

Erledigt

Brauche Hilfe bei Skript-Erstellung

Beitrag von „Maddeen“ vom 27. Juli 2019, 18:04

Hi zusammen,

mich plagt leider seit langem das Problem, dass mein Papierkorb sich selber zerstört

Dann muss ich die folgenden commands durchlaufen lassen und danach funktioniert wieder alles - ca. 2-3 Tage.

Und dann gehts wieder von neuem los.

Bin auch nicht alleine (siehe [LINK](#))

Daher die Frage - wie bekomme ich die folgenden commands in ein Skript, was ich dann bei jedem boot automatisch ausführen lassen kann?

Vielen Dank und schönes Wochenende

Code

1. `sudo rm -rf ~/.Trash`
2. `sudo rm -rf /Volumes/*.Trashes`
3. `mkdir ~/.Trash`
4. `sudo chown $UID ~/.Trash`
5. `chmod u+rwx ~/.Trash`

Beitrag von „sunraid“ vom 27. Juli 2019, 21:41

Versuche es mal hiermit:

<https://developer.apple.com/li...hnotes/tn2065/index.html>

Beitrag von „iPhoneTruth“ vom 28. Juli 2019, 09:18

Ich bin jetzt nicht der große Scriptersteller, aber vielleicht klappt es mit dem angefügten.

Beitrag von „Maddeen“ vom 28. Juli 2019, 11:38

[sunraid](#) - danke, aber das übersteigt meine Fähigkeiten... aber vielleicht mit Hilfe von iPhoneTruth 😊

[iPhoneTruth](#) - leider nein - er wirft folgenden Fehler:

Code

1. error "„Finder“ hat einen Fehler erhalten: sudo: no tty present and no askpass program specified" number 1

Ich glaube der kennt kein Sudo oder? In dem Link von Sunraid steht sowas wie:

Code

1. do shell script "command" user name "me" password "mypassword" with administrator privileges

Also muss doch aus deinem

Code

1. tell application "Finder"
2. activate
3. do shell script "sudo rm -rf ~/.Trash"

ein

Code

1. tell application "Finder"
2. activate
3. do shell script "rm -rf ~/.Trash" user name "me" password "mypassword" with administrator privileges

werden, oder? 🤔

Update: Ich habe das Script mal wie folgt angepasst.

Code

1. tell application "Finder"
2. activate
3. do shell script "rm -rf ~/.Trash" user name "myUsername" password "myPassword" with administrator privileges
4. do shell script "rm -rf /Volumes/*/Trashes" user name "myUsername" password "myPassword" with administrator privileges
5. do shell script "mkdir ~/.Trash" user name "myUsername" password "myPassword" with administrator privileges
6. do shell script "chown \$UID ~/.Trash" user name "myUsername" password "myPassword" with administrator privileges
7. do shell script "chmod u+rwx ~/.Trash" user name "myUsername" password "myPassword" with administrator privileges
8. end tell

Das Ergebnis lt. ScriptEditor ist jetzt --> "" (was auch immer das bedeutet) 😊

Aber es scheint zu funktionieren... werde noch was testen und gucken, dass ich es in den Autostart bekomme ... Danke nochmal!!

Beitrag von „sunraid“ vom 28. Juli 2019, 12:33

Sudo weglassen

tell *application* "Finder"

activate

do shell script "rm -rf ~/.Trash" **with** administrator privileges

end tell

Wenn du das so vorgibst musst du nicht dein Passwort in das Script einfügen. Es erscheint eine Dialogbox mit der PWabfrage...

Bin gerade überfragt, ob du bei jedem do shell "with administrator privileges" anhängen muß. Einfach mal probieren

EDIT: Wenn ich das richtig verstehen,

"...Once a script is correctly authenticated, it will not ask for authentication again for five minutes..."

reicht es "with administrator privileges" einmal beim ersten do shell anzuhängen!

Beitrag von „Sascha_77“ vom 28. Juli 2019, 12:43

Hab dir mal ne App gebastelt, die Du nur in Startobjekte legen musst. Du musst aber noch im Bundle selber in der Datei **script**

Shell-Script

1. #!/bin/bash
- 2.
3. password="DEIN_ROOT_PASSWORT"
- 4.
5. echo "\$password" | sudo -S rm -rf ~/.Trash
6. echo "\$password" | sudo -S rm -rf /Volumes/*/Trashes
7. mkdir ~/.Trash
8. echo "\$password" | sudo -S chown \$UID ~/.Trash
9. chmod u+rwX ~/.Trash

dein Rootpasswort hinterlegen. Dann macht er alles bei einem Neustart automatisch. Ich kann es dir auch so machen, dass er dich jedesmal nach dem Passwort fragt und es nicht fest hinterlegt ist.

EDIT:

War noch ein Fehler drin. Habs neu hochgeladen.

Beitrag von „sunraid“ vom 28. Juli 2019, 12:48

Hi Sascha,

kannst du mal kurz erläutern, wie man aus dem Shell-Script eine APP macht!

Würde mich echt interessieren, dann muß man vorhandene Script nicht immer mit AppleScript aufpimpen...

Gruß und Danke!

Joe

Beitrag von „Sascha_77“ vom 28. Juli 2019, 12:49

Hab das Script oben geändert. Da war noch ein Fehler drin. Bitte nochmal neu runterladen.

[sunraid](#)

Seit zig Jahren mein guter Freund:

<https://sveinbjorn.org/platypus>

Beitrag von „sunraid“ vom 28. Juli 2019, 12:51

[Sascha_77](#) Cool!!! Wenn du noch mehr solcher Schmankerl auf Lager hast, die einem das Leben, eh proggen vereinfachen, immer her damit!

Beitrag von „Sascha_77“ vom 28. Juli 2019, 12:57

Hat zwar nix mit proggen als solches zu tun

<https://www.pointworks.de/software/custommenu-3>

Lässt ein Custom Menu (und damit meine ich auch ein **echtes** Custom Menu) direkt am Mauszeiger aufspringen. Es gibt zwar vom System her das Menu mit der rechten Maustaste aber das kann man nicht wirklich customizen. Custommenu3 ist das einzige Tool dieser Art was auf aktuellen OS läuft. Es gibst zwar auch 2 oder 3 ältere Tools die sowas machen aber keines ist noch lauffähig, da schon sehr alt und nicht mehr weiterentwickelt.

Ich rufe das Menu z.b. mit der mittleren Maustaste auf. Schneller sind nur Keyboard Shortcuts.

Beitrag von „Maddeen“ vom 28. Juli 2019, 13:13

[Sascha_77](#) - das ist ja mal geil!! Super - dank dir.

Bin jetzt einfach hingegangen und habe die Datei script mit sublime editiert und gespeichert - scheint zu laufen.

Ich werde die App jetzt mal in den Autostart verfrachten und gucken, ob das Problem der Vergangenheit angehört 😊

Beitrag von „ResEdit“ vom 28. Juli 2019, 21:24

Darf ich mich mal bitte hier kurz dran hängen und um einen Tipp bzgl. eines Applescript bitten?



Ich möchte gerne für die interne Projektabrechnung mit dem Kontextmenü im Finder Textfiles in einem vorher ausgewählten Verzeichnis anlegen. Das klappt ganz rudimentär mit diesem Script:

Code

1. tell application "Finder"
2. set thisFolder to the target of the front window as alias
3. make new file at thisFolder with properties {name:"YYYY.MM.DD - HH.MM.txt", file type:"TEXT", creator type:"ttx"} }
4. end tell

"YYYY.MM.DD - HH.MM" steht dann für 2019.07.28 - 21.05.txt. Das mit dem Datum im Namen kriege ich schon mal nicht hin. 🤔

Ich schaffe es gerade, einen Namen ("test.txt") anzugeben und das klappt auch und das File wird auch ins gewählte Verzeichnis geschrieben.

ABER! Das hat 0 Byte und mir wäre lieber, es hätte "zur Sicherheit" ein paar Byte "Inhalt". Der Text in der Datei und der Name können identisch sein - also steht dann in der Datei der Text "2019.07.28 - 21.05.txt". Mit dem Schreiben in die Datei habe ich schon alles mögliche probiert und bin gescheitert. 🤔

Über den selben Script sollte noch ein weiteres File im Verzeichnis angelegt werden, das hat den Namens Kürzel des Bearbeiters (und konsequenterweise steht der auch als Text in der Datei drinnen). Dieser Name wird einmal im Script definiert und ändert sich nicht mehr.

Hintergrund ist der, dass man im Arbeitsverzeichnis später genau sehen kann, wer wann welchen Job übergeben, bzw übernommen hat. Klingt ungewöhnlich, klappt aber schon seit Jahren bei uns super. Ist allerdings eine Arbeit für Blöde, die Dateien jedes mal händisch anzulegen und die Mitarbeiter sind ganz großartig darin, sich diese Mühe zu sparen oder vergessen es einfach. Deshalb möchte ich das gerne als Script haben, den ich später als Dienste im Kontextmenü habe.

[Sascha_77](#) (oder andere Cracks hier) - ich wäre euch zu ewigem Dank verpflichtet!!!!

Beitrag von „Sascha_77“ vom 28. Juli 2019, 22:33

Hier bitte. Legt die Sachen wie gewünscht in dem **markierten** Ordner an. Datum musst Du dir dann noch nach Deiner Syntax zurechtbiegen im Workflow.

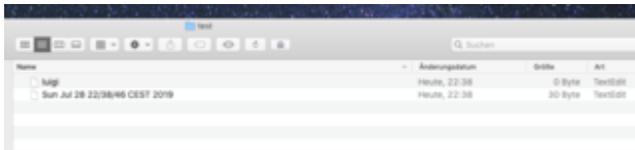
Beitrag von „ResEdit“ vom 28. Juli 2019, 22:37

Klasse - und das noch zu später Stunde, vielen Dank! Ich schau mir das morgen in Ruhe an.



Beitrag von „Sascha_77“ vom 28. Juli 2019, 22:39

Kein Ding. Sieht am Ende so aus:



Beitrag von „ResEdit“ vom 29. Juli 2019, 06:31

Hammer! Nur eine Frage noch: Bei der Ausführung ertönt immer ein Warnton, ist das normal? Mit dem Datum muss ich noch mal schauen. Auf jeden Fall ist das eine riesige Hilfe.

Hintergrund: Anhand der Files in den Arbeitsverzeichnissen räumt Hazel im Hintergrund auf, bzw. liefert Input für die Buchhaltung. Das klappt aber nur, wenn die entsprechenden Namen der Files zu 100% der Vorgabe entsprechen. Wenn z.B. einer versehentlich das Datum falsch formatiert, klappt es nicht mehr. Was müsste ich beispielsweise ändern, damit das Datum unabhängig von den Sprach- und Systemeinstellungen immer in "fest gegossener Form" ausgegeben wird?

Edit: Erledigt, habe jetzt geändert in `Datum=$(date +"%Y.%m.%d - %H.%M")`

Passt!!!!

Shell-Scripts sind ja eine richtig geile Sache!

Beitrag von „Sascha_77“ vom 29. Juli 2019, 07:25

[Zitat von ResEdit](#)

Shell-Scripts sind ja eine richtig geile Sache!

Kann man sagen. Alles was ich mir an Scripten so baue ist ausschließlich in Bash. Selbst der Kern des Kext Updaters ist reines Bash.

Beitrag von „ResEdit“ vom 29. Juli 2019, 08:22

Bin wirklich ganz hin und weg. Habe mich Jahrzehnte vor dieser Sache gedrückt. Werde das ab jetzt definitiv intensiver angehen, das Thema.

Übrigens ist die Warnmeldung weg, nachdem ich im Workflow die Kriterien definiert habe (oberhalb des Scripts im Screenshot).

Nochmals Tausend Dank!

