

## Kurzanleitung ScriptDeveloper SDV V3.08.04 (12. Mai 2019)

Der Scriptdeveloper (SDV) soll ein Hilfsmittel im Alltag bei der Erstellung von Homematic Skripten und deren Tests darstellen. Ein gewisses Wissen über Scripterstellung sowie den Aufbau einer CCU wird vorausgesetzt.

Die Software läuft auf Windows PC, ist bei nicht kommerzieller Nutzung Freeware.

Da mittlerweile aber schon einige tiefgreifende Operationen möglich sind, sind Löschfunktionen erst nach Drücken von Unlock  zugänglich.

Trotzdem an der Stelle der Hinweis, welcher auch beim ersten Start des Programmes bestätigt werden muss:

Dies ist eine BetaTestversion.

Die Verwendung dieser Software erfolgt auf eigenes Risiko  
Der Autor dieser Software übernimmt keine Haftung für direkte oder indirekte Schäden, welche sich aus der Benutzung dieser Software ergeben sollten.  
Eine kommerzielle Nutzung dieser Software ist untersagt

Ich bin einverstanden (Ja, Nein, wobei nein zum Programmende führt)

Hinweise über undokumentierte Methoden, die im Alltag nützlich sind aber ich bis jetzt auch noch nicht kannte, nehme ich gerne an und baue die auch gerne hier in das Programm mit ein.

# Inhalt

Kurzanleitung ScriptDeveloper SDV V3.08.04 (12. Mai 2019).....	1
1. Installation.....	4
1.1 Lizensierung.....	6
1.2 Donate .....	11
1.3 Systemvoraussetzungen.....	12
1.4 Was tut's bis jetzt .....	12
1.5 Bekannte Einschränkungen / Bugs.....	13
1.6 Geänderte Anforderung an Auflösung .....	13
1.7 Changelog .....	14
1.7.1 Changelog 03.08.04 LCL.....	14
1.7.2 Changelog 03.08.01 LCL.....	14
1.7.3 Changelog 03.07.13 LZL.....	14
1.7.4 Changelog 03.07.12 LZL.....	14
1.7.5 Changelog 03.07.11 LZL.....	14
1.7.6 Changelog 03.07.10 LZL.....	14
1.7.7 Changelog 03.07.07 LZL.....	15
1.7.8 Changelog 03.07.05 LZL.....	15
1.7.9 Changelog 03.07.04 LZL.....	15
1.8 Neue Authentifizierungsmechanismen Raspberrymatik .....	16
1.8.1 SSL.....	16
1.8.2 Authentifizierung über Nutzernname / Passwort.....	16
2. Oberfläche .....	18
2.1 Word Markup (seit 03.07.02).....	19
2.2 Blockweise auskommentieren.....	20
3 Scripteditor.....	21
3.1 Voreinstellungen Editor.....	23
3.2 Vervollständigen Funktion .....	24
3.3 Methodenhilfe.....	25
4 Inspektor.....	26
4.1 Selektionswahl: DomScan .....	27
4.2 Selektionskriterium Types .....	30
4.3 Zusätzliche Selektionsbedingungen .....	31
4.4 Daten aus Inspektor in Editor übernehmen .....	38
4.4.1 Mehrfachauswahl als Enum String .....	40
4.4.2 Übernahme von einem Script aus einem Programm direkt in den Editor .....	42

4.5 Selektion von Selektion .....	45
4.6 Objekte löschen.....	46
4.7 Anwenderdefinierte Sichten .....	48
4.8 Browsing durch Rekursionsebenen.....	49
4.9 Auflösen komplexer Objekte.....	52
4.9.1 Auflösen von Devices .....	52
4.9.2 Auflösen von Programmen.....	53
4.10 Zeitmodule .....	54
4.11 Suchen in Scripten nach Variablen, Devices etc.....	55
4.12 Volltextsuche in Scripten.....	58
4.13 SingleDestinations in ihrer Reihenfolge ändern.....	60
5 Favoritenansicht .....	63
5.1 Hinzufügen von Objekten in die Favoriten.....	63
5.2 Verwendete Objekte eines Programmes in die Favoriten laden .....	64
5.3 Favoritenliste löschen.....	64
5.4 Einschränkungen .....	64
6 Kleine Helfer im Alltag.....	65
6.1 Backups.....	65
6.1.1 Räume.....	65
6.1.2 Gewerke .....	65
6.1.3 Systemvariablen .....	67
6.1.4 Devices und Kanäle.....	67
6.1.5 Backup Programme .....	68
6.2 Umbenennen von Kanälen von Geräten .....	72
6.3 Paramset Master .....	73
6.4 Rega Push auf Datenpunkte via Rega event.....	74
7 Diagnosebild .....	76
7.1 SSH Funktionalität .....	78
7.2 Programm Condition Konsistenz.....	80
7.3 Programmanalyse.....	82
7.4 Systemvariablen Metadatenkonsistenz .....	86
7.5 Systemvariablen Test auf NirvanaChannelBezüge .....	88
7.6 Check Internal Sysvar .....	90
8 License Disclaimer .....	91

## 1. Installation

Das \*.rar File in ein beliebiges Verzeichnis entpacken. Ein Installer ist nicht notwendig. In diesem Verzeichnis befindet sich auch das Konfigurationsfile SDV.INI. Bei der erstmaligen Verwendung muss dieses angepasst werden

```
[LAST]
DATEI=c:\MTH\Homematic\NewScript.hsc
; Zusätzliche Dateitypen zur Filterung
; *.BMP;*.JPG;*.GIF
; die Aufzählung muss mit Semikolon getrennt sein
FILETYPE=                                Beispiel *.htm;*.scr
; Scriptdir
; ist Scriptdir leer, so wird Standartmäßig das Verzeichnis
; genommen, in dem der SDV gestartet wird
SCRIPTDIR=                                Beispiel c:\Scriptdir\

[HOST]
NICKNAME=DerNickNameAusDerLizenzenanfrage
CUXD=CUXD.CUX2801001:5
SYSTEMEXEC=true                           Aufruf wenn möglich über systemExec statt CuxD
LICENCE1=DerLizenzschlüssel1
LICENCE2=DerLizenzschlüssel2

[CCU1]                                     Neuer Schlüssel
IP=192.168.2.19                            Ip Von CCU1
USERNAME=ExternAdmin                        Nutzernname auf der CCU1
PASSWORD=XXXXXXXXXXXXX                      Passwort der Nutzers auf der CCU1
USEHTTPS=true                               Zugriff über HTTPS
SSHUSERNAME=root                            root wenn SSH Zugriff erwünscht
SSHPW=xxxxx^                                Das SSH Passwort der CCU eintragen

[CCU2]                                     Dito wie bei CCU1
IP=192.168.2.6
USERNAME=
PASSWORD=
USEHTTPS=false
SSHUSERNAME=root                            root wenn SSH Zugriff erwünscht
SSHPW=xxxxx^                                Das SSH Passwort der CCU eintragen

[HOSTCCU]
IP=192.168.2.19                           Die Werte der aktuellen CCU bei Start
                                            im Idealfall hier die Daten von CCU1 oder 2
                                            eintragen für 1. Start
USERNAME=ExternAdmin
PASSWORD=XXXXXXXXXXXXX
USEHTTPS=True
SSHUSERNAME=root                            root wenn SSH Zugriff erwünscht
SSHPW=xxxxx^                                Das SSH Passwort der CCU eintragen

[SECURITY]                                    Die Ports die der SDV benutzt
HTTPREGASCRIPt=8181
HTTPREGAXMLRPC=1999
HTTPSREGASCRIPt=48181
HTTPSREGAXMLRPC=41999
```

[ENUM\_NORM]

← Ab hier kommen dann interne Werte, Finger Weg

C1=65

C2=200

C3=293

C4=65

[ENUM\_MAX]

C1=65

C2=200

C3=293

C4=65

Hinweis für CCU2 Nutzer oder Nutzer älterer Firmware: Wenn sich kein Anfrageschlüssel generieren lässt (keine Hashs erkannt), in der INI Datei prüfen, ob dort der entsprechenden CCU der Eintrag USEHTTPS=True steht. Damit klappt es nicht. In diesem Fall muss der Eintrag in USEHTTPS=False geändert werden, dann klappt auch. Alte Firmwares mögen es auch nicht, wenn über Nutzernname und PW ein authentifizierter Zugriff über Name:PW@Host versucht wird. Bei älteren Firmware oder CCU keinen Nutzernamen oder Passwort eintragen bei CCU1 / CCU2 CCUHOST

Bei CCU2:

Auf jeden Fall USEHTTPS=false sowie USERNAME= und PASSWORD= (Username und Password leerlassen , sonst gibt's keine HashAnfrage

Bei CCU3:

Bei Authentifizierung auf CCU3 ein: USERNAME und PASSWORT des Admins müssen eingetragen sein

BestandsNutzer:

Die Ini Datei hat sich ziemlich vergrößert. Hilfreich ist, in der mitgelieferten INI alle Schlüssel ab [CFG\_ChanView1\_Methods]

Address=True.....

Und in die bestehende INI Datei anzuhängen. Erspart eine Menge manueller Hakensetzen.

Warum CUxD ? Der SDV Version 2.x nutzte noch pscp für den Zugriff auf die Logdatei und auf das System. Dies war immer ein Schwachpunkt (zusätzliches Programm, Bestätigung Serverzertifikat. Dies wird jetzt mit CUxD realisiert. Es muss ein Kanal angegeben werden auf einem CUxD exec Gerät, auf das der SDV exclusiven Zugriff hat. Auf Systemen ohne CUxD kann der SDV nicht eingesetzt werden.

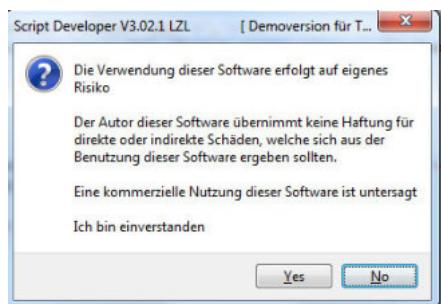
## 1.1 Lizenierung

Der SDV ist bei nicht kommerzieller Nutzung Freeware. Trotzdem habe ich mich entschlossen, aufgrund von Erfahrungen der Vergangenheit den Nutzerkreis oder die möglichen Features bestimmter Nutzer einzuschränken. Dies geschieht durch Vergabe von bis zu 2 Lizenzschlüsseln. Der SDV ist dadurch an bis zu 2 CCU / Raspberrymatic gepaart.

Wie arbeitet das ?

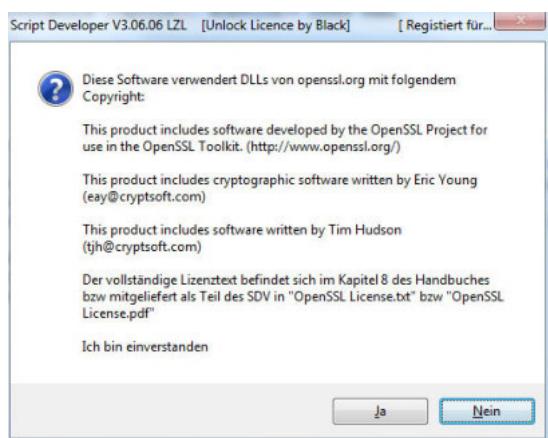
Der SDV telefoniert nicht nach Hause. Um eine Lizenz anzufragen ist folgender Weg einzuschlagen.

1. Die Konfigurationsdatei SDV.INI mit einem Editor öffnen.
2. Nickname anpassen
3. IP der CCU 1 eintragen
4. IP der CCU 2 eintragen
5. CUXD Kanal eintragen
6. Konfiguration abspeichern
7. Script Developer starten



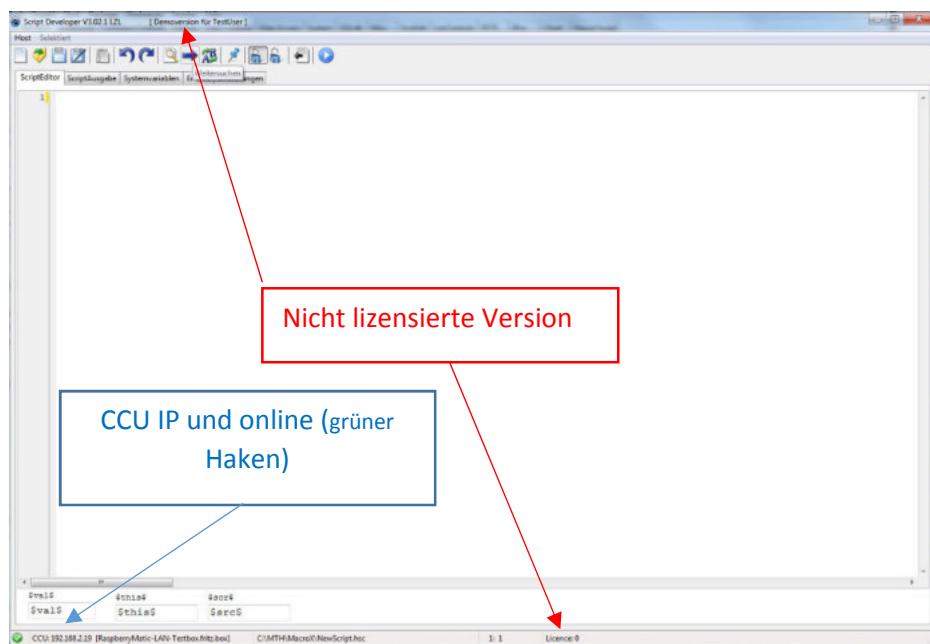
Bei allerersten Start muss dieses Fenster mit yes bestätigt werden. No führt so einem sofortigen Programmabbruch

Ab der Version 3.06.06 befinden sich in dem Package des SDV 2 DLL's von OpenSSL.org, welche für den HTTPS Zugang zur CCU benötigt werden. Hierbei ist einmalig ebenfalls eine Zustimmung über die Verwendung der Software nötig. Der komplette Lizenztext kann im Kapitel 8 nachgelesen werden bzw. ist als Datei mit in dem Package enthalten.

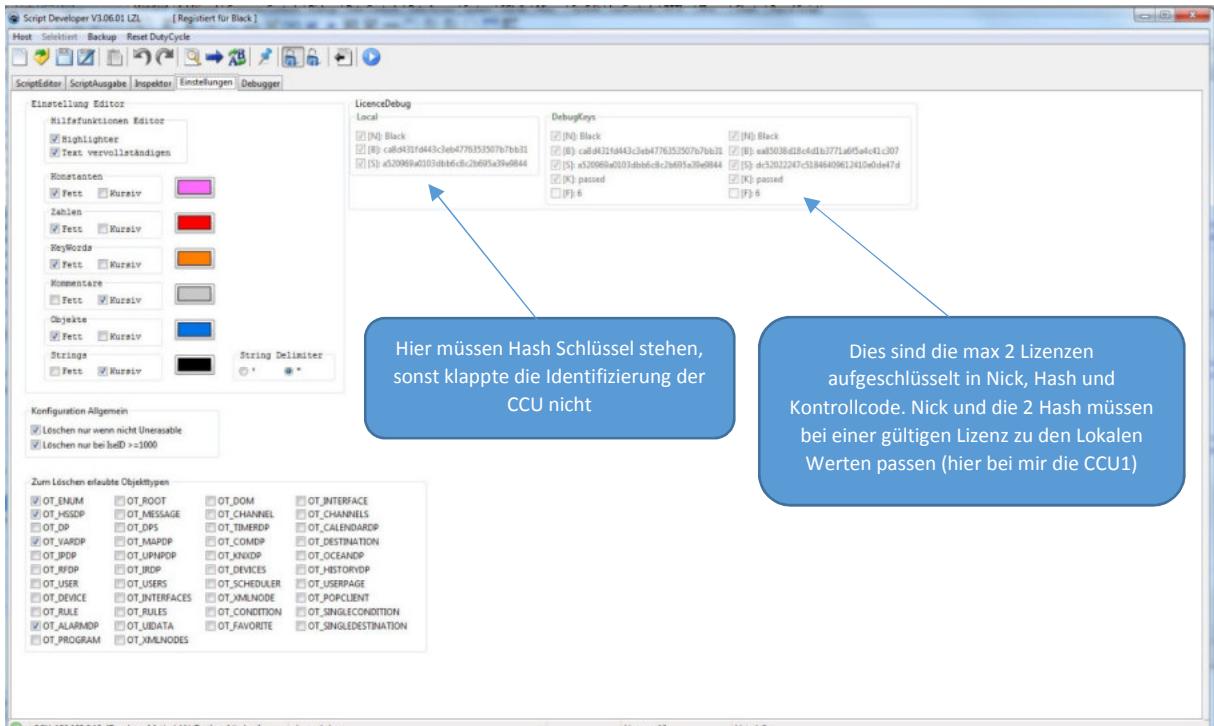


„Nein“ führt auch hier zu einem sofortigen Programmabbruch

Bei Bestätigung mit Yes startet nun zum erstenmal der SDV als Demoversion



Wenn die CCU, für die der Schlüssel angefragt werden soll, als grün angezeigt wird, bitte vorher einmal unter dem Reiter Einstellungen kontrollieren

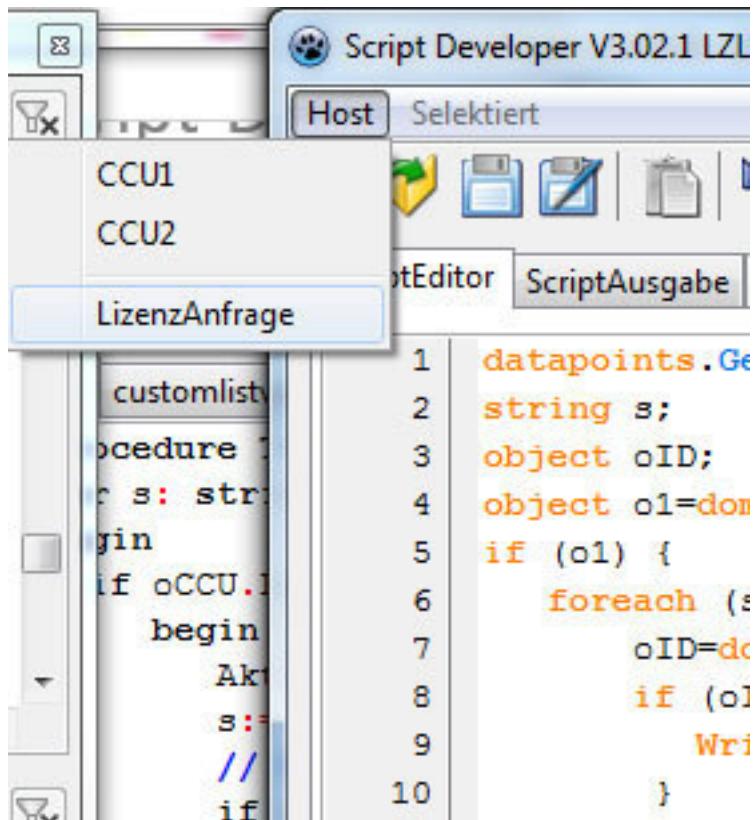


Aus einer Anfrage ohne unter local sinnige Einträge zu sehen lässt sich kein gültiger key generieren.

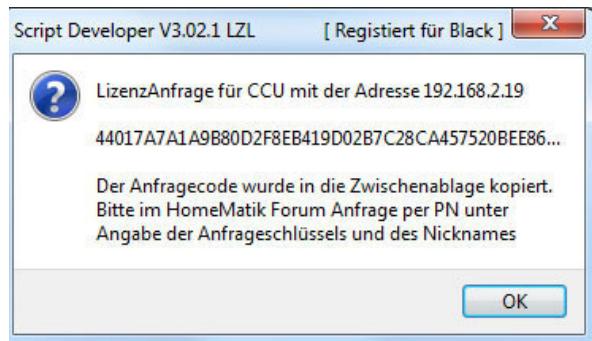
Aufgrund einer Programmungsgenauigkeit bei der Ausformulierung einer RegEx ist es leider sehr wahrscheinlich, dass alte Lizenzschlüssel vor 3.5.2 als ungültig erkannt werden. Sorry dafür, in dem Fall bitte die Anfrageschlüssel neu erstellen (wenn die Hashwerte gültig sind) und mir schicken. Der SDV 3.5.0 ist davon nicht betroffen, also am besten den 3.5.2 in ein neues Verzeichnis entpacken, die SDV.INI von dem 3.5.0 kopieren ins 3.5.2 Verzeichnis und neue Anfrage machen unter dem 3.5.2, da die alte INI nicht verändert wurde im alten Verzeichnis kann solange dann noch mit der 3.5.0 Version gearbeitet werden.

Im Auslieferungszustand ist unter NICKNAME= noch testUser eingetragen. Mit TestUser lässt sich kein Key generieren. Hierbei dann bitte die INI nochmal anpassen und den SDV neu starten-

Für die weiteren Schritte muss der SDV mit der CCU verbunden sein und die CCU auch als online erkannt worden sein.



unter Host auf Lizenzanfrage drücken. Als nächstes öffnet sich ein Fenster mit einem Anfrage Hexstring.



Der Hexstring ist in die Zwischenanlage kopiert und kann in beliebige Text Dokumente eingefügt werden. Als nächste dann im Homematik.de Forum eine PN an mich schreiben mit dem String und Angabe des Nicknames, welcher zum Zeitpunkt der Lizenzanfrage in der INI Datei eingetragen war.

Was enthält dieser Hexstring ?

Kodiert und verschlüsselt: 1. den Nicknamen, 2. die Seriennummer des Funkmodules der verbundenen CCU , einen VerifizierCode von mir.

Die Seriennummer des Funkmodules ist nötig zur Verifizierung des Pairings. Diese wird bei mir nirgends gespeichert, mit diesem Hexschlüssel wird nach der Anfrage der LizenzLevel definiert und ebenfalls in einen Hexstring kopiert. Dieser dann zurückgesandte Hexstring wird unter Licence1 oder Licence2 in der INI Datei eingetragen. Es kann mit bis zu 2 CCU bearbeitet werden, sollte ein Lizenzlevel höherwertiger sein so gilt dieser höherwertige Level für beide CCUs.

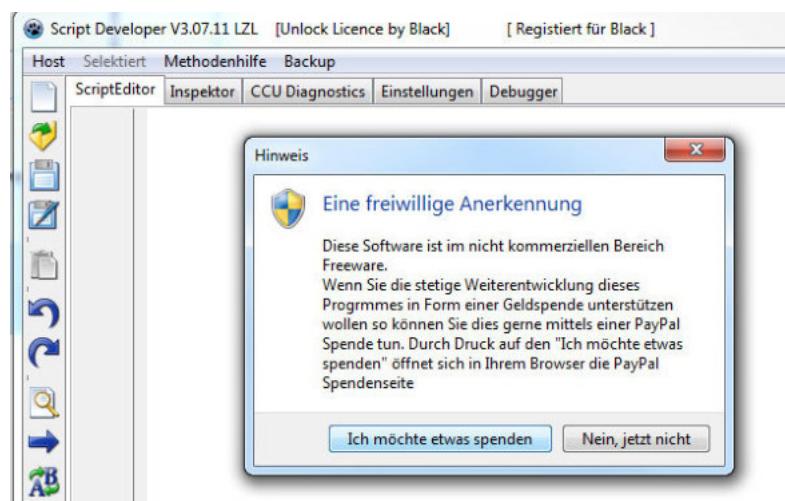
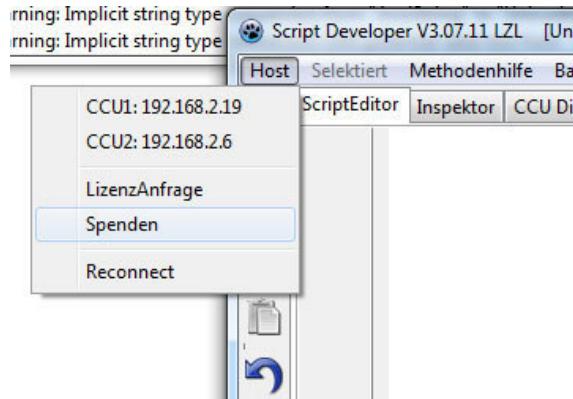
Wer mit diesem Verfahren nicht einverstanden ist, möge bitte an dieser Stelle die PDF Datei schließen und kann die Dateien beruhigt löschen.

Geplant hab ich folgende Lizenzabstufungen

Level	Editor	Script ausführen	Highlighter Und Vervollständiger	Enums	SysVar	Programs	Backup Restore		Special Funktions (resetDC, restor950 etc)
0	X								
1	X	X							
2	X	X	X						
3	X	X	X	X	X				
4	X	X	X	X	X	X			
5	X	X	X	X	X	X	X		
6									
7	X	X	X	X	X	X	X		X

## 1.2 Donate

Es besteht die Möglichkeit, dem Autor dieses Programmes für seine Arbeit eine Spende über Paypal zukommen zu lassen. Diese Möglichkeit ist völlig losgelöst von der Vergabe von Lizenzschlüsseln und rein freiwillig.



## 1.3 Systemvoraussetzungen

Der SDV lief bisher in Testinstallionen unter WIN 7 64/32 bit, und unter Win 10 64bit. Da unter recht konservativen Compilereinstellungen übersetzt wurde, sollte er eigentlich unter allen Windows Version laufen (ab Win 7). In einer AppleVM haben es auch schon welche stabil zum laufen bekommen.

Auf der Homematic-Seite wurde bei mir auf einer Raspberrymatic 3.43.15.20190323 und 3.41.11.20181126 getestet.

Auf einer CCU sind die erzeugen internen Progs auch lauffähig, wenn Rega-Community eingestellt wird. Unter Legacy läuft es NICHT !

Eventuell werde ich im Laufe des Jahres 2019 mal schauen, ob ichs nativ unter Debian auch kompiliert und zum Laufen bekomme.

## 1.4 Was tuts bis jetzt

Der Editor funktioniert inkl. Suchen und Suchen / ersetzen. Der Highlighter und der Code Vervollständiger arbeiten auch.

Undo / Redo arbeiten

Script ausführen arbeitet und liefert wie in der alten Version die antworten der CCU.

Enums und Sysvars arbeiten auch schon inkl Detaildaten und Editermöglichkeiten.

Darstellbarkeit zumindest der Grundmethoden aller Objekte

DomScan

Devices

Aufschlüsseln der MetaDaten

Datenpunkte

Kanäle

Programme

Favoriten

User

Paramset Master

Script Bearbeitung aus Programmen heraus

Verändert von Verzögert um von Scripten

Verändern von Retriggern

Komplettes Aufschlüsseln von Programmen in Rules, Subrules, Conditions, SingleConditions,

Destination und SingleDestination

MarkupUp im Editor

Suchen in Scripten nach Namen von Systemvariablen, Devices, Channels, Rooms und Functions

Suchen in Scripten nach Seriennummer von Devices und Channels

VollTextsuche in Scripten auch mit Regex Ausdrücken

Backup von Räumen, Gewerken, Favoriten, Systemvariablen, Devicenamen und Programmen

## 1.5 Bekannte Einschränkungen / Bugs

Auswahldialoge sind auf Englisch. Weiß ich, zurzeit benutze ich die in der Laufzeitumgebung integrierten Dialoge, und die sind leider trotz Landeseinstellung englisch.

Folding im Editor arbeitet noch nicht. Wenn der Rest läuft gucke ich da mal nach.

Kommentare im Script müssen als !- geschrieben werden. Kann man sich dran gewöhnen, das anzupassen wäre ein Haufen Aufwand, da EQ3 ja klugerweise Negation und Kommentar mit demselben Zeichen bedacht hat. Hurra. Ich kann jedenfalls mit dem !- gut leben, folglich ist die Chance, das ich das ändere, recht gering: xD

Aufgrund dessen, dass als Middleware bei mir IOBroker läuft und ich die Diagramm und die History Funktion der CCU nicht nutze, werde ich diese im SDV auch nicht ausprogrammieren.

## 1.6 Geänderte Anforderung an Auflösung

Nach Rückmeldung aus dem Testerkreis bezüglich dem Herabsetzen der nötigen Bildschirmauflösung habe ich dieses etwas neu aufgebaut.

Der SDV startet mit Höhe 769 und Breite 1300 und kann angepasst werden. Die Einstellungen werden beim Verlassen gespeichert.

Mit dem Kommandozeilenparameter SDV\_xxxx formreset kann eine Bildschirmauflösung wieder auf den Grundzustand zurückgesetzt werden. Alternativ die Einträge unter dem Schlüssel FORMVIEW in der INI Datei löschen.

## 1.7 Changelog

### 1.7.1 Changelog 03.08.04 LCL

- ein paar weitere Methoden hinzugefügt (auch die neuen der aktuellen RMMatik)
- Backup Restore Funktionalität nun auch für Programme (Einzelne oder mehrere)
- Möglichkeit nun mehrere Instanzen des SDV zu öffnen, ohne dass sich die INI Dateien in die Quere kommen (Nur die erste geöffnete Instanz, (Main) hat Schreibrechte auf die INI)
- Möglichkeit, dem SDV als Kommandozeilenparameter einen Dateinamen mitzugeben, damit ist bei Doppelklicken auf eine Datei automatisches öffnen des Scriptes im SDV möglich

### 1.7.2 Changelog 03.08.01 LCL

- ein paar Änderungen in der internen Abarbeitung als Vorbereitung für die nächsten Steps
- Selektionshandling verbessert (ist nun schneller und Bildaufbau ist ruhiger, vor allem bei Multiselektionen)
- Condition Inkonsistenz beseitigen verbessert (hat nun auch mein Produktivsystem glattgezogen)
- SSH kommt nun mit CUxd Aus und braucht plink nun doch nicht mehr (auch im Hinblick auf irgendwann mal Linux)
- Ein paar Methoden und Konstanten noch hinzugefügt
- Editor hat Funktion bekommen für Block auskommentieren und Auskommentieren entfernen (fand ich nützlich für Programmeentwickeln zum Testen)
- Masterparameter können direkt in den Editor als programmCode übernommen werden (Level 6) (unter Berücksichtigung vom Gerät, dem Kanal und dem ValueType des Parameters)
- aus einer Selektierten Singledestination kann der Editor unter Selektiert mit dem Neuen Menüpunkt "Singledestination Zugriff" direkt den programmcode zum selektieren der SDest erzeugen, um verzögert um in einem eigenen Script zu bearbeiten
- unter Methodenhilfe kann der Code zum Berechnen der "verzögert um" zeit direkt abgerufen werden.
- Sichten für Systemvariablen nun endlich fertig ausprogrammiert
- Favoritenansicht eingefügt Favoriten können aus der Listenübersicht übernommen werden, ein Doppelklick in der Favoritenansicht auf ein Object führt zum Laden der Favoritenansicht in die Listenübersicht mit detailldarstellung des Selektierten Objectes (undo redo Browsing ist dabei berücksichtigt)

### 1.7.3 Changelog 03.07.13 LZL

- eine rudimentäre SSH Funktionalität.
- Condition Inkonsistenzen sind nun auch behebbar (war behebbar ist)

### 1.7.4 Changelog 03.07.12 LZL

- kleiner Fix bei Gleitkommawerteingabe von Sysvars, DP (Hier wurde der Nachkommateil unterdrückt)
- Bugfix bei der Fensterdarstellung im Detail- Edit Feld
- Firmware der verbundenen CCU wird nun in der Statusbar angezeigt
- Fix für Level 7 Anwendung aufgrund Firmware 3.45.5 bei RMatik

### 1.7.5 Changelog 03.07.11 LZL

Kleine Bugfixes  
Tab Scriptausgabe entfernt, dafür sitzt die Scriptausgabe nun in einem Fenster unter dem Editor (Größe anpassbar)  
Neue Keys eingeführt, um Script Run und Script Test auch über FKeys zu starten  
Mit dieser Version kennt der SDV 620 Rega-Methoden und Konstanten  
Methodenhilfe eingeführt

### 1.7.6 Changelog 03.07.10 LZL

Kleine Bugfixes  
Properties von Zeitmodulen und Singledestination / Singleconditions über Dropboxes änderbar  
Systemvariablen Metadaten Check und Reparatur eingeführt  
Systemvariablen NirvanaChannel Check und Reparatur eingeführt

## 1.7.7 Changelog 03.07.07 LZL

### Kleine Bugfixes

Bugfix: Alt-S aus dem Editor brachte nicht die Special Spalte: geändert in Alt-p  
Ein Paar Methoden hinzugefügt  
Reiter Scriptsuche fertiggestellt (ab level 6)  
Damit nun möglich: Volltextsuche in allen Scripten  
Selektiertbar, auch suchen in GeisterDestinations möglich  
Suchen auch als Regular Expression möglich  
Beschleunigte Suche in Ram möglich  
ProgrammObjektanalyse

## 1.7.8 Changelog 03.07.05 LZL

### Kleine Bugfixes

Bugfix bei Programme Testen („\n“ in einem Kommentar führte zu einem Fehler) Bugfix tut nur auf RM, auf der die Rega bezüglich Doublequote gefixt wurde.  
Im Editor folgende Tastenkürzel:  
Ctrl F – Suchen Dialog, Ctrl R Replace Dialog, F3 – Weitersuchen  
Reports werden nun Automatisch ermittelt  
Alternative Endungen für Scripte in INI hinzugefügt  
Alternatives Scriptverzeichnis in INI hinzugefügt

## 1.7.9 Changelog 03.07.04 LZL

Auflösung geändert und ein wenig das Design  
Programm Testen hinzugefügt  
Suchen in Scripten nach Seriennummern von Devices / Kanälen  
Ein paar weiterer Methoden eingepflegt

## 1.8 Neue Authentifizierungsmechanismen Raspberrymatik

Die neuen Versionen der Raspberrymatik unterstützen SSL und Authentifizierung über Nutzername / Password. Nach Wunsch und Tips aus dem Testerkreis habe ich auch dies versucht umzusetzen.

### 1.8.1 SSL

SSL wird nun realisiert über die OpenSSL Library, welche auch von der Lazarus Foundation empfohlen wird:

Quelle: [https://sourceforge.net/p/lazarus-ccr/s...en\\_ssl.pas](https://sourceforge.net/p/lazarus-ccr/s...en_ssl.pas).

Der SDV liegt als 32bit Compilat vor, wenn man sich die dazu benötigten DLL's selber herunterladen möchte, die Quellen sind folgende;

[http://packages.lazarus-ide.org/openssl-1.0.2j-x64\\_86-win64.zip](http://packages.lazarus-ide.org/openssl-1.0.2j-x64_86-win64.zip) für die 64 Bit Version  
und  
<http://packages.lazarus-ide.org/openssl-1.0.2j-i386-win32.zip> für die 32 Bit Version

Das Zip File auspacken und die 2 DLL's in das Verzeichnis kopieren, im dem sich auch die SDV.Exe befindet. In dem SDV rar File befindet sich auch die 32 Bit Version der DLL's inkl. dem Lizenztext als TXT und PDF. Damit sollte der Zugriff über HTTPS schon mal funktionieren.

### 1.8.2 Authentifizierung über Nutzername / Passwort.

Die Struktur der INI Datei wurde geändert, damit sich pro CCU nun auswählen lässt zwischen HTTP und HTTPS Zugriff und die Authentifizierung Nutzername / Passwort wahlweise genutzt wird.

Wird bei Nutzername oder Passwort nix eingetragen, so generiert der SDV einen Zugriff ohne die Kennung NN:PW@HOST. Sind beide Werte eingetragen, so wird der Zugriff überNN:PW@HOST generiert.

Beispielhafter Aufbau der neuen INI

```
[LAST]
DATEI=c:\MTH\Homematic\NewScript.hsc

[HOST]
NICKNAME=DerNickNameAusDerLizenzenanfrage
CUxD=CUxD.CUX2801001:5
SYSTEMEXEC=true
LICENCE1=DerLizenzschlüssel1
LICENCE2=DerLizenzschlüssel2

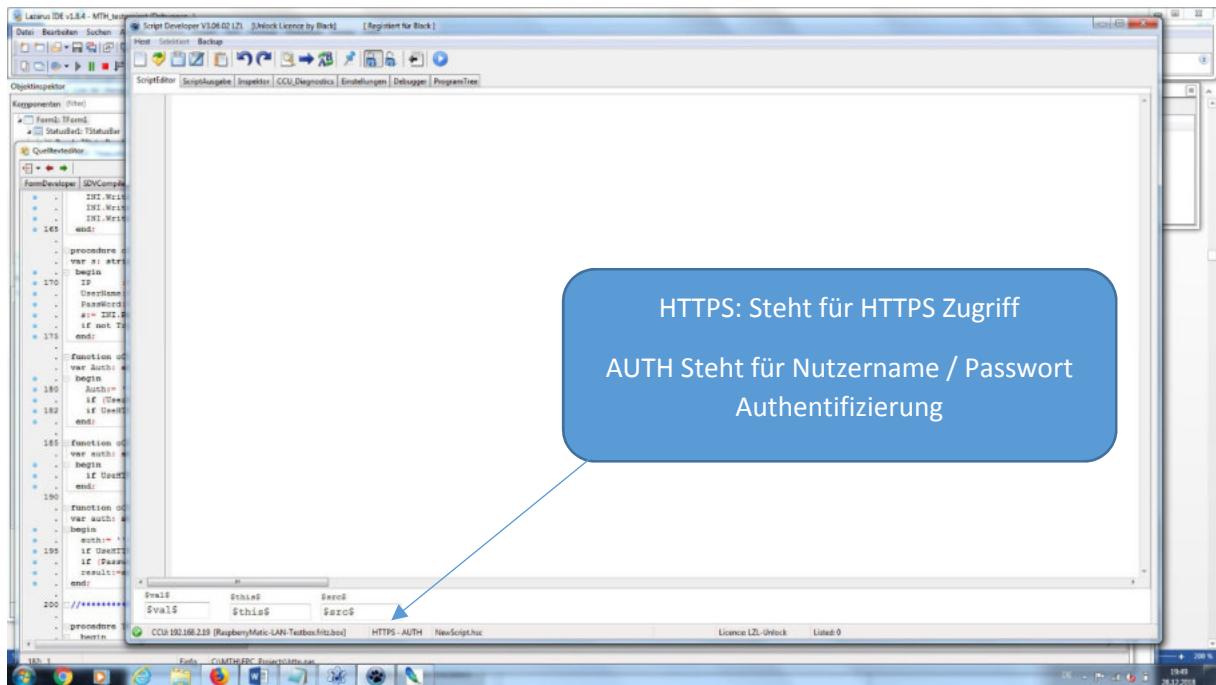
[CCU1]                               Neuer Schlüssel
IP=192.168.2.19                      Ip Von CCU1
USERNAME=Admin                         Nutzernname auf der CCU1
PASSWORD=XXXXXXXXXXXX                  Passwort der Nutzers auf der CCU1
USEHTTPS=true                          Zugriff über HTTPS

[CCU2]                               Dito wie bei CCU1
IP=192.168.2.6
USERNAME=
PASSWORD=
USEHTTPS=false

[HOSTCCU]                             Die Werte der aktuellen CCU bei Start
IP=192.168.2.19                      im Idealfall hier die Daten von CCU1 oder 2
                                         eintragen für 1. Start
USERNAME=Admin
PASSWORD=XXXXXXXXXXXX
USEHTTPS=True

[SECURITY]                            Die Ports die der SDV benutzt
HTTPREGASCRPT=8181
HTTPREGAXMLRPC=1999
HTTPSREGASCRPT=48181
HTTPSREGAXMLRPC=41999
```

Die gelben Texte müssen auf jeden in einer Bestehenden INI geändert werden. Ist das alles richtig gemacht, so zeigt der SDV im Statusfeld auch die Art des Zugriffs an:

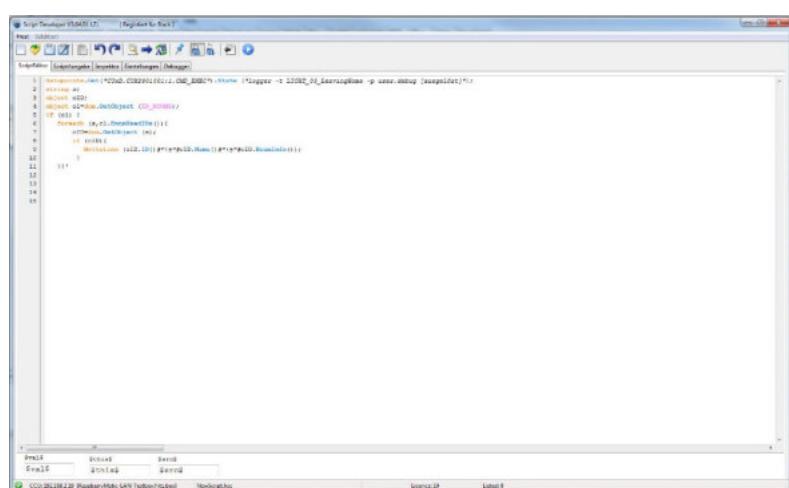
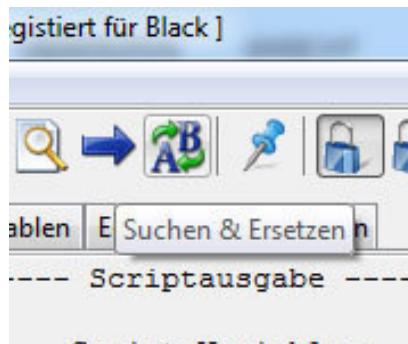


Ein Debugger Breakpoint im SDV zeigt hier den zusammengebauten HostString, der bei HTTPS und Authentifizierung benutzt wird



## 2. Oberfläche

Zu fast allen Funktionen sind die Hint parametriert, so dass es da Hilfestellung gibt.



Im Menüreiter **Scripte** finden sich die Einstellungen zum Anlegen eines neuen Scripts , zum Laden eines bestehenden Scripts und zum Speichern eines Scripts im Scripteditor sowie speichern unter.

In der Statuszeile finden sich Information über:

1. IP der Host CCU
2. DateiNames des Scriptes im Scripteditor
3. Anzahl der Elemente in der Listendarstellung

Folgende Shortcuts wurden implementiert:

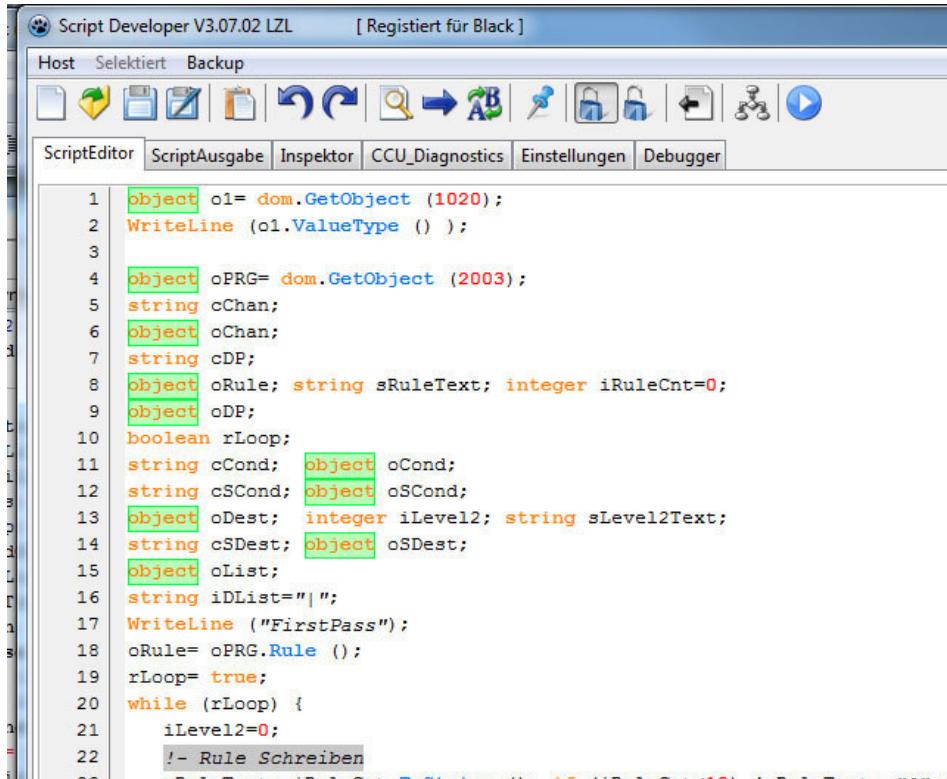
Ctrl-F: Suchen Dialog

Ctrl-R: Ersetzen Dialog

F3: Weitersuchen

## 2.1 Word Markup (seit 03.07.02)

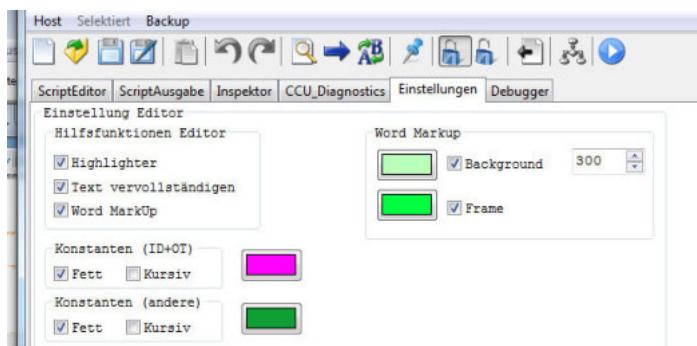
Der Editor beherrscht nun auch Wordmarkup. Bei einem Word, welches angeklickt wurde, oder geschrieben wurde (auf dem sich der Focus befindet ^^), werden sämtliche Vorkommen im Text farblich hervorgehoben



The screenshot shows the Script Developer interface with the title bar "Script Developer V3.07.02 LZL [Registiert für Black]". The menu bar includes "Host", "Selektiert", and "Backup". Below the menu is a toolbar with various icons. The main window contains a tab bar with "ScriptEditor" selected, followed by "ScriptAusgabe", "Inspektor", "CCU\_Diagnostics", "Einstellungen", and "Debugger". The code editor area displays the following script:

```
1 object o1= dom.GetObject (1020);
2 WriteLine (o1.ValueType () );
3
4 object oPRG= dom.GetObject (2003);
5 string cChan;
6 object oChan;
7 string cDP;
8 object oRule; string sRuleText; integer iRuleCnt=0;
9 object oDP;
10 boolean rLoop;
11 string cCond; object oCond;
12 string cSCond; object oSCond;
13 object oDest; integer iLevel2; string sLevel2Text;
14 string cSDest; object oSDest;
15 object oList;
16 string iDList="| ";
17 WriteLine ("FirstPass");
18 oRule= oPRG.Rule ();
19 rLoop= true;
20 while (rLoop) {
21     iLevel2=0;
22     !- Rule Schreiben
```

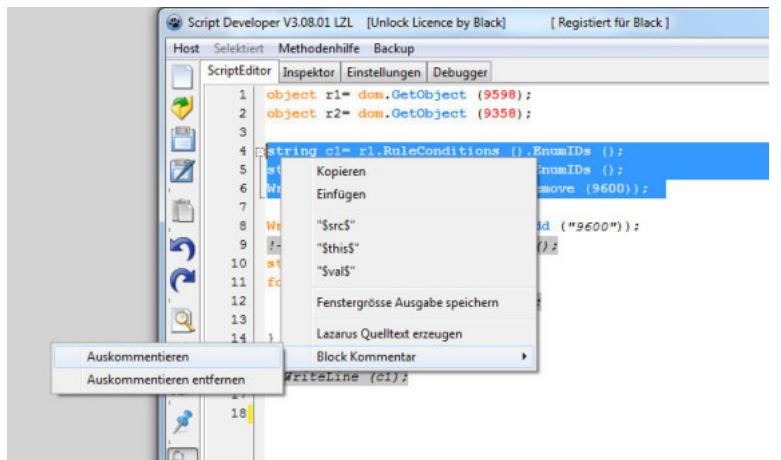
Die Funktion ist konfigurierbar unter Einstellungen



Die Einstellungen sollten selbsterklärend sein, die Zahlenangabe bezieht sich auf die Zeit in ms, ab wann seit dem letzten Tastenanschlag die Hervorhebung beginnt.

## 2.2 Blockweise auskommentieren.

Ab der Version V3.08.01 kann der SDV Editor Blöcke zum Testen auskommentieren oder wieder die Auskommentierung entfernen.



```
object r1= dom.GetObject (9598);
object r2= dom.GetObject (9358);

!- string c1= r1.RuleConditions ().EnumIDs ();
!- string c2= r2.RuleConditions ().EnumIDs ();
!- WriteLine (r1.RuleConditions().Remove (9600));

WriteLine (r1.RuleConditions().Add ("9600"));
!-r2.RuleConditions().RemoveAll ();
string sCND;
foreach (sCND,c1) {
    !-r1.RuleConditions.Add (sCND);
    WriteLine ("---#sCND#---");
}
!-WriteLine (c1);
```

## 3 Scripteditor

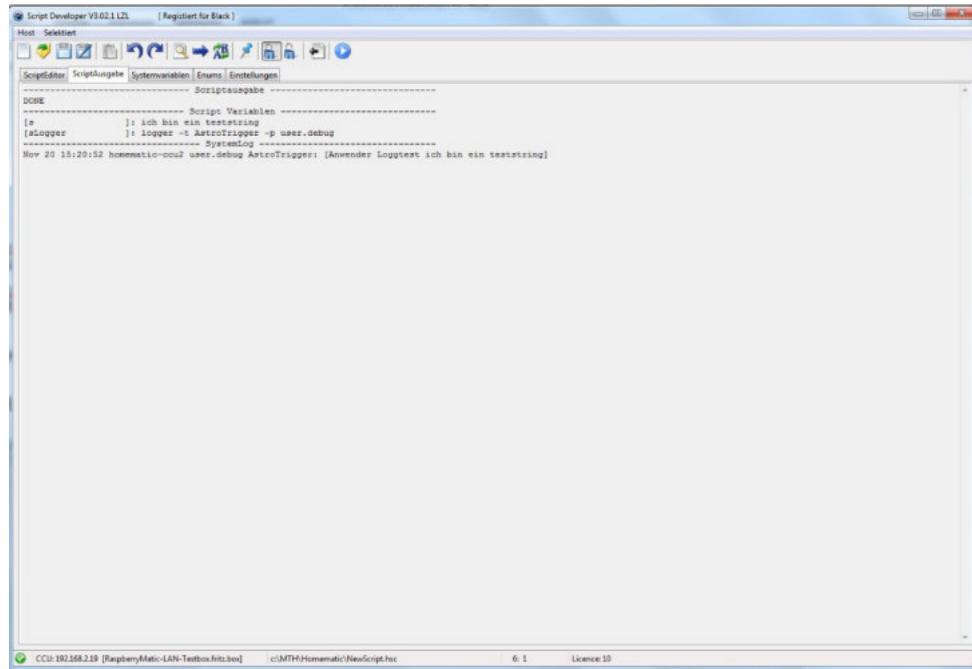
Im Scripteditor werden die Scripte geschrieben oder geladen, die mittels Run Script oder  an die CCU zum Ausführen gesendet werden. Das Scriptergebnis wird dann im Reiter Ausgabe angezeigt. Dieses kleine TestScript zum Beispiel:

```
string s= "ich bin ein teststring";
string sLogger      = "logger -t AstroTrigger -p user.debug ";

datapoints.Get("CUXD.CUX2801001:1.CMD_EXEC").State (sLogger # "[Anwender Loggtest " # s # "]");
WriteLine ("DONE");
```

Hier testweile aus State

Erzeugt folgende Ausgabe:



The screenshot shows the 'Scriptausgabe' tab of the Script Developer software. The output window displays the following text:

```
----- Script Variablen -----
[s] s= ich bin ein teststring
[s] sLogger      = logger -t AstroTrigger -p user.debug
----- SystemLog -----
Now 20 15:20:52 homematic-ccu2 user.debug AstroTrigger: [Anwender Loggtest ich bin ein teststring]
```

At the bottom of the window, it shows the host as 'CCU:192.168.2.19 [RaspberryMatic-LAN-Testbox.fritz.box]' and the file path as 'c:\(MTH)\Homematic\NewScript.hsc'. The status bar also indicates '6: 1' and 'Licence 10'.

Script Ausgabe stellt alles dar, was in dem Script mit Write, WriteLine oder Derivaten zur Ausgabe gebracht wurde,

Unter lokale Script variablen stehen die Variablen welche im Script definiert wurden mit ihren Namen. In dem Fall hier sind das die Beiden String Variablen s und sLogger.

Wurde via Userlog ein Eintrag im Logfile erzeugt, so wird dieser nach Scriptende auch hier angezeigt.

Sollte in dem Script ein Fehler sein (hier testweise State zu Stat geändert) erhält man die gleiche Ausgabe wie im Syslog:

```
[----- Fehler im Script -----]  
Jun 15 12:41:49 homematic-raspi local0.err ReGaHss: Error: IseESP::SyntaxError= Error 1 at  
row 4 col 88 near ^ (sLogger # "[Anwender Logtest " # s # "]");^M WriteLine ("DONE");^M  
[iseESP.cpp:1121]  
Jun 15 12:41:49 homematic-raspi local0.err ReGaHss: Error: ParseProgram: SyntaxError=  
(sLogger # "[Anwender Logtest " # s # "]");^M WriteLine ("DONE"); [iseESP.cpp:374]
```

Ab 3.07.04 : Script testen eingefügt, hierbei wird das Skript nur nach Fehlern geparst aber nicht ausgeführt (Syntax Check). Funktionsweise äquivalent zu der KlickiBunti Funktion: Script Testen.

Dazu im Scripteditor auf  drücken, es wird dann der Syntaxcheck der CCU aufgerufen. Bei erfolgreichem Syntaxcheck erfolgt für 2 Sekunden die Meldung

Test Script erfolgreich

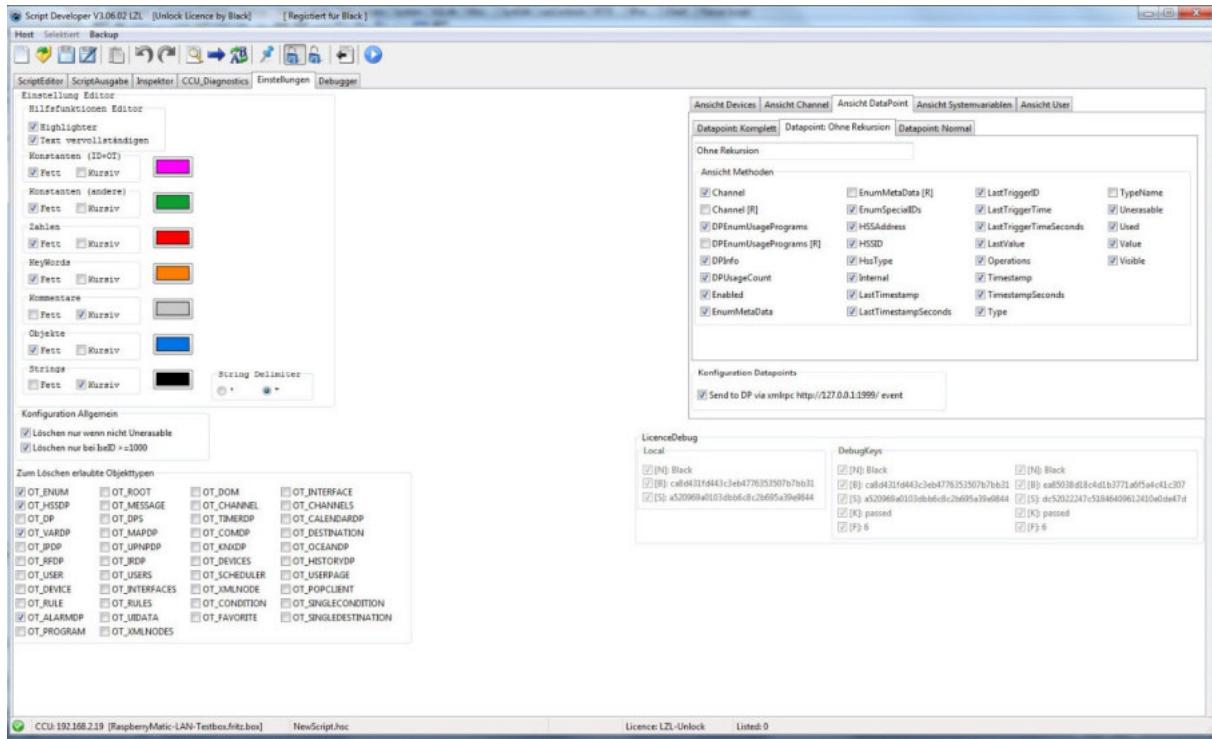
Trat ein Fehler auf, so erfolgt Sprung auf den Reiter Scriptausgabe mit Anzeige der Fehlermeldung

Bestandsnutzer können den folgenden Schlüssel in die SDV.INI schreiben:

```
[SCRIPTRUN]  
RUN=F1  
TEST=F12
```

Die legt Script RUN auf F1 und Script testen auf F12. Kann selber angepasst werden. F3 ist reserviert !!! für suche erneut

### 3.1 Voreinstellungen Editor



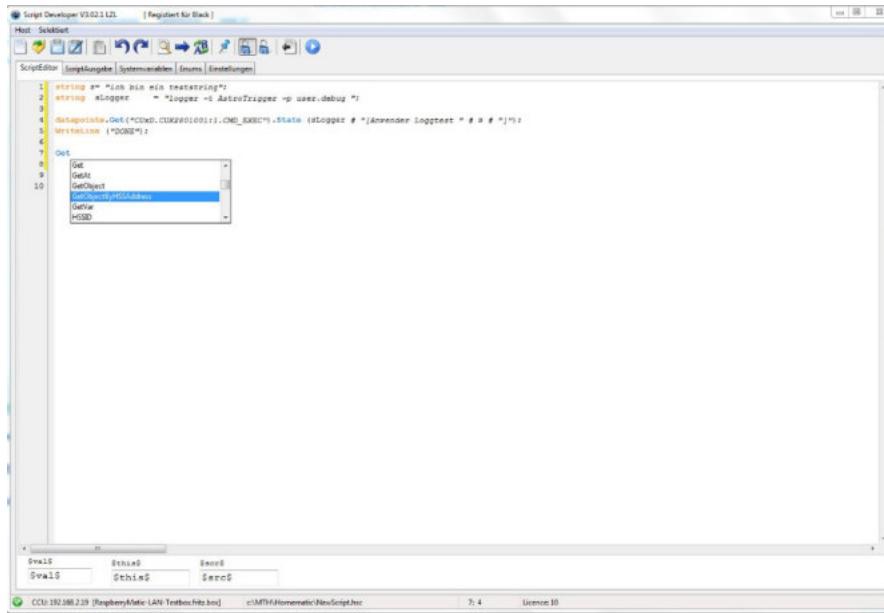
Hier kann nach Vorliebe der Highlighter konfiguriert oder auch ausgeschaltet werden

Ebenso lässt sich die Vervollständigen Funktion an oder abwählen.

Vorbedingung natürlich: Lizenzlevel muss vorhanden sein.

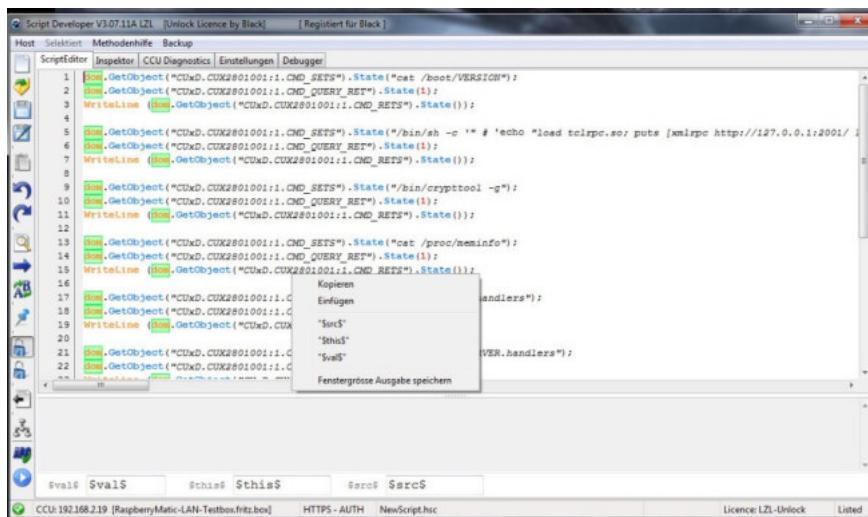
## 3.2 Vervollständigen Funktion

Methoden und Konstanten Namen muss man sich nicht auswendig merken. Der Editor verfügt über einen Auto Vervollständiger. Man schreibt den Wortanfang, hier z.B Get , drückt Strg+Space und wählt in dem sich öffnenden Menü die passende Funktion aus.



Nach Druck von Enter erscheint das Wort im Editor.

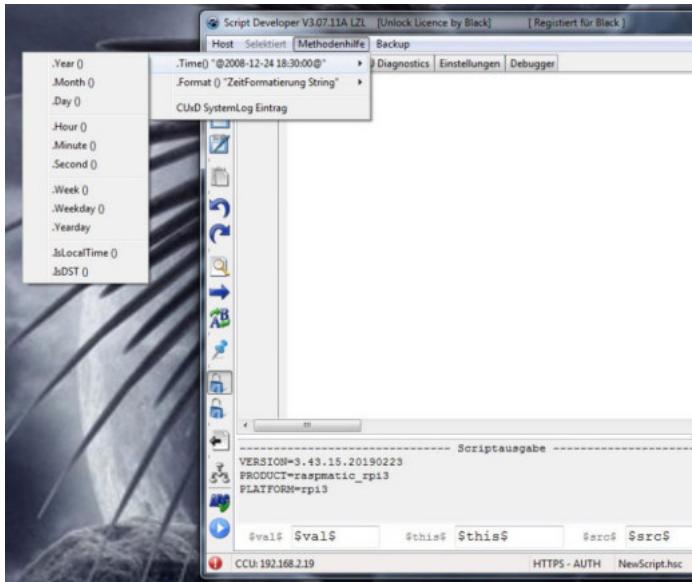
Die Grösse des Ausgabefensters kann abgespeichert werden.  
Rechte Maustaste in den Editor und Fenstergrösse Ausgabe speichern



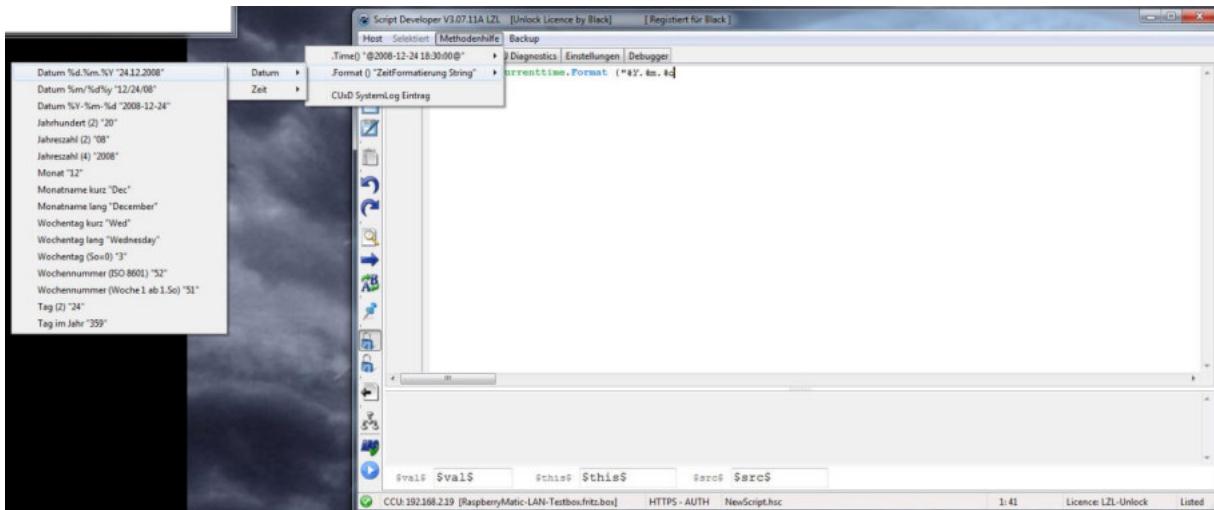
### 3.3 Methodenhilfe

Da es mich immer nervte, bei bestimmten Funktionen doch in Handbuch gucken zu müssen, gibt es nun den Menüpunkt Methodenhilfe (Aktiv im Editor)

Damit lassen sich zu Themen Time, Zeit, und Zeitformatierung Funktionen und die Formatstring schnell finden



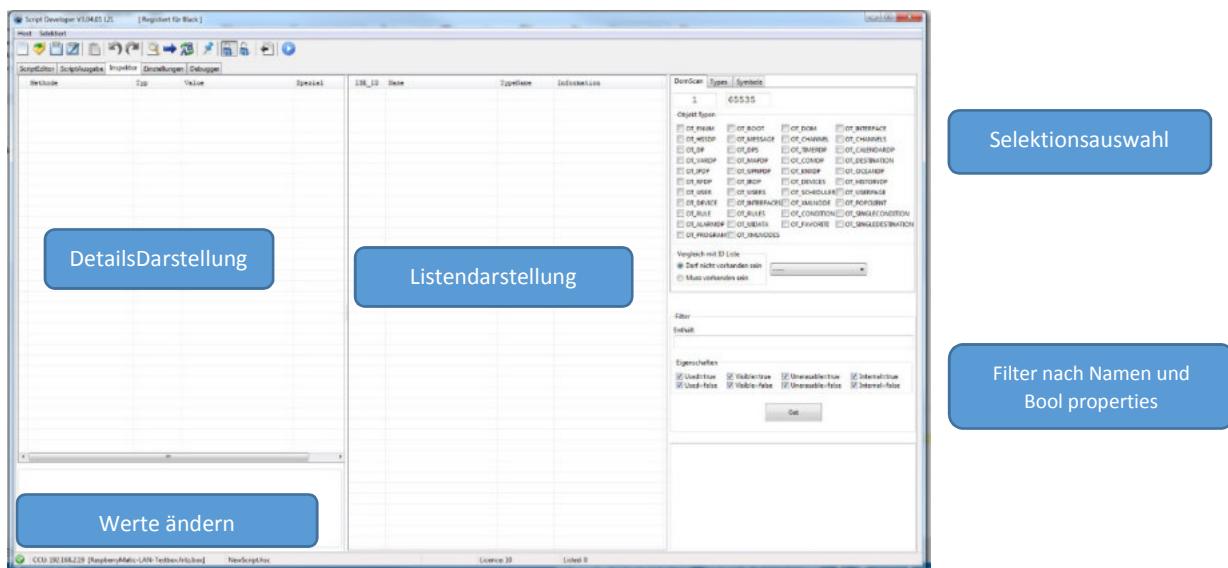
Durch Druck wird die Passende Funktion ausgewählt. Dito bei Zeitformatierung



## 4 Inspektor

Der Inspektor dient zum Suchen, Anzeigen und Ändern von Objekten auf der CCU/RaspberryMatik.

Es existieren verschiedene Selektionskriterien.



Filteroptionen:

Auswahl der Aufzählungen (Räume, Gewerke, Favoriten, Interfaces , Systemvariablen sind bisher implementiert)

Enthält: leerer Eintrag = es wird nicht nach enthaltener Buchstabensequenz selektiert  
Eingegebener Text. Die Systemvariable muss im Namen die Buchstabensequenz enthalten.

Eigenschaften: Es wird nach den Eigenschaften Used, Visible,Unerasable und Internal selektiert.  
Am Beispiel used:

1. Kein Haken bei Used= false und kein Haken bei Used= true  
Die Eigenschaft Used wird bei der Auswahl nicht beachtet
2. Haken bei Used= false und Haken bei Used= true  
Die Eigenschaft Used wird bei der Auswahl nicht beachtet
3. Kein Haken bei Used= false und Haken bei Used= true  
Um gelistet zu werden muss das Objekt die Eigenschaft Used=true haben
4. Haken bei Used= false und kein Haken bei Used= true  
Um gelistet zu werden muss das Objekt die Eigenschaft Used=false haben

## 4.1 Selektionswahl: DomScan

DomScan		Types	Symbole																																												
1	65535																																														
<b>Objekt Typen</b> <table border="1"> <tr><td><input type="checkbox"/> OT_ENUM</td><td><input type="checkbox"/> OT_ROOT</td><td><input type="checkbox"/> OT_DOM</td><td><input type="checkbox"/> OT_INTERFACE</td></tr> <tr><td><input type="checkbox"/> OT_HSSDP</td><td><input type="checkbox"/> OT_MESSAGE</td><td><input type="checkbox"/> OT_CHANNEL</td><td><input type="checkbox"/> OT_CHANNELS</td></tr> <tr><td><input type="checkbox"/> OT_DP</td><td><input type="checkbox"/> OT_DPS</td><td><input type="checkbox"/> OT_TIMERDP</td><td><input type="checkbox"/> OT_CALENDARDP</td></tr> <tr><td><input type="checkbox"/> OT_VARDP</td><td><input type="checkbox"/> OT_MAPDP</td><td><input type="checkbox"/> OT_COMDP</td><td><input type="checkbox"/> OT_DESTINATION</td></tr> <tr><td><input type="checkbox"/> OT_IPDP</td><td><input type="checkbox"/> OT_UPNPD</td><td><input type="checkbox"/> OT_KNXDP</td><td><input type="checkbox"/> OT_OCEANDP</td></tr> <tr><td><input type="checkbox"/> OT_RFDP</td><td><input type="checkbox"/> OT_IRDP</td><td><input type="checkbox"/> OT_DEVICES</td><td><input type="checkbox"/> OT_HISTORYDP</td></tr> <tr><td><input type="checkbox"/> OT_USER</td><td><input type="checkbox"/> OT_USERS</td><td><input type="checkbox"/> OT_SCHEDULER</td><td><input type="checkbox"/> OT_USERPAGE</td></tr> <tr><td><input type="checkbox"/> OT_DEVICE</td><td><input type="checkbox"/> OT_INTERFACES</td><td><input type="checkbox"/> OT_XMLNODE</td><td><input type="checkbox"/> OT_POPCLIENT</td></tr> <tr><td><input type="checkbox"/> OT_RULE</td><td><input type="checkbox"/> OT_RULES</td><td><input type="checkbox"/> OT_CONDITION</td><td><input type="checkbox"/> OT_SINGLECONDITION</td></tr> <tr><td><input type="checkbox"/> OT_ALARMDP</td><td><input type="checkbox"/> OT_UIDATA</td><td><input type="checkbox"/> OT_FAVORITE</td><td><input type="checkbox"/> OT_SINGLEDESTINATION</td></tr> <tr><td><input type="checkbox"/> OT_PROGRAM</td><td><input type="checkbox"/> OT_XMENODES</td><td></td><td></td></tr> </table>				<input type="checkbox"/> OT_ENUM	<input type="checkbox"/> OT_ROOT	<input type="checkbox"/> OT_DOM	<input type="checkbox"/> OT_INTERFACE	<input type="checkbox"/> OT_HSSDP	<input type="checkbox"/> OT_MESSAGE	<input type="checkbox"/> OT_CHANNEL	<input type="checkbox"/> OT_CHANNELS	<input type="checkbox"/> OT_DP	<input type="checkbox"/> OT_DPS	<input type="checkbox"/> OT_TIMERDP	<input type="checkbox"/> OT_CALENDARDP	<input type="checkbox"/> OT_VARDP	<input type="checkbox"/> OT_MAPDP	<input type="checkbox"/> OT_COMDP	<input type="checkbox"/> OT_DESTINATION	<input type="checkbox"/> OT_IPDP	<input type="checkbox"/> OT_UPNPD	<input type="checkbox"/> OT_KNXDP	<input type="checkbox"/> OT_OCEANDP	<input type="checkbox"/> OT_RFDP	<input type="checkbox"/> OT_IRDP	<input type="checkbox"/> OT_DEVICES	<input type="checkbox"/> OT_HISTORYDP	<input type="checkbox"/> OT_USER	<input type="checkbox"/> OT_USERS	<input type="checkbox"/> OT_SCHEDULER	<input type="checkbox"/> OT_USERPAGE	<input type="checkbox"/> OT_DEVICE	<input type="checkbox"/> OT_INTERFACES	<input type="checkbox"/> OT_XMLNODE	<input type="checkbox"/> OT_POPCLIENT	<input type="checkbox"/> OT_RULE	<input type="checkbox"/> OT_RULES	<input type="checkbox"/> OT_CONDITION	<input type="checkbox"/> OT_SINGLECONDITION	<input type="checkbox"/> OT_ALARMDP	<input type="checkbox"/> OT_UIDATA	<input type="checkbox"/> OT_FAVORITE	<input type="checkbox"/> OT_SINGLEDESTINATION	<input type="checkbox"/> OT_PROGRAM	<input type="checkbox"/> OT_XMENODES		
<input type="checkbox"/> OT_ENUM	<input type="checkbox"/> OT_ROOT	<input type="checkbox"/> OT_DOM	<input type="checkbox"/> OT_INTERFACE																																												
<input type="checkbox"/> OT_HSSDP	<input type="checkbox"/> OT_MESSAGE	<input type="checkbox"/> OT_CHANNEL	<input type="checkbox"/> OT_CHANNELS																																												
<input type="checkbox"/> OT_DP	<input type="checkbox"/> OT_DPS	<input type="checkbox"/> OT_TIMERDP	<input type="checkbox"/> OT_CALENDARDP																																												
<input type="checkbox"/> OT_VARDP	<input type="checkbox"/> OT_MAPDP	<input type="checkbox"/> OT_COMDP	<input type="checkbox"/> OT_DESTINATION																																												
<input type="checkbox"/> OT_IPDP	<input type="checkbox"/> OT_UPNPD	<input type="checkbox"/> OT_KNXDP	<input type="checkbox"/> OT_OCEANDP																																												
<input type="checkbox"/> OT_RFDP	<input type="checkbox"/> OT_IRDP	<input type="checkbox"/> OT_DEVICES	<input type="checkbox"/> OT_HISTORYDP																																												
<input type="checkbox"/> OT_USER	<input type="checkbox"/> OT_USERS	<input type="checkbox"/> OT_SCHEDULER	<input type="checkbox"/> OT_USERPAGE																																												
<input type="checkbox"/> OT_DEVICE	<input type="checkbox"/> OT_INTERFACES	<input type="checkbox"/> OT_XMLNODE	<input type="checkbox"/> OT_POPCLIENT																																												
<input type="checkbox"/> OT_RULE	<input type="checkbox"/> OT_RULES	<input type="checkbox"/> OT_CONDITION	<input type="checkbox"/> OT_SINGLECONDITION																																												
<input type="checkbox"/> OT_ALARMDP	<input type="checkbox"/> OT_UIDATA	<input type="checkbox"/> OT_FAVORITE	<input type="checkbox"/> OT_SINGLEDESTINATION																																												
<input type="checkbox"/> OT_PROGRAM	<input type="checkbox"/> OT_XMENODES																																														
<b>Vergleich mit ID Liste</b> <p><input checked="" type="radio"/> Darf nicht vorhanden sein <input type="text" value="----"/></p> <p><input type="radio"/> Muss vorhanden sein</p>																																															

Eingabe des Scan Bereiches der IseID's (hier von 1-65535)

Achtung

Schrott Eingabe von Millionenwerten werden die CCU lahmlegen. Der SDV ist schließlich kein Spielzeug sondern ein Werkzeug, man sollte schon wissen, was man tut.

Damit ein Objekt Selektiert wird, muss es die angeklickte Objekteigenschaft haben.

Mehrfachangaben sind möglich

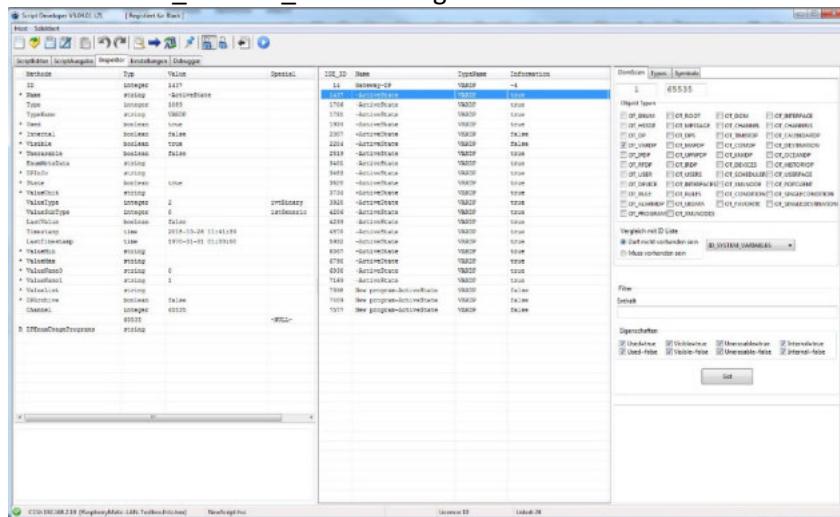
Beispiel für Suchen aller Objekte mit der Eigenschaft OT\_DEVICE im Bereich der ISE\_Nummern 1-65535

The screenshot shows the Script Developer V3.08.1 interface with the 'DomScan' dialog open. The 'DomScan' tab is selected, showing the range '1' to '65535'. The 'Object Types' section lists various object types with checkboxes. The 'Vergleich mit ID Liste' section has a radio button for 'Darf nicht vorhanden sein' (must not exist) and a dropdown set to '----'. The results table on the left shows several entries, and the right side shows the selected object types and a 'Get' button.

Anklicken eines Wertes in der Listendarstellung öffnet die Detaildarstellung des Objektes.

Ebenso ist es möglich, im DomScan Bereich Einträge zu suchen, welche beispielsweise nicht in den Aufzählungen gelistet sind.

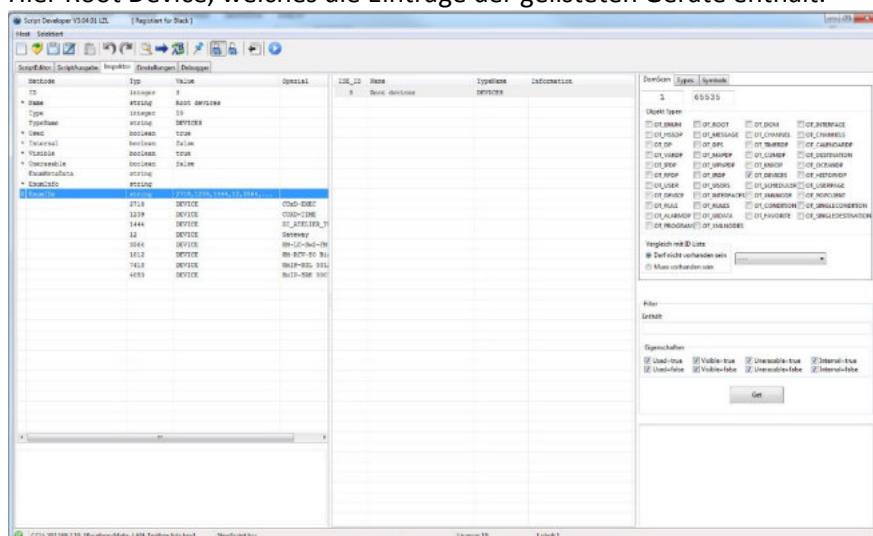
Hier Beispieldeweise: Scanlauf über alle Objekte aus DOM mit der Eigenschaft VARDP, die aber nicht unter ID\_SYSTEM\_VARIABLES gelistet sind:



Hier tauchen dann einige interne Datenpunkte auf, im dem Falle sind die –ActiveState keine Leichen, sondern der Anwahl Punkt Programm aktiv unter Programme.

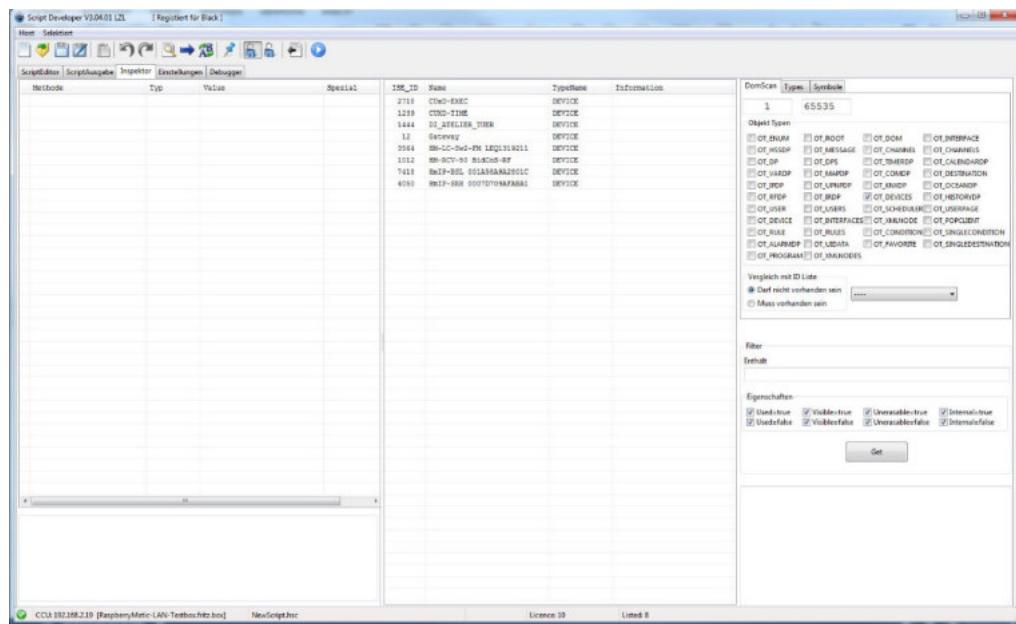
Weiterhin ist auch rekursives Arbeiten nun möglich

Hier Root Device, welches die Einträge der gelisteten Geräte enthält.

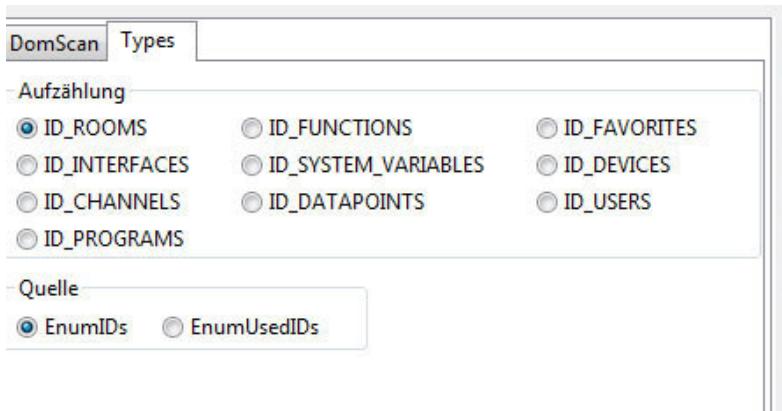


Das „R“ in der ersten Spalte zeigt an, dass ein Rekursiver Aufruf möglich ist. Doppelklicken löst und die Liste auf und lädt diese in die Listenauswahl. Ist es nur ein Wert, so wird dieser Wert auch direkt in der Detailansicht geöffnet.

Die neue Auswahllist steht nun zur weiteren Bearbeitung bereit.



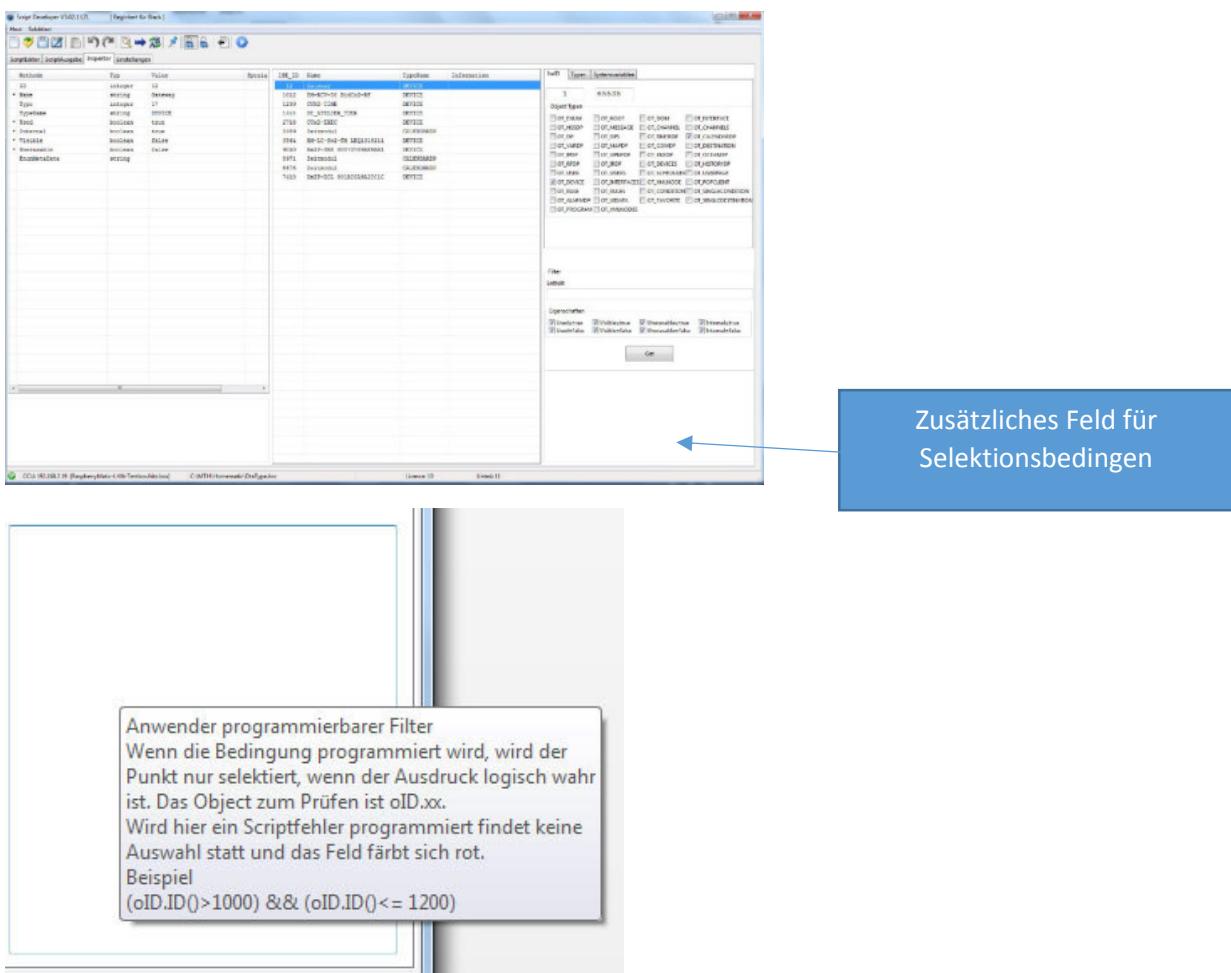
## 4.2 Selektionskriterium Types



Hierbei wird wie schon in der Version 3.2 in festen Bereichen gesucht und selektiert. Schneller und einfacher zu handeln als die Objekt Selektion, dafür nicht so umfangreich.

### 4.3 Zusätzliche Selektionsbedingungen

Durch Druck auf Get wird die Liste gemäß Selektion von der CCU angefordert, aufbereitet und dargestellt. (Lizenzlevel vorausgesetzt)



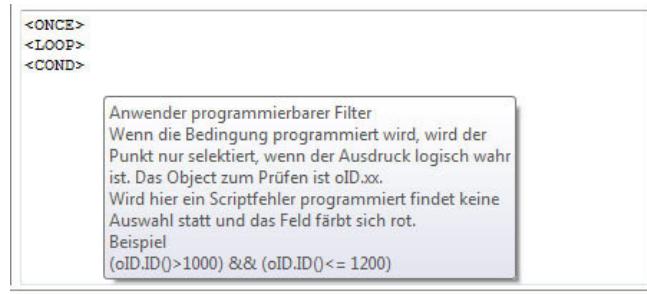
Filter sind ein mächtiges Werkzeug zum komplexen Eingrenzen und für Komplexe Abfragen.

Für die Filter existiert mittlerweile ein Kontext Menü mit rechter Maustaste:



Filter löschen entfernt sämtliche Filterbedingungen

Neuer Filter legt von der Syntax einen neuen, leeren Filter an



Mit Filter laden und speichern lassen sich nun Anwenderfilter als \*.flt Datei im Verzeichnis des SDV abspeichern.

Ein Filter besteht aus den 3 Abschnitten:

<ONCE> Der Text dahinter wird am Anfang des internen Abfragescriptes quasi im einmaligen Durchlauf eingefügt. Normalerweise stehen hier Definitionen, welche nicht bei jedem Durchlauf aktualisiert werden müssen

<LOOP> Der Text dahinter wird im Zyklischen Durchlauf des Programmes innerhalb der Programmschleife eingefügt.

<COND> der Text hinter COND wird in die IF Abfrage eingefügt, welche letztlich das Objekt zur Darstellung in der Liste selektiert.

Vereinfachter Ablauf: so sieht vereinfacht das Listenselektionsprogramm aus:

```
object oID;
string s;
foreach (s,Schleifenbedingung) {
oID= dom.GetObject (s);
if (oID) {
    if (ElementBedingung) {WriteLine („Element in Liste: „ # oID.ID () );
}
}
```

Ein Anwenderdefinierter Filter wird dann in diese Grundschleife so eingebaut:

```
object oID;
string s;
ONCETEXT;

foreach (s,Schleifenbedingung) {
oID= dom.GetObject (s);
if (oID) {
    LOOPTEXT;
    if (ElementBedingung && (CONDTEXT)) {WriteLine („Element in Liste: „ # oID.ID () );
}
}
```

An diesem Kleinen Filter mal verdeutlicht:

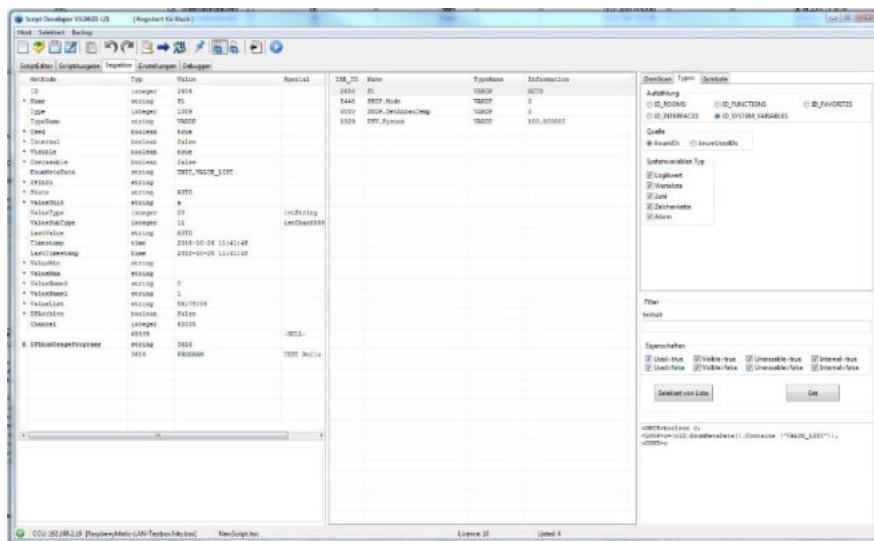
```
<ONCE>boolean c;  
<LOOP>c=(oID.EnumMetaData().Contains ("VALUE_LIST"));  
<COND>c
```

Daraus generiert der Scriptdeveloper folgende interne Filterabfrage:

```
object oID;
string s;
boolean c;

foreach (s,Schleifenbedingung) {
oID= dom.GetObject (s);
if (oID) {
    c=(oID.EnumMetaData().Contains ("VALUE LIST"));
    if (ElementBedingung && (c)) {WriteLine („Element in Liste: „ # oID.ID ());
}
}
```

Filtert aus der Gruppe der Systemvariablen alle, in deren Eigenschaft EnumMetadata das Wort VALUE\_LIST vorkommt.



So lassen sich dann Filter in epischer Komplexität basteln, die man über die RegaDom stülpen kann. Zu beachten, die folgenden Variablennamen sind schon Intern vorbelegt:

**object** **old:** darf benutzt werden, ist der Bezug auf das Objekt, welches im Filter überprüft werden soll

var y: intern benutzt zur Typerkennung: Fingers weg

string slInfo: intern benutzt zur Listengenerierung; Ein

boolean b: interner Fildar, auch Finger weg

boolean b: interner Finder, auch Finger weg  
string done: auch interne Benutzung, auch

string done: auch interne Benutzung, auch Finger weg

Die Filterbedingung wird in HM Script ausformuliert. Das gefundene Object kommt nur in die Liste, wenn die ausformulierte Bedingung True ist. Das Teil ist mächtig, aber auch nicht ungefährlich, man kann auch Müll als Bedingung schreiben. Dabei kommt dann aber eine Warnung:

```
<ONCE>boolean c;  
<LOOP>c=(oID.EnumMetaData().Contains ("VALUE_LIST");  
<COND>c
```

Anwender progr  
Wenn die Bedin  
Punkt nur selekt  
ist. Das Object zu

Bedingung ist falsch: erzeugt Scriptfehler  
(Klammer zu fehlt). In dem fall färbt sich nach  
Druck auf Get das Feld rot

Der rar Datei liegen Standardmäßig nun schon mal 2 Filter bei:

PROGRAM\_GeisterProg\_CopyID - Filter um Geisterprogramme mit gesetzter CopyID zu finden

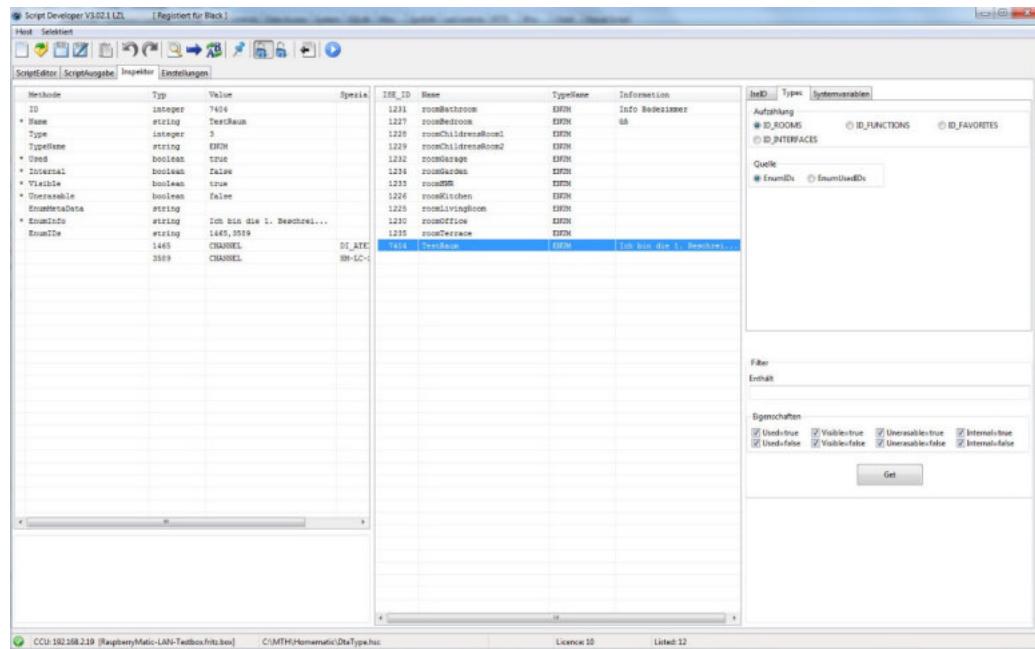
SYSVAR\_VerwaisterChannel – Filter um Systemvariablen zu finden, deren Channel verweis in Nirvana zeigt

Durch Click auf die Beschreibungszeile IsID bzw Name können die Felder entsprechend sortiert werden.

Click auf eine selektierte Aufzählung öffnet im Detailfenster die Methodenansicht des Objektes

### Changelog V3.03xx

Da die internen Sortieralgorithmen suboptimal arbeiteten, hat das ListView Object neue selektive Sortieralgorithmen bekommen. IsID sortiert nun wie man erwartet nach Integer aufsteigend, Name sortiert alphabetisch aufsteigend, TypeName sortiert alphabetisch, sind die Typenames gleich, wird innerhalb gleicher Typenames nach IsID numerisch sortiert.



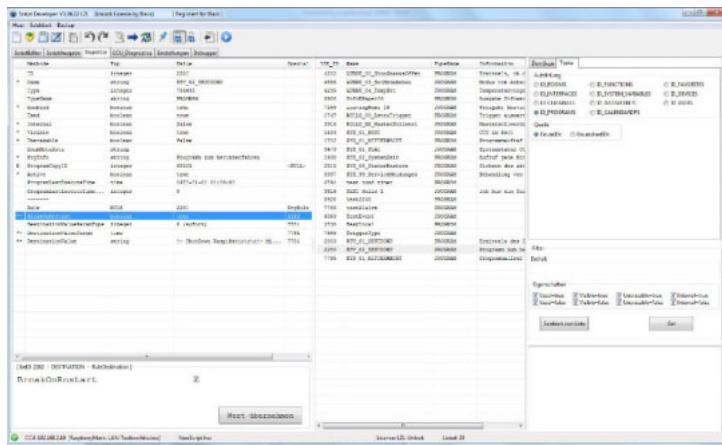
Die Spaltenbreite kann sowohl in Normalansicht als auch in Maximize separat eingestellt werden  
(Das Programm sollte sich die Breiten merken und je nach Darstellungsart automatisch wieder einstellen, sollte...)

Dargestellt werden die Methode, der Vartype und die Property.

Bei den Aufzählungen wird jeweils eine Rekursionsstufe aufgelöst, um an die Detailinformationen zu kommen. Hier die Liste der Channels, die diesen Raum verwenden, aufgelöst in die ID, der Typ (hier Channels und der Name des Channels)

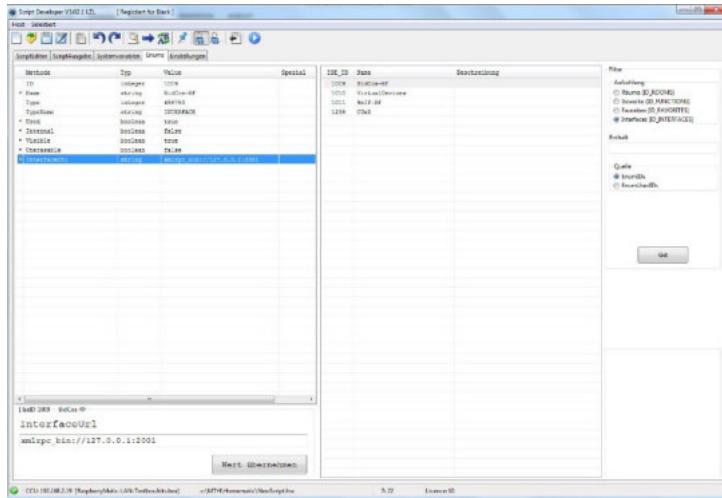
Properties, die in der ersten Zeile mit einem Stern (\*) gekennzeichnet sind, können in ihrem Wert geändert werden.

Ab version 3.06.04: Properties, die in der ersten Zeile mit einem Doppelstern (\*\*\*) gekennzeichnet sind, können ebenfalls verändert werden. Der Doppelstern bedeutet dabei, die geänderte Property nicht element der Haupt ID ist, sondern sich rekursiv in einer untergeordneten Rekursionsebene befand.

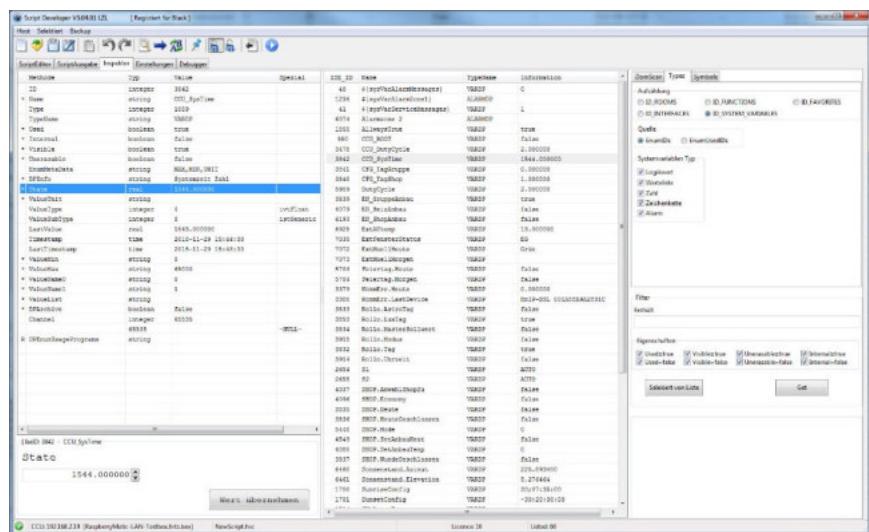


BreakOnRestart ist hier nicht Member von OT\_PROGRAM sondern rekursiv von der entsprechenden Rule bzw SubRule.

Dazu auf die Zeile klicken



Nach Click auf Wert übernehmen wird der Wert in der CCU geändert. Also Vorsichtig mit dieser Funktion umgehen, hier gibt es kein redo.

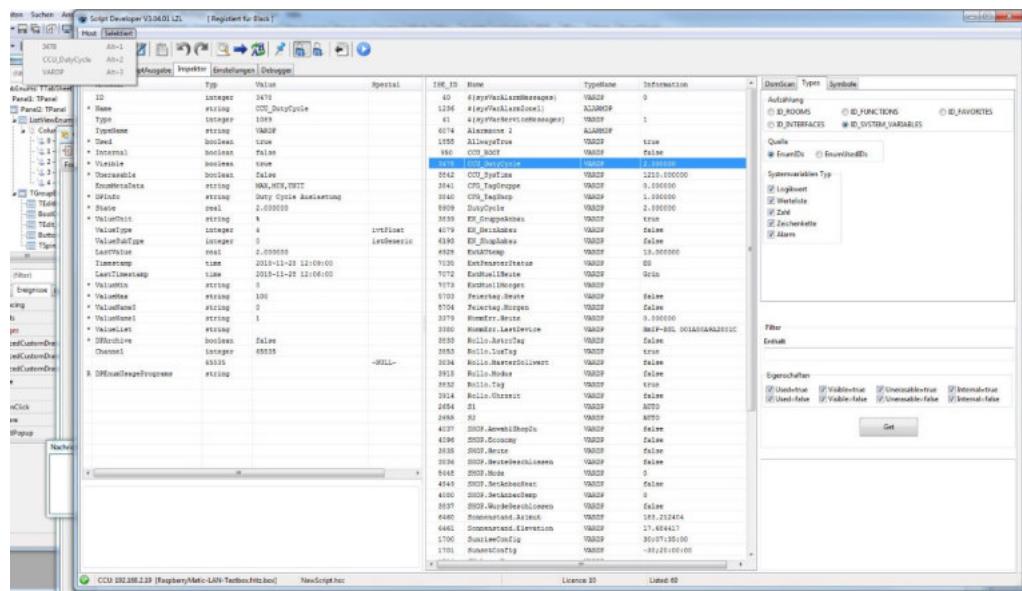


## 4.4 Daten aus Inspektor in Editor übernehmen

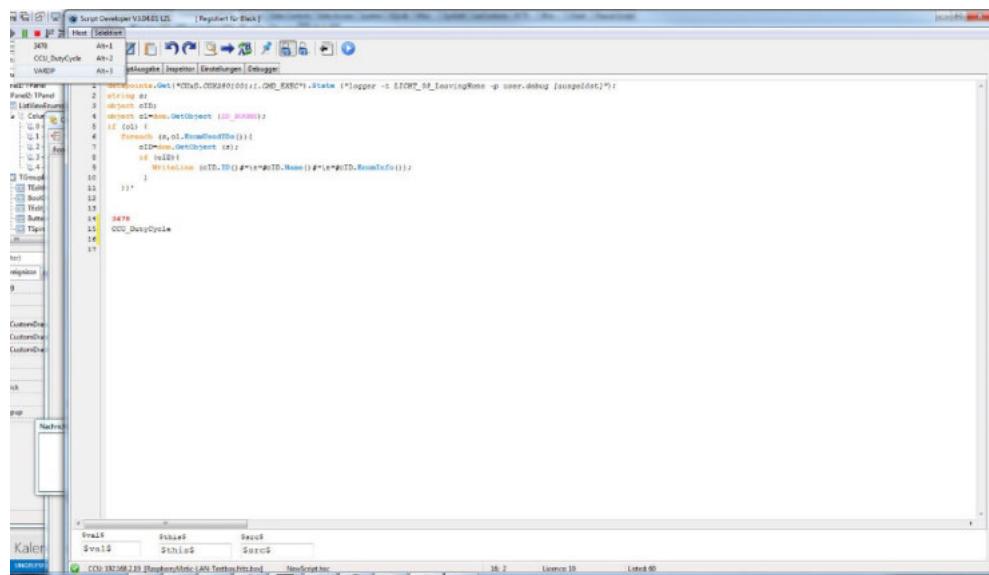
Zur schnelleren und auch möglichst fehlerfreien Bearbeitung besteht die Möglichkeit, Daten aus dem Inspektor direkt in den Editor zu übernehmen.

Immer wenn im Inspektor in den beiden Listviews auf eine Eigenschaft geklickt wurde, stehen diese Daten dann im Editor unter Selektiert zu verfugung.

Hier Klick auf die Systemvariable



Unter selektiert sind die Eigenschaften herausgefiltert worden und lassen sich im Editor entweder durch das Menü selektiert oder durch die Kurztasten Alt-1: ID, Alt-2: Name und Alt 3: Eigenschaft bzw. Methode einfügen.



Im Detailauswahlfeld wird bei klicken auf die Methode auch noch der Methodenname gespeichert, der sich dann auch durch Alt-2 einfügen lässt

Ab 3.06.01

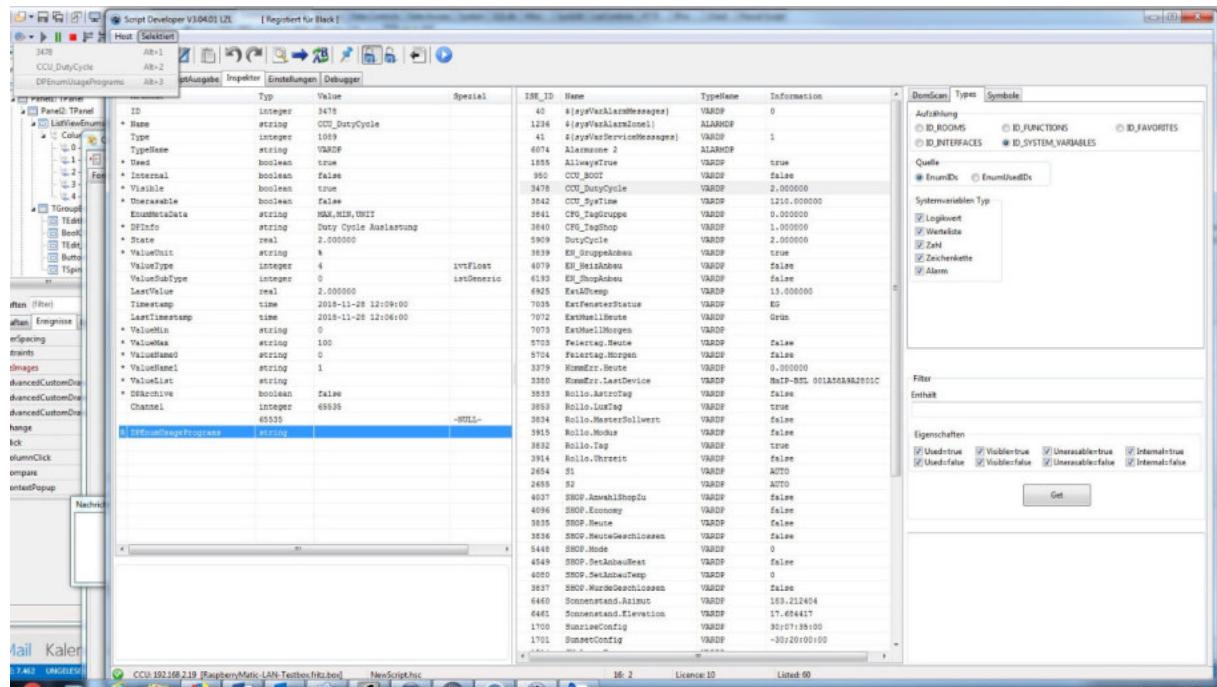
Mit den Tasten lassen sich folgende Spalteninhalte, welche vorher im Detailfeld selektiert worden sind, in den Editor übernehmen

Alt+m: Inhalt von Methode

Alt+t: Inhalt von typ

Alt+v: Inhalt von Value

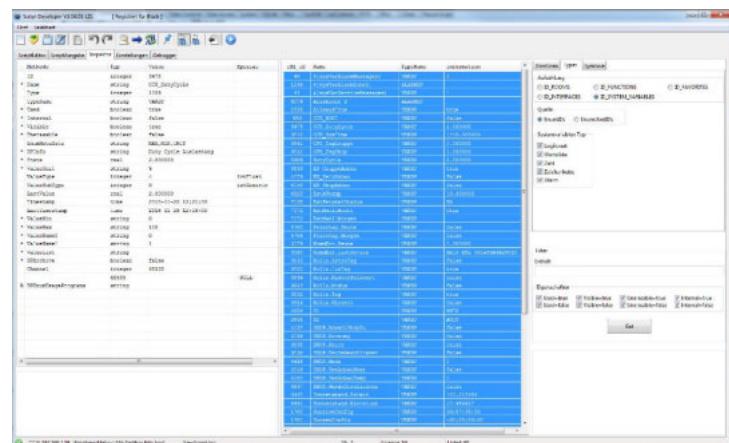
Alt+p: Inhalt von Spezial



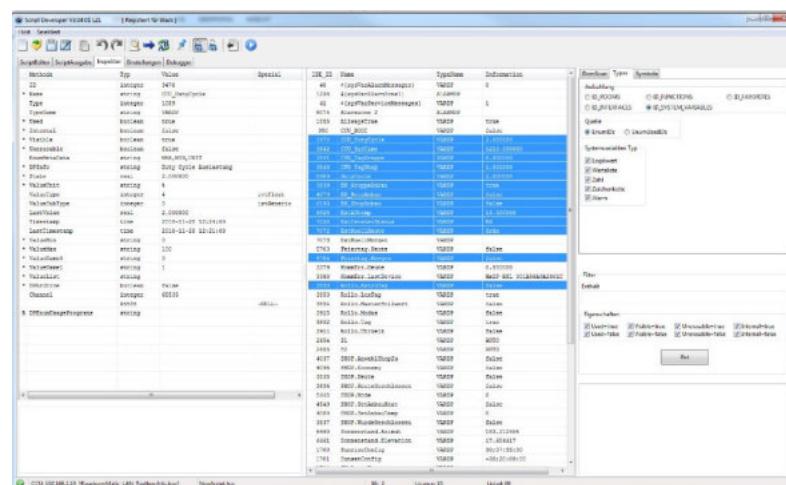
#### 4.4.1 Mehrfachauswahl als Enum String

Es lassen sich im Hauptauswahlfeld Mehrfachselektionen vornehmen.

**Ctrl-A : alle auswählen.**



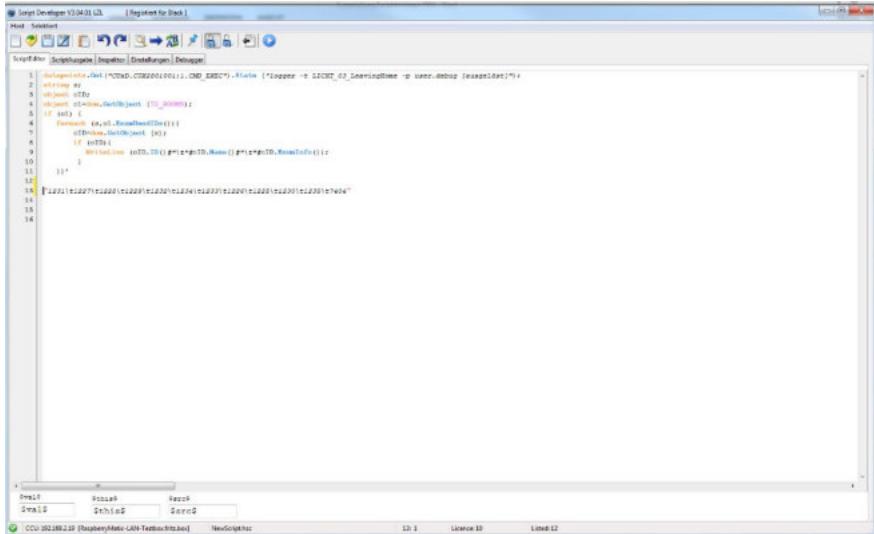
Oder die Übliche Mausbedienung:



Ab Version 3.07.07.

In der Listendarstellung kann nun mit den Cursortasten navigiert werden, ein Druck auf Return öffnet von dem selektierten Objekt die Detaildarstellung.

Mit der Taste markt sich der Inspektor die Auswahl, welche sich dann Script Konform im Editor Als ID-Enum durch die Einfügen Clipboard Taste, welche nach dem Pin Druck nicht mehr grau ist, lassen sie die selektierten ID,s im Editor einfügen (z.B. zur Verarbeitung in einem Script als foreach)



Die Pinliste funktioniert nicht nur mit dem Editor, auch im Inspektor lässt sich eine mit dem Pin gemerkte Selektionsliste wieder in die Auswahl laden:

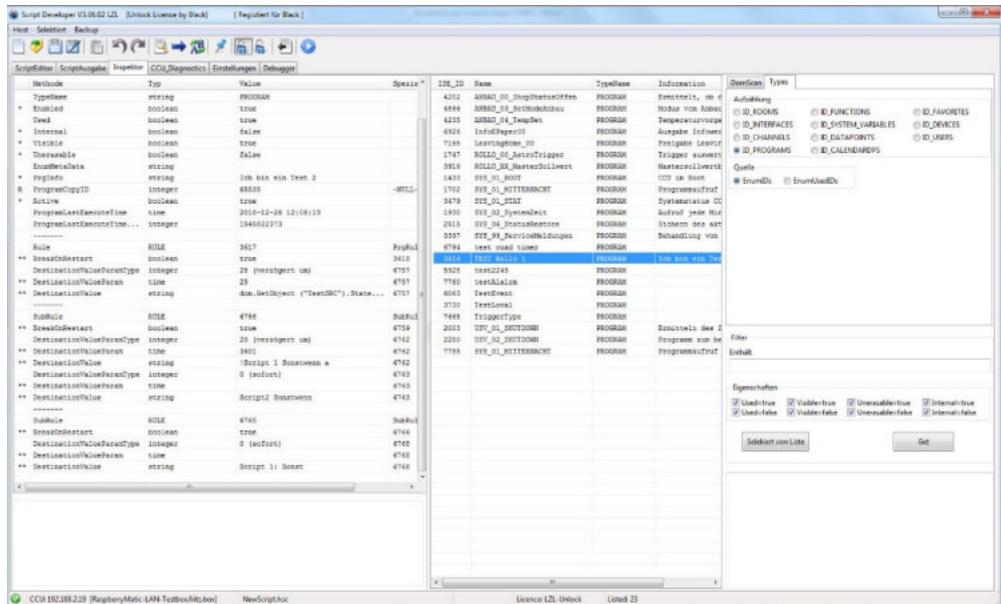
Mit rechter Maustaste im Mittleren Feld die Funktion „Einfügen aus Pinliste“ anwählen und die Sicherheitsabfrage bestätigen,



dann befinden sich die Selektierten Elemente wieder im mittleren Feld.

## 4.4.2 Übernahme von einem Script aus einem Programm direkt in den Editor

In der Darstellung eines Programmes werden in der Detaildarstellung auch die Rules und Subrules mit ihren Destinations/SingleDestinations aufgelöst, wenn diese ein Script enthalten



Dies entspricht der tabellarischen Darstellung dieses Programmes

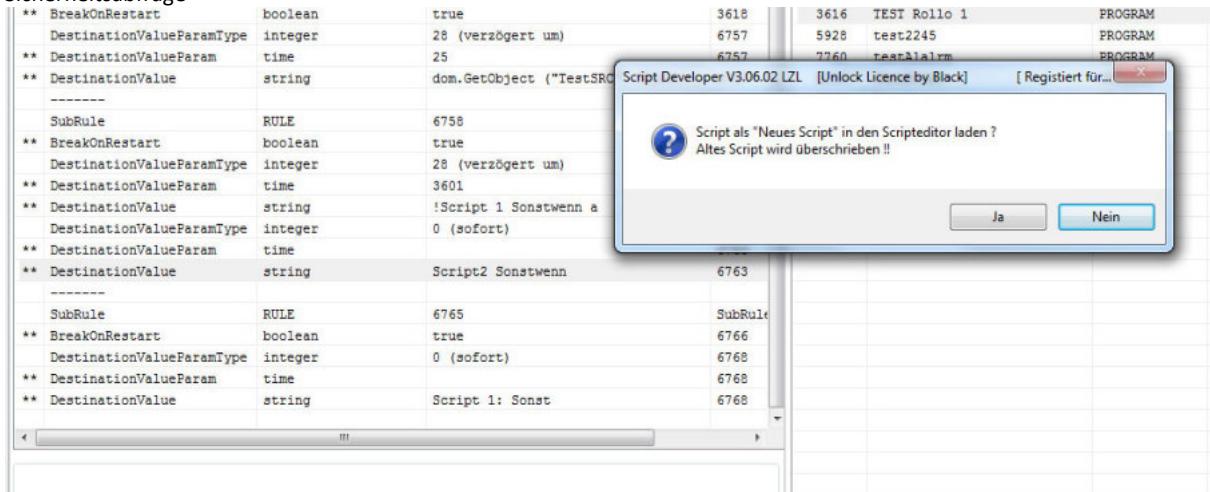
The screenshot shows the RaspberryMatic software interface under 'Programme und Verknüpfungen > Programme > Programmherstellung'. It displays a complex logic structure for a 'TEST Rollo 1' program. The logic includes several 'Bedingung: Wenn...' (Condition: If...) blocks and 'Aktivität: Dann...' (Action: Then...) blocks. One rule involves setting a state to 'S1 sofort' and then performing an action. Another rule involves setting a state to 'CFC\_TapGruppe' and then performing an action. The interface also shows 'Systemzustand' (System State) and 'Aktion' (Action) columns. At the bottom, there are buttons for 'Abbrechen' (Cancel), 'OK', 'Einstellungen als neues Programm speichern' (Save settings as new program), 'Skript testen' (Test script), and 'Drucken' (Print).

Rule entspricht hier der DANN Aktivität, dort wurde auch das Script gefunden, welches mit dom.GetObject („TESTSCR.....“) beginnt

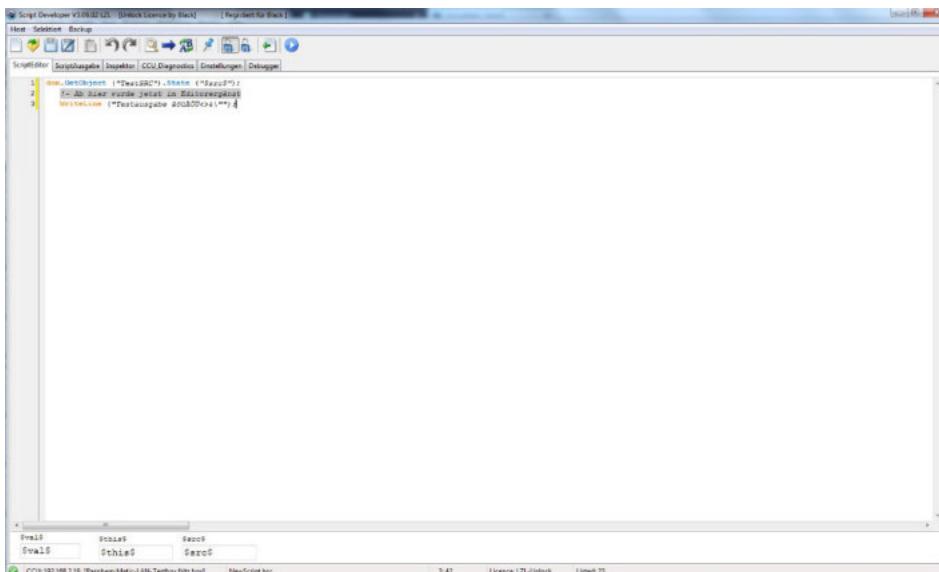
Die erste Subrule enthält dann die beiden Scripte aus der SonstWenn Bedingung.

Die Letzte Subrule entspricht der Sonst Bedingung. Nach dieser Auflistung lassen sich die Scripte tabellatisch im SDV wiederfinden. Zur Hilfe wird in der Spalte Value die ersten 70 Zeichen des Scriptes dargestellt.

Klick auf den Doppelstern (Doppelstern= Element wurde rekursiv aus dem Hauptelement aufgelöst) führt nach einer Sicherheitsabfrage



Zum Laden des Scriptes in den Scripteditor. Dort können dann die Änderungen durchgeführt oder Tests gemacht werden.

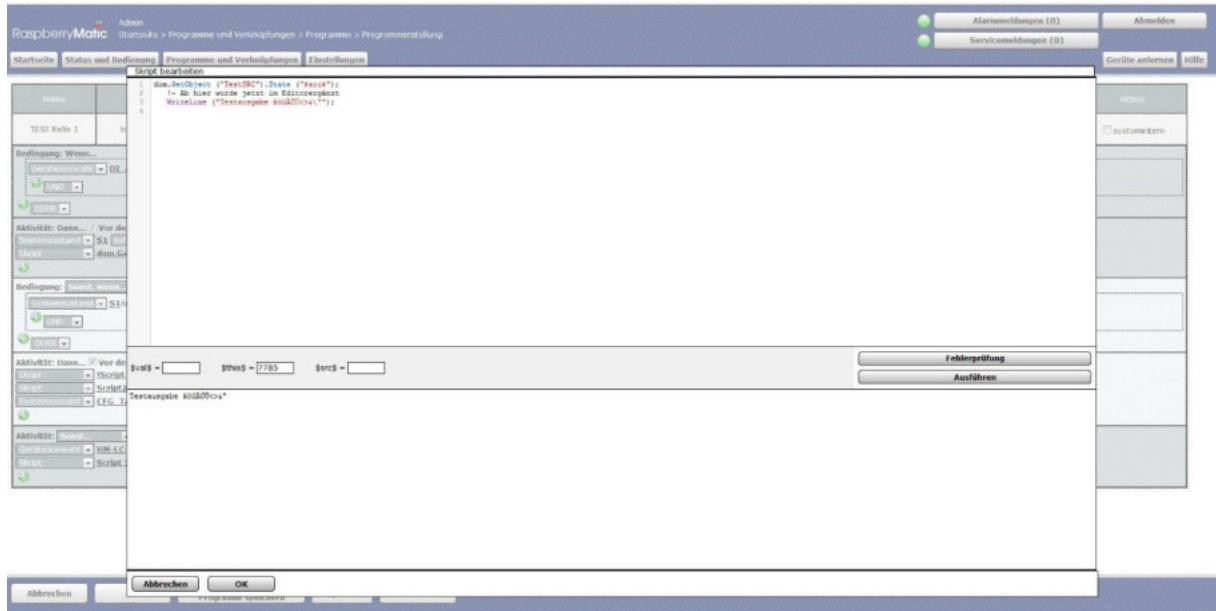


Danach kann durch Drücken der nun nicht mehr grauen Taste



Das Programm aus dem Editor direkt in das Programm der CCU geladen werden. Im Hint wird als Gedächtnissstütze der Name des Programmes mit angezeigt. Einschränkung: wird im Inspektor nun rumgeklickt und eine andere Property angewählt oder etwas anderes Selektiert. Wird laden in CCU Programm wieder grau (Grund: interner Bezug geht dann natürlich verloren) Dann ist kein direkter Upload mehr möglich (nur über im Editor CtrlA-CtrlC, im Inspektor neu Selektieren und laden des Scriptes erneut in den Editor, im Editor Ctrl-A, Del, Ctrl-V, Upload).

Auf der CCU lässt dich dann das geänderte und wieder uploadete Script öffnen, überprüfen und auch Ausführen.

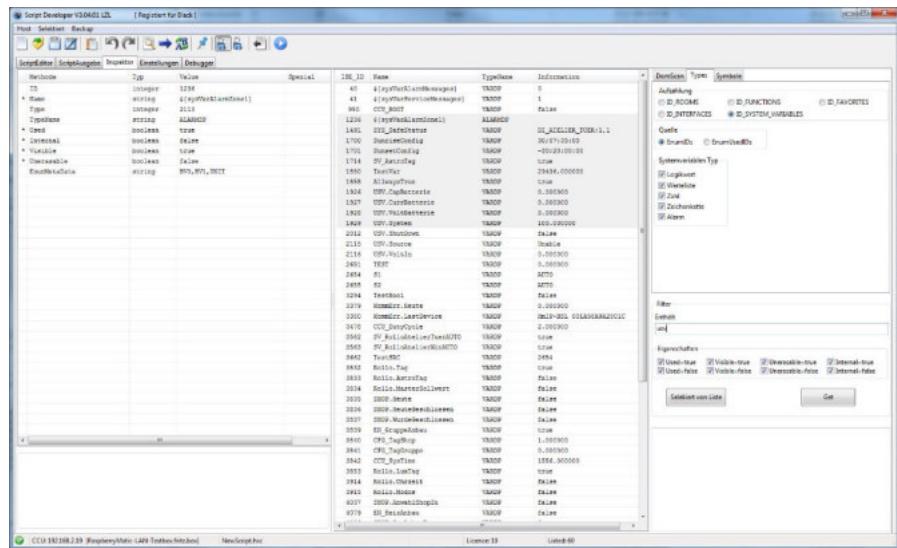


Das Programm würde nun das neue, geänderte Script bei Triggerung ausführen.

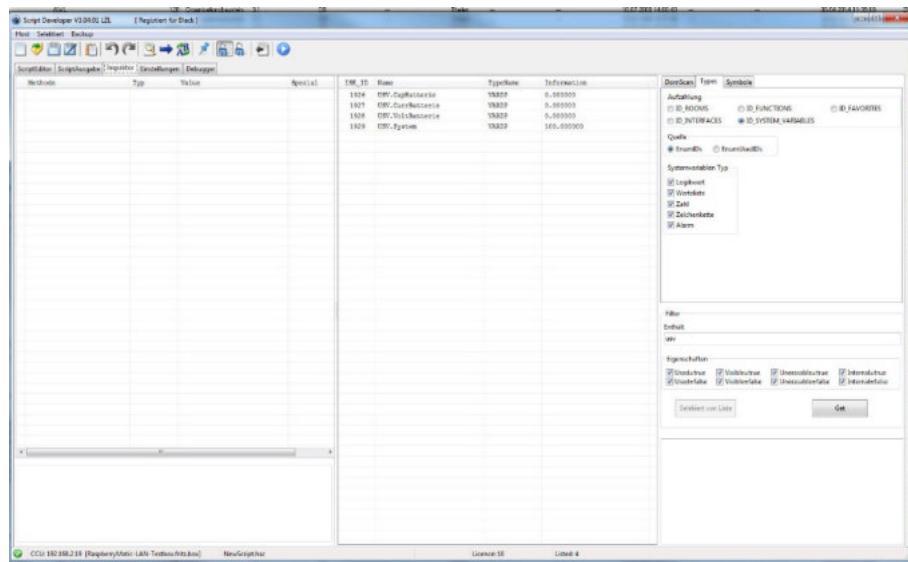
## 4.5 Selektion von Selektion

Befinden sich Daten in der Listendarstellung, so können daraus Bereiche selektiert werden und über diesen manuell selektierten Bereich die Auswahlfilter geschickt werden.

Hier Beispiel



Selektierter Bereich von Systemvariablen, die hier darauf gefiltert werden sollen, dass der Name den String „usv“ enthält. Es müssen 4 Sysvars gefunden werden, die IDS 2012,2115,2116 werden hier nicht berücksichtigt, da diese sind selektiert sind. Bei <Druck auf: Selektiert von Liste: ergibt sich dann

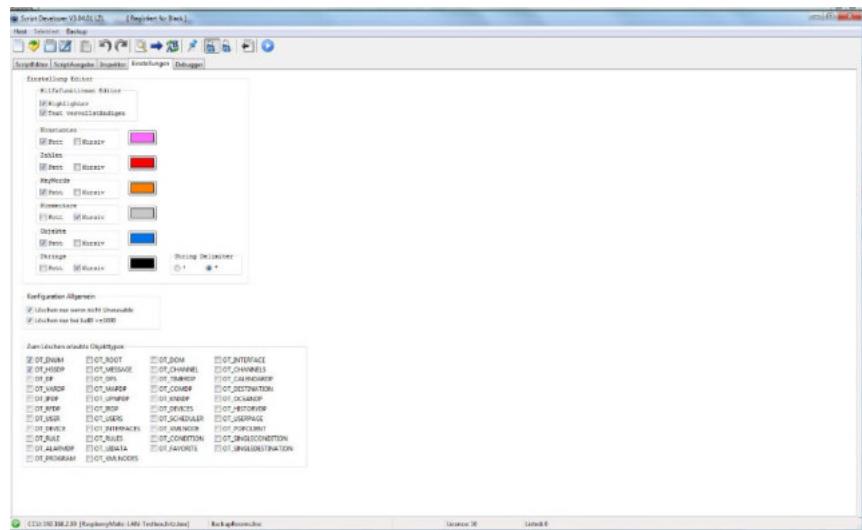


## 4.6 Objekte löschen

Objekte können vom SDV direkt auf der CCU gelöscht werden. Die Verantwortung, welche Objekte gelöscht werden, obliegt dem jeweiligen Anwender. Für die Löschfunktion gibt es KEIN Redo. Bevor derartige Bearbeitungen gemacht werden, IMMER vorher ein Backup machen.

Redo geht nur über restore !

Um Versehentliches löschen zu verhindern, sind ein paar Schutzmechanismen eingebaut. Generell sind Löschfunktion blockiert, wenn das Schloss in der Menüleiste auf zu steht. Um Löschen generell Freizugeben muss das schloss auf „Offen“ stehen.



Unter Einstellungen befinden sich noch ein paar Einstellungen, die Löschmöglichkeiten eingrenzen:

Löschen nur wenn nicht Unerasable: Jedes Objekt auf der CCU hat eine Property namens unerasable. (unlösbar) Ist der Haken gesetzt, geht löschen nur wenn das Objekt nicht auf unerasable = checked steht. Um nicht löscharbe Elemente zu löschen entweder:

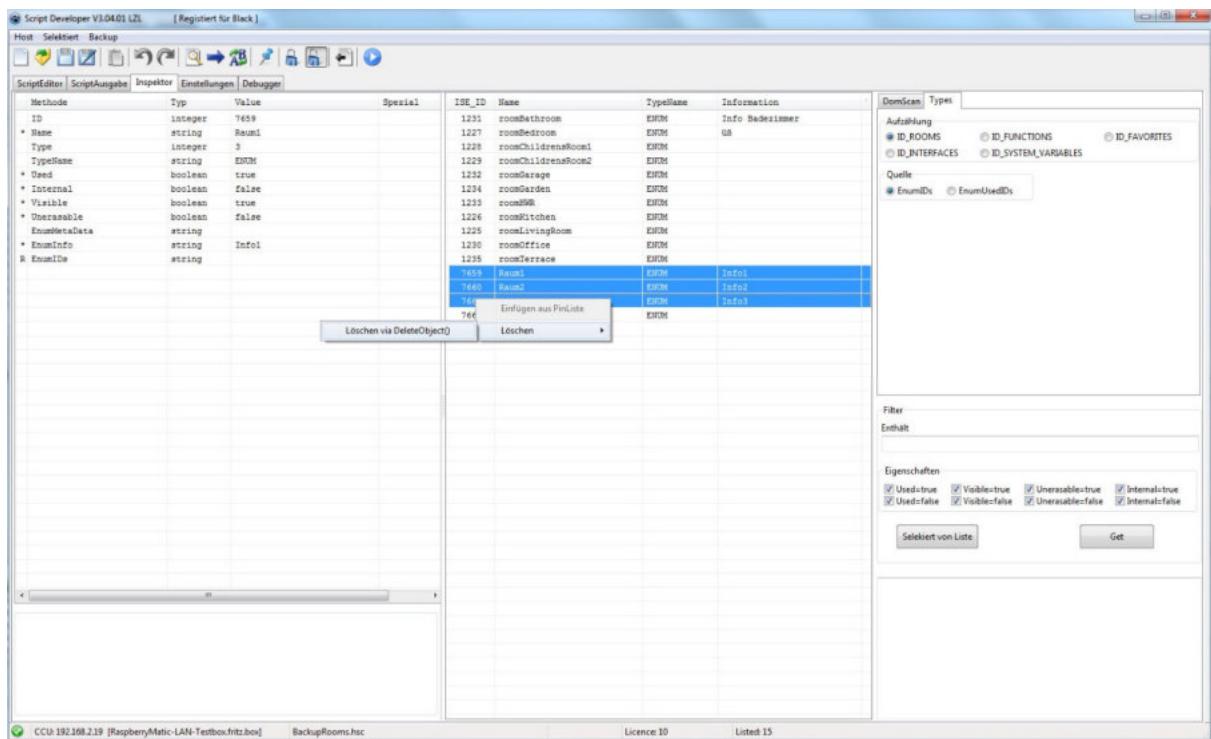
Im Inspektor unter Detailsview die Property entfernen (gilt nur für das Objekt), oder hier den Haken wegmachen (gilt für alle)

Löschen nur wenn ID $\geq$ 1000. Dieser haken verhindert, dass man versehentlich Interne IDs der CCU (normalerweise unter kleiner 1000 angelegt) löscht. Will man in dem Bereich löschen, muss der hier explizit manuell unchecked werden.

Die Einstellungen werden NICHT gespeichert, bei jedem Neustart des SDV sind diese beiden Einstellungen wieder checked.

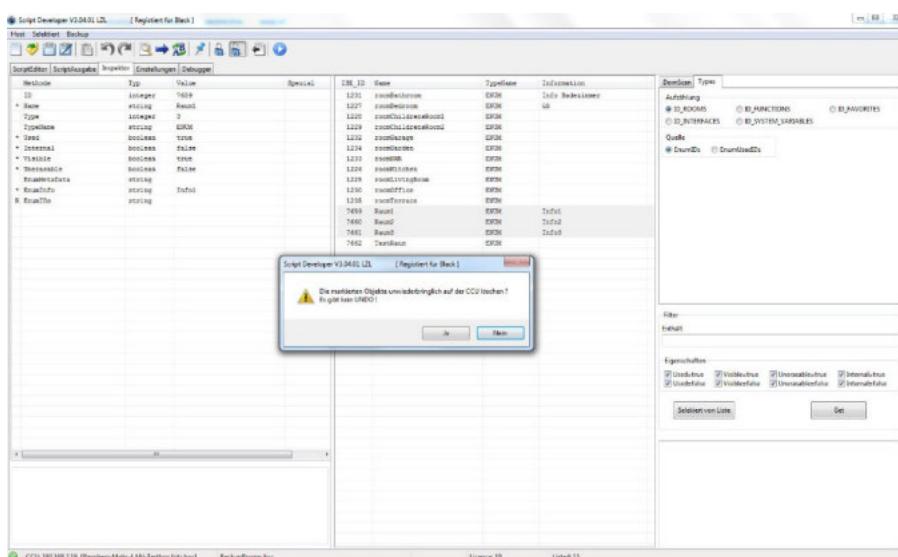
Löschen von Objekten. Die Letzte Sicherheit: ein zu lösches Objekt muss einen hier gecheckten Objekttyp haben, sonst wird es nicht gelöscht.

Löschen läuft so ab:



Objekte filtern und markieren, rechte Maustaste, Löschen, Löschen via DeleteObject ()

Mehrfachselektion ist möglich

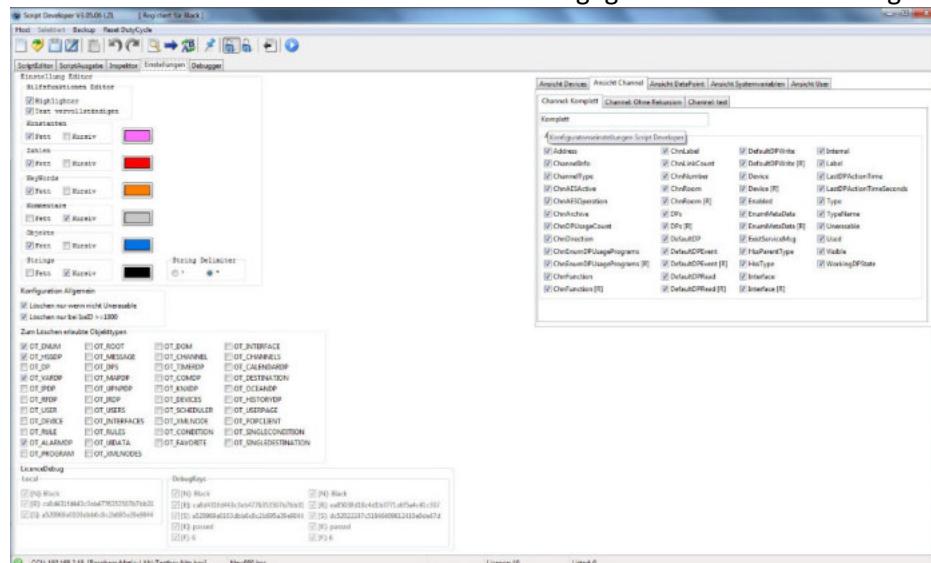


Nach dieser Sicherheitsabfrage sind die Objekte dann weg.. Zurück geht's dann nur mit Restore.

## 4.7 Anwenderdefinierte Sichten

Die Detailansichten können stellenweise sehr umfangreich sein und auf den ersten Blick mit Information zuwerfen. Deshalb ist es möglich, für manche Objekte drei Anwenderspezifische Sichten zu definieren.

Es werden dann in der Detailansicht nur die freigegebenen Methoden dargestellt.



Sichten können mit eigenen Namen versehen werden.

Die Entstellungen werden beim Verlassen gespeichert. Die jeweils geöffnete Sicht wird dann für das gefundene Objekt angewendet.

## 4.8 Browsing durch Rekursionsebenen

Seit der Version 3.06.04 verfügt der SDV über einen UNDO/REDO Stack im Inspektor. Dies bedeutet, dass immer, wenn in eine Rekursionsebene gesprungen wird, sich die Einträge im Selektionsfeld gemerkt werden und man über Undo / Redo dann zwischen den Ebenen hin und herspringen kann. Ein GET löscht dabei immer den Undo Stack

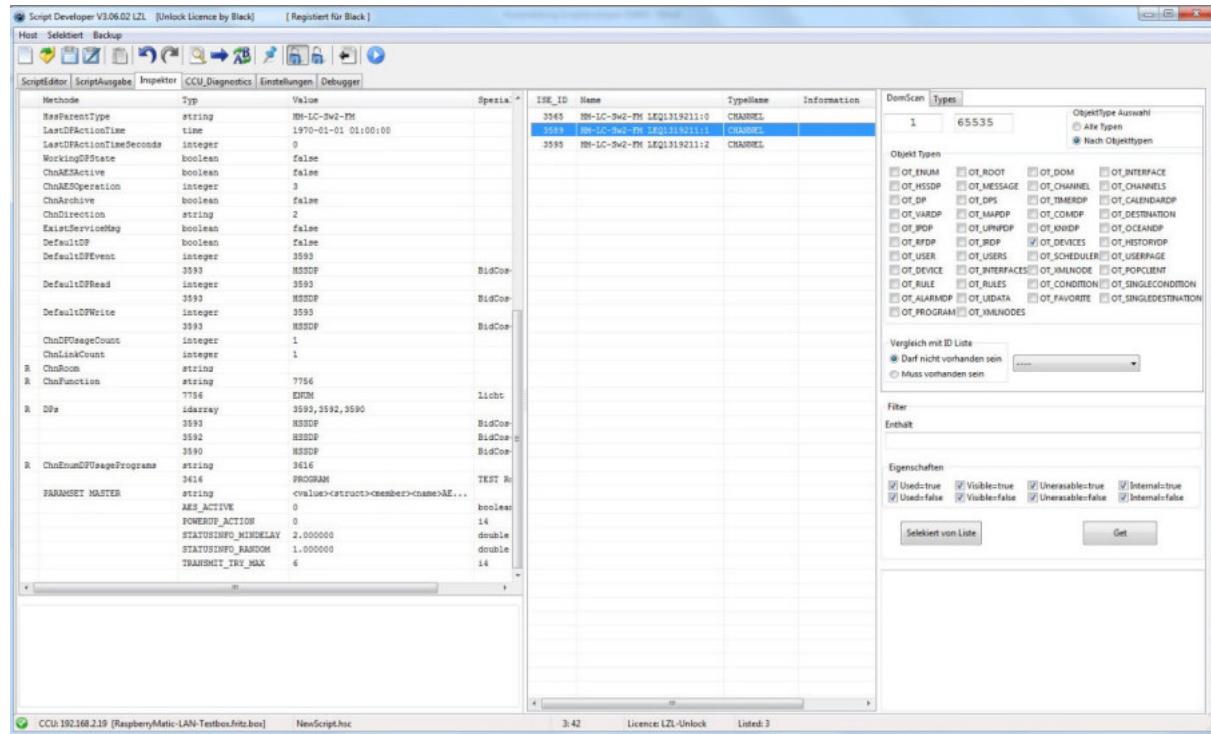
### Hier Beispiel Einsprung über RootDevices

Methode	Typ	Value	Spezial	IS_E_ID	Name	TypeName	Information
ID	integer	3					
Name	string	Root devices					
Type	integer	19					
TypeName	string	DEVICES					
Enabled	boolean	true					
Used	boolean	true					
Internal	boolean	false					
Visible	boolean	true					
Unreachable	boolean	false					
EnumMetaData	string						
EnumInfo	string						
R_EnumIds	string	2718,1239,1444,12,3564,1012,741..., 2718 1239 1444 12 3564 1012 7418 7624 7886 4050	DEVICE CUDX-TIME DL_ATELIER Gateway HM-LC-SwG HM-RCV-SC RaIP-BSL RaIP-SLO RaIP-SPI RaIP-SRR				

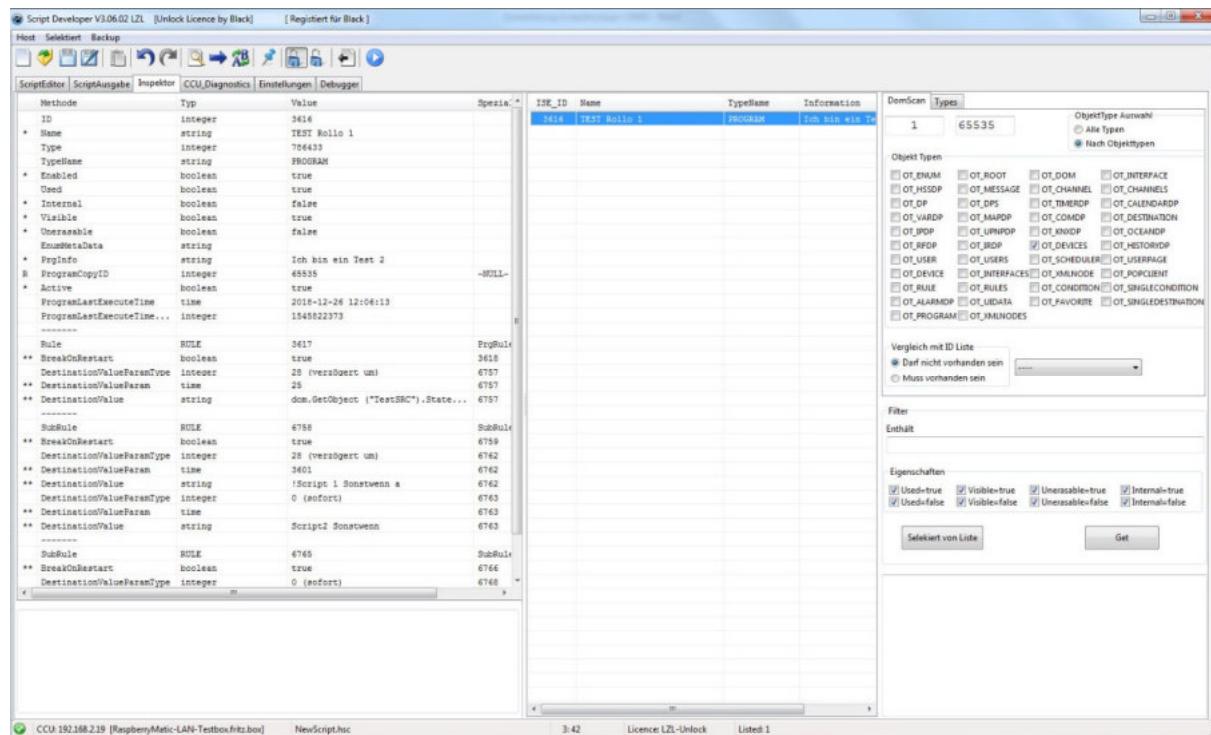
Rekursiv Aufgelöst die EnumIDs und Anwahl des 2 Fach Schaltaktors (Hier zu sehen, Undo ist schon anwählbar geworden)

Methode	Typ	Value	Spezial	IS_E_ID	Name	TypeName	Information
ID	integer	3564					
Name	string	HM-LC-Sw2-PW LEQ1319211					
Type	integer	17					
TypeName	string	DEVICE					
Enabled	boolean	true					
Used	boolean	true					
Internal	boolean	false					
Visible	boolean	true					
Unreachable	boolean	false					
EnumMetaData	string	AUTOCONF, DEVDESC, MASTERDESC, FAR..., DEVDESC MASTERDESC PARAMETERS enabledServiceMsg operateGroupOnly	string [ADDRESS:"LEQ1319211", CHILDREN:... string [CONF_BUTTON_TIME] [DEFAULT:255,... string MASTER true false				
DeviceInfo	string						
Interface	integer	1009	INTERFACE				
Address	string	LEQ1319211					
Label	string	HM-LC-Sw2-PW					
NetType	string	HM-LC-Sw2-PW					
Channels().Count	integer	3					
R_Channels	idarray	3569,3595 3595	CHANNEL CHANNEL				
CreatingCompleted	boolean	true					
LastTestCompletedTime	time	1970-01-01 01:00:00					
LastTestCompletedTimeS...	integer	0					

## Und rekursiv weiter über die Channels des Devices

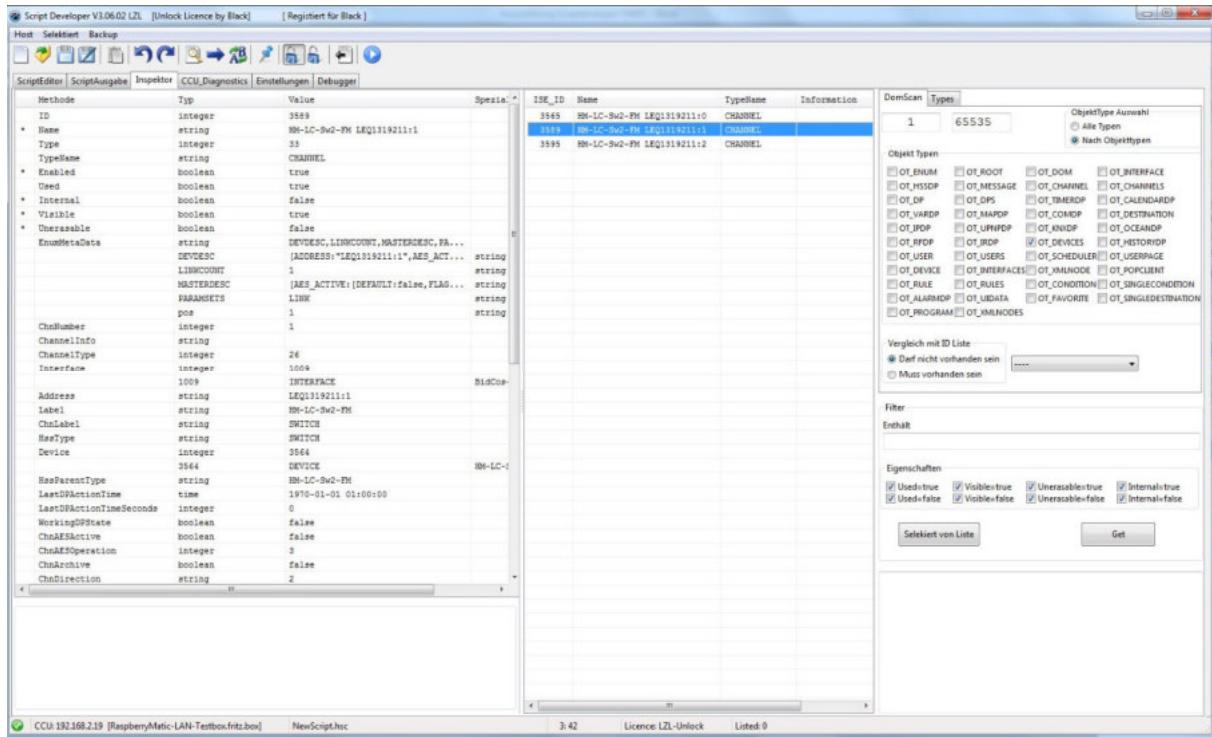


Mal in die Verwendung in dem Programm schauen über ChnEnumDPUsePrograms



Will ich jetzt aber wieder in den Channel zurück, so war dies in der alten Version nur mit Beginn der Selektion von ganz vorne angesagt.

Ab der 3.06.04 bin ich mit Undo eine Rekursionsebene zurück, hier in der Kanalauswahl, ab der ich direkt weitermachen kann



Ab Version 3.06.06 merkt sich der SDV zusätzlich zum Inhalt des Selektionsfeldes auch noch das zuletzt angezeigte Objekt in der Details view und stellt diese Ansicht auch wieder her (So das Objekt noch existent ist)

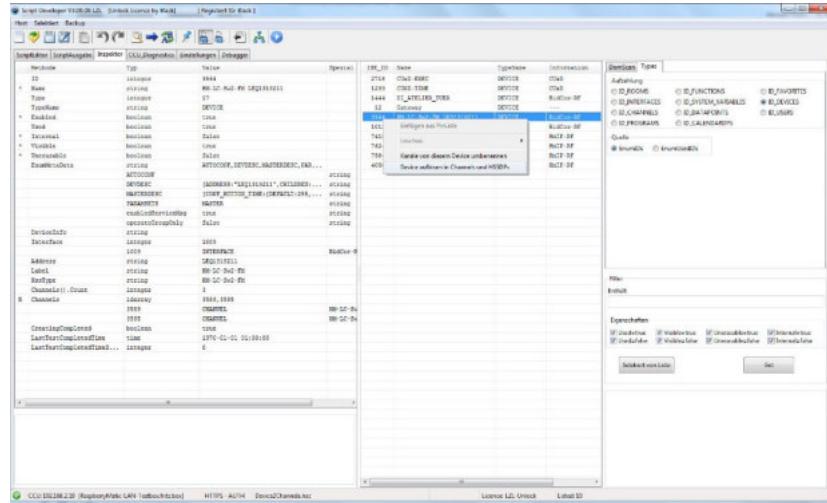
## 4.9 Auflösen komplexer Objekte

Der SDV wird die Möglichkeit haben, komplexe zusammen gesetzte Objekte aufzulösen und zur Bearbeitung zur Verfügung zu stellen.

Beispielsweise Devices und auch Programme.

### 4.9.1 Auflösen von Devices

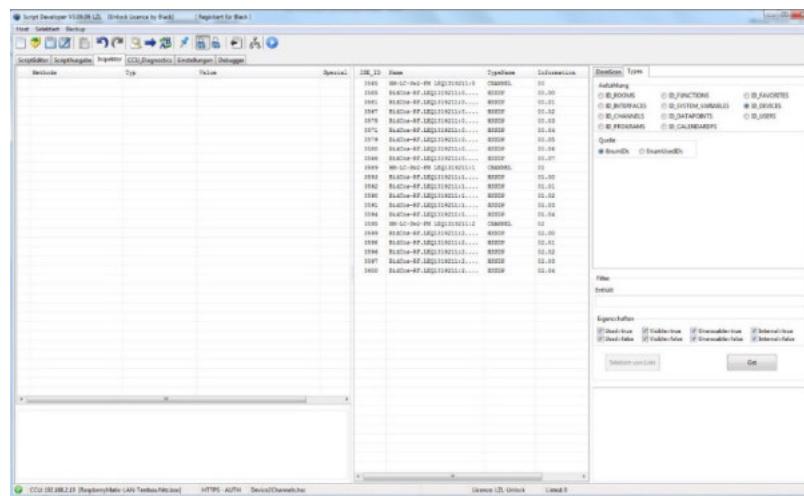
Ein Device besteht ja aus Channels und die Channels wiederum aus Datenpunkten



Wenn das Selektierte Objekt ein Device ist, so hat das PopUp Menü auf der rechten Maustaste nun auch das Feld: Device auflösen in Channels und HSSDPs.

Alternativ über den neuen Menüpunkt . Dieser ist nicht ausgegraut, wenn sich ein Objekt auflösen lässt.

Als Ergebnis erhält man :

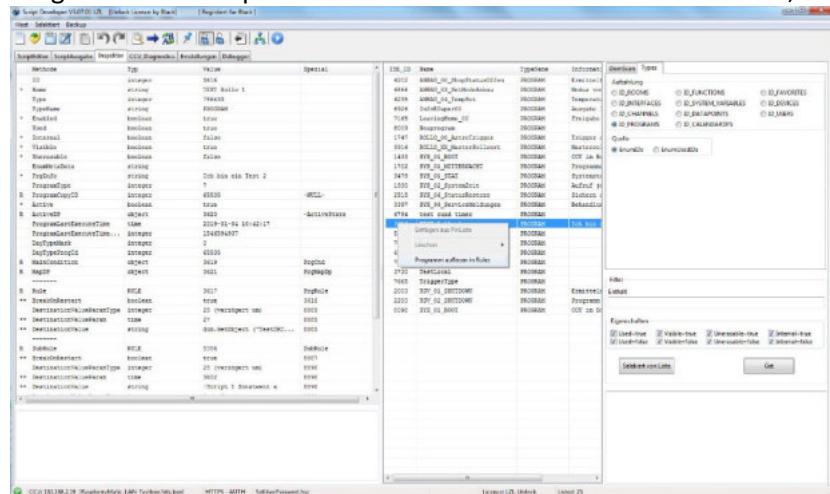


Diese Liste lässt sich dann weiter untersuchen mit den schon beschriebenen Arbeitsweisen (Auch Undo /Redo)

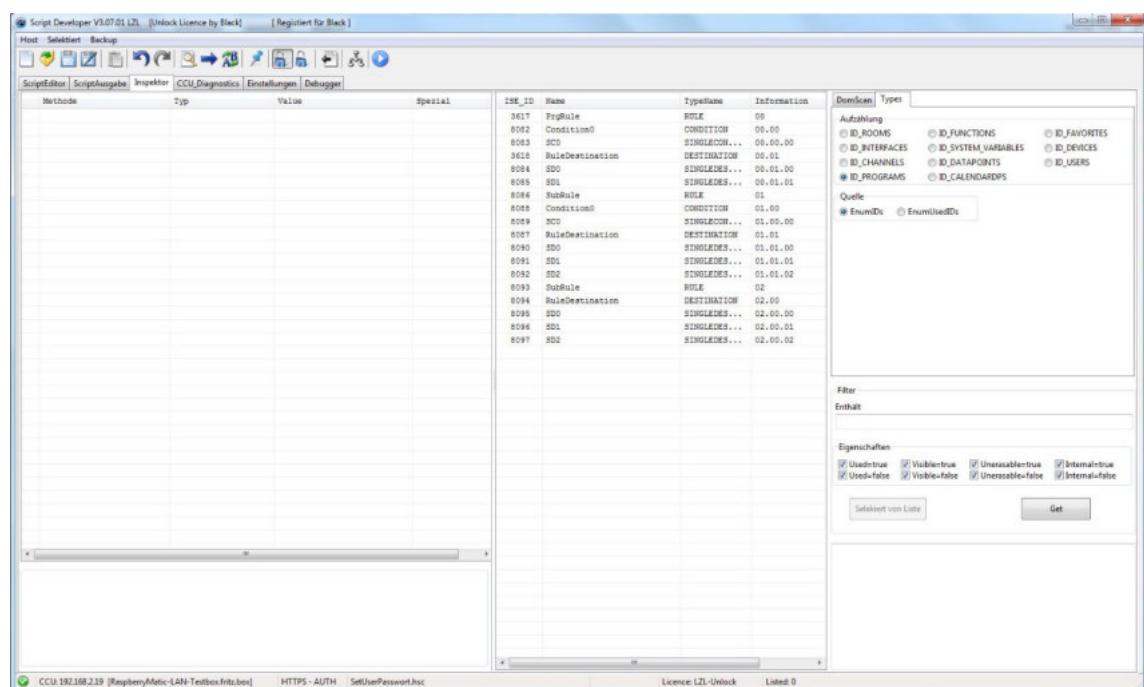
## 4.9.2 Auflösen von Programmen

Ein Programm besteht aus Rules (Regeln bzw den Subrules) und die jeweils aus den Conditions (und ihren untergeordneten SingleConditions sowie den Destinations und den untergeordneten Single Destinations

Aufgelöst wird adäquat zu den Devices: Auswahl über selektieren, dann rechte Maustaste und



Alternativ über den neuen Menüpunkt . Dieser ist nicht ausgegraunt, wenn sich ein Objekt auflösen lässt.



Die Sortierung über Information erlaubt eine chronologische Sortierung nach Auftreten der Objekte in einem Programm. Diese Liste lässt sich nun in der Detailansicht weiter untersuchen

Script Developer V3.07.01 L2L [Unlock Licence by Black] | Registert für Black | Host: Selektiert Backup

ScriptEditor ScriptAusgabe Inspector CCU\_Diagnostics Einstellungen Debugger

Methode	Typ	Value	Spezial	ISID	Name	TypeName	Information
ID	integer	8090		3417	FrgRule	RULE	00
* Name	string	SD0		8082	Condition0	CONDITION	00.00
Type	integer	6291457	SINGLEDESTINATION	8083	SCO	SINGLECOM...	00.00.00
* TypeName	string			3418	RuleDestination	DESTINATION	00.01
* Enabled	boolean	true		8084	SD0	SINGLEDES...	00.01.00
Used	boolean	true		8085	SD1	SINGLEDES...	00.01.01
* Internal	boolean	false		8086	SubRule	RULE	01
* Visible	boolean	true		8089	Condition0	CONDITION	01.00
* Unusable	boolean	false		8090	SCO	SINGLECOM...	01.00.00
EnumMetaData	string			8087	RuleDestination	DESTINATION	01.01
R DestinationChannel	integer	65535	-NULL-	8090	SD0	SINGLEDES...	01.01.00
R DestinationDF	integer	65535	-NULL-	8091	SD1	SINGLEDES...	01.01.01
DestinationParam	integer	20	intString	8092	SD2	SINGLEDES...	01.01.02
R DestinationParent	integer	8087	DESTINATION	8093	SubRule	RULE	02
R DestinationProgram	integer	3416	PROGRAM	8094	RuleDestination	DESTINATION	02.00
DestinationValue	integer	20	intString	8095	SD0	SINGLEDES...	02.00.00
* DestinationValueParamType	integer	20	intDelay	8096	SD1	SINGLEDES...	02.00.01
* DestinationValueParam	time	3602		8097	SD2	SINGLEDES...	02.00.02
DestinationValueParamType	string	'Script 1 Sonnena...					
DestinationValueParamType	integer	20	intDelay				
DestinationValueParam	time	3602					

DomScan Types

Aufzählung

- ID\_ROOMS       ID\_FUNCTIONS       ID\_FAVORITES
- ID\_INTERFACES       ID\_SYSTEM\_VARIABLES       ID\_DEVICES
- ID\_CHANNELS       ID\_DATAPOINTS       ID\_USERS
- ID\_PROGRAMS       ID\_CALENDAROPDS

Quelle

- EnumIDs       EnumUsedIDs

Filter

Eigenschaften

Used=true     Visible=true     Unusable=false     Internal=true  
 Used=false     Visible=false     Unusable=true     Internal=false

Selektiert von Liste      Get

CCU 192.168.2.19 [Raspberry/Matic-LAN-Testbox.fritz.box]      HTTPS - AUTH      SetUserPassword.hsc      Licence: L2L-Unlock      Listed: 0

## 4.10 Zeitmodule

Zeitmodule lassen sich nun auch in Detailansicht darstellen

Script Developer V3.07.01 L2L [Unlock Licence by Black] | Registert für Black | Host: Selektiert Backup

ScriptEditor ScriptAusgabe Inspector CCU\_Diagnostics Einstellungen Debugger

Methode	Typ	Value	Spezial	ISID	Name	TypeName	Information
* Internal	boolean	false		3489	Zeitmodul	CALENDAROP	
* Visible	boolean	true		5971	Zeitmodul	CALENDAROP	
* Unusable	boolean	false		4678	Zeitmodul	CALENDAROP	
EnumMetaData	string						
Date	boolean	true					
Time	time	1970-01-01 01:00:00					
TimeSeconds	integer	0					
TimerType	integer	4	ttPeriodic				
CalDuration	integer	0					
CalRepeatTime	time	1970-01-01 01:00:00					
CalRepeatTimeSeconds	integer	0					
Weekdays	integer	0					
Weekdays	integer	60					
Period	integer	60					
Begin	time	2018-01-19 00:00:00					
End	time	1970-01-01 01:00:00					
CalRepetitionCount	integer	0					
SumOffsetType	integer	0	setNone				
Next	time	2019-01-04 22:23:00					
Holidays	boolean	false					
VacationCheck	boolean	false					
VacationCheckType	boolean	false					
BeginSeconds	integer	1516316400					
EndSeconds	integer	0					
DPInfo	string						
DPUsageCount	integer	1					
R DPUsagePrograms	string	1930	PROGRAM	SYS_02_System			
Timestamp	time	2019-01-04 22:22:00					
LastValue	boolean	true					
LastTimestamp	time	2019-01-04 22:21:00					
LastTimestampSeconds	integer	1544688460					
LastTriggerID	integer	65535					
LastTriggerTime	time	1970-01-01 01:00:00					

DomScan Types

Aufzählung

- ID\_ROOMS       ID\_FUNCTIONS       ID\_FAVORITES
- ID\_INTERFACES       ID\_SYSTEM\_VARIABLES       ID\_DEVICES
- ID\_CHANNELS       ID\_DATAPOINTS       ID\_USERS
- ID\_PROGRAMS       ID\_CALENDAROPDS

Quelle

- EnumIDs       EnumUsedIDs

Filter

Eigenschaften

Used=true     Visible=true     Unusable=false     Internal=true  
 Used=false     Visible=false     Unusable=true     Internal=false

Selektiert von Liste      Get

CCU 192.168.2.19 [Raspberry/Matic-LAN-Testbox.fritz.box]      HTTPS - AUTH      SetUserPassword.hsc      Licence: L2L-Unlock      Listed: 3

## 4.11 Suchen in Scripten nach Variablen, Devices etc

Ab der Version 3.07.02 kann in den vorhandenen Scripten nach dem Vorkommen von Systemvariablen, Geräten, Kanälen, Räumen und Gewerken gesucht werden.

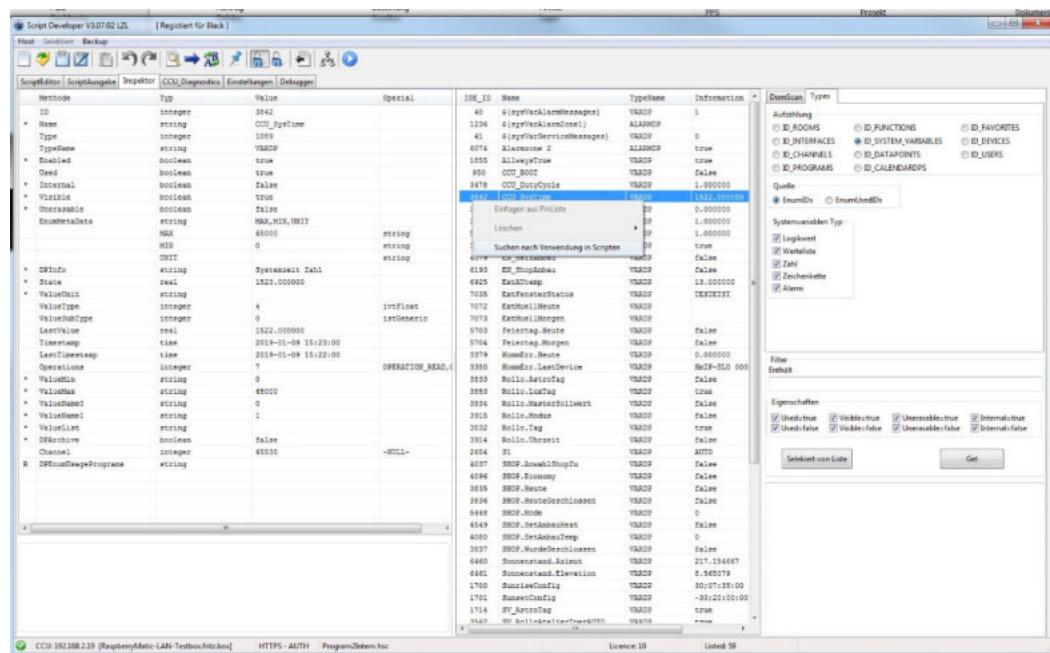
Dazu werden sämtliche SingleConditions, die als Property ein Script enthalten, gesucht und dann via Stringvergleich nach dem oder den Vorkommenden Namen abgesucht.

Vorgehensweise:

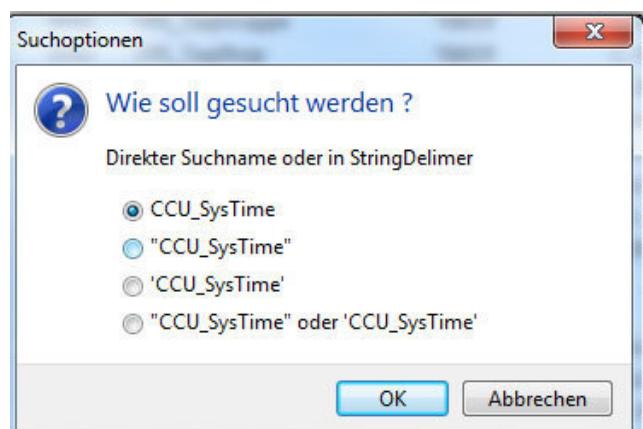
Markieren der oder auch mehrere Suchvariablen (Können Systemvariablen, Devices, Kanäle, Räume und Gewerke sein) Rechte Maustaste und suchen nach Verwendung in Scripten

Es kann hierbei gesucht werden:

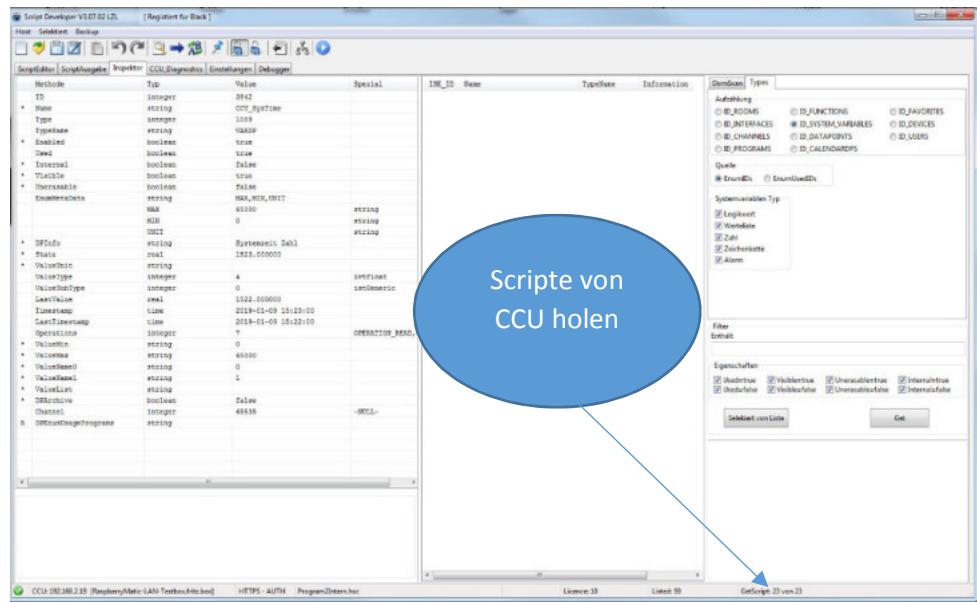
- Suchen "Name" nach Verwendung in Scripten : Suche nach dem Namen (SV, Alarme, Device, Chans)
- Suchen "Address" nach Verwendung in Scripten : Sucht nach der Seriennummer (Address) von Devices und Channels



Im darauf sich öffnenden Dialog festlegen ob der reine Text oder der Text in StringDelimitern gesucht werden soll

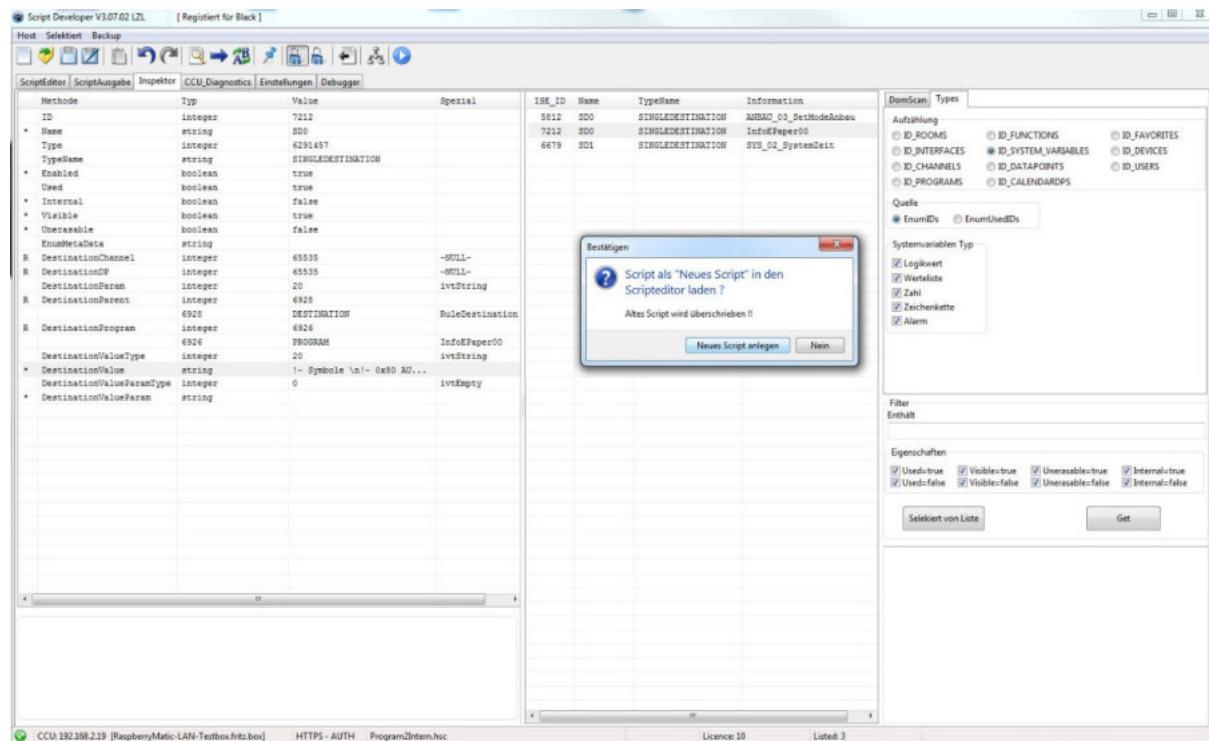


Daraufhin werden erstmal alle Rule in ihre Destinations aufgedröselt und eine Liste angelegt, wie viele Scripte es in diesen SingleDestinations dann gibt. Diese werden in den PC geladen. Das geht recht Fix, um der CCU zwischenzeitlich Zeit für Ihre Aufgaben zu lassen, wird direkt nach dem Empfang PC seitig die Stringanalyse gemacht und die Ergebnisliste aufbereitet.



Wenn fertig, gibt es eine Liste der Singledestinations, die die Scripte enthalten, in welchem der gesuchte Name vorhanden ist. Im Informationsfeld wird auch noch der zu der SingleDestination gehörendem Programm angezeigt:

Das Script lässt sich dann nach Rückfrage öffnen:



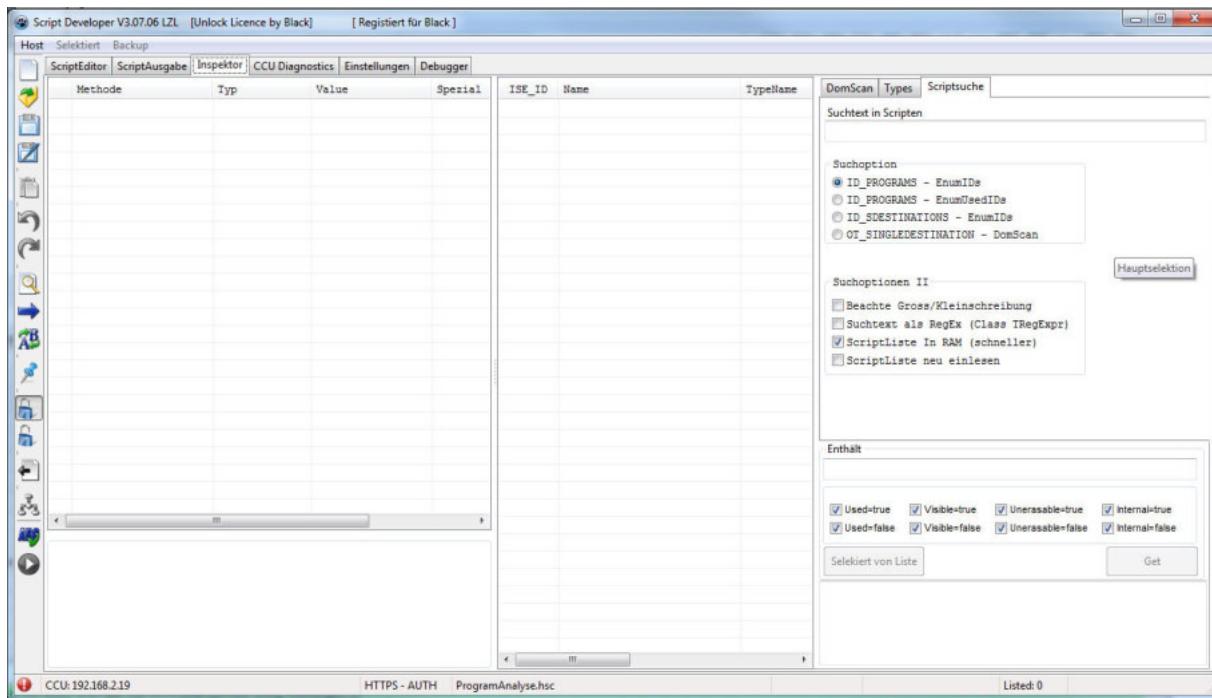
Der Editor öffnet sich und es wird direkt zum ersten Vorkommen des zu suchenden Namens gesprungen. bei mehrfachem Vorkommen sind alle Stellen gemäß den MarkUp Einstellungen hervorgehoben. Mit dem Pfeil für Weitersuchen lässt sich dann auch durch den Text nach dem Namen durchsuchen unabhängig von der Hervorhebung. Bei suchen Ersetzen ist als Suchbegriff der zu suchende Name schon vorbesetzt. Ersetzen automatisiert ist nicht vorgesehen, da sollte als letzte Instanz der Mensch das letzte Wort haben. Ein geändertes Skript lässt sich auch direkt wieder auf die CCU in das ursprüngliche Programm hochladen

The screenshot shows the Script Developer application interface with the following details:

- Title Bar:** Script Developer V3.07.02 L2L | Registriert für Block |
- Toolbar:** Includes icons for Home, Selection, Backup, and various file operations.
- Script Editor:** The main area displays a script in C# syntax. The script handles various system variables and icons, including:
  - Variables for LANG (e.g., LANG\_LANG, LANG\_KURZ).
  - Variables for Signals (e.g., Signal\_AUS, Signal\_RotesBlitzen, Signal\_GruenesBlitzen, Signal\_OrangenesBlitzen).
  - A comment block for the serial number of the HM-Display (HWQ1443971).
  - Code to get the status of the ExtAutemp window.
  - Logic to determine the status of the Müllstatus window based on current time and configuration.
- Script Output:** Shows the output of the script execution, with sections for Svals and Svals.
- Bottom Status Bar:** CCU 192.168.2.19 | RaspberryMatic-LAN-Test-Box | HTTPS-AUTH | NewScript.hsc | Licence 10 | Listed 3

## 4.12 Volltextsuche in Scripten

Ab Version 03.07.07 existiert eine Volltextsuche (ab Level 6)



Suchtext ist selbsterklärend.

Entweder wird nach dem Vorkommen des Textes in dem Script gesucht oder aber, wenn der Haken bei Suchtext als Regex gesetzt wurde, wird dieser Suchtext als regular Expression ausgewertet. (Da hier die Class TRegExpr von Lazarus verwendet wurde... das ganze in POSIX Syntax)

### ID\_Programs – EnumIDs

Es wird in der Aufzählung ID\_PROGRAMS , alle dort gelisteten IDs iteriert und in allen Skripten gesucht, die dort enthalten sind.

### ID\_Programs – EnumUsedIDs

Es wird in der Aufzählung ID\_PROGRAMS , alle dort gelisteten EnumUsedIDs iteriert und in allen Skripten gesucht, die dort enthalten sind.

### ID\_SDESTINATIONS – EnumIDs

Es wird in der Aufzählung ID\_SDESTINATIONS , alle dort gelisteten IDs iteriert und in allen Skripten gesucht, die dort enthalten sind.

### ID\_SINGLEDESTINATION – DomScan

Es wird die gesamte Regadom nach Objecten vom Typ Singledestination durchsucht und in allen Skripten gesucht, die dort enthalten sind. Diese Suchmethode dauert am längsten, findet aber auch Geisterobjekte.

### **Beachte Gross/Kleinschreibung**

Selbsterklärend, ist der Haken gesetzt, muss das Wort genauso in Gross Kleinschreibung vorhanden sein, ansonsten ist Gross Kleinschreibung egal

### **Suchtext als Regex**

Der Suchtext wird als Regulärer Ausdruck interpretiert. `.*Son.t.*` findet alle Script die das Wort Sonst, aber auch z.B. Sonat enthalten. Syntax nachzulesen unter

[https://en.wikipedia.org/wiki/Regular\\_expression#POSIX%20Basic%20Regular%20Expressions](https://en.wikipedia.org/wiki/Regular_expression#POSIX%20Basic%20Regular%20Expressions)

### **Scriptliste in Ram**

Diese Suchoption ist schneller, hierbei wird nur beim ersten Suchlauf die Skriptliste in den Rechner geladen. Wenn mehrere Suchläufe gemacht werden, wird ab dem zweiten Durchlauf mit den Scripten aus dem Ram gearbeitet. Dies ist wesentlich schneller als jedes Mal die Skripte neu von der CCU zu laden.

### **Scriptliste neu einlesen**

Wurden Skripte geändert oder Skriptänderungen zwischenzeitlich auf der CCU gemacht, so sollte dieser Haken gesetzt werden. Beim nächsten Suchlauf wird die einmalig Skriptliste neu aus der CCU geladen

## 4.13 SingleDestinations in ihrer Reihenfolge ändern

Eigentlich ein Wunsch von einem der Tester. Die Reihenfolge der Anweisungen in einem Programm lässt sich abändern.

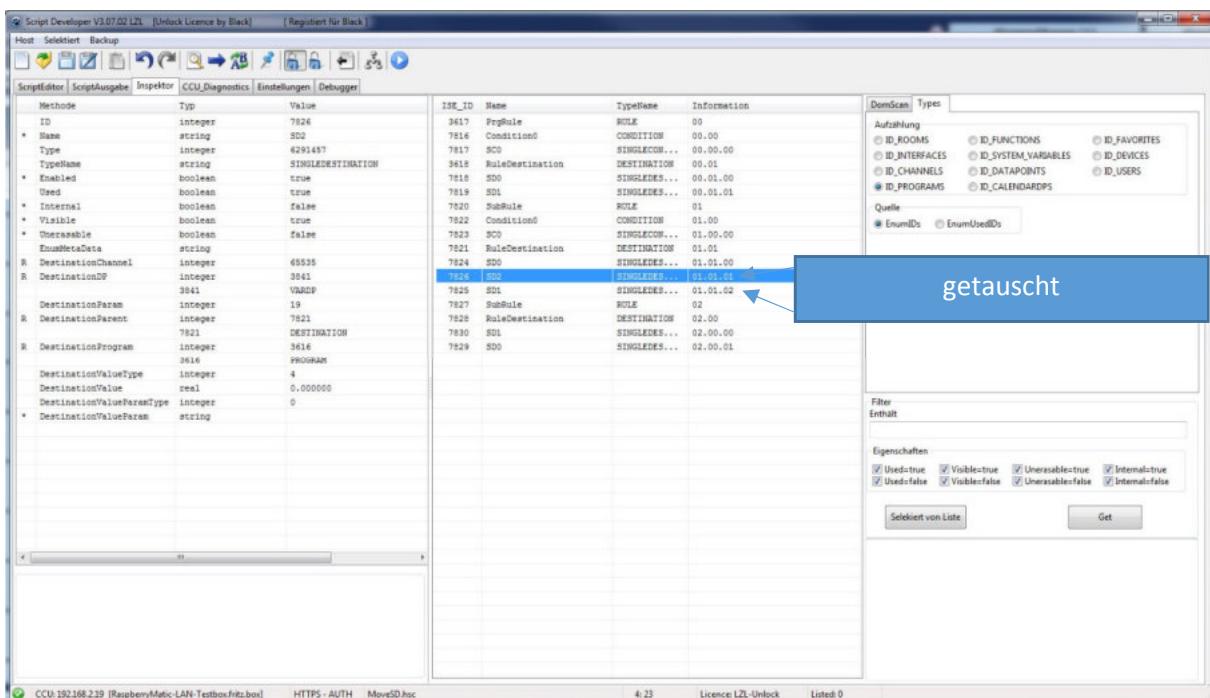
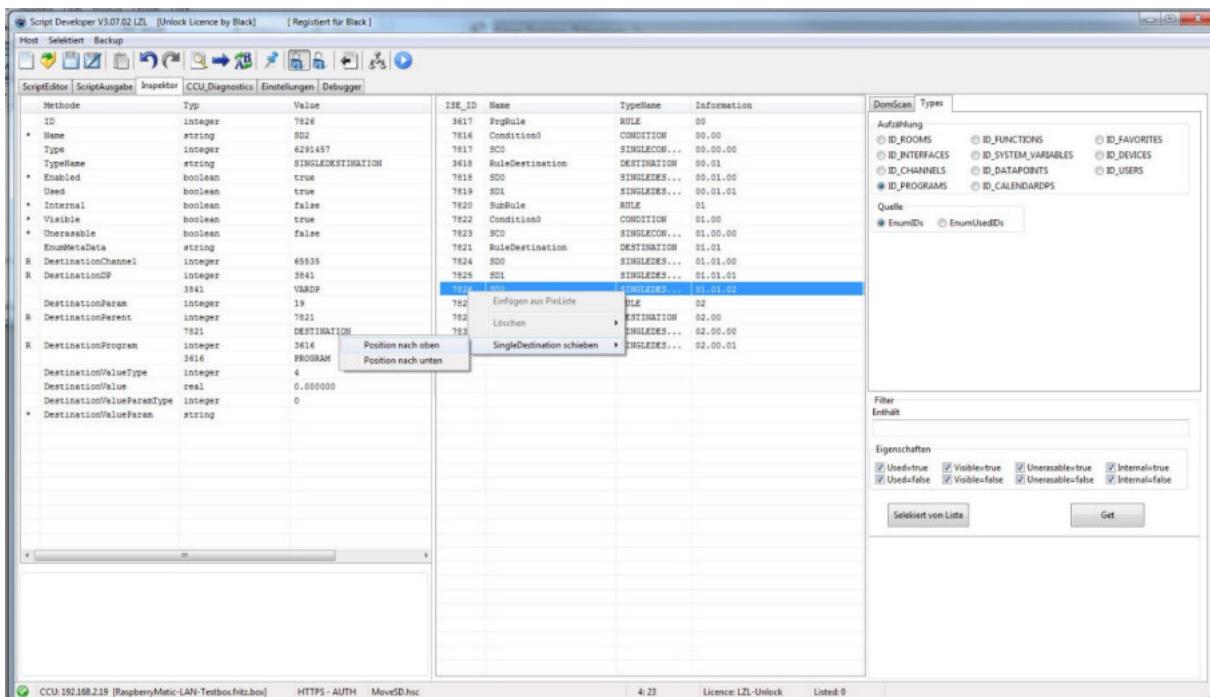
Vorher:

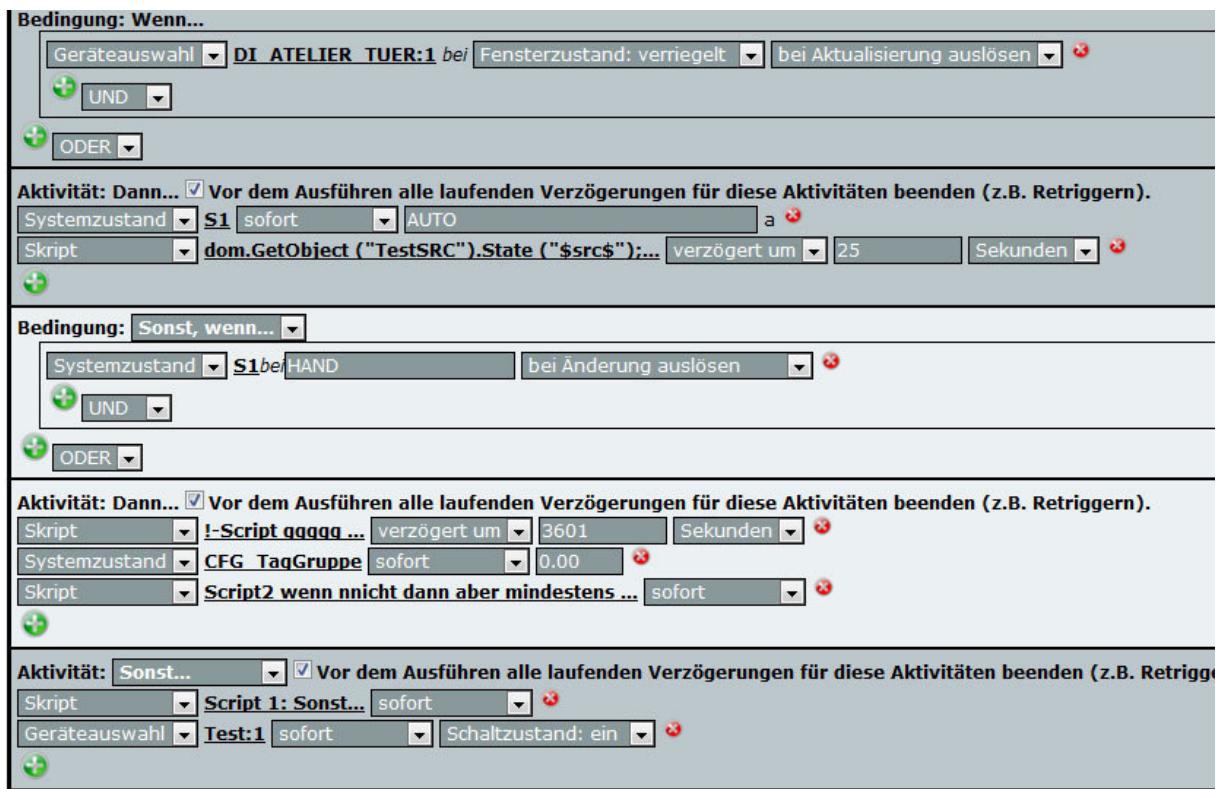
Name	Beschreibung	Bedingung (Wenn...)
TEST Rollo 1	Ich bin ein Test 2	Kanalzustand: DI_ATELIER_TUER:1 bei Fensterzustand: verriegelt bei Aktualisierung auslös
<b>Bedingung: Wenn...</b>		
Geräteauswahl ▼ DI_ATELIER_TUER:1 bei Fensterzustand: verriegelt □ bei Aktualisierung auslösen □		
+ UND □		
+ ODER □		
<b>Aktivität: Dann... ✓ Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).</b>		
Systemzustand □ S1 sofort □ AUTO □ a ↗		
Skript □ dom.GetObject("TestSRC").State("\$src\$"); verzögert um □ 25 □ Sekunden □		
+ ↗		
<b>Bedingung: Sonst, wenn...</b>		
Systemzustand □ S1 bei HAND □ bei Aenderung auslösen □		
+ UND □		
+ ODER □		
<b>Aktivität: Dann... ✓ Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).</b>		
Skript □ !-Script qqqq... verzögert um □ 3601 □ Sekunden □		
Skript □ Script2 wenn nnicht dann aber mindestens... □ sofort □		
Systemzustand □ CFG_TagGruppe □ sofort □ 0.00 □		
+ ↗		
<b>Aktivität: Sonst... ✓ Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).</b>		
Skript □ Script 1: Sonst... □ sofort □		
Geräteauswahl □ Test:1 □ sofort □ Schaltzustand: ein □		
+ ↗		

Das Aussehen im SDV

Sieht hier so aus

Index	ID	Name	Type	Information
3417	ProgRule		RULE	00
7814	Condition0		CONDITION	00-00
7817	SDO		SINGLEREAD...	00-00-00
3418	RuleDestination		DESTINATION	00-01
7818	SDO		SINGLEREAD...	00-01-00
7819	SDO		SINGLEREAD...	00-01-01
7820	SDO		SINGLEREAD...	00-01-10
7821	SubRule		RULE	01
7822	Condition0		CONDITION	01-00
7823	SDO		SINGLEREAD...	01-00-00
7824	RuleDestination		DESTINATION	01-01
7825	SDO		SINGLEREAD...	01-01-00
7826	SDO		SINGLEREAD...	01-01-01
7827	SubRule		RULE	02
7828	RuleDestination		DESTINATION	02-00
7829	SDO		SINGLEREAD...	02-00-00
7830	SDO		SINGLEREAD...	02-00-01

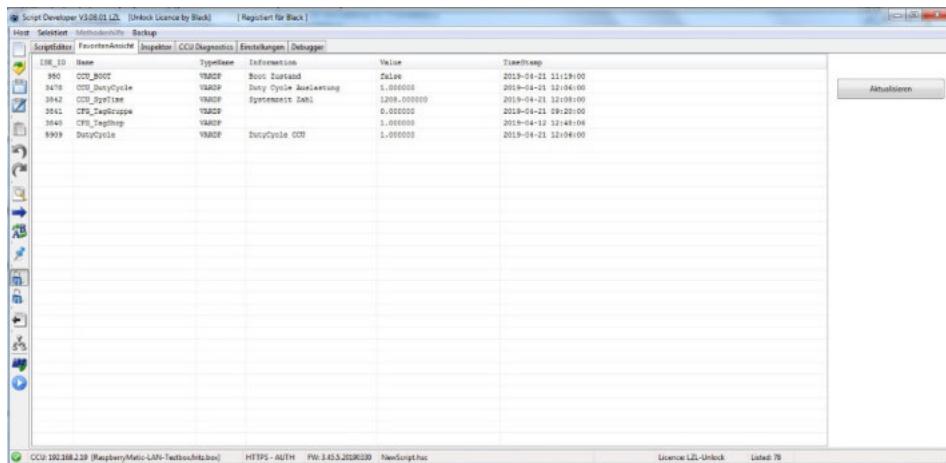




Und hier im WebUI Script

## 5 Favoritenansicht

Seit der Version V3.08.01 gibt es die Favoritenansicht. Diese soll dazu dienen, einen VariablenPool zusammenzustellen , dessen Werte und Zeitstempel sich übersichtlich darstellen und beobachten lassen.



Aktualisieren führt zum einen Nachladen der aktuellen Werte aus der CCU. In die Favoriten können Systemvariablen, Alarne, Datenpunkte und Programme übernommen werden.

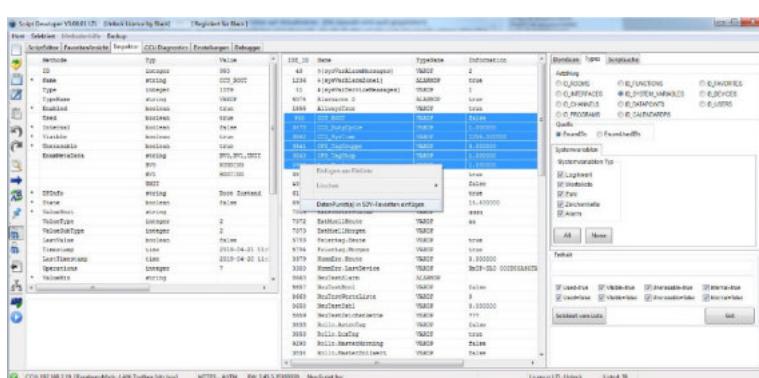
Will man einen Datenpunkt mal eben ändern, einen Doppelklick auf die Zeile der Favoriten, daraufhin wird unter Berücksichtigung des Undo / redo Stacks die Favoriten in die Listenansicht geladen und das angeklickte Objekt zur Bearbeitung in der Detailansicht geöffnet.

Der Inhalt der Favoritenliste wird in der SDV.INI automatisch gespeichert, und steht bei einem Neustart auch wieder zu Verfügung.

Mit Öffnen des Reiters FavoritesAnsicht werden die aktuellen Daten frisch von der CCU geholt

### 5.1 Hinzufügen von Objekten in die Favoriten

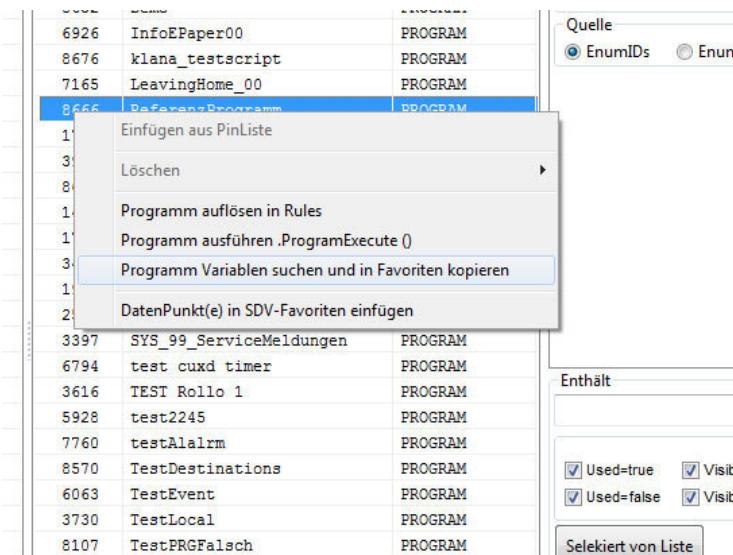
Objekte können aus der Listenansicht des Inspektors in die Favoriten geladen werden. Dazu Objekte Selektieren rechte Maustaste und



Neue Datenpunkte werden immer angehängt. Doppelte IDs werden unterdrückt, zweimal die gleiche Sysvar in den Favoriten geht also nicht.

## 5.2 Verwendete Objekte eines Programmes in die Favoriten laden

Zur Fehlersuche oder zum Testen möchte man öfters alle Werte, die ein Programm beeinflusst oder die von einem Programm beeinflusst werden, im Überblick haben. Dies geht mit dem SDV recht einfach. Dazu in der Listenansicht im Inspektor ein Programm selektieren, rechte Maustaste und „Programm variablen suchen und in Favoriten kopieren“



Darauf löst der SDV das Programmobjekt intern rekursiv auf und schreibt alle gefundenen Systemvariablen und Datenpunkte von Geräten in die Favoritenliste.

## 5.3 Favoritenliste löschen

In die Favoriten reinklicken, rechte Maustaste – Favoritenliste löschen

Es wird natürlich nur die interne Favoritenliste des SDV gelöscht, auf der CCU wird da nicht gelöscht oder verändert.

## 5.4 Einschränkungen

Die Favoritenliste arbeitet IseID orientiert. Wird von der CCU ein Object gelöscht, beispielsweise eine Systemvariable, so wird mit dem nächsten Aktualisieren diese auch von der Favoritenansicht gelöscht, da der ISEID Bezug nun ins Leere greift.

Beim Arbeiten mit 2 CCU's bedeutet das:

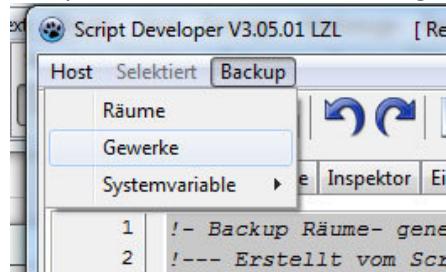
Ein Wechsel der CCU als Host im SDV führt immer auch zu einer komplett anderen ISEID Liste. So wird es dann vorkommen, dass die favoritenliste auf einmal leer sein wird, wenn der Host gewechselt wird.

## 6 Kleine Helfer im Alltag

### 6.1 Backups

Von relevanten Objekten können Backups gemacht werden. Diese ersetzen KEIN richtiges SystemBackup an der CCU !!!  
Vielmehr dienen diese im Falle eines Umzuges von einem alten System auf ein Neusystem als Hilfestellung, wenn man das alte Systembackup nicht benutzen will (Loswerden von in den Jahren angesammelten Leichen), oder aber ein inkonsistentes System.

Den passenden Lizenzlevel vorausgesetzt, findet sich die Backups hier:



Devices müssen VORHER manuell umgezogen worden sein über ablernen und neu anlernen.  
Und die Geräte müssen, damit die Backups von Räumen und Gewerken sinnig arbeiten können, wieder ihre „alten“ Namen haben.  
Siehe dazu auch die passende EQ3 Dokumentation. Der SDV legt keine neuen Devices oder Direktverbindungen an.

#### 6.1.1 Räume

Auf dem Bestandssystem wird ein Restore Programm erzeugt und lässt sich anschließend auf dem PC abspeichern. Der vorgeschlagene Dateiname ist dabei backup\_Rooms\_ + datum und Uhrzeit der Generierung.

Auf dem „Neusystem“ lässt sich dieses Programm über den SDV via Scripteditor dann starten.  
Dabei passiert folgendes:

Es wird geprüft, ob dort schon ein Raum mit dem Namen „XX“ existiert. Wenn ja, gut, wenn nein, wird dieser Raum neu anlegt, mit Namen und Beschreibung versehen und in ID\_ROOMS eingehängt. Waren dem alten Raum Kanäle zugeordnet, so versucht der SDV nun diese Kanäle des Altsystems über ihren Kanalnamen zu identifizieren. Ist dieses erfolgreich, so wird dieser Kanal dem Raum hinzugefügt.

#### 6.1.2 Gewerke

Auf dem Bestandssystem wird ein Restore Programm erzeugt und lässt sich anschließend auf dem PC abspeichern. Der vorgeschlagene Dateiname ist dabei backup\_Functions\_ + datum und Uhrzeit der Generierung.

Auf dem „Neusystem“ lässt sich dieses Programm über den SDV via Scripteditor dann starten.  
Dabei passiert folgendes:

Es wird geprüft, ob dort schon ein Gewerk mit dem Namen „XX“ existiert. Wenn ja, gut, wenn nein, wird dieses Gewerk neu anlegt, mit Namen und Beschreibung versehen und in ID\_FUNCTIONS eingehängt.

Waren dem alten Gewerk Kanäle zugeordnet, so versucht der SDV nun diese Kanäle des Altsystems über ihren Kanalnamen zu identifizieren. Ist dieses erfolgreich, so wird dieser Kanal dem Gewerk hinzugefügt.

### 6.1.3 Systemvariablen

Der komplizierteste Part.

Auf dem Bestandssystem wird ein Restore Programm erzeugt und lässt sich anschließend auf dem PC abspeichern. Der vorgeschlagene Dateiname ist dabei backup\_Sysvars\_ + Datum und Uhrzeit der Generierung.

Auf dem „Neusystem“ lässt sich dieses Programm über den SDV via Scripteditor dann laden.

Hierbei können noch folgende Einstellungen in dem Programm Kopf vorgenommen werden:

```
----- Scriptausgabe -----
!-      Backup SystemVariablen vom 06.12.2018 13:21:02
!-      Erstellt mit Script Developer V3.04 by Black 2018
!----- Diese Zeilen Anpassen -----
boolean bcreate= true; !- Anlegen, wenn noch nicht existierte
boolean bupdate= true; !- Wert Updaten, wenn vorhanden und gleicher Typ
boolean barchive= false; !- false: immer restore mit DPArchive (false), true: restore mit
altem Wert
```

bcreate:

true: wenn die Systemvariable noch nicht existiert wird diese angelegt und in ID\_SYSTEM\_VARIABLES eingehängt.

False: wenn die Systemvariable noch nicht existierte, wird auch nix gemacht.

bupdate:

true: wenn die Systemvariable schon existierte und diese den gleichen Typ hat, wird der State wert aus dem Backup in die variable geschrieben. Wenn nicht der gleiche Typ- passiert nix

false: wenn die Systemvariable schon existiert- wird nix gemacht

barchive: (nur bei Neuanlage)

true: beim Restore wird die Archiv Option der Systemvariable aus dem Backup genommen.

False: es wird immer ohne Archiv Option angelegt beim Restore.

Der SDV unterscheidet dabei von sich aus zwischen Alarm und Systemvariable. Bei Alarm wird nicht der Zustand (AllsArmed) verändert. Heisst: bei Neu Anlage sind die Alarne immer scharf, auch wenn dieser Alarm vorher im Alt System über AIArm (false) unscharf geschaltet wurde !

Zugeordnete Channels werden ebenfalls versucht zu rekonstruieren, so sich der Kanal über den alten Kanalnamen identifizieren lässt (s.a. Räume und Gewerke)

### 6.1.4 Devices und Kanäle

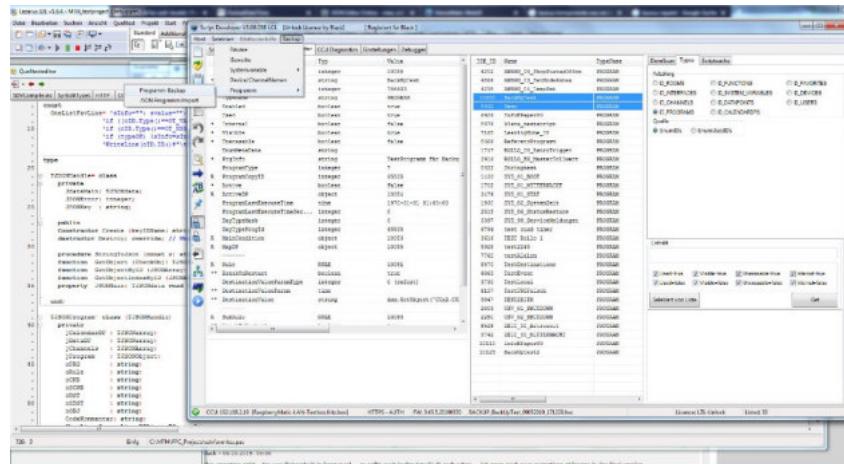
Bei diesem Backup werden die Namen der Kanäle und Geräte gesichert. Die Identifikation erfolgt später über das Interface und die Seriennummer, die der Kanäle durch Durchiterieren und Vergleich mit ChnNumber Methode.

Hilfreich beim Umzug von einem System auf ein anderes System. Nachdem die Geräte abgelernt und am neuen System MANUELL !!!! angelernt wurden, kann das Restore Programm die alten Namen anhand der Seriennummern wiederherstellen. Anschließend können die Raum / Gewerk und Systemvariablen Restore gemacht werden.

### 6.1.5 Backup Programme

Über diesen Punkt kann von einem oder mehreren WEB-UI Programmen Backups gezogen und als Datei (JSON bzw Ausführbares HM-Script) abgespeichert werden. Mit diesen Programmen lässt dich ein WEB-Ui Programm vollständig wiederherstellen. Dabei arbeitet der restore nicht ISEID bezogen, mit dem Restore Programm lässt sich ein WebUI Programm auch auf einem komplett anderen System wiederherstellen, so natürlich die verwendeten Datenpunkte in symbolischer Adressierung und vom richtigen Typ vorhanden sind:

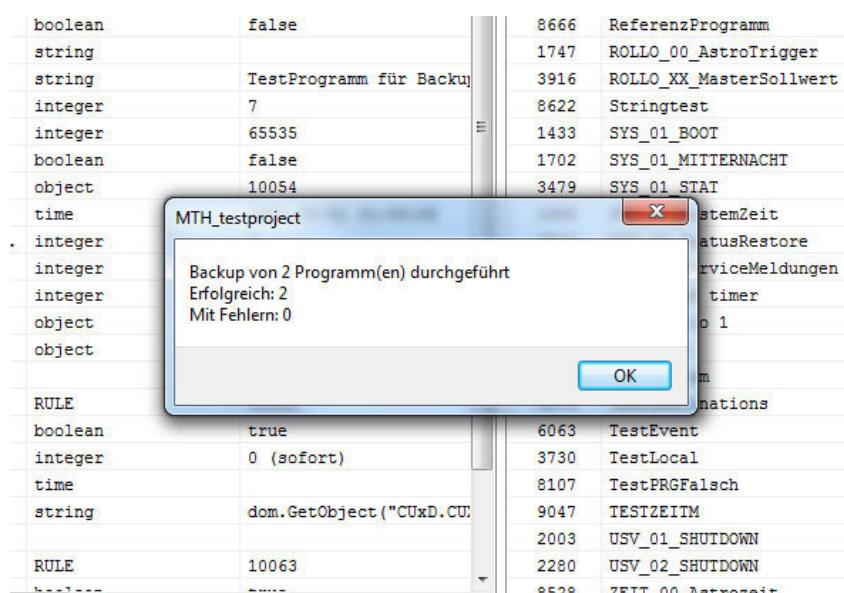
Programme in der Listenauswahl selektieren, eins oder mehrere...



Danach in dem sich öffnenden Dialog das Verzeichnis Auswählen, wo das Backup hin gespeichert werden soll:

Der Programmname heisst 'BACKUP\_ + ProgrammName + \_ + TTMMYYYY\_HHMMSS.hsc bzw .json danach werden die Backups der ausgewählten Programme angelegt.  
werden Objekte selektiert, die keine Programme sind, so werden diese natürlich ignoriert.

So sollte es aussehen, wenn alles durchlief:



ab V8.03.04 wird auch noch die Laufzeit für das Backup in Sekunden angezeigt. In der unteren Infozeile erfolgt Anzeige ganz rechts, welches Programm von Wievielen gerade im Backprozess ist.

danach liegen im Verzeichnis 2 Files.

das hsc kann z.b. in den SDV geladen werden und dort ausgeführt werden.  
Der Kopf sieht immer so aus:

die ersten Zeilen können auch Händisch angepasst werden, wenn man den Weiß, was man tut.

Programmname ist selbsterklärend, die Einträge unter verwendete Kanäle und verwendete Systemvariablen können verändert werden. (beispielsweise Rollo Programm, wenn man einen neuen Aktor benutzt... oder Hm auf HMIP umbaut. es muss nur der alte auf den neuen Kanal geändert werden, das ID zusammenbasteln Gedöns macht dann das Programm. wo was substituiert wurde steht auch als Test im Programm. Hier sieht man es besser, Der SDV trägt in die Kommentarzeilen unter der Tabelle ein, welcher Datenpunkt wie über Kanal - HSSID zusammengesetzt wurde.

Zum Restore.

Es wird überprüft, ob das Programm schon existiert... natürlich Abbruch.  
auch ob alle verwendeten Datenpunkte (Sysvars Kanäle oder DPs) vorhanden sind und auch den  
richtigen Typ haben. Originale Sysvar BOOL und auf dem neuen System sysvar String erkennt der SDV  
und bricht vor der Generierung ab.

Bei einem Erfolgreichen Durchlauf sieht die Scriptausgabe so aus:

```
Backup erstellt vom SDV V3.08.03B ICL am 09.05.2019 19:43:41
CCU ProgrammName: "BackUpTest"
CCU ProgrammInfo: "TestProgramm für Backup Restore"
-----
Test auf Existenz referenzierter Channels
Test auf Existenz und Typkonsistenz referenzierter Systemvariablen
    OK.... SYSVAR TP1
    OK.... SYSVAR TestVar
Test auf Existenz und Typkonsistenz referenzierter Geräte-Datenpunkte
Restore Programm von Program "BackUpTest" erfolgreich durchgelaufen
----- Script Variablen -----
```

Im Hinblick z.B auf Gerätetausch... Gerät selektieren rechte Maustaste und dann WEBUI Programme von diesem Gerät.

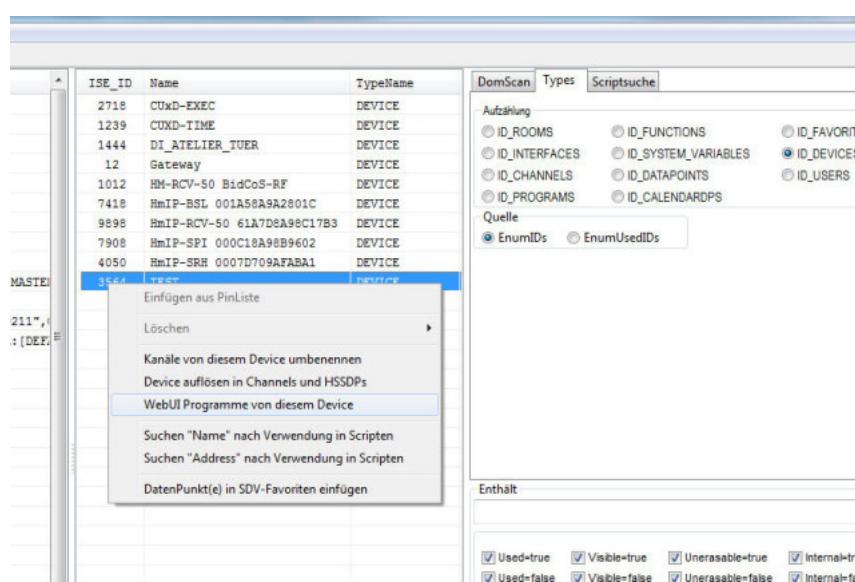
Funktion ist unspektakulär kann die WebUI auch (habt nur programmiert, weil ein Device keine idarray hat, wo die verwendeten Programme drinstehen)

Diese gefundenen Programme kann man nun selektieren, und backup machen.

Im Editor bearbeitet lassen sich nun z.b. die Kanalbezeichner ändern (auch HM auf HMIP z.B.).

dann die gefundenen Programme löschen und die neuen Restoren.. Gerätetausch von HM-nach HMIP (Das neue Gerät mit den Kanälen MUSS natürlich vorhanden sein und die Datenpunkte auch den gleichen Namen (z.B. level) und Typ haben..)

natürlich müssen die neuen Kanal Bezeichner richtig eingegeben worden sein, sonst gibt das Shit in - Shit out Prinzip



## 6.2 Umbenennen von Kanälen von Geräten

Wer hatte nicht schon alles die Freude, z.B. an einem neu angelernten IP Gerät mit 14 Kanälen die Namen neu zu vergeben. Dies geht nun schneller.

Das Device wird selektiert und der Name der Device geändert.

Anschliessend rechte maustaste auf das Device in der Listendarstellung und Punkt auswählen:

ISE_ID	Name	TypeName	Information
2718	CUxD-EXEC	DEVICE	CUxD
1239	CUXD-TIME	DEVICE	CUxD
1444	DI_ATELIER_TUER	DEVICE	BidCos-RF
12	Gateway	DEVICE	---
3564	HM-LC-Sw2-FM LEQ1319211	DEVICE	BidCos-RF
1012	HM-RCV-50 BidCos-RF	DEVICE	BidCos-RF
74	Einfügen aus PinListe		HmIP-RF
76	Löschen		HmIP-RF
75			HmIP-RF
40	Kanäle von diesem Device umbenennen		HmIP-RF

Rückfrage mit Ja bestätigen und die Kanäle werden so benannt:

Device: DeviceName

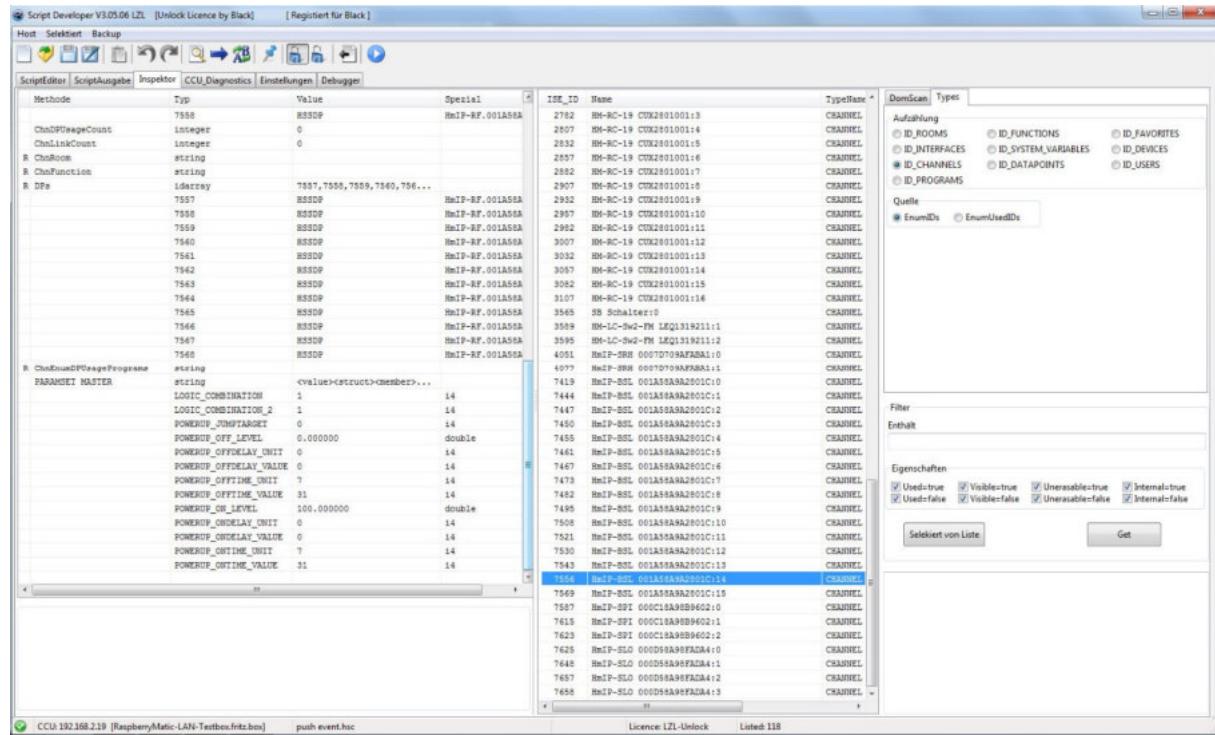
Kanal0 : DeviceName:0

Kanal1 : Devicename:1

Etc...

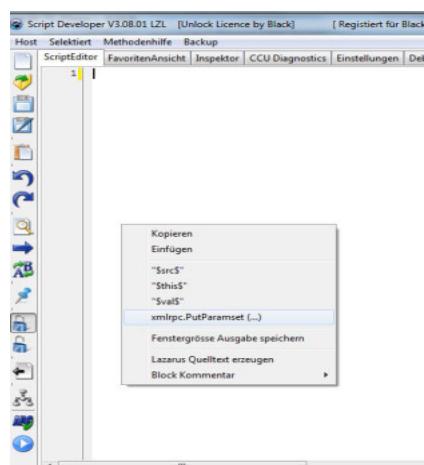
## 6.3 Paramset Master

Bei Device, Kanälen, die einen Paramset Master Haben, wird dieser mit angezeigt (wenn in der Sicht konfiguriert und mind. Level 6)

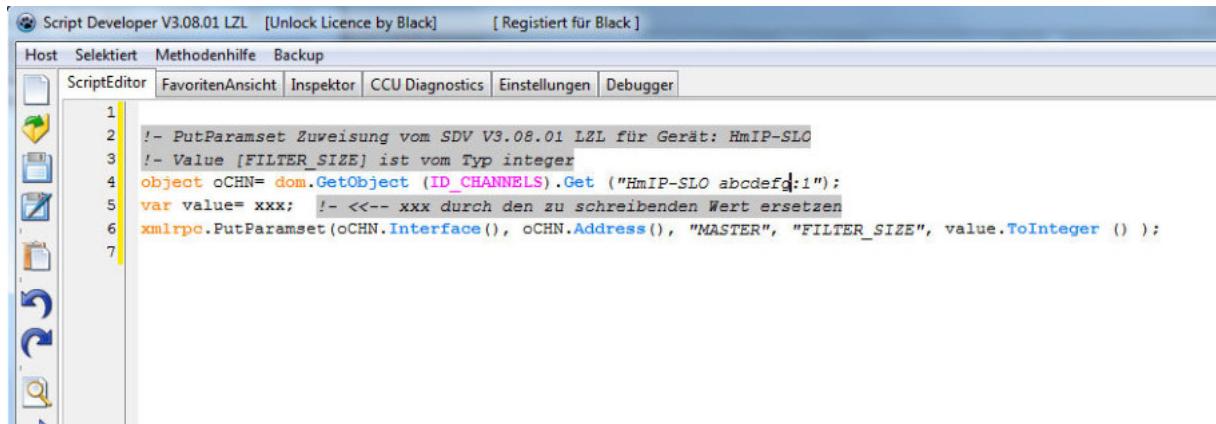


Die Werte lassen sich dann über PutParamset verändern in einem Script

Ab Version V3.08.01. Einzelne Parameter des Mastersatzes lassen sich nun auch markieren in der Detailansicht. Im Editor lässt sich nun Automatisiert der Code für die Manipulation dieses Masterparameters erzeugen.



Dies erzeugt nun automatisiert den richtigen Code, in Abhängigkeit von: welches Gerät ist es (HMIP, HM, CUXD o.ä) und dem erwarteten Datentyp.



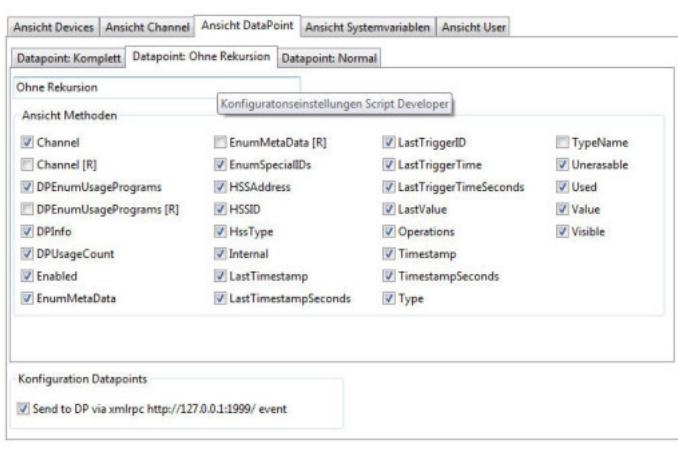
```

1  !- PutParamset Zuweisung vom SDV V3.08.01 LZL für Gerät: HmIP-SLO
2  !- Value [FILTER_SIZE] ist vom Typ integer
3  object oCHN= dom.GetObject (ID_CHANNELS).Get ("HmIP-SLO abcdefg:1");
4  var value= xxx; !- <<- xxx durch den zu schreibenden Wert ersetzen
5  xmirpc.PutParamset(oCHN.Interface(), oCHN.Address(), "MASTER", "FILTER_SIZE", value.ToInteger () );
6
7

```

## 6.4 Rega Push auf Datenpunkte via Rega event

**Rega Push:** Damit lassen sich Datenpunkte innerhalb der Rega verändern, die sich normalerweise nicht verändern lassen. Z.B. Batteriefehler eines Netzaktors ^^. Setzt voraus, dass dieser Haken im der Konfiguration Datenpunkte gesetzt wurde:



Konfigurationseinstellungen Script Developer			
<input checked="" type="checkbox"/> Channel	<input type="checkbox"/> EnumMetaData [R]	<input checked="" type="checkbox"/> LastTriggerID	<input type="checkbox"/> TypeName
<input type="checkbox"/> Channel [R]	<input checked="" type="checkbox"/> EnumSpecialIDs	<input checked="" type="checkbox"/> LastTriggerTime	<input checked="" type="checkbox"/> Unerasable
<input checked="" type="checkbox"/> DPEnumUsagePrograms	<input checked="" type="checkbox"/> HSSAddress	<input checked="" type="checkbox"/> LastTriggerTimeSeconds	<input checked="" type="checkbox"/> Used
<input type="checkbox"/> DPEnumUsagePrograms [R]	<input checked="" type="checkbox"/> HSSID	<input checked="" type="checkbox"/> LastValue	<input checked="" type="checkbox"/> Value
<input checked="" type="checkbox"/> DPInfo	<input checked="" type="checkbox"/> HstType	<input checked="" type="checkbox"/> Operations	<input checked="" type="checkbox"/> Visible
<input checked="" type="checkbox"/> DPUsageCount	<input checked="" type="checkbox"/> Internal	<input checked="" type="checkbox"/> Timestamp	
<input checked="" type="checkbox"/> Enabled	<input checked="" type="checkbox"/> LastTimestamp	<input checked="" type="checkbox"/> TimestampSeconds	
<input checked="" type="checkbox"/> EnumMetaData	<input checked="" type="checkbox"/> LastTimestampSeconds	<input checked="" type="checkbox"/> Type	

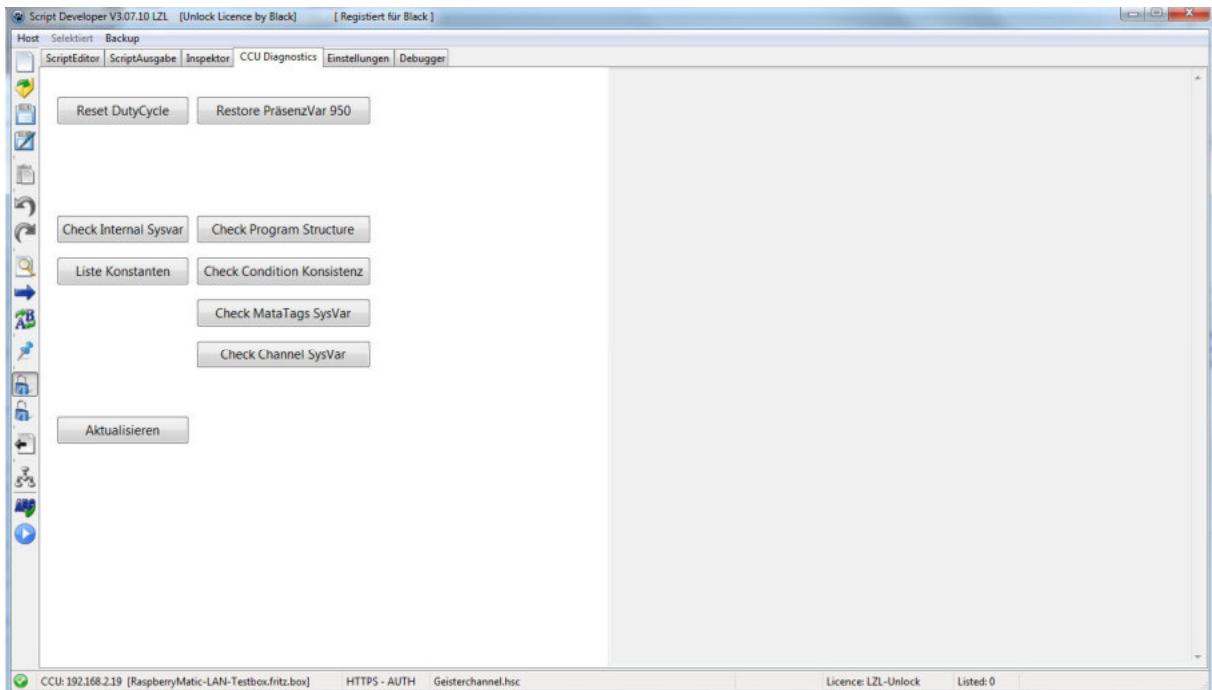
Konfiguration Datapoints  
 Send to DP via xmlrpc http://127.0.0.1:1999/ event

Des weiteren braucht es in der Sicht Freigabe auf TypeName (SDV prüft auf HSSDP) und anklicken von Value. (Level 7 braucht es dafür auch)

Hat der DP die Eigenschaft Write, wird ein ganz normales State (xxx) ausgeführt. Hat er diese nicht, macht der SDV bei Wertänderung von sich aus ein Rega Event auf den DP.

(Wenn die Freigabebedingungen passen)

## 7 Diagnosebild



Hier auch Schnellstatus Programme mit gesetzter Copy ID. Ist nach Neustart zentrale immer noch der Wert <>0... Geisterprogramm

## Alphabetische Auflistung aller bisher bekannten Systemkonstanten mit ihren Werten

The screenshot shows the 'ScriptDeveloper' application window with the title 'Script Developer V3.07.10 LZL [Unlock Licence by Black]'. The menu bar includes 'Host', 'Selektiert', 'Backup', 'ScriptEditor', 'ScriptAusgabe', 'Inspektor', 'CCU Diagnostics', 'Einstellungen', and 'Debugger'. Below the menu is a toolbar with icons for file operations like Open, Save, Print, and a search function. A central panel contains several buttons: 'Reset DutyCycle', 'Restore PräsenzVar 950', 'Check Internal Sysvar', 'Check Program Structure', 'Liste Konstanten' (which is highlighted in blue), 'Check Condition Konsistenz', 'Check MataTags SysVar', and 'Check Channel SysVar'. At the bottom left is an 'Aktualisieren' button. To the right is a large list of system constants with their values:

ID_ALARM_MAPS	:	700
ID_ALARMS	:	701
ID_CALENDARDPs	:	34
ID_CHANNEL_COMMUNICATION	:	24
ID_CHANNEL_STATE_VARIABLES	:	23
ID_CHANNELS	:	4
ID_CIRCUITS	:	501
ID_CONDITIONS	:	35
ID_CONTACTS	:	601
ID_DATAPOINTS	:	5
ID_DESTINATIONS	:	37
ID_DEVICES	:	3
ID_DOM	:	1
ID_ENOCEAN	:	850
ID_ENOCEAN_BEGIN	:	851
ID_ENOCEAN_LASTREC	:	892
ID_ERROR	:	65535
ID_EVENTING	:	11
ID_FAVORITES	:	201
ID_FUNCTIONS	:	151
ID_GATEWAYCONFIG	:	20
ID_GW_CHANNEL	:	13
ID_GW_DATAPOINT	:	14
ID_GW_DEVICE	:	12
ID_GW_SYSALARM	:	40
ID_GW_SYSERVICE	:	41
ID_HISTORYDPs	:	16
ID_INTERFACES	:	9
ID_INTERNALCHANNEL	:	100
ID_IP_DP_GW	:	39
ID_LINKS	:	301
ID_MESSAGES	:	31
ID_PRESENCE_SIMULATION	:	19
ID_PRESENT	:	950
ID_PROGRAMS	:	15
ID_RF	:	900
ID_RF_BEGIN	:	901
ID_RF_LASTREC	:	942
ID_ROOMS	:	101
ID_ROOT	:	2
ID_RULES	:	33
ID_RUNTIMECONFIG	:	21

At the bottom of the window, status bars show 'CCU: 192.168.2.19 [RaspberryMatic-LAN-Testbox.fritz.box]', 'HTTPS - AUTH', 'Geisterchannel.hsc', 'Licence: LZL-Unlock', and 'Listed: 0'.

## 7.1 SSH Funktionalität

Wenn auf der CCU / Raspberrymatik SSH freigegeben ist, ist es möglich, ab Level 5 eine rudimentäre SSH Funktionalität zu nutzen.

Was ist dazu nötig ?

In der SDV Ini müssen die nötigen Schlüssel eingegeben sein.

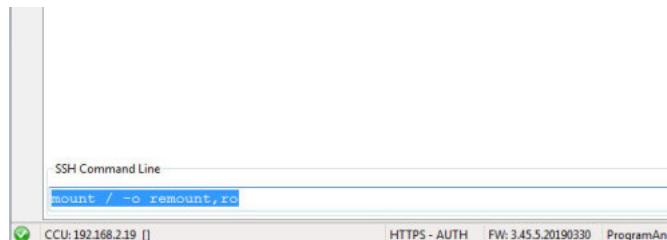
```
[CCU1]
IP=192.168.2.xx
USERNAME=xxxx
PASSWORD=yyyy
USEHTTPS=true
SSHUSERNAME=root      Bestandsnutzer: diese schlüssel hinzufügen
SSHPW=sshpasswort
```

```
[CCU2]
IP=192.168.2.xx
USERNAME=
PASSWORD=
USEHTTPS=false
SSHUSERNAME=root
SSHPW=
```

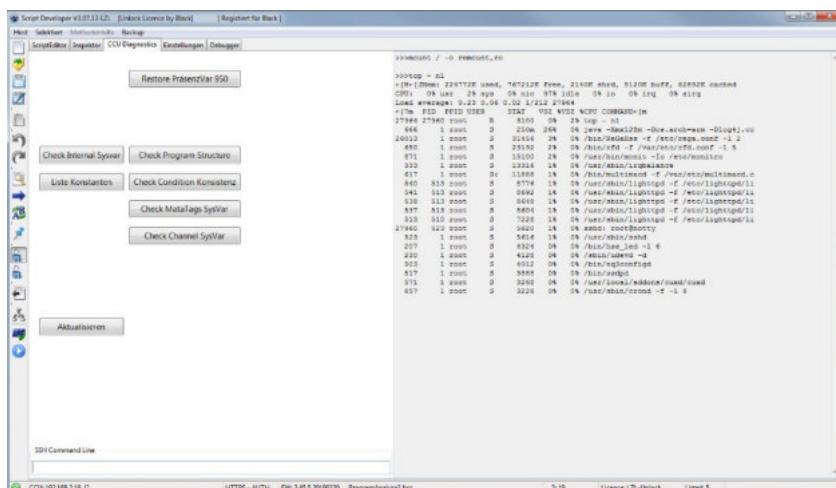
```
[HOSTCCU]
IP=192.168.2.xx
USERNAME=
PASSWORD=
USEHTTPS=False
SSHUSERNAME=root
SSHPW=
```

Ab der 3.08.xx kommt der SDV wieder ohne das PLink aus, hier wird dies nun über CUxD realisiert.

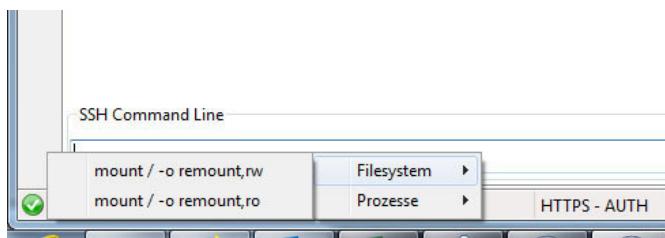
In dieser Zeile lassen sich nun einfache SSH Kommandos ausführen.



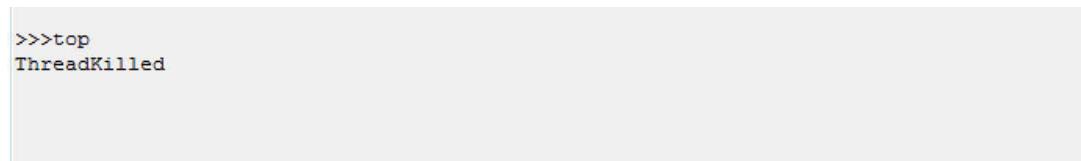
Dies soll und ist aber kein Ersatz für eine putty Konsole, eher gedacht für ich muss mal eben schnell einen Befehl ausführen oder man hat nicht die Befehlsfolgen im Kopf sondern benutzt die Hilfe. Programme dort zu starten oder z.b. top zusehen geht nicht und ist auch nicht geplant, das kann putty ganz klar besser. zweimal Rad erfinden ist auch nicht nötig.



Einige Funktionen sind als Hilfe mit rechter Maustaste auf die Kommandozeile programmiert, wird bei Bedarf von mir noch erweitert, Vorschläge dazu sind auch willkommen.



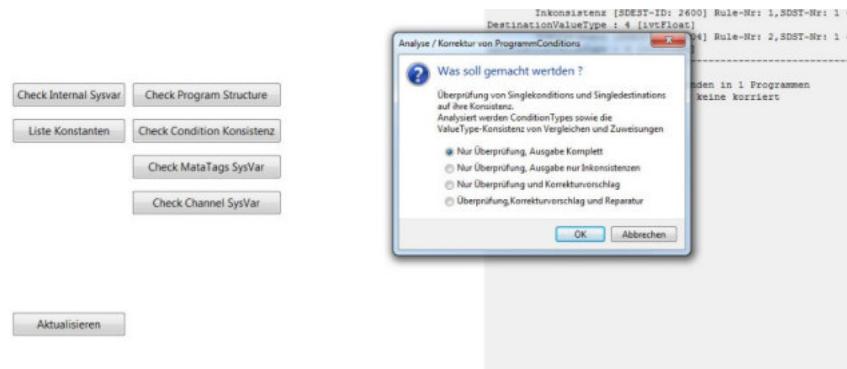
Programme dort zu starten oder z.b. top zusehen geht nicht und ist auch nicht geplant, das kann putty ganz klar besser. zweimal Rad erfinden ist auch nicht nötig. Also eine Ausführung von top rennt nach 10 Sekunden in den Thread Timeout Kill.



## 7.2 Programm Condition Konsistenz.

Durch häufiges Ändern, oder auch internen Fehlern in der WEB-UI kann es vorkommen, dass eigentlich in sich richtig ausschauende Programme in Ihren Objektstrukturen logische Fehler aufweisen und nicht mehr triggern oder ansonsten unlogisches Verhalten aufweisen. Der bekannteste Fehler z.B. ist ein Programm mit einem Stringvergleich welches aber reproduzierbar nicht triggern wird, weil die WebUi aus dem bei Strings eigentlich vorgesehenem ConditionType 1 (bei) eine 6 (ist kleiner bis gleich) macht.

Als Vorstufe zu der Erweiterung Program backups kann der SDV nun mir bekannte logische Inkonsistenzen erkennen, aufzeigen, und wenn möglich, einen Korrekturvorschlag nennen und diesen auf Anweisung auch ausführen.



Auswahl ist selbsterklärend.

Stufe 4 (mit Reparatur sollte immer erst nach Stufe 3 (Überprüfung mit Korrekturvorschlag) benutzt werden. Hier werden die Fehler aufgeführt und ein Lösungsvorschlag aufgezeigt. Wenn dieser schlüssig ist, gut, kann man den SDV drauf loslassen, wenns arg schräg aussieht kann oder muss man die Korrektur manuell machen.

```
ProgrammAnalyse Konsistenz SingleConditions & Singledestinations
Black in April 2019
-----
Analyse Program: Alarmtest [ID: 8528]
    Inkonsistenz [SCND-ID: 8567] Rule-Nr: 1,CND-Nr: 1, SCND-Nr: 1 -- LeftVal: 2 [ivtBinary], RightVal: 4 [ivtFloat]
        Ursache: Boolvergleich ivtBinary Links mit nicht ivtFloat Wert rechts
        Umrechnen von rechts Wert 15.000000 [ivtFloat] in true [ivtBoolean]
        Kann vom SDV korrigiert werden
-----
Analyse Program: Stringtest [ID: 8622]
    Inkonsistenz [SCND-ID: 9258] Rule-Nr: 1,CND-Nr: 1, SCND-Nr: 1 -- LeftVal: 20 [ivtString], RightVal: 16 [ivtInteger]
        Ursache: Stringvergleich ivtString Links mit nicht ivtInteger Wert rechts
        Umrechnen von rechts Wert 25 [ivtInteger] in 25 [ivtString]
        Kann vom SDV korrigiert werden
-----
Analyse Program: USV_01_SHUTDOWN [ID: 2003]
    Inkonsistenz [SDEST-ID: 2600] Rule-Nr: 1,SDST-Nr: 1 -- DestinationDP : 2 [ivtBinary], DestinationValueType : 4 [ivtFloat]
    Inkonsistenz [SDEST-ID: 2604] Rule-Nr: 2,SDST-Nr: 1 -- DestinationDP : 2 [ivtBinary], DestinationValueType : 4 [ivtFloat]
-----
Geprüft: 37 Programme
Dabei wurden 4 Inkonsistenzen gefunden in 3 Programmen
Von diesen 4 Inkonsistenzen wurden keine korrigiert
```

In diesem Beispiel werden gefundene Inkonsistenzen aufgezeigt, die Ursache dargelegt und die Lösung und ob der SDV diese, wenn er mit Reparatur gestartet wird, diese korrigieren kann.

Wenn ja, wird im einem Korrekturlauf diese Inkonsistenzen beseitigt.

Der SDV verändert dabei Programmobjekte..  
AUF JEDEN FALL VORHER EIN BACKUP MACHEN !

Es gibt kein UNDO, das geht nur über Restore

```
ProgrammAnalyse Konsistenz SingleConditions & Singledestinations
Black in April 2019

Analyse Program: Alarmtest [ID: 8528]
Inkonsistenz [SCND-ID: 8567] Rule-Nr: 1,CND-Nr: 1, SCND-Nr: 1 -- LeftVal: 2 [ivtBinary], RightVal: 4 [ivtFloat]
Ursache Boolvergleich ivtBinary Links mit nicht ivtFloat Wert rechts
Umrechnen von rechts Wert 15.000000 [ivtFloat] in true [ivtBinary]
Kann vom SDV korrigiert werden
Korrigiert durch RightValValType (ivtBinary)

Analyse Program: Stringtest [ID: 8622]
Inkonsistenz [SCND-ID: 9258] Rule-Nr: 1,CND-Nr: 1, SCND-Nr: 1 -- LeftVal: 20 [ivtString], RightVal: 16 [ivtInteger]
Ursache Stringvergleich ivtString Links mit nicht ivtInteger Wert rechts
Umrechnen von rechts Wert 25 [ivtInteger] in 25 [ivtString]
Kann vom SDV korrigiert werden
Korrigiert durch RightValValType (ivtString)

Analyse Program: USV_01_SHUTDOWN [ID: 2003]
Inkonsistenz [SDEST-ID: 2600] Rule-Nr: 1,SDST-Nr: 1 -- DestinationDP : 2 [ivtBinary], DestinationValueType : 4 [ivtFloat]
Inkonsistenz [SDEST-ID: 2604] Rule-Nr: 2,SDST-Nr: 1 -- DestinationDP : 2 [ivtBinary], DestinationValueType : 4 [ivtFloat]

Geprüft: 37 Programme
Dabei wurden 4 Inkonsistenzen gefunden in 3 Programmen
Von diesen 4 Inkonsistenzen wurden 2 korrigiert
```

Ein nochmaliger Lauf zeigt diese 2 korrigierbaren nun nicht mehr als inkonsistent an.

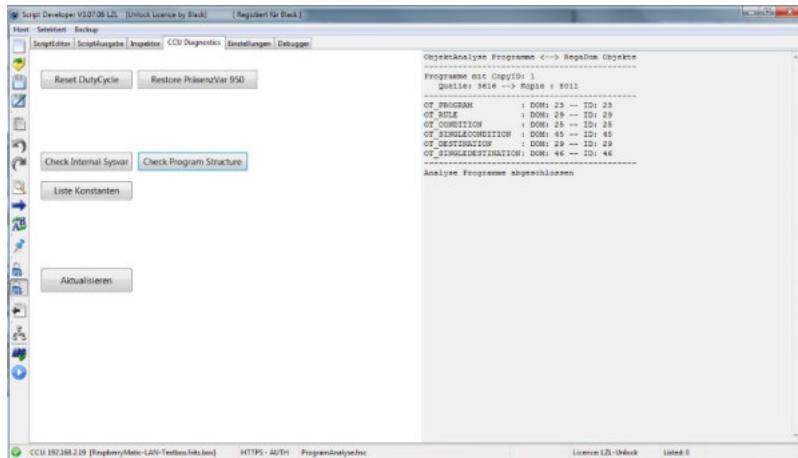
```
ProgrammAnalyse Konsistenz SingleConditions & Singledestinations
Black in April 2019

Analyse Program: USV_01_SHUTDOWN [ID: 2003]
Inkonsistenz [SDEST-ID: 2600] Rule-Nr: 1,SDST-Nr: 1 -- DestinationDP : 2 [ivtBinary], DestinationValueType : 4 [ivtFloat]
Inkonsistenz [SDEST-ID: 2604] Rule-Nr: 2,SDST-Nr: 1 -- DestinationDP : 2 [ivtBinary], DestinationValueType : 4 [ivtFloat]

Geprüft: 37 Programme
Dabei wurden 2 Inkonsistenzen gefunden in 1 Programmen
Von diesen 2 Inkonsistenzen wurden keine korrigiert
```

## 7.3 Programmanalyse

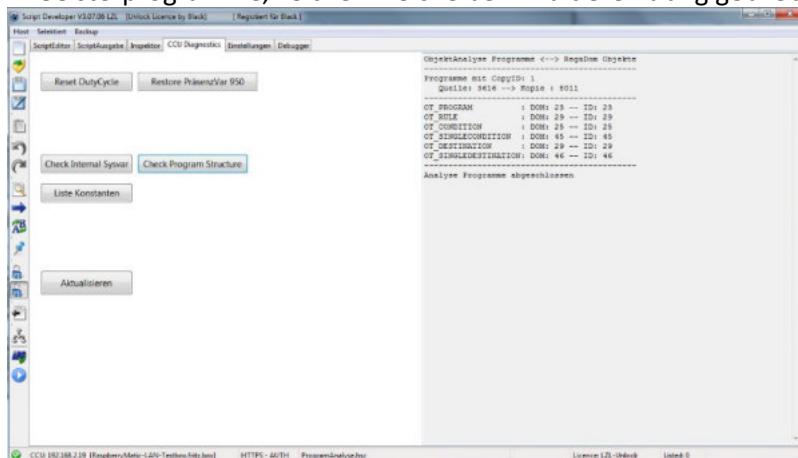
Mittels der Programmanalyse kann die Konsistenz der Programmobjekte überprüft werden. So sieht der Aufruf auf, wenn die Prüfroutine erfolgreich über die Programme gelaufen ist.



Keine Einträge in ProgrammCopyID, die weiteren Tests ergeben eine Übereinstimmung zwischen dem DomScan und der Analyser der Programme

### Mögliche Fehlerbilder:

#### 1. Geisterprogramme, Leichen welche beim Editieren übrig geblieben sind.



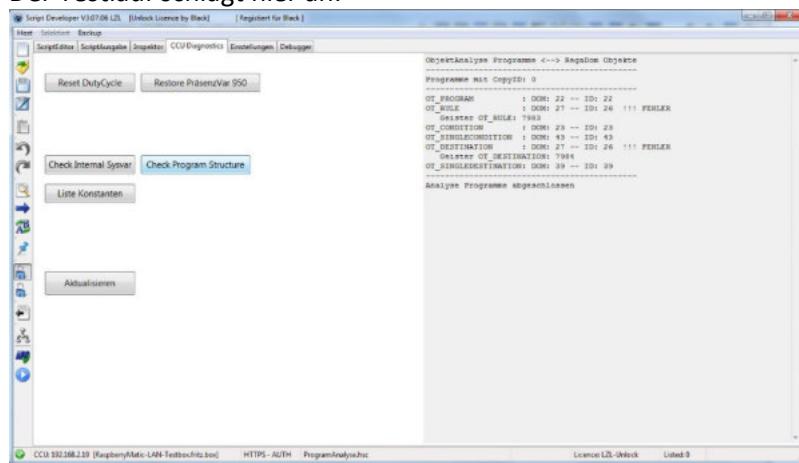
Hier wird das ursprüngliche Programm und die (Geisterkopie) daraus angezeigt. Der erste Versuch wäre: einen Reboot zu versuchen und zu überprüfen, ist der Eintrag des Geisterprogrammes weg oder nicht. Wenn nicht, kann manuell versucht werden, die Objekte der Kopie (nicht des Quellprogrammes) zu löschen

#### 2. Programmfragmentreste

In der Regadom tummeln sich Reste ehemaliger Programme, welche aber nicht mehr in der Struktur unter ID\_PROGRAMS auftauchen.

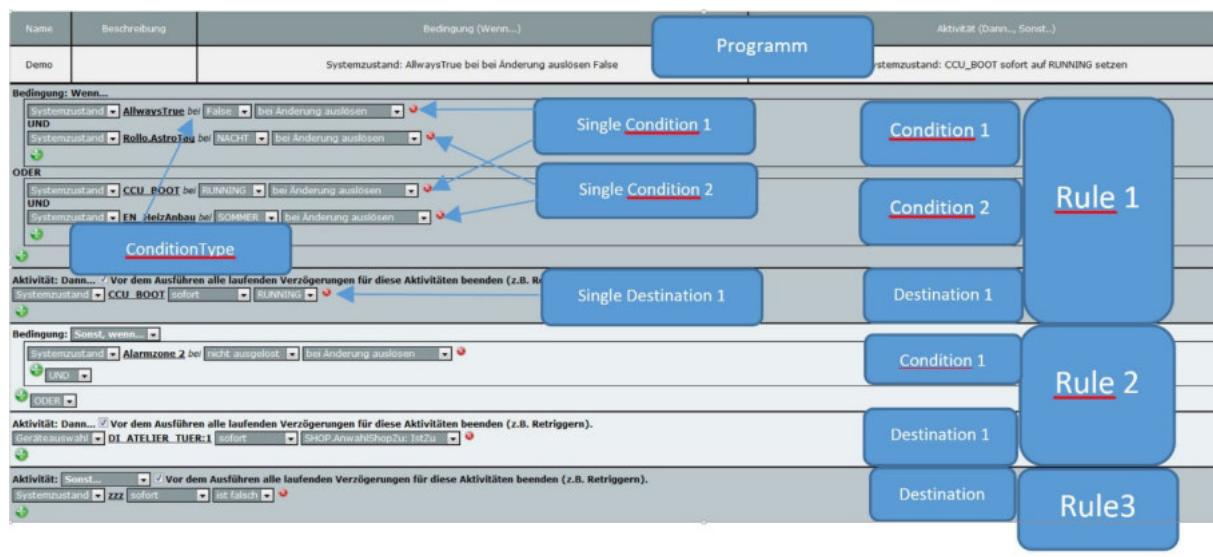
Provoziert habe ich es hier in dem Testfall, in dem ich ein RULE Object mal händisch angelegt habe.

Der Testlauf schlägt hier an:

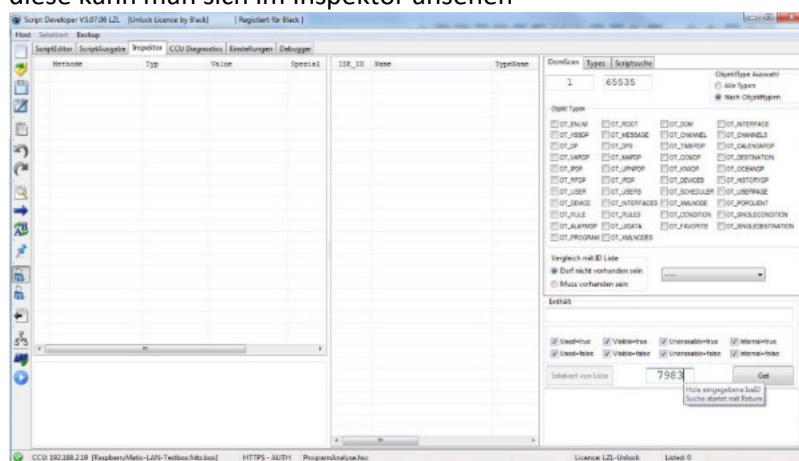


Es wird unter den Objecten OT\_RULE und OT\_DESTINATION angezeigt, dass sich in der Rega 2 Objekte tummeln, welche keinen Bezug zu dem Inhalt eines Programmobjektes haben. (2 deshalb, weil ein dom.CreateObject (OT\_RULE) auch gleichzeitig das zu der Rule gehörende OT\_DESTINATION Object erzeugt)

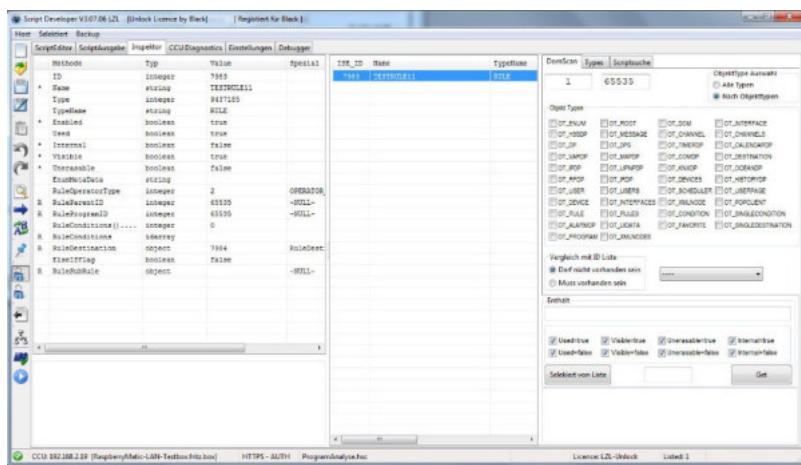
Ein Programm ist auf der CCU so Aufgebaut:



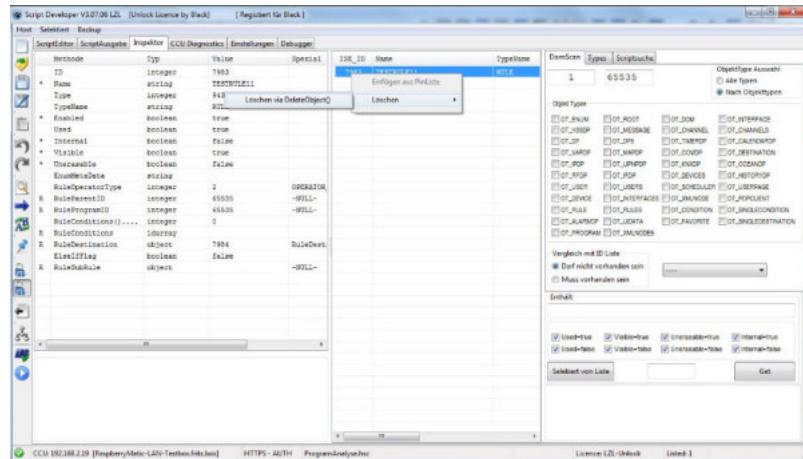
Die isID des gefundenen Objektes wird ausgegeben  
diese kann man sich im Inspektor ansehen



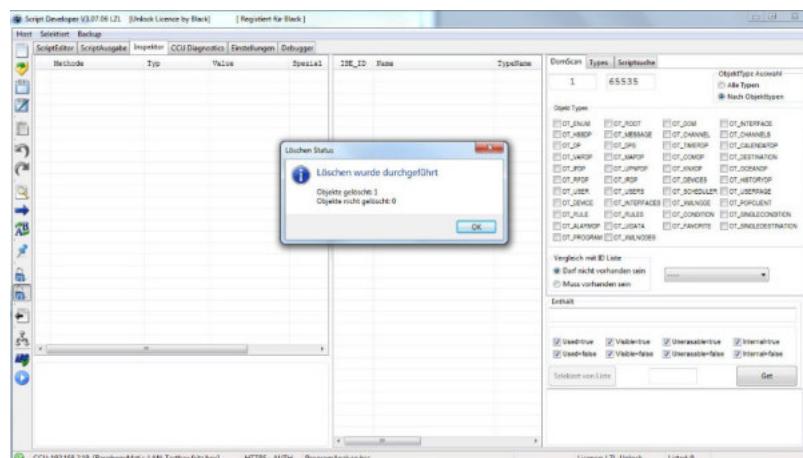
Dazu die gefundene ID in die Suchmaske eintragen und Enter drücken



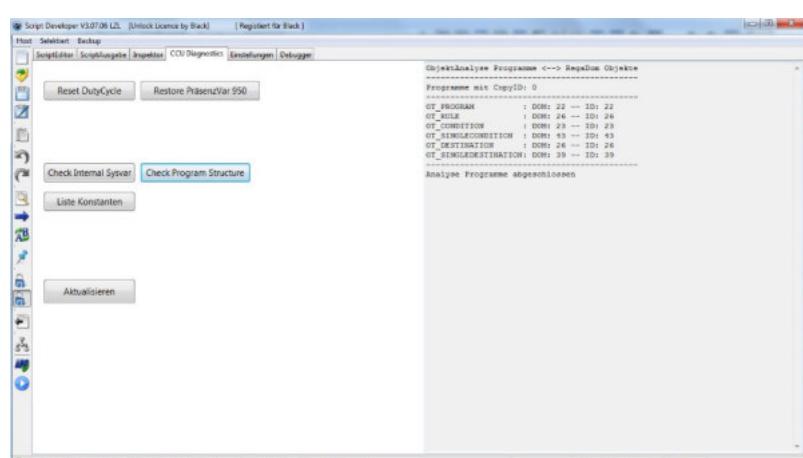
Man sieht, ein Object ohne Bezüge. es kann also entfernt werden. Löschen Freigeben, Programmunterobjekte müssen auch freigeben sein zum löschen, dann



es wurde gekillt



Die abschliessende Kontrolle zeigt:



Es ist weg.

**Und immer die Erinnerung: vor Löschen immer ein Backup machen, bei löschen gibt es kein Undo.**

## 7.4 Systemvariablen Metadatenkonsistenz

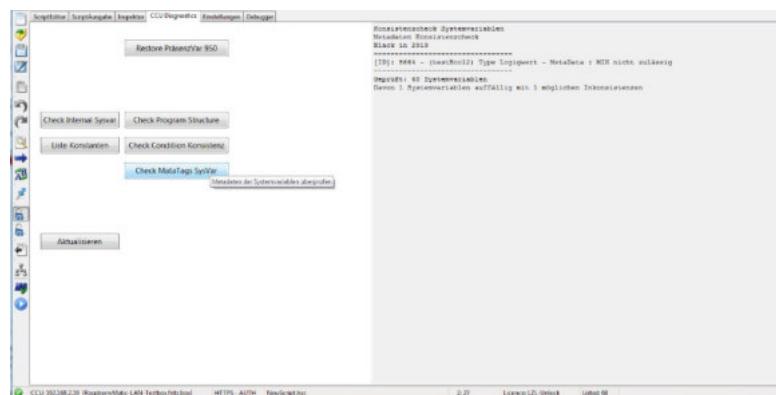
Dieser Menüpunkt überprüft die Systemdaten auf Konsistenz von dem Typ, welcher durch ValueType und ValueSubtype spezifiziert ist und den Einträgen unter EnumMetadata.

Druck erzeugt folgendes Menü:



### Punkt 1: nur Prüfen

Bringt eine Ausgabe in der Form, wie das Script, welches ich im Forum Online gestellt hatte. Mehr nicht. Er tut selber nix.



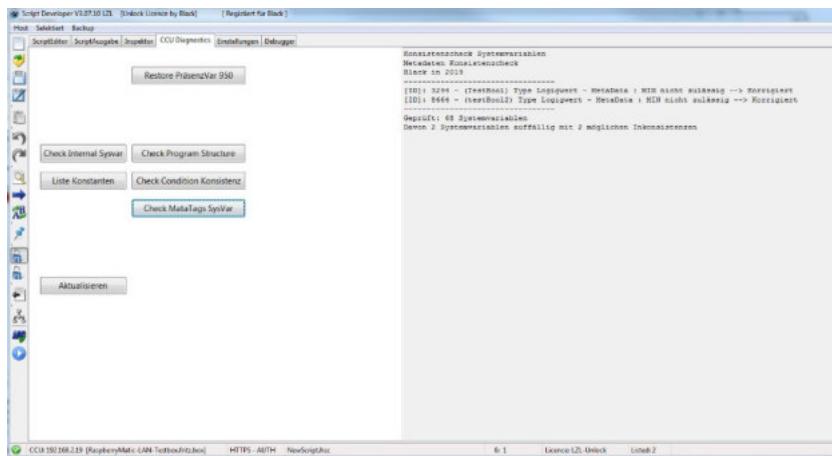
### Punkt 2: Nur Prüfen und Ergebnis in Explorer

Genau wie Punkt 1, es wird eine Überprüfung gemacht, zusätzlich wird das Ergebnis noch in die Auswahlliste des Inspektors geladen. Dort kann überprüft und gegenbenenfalls auch manuell korrigiert werden.

### Punkt 3: Korrigieren und Ergebnis in den Inspektor

Hier wird automatisch korrigiert und das Ergebnis dann in den Inspektor geladen.

VORHER AUF JEDENFALL EIN BACKUP MACHEN. Es gibt kein Undo hierbei.

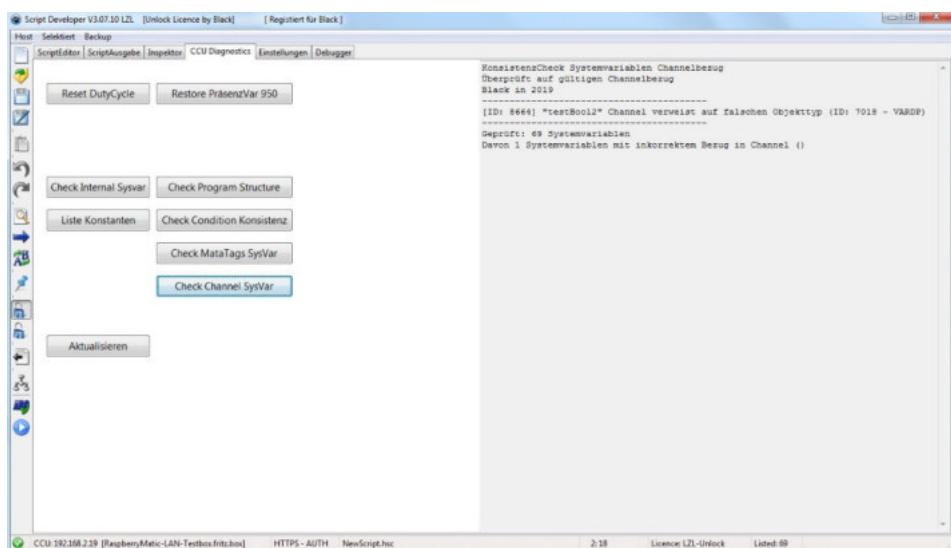
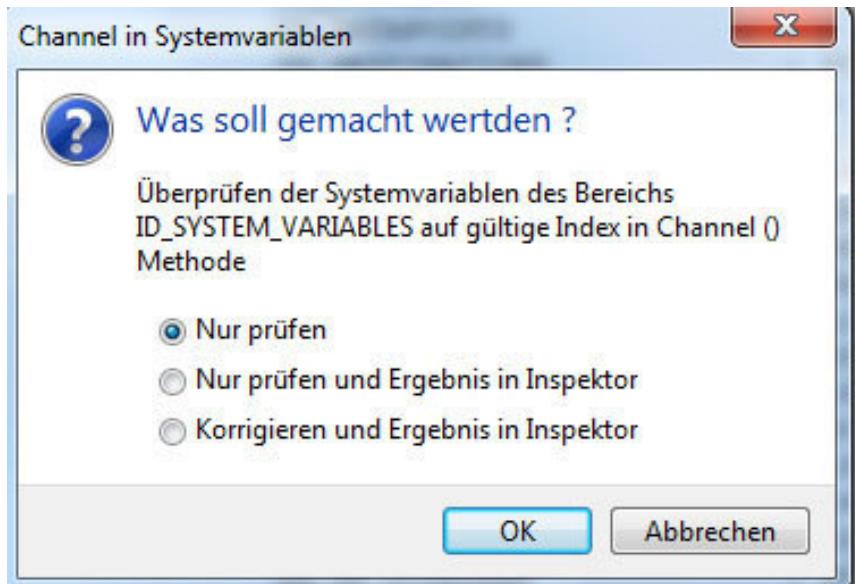


Ein nochmaliger Durchlauf bringt dann die Fehlerfrei-Nachricht.

```
Konsistenzcheck Systemvariablen
Metadaten Konsistenzcheck
Black in 2019
-----
-----
Geprüft: 68 Systemvariablen
Keine Auffälligkeiten festgestellt
```

## 7.5 Systemvariablen Test auf NirvanaChannelBezüge

Es wird überprüft, ob die Methode Channel () entweder Leer ist (ID\_ERROR) oder auf einen gültiges Channel Objekt verweist. Wenn nicht, handelt es sich um einen NirvanaChannelBezug.  
Hier auch: Auswahl was gemacht werden soll

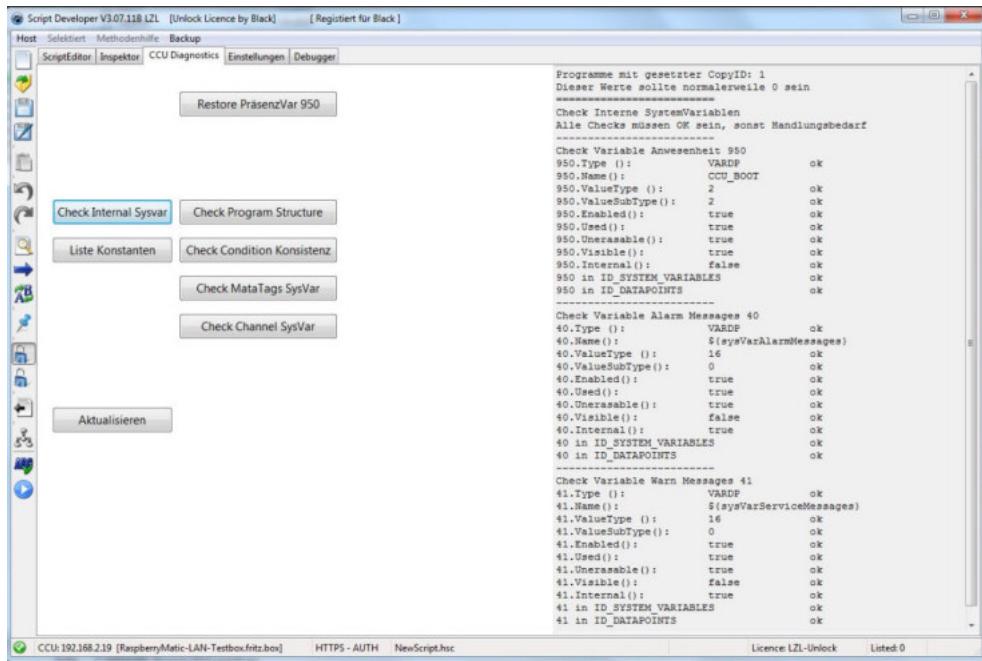


Die Auswahl Korrigieren bereinigt dann diese automatisch und lädt die veränderten Systemvariablen in den Inspektor.

```
KonsistenzCheck Systemvariablen Channelbezug
Überprüft auf gültigen Channelbezug
Black in 2019
-----
[ID: 8664] "testBool2" Channel verweist auf falschen Objekttyp (ID: 7018 - VARDP) --> korrigiert
-----
Geprüft: 69 Systemvariablen
Davon 1 Systemvariablen mit inkorrekttem Bezug in Channel ()
```

## 7.6 Check Internal Sysvar

Hierbei werden die Systeminternen auf Konsistenz geprüft (ID 950 Anwesenheit, ID 40 und 41 , Warnung und Alarm)



Alles Tests sollten OK melden, bei Fehler besteht Handlungsbedarf.

## 8 License Disclaimer

Mit diesem Programm werden 2 DLLs des OpenSSL Project mitgeliefert

Original Lizenztext:

### LICENSE ISSUES

---

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

### OpenSSL License

---

```
* =====
* Copyright (c) 1998-2016 The OpenSSL Project. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in
*    the documentation and/or other materials provided with the
*    distribution.
*
* 3. All advertising materials mentioning features or use of this
*    software must display the following acknowledgment:
*    "This product includes software developed by the OpenSSL Project
*    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
*
* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
*    endorse or promote products derived from this software without
*    prior written permission. For written permission, please contact
*    openssl-core@openssl.org.
*
* 5. Products derived from this software may not be called "OpenSSL"
*    nor may "OpenSSL" appear in their names without prior written
*    permission of the OpenSSL Project.
*
* 6. Redistributions of any form whatsoever must retain the following
*    acknowledgment:
```

\* "This product includes software developed by the OpenSSL Project  
\* for use in the OpenSSL Toolkit (<http://www.openssl.org/>)"  
\*  
\* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY  
\* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
\* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR  
\* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR  
\* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,  
\* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT  
\* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;  
\* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
\* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,  
\* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)  
\* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED  
\* OF THE POSSIBILITY OF SUCH DAMAGE.  
\* ======  
\*  
\* This product includes cryptographic software written by Eric Young  
\* (eay@cryptsoft.com). This product includes software written by Tim  
\* Hudson (tjh@cryptsoft.com).  
\*  
\*  
\* Original SSLeay License

-----  
\* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)  
\* All rights reserved.  
\*  
\* This package is an SSL implementation written  
\* by Eric Young (eay@cryptsoft.com).  
\* The implementation was written so as to conform with Netscapes SSL.  
\*  
\* This library is free for commercial and non-commercial use as long as  
\* the following conditions are aheared to. The following conditions  
\* apply to all code found in this distribution, be it the RC4, RSA,  
\* Ihash, DES, etc., code; not just the SSL code. The SSL documentation  
\* included with this distribution is covered by the same copyright terms  
\* except that the holder is Tim Hudson (tjh@cryptsoft.com).  
\*  
\* Copyright remains Eric Young's, and as such any Copyright notices in  
\* the code are not to be removed.  
\* If this package is used in a product, Eric Young should be given attribution  
\* as the author of the parts of the library used.  
\* This can be in the form of a textual message at program startup or  
\* in documentation (online or textual) provided with the package.  
\*  
\* Redistribution and use in source and binary forms, with or without  
\* modification, are permitted provided that the following conditions  
\* are met:  
\* 1. Redistributions of source code must retain the copyright  
\* notice, this list of conditions and the following disclaimer.  
\* 2. Redistributions in binary form must reproduce the above copyright  
\* notice, this list of conditions and the following disclaimer in the

\* documentation and/or other materials provided with the distribution.

\* 3. All advertising materials mentioning features or use of this software

\* must display the following acknowledgement:

\* "This product includes cryptographic software written by

\* Eric Young (eay@cryptsoft.com)"

\* The word 'cryptographic' can be left out if the routines from the library

\* being used are not cryptographic related :-).

\* 4. If you include any Windows specific code (or a derivative thereof) from

\* the apps directory (application code) you must include an acknowledgement:

\* "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

\*

\* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND

\* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

\* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

\* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE

\* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

\* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS

\* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

\* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT

\* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY

\* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF

\* SUCH DAMAGE.

\*

\* The licence and distribution terms for any publically available version or

\* derivative of this code cannot be changed. i.e. this code cannot simply be

\* copied and put under another distribution licence

\* [including the GNU Public Licence.]

\*/