

VCDScripter Handbuch (Version 1.2)

Autor: hadez16, 07.08.19

Kontakt: vcdscripiter@vcds.de

Inhalt

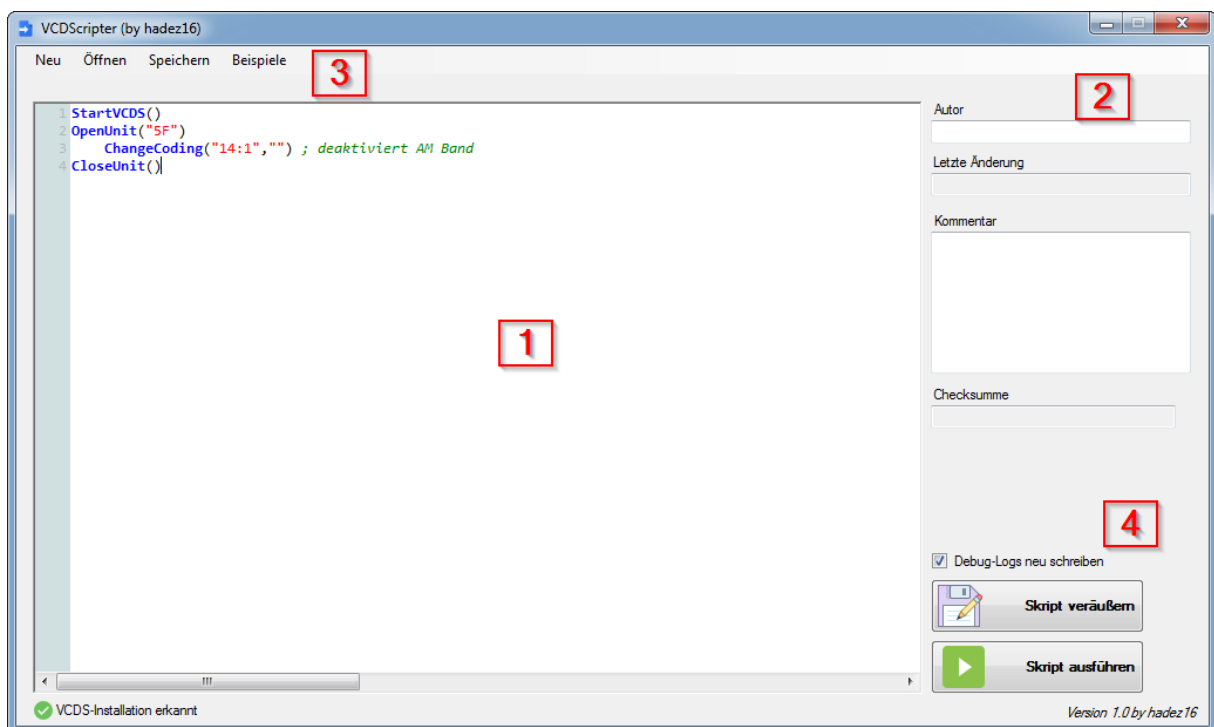
VCDScripiter Handbuch (Version 1.1)	1
1) Einleitung	2
2) Überblick	2
3) Skript-Befehle	3
StartVCDS()	3
CreateAdpMap("...", "...")	3
CreateHexAdpMap("...", "...")	4
CreateAdpMapDump()	4
CreateHexAdpMapDump()	4
CreateAutoscan("...")	4
CheckForUnits("...", 0/1)	5
OpenUnit("...")	5
TestComponent("...", 0/1)	6
TestPartNo("...", 0/1)	6
TestAdpChannel("...", "...", 0/1)	6
ClearDTCs()	7
LoginToUnit("...")	7
ChangeCoding("...", "...")	7
ChangeCodingInSub("...", "...", "...")	8
ChangeAdpChannel("...", "...")	8
CloseUnit()	10
ClearAllDTCs()	10
Gültigkeitsbereiche der Befehle	10
Kommentare	10
4) Speichern & Versiegeln eines Skripts	11
5) Skripte veräußern	12
6) Skripte starten	12
7) Limitationen und Hinweise	13
a) Limitationen	13
b) Fehlerbehandlung	13
8) Beispiel Anhängerkupplung MQB Facelift	14

1) Einleitung

Die Applikation "VCDScripter" ist ein Werkzeug zur Automation von Vorgängen innerhalb von Ross-Tech (LLC) VCDS[®] (nachfolgend "VCDS"). Zu diesem Zweck wird eine eigene Skript-Sprache verwendet, die Schritte innerhalb der Applikation automatisiert.

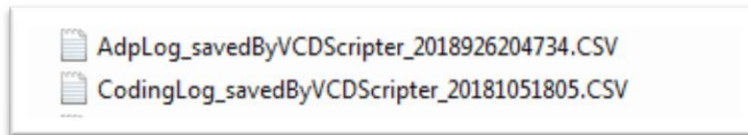
Der Autor steht in keiner Geschäftsbeziehung mit Ross-Tech. Demnach ist auch die Qualität des VCDScripters nicht in Verbindung mit der Qualität von VCDS zu bringen. "VCDScripter" ist ein Hobby-Projekt einer Privatperson und verfolgt weder kommerzielle Absichten, noch wird es in kommerzieller Verbindung mit Ross-Tech LLC.

2) Überblick



- 1) Der Textbereich für die Programmierung des Automations-Skripts
- 2) Textfelder, die zusammen mit der Skriptdatei gespeichert und beim Öffnen erneut geladen werden
 - a. Freitext-Feld für den Autor
 - b. Automatischer Zeitstempel der letzten Änderung / des letzten Speicher-Datums
 - c. Freitext-Feld für den Kommentar
 - d. Checksumme im Falle einer Versiegelung des Skripts (siehe "Skript speichern")
- 3) Menüpunkte Neu, Öffnen, Speichern und Beispiele
 - a. Neu: Leert alle Eingaben
 - b. Öffnen: Öffnet ein abgespeichertes Skript im gültigen .vsf Format

- c. Speichern: Speichert ein Skript im .vsf Format
- d. Beispiele: Öffnet ein weiteres Menü mit länderspezifischen Beispiel-Skripts
- 4) Optionen zum Starten oder Veräußern des Skripts
 - a. Debug-Logs neu schreiben: Sichert die aktuelle CodingLog.csv und AdpLog.csv mit aktuellen Zeitstempel, sodass das Skript neue Logfiles schreibt



- b. Skript veräußern: Speichert das Skript als autark ausführbare .exe Datei, die weitergegeben und unabhängig von der VCDScripter-Umgebung ausgeführt werden kann
- c. Skript ausführen: Startet das Skript sofort

3) Skript-Befehle

Jeder Befehl sollte mit einem Zeilenumbruch abgeschlossen werden.

StartVCDS()

Parameter: Keine

Obligatorischer Befehl zu Beginn jedes Skripts.

Startet VCDS. Beim Start eines Skripts sollte VCDS geschlossen sein!

CreateAdpMap("...", "...")

Parameter:

- Steuergerät-Adresse
- Steuergerät-Login (optional)

Erstellt ein CSV Steuergeräte-Abbild eines Steuergeräts unter möglicher Angabe des Logins.

```
1 StartVCDS()
2 CreateAdpMap("A5", "")
3 CreateAdpMap("5F", "")
4 OpenUnit("A5")
```

Hinweis: Bei UDS-Steuergeräten ist in der Regel kein Login notwendig um die Kanäle auslesen zu können.

Beispiel: `CreateAdpMap("10", "")`

Der Befehl ist nur außerhalb eines Steuergeräts erlaubt!

CreateHexAdpMap("...", "...")

Parameter:

- Steuergerät-Adresse
- Steuergerät-Login (optional)

```
1 StartVCDS()  
2 CreateHexAdpMap("09", "")  
3 OpenUnit("09")
```

Erstellt ein RAW-/HEX-Steuergeräte-Abbild eines Steuergeräts unter möglicher Angabe des Logins.

Hinweis: Bei UDS-Steuergeräten ist in der Regel kein Login notwendig um die Kanäle auslesen zu können.

Beispiel: `CreateHexAdpMap("10", "")`

Der Befehl ist nur außerhalb eines Steuergeräts erlaubt!

CreateAdpMapDump()

Parameter: Keine

Erstellt CSV Steuergeräte-Abbilder von allen Steuergeräten

Beispiel: `CreateAdpMapDump()`

Der Befehl ist nur außerhalb eines Steuergeräts erlaubt!

CreateHexAdpMapDump()

Parameter: Keine

Erstellt RAW-/HEX-Steuergeräte-Abbilder von allen Steuergeräten

Beispiel: `CreateHexAdpMapDump()`

Der Befehl ist nur außerhalb eines Steuergeräts erlaubt!

CreateAutoscan("...")

Parameter:

- Titel (KFZ-Kennzeichen)

Erstellt einen Auto-Scan unter dem optional angegebenen Titel bzw. KFZ-Kennzeichen. Dieser wird standardmäßig im Logs-Verzeichnis von VCDS abgelegt.

Beispiel: `CreateAutoscan("VOR_VZE_CODIERUNG")`

Der Befehl ist nur außerhalb eines Steuergeräts erlaubt!

CheckForUnits("...", 0/1)

Parameter:

- Komma-separierte Liste von Steuergerät-Adressen
- 0 oder 1
 - 0: übergeht nicht gefundene Steuergeräte im weiteren Skript
 - 1: Beendet das Skript mit Fehler, sollte eine angegebene Adresse nicht verbaut sein

Der Befehl ist nur nach StartVCDS() erlaubt und prüft auf das Vorhandensein von Steuergeräten.

Beispiel: `CheckForUnits("13,A5,4B", 1)`

Erläuterung: Prüft ob die Steuergeräte 13, A5 und 4B verbaut sind. Bricht mit Fehler ab falls eines davon nicht gefunden wird.

Beispiel: `CheckForUnits("13,3C", 0)`

Erläuterung: Prüft ob die Steuergeräte 13 und 3C verbaut sind. Im Weiteren Skript werden Steuergeräte-Blöcke übersprungen, die nicht vorhandene Steuergeräte behandeln. Der rot umrandete Teil wird somit schlicht nicht ausgeführt, sollte das Steuergerät 13 nicht verbaut sein.

```
1 StartVCDS()
2 CheckForUnits("10,13",0)
3 OpenUnit("10")
4   ClearDTCs()
5 CloseUnit()
6 OpenUnit("13")
7   ChangeCoding("3:3","")
8   ClearDTCs()
9 CloseUnit()
```

OpenUnit("...")

Parameter:

- Steuergerät-Adresse

Öffnet ein Steuergerät um weitere Aktionen darin durchzuführen

Beispiel: `OpenUnit("10")`

Erläuterung: Öffnet das PDC-Steuergerät

TestComponent("...", 0/1)

Parameter:

- Zeichenfolge
- 0 oder 1
 - 0: überspringt bei nicht-Übereinstimmung die weiteren Befehle
 - 1: Beendet das Skript mit Fehler, sollte die Zeichenfolge nicht gefunden werden

Prüft auf das Vorhandensein einer Zeichenfolge in der Textzeile "Bauteil" innerhalb des Steuergeräts.

Beispiel: `TestComponent("PDC 8 Kanal", 0)`

Erläuterung: Prüft auf die Zeichenfolge "PDC 8 Kanal" in der Zeile Bauteil. Überspringt weitere Befehle bis zum nächsten Steuergerät, sollte die Zeichenfolge nicht übereinstimmen.



TestPartNo("...", 0/1)

Parameter:

- Zeichenfolge
- 0 oder 1
 - 0: überspringt bei nicht-Übereinstimmung die weiteren Befehle
 - 1: Beendet das Skript mit Fehler, sollte die Zeichenfolge nicht gefunden werden

Prüft auf das Vorhandensein einer Zeichenfolge in der Textzeile "Teilenummer" innerhalb des Steuergeräts.

Beispiel: `TestPartNo("5QA 919 294", 1)`

Erläuterung: Prüft auf die Zeichenfolge "5QA 919 294" in der Zeile Teilenummer. Bricht mit Fehler ab, sollte die Zeichenfolge nicht gefunden werden



TestAdpChannel("...", "..." , 0/1)

Parameter:

- Zeichenfolge (Titel oder teilweiser Titel des Anpassungskanals)
- Zeichenfolge (optional: gespeicherter Wert des Kanals)
- 0 oder 1

- 0: überspringt bei nicht-Übereinstimmung die weiteren Befehle
- 1: Beendet das Skript mit Fehler, sollte die Zeichenfolge nicht gefunden werden

Prüft vor dem Fortfahren auf die Existenz eines Anpassungskanals im Steuergerät, und optional noch nach dem gespeicherten Wert innerhalb des Kanals.

Beispiel: `TestAdpChannel("Erweiterte_Fernlichtsteuerung", "", 0)`

Erläuterung: Prüft auf die Existenz des Anpassungskanals im 1. Parameter, und übergeht das komplette Steuergerät wenn dieser nicht existiert. (Wichtig um MQB BCMs auseinander zu halten)

Beispiel: `TestAdpChannel("Erweiterte_Fernlichtsteuerung", "AFS: FLA: Fernlicht (GLW:MDF)", 1)`

Erläuterung: Prüft neben der Existenz des Anpassungskanals im 1. Parameter auch auf den gespeicherten Wert im 2. Parameter. Bricht mit Fehler ab, sollte beides nicht gegeben sein.

ClearDTCs()

Parameter: Keine

Löscht den Fehlerspeicher des aktuell geöffneten Steuergeräts. Damit die Funktion ausgeführt werden kann muss kein Fehler im Speicher abgelegt sein.

LoginToUnit("...")

Parameter:

- Login-Code für Steuergerät

Führt einen Login (Zugriffsberechtigung) im aktuell geöffneten Steuergerät durch.

Beispiel: `LoginToUnit("31347")`

ChangeCoding("...", "...")

Parameter:

- Zu aktivierende Bits (Parameter 1)
- Zu deaktivierende Bits (Parameter 2)

Ändert die Codierung eines Steuergeräts. Es ist möglich Bits zu aktivieren und zu deaktivieren

Die Syntax eines Parameters ist wie folgt:

`[Byte]:[Bit],[Bit]/[Byte]:[Bit],[Bit]`

Beispiele:

- `ChangeCoding("4:1,2,3/5:0", "")`
 - Aktiviert Byte 4 Bit 1, 2 und 3 sowie Byte 5 Bit 0
- `ChangeCoding("6:7", "2:5,6")`
 - Aktiviert Byte 6 Bit 6, Deaktiviert Byte 2 Bit 5 und Bit 6
- `ChangeCoding("", "1:7")`
 - Deaktiviert Byte 1 Bit 7

Möchte man lediglich aktivieren bzw. deaktivieren, lässt man den anderen Parameter leer ("").

Es ist nicht notwendig, dass sich die IST-Codierung von der SOLL-Codierung unterscheidet. Sollte nach der Änderung die gleiche Codierung herauskommen, wird diese erneut bestätigt.

Versucht man ein Byte zu ändern welches nicht existiert (Codierung kürzer), wird das Skript mit einem Fehler abgebrochen.

Hinweis: Kurze Codierungen werden nicht unterstützt!

ChangeCodingInSub("...", "...", "...")

Parameter:

- Subsystem-Zeichenfolge (Parameter 1)
- Zu aktivierende Bits (Parameter 2)
- Zu deaktivierende Bits (Parameter 3)

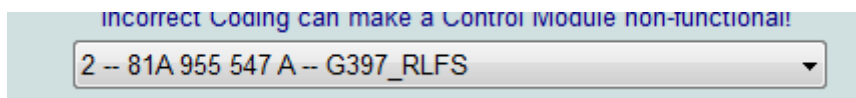
Ändert die Codierung eines Subsystems (bspw. Regensensor).

Es gilt die gleiche Syntax für Parameter 2 und 3 entlang "ChangeCoding".

In Parameter 1 wird eine Zeichenfolge angegeben, die in der Subsystem-Liste gefunden werden muss.

Beispiel: `ChangeCodingInSub("RLFS", "0:2", "")`

- Aktiviert Byte 0 Bit 2 im RLFS Subsystem, sofern man vorher das Steuergerät 09 geöffnet hat.



ChangeAdpChannel("...", "...")

Parameter:

- Zeichenfolge für Titel des Anpassungskanals
- Genauer Zielwert (neuer Wert)

Ändert einen Anpassungskanal auf einen neuen Wert.

Beispiel: `ChangeAdpChannel("Automatische Aktivierung", "aktiv")`

Die Zeichenfolge muss einen Treffer in der Auswahlliste der Anpassungskanäle hervorbringen, sonst wird das Skript mit einem Fehler abgebrochen.

Im Falle mehrerer Treffer:

Fördert eine Zeichenfolge ("road_sign_identification") mehrere Treffer zu Tage, die alle die gesuchte Zeichenfolge enthalten, so wird der erste Kanal ausgewählt und geändert!

Der Zielwert muss genau einem Wert entsprechen, der in der Auswahlliste "Neuer Wert" verfügbar ist.

Das Skript wird mit einer Fehlermeldung abgebrochen falls dies nicht gegeben sein sollte!

Hinweis: Anpassungskanäle ohne Auswahl-Liste sondern mit reiner Werte-Eingabe sind ebenso unterstützt.

Einschränkung: Anpassungen von CAN-Steuergeräten, die mit einer Kanalnummer arbeiten, sind nicht unterstützt!

CloseUnit()

Parameter: Keine

Schließt das aktuell geöffnete Steuergerät.

ClearAllDTCs()

Parameter: Keine

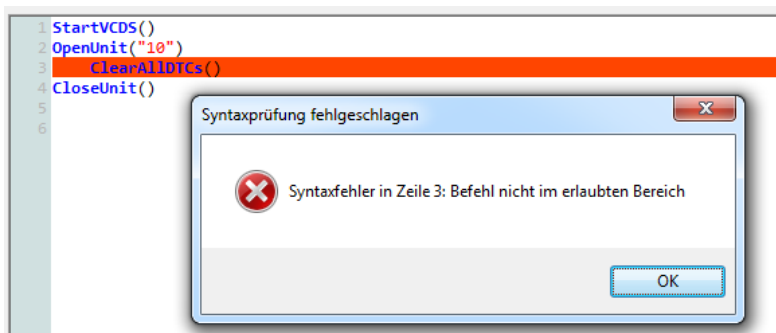
Löscht alle Fehlerspeicher.

Der Befehl ist nur außerhalb eines Steuergeräts erlaubt!

Gültigkeitsbereiche der Befehle

```
1 StartVCDS()  
2 CheckForUnits("10,09",1)  
3 OpenUnit("10")  
4   TestComponent("PDC",1)  
5   TestPartNo("294",1)  
6   ChangeAdpChannel("Automatische aktivierung","aktiviert")  
7   ClearDTCs()  
8 CloseUnit()  
9 OpenUnit("09")  
10  LoginToUnit("31347")  
11  ChangeAdpChannel("Fernlichtassistent reset","nicht aktiv")  
12 CloseUnit()  
13  
14 ClearAllDTCs()
```

Es sollte somit intuitiv recht klar sein, welche Befehle innerhalb eines Steuergeräts funktionieren, und welche außerhalb. Lediglich `CheckForUnits`, sowie `ClearAllDTCs` sind außerhalb eines Steuergeräts erlaubt. Der Editor wird auf eine Verletzung dessen hinweisen.



Kommentare

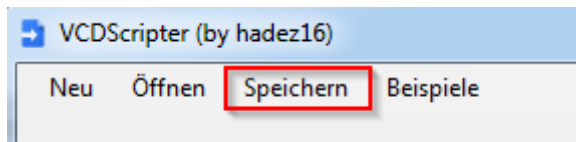
Innerhalb des Skripts können Kommentare verwendet werden. Diese werden mit Semikolon eingeleitet.

```

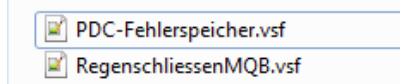
1 StartVCDS() ; dies ist ein Kommentar
2 ; noch ein Kommentar
3 OpenUnit("10") ; öffne Steuergerät 10
4 |

```

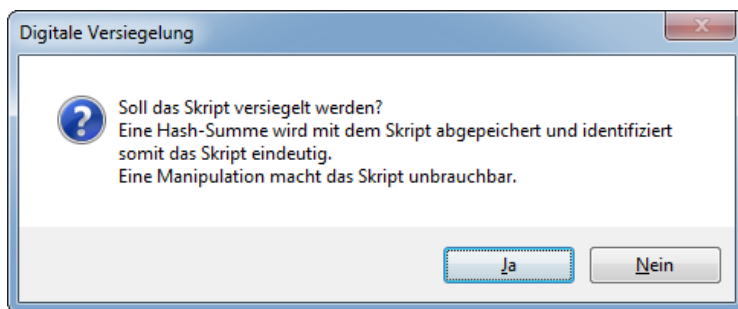
4) Speichern & Versiegeln eines Skripts



Über den Speichern-Dialog kann das aktuelle Skript samt Zusatzinformationen (Autor, Kommentar, falls angegeben) in einer Datei abgespeichert werden.



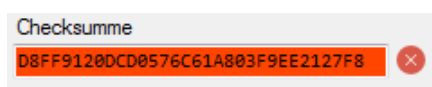
Beim Abspeichern erhält man eine Abfrage, ob man das Skript "versiegeln" möchte.



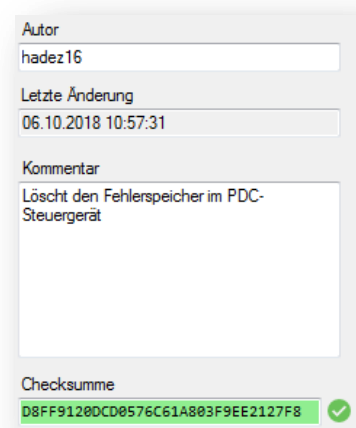
Bestätigt man dies mit "Nein", so wird das Skript einfach in die angegebene Datei abgespeichert.

Der Klick auf "Ja" berechnet zusätzlich noch eine eindeutige Prüfsumme, um das Skript und dessen Informationen absolut eindeutig zu identifizieren.

Lädt man das Skript erneut in den Editor, wird die Prüfsumme erneut berechnet. Stimmt diese überein, so ist die Authentizität des Skripts sichergestellt. Wird das Skript auf irgendeine Art und Weise manipuliert, stimmen die Prüfsummen nicht mehr überein.



Die Versiegelung bietet sich für den Fall an, dass Skripte innerhalb der Community geteilt werden und man sicher gehen möchte, dass der Autor sein Werk durch eine eindeutige Prüfsumme schützen kann.

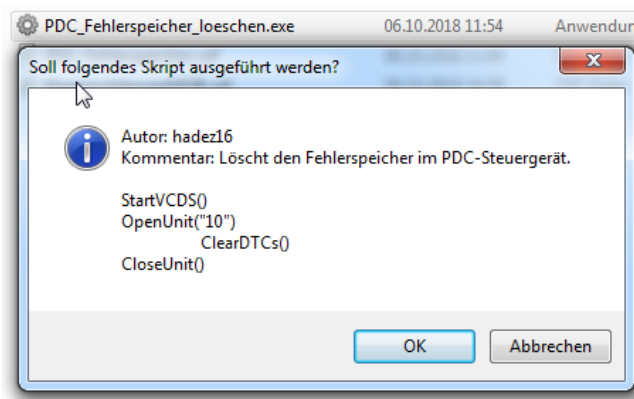


5) Skripte veräußern

Der Button "Skript veräußern" führt das Skript nicht aus, sondern bietet es dieses als eigenständig ausführbare EXE-Datei abzuspeichern.

Somit kann jeder VCDS-Benutzer auch ohne die VCDScripser Applikation ein bereitgestelltes Skript für sein Fahrzeug ausführen.

Beim Klick auf die ausführbare Datei wird abermals nachgefragt, unter Anzeige der enthaltenen Skript-Schritte, ob man das Skript ausführen möchte:

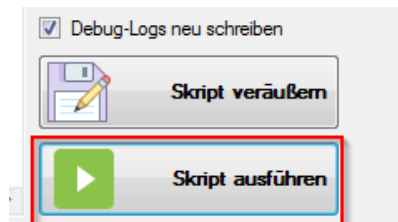
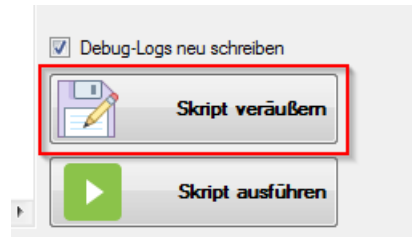


Wichtiger Hinweis: Es ist sehr wahrscheinlich, dass ein installierter **Virens Scanner** die Datei als schadhaft identifiziert und somit ggf. die Ausführung automatisch verhindert. Die ausführbare Datei enthält keinerlei schadhaften Code, jedoch werden die Aktionen Skripts von Virenscannern als schadhaft interpretiert.

6) Skripte starten

Der Button "Skript starten" führt das im Editor befindliche Skript sofort aus.

Wichtig ist, dass VCDS beendet ist, da es durch das Skript automatisch gestartet wird.



7) Limitationen und Hinweise

a) Limitationen

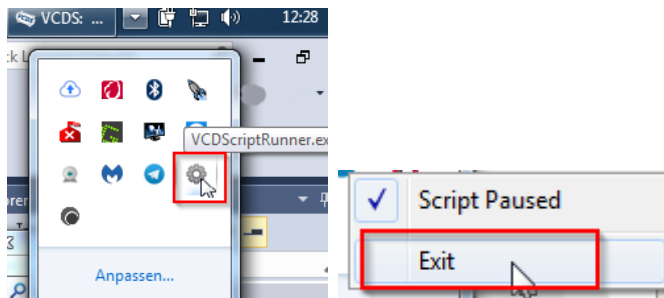
Der VCDScripiter bringt folgende Einschränkungen und Limitationen mit sich:

- Es werden nur Fahrzeuge unterstützt, bei denen VCDS direkt die verbauten Steuergeräte erkennt. (Verbaut-Reiter in der Steuergerätauswahl)
- Kurze/Dezimale Codierungen sind nicht unterstützt
- Anpassungskanäle auf Kanalnummer-Basis sind nicht unterstützt
- Es werden lediglich die neuen Interfaces HEX-V2 und HEX-NET unterstützt

b) Fehlerbehandlung

Die Logiken des VCDScripsters sehen vor, dass automatisch erkannt wird wenn das Skript zum Stillstand gekommen ist durch höhere Gewalt oder bisher unbekannte Verhaltensweisen.

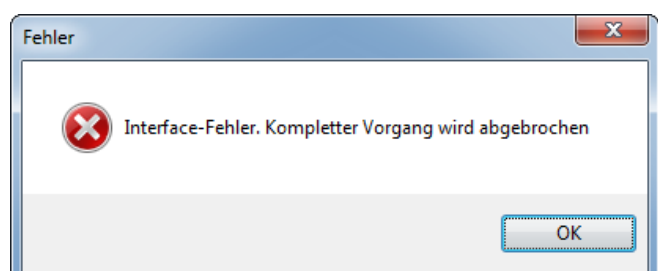
So sollte sichergestellt sein, dass das Skript sich automatisch beendet. Sollte sich das Skript nicht ordnungsgemäß beenden, entweder im Falle eines Fehlers, bei erfolgreichem Abschluss, oder nach automatisch erkannter Inaktivität, muss das Skript im System-Tray von Windows manuell beendet werden: (Rechtsklick auf das Zahnrad-Symbol, Exit)



Wichtiger Hinweis bezüglich Fehlerbehandlung:

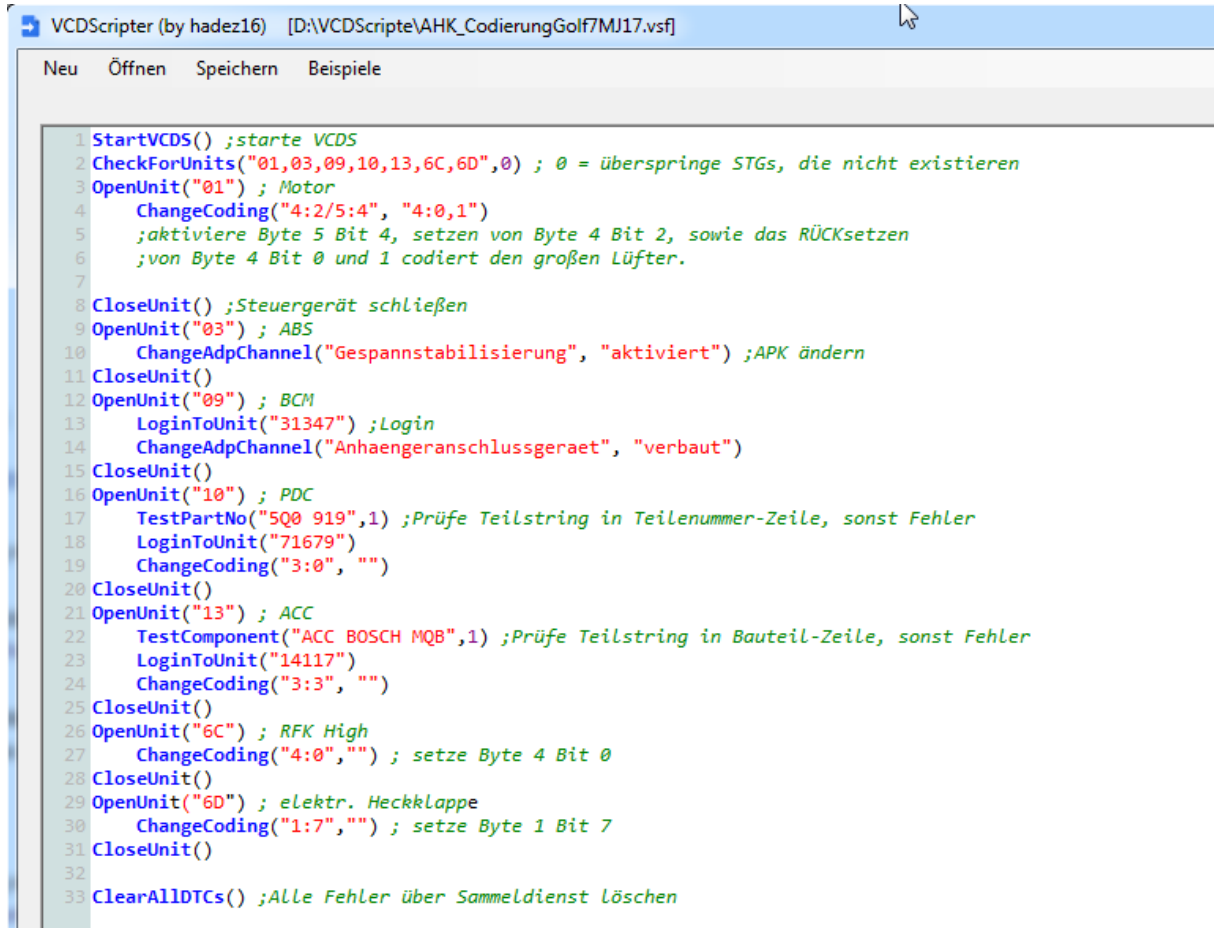
Sollte das Skript durch einen Fehler abbrechen (Steuergerät nicht erreichbar, Anpassungskanal nicht vorhanden,) so erfolgt kein automatischer Rollback auf den Ausgangszustand.

Wurden folglich schon Codierungen etc. durchgeführt, so muss man dies beachten und ggf. manuell auf den Ausgangszustand ändern. Das Neu-Schreiben der Debug-Logs bietet sich hier an, um einen Überblick über die erfolgten Änderungen zu haben.



8) Beispiel Anhängerkupplung MQB Facelift

Ein Zusammenspiel der Befehle hier an einer Variante der AHK-Freischaltung am Beispiel MQB Facelift: (Kein Anspruch auf 100% Korrektheit)



The screenshot shows the VCDScripter application window. The title bar reads "VCDScripter (by hadez16) [D:\VCDScripte\AHK_CodierungGolf7MJ17.vsf]". The menu bar includes "Neu", "Öffnen", "Speichern", and "Beispiele". The script content is as follows:

```
1 StartVCDs() ;starte VCDs
2 CheckForUnits("01,03,09,10,13,6C,6D",0) ; 0 = überspringe STGs, die nicht existieren
3 OpenUnit("01") ; Motor
4     ChangeCoding("4:2/5:4", "4:0,1")
5     ;aktiviere Byte 5 Bit 4, setzen von Byte 4 Bit 2, sowie das RÜCKsetzen
6     ;von Byte 4 Bit 0 und 1 codiert den großen Lüfter.
7
8 CloseUnit() ;Steuergerät schließen
9 OpenUnit("03") ; ABS
10     ChangeAdpChannel("Gespannstabilisierung", "aktiviert") ;APK ändern
11 CloseUnit()
12 OpenUnit("09") ; BCM
13     LoginToUnit("31347") ;Login
14     ChangeAdpChannel("Anhaengeranschlussgeraet", "verbaut")
15 CloseUnit()
16 OpenUnit("10") ; PDC
17     TestPartNo("5Q0 919",1) ;Prüfe Teilstring in Teilenummer-Zeile, sonst Fehler
18     LoginToUnit("71679")
19     ChangeCoding("3:0", "")
20 CloseUnit()
21 OpenUnit("13") ; ACC
22     TestComponent("ACC BOSCH MQB",1) ;Prüfe Teilstring in Bauteil-Zeile, sonst Fehler
23     LoginToUnit("14117")
24     ChangeCoding("3:3", "")
25 CloseUnit()
26 OpenUnit("6C") ; RFK High
27     ChangeCoding("4:0", "") ; setze Byte 4 Bit 0
28 CloseUnit()
29 OpenUnit("6D") ; elektr. Heckklappe
30     ChangeCoding("1:7", "") ; setze Byte 1 Bit 7
31 CloseUnit()
32
33 ClearAllDTCs() ;Alle Fehler über Sammeldienst Löschen
```