und drinn ist es 😄

Beitrag von „Sascha_77“ vom 24. Dezember 2015, 07:56

Jut. Ich sehe Du hast es unter Notebook gepackt. Macht es nicht Sinn das unter Tools zu packen? Auf einem Desktoprechner kann man das ja auch gut gebrauchen unter Umständen

(für einen VPN-Reconnect z.B.).

Beitrag von „griven“ vom 24. Dezember 2015, 23:53

War mir selbst nicht so ganz sicher wo ich es einpacken soll, da war die Lösung in der alten Software flexibler denn da konnte man einfach beides anklicken...

Mal sehen ich packe es vielleicht auch noch mal zu den Tools...

Beitrag von „burzlbaum“ vom 25. Dezember 2015, 20:39

Könnte dadurch auch das Sleep Problem mit einigen ATI Grafikkarten gelöst werden?

Beitrag von „Sascha_77“ vom 26. Dezember 2015, 18:22

Eher nicht. Das Tool führt lediglich Scripte aus. Das einzige was man versuchen könnte ist, dass beim Aufwachen die GFX-Kext geunloaded und direkt wieder geloaded werden. In wiefern das im laufenden Betrieb überhaupt möglich ist entzieht sich meiner Kenntniss.

Beitrag von „anonymous_writer“ vom 28. Februar 2018, 18:37

Ist zwar etwas älter der Beitrag aber das Tool erfüllt auch unter High Sierra seinen Dienst. Dachte mir daher, stelle hier mal ein praktisches Beispiel ein für einen Kext der nicht so arbeitet wie er soll.

Problem bei mir:

Der Kext VoodooI2C.kext erzeugt ab und zu bei mir eine Kernel Panic im Sleepmodus.

Außerdem funktioniert er nach Sleep nicht mehr.

Daher war mein Lösungsansatz den Kext vor dem Sleep zu entladen und danach wieder zu laden. Funktioniert super. 😊

Verwenden kann man das bei jedem Kext der sich im **laufenden Modus entladen lässt**. Geht leider nicht mit jedem Kext.

Zur Installation gibt es vom Entwickler eine Installation Routine. Da ich die hier noch nicht gefunden habe, hänge ich die auch mal an.

Code

```
1. ?#!/bin/bash
2.
3.
4. # acquire sudo at the beginning
5. sudo -v
6.
7.
8. # Keep-alive: update existing `sudo` time stamp until `.osx` has finished
9. while true; do sudo -n true; sleep 60; kill -0 "$$" || exit; done 2>/dev/null &
10.
11.
12. # unload launch agents
13. sudo launchctl unload /Library/LaunchDaemons/de.bernhard-baehr.sleepwatcher.plist
14. launchctl unload ~/Library/LaunchAgents/de.bernhard-baehr.sleepwatcher.plist
15.
16.
17. # remove plist launchagents
18. sudo rm -f /Library/LaunchDaemons/de.bernhard-baehr.sleepwatcher.plist
19. rm -f ~/Library/LaunchAgents/de.bernhard-baehr.sleepwatcher.plist
20.
21.
22. # remove executable and man files
23. sudo rm -f /usr/local/sbin/sleepwatcher
24. sudo rm -f /usr/local/share/man/man8/sleepwatcher.8
25.
26.
```

```
27. # download sleepwatcher package, untar, and cd into directory
28. curl --remote-name "http://www.bernhard-baehr.de/sleepwatcher_2.2.tgz"
29. tar xvzf sleepwatcher_2.2.tgz 2>/dev/null
30. cd sleepwatcher_2.2
31.
32.
33. # create folders necessary for installation
34. sudo mkdir -p /usr/local/sbin /usr/local/share/man/man8
35.
36.
37. # move files into installation folders
38. sudo cp sleepwatcher /usr/local/sbin
39. sudo cp sleepwatcher.8 /usr/local/share/man/man8
40. sudo          cp          config/de.bernhard-baehr.sleepwatcher-20compatibility.plist
    /Library/LaunchDaemons/de.bernhard-baehr.sleepwatcher.plist
41.
42.
43. sleep 1
44.
45.
46. # load launch agent
47. sudo launchctl load -w -F /Library/LaunchDaemons/de.bernhard-baehr.sleepwatcher.plist
48.
49.
50. # create script in local user directory and make them executable
51. sudo touch /etc/rc.wakeup
52. sudo touch /etc/rc.sleep
53. sudo chmod +x /etc/rc.sleep /etc/rc.wakeup
```

Alles anzeigen

Und hier ein Beispiel für die Dateien welche das Programm aufrufen.

/etc/rc.sleep:

Code

```
1. ?#!/bin/bash
2.
3.
4. # Please replace "yourpasswordhere" with your login password, make sure you put your
   password in between the quotes, like, Ex324Ins should be "Ex324Ins"
```

- 5.
- 6.
7. userPass="yourpasswordhere"
- 8.
- 9.
10. echo \$userPass | sudo -v -S &>/dev/null
- 11.
- 12.
13. # Assuming kext is installed at /S/L/E, Change path if not.
- 14.
- 15.
16. sudo kextunload /System/VoodooI2C.kext
- 17.
- 18.
19. exit

Alles anzeigen

/etc/rc.wakeup:

Code

1. ?#!/bin/bash
- 2.
- 3.
4. # Please replace "yourpasswordhere" with your login password, make sure you put your password in between the quotes, like, Ex324Ins should be "Ex324Ins"
- 5.
- 6.
7. userPass="yourpasswordhere"
- 8.
- 9.
10. echo \$userPass | sudo -v -S &>/dev/null
- 11.
- 12.
13. # Assuming kext is installed at /S/L/E, Change path if not.
- 14.
- 15.
16. sleep 4
- 17.
- 18.
19. sudo kextload /System/VoodooI2C.kext
- 20.
- 21.
22. exit

Alles anzeigen

Der Kext wird bei mir über Clover geladen. Damit ich den entladen und wieder laden kann habe ich eine Kopie hier:

"/System/VoodooI2C.kext" kopiert.

Damit der auch geladen und entladen wird müssen die Rechte richtig gestellt werden.

Code

1. sudo chmod -R 755 /System/VoodooI2C.kext

Code

1. sudo chown -R root:wheel /System/VoodooI2C.kext

Das war es eigentlich schon. Vielleicht hilft es jemanden sein Sleep Problem zu lösen.

Beitrag von „Noir0SX“ vom 28. Februar 2018, 19:43

"/System/AppleALC.kext" kopiert. Das war glaube mehr der Wunsch. Sollte bestimmt "/System/VoodooI2C.kext" kopiert. heißen.

Beitrag von „anonymous_writer“ vom 28. Februar 2018, 20:19

Hallo [@BlackOSX](#),

Stimmt, war ein Überbleibsel vom Test mit dem AppleALC.kext. Leider kann man denn nicht im laufenden Prozess entladen, aber vielleicht bekommt das einer hin.

Zusammen mit dem CodecCommander.kext im Clover Ordner funktioniert der Ton bei mir wunderbar nach Sleep.

Beitrag von „Sascha_77“ vom 19. März 2018, 21:27

Ich habe mich gerade noch mal etwas mit dem Tool auseinandergesetzt. Mir war gar nicht bewusst was für Überwachungsfunktionen da noch drinstecken. Z.b., habe ich hier ein T440 welches in den Sleepmodus geht sobald das Display auf Standby geht. Ich musste in den Energieoptionen den autom. Ruhezustand des Monitors auf "Nie" setzen damit das aufhört. Allerdings muss ich jetzt immer über einen Hot Corner das Display manuell in Standby schicken. Mittels diesem Befehls macht er das nach 5 Minuten automatisch sofern die Maus nicht bewegt oder auf der Tastatur getippt wird.

Code

1. `sleepwatcher -d -i "pmset displaysleepnow" -t 3000`

Und das hier ist die gesamte Palette ... kann sich sehen lassen:

Spoiler anzeigen

Klein aber fein kann man da nur sagen. 😊

Beitrag von „Hessi“ vom 18. Januar 2020, 16:59

[Sascha 77](#) : Habe mich exakt an Deine Vorgaben gehalten. Wenn ich nun eines meiner Scripte aufrufen will, bleibt Sleepwatcher augenscheinlich hängen, bis ich es mit ctrl-c beende.

Code

1. Michael:~ michael\$ /usr/local/sbin/sleepwatcher --verbose --sleep /Users/michael/Documents/shell/sleepwatcher/wakeup.sh
2. ^Csleepwatcher: got SIGINT - exiting
3. Michael:~ michael\$

Du bist nicht der Entwickler, ist klar, aber hast Du eine Idee?

Meine Scripte:

sleep.sh

Code

1. osascript
2. tell application "Phoebetria"
3. quit
4. end tell

wakeup.sh

Code

1. osascript
2. tell application "Phoebetria"
3. activate
4. end tell

Als Applescript im Editor ohne "osascript" ausgeführt funktionieren die Scripte (aber ich habe genau null Ahnung von Applescript).

Danke
Gruß
Michael

P.S.: Ist Dein A1200 einer der "Neugeräte" aus der Aktion vom Pedro, oder hast Du den schon lange aufgerüstet?

Beitrag von „anonymous_writer“ vom 9. März 2021, 19:33

Der Sleepwatcher läuft etwas verändert auch unter BigSur 1A. Er kann für alle möglichen Tasks verwendet werden die man so machen möchte wenn der Rechner in denn Ruhezustand geht oder von diesem Aufwacht.

Die Angepasste Konfiguration habe ich hier angehängt.

Installation wie folgt:

1. Die Zip auf dem Desktop entpacken.
2. Im Terminal in den Paket Ordner wechseln.

Code

1. `cd ~/Desktop/sleepwatcher_2.2.1_BigSur`
3. Die nötigen Dateien installieren.

Code

1. `sudo mkdir -p /usr/local/share/man/man8`
2. `sudo cp sleepwatcher /usr/local/bin`
3. `sudo cp sleepwatcher.8 /usr/local/share/man/man8`
4. `sudo cp ./config/de.bernhard-baehr.sleepwatcher-20compatibility.plist /Library/LaunchDaemons/de.bernhard-baehr.sleepwatcher.plist`

sleepwatcher läuft hiermit bereits. Mit

Code

1. `man sleepwatcher`

kann die Hilfe zum Programm aufgerufen werden.

4. Die Script Dateien erstellen für Sleep und Wake.

Code

1. `sudo nano /etc/rc.wakeup`
2. `sudo nano /etc/rc.sleep`

und mit irgendwas befühlen zum Beispiel als Test das hier

Shell-Script

1. `#!/bin/sh`
2. `mkdir /<UserName>/Desktop/Ich_war_hier`
3. `exit`

6. Die Dateien ausführbar machen

Code

1. `sudo chmod +x /etc/rc.sleep /etc/rc.wakeup`

Nach einem Neustart läuft sleepwatcher und macht exact das was in den beiden Dateien rc.wakeup und rc.sleep steht.

Beitrag von „Sascha_77“ vom 19. Mai 2021, 15:32

Habe mich mal drangesetzt und angefangen eine GUI dafür zu bauen. Irgendwie gibt es kein Tool (kostenlos) was diese Funktionen von sleepwatcher alle beinhaltet. Zumindest habe ich keins gefunden. 🤔 BTT (BetterTouchTool) sollte das zwar können aber es ist ja kostenpflichtig. Und BTT ist ziemlich umfangreich. Das wäre mit Kanonen auf Spatzen geschossen für so eine Funktion.



Beitrag von „polaplex“ vom 19. Mai 2021, 16:21

Sieht super aus. Was meinst du wann kann man es runterladen und auf dem M1 sollte es doch auch laufen oder ?

Beitrag von „Sascha_77“ vom 20. Mai 2021, 14:50

[polaplex](#)

Release siehe hier 😊

[Sleep-o-Mat \(sleepwatcher GUI\)](#)

Beitrag von „polaplex“ vom 20. Mai 2021, 18:17

Hast du die ganze nacht durch gemacht ? lol. Super Arbeit 😊

Beitrag von „Sascha_77“ vom 21. Mai 2021, 18:26

Nein aber es ging besser als ich erwartet habe.

EDIT:

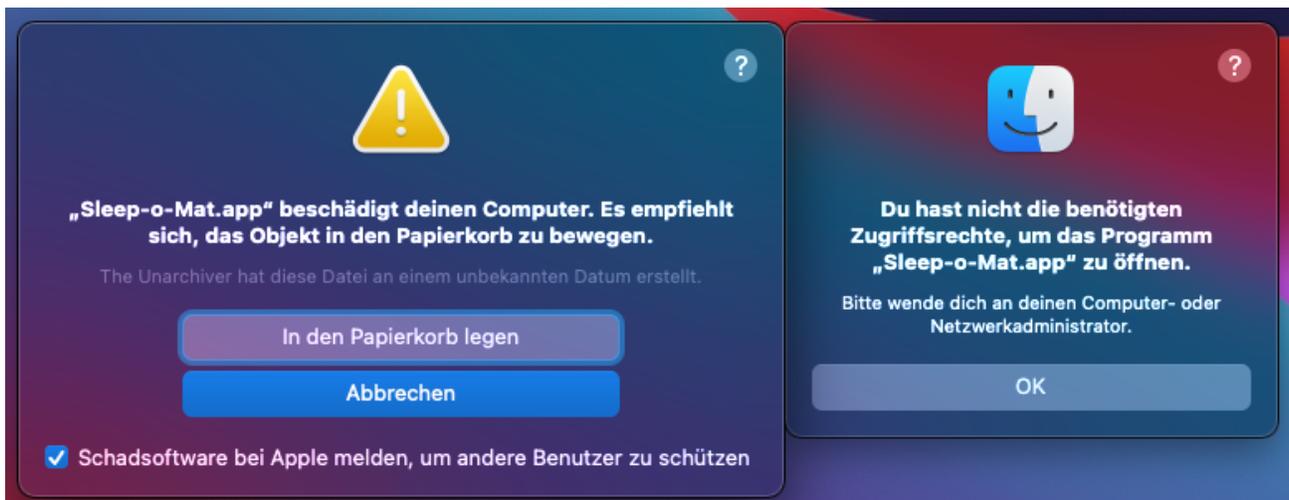
Gab nun ein kleines Update. Man kann jetzt einen Standardtexteditor festsetzen. Desweiteren kann man sich Vorlagen via Button erstellen und diese mit dem Editorbutton rechts davon direkt bearbeiten.



Beitrag von „LetsGo“ vom 11. Oktober 2021, 13:27

[Sascha 77](#)

Hallo. Benutze gerade Sleepwatcher, damit Undervolting (Voltageshift) nach dem Sleep sofort wieder aktiv ist. Dann habe ich dein nettes GUI gefunden und wollte das mal testen. Aber ich bekomme immer folgende Meldung, wenn ich das Programm unter BigSur ausführen möchte. Hast du eine Idee?



Beitrag von „anonymous_writer“ vom 11. Oktober 2021, 13:56

Rechst Klick im Finder auf das Programm und dann öffnen geht nicht?

Beitrag von „LetsGo“ vom 11. Oktober 2021, 14:07

Nein. Die Fehlermeldungen kommen eben, wenn ich die App mit Rechtsklick öffnen möchte. Und unter Systemeinstellungen/Datenschutz gibts auch nichts, damit ich die App vertrauenswürdig machen könnte.

Beitrag von „Sascha_77“ vom 11. Oktober 2021, 19:00

Geht vermutlich nur dann wenn [SIP](#) deaktiviert ist?

Wenn man bei Apple als Entwickler nicht die 99 Dollar im Jahr löhnt ist man unverifizierter Entwickler und kriegt da den ein oder anderen Knüppel zwischen die Beine geworfen. Bzw. ich veröffentliche meine Apps unsigned weil die sonst oft direkt abstürzen wenn ich die unverifiziert signiere. Das ist sone Sache die mich bei Apple ziemlich ank..... Und natürlich macht Apple das nur damit der User vor Schadsoftware weitesgehend sicher ist. Na aber klar doch. 🙄

Sorry musste jetzt mal sein. Regt mich einfach auf.

Beitrag von „kaneske“ vom 11. Oktober 2021, 21:52

Interessant wäre doch [Sascha_77](#) :

Wenn du deinen KU in den MAS bringen wollen würdest, als NoBrainer sagen wir mal, für 99ct...

Da wäre der DEV Account ruckzuck wieder drin trotz AAPL_Tax...

...netter Feldversuch ob die das Teil zulassen

sorry Steilvorlage Post vorher

Beitrag von „Sascha_77“ vom 12. Oktober 2021, 06:01

Das Teil entspräche denke ich nicht den MAS Richtlinien. Nicht weils für Hackintosh ist sondern wie das Teil aufs System zugreift und was alles im Bundle enthalten ist. Das würde glaube ich nie durchkommen.

Das Blöde ist das, um es einfach mal zu versuchen, man die 99 Schleifen schon gelöhnt haben muss. Sonst würde ich mir den Spaß einfach mal machen.