

Erledigt

Wacom Intuos S funktioniert nicht nach dem Aufwachen

Beitrag von „mabam“ vom 30. Juni 2019, 00:18

So, falls irgendwer mal ein ähnliches Problem hat, möchte ich meine letztendliche Konfiguration hier dokumentieren:

Ich wollte mit sleepwatcher keinen externen Skript ausführen, obwohl das eigentlich so gedacht ist. Denn sonst muss ich neben dem Launch Agent unter /Library/LaunchAgents/org.mabam.wacom.sleepwatcher.plist noch irgendwo anders einen Skript unterbringen, der aber eigentlich zum Launch Agent gehört und ein recht einsames Dasein pflegen würde. Ich hab' lieber übersichtlich alles in einem, damit ich auch Jahre später noch auf einen Blick sehen kann, was wozu gehört.

Wenn ich im Launch Agent Variablen ins Argument hinter /usr/local/sbin/sleepwatcher schreibe, expandiert letzterer diese einmal und merkt sich den daraus resultierenden Wert für jegliche folgende Ausführung. Sollte ich in der Zwischenzeit das Intuos also in einen anderen USB-Port gesteckt haben, versagt sleepwatcher.

Gelöst habe ich das, indem der Launch Agent beim Laden den benötigten Skript nach /tmp schreibt, wo sleepwatcher ihn dann ausführt. So werden die Variablen bei jedem Ausführen neu expandiert und ich kann das Intuos nach Lust und Laune umstöpseln. Beim Runterfahren wird der Skript vom System gelöscht und beim nächsten Hochfahren vom Launch Agent wieder erstellt (ein ewiger Kampf also ... 😊). Auf diese Weise ist alles Nötige einzig im Launch Agent enthalten.

Das Problem, dass Dateien in /tmp nicht ausführbar gemacht werden können, wird umgangen, indem das Argument hinter /usr/local/sbin/sleepwatcher mit /bin/bash beginnt.

Hier der Launch Agent:

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>org.mabam.wacom.sleepwatcher</string>
7. <key>LimitLoadToSessionType</key>
8. <array>
9. <string>LoginWindow</string>
10. <string>Aqua</string>
11. </array>
12. <key>ProgramArguments</key>
13. <array>
14. <string>/bin/bash</string>
15. <string>-c</string>
16. <string>printf '# Get current status from uhubctl; parse it to only keep the ID\n# of the hub with the port no. the Intuos S is connected to.\nuhubctlHubPort=\$(/usr/local/bin/uhubctl | awk -F '[':]'\ '"/Current status for hub/{u=\$5} /Intuos S/ {print u; print \$4}')'\n\n# Split the two lines from above into two separate variables.\nuhubctlHub=\$(printf \$uhubctlHubPort)\nuhubctlPort=\$(printf "\${uhubctlHubPort}##\$'\n\n}')'\n\n# Unpower the Intuos S using the generated variables.\n/usr/local/bin/uhubctl -a off -el \$uhubctlHub -p \$uhubctlPort\n' > /tmp/org.mabam.wacom.sleepwatcher.sh; /usr/local/sbin/sleepwatcher -Vs "/bin/bash /tmp/org.mabam.wacom.sleepwatcher.sh"</string>
17. </array>
18. <key>RunAtLoad</key>
19. <true/>
20. <key>KeepAlive</key>
21. <true/>
22. </dict>
23. </plist>

Alles anzeigen

Ich habe inzwischen übrigens festgestellt, dass mein Dell 1704FPT-Monitor über ein per uhubctl schaltbares USB 2-Hub verfügt. Wenn ich das mit obiger Lösung verwende, wacht der Rechner jedoch sofort wieder auf, nachdem er im Sleep-Modus angelangt ist. Seltsam. Wenn ich das AmazonBasics-Hub (USB 3) verwende, funktioniert alles einwandfrei.