

Erledigt

Java-App für Lion unter Catalina laufen lassen; JVM startet nicht?

Beitrag von „LigH“ vom 9. März 2020, 09:38

Hallo.

Ein Bekannter hat eine Auerswald Telefonanlage. Diese lässt sich mittels einiger Java-Applikationen sehr komfortabel steuern. Unter Windows und unter MacOS älterer Generationen liefen diese auch noch problemlos (z.B. COMset 2.7.2 für Lion von 2011).

Seit der Umstellung auf Catalina starten die Apps nun nicht mehr. Laut meinem Wissensstand sollte es aber für Java-Applikationen relativ egal sein, auf welchem System sie laufen, solange auf dem System nur eine JRE mit kompatibler Version läuft.

Ich selbst kenne mich mit Macs auch nicht komplett aus. Mit einer Finder-Suche habe ich zunächst nichts entdeckt, was auf eine Java-Installation hindeutet, und habe deshalb ein aktuelles AdoptOpenJRE v8 installiert; nach meiner Erfahrung ist Java 8 ja noch recht kompatibel mit Programmen, die "ab Java 6" laufen sollen. Das brachte die COMset-App von Auerswald aber auch nicht zum Starten.

Dann bin ich mal in den App-Container hinein und habe mir den Inhalt angeschaut - ohne Hintergrundwissen, ob das sinnvoll ist; immerhin bekam ich so ein klein wenig mehr an verwertbaren Informationen: Beim Start eines `JavaApplicationStub` öffnete sich eine Unix-Shell, der man ein paar Fehlermeldungen entnehmen kann.

Code

1. `/Applications/Auerswald/Lion/COMset-2.7.2.app/Contents/MacOS/JavaApplicationStub ; exit;`
2. `Macintosh:~ info$ /Applications/Auerswald/Lion/COMset-2.7.2.app/Contents/MacOS/JavaApplicationStub ; exit;`

3. JavaVM: Failed to load JVM: /Library/Java/JavaVirtualMachines/adoptopenjdk-8.jre/Contents/Home/bundle/Libraries/libserver.dylib
4. JavaVM: Failed to load JVM: /Library/Java/JavaVirtualMachines/adoptopenjdk-8.jre/Contents/Home/bundle/Libraries/libserver.dylib
5. JavaVM FATAL: Failed to load the jvm library.
6. [JavaAppLauncher Error] JNI_CreateJavaVM() failed, error: -1
7. logout

Die virtuelle Maschine startet also nicht. Also sieht das eher nach einem grundlegenden Problem aus (evtl. mit AdoptOpenJRE auf dem speziellen System), weniger nach einem spezifisch für die Auerswald-Software, oder?

Bitte beachten:

Ich habe keinen täglichen Zugriff auf den betreffenden Rechner, kann also bei Nachfragen und Vorschlägen nicht umgehend was ausprobieren. Ich muss noch mal die exakte Bezeichnung erfragen...

Beitrag von „Aluveitie“ vom 9. März 2020, 09:40

Gerade bei älteren Applikationen kann es sein das diese nicht mit dem OpenJDK kompatibel sind. Probier vielleicht mal mit der Oracle JDK.

Auf meinem Hackintosh habe ich bis jetzt nur mit OpenJDK 13 gearbeitet, aber soweit keine Probleme gehabt.

Beitrag von „Veemyu“ vom 9. März 2020, 13:02

Du kanns einfach die Java OSX 1.6.0 JDK mit Pacifist unpacken und dann mit ./ und im java/bin von jdk1.6 das programm starten wie in diesem Video

<https://www.youtube.com/watch?v=D5PleVzjC7A>

Java 6 Download

https://support.apple.com/kb/DL1572?locale=en_US

Pacifist

<https://www.charlessoft.com>

Viel Spaß, sollte funktionieren.

Musst halt nur ein wenig rumwurschteln, da dass Programm leider keine Standalone .jar ist

Beitrag von „LigH“ vom 9. März 2020, 13:25

Vielleicht könnte ich als Dipl.-Inf. da rumwurschteln.

Aber der Besitzer dieses iMac ist Mac-User ohne jede Unix-Erfahrung. Bei dem muss das Klicki-bunti laufen, oder es läuft gar nicht.

Beitrag von „Veemyu“ vom 9. März 2020, 13:48

Kann ich die Software mal als Download haben und die ausprobieren?

Dann kann man ja ein Paket erstellen und ne .App so dass wieder alles geht

Weil Java 6 läuft auf Catalina Prima (Nicht als Hauptinstallation) aber seitens dran und das könnte man ja irgendwie bundlen wenn ich mehr über das Programm weiss

Beitrag von „LigH“ vom 9. März 2020, 14:13

[Service TK-Anlagen](#) - ist eine etwas veraltete Seite, da kann man bei einigen Klicks auch weggeleitet werden...

Im Abschnitt **COMmander® Basic/COMmander® Basic 19"** (etwa 1/3 runterscrollen) findet man Installationspakete für COMset, COMtools, COMlist... jeweils für Mac OS X (Snow)Leopard von 2009 und Lion von 2011. Die lassen sich installieren, wenn man nach dem Warnhinweis, dass diese Pakete nicht von einem bekannt vertrauenswürdigen Anwender signiert wurden, in den Systemeinstellungen bei Sicherheit dem gerade abgewiesenen Programm dennoch die Installation erlaubt.

Die URLs linken zwar zu *.zip-Dateien, aber Safari hatte sie mit einer anderen Endung heruntergeladen (war irgendwas mit m und 4 Buchstaben, weiß ich heute nicht mehr). Grundsätzlich sollte diese Software auch kein Sicherheitsrisiko sein: Sie kommuniziert mit einer Telefonanlage über eine serielle Schnittstelle (z.B. USB), wenn da eine erkannt wurde, ansonsten macht sie nicht viel außer Konfigurationsdateien speichern.

Beitrag von „Arkturus“ vom 9. März 2020, 18:52

32-bit apps laufen nicht mehr unter CT. Downgrade auf Mojave könnte Dir helfen.

Beitrag von „LigH“ vom 9. März 2020, 20:19

Aber das ist doch Java. Sollte Java-Bytecode nicht weitgehend plattformunabhängig sein? Abgesehen von plattformspezifischen Geräte-Bibliotheken... 🤔 Enthalten diese Apps sowas?

Beitrag von „Arkturus“ vom 9. März 2020, 20:23

welche Version darf es denn sein?



EDIT: Sorry, auch Mojave hilft nicht weiter



Beitrag von „LigH“ vom 9. März 2020, 20:55

Das ist eine tolle Antwort, [Arkturus](#) ... aber für welches Problem?

Die Software von Auerswald besteht aus compilierten Java-Klassen im Java-Bytecode, soweit ich weiß. Zum Ausführen solcher Software sollte es ziemlich egal sein, ob auf dem Mac eine Java Runtime Engine in 32-bit-Code oder in 64-bit-Code läuft, Hauptsache die JRE läuft auf diesem System. Und dann sollte die Auerswald Java-Software in dieser JRE laufen.

Eine Java-v6-Runtime wird man heute schwer auftreiben können, und wenn, kriegt man sie offenbar nur über Umwege auf einem aktuellen Mac OS X zum Laufen, so wie ich das den

obigen Links entnommen habe. Deshalb hatte ich gehofft, die Java v8 Runtime von AdoptOpenJDK würde als Ersatz funktionieren. Einfach installieren ließ sie sich ja. Aus welchen Gründen exakt nun deren Virtuelle Maschine nicht startet, entnehme ich deiner Antwort nicht.

Beitrag von „al6042“ vom 9. März 2020, 21:32

Ich hatte früher auch die dedizierte "Java for OSX"-Version auf der Kiste.

Herunterladen konnte man die von <https://support.apple.com/kb/DL1572>

Ob die sich aber noch unter Catalina installieren und nutzen lässt, weiss ich nicht.

Ansonsten sollte doch auch die [Version 8 von Oracle](#) funktionieren, oder?

Beitrag von „LigH“ vom 9. März 2020, 22:05

Seit einigen Monaten braucht man ja zum Download einer Windows-JRE von Oracle ein Oracle-Konto, da haben sich die Lizenzbedingungen geändert. Deshalb suchen ja so viele nach Alternativen, die nicht so kundenknebelnd sind. Ob das bei Oracle mit einer Mac-JRE auch so ist, weiß ich noch nicht... an den betreffenden Rechner komme ich erst wieder in einigen Wochen.

Beitrag von „MacGrummel“ vom 10. März 2020, 10:45

Java 6 ist nicht Version 8 von Oracle. Leider sind mit dem endgültigen Rauswurf von Java 6 unter Catalina (und der eigentlich fast nicht funktionierenden Implementation schon unter Mojave) durchaus einige 64-bit-Programme nicht mehr lauffähig, Photoshop ohne KLAUT z.B...

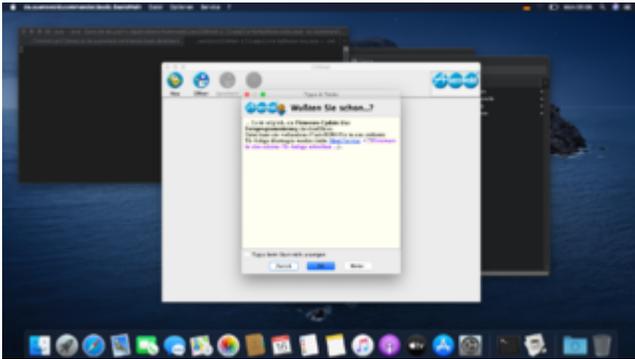
Du kannst aber auch ältere Mac-Betriebssysteme unterdessen problemlos in eine VM installieren, wenn das hilft.

Und die Lizenzbedingungen haben sich bisher nur für Geschäftskunden geändert, die sollen

richtig zahlen. Was natürlich Blödsinn ist, wenn den Privat-Kunden gleichzeitig mitgeteilt wird, dass Java bald ganz abgeschaltet wird..

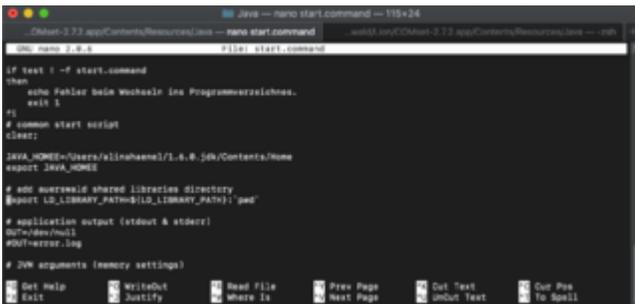
Beitrag von „Veemyu“ vom 16. März 2020, 13:01

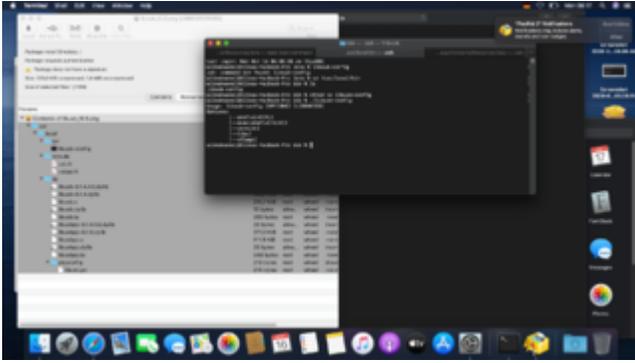
Sodele also das COMset 2.7.2 für Lion läuft prima



Und das LibUSB Package läuft auch noch auf macOS Catalina

Ich habe Java for OSX 2017 für Lion entpackt mit pacifist und dann die start.command angepasst und so auch die App gestartet wenn man das noch bundled und in ne neue App schmeißt sollte alles gehen





Beitrag von „LigH“ vom 16. März 2020, 18:46

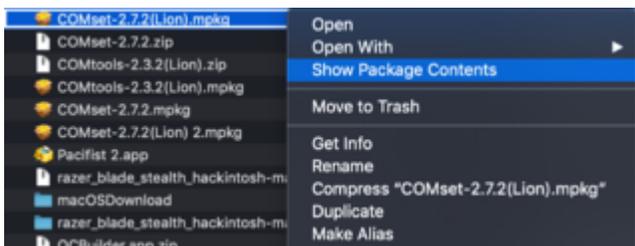


Klingt prima, Alina. Keine Hektik ... hat Zeit.

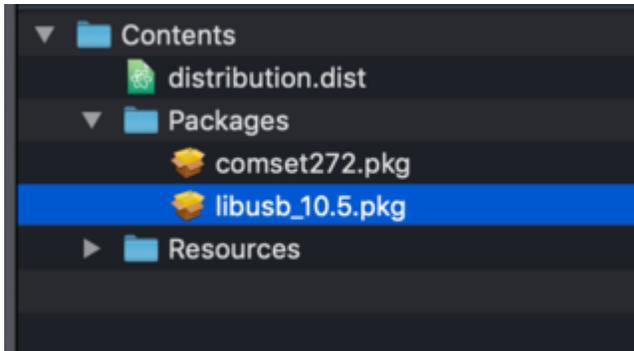
Beitrag von „Veemyu“ vom 17. März 2020, 22:44

So hiermit solltest du es hinbekommen mit meiner Tollen Bilder Galerie

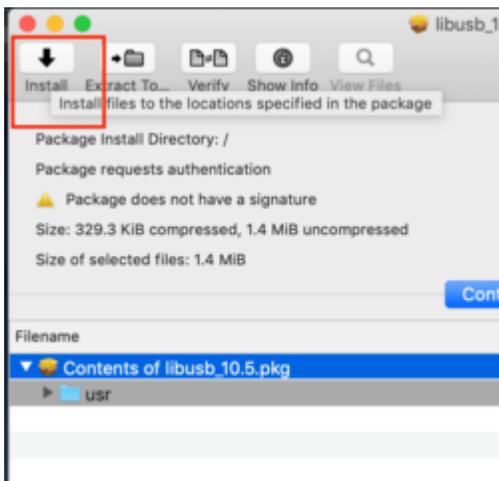
Also erstmal Auerswald Comset runter laden unzippen und bei der .mpkg mal Show Package Contents Machen



Darin befinden sich einmal die eigentlichen Comset Package die man einfach mit einem Doppelklick öffnen kann und dann installiert.



Libusb ist eine andere Geschichte das .pkg muss man mit Pacifist von <http://charlessoft.com> öffnen und dann wie hier zu sehen installieren



Jetzt nur noch ein Terminal öffnen und eingeben

Code

1. `sudo chmod +x /usr/local/bin/libusb-config`

Um LibUSB läuffähig zu machen

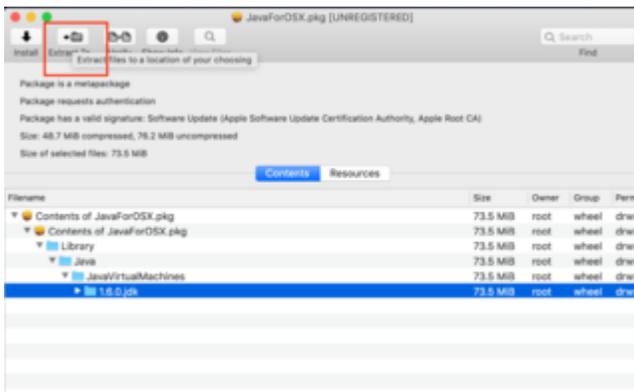
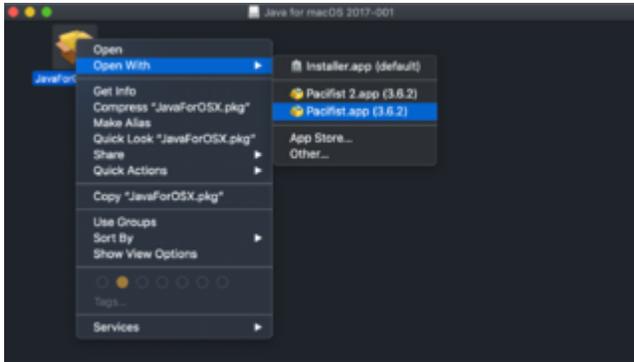
Jetzt geht es dran Java zu "installieren" bzw. irgendwo in einen Ordner zu schmeißen wie z.B. Documents oder den Home (/Users/<username>)folder

Also erstmal JavaOSX 2017 von Apple runterladen
https://support.apple.com/kb/DL1572?locale=en_US

Und dann die DMG öffnen und mounten

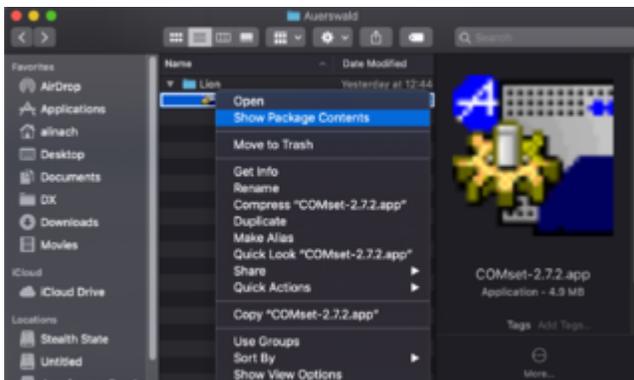
Anschließend die darin befindlich .pkg mit Pacifist öffnen bis zum 1.6.0.jdk Ordner vorschreiten

und diesen dann in z.B Documents rein Entpacken

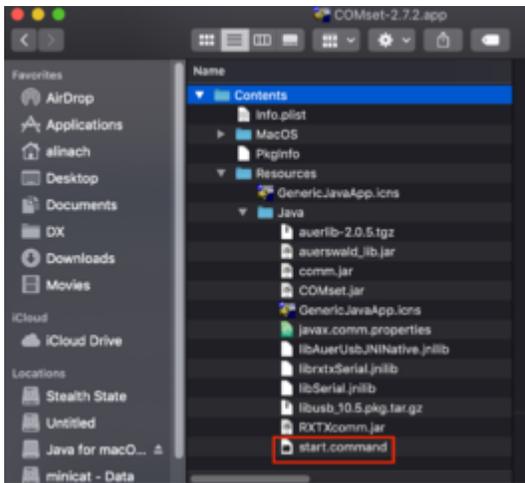


Jetzt haben wir "Java installiert" haben den Auerswald Ordner in /Applications und libusb installiert

Jetzt müssen wir in /Applications/Auerswald im Finder und dann Show Package Contents



Und schliesslich Contents/Resources/Java



Jetzt sind wir auch bei der start.command angekommen

Welche wir jetzt Modifizieren werden

Erstmal mit nem guten Text Editor öffnen

Atom oder Xcode. (nano würde auch gehen) TextEdit zerstört die Datei wahrscheinlich

```
start.command
start.command > No Selection

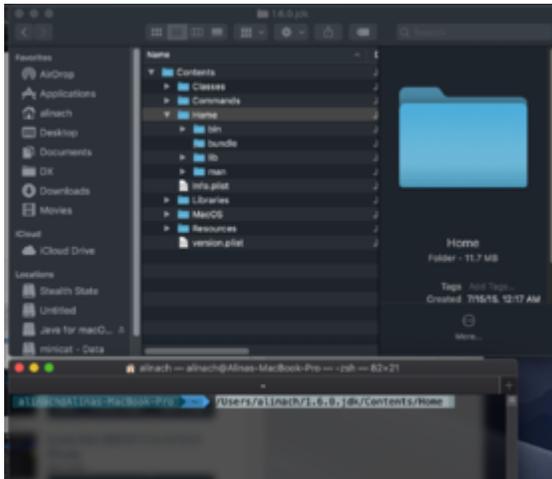
6 # local classpath
7 LOCAL_CLASSPATH=auerswald_lib.jar
8 LOCAL_CLASSPATH=$LOCAL_CLASSPATH:comm.jar
9 LOCAL_CLASSPATH=$LOCAL_CLASSPATH:RXTXcomm.jar
10 LOCAL_CLASSPATH=$LOCAL_CLASSPATH:COMset.jar
11
12 # java main class
13 MAIN_JAVA_CLASS=de.auerswald.commander.basic.BasicMain
14
15 # change to application directory
16 cd `dirname $0`
17
18 # check for correct current directory
19 if test ! -f start.command
20 then
21     echo Fehler beim Wechseln ins Programmverzeichnis.
22     exit 1
23 fi
24 # common start script
25 clear;
26
27 JAVA_HOME=/System/Library/Frameworks/JavaVM.framework/Home
28 export JAVA_HOME
29
30 # add auerswald shared libraries directory
31 export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:`pwd`
32
33 # application output (stdout & stderr)
34 OUT=/dev/null
35 #OUT=error.log
36
37 # JVM arguments (memory settings)
38 #JVM_ARGS="-Xms32M -Xmx64M"
39 JVM_ARGS=-Djava.library.path=$LD_LIBRARY_PATH
40 #JVM_ARGS=
41
42 # start the application
43 exec $JAVA_HOME/bin/java $JVM_ARGS -cp $LOCAL_CLASSPATH \
44     $MAIN_JAVA_CLASS > $OUT 2>&1
45
```

Und dann es so abändern

```
start.command
start.command > No Selection
1  #! /bin/sh
2  #
3  # Startup script for Auerswald Application
4  #
5
6  # local classpath
7  LOCAL_CLASSPATH=auerswald_lib.jar
8  LOCAL_CLASSPATH=$LOCAL_CLASSPATH:comm.jar
9  LOCAL_CLASSPATH=$LOCAL_CLASSPATH:RXTXcomm.jar
10 LOCAL_CLASSPATH=$LOCAL_CLASSPATH:COMset.jar
11
12 # java main class
13 MAIN_JAVA_CLASS=de.auerswald.commander.basic.BasicMain
14
15 # change to application directory
16 cd `dirname $0`
17
18 # check for correct current directory
19 if test ! -f start.command
20 then
21     echo Fehler beim Wechseln ins Programmverzeichnis.
22     exit 1
23 fi
24 # common start script
25 clear;
26
27 JAVA_HOME=/Users/alinach/1.6.0.jdk/Contents/Home
28 export JAVA_HOME
29
30 # add auerswald shared libraries directory
31 export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:'pwd'
32
33 # application output (stdout & stderr)
34 OUT=/dev/null
35 #OUT=error.log
36
37 # JVM arguments (memory settings)
38 #JVM_ARGS="-Xms32M -Xmx64M"
39 JVM_ARGS=-Djava.library.path=$LD_LIBRARY_PATH
40 #JVM_ARGS=
41
42 # start the application
43 exec $JAVA_HOME/bin/java $JVM_ARGS -cp $LOCAL_CLASSPATH \
44     $MAIN_JAVA_CLASS > $OUT 2>&1
45
```

Alles wo JAVA_HOME steht bekommt ein E dazu spendiert, einfach damit die JAVA_HOME Environment Variable nicht einfach so überschrieben wird und dann eine möglich Java 8 / 10 Installation zerschiesst

Den JAVA_HOME= Ordner Pfad finden wir folgender maßen heraus wir öffnen den exportierten 1.6.0.jdk ordner gehen in Contents/Home und ziehen den Home Ordner in ein offenes Terminal somit erhalten wir den Pfad den wir dann einfach in die start.command reinpasten können



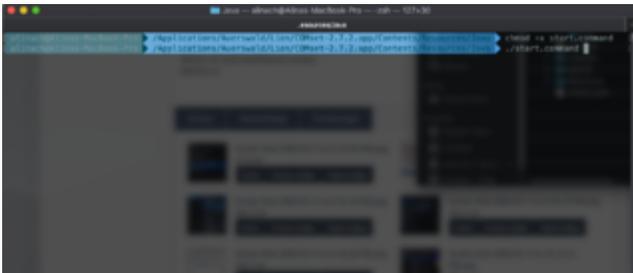
Jetzt haben wir alles fertig.

Jetzt nur noch mit dem Terminal in den Comset Java Ordner

Code

1. `cd /Applications/Auerswald/Lion/COMset-2.7.2.app/Contents/Resources/Java`

Dann noch schnell die start.command ausführbar machen mit:



Und dann mal im Selben Terminal anschließend

Code

1. `./start.command`

Und tada das programm geht. Zumindest nach augen scheinen



Und deswegen wäre es toll wenn diese Methode mal ausprobiert werden könnte

Bevor ich es ver-user-friendlye

[LigH](#)

Gutes gelingen, lass von dir hören!

Beitrag von „LigH“ vom 18. März 2020, 22:05

Ich danke vielmals für die ausführliche Anleitung.

Das Testen wird etwas dauern, wie gesagt, an diesen Rechner komme ich nur mal alle paar Wochen ...

Beitrag von „Veemyu“ vom 19. März 2020, 20:48

Alles gut! Nur bevor ich das alles in einen Easy installer oder DMG schmeiße sollte es vorher einfach getestet werden ob es so schon funktioniert, oder halt nicht.

Wann das geschieht ist wurscht 😜