

Musterlösung für
Schulen in
Baden-Württemberg

paedML® Windows



Opsi-Paketerstellung Teil 2

Paketerstellung mit AutoIT

Netzwerke/Windows

Juni 2016

Lizenz: CC-BY-NC-SA



Impressum

Herausgeber

Zentrale Konzeptionsgruppe Netze (ZKN)

an der Landesakademie für Lehrerfortbildung und Personalentwicklung an Schulen

Autoren

Adrian Koch

Johannes Kühn

Martin Resch

Endredaktion

Adrian Koch

Weitere Informationen

<http://www.lehrerfortbildung-bw.de/netz/>

Veröffentlicht

2016, 1. Fassung

INHALTSVERZEICHNIS

1. OPSI-Paketerstellung Teil 2.....	1
1.1. Allgemeines.....	1
1.2. Paketerstellung mit AutoIT.....	2
1.2.1. Das Rohpaket VorlageAutoIT.....	3
1.2.2. Übung 1: PC_Extern vorbereiten.....	4
1.2.3. Übung 2: VeraCrypt „händisch“ installieren und deinstallieren.....	6
1.2.4. Übung 3: Opsi-Rohpaket für die Installation von VeraCrypt mit AutoIT vorbereiten.....	6
1.2.5. Übung 3: Funktionstest.....	9
1.2.6. Übung 3: Produkt-ID und Name anpassen.....	9
1.2.7. Übung 4: Angepasste Installation am Beispiel Everything.....	10
2. Links und Quellen.....	1

1. OPSI-PAKETERSTELLUNG TEIL 2

1.1. Allgemeines

Diese Anleitung zeigt Ihnen, wie Sie opsi-Pakete mit Hilfe von autoit erstellen können. Der Vorgang gliedert sich grob in drei Schritte.

1. Sie installieren ein Programm händisch und beobachten den Vorgang und die notwendigen Tastatureingaben.
2. Anhand dieser Notizen und einer Rohpaketvorlage erstellen Sie die Skripte für Installation und Deinstallation. Mit opsi-WinInst testen Sie, ob sich das Paket installieren und deinstallieren lässt.
3. Sie transportieren das Paket zum AdminPC. Mit dem opsi-PakageBuilder packen Sie das Rohpaket zu einem opsi-Paket und hinterlegen es auf dem Opsi-Server, so dass es den Clients in der paedML schnell zugewiesen werden kann.

Hintergründe

Ein *Rohpaket* besteht aus einem Ordner, der auf einem beliebigen Rechner erstellt werden kann und dann meist auf dem Opsiserver in der Workbench abgelegt ist. Er enthält zwei Unterordner, für die Installationsdateien (samt Installationsskript) und für Opsiinformationen wie z.B. die Versionsnummer.

Name	Größe	Typ
CLIENT_DATA		Dateiordner
OPSI		Dateiordner
shockwave12_2_12.2.0.162-1.opsi	23.437 KB	OPSI-Datei

In dieser Form kann man das Paket bearbeiten, z.B. Dateien hinzufügen oder die Skripte mit einem Texteditor verändern.

Die eigentliche .opsi-Datei (Paket) ist ein komprimiertes Archiv, ähnlich wie eine .zip-Datei. Nur in dieser Form kann das Programm in die Opsi-Datenbank übertragen werden.

PackageBuilder ist Hilfsprogramm, das für den nicht so versierten Anwender, der unter Windows arbeiten möchte, zur Verfügung steht. Es erlaubt eine einfache Erstellung von Opsi-Paketen ohne sich in die Linux Kommandozeilenbefehle einarbeiten zu müssen.

WinInst ist der Skriptinterpreter von Opsi auf den Windowsclients. Er bringt eine grafische Oberfläche zum Testen und Debuggen mit sich.

Wünschenswert wäre es, wenn Ihnen dazu ein PC zur Verfügung steht, den Sie immer wieder auf einen definierten Zustand zurücksetzen können. Denkbar wäre

- eine zusätzliche virtuelle Maschine im ESXi
- eine virtuelle Maschine in VMWare Workstation (kostenpflichtig) VirtualBox oder hyper-V (freeware). Auf diese Weise könnten Sie prinzipiell Ihre Pakete zu Hause erstellen.
- ein realer Rechner, der mit opsi-local-image-restore immer wieder zurück gesetzt wird.

In der Schulungsumgebung finden Sie einen zusätzlichen Client vor, den PC-Extern. Dieser PC ist nicht ins Netzwerk eingebunden. In dieser Anleitung erfahren Sie, wie Sie diesen Client zur

Erstellung von Opsi-Paketen nutzen können.

Notwendige Dateien:

- *OpsiDetector 4.0.6.132.msi*
- *OpsiWinInst 4.11.5.14.msi*
- *Start-hta.cmd*
- *Deinstallationsroutine auslesen.html*
- *VorlageAutoIt.zip*
- *VeraCrypt Setup 1.19* (für unten aufgeführte Übung)
- *vlc-2.2.1-win32.exe* (für eventuell weitere Übung)

1.2. Paketerstellung mit AutoIT

Es gibt immer wieder Softwareprodukte, die keine unbeaufsichtigte Installation anbieten und für die somit eine Paketerstellung mit dem OpsiDectector nicht möglich ist. In diesem Fall müsste man wieder die aus der paedML 2.x bekannte Lösung mit LMZ-Repacker ein MSI-Paket erstellen und dieses mit dem OpsiDectector von MSI in ein opsi-Paket umwandeln. Diese Methode wird aber auch nicht in allen Fällen erfolgreich sein. Insbesondere Programme die hardwarenah arbeiten.

Eine weitere Möglichkeit ist das Aufzeichnen der Maus- und Tastatureingaben während der Installation und Deinstallation einer Software, die dann später automatisiert ausgeführt werden können. Somit kann jede Software installiert werden, die sich auch händisch installierten lässt. Für diese Automatismen bietet sich die Script-Sprache AutoIT www.autoitscript.com an, die sich in opsi-Scripte einbinden lässt. Aus dem umfangreichen AutoIT-Paket wird später lediglich der Batch-Interpreter autoit3.exe verwendet. Die wichtigsten AutoIT-Befehle lauten wie folgt:

Befehl	Beschreibung
WinWait("VeraCrypt Setup 1.19")	Wartet bis das Fenster mit dem angegeben Text in der Titelleiste erscheint
Sleep(2000)	Wartet 2 Sekunden
Send("n")	Betätigt die Taste n (oft für N ext) = weiter)
Send("{TAB}")	Betätigt die TAB-Taste
Send("{UP}")	Betätigt die nach-oben-Taste
Send("{DOWN}")	Betätigt die nach-unten-Taste
Send("{ENTER}")	Betätigt die Eingabetaste (oft für weiter)
Send("{}")	Entfernt Haken in einer Selectbox
Send("{+}")	Setzt Haken in einer Selectbox
WinWait("", "successfully installed.")	Wartet auf ein Fenster, welches im Fenstertext den Text "successfully installed." enthält. Hinweis: Die Titelleiste wird nicht ausgewertet
BlockInput(1)	Eingabe Maus und Tastatur sperren
BlockInput(0)	Eingabe Maus und Tastatur zulassen

Weitere Send-Befehle finden sich hier: <https://autoit.de/onlinehilfe/functions/Send.htm>

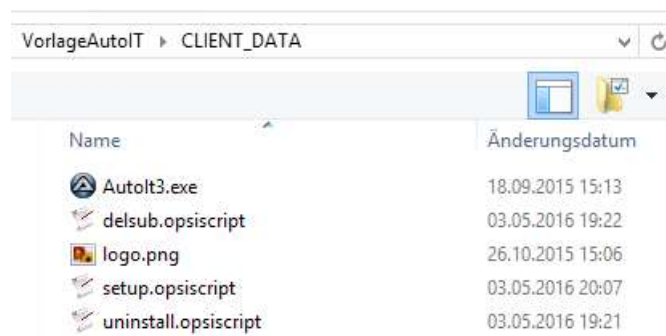
Die Anwendung von AutoIT in opsi-Scripten soll nun an der Problemsoftware „VeraCrypt“ gezeigt werden, die mit den bisherigen Methoden nicht paketierte werden kann.

1.2.1. Das Rohpaket VorlageAutoIT

Das Rohpaket VorlageAutoIT liegt als ZIP-Paket vor und beinhaltet die beiden Ordner CLIENT_DATA und OPSI.

Name	Änderungsdatum	Typ
CLIENT_DATA	12.11.2016 19:26	Dateiordner
OPSI	12.11.2016 19:26	Dateiordner

Der Ordner CLIENT_DATA:



Name	Änderungsdatum
Autolt3.exe	18.09.2015 15:13
delsub.opsiscript	03.05.2016 19:22
logo.png	26.10.2015 15:06
setup.opsiscript	03.05.2016 20:07
uninstall.opsiscript	03.05.2016 19:21

Neben dem Programm Autolt3.exe sind hier die drei Vorlageskriptdateien setup.opsiscript, uninstall.opsiscript und delsub.opsiscript.

```
setup.opsiscript
1  [Actions]
2
3  Sub "%ScriptPath%\delsub.opsiscript"
4
5  ShowBitmap "%ScriptPath%\logo.png" "Installiere xxxxxx"
6
7  ExecWith_autoit "%SCRIPTPATH%\autoit3.exe" WINST /letThemGo
8  winbatch_setup_aufrufen /Sysnative
9  killtask "autoit3.exe"
10
11 [winbatch_setup_aufrufen]
12 "%SCRIPTPATH%\xxxxxx.exe"
13
14 [ExecWith_autoit]
15 ;WinWait("Fenstertitel")
16 ;Sleep(2000)
17 ;Send("n")
18 ;Send("{ENTER}")
19 ;WinWait("", "Fensterstext")
20 Exit
```

in Zeile 3 der Setup-Routine wird zunächst die Uninstall-Routine aufgerufen, um vor der Installation eine mögliche Vorversion zu deinstallieren.

Anpassungen des Setup-Skripts erfolgen anhand des unter 1.2.4 folgenden Übungsbeispiels in den gelb unterlegten Bereichen.

```
delsub.opsiscript
1 ShowBitmap "%ScriptPath%\logo.png" "Deinstalliere xxxxxx"
2
3 if FileExists("xxxxx")
4
5     ExecWith_autoit2 "%SCRIPTPATH%\autoit3.exe" WINST /letThemGo
6     ; /WaitforProcessending "Au_.exe" notwendig bei nsis
7     winbatch_setup_aufrufen /Sysnative
8     killtask "autoit3.exe"
9
10 endif
11
12 [winbatch_setup_aufrufen]
13 xxxxxx
14
15 [ExecWith_autoit2]
16 ;WinWait("Fenstertitel")
17 ;Sleep(2000)
18 ;Send("n")
19 ;Send("{ENTER}")
20 ;WinWait("", "Fensterstext")
21 Exit
```

Anpassungen des Delsub-Skripts erfolgen anhand des unter 1.2.4 folgenden Übungsbeispiels.

```
uninstall.opsiscript
1 [Actions]
2
3 Sub "%ScriptPath%\delsub.opsiscript"
```

Das Uninstall-Skript bleibt unverändert. Er ruft lediglich die Delsub-Routine auf.

Das Bild logo.png kann durch das Logo des jeweils zu installierenden Programms ausgetauscht werden (Größe ca. 160x160 Pixel).

1.2.2. Übung 1: PC_Extern vorbereiten

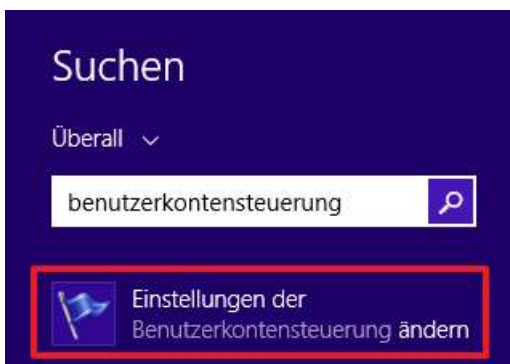
Für die Paketerstellung wird in der Schulungsumgebung der PC_Extern verwendet, der entsprechend vorbereitet werden muss.

1. Melden Sie sich mit dem bereits eingerichtete lokalen Administrator *Chef* und dem Kennwort *muster* am PC_Extern an.
2. Kopieren Sie die folgenden Programme auf den Desktop der virtuellen Maschine:
 - Deinstallationsroutine auslesen
 - OpsiWinInst 4.11.5.14.msi
 - Simple ScriptEditor 3.0.msi
3. Installieren Sie die beiden Programme OpsiWinInst 4.11.5.14.msi u. Simple ScriptEditor 3.0.msi durch Doppelklick
4. Löschen Sie nach der Installation die beiden Setup-Dateien.
5. Legen Sie den Ordner [C:\opsi-Pakete](#) an.

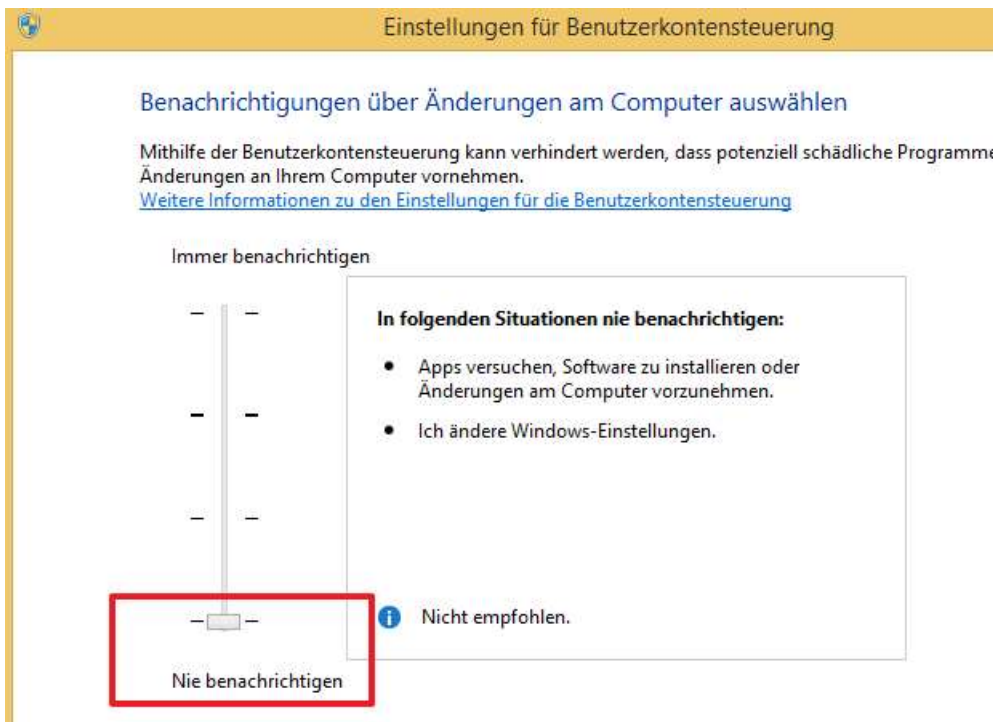
6. In der paedML 3.0 ist an den Clients die Benutzerkontensteuerung abgeschaltet. Diese muss auch beim PC_Extern deaktiviert werden. Klicken Sie mit der rechten Maustaste auf das Windows-Startsymbol und wählen Sie „Suchen“ aus.



7. Geben Sie als Suchbegriff „benutzerkontensteuerung“ ein.

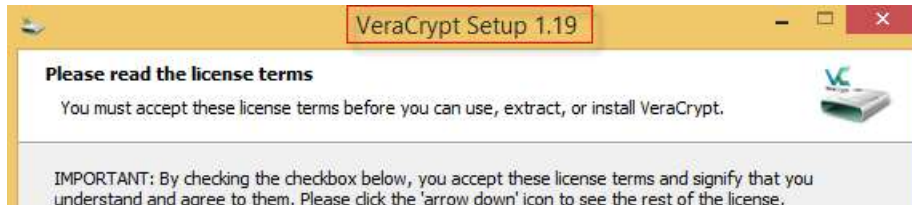


8. Verschieben Sie den Regler nach ganz unten und bestätigen Sie mit „OK“.



1.2.3. Übung 2: VeraCrypt „händisch“ installieren und deinstallieren

1. Kopieren Sie die Setup-Datei von VeraCrypt in den PC_Extern.
2. Starten Sie das Setup.
3. Notieren Sie sich den vollständigen Text in der Titelleiste des Setup-Assistenten.

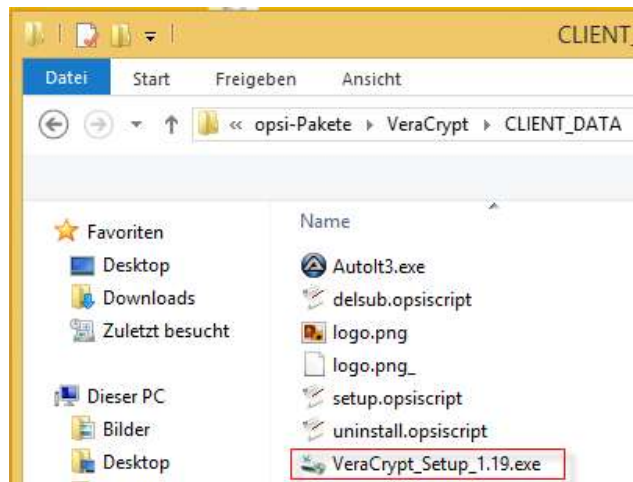


4. Betätigen Sie die „Alt“-Taste. Durch diese Aktion werden die meisten Setup-Assistenten auf den Tastatur-Betrieb umgeschaltet. Die unterstrichenen Buchstaben zeigen die möglichen Tastatureingaben.
5. Notieren Sie sich die Tastaturklicks während des Setup-Assistenten (z. B. n für Next = weiter, alternativ geht auch Enter als weiter).
6. Am Ende des Setup-Assistenten beginnt die Installation, die immer etwas dauern kann. Notieren Sie sich einen Textteil aus dem anschließend erscheinenden Fenster. Hier wird es der Text "successfully installed" sein.
7. Notieren Sie sich die Tastatureingaben die anschließend noch folgen.
8. Wiederholen Sie die Schritte für die Deinstallation.
9. Die Notizen sollten dann wie folgt aussehen:

Notizen Installation	Notizen Deinstallation
Titel "VeraCrypt Setup 1.19" a n (alternativ Enter) n r i Fenstertext "successfully installed" Enter f n	Titel "VeraCrypt Setup" r u Fenstertext "successfully uninstalled" Enter f

1.2.4. Übung 3: Opsi-Rohpaket für die Installation von VeraCrypt mit AutoIT vorbereiten

1. Kopieren Sie die Datei VorlageAutoIT.zip auf den Desktop des PC_Extern.
2. Entpacken Sie diese Datei im Ordner [c:\opsi-Pakete](#).
3. Benennen Sie den Ordner C:\opsi-Pakete\VorlageAutoIT in C:\opsi-Pakete\VeraCrypt um.
4. Wechseln Sie in den Ordner C:\opsi-Pakete\VeraCrypt\CLIENT_DATA und kopieren Sie in diesen Ordner die Setup-Datei von VeraCrypt



5. Doppelklicken Sie auf die Datei „setup.opsiscript“ und nehmen Sie die gelb hervorgehobenen Änderungen vor. Die Befehle für AutoIT entsprechen den Notizen bei der Installation. Wichtig ist, dass vor jedem Befehl eine Verzögerungszeit von mindestens 2 Sekunden eingebaut wird. Speichern Sie diese Datei nach der Änderung ab.

```
[Actions]
```

```
Sub "%ScriptPath%\delsub.opsiscript"
```

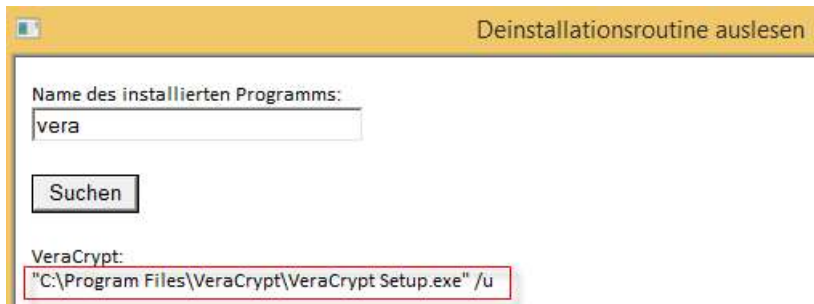
```
ShowBitmap "%ScriptPath%\logo.png" "Installiere VeraCrypt"
```

```
ExecWith_autoit "%SCRIPTPATH%\autoit3.exe" WINST /letThemGo  
winbatch_setup_aufrufen /Sysnative  
killtask "autoit3.exe"
```

```
[winbatch_setup_aufrufen]  
"%SCRIPTPATH%\VeraCrypt_Setup_1.19.exe"
```

```
[ExecWith_autoit]  
WinWait("VeraCrypt Setup 1.19")  
Sleep(2000)  
Send("a")  
Sleep(2000)  
Send("n") 9  
Sleep(2000)  
Send("n")  
Sleep(2000)  
Send("r")  
Sleep(2000)  
Send("i")  
Sleep(2000)  
WinWait("", "successfully installed.")  
Sleep(2000)  
Send("{ENTER}")  
Sleep(2000)  
Send("f")  
Sleep(2000)  
Send("n")  
Exit
```

- Doppelklicken Sie auf die Datei „delsub.opsiscript“ und nehmen Sie die gelb hervorgehobenen Änderungen vor. Die Programmzeile für die Deinstallation erhalten Sie mit dem Tool „Deinstallationsroutine auslesen\Start-hta“ und der Eingabe von einem Teil des Programmnamens.



Die Befehle für AutoIT entsprechen den Notizen bei der Deinstallation. Wichtig ist, dass vor jedem Befehl eine Verzögerungszeit von mindestens 2 Sekunden eingebaut wird. Speichern Sie diese Datei nach der Änderung ab.

```
ShowBitmap "%ScriptPath%\logo.png" "Deinstalliere VeraCrypt"

if FileExists("C:\Program Files\VeraCrypt\VeraCrypt Setup.exe")

    ExecWith_autoit2 "%SCRIPTPATH%\autoit3.exe" WINST /letThemGo
    ; /WaitforProcessending "Au_.exe" notwendig bei nsis
    winbatch_setup_aufrufen /Sysnative
    killtask "autoit3.exe"

endif

[winbatch_setup_aufrufen]
"C:\Program Files\VeraCrypt\VeraCrypt Setup.exe" /u

[ExecWith_autoit2]
WinWait("VeraCrypt Setup")
Sleep(2000)
Send("r")
Sleep(2000)
Send("u")
Sleep(2000)
WinWait("", "successfully uninstalled")
Sleep(2000)
Send("{ENTER}")
Sleep(2000)
Send("f")
Exit
```

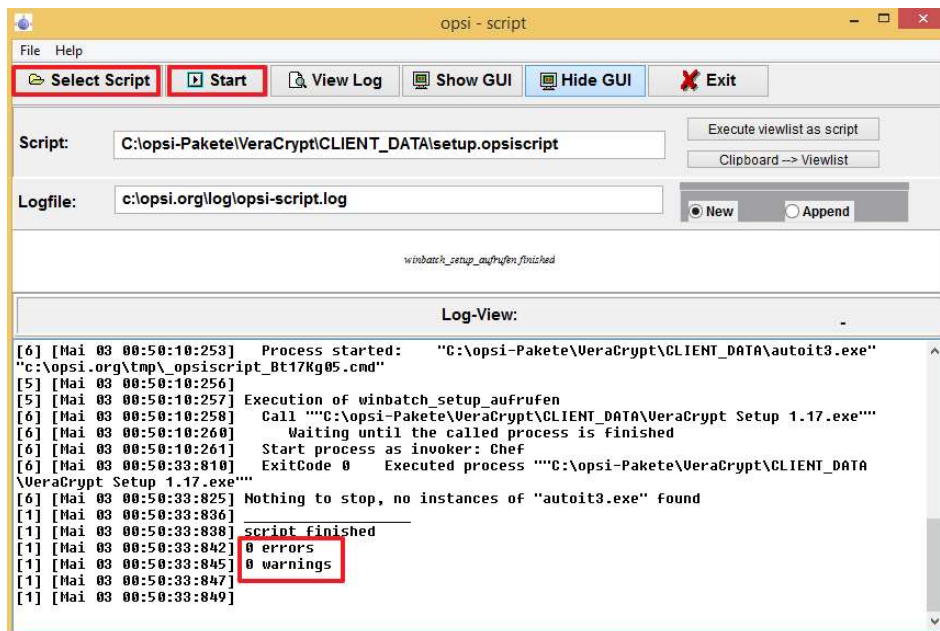
Das für die Deinstallation verantwortliche Skript *uninstall.opsiscript* beinhaltet lediglich die beiden Zeilen

```
[Actions]

Sub "%ScriptPath%\delsub.opsiscript"
```

1.2.5. Übung 3: Funktionstest

1. Doppelklicken Sie auf das Desktop-Symbol „Opsi-Wininst“.
2. Wählen Sie das Setup-Script aus und klicken Sie auf „Start“. Betätigen Sie während der Ausführung nicht die Tastatur oder die Maus. Beachten Sie auch, dass durch die Einbindung von AutoIT in opsi-Skripte nicht alle Klicks während der Installation angezeigt werden. Warten Sie auch, bis die Meldung „0 errors, 0 warnings“ angezeigt wird.



3. Wiederholen Sie die Schritte für das uninstall-Script.

1.2.6. Übung 3: Produkt-ID und Name anpassen

In den bisherigen Beschreibungen wurde die Produkt-ID und der Paketname erst im opsi-PackageBuilder festgelegt. Hier wird gezeigt, dass dies bereits im opsi-Rohpaket möglich ist und somit die spätere Anpassung entfallen kann. Gehen Sie wie folgt vor:

1. Beenden Sie den ScriptEditor (falls gestartet).
2. Starten Sie den ScriptEditor.
3. Öffnen Sie mit diesem die Datei C:\opsi-Pakete\VeraCrypt\OPSI\control und nehmen Sie die gelb hervorgehobenen Änderungen vor. Bitte beachten Sie, dass bei der Produkt-ID nur Kleinschrift und keine Umlaute zugelassen sind. Speichern Sie diese Datei nach der Änderung ab.

```
[Package]
version: 1
depends:
incremental: False
```

```
[Product]
type: localboot
id: veracrypt
name: VeraCrypt
```

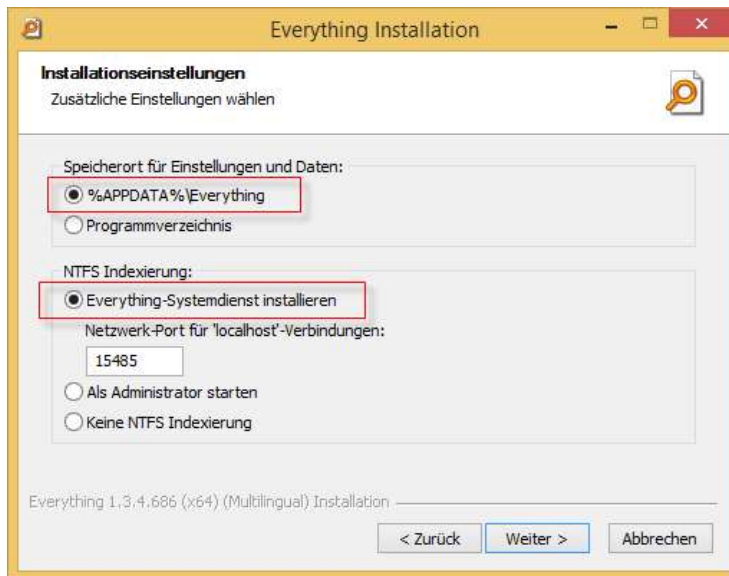
```
description:  
version: 1.19  
setupScript: setup.opsiscript  
uninstallScript: uninstall.opsiscript
```

1.2.7. Übung 4: Angepasste Installation am Beispiel Everything

1. Das Programm Everything finden Sie in den bereitgestellten Materialien.

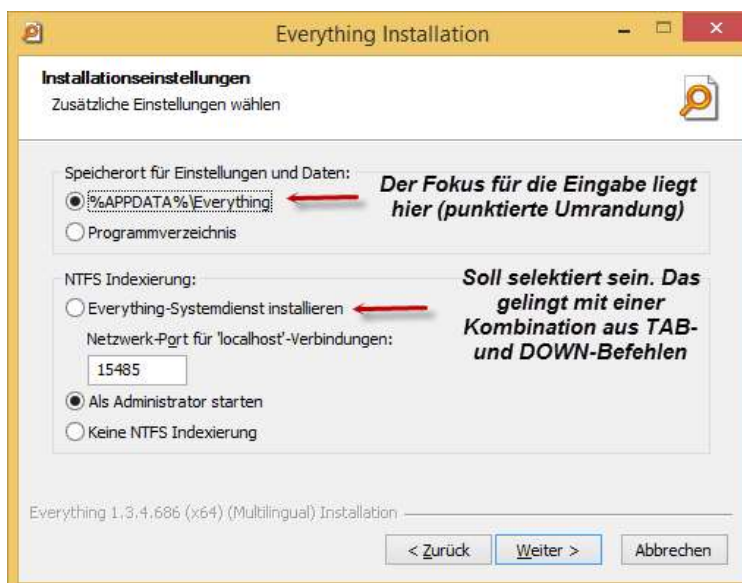
Führen Sie wie unter 1.2.3 gezeigt eine „händische“ Installation und Deinstallation durch und notieren Sie sich die Tastenfolge.

Bei der Installation sollen folgende Anpassungen vorgenommen werden.



Dies obige Auswahl ist standardmäßig in der händischen Installation getroffen. Man könnte also direkt mit ‚w‘ weiterklicken.

Beim Test des AutoIT-Setupskripts bekommt man aber folgende Einstellung:





2. Bereiten Sie eine Kopie des Rohpakets VorlageAutoIT entsprechend dem Punkt 1.2.4 für die Aufnahme des Installationsprogramms von Everything vor.
3. Nehmen Sie entsprechend Ihrer Aufzeichnungen die notwendigen Änderungen an den Skripten *setup.opsiscript* und *delsub.opsiscript* vor.

```
1 [Actions]
2 Sub "%ScriptPath%\delsub.opsiscript" /waitseconds 10
3
4 ShowBitmap "%ScriptPath%\logo.png" "Installiere Everything"
5
6 ExecWith_autoit "%SCRIPTPATH%\autoit3.exe" WINST /letThemGo
7 winbatch_setup_aufrufen /Sysnative
8 killtask "autoit3.exe"
9 |
10 [winbatch_setup_aufrufen]
11 "%SCRIPTPATH%\Everything-1.3.4.686.x64.exe"
12
13
14 [ExecWith_autoit]
15 BlockInput (1) ← Tastatur und Maus ausschalten
16 WinWait("Everything Installation")
17 Sleep(2000)
18 Send("{ENTER}")
19 Sleep(2000)
20 Send("{ENTER}")
21 Sleep(2000)
22 Send("{TAB}")
23 Sleep(2000)
24 Send("w") ← weiterschalten zum Fenster
    Installationseinstellungen
```

```
25 Sleep(2000)
26 Send("{TAB}")
27 Sleep(2000)
28 Send("{TAB}")
29 Sleep(2000)
30 Send("{TAB}")
31 Sleep(2000)
32 Send("{TAB}")
33 Sleep(2000)
34 Send("{DOWN}")
35 Sleep(2000)
36 Send("{DOWN}")
37 Sleep(2000)
38 Send("w")
39 Sleep(2000)
40 Send("k")
41 Sleep(2000)
42 Send("i")
43 Sleep(3000)
44 Send("{-}")
45 Sleep(2000)
46 Send("f")
47 BlockInput(0)
48 ;Send("{ENTER}")
49 ;WinWait("", "Fenster text")
50 Exit
```

in diesem Block wird die gewünschte Einstellung vorgenommen

Tastatur und Maus wieder einschalten

Würde man die Tastatur- und Mausfunktion nicht ausschalten, würde während der Installation der Fokus auf das Installationsfenster verloren gehen und die Installation scheitern.

4. Führen Sie entsprechend 1.2.5 einen Funktionstest durch.

```
delsub.opsiscript
1 ShowBitmap "%ScriptPath%\logo.png" "Deinstalliere Everything"
2
3 if FileExists("C:\Program Files\Everything\Everything.exe")
4
5   ExecWith_autoit "%SCRIPTPATH%\autoit3.exe" WINST /LetThemGo
6   ;/WaitforProcessending "Au_.exe" notwendig bei nsis
7
8   winbatch_setup_aufrufen /Sysnative /WaitSeconds 10
9   killtask "autoit3.exe"
10
11 endif
12
13 [winbatch_setup_aufrufen]
14 "C:\Program Files\Everything\Uninstall.exe"
15
16 [ExecWith_autoit]
17 BlockInput(1)
18 WinWait("Everything Installation")
19 Sleep(2000)
20 Send("{ENTER}")
21 Sleep(2000)
22 Send("d")
23 Sleep(2000)
24 Send("f")
25 BlockInput(0)
26 ;Send("{ENTER}")
27 ;WinWait("", "Fenstertext")
28 Exit
```

ist notwendig, weil sonst AutoIT fertig ist, bevor das Uninstallprogramm fertig ist.

5. Die Deinstallation klappt nur, wenn man in Zeile 8 den Schalter WaitSeconds <Anzahl Sekunden> verwendet
6. Abschließend sollten Sie noch entsprechend 1.2.6 Produkt-ID, Namen und Versionsnummer anpassen.

Mögliche weitere Übungen mit Programmen, die sich mit den bisherigen Techniken nicht repacketieren lassen:

- Lego Mindstorm wäre schön, jedoch geht eine Silentinstallation
- eventuell eine angepasste Installation des vlc-Players



2. LINKS UND QUELLEN

AutoIT-Onlinedokumentation: <https://autoit.de/onlinehilfe/>

AutoIT-Funktionsreferenz (deutsch): <https://autoit.de/onlinehilfe/functions.htm>

AutoIT-Sendbefehle (deutsch): <https://autoit.de/onlinehilfe/functions/Send.htm>