

Kurzanleitung ScriptDeveloper SDV V3.10.02 (10. Januar 2020)

Der Scriptdeveloper (SDV) soll ein Hilfsmittel im Alltag bei der Erstellung von Homematic Skripten und deren Tests darstellen. Ein gewisses Wissen über Skripterstellung 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 V3.10.02 (10. Januar 2020)	1
1. Installation	6
1.1 Lizenzierung	8
1.2 Donate	13
1.3 Systemvoraussetzungen	14
1.4 Was tut es bis jetzt	15
1.5 Bekannte Einschränkungen / Bugs	16
1.6 Geänderte Anforderung an Auflösung	16
1.6.1 Mehrfacher Start	16
1.7 Changelog	17
1.7.1 Changelog 03.10.02 LCL	17
1.7.2 Changelog 03.10.01 LCL	17
1.7.3 Changelog 03.09.05 LCL	17
1.7.4 Changelog 03.09.04 LCL	17
1.7.5 Changelog 03.09.03 LCL	17
1.7.6 Changelog 03.09.02 LCL	18
1.7.7 Changelog 03.09.01 LCL	18
1.7.8 Changelog 03.08.14 LCL	18
1.7.9 Changelog 03.08.13 LCL	18
1.7.10 Changelog 03.08.12 (A-D) LCL	18
1.7.11 Changelog 03.08.11 LCL	18
1.7.12 Changelog 03.08.10 LCL	18
1.7.13 Changelog 03.08.09 LCL	18
1.7.14 Changelog 03.08.07 LCL	19
1.7.15 Changelog 03.08.05 LCL	19
1.7.16 Changelog 03.08.04 LCL	19
1.7.17 Changelog 03.08.01 LCL	19
1.7.18 Changelog 03.07.13 LZL	19
1.7.19 Changelog 03.07.12 LZL	19
1.7.20 Changelog 03.07.11 LZL	20
1.7.21 Changelog 03.07.10 LZL	20
1.7.22 Changelog 03.07.07 LZL	20
1.7.23 Changelog 03.07.05 LZL	20
1.7.24 Changelog 03.07.04 LZL	20
1.8 Neue Authentifizierungsmechanismen Raspberrymatik	21

1.8.1 SSL.....	21
1.8.2 Authentifizierung über Nutzernamen / Passwort.....	21
1.9 Ein Wort des Autors zum Thema „gewerbliche Nutzung“	23
2. Oberfläche.....	24
2.1 Word Markup (seit 03.07.02).....	25
2.1.1 SyncroEdit (ab 3.08.10)	26
2.1.2 Schriftgrößen und Schriftart (ab 3.08.07)	27
2.2 Blockweise auskommentieren.	28
3 Scripteditor.....	29
3.1 Voreinstellungen Editor.....	31
3.2 Vervollständigen Funktion	32
3.3 Methodenhilfe.....	34
3.4 Symbolischer Zugriff auf selektierte Datenpunkte des Inspektors	36
3.5 Automatische Codeerzeugung für Read/Write von Master/Linksets	37
4 Inspektor.....	39
4.1 Selektionswahl: DomScan	40
4.2 Selektionskriterium Types	43
4.3 Zusätzliche Selektionsbedingungen	44
4.4 Daten aus Inspektor in Editor übernehmen.....	51
4.4.1 Mehrfachauswahl als Enum String.....	53
4.4.2 Übernahme von einem Script aus einem Programm direkt in den Editor.....	55
4.5 Selektion von Selektion	58
4.6 Objekte löschen.....	59
4.7 Anwenderdefinierte Sichten	61
4.8 Browsing durch Rekursionsebenen.....	62
4.9 Auflösen komplexer Objekte.....	65
4.9.1 Auflösen von Devices	65
4.9.2 Auflösen von Programmen.....	67
4.10 Zeitmodule	68
4.11 Suchen in Scripten nach Variablen, Devices etc.....	69
4.12 Volltextsuche in Scripten.....	72
4.13 Suchen Verwendungsstelle in WEB-UI Programmen (ab 3.09.04).....	74
4.14 Suchen „Name“ oder „Adress“ nach Verwendung in Scripten	75
4.15 Programm suchen verwendete Systemvariablen	75
4.16 SingleDestinations in ihrer Reihenfolge ändern.....	76
4.17 History Data Points und Systemprotokoll	79

5 Favoritenansicht	80
5.1 Hinzufügen von Objekten in die Favoriten.....	80
5.2 Verwendete Objekte eines Programmes in die Favoriten laden	81
5.3 Favoritenliste löschen.....	81
5.4 Einschränkungen	81
6 Kleine Helfer im Alltag.....	82
6.1 Backups.....	82
6.1.1 Räume.....	82
6.1.2 Gewerke	83
6.1.3 Systemvariablen	84
6.1.4 Devices und Kanäle.....	84
6.1.5 Backup Programme	85
6.1.6 Backup Masterset.....	90
6.1.7 Backup Linkset.....	91
6.1.8 Backup Gerät von der WebUI.....	92
6.2 Umbenennen von Kanälen von Geräten	94
6.3 Paramset Master	95
6.4 Linkset eines Gerätes.....	97
6.5 Rega Push auf Datenpunkte via Rega event.....	98
6.6 Querverweise zu Datenpunkten in WebUI-Programmen	99
7 Diagnosebild	100
7.1 Allgemein.....	100
7.1.1 Liste Konstanten	100
7.1.2 Regadom.....	100
7.1.3 Systemübersicht	100
7.1.4 CacheInfo.....	101
7.1.5 Regadom Aktualisierung.....	101
7.1 SSH Funktionalität	102
7.1.1 SSH Realisierung (CUxD ab 2.3.1 oder plink.exe)	104
7.2 Programm Condition Konsistenz.....	105
7.3 Programmanalyse.....	107
7.4 Systemvariablen Metadatenkonsistenz	111
7.5 Systemvariablen Test auf NirvanaChannelBezüge	113
7.6 Test und Korrektur Zeitmodule	114
7.7 Test und Korrektur HistoryDPs.....	115
7.8 Check Internal Sysvar	117

7.8	Check Objects	117
7.9	Check Devices <-> XMLRPC	118
7.10	Check Devices <-> XMLRPC	118
8	Gerätekopieren	120
8.1	Kopieren Masterset	121
8.2	Gerätetausch	123
8.3	Kopieren Week Profiles	123
8.4	Kopieren Heizprofile	124
9	License Disclaimer	127
9.1	Compiler	129

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 Filetypen 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=DerNickNameAusDerLizenzanfrage
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     Nutzername auf der CCU1
PASSWORD=XXXXXXXXXX     Passwort der Nutzers auf der CCU1
USEHTTPS=true           Zugriff über HTTPS
SSHUSERNAME=root        root wenn SSH Zugriff erwünscht
SSHPW=xxxx^            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=xxxx^            Das SSH Passwort der CCU eintragen

[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=ExternAdmin
PASSWORD=XXXXXXXXXX
USEHTTPS=True
SSHUSERNAME=root        root wenn SSH Zugriff erwünscht
SSHPW=xxxx^            Das SSH Passwort der CCU eintragen

[SECURITY]                Die Ports die der SDV benutzt
HTTPREGASCRIP T=8181
HTTPREGAXMLRPC=1999
HTTPSREGASCRIP T=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 Nutzernamen 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 exklusiven Zugriff hat. Auf Systemen ohne CUxD kann der SDV nicht eingesetzt werden.

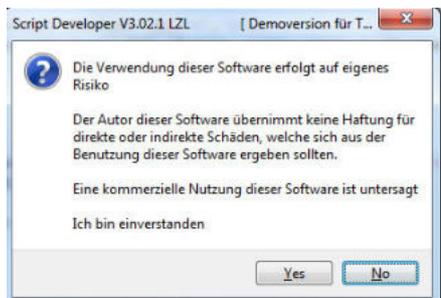
1.1 Lizenzierung

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 einzugrenzen. Dies geschieht durch Vergabe von bis zu 2 Lizenzschlüsseln. Der SDV ist dadurch an bis zu 2 CCU / Raspberrymatic gepairt.

Wie arbeitet das ?

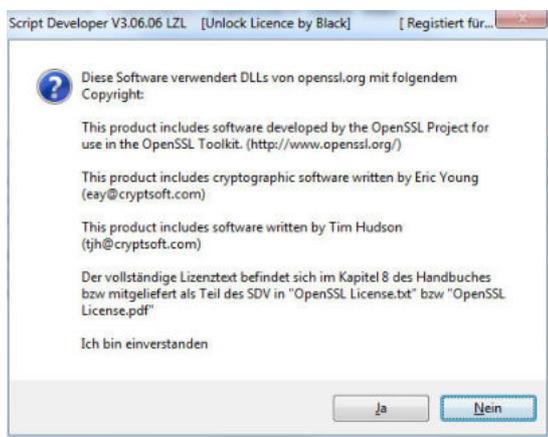
Der SDV telefoniert nicht nach Hause. Um eine Lizenz anzufragen ist folgender Weg einzuschlagen.

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



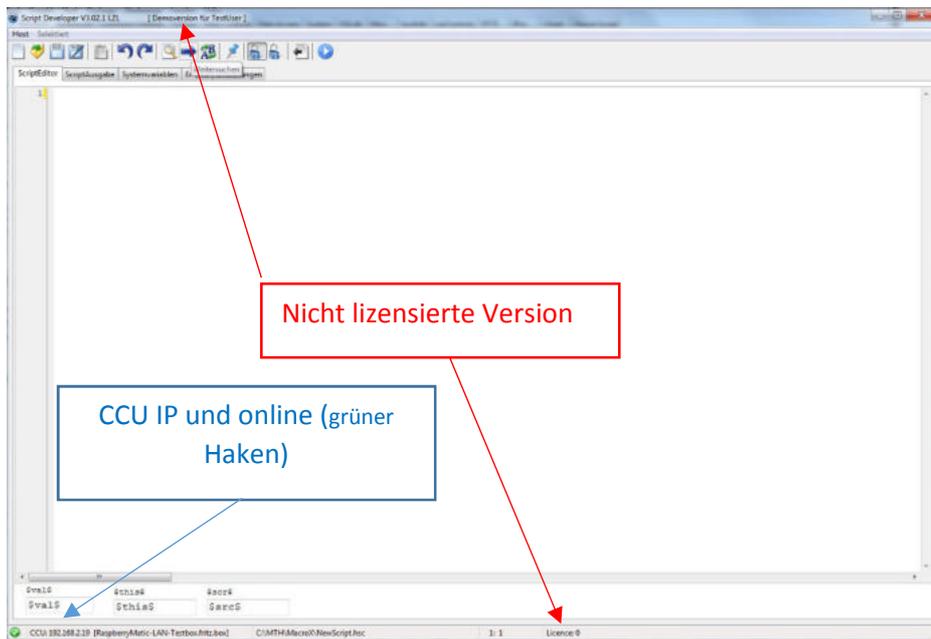
Bei allerersten Start muss dieses Fenster mit yes bestätigt werden. No führt so einem sofortigen Programmabbruch

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 9 nachgelesen werden bzw. ist als Datei mit in dem Package enthalten.

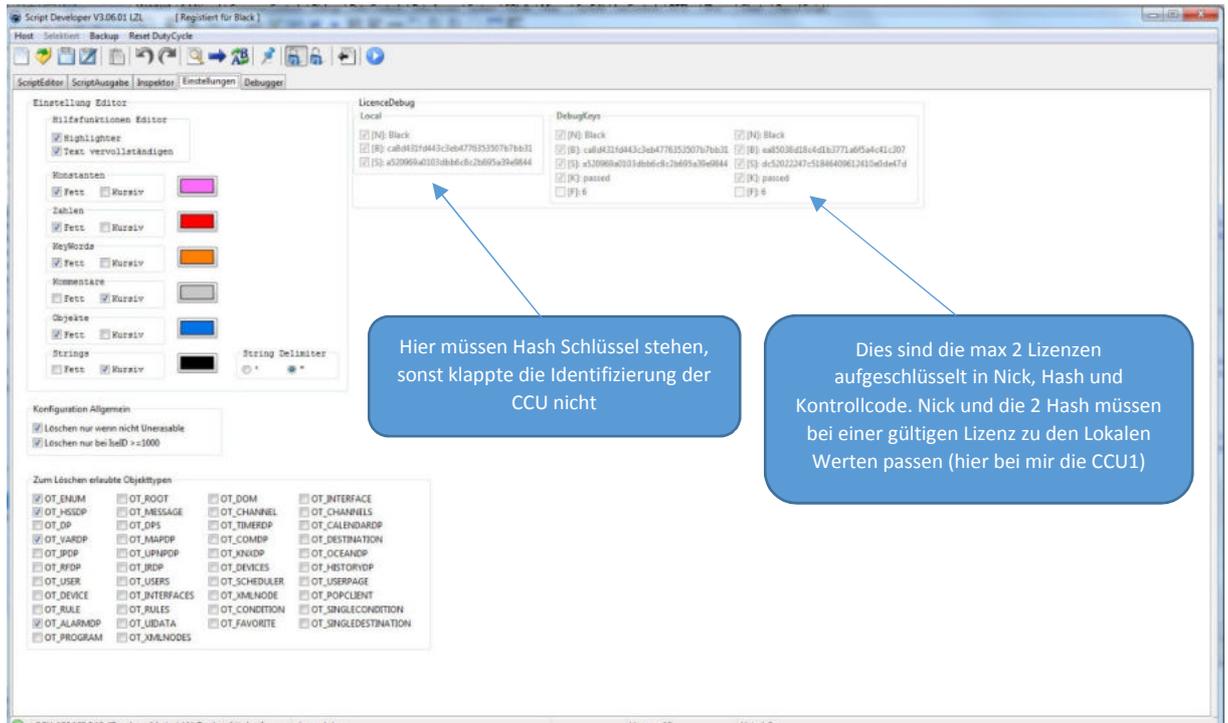


„Nein“ führt auch hier zu einem sofortigen Programmabbruch

Bei Bestätigung mit Yes startet nun zum erstmal 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

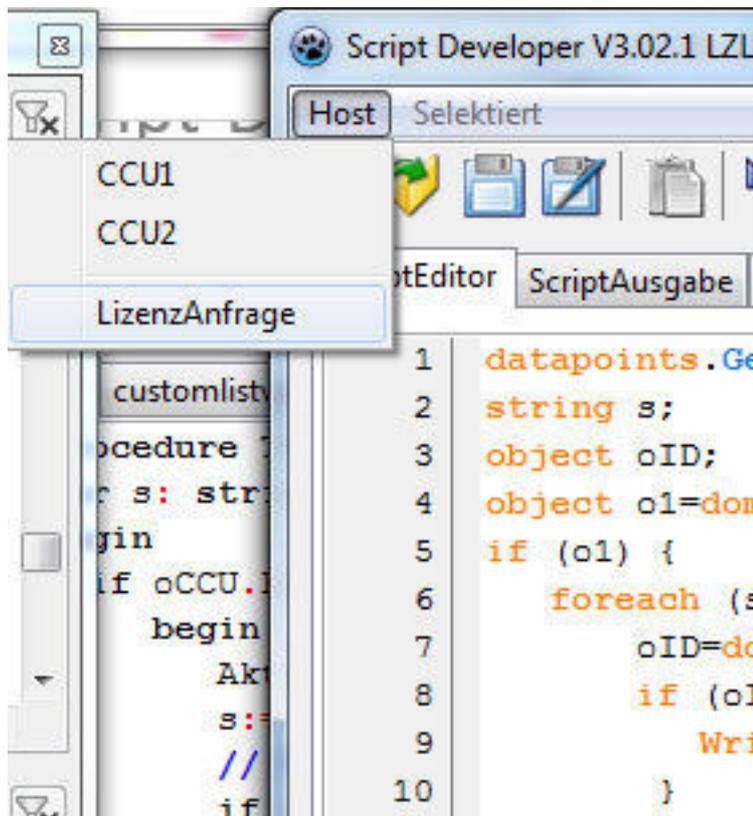


Aus einer Anfrage ohne unter local sinnige Einträge zu sehen lässt sich kein gültiger key generieren.

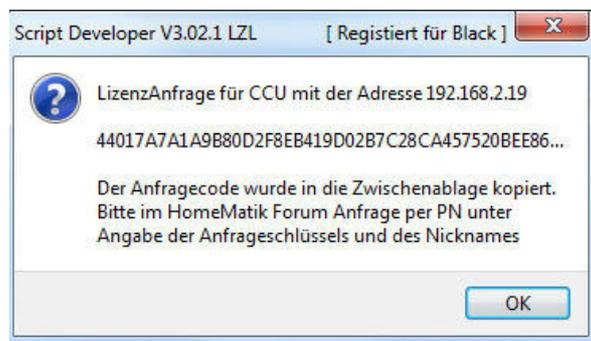
Aufgrund einer Programmgenauigkeit bei der Ausformulierung einer RegEx ist es leider sehr wahrscheinlich, dass alte Lizenzschlüssel vor 3.5.2 als ungültig erkannt werden. Sorry dafür, in dem Fall bitte die Anfrageschlüssel neu erstellen (wenn die Hashwerte gültig sind) und mir schicken. Der SDV 3.5.0 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.

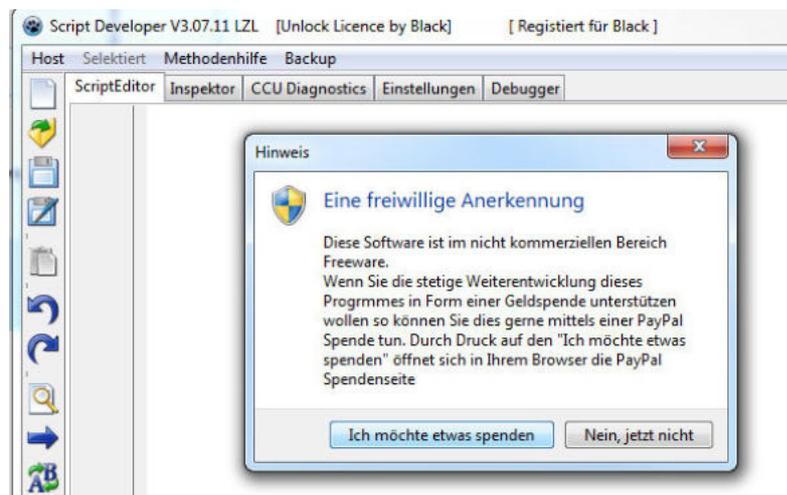
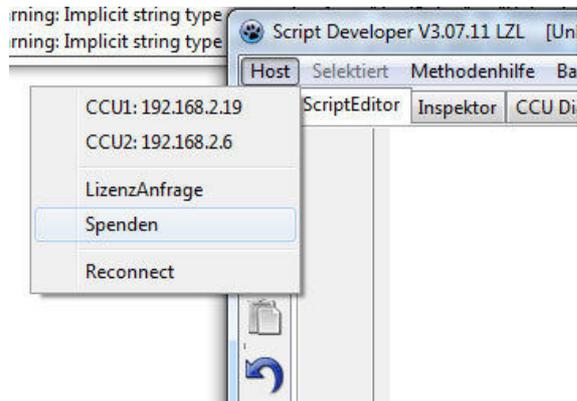
Wer mit diesem Verfahren nicht einverstanden ist, möge bitte an dieser Stelle die PDF Datei schließen und kann die Dateien beruhigt löschen.

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 Testinstallationen 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 3.47.18.20190918 getestet.

Auf einer CCU sind die erzeugten 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 ich es nativ unter Debian auch kompiliert und zum Laufen bekomme. Wahrscheinlich wird dieses Projekt aber zurückgestellt, da leider das Threathandling unter Linux sich völlig anders wie Windows verhält.

Ab der Version 3.09.xx wird der SDV unter Lazarus 2.0.4 als 64Bit Programm compiliert.

In dem Package sind nun auch die SSL DLLs für die 64bit Version enthalten. In einem Bestandssystem müssen dann die DLLs ausgetauscht werden. Ein 64bit SDV mit 32bit DLLs führt zu einem SIGTERM.

1.4 Was tut es 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 Editiermöglichkeiten.

Darstellbarkeit zumindest der Grundmethoden aller Objekte

DomScan

Devices

Aufschlüsseln der MetaDaten

Datenpunkte

Kanäle

Programme

Favoriten

User

Paramset Master

Paramset Links

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

Backup von Mastersets

Backup von LinkSets

Backup von WebUI Einstellungen eines Gerätes (Zuordnung zu Räumen, Gewerken Favoriten und Programmen)

Diverse Konsistenztests

Auflösen von Direktverbindungen

Backup von WEBUI Programmen und Einstellungen eines Gerätes

Devicekopien auch unter „ähnlichen“ Geräten

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. (Edit: History ist mittlerweile programmiert, Diagramm gibt es aber nicht und wird es auch nicht geben)

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.

1.6.1 Mehrfacher Start

Der SDV kann mittlerweile mit mehreren Instanzen gleichzeitig gestartet werden. Dabei hat allerdings nur die zuerst gestartete Instanz Schreibrechte auf die INI Datei. So lassen sich mehrere Scripte gleichzeitig bearbeiten und testen oder auch gleichzeitig mehrere Inspektoren benutzen.

Zu empfehlen ist auch, z.B. die Dateierweiterung .hsc mit dem SDV zu verknüpfen, so öffnet ein Doppelklick auf eine Scriptdatei dann automatisch den SDV mit dem geladenen Script

1.7 Changelog

1.7.1 Changelog 03.10.02 LCL

Ein paar kleinere Bugfixes
Gerätekopien Mastersets zwischen gleichen und ähnlichen Devices
Gerätekopien IP-Wochenprofile auch zwischen unterschiedlichen IP Geräten
Gerätekopien Heizprofile übergreifend zwischen HM, HMIP und Heizgruppen möglich

1.7.2 Changelog 03.10.01 LCL

Internes zwischenrelease, reiner teststatus

1.7.3 Changelog 03.09.05 LCL

Ein paar kleinere Bugfixes
Programmkonsistenztest verschärft (Zeitmodule und Scripte und GeisterChannel verweise)
Backup von Räumen Gewerken Fix eingeführt (Enumtypes sind durchgegangen)
Korrekturlauf für Enums eingeführt, die diese Bezüge reparieren können
Fix bzw Ungenauigkeit bei Backup Linkset für ein gerät beseitigt
Aus Mastersets / Linksets lassen sich Werte markieren und zum Auslesen der entsprechende Scriptcode generieren
Korrekturlauf CheckObjects (Richtiger Verweis und korrekter Gegenverweis)
Möglichkeit nach Name () oder Adress () eines Gerätes/Channel in Scripten zu suchen
Möglichkeit in einem Programm die verwendeten Systemvariablen zu suchen
CheckDevices gegen XMLRPC Schnittstellenprozess eingeführt
Löschmöglichkeit von Devices nun auch xmlrpc.DeleteDevices ()
Backup Device und Channelnamen nun über die Selektieren Devices, nicht mehr über alle.

1.7.4 Changelog 03.09.04 LCL

Mittlerweile 64 Bit Compiliert (verwendete RAD: 2.0.4)
Diverse kleinere Bugfixes
Auflösen von Linksets möglich
Ändern von Namen und Beschreibung von Linksets
Backup eines Linksets
Parameter im Linkset können markiert werden und automatischer Editorcodeerzeugt werden
Editor kann mit Human friendly JSON umgehen
Auflösen von Direktverbindungen selektierte Geräte
Sortieren der Direktverbindungen nach Sender, Empfänger
Finden Gleichartiger Geräte
Echtzeithilfe eingefügt
Automatische Codegenerierung für Datenpunkte (Vollsymbolischer Zugriff)
Konfigurierbarer Threatkill Timeout
Für die jeweiligen CCUs separates BackupDir eingefügt
Restore Möglichkeit, wenn ein Device gelöscht wurde, um sämtliche Punkte der WEBUI wieder zu rekonstruieren
Linkset Auflösen von Level6 auf Level 5 herabgesetzt
Master und Linkset Level von 6 auf 5 heruntergesetzt
Logmöglichkeit für Backup programme eingeführt
Möglichkeit benutzerdefinierte Dateinamen für Programmbackup

1.7.5 Changelog 03.09.03 LCL

Mehrere kleinere Bugfixes
Restore Möglichkeit, wenn ein Device gelöscht wurde, um sämtliche Punkte der WEBUI wieder zu rekonstruieren
Linkset Auflösen von Level6 auf Level 5 herabgesetzt
Master und Linkset Level von 6 auf 5 heruntergesetzt
Logmöglichkeit für Backup programme eingeführt

1.7.6 Changelog 03.09.02 LCL

Kleinere Bugfixes

Parameter im Linkset können markiert werden und automatischer Editorcodeerzeugt werden

Editor kann mit Human friendly JSON umgehen

Auflösen von Direktverbindungen selektierte Geräte

Sortieren der Direktverbindungen nach Sender, Empfänger

Finden Gleichartiger Geräte

Echtzeithilfe eingefügt

Automatische Codegenerierung für Datenpunkte (Vollsymbolischer Zugriff)

Konfigurierbarer Threatkill Timeout

Für die jeweiligen CCUs separates BackupDir eingefügt

1.7.7 Changelog 03.09.01 LCL

Update der RAD auf 2.0.4

Umbau im Untergrund, um die Classen für die geplanten Master und Linkset Manipulationen vorzubereiten

Auflösen von Linksets möglich

Ändern von Namen und Beschreibung von Linksets

Backup eines Linksets

1.7.8 Changelog 03.08.14 LCL

Letzte Version unter 32 bit unter Lazarus 1.8.4

Backup Devices eingeführt

Auflösen von Linksets möglich

1.7.9 Changelog 03.08.13 LCL

Inspektor: Device mit allen Kanälen aus Raumlisten löschen

Inspektor: Device mit allen Kanälen aus Gewerkelisten löschen

Konsistenzcheck History DPS (Ungültige und verwaiste History DPs aufspüren und beseitigen)

1.7.10 Changelog 03.08.12 (A-D) LCL

Proof of Concept Version mit Umgebauter interner Verarbeitung des Highlighters

B-Hash Berechnung entfernt, EQ3 kriegt unter bestimmten Voraussetzungen keine valide B-Ausgabe hin)

Filter für HUE und Lightify Geräte dem Package hinzugefügt

Möglichkeit 10 Eigener Methoedenhilfen mit Tastenkurzanwahl

1.7.11 Changelog 03.08.11 LCL

Verbesserung im Editor (Stringdelimiter angabe nicht mehr nötig)

Highlighter verbessert in Hinsicht auf String und Kommentarverarbeitung

Ansatz von Echtzeit Hilfe zu Methoden

Möglichkeit ungültiger Dateinamen durch ein Backup-Programme beseitigt

Diverse Kleinere Bugfixes

1.7.12 Changelog 03.08.10 LCL

SynchoEdit eingefügt

Parallele Completions eingefügt

Echtzeit Completion für definierte Variablen eingefügt

1.7.13 Changelog 03.08.09 LCL

Interne Modifikationen, so das für Scriptausführen nun nicht mehr ein interner Aufruf con CUXD gebraucht wird

Wahlmöglichkeit über INI Datei: SSH Funktionalität über pling oder über CUXD

Ein paar kleinere Fixes zur Verbereitung auf den Syntax checker

Menügesteuerter RTU Upate

Für Raspberrymatik muss die CUXD version Ungleich 2.3.0 sein !!!! aktuell heute 2.3.1

1.7.14 Changelog 03.08.07 LCL

Sortieralgorithmus der Listendarstellung optimiert
Erkennen und Beseitigen von Conditionchannel und Destinationchannel Inkonsistenzen hinzugefügt
ein kleiner Bugfix bei einem Ausgabertext

1.7.15 Changelog 03.08.05 LCL

- Bugfix : Backup von Räumen / Gewerken und Sysvars speicherte nicht als Datei. (behooben)
- Neu: - SDV kann SingleConditions auch so bearbeiten, das indirekte Vergleiche möglich sind. (Patch für Konsistenzcheck fehlt aber noch !)
- Editor und Listendarstellung Schrift und Grösse änderbar
- History Datapoints hinzu

1.7.16 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

1.7.17 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 detaildarstellung des Selektierten Objectes (undo redo Browsing ist dabei berücksichtigt)

1.7.18 Changelog 03.07.13 LZL

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

1.7.19 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

1.7.20 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

1.7.21 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

1.7.22 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

Selektierbar, auch suchen in GeisterDestinations möglich

Suchen auch als Regular Expression möglich

Beschleunigte Suche in Ram möglich

ProgrammObjektanalyse

1.7.23 Changelog 03.07.05 LZL

Kleine Bugfixes

Bugfix bei Programme Testen („\n“ in einem Kommentar führte zu einem Fehler) Bugfix tuts 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

Regaports werden nun Automatisch ermittelt

Alternative Endungen für Scripte in INI hinzugefügt

Alternatives Scriptverzeichnis in INI hinzugefügt

1.7.24 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 Nutzernamen / Passwort. 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.

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 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 Nutzernamen / Passwort.

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

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

Beispielhafter Aufbau der neuen INI

[LAST]

DATEI=c:\MTH\Homematic\NewScript.hsc

[HOST]

NICKNAME=DerNickNameAusDerLizenzanfrage

CUXD=CUXD.CUX2801001:5

SYSTEMEXEC=true

LICENCE1=DerLizenzschlüssel1

LICENCE2=DerLizenzschlüssel2

[CCU1]

Neuer Schlüssel

IP=192.168.2.19

Ip Von CCU1

USERNAME=Admin

Nutzernamen auf der CCU1

PASSWORD=XXXXXXXXXX

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=XXXXXXXXXX

USEHTTPS=True

[SECURITY]

Die Ports die der SDV benutzt

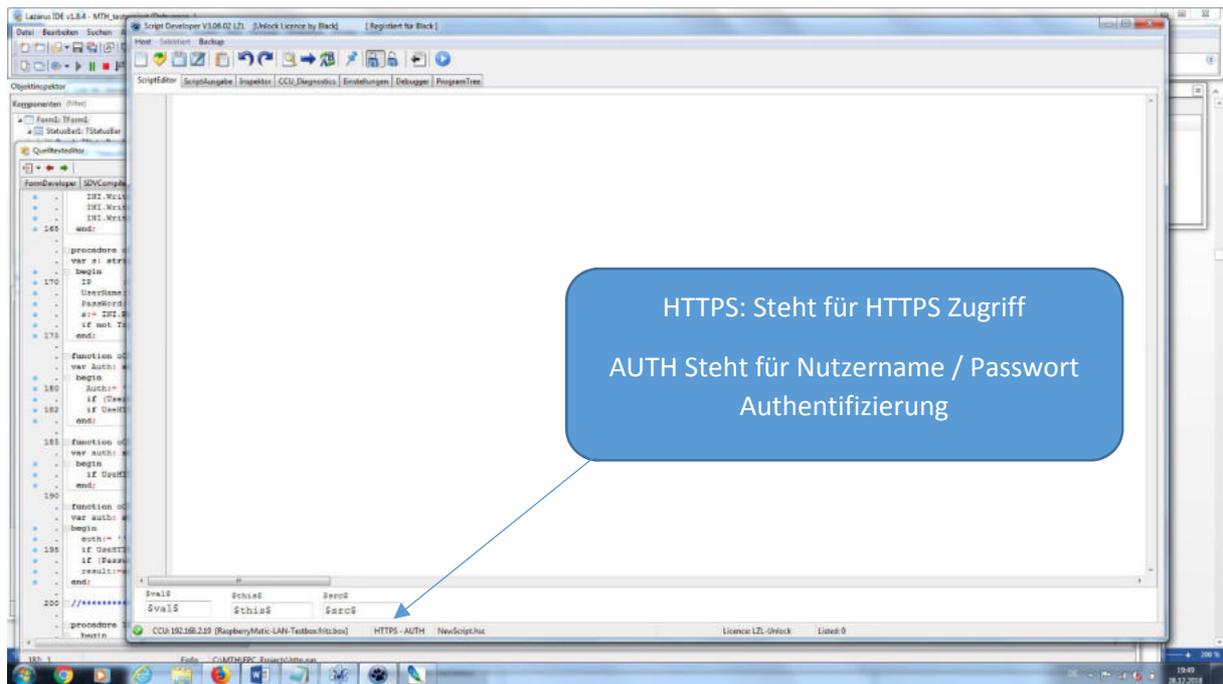
HTTPREGASCRIPT=8181

HTTPREGAXMLRPC=1999

HTTPSREGASCRIPT=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



1.9 Ein Wort des Authors zum Thema „gewerbliche Nutzung“

Aufgrund der Tatsache, dass ich von einem Supplier versehentlich hörte, man freue sich schon auf die Backup Funktionalitäten der Direktverknüpfungen, Blöd nur, dass ich die kommerzielle Nutzung explizit untersagt habe. Immerhin zwingt mich diese kommerzielle Unverfrorenheit ein wenig an dem Ausgabeformat zu feilen. Das derartige Vervielfältigungsmechanismen einen immensen zeitlichen Vorteil bei der Implementierung von baugleichen Anlagen bieten leuchtet allen ein. Möglichkeit wäre diese Funktion dann ganz zu sperren oder per CompilerDirective erst gar nicht einzucompilieren. Hilft aber allen anderen privaten Anwendern nicht.

Also bleibt mir nur um das zu unterbinden bzw. möglichst zu erschweren:

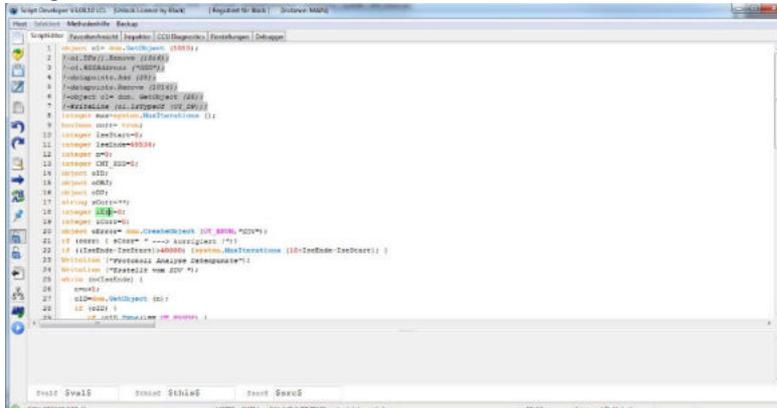
Als Ausgabeformat für das Backup von Geräten wird es nur noch JSON geben.

Damit ist direktes 1 zu 1 Kopien ziehen aus einer Masteranlage zwar möglich, das JSON kann auch manuell angepasst werden, allerdings lassen sich diese JSON Files aber aufgrund des Pairings nicht mehr in die Child Anlagen einspielen. Als privater Anwender sollte man davon nichts merken, der SDV verbindet sich ja mit der CCU und kann das JSON in ein Programm umrechnen und damit das Restore ausführen.

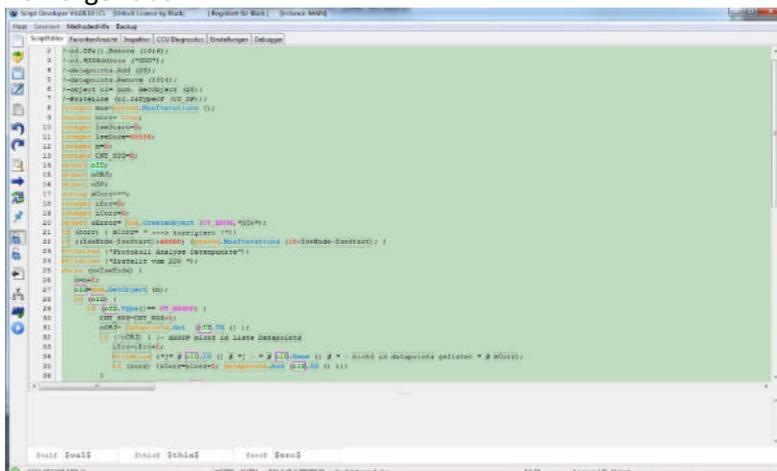
Ich hoffe der Schritt von mir ist nachvollziehbar

2.1.1 SyncroEdit (ab 3.08.10)

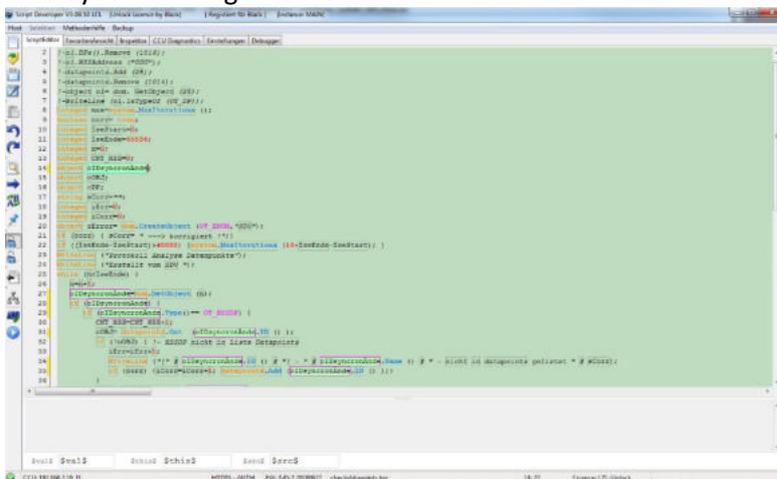
Mit SyncroEdit lässt sich in einem Bereich synchron an mehreren gleichen Stellen ändern.
Vorgehensweise:



Den Textbereich in dem geändert werden soll, markieren, anschliessend mit Strg-J zum Synchroneditieren vorbereiten. Der Bereich wird dann mintgrün, die Worte werden hervorgehoben und ich klick nun dahin wo ich ändern will. Hier Hinter das oid (ist durch den markup auch hellgrün hervorgehoben

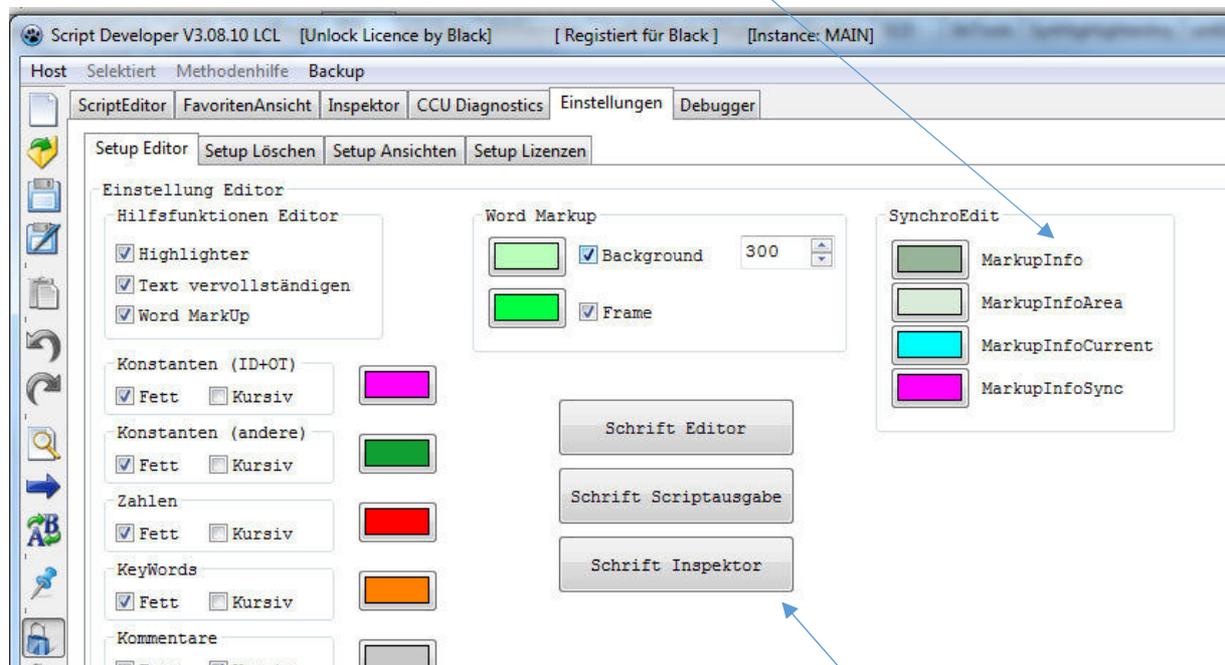


Jetzt kann man schreiben und an allen hellgrün hervorgehobenen Stellen im Mindgrünen bereich wird Synchron mitgeändert.



Die Markierung aufheben geht dann mit Escape.

Die Farben lassen sich natürlich in Setup Menü einstellen

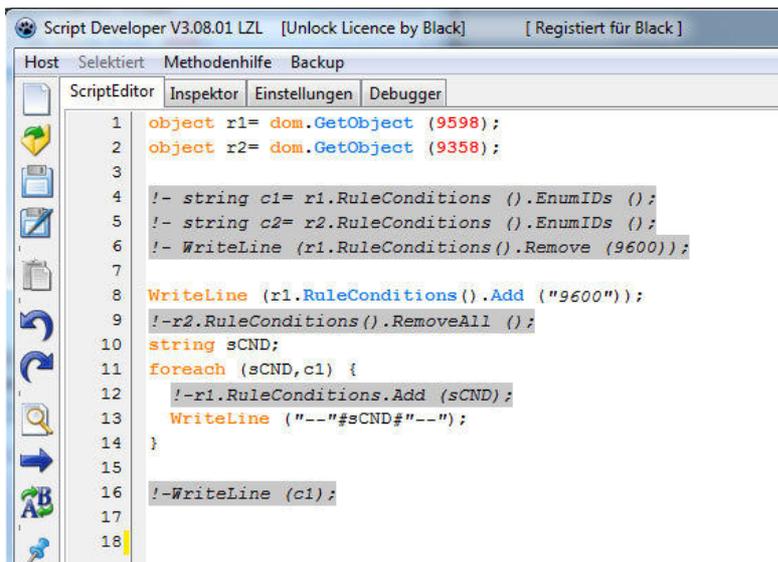
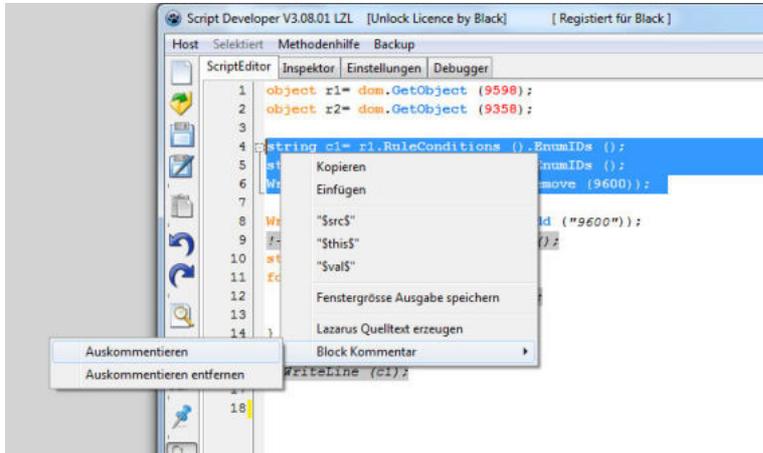


2.1.2 Schriftgrößen und Schriftart (ab 3.08.07)

Einstellungen für Schriftwart Editor, dem Editor Ausgabefenster und im Inspektor

2.2 Blockweise auskommentieren.

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



3 Scripteditor

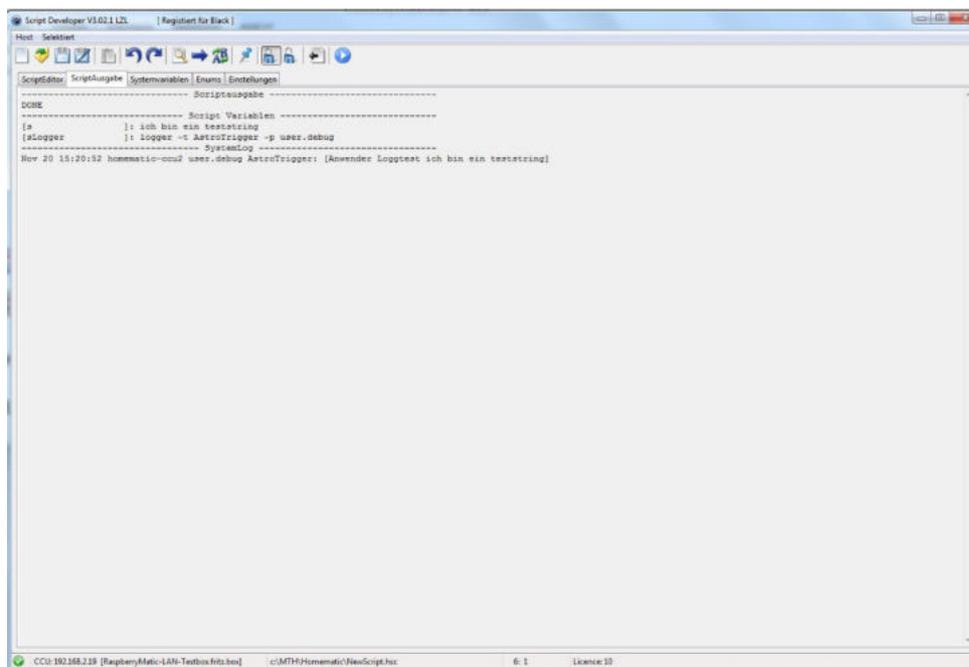
Im Scripteditor werden die Scripte geschrieben oder geladen, die mittels Run Script oder  an die CCU zum Ausführen gesendet werden. Das Scriptergebnis wird dann im Reiter Ausgabe angezeigt. Dieses kleine TestScript zum Beispiel:

```
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:



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 Loggtest " # s # "]);^M WriteLine ("DONE");^M
[iseESP.cpp:1121]
Jun 15 12:41:49 homematic-raspi local0.err ReGaHss: Error: ParseProgram: SyntaxError=
(sLogger # "[Anwender Loggtest " # s # "]);^M WriteLine ("DONE"); [iseESP.cpp:374]
```

Ab 3.07.04 : Script testen eingefügt, hierbei wird das Skript nur nach Fehlern geparkt 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

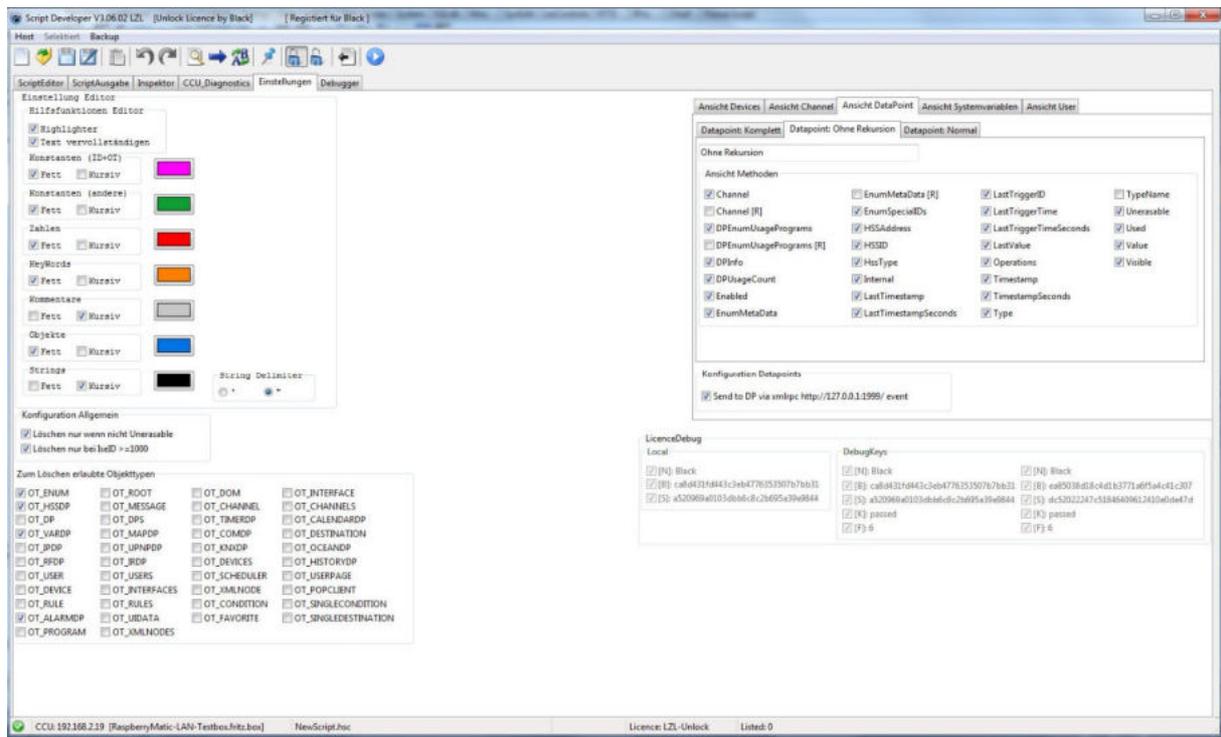
Trat ein Fehler auf, so erfolgt Sprung auf den Reiter Scriptausgabe mit Anzeige der Fehlermeldung

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



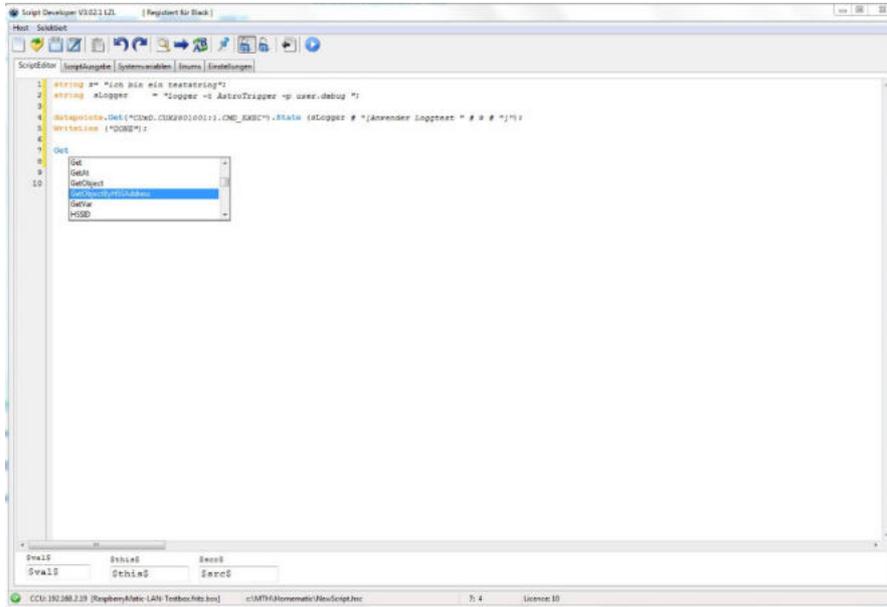
Hier kann nach Vorliebe der Highlighter konfiguriert oder auch ausgeschaltet werden

Ebenso lässt sich die Vervollständigen Funktion an oder abwählen.

Vorbedingung natürlich: Lizenzlevel muss vorhanden sein.

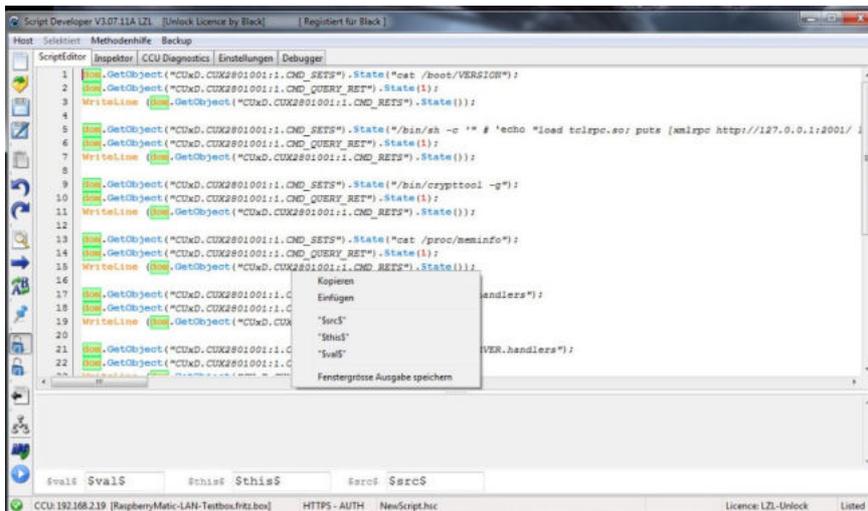
3.2 Vervollständigen Funktion

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



Ab V3.08.10

Erweiterung der Vervollständigen Funktion

Folgende Vervollständiger sind nun parallel implementiert:

Ctrl-Space : Die bekannte komplette Version

Ctrl-Shift-M: Nur Regascript Methoden

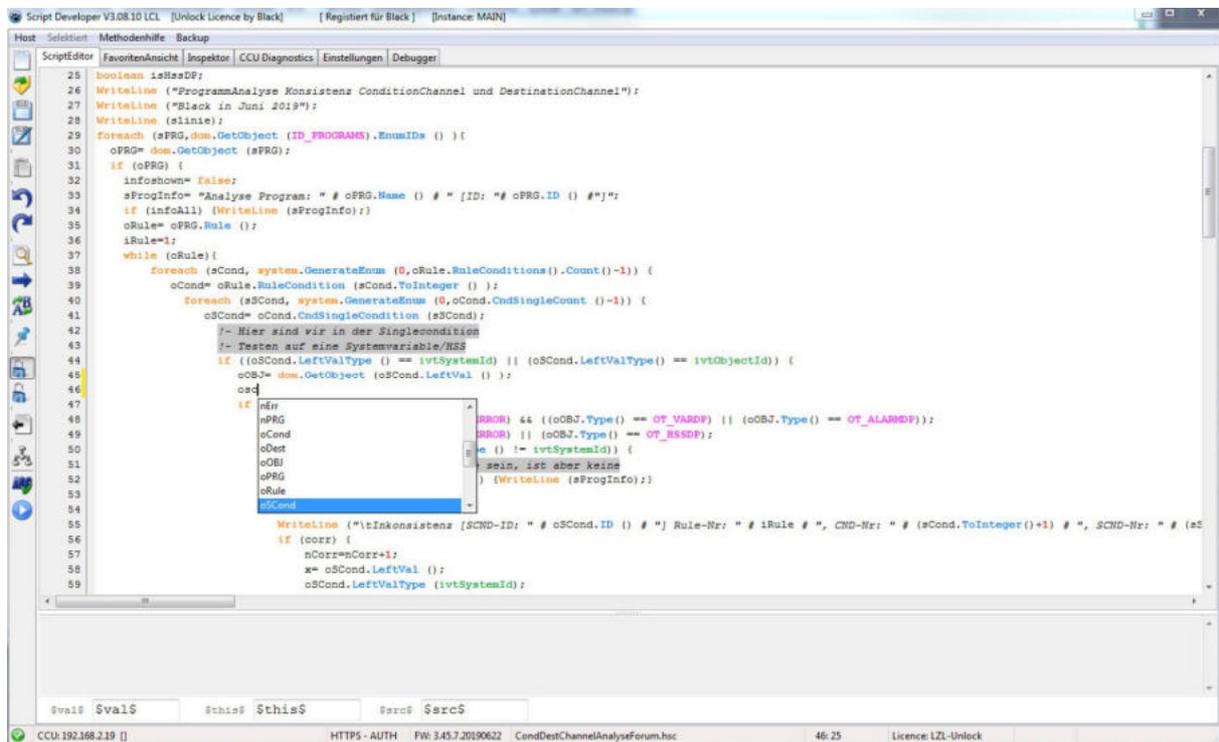
Ctrl-Shift-I: Regascript Konstanten ID_XXX

Ctrl-Shift-O: Regascript Konstanten OT_XX

Ctrl-Shift-S: Untermenge Regascript Methoden (StringMethoden)

Ctrl-Shift-C: Untermenge Regascript Methoden (Mathematische Methoden)

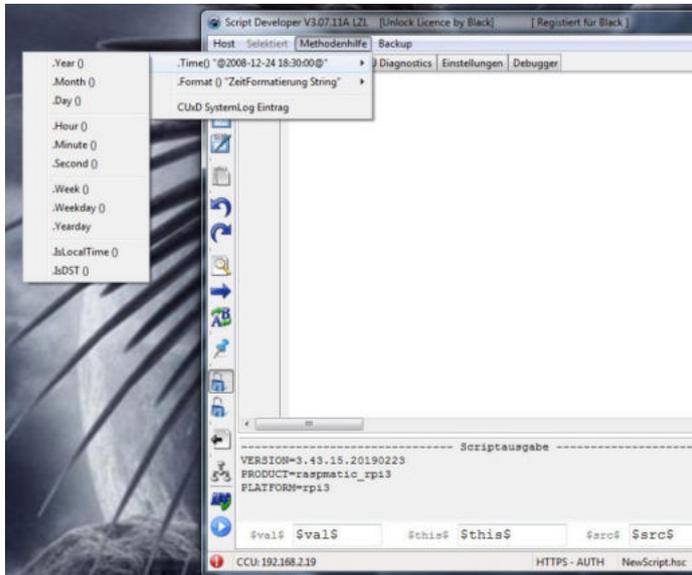
Ctrl-Shift-D: Deklarationen: hier wird in Echtzeit das Script analysiert und die gefunden Variablendeklarationen vorgeschlagen



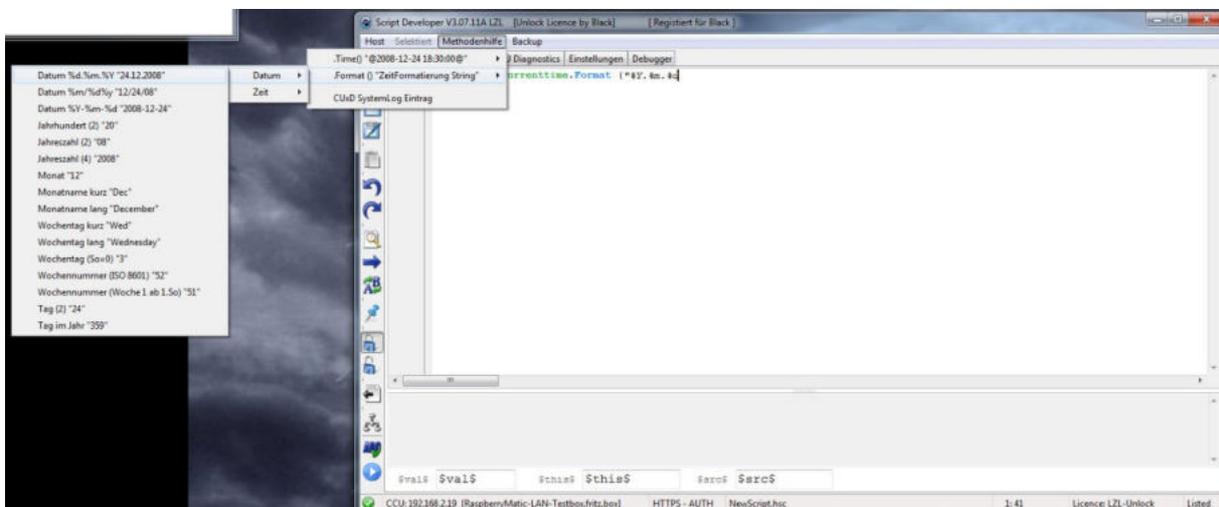
3.3 Methodenhilfe

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



Ab dem Version 3.08.12D verfügt der SDV über die Möglichkeit, bis zu zehn eigene Methodenhilfen zu definieren. Diese werden in der INI angelegt.

```
; Common escape sequences
; \\      \ (a single backslash, escaping the escape character)
; \'      Apostrophe
; \"      Double quotes
; \t      Tab character
; \r      Carriage return
; \n      Line feed
; \;      Semicolon
; \#      Number sign
; \=      Equals sign
; \:      Colon
```

```
[METHODHELP1]
NAME=dom.GetObject
TEXT=dom.GetObject (
```

```
[METHODHELP2]
NAME=
TEXT=
```

```
[METHODHELP3]
NAME=
TEXT=
```

```
[METHODHELP4]
NAME=
TEXT=
```

```
[METHODHELP5]
NAME=
TEXT=
```

```
[METHODHELP6]
NAME=
TEXT=
```

```
[METHODHELP7]
NAME=
TEXT=
```

```
[METHODHELP8]
NAME=
TEXT=
```

```
[METHODHELP9]
NAME=
TEXT=
```

```
[METHODHELP10]
NAME=
TEXT=
```

Sollte fast selbsterklärend sein, mit Name wird definiert, was in der Menüzeile dargestellt wird, mit text wird das definiert, was dann im Editor eingefügt wird. Es muss mindestens ein Eigener MenüHelp definiert sein, damit der menüpunkt unter Methodenhilfe dargestellt wird.

Ebenso ist direkte Verwendung von Kurztasten möglich Shift+Ctrl + 1,2,3...9,0

dabei bitte auf die Falle aufpassen !

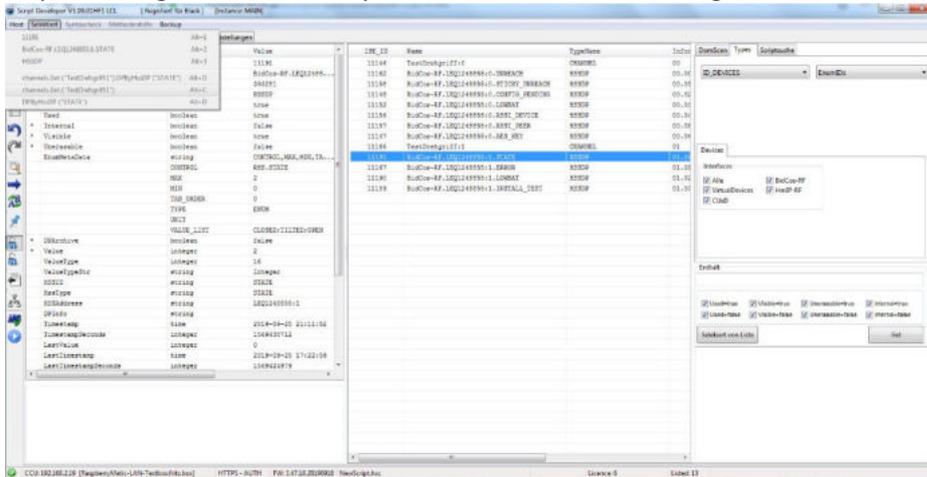
ich hab extra die Kommentare da mit reingesetzt, sie dort aufgeführten Sonderzeichen müssen gequotation sein rechts des Gleichheitszeichens.

heißt will ich dom.GetObject (" "); als Text schreiben so muss ich eintragen
TEXT=dom.GetObject (\ " \ " \);

Zweizeilig geht auch, dann muss \n benutzt werden.

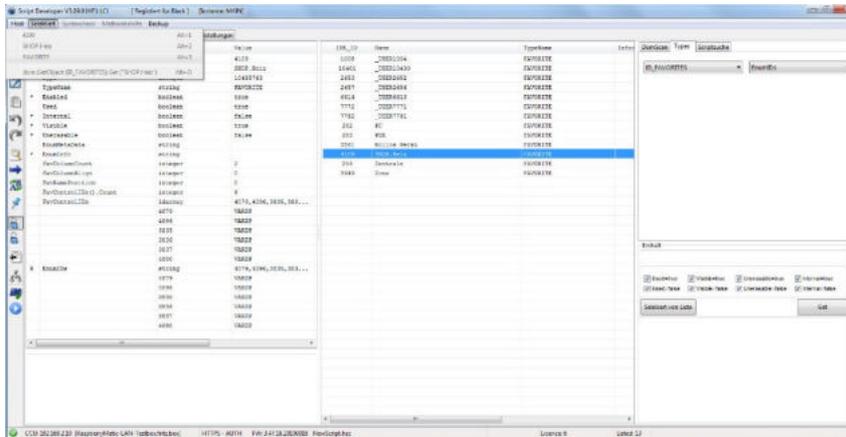
3.4 Symbolischer Zugriff auf selektierte Datenpunkte des Inspektors

Zur Vereinfachung beim Programmieren besteht die Möglichkeit, einen Datenpunkt, welcher im Inspektor angewählt wurde, symbolisch in den Editor einfügen zu können.



Bei einem HSSDP werden 3 Möglichkeiten angeboten (Menüpunkt selektiert)
 Vollzugriff über Kanal.HSSDP
 Nur Kanal
 Nur HSSDP

Fast alle typen (Favoriten, User, VarDP, AlarmDP) werden automatisch richtig erkannt und dafür der richtige Symbolische Zugriff vorgeschlagen.

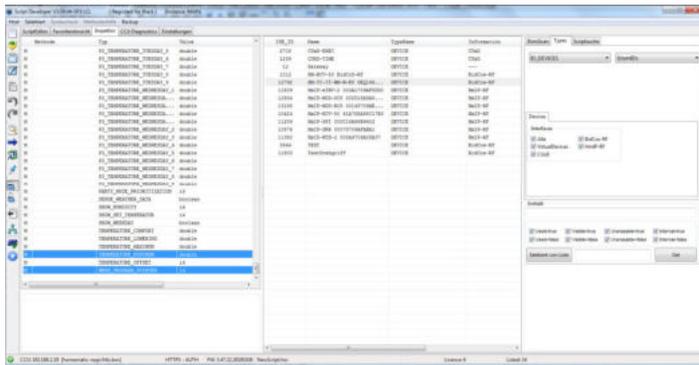


Eine einfache, schnelle und effektive Methode um unnötige Schreibfehler zu vermeiden.

3.5 Automatische Codeerzeugung für Read/Write von Master/Linksets

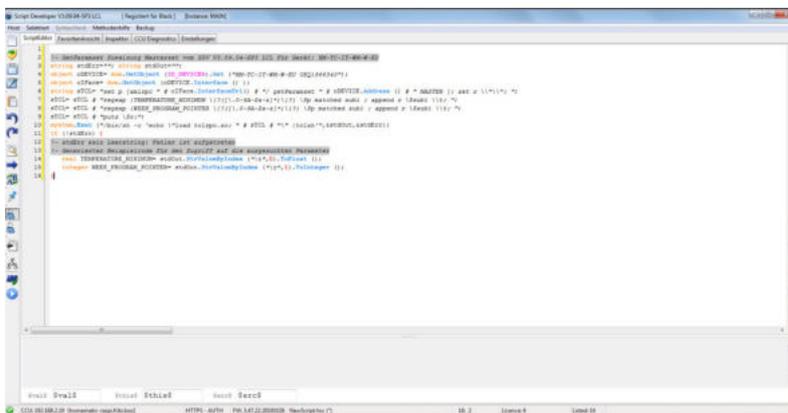
Im Editor kann automatisch Code erzeugt werden für das Schreiben von Master/Linksets (putParamset) oder für das Lesen von Master/Linksets (getParamset).

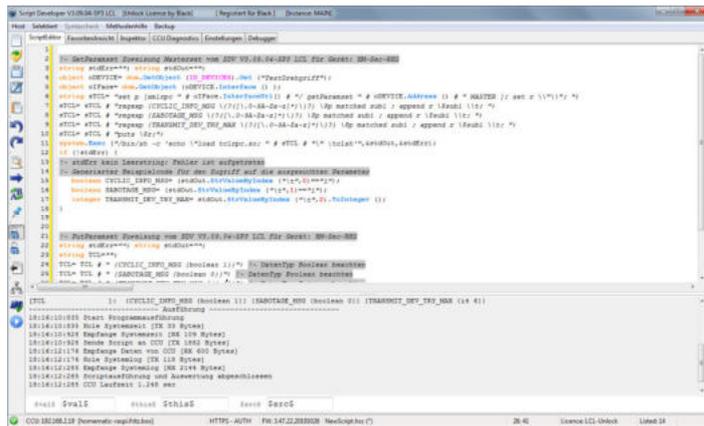
Dazu müssen im Vorfeld im Inspektor die gewünschten Masterparameter (M) oder Linkparameter (L) selektiert werden. Dann hat man im Editor auf der rechten Maustaste im Menü die Auswahl Xmlrpc.PutParamset (Code zum Beschreiben dieser Parameter generieren) bzw xmlrpc.GetParamSet (Code zum Auslesen dieser Parameter generieren)



Dann wechseln in den Editor. Dort wo der Abfragetext in das Script eingefügt werden soll: rechte Maustaste - xmlrpc.getParamset

dann erzeugt der SDV den richtigen Code für die Abfrage (Device, Channel wird richtig ausgewählt, die Regex wird richtig definiert und aus dem Ergebnis werden 2 Scriptvariablen mit dem Namen des Mastersetparameters mit dem richtigen VariablenTyp angelegt.

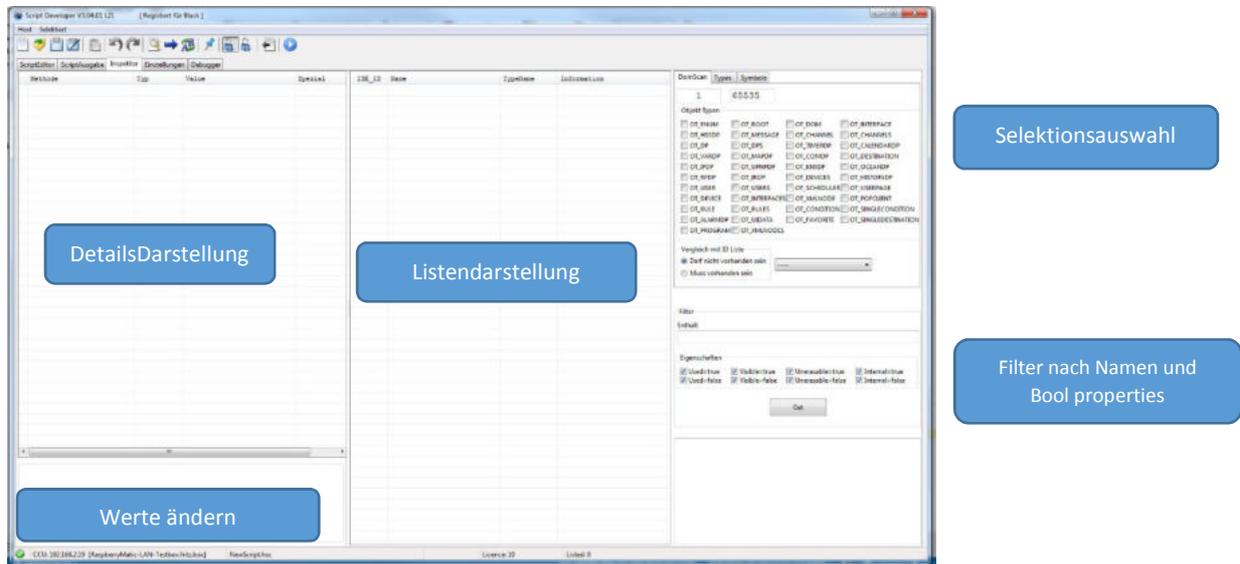




4 Inspektor

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 Eingegabener 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

Objekt Typen

OT_ENUM OT_ROOT OT_DOM OT_INTERFACE

OT_HSSDP OT_MESSAGE OT_CHANNEL OT_CHANNELS

OT_DP OT_DPS OT_TIMERDP OT_CALENDARDP

OT_VARDP OT_MAPDP OT_COMDP OT_DESTINATION

OT_IPDP OT_UPNPD OT_KNXDP OT_OCEANDP

OT_RFDP OT_IRDP OT_DEVICES OT_HISTORYDP

OT_USER OT_USERS OT_SCHEDULER OT_USERPAGE

OT_DEVICE OT_INTERFACES OT_XMLNODE OT_POPCLIENT

OT_RULE OT_RULES OT_CONDITION OT_SINGLECONDITION

OT_ALARMDP OT_UIDATA OT_FAVORITE OT_SINGLEDESTINATION

OT_PROGRAM OT_XMLNODES

Vergleich mit ID Liste

Darf nicht vorhanden sein

Muss vorhanden sein

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

ISE_ID	Name	Typename	Subsystem
1	Domscan	DEVICE	
1012	BB-WT-10-BUSCH-42	DEVICE	
1018	CHD-TIME	DEVICE	
1444	SI_BESTELLA_TIER	DEVICE	
1718	OTAD-IBEC	DEVICE	
1809	WALDORF	CALLERGROUP	
1864	BB-SO-100-10-10011	DEVICE	
1880	BB-SO-100-10-10011	DEVICE	
1881	BB-SO-100-10-10011	DEVICE	
1876	BB-SO-100-10-10011	CALLERGROUP	
1818	BB-SO-100-10-10011	DEVICE	

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

4.2 Selektionskriterium Types

The screenshot shows a dialog box titled "DomScan" with a sub-tab "Types". It contains two sections: "Aufzählung" and "Quelle".

Aufzählung

<input checked="" type="radio"/> ID_ROOMS	<input type="radio"/> ID_FUNCTIONS	<input type="radio"/> ID_FAVORITES
<input type="radio"/> ID_INTERFACES	<input type="radio"/> ID_SYSTEM_VARIABLES	<input type="radio"/> ID_DEVICES
<input type="radio"/> ID_CHANNELS	<input type="radio"/> ID_DATAPOINTS	<input type="radio"/> ID_USERS
<input type="radio"/> ID_PROGRAMS		

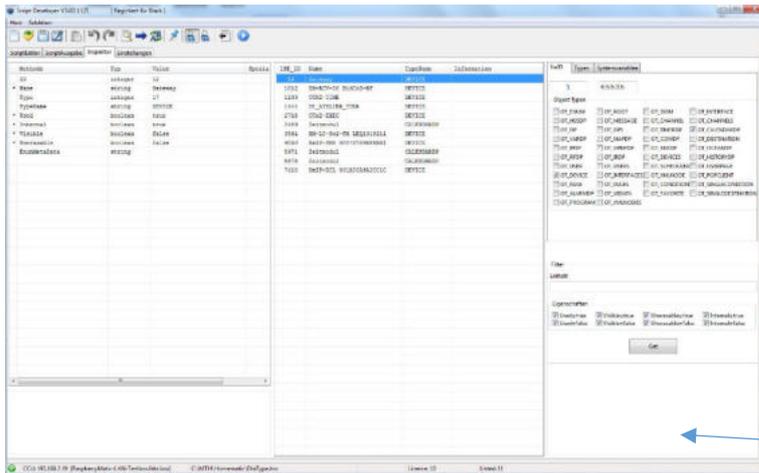
Quelle

<input checked="" type="radio"/> EnumIDs	<input type="radio"/> EnumUsedIDs
--	-----------------------------------

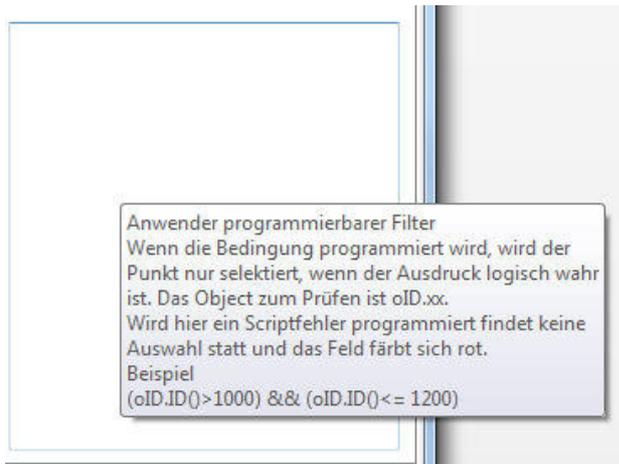
Hierbei wird wie schon in der Version 3.2 in festen Bereichen gesucht und selektiert. Schneller und einfacher zu handeln als die Objekt Selektion, dafür nicht so umfangreich.

4.3 Zusätzliche Selektionsbedingungen

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

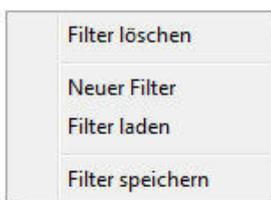


Zusätzliches Feld für Selektionsbedingungen



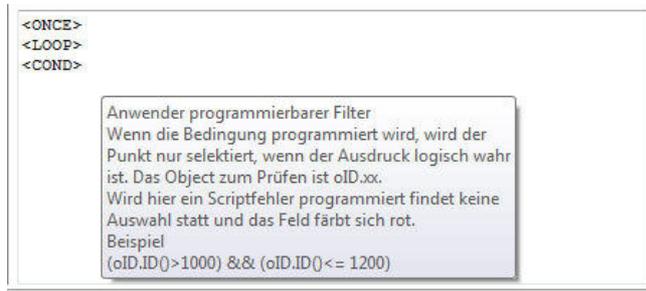
Filter sind ein mächtiges Werkzeug zum komplexen Eingrenzen und für Komplexe Abfragen.

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:

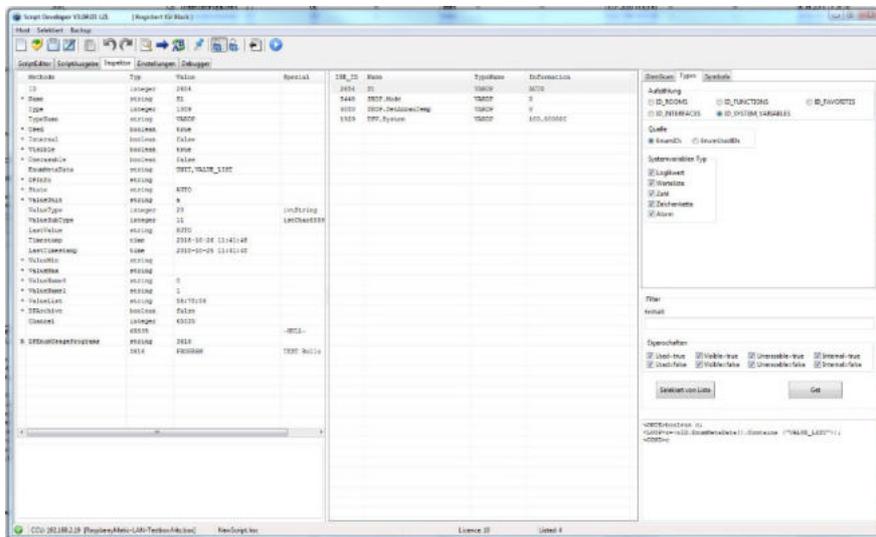
```
<ONCE>boolean c;
<LOOP>c=(oID.EnumMetaData().Contains ("VALUE_LIST"));
<COND>c
```

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 Variablenamen sind schon Intern vorbelegt:

- object oID: darf benutzt werden, ist der Bezug auf das Objekt, welches im Filter überprüft werden soll
- var v: intern benutzt zur Typerkennung: Fingers weg
- string sInfo: intern benutzt zur Listengenerierung: Fingers weg
- boolean b: interner Filter, auch Finger weg
- 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 programmiert
Wenn die Bedingung
Punkt nur selektiert
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

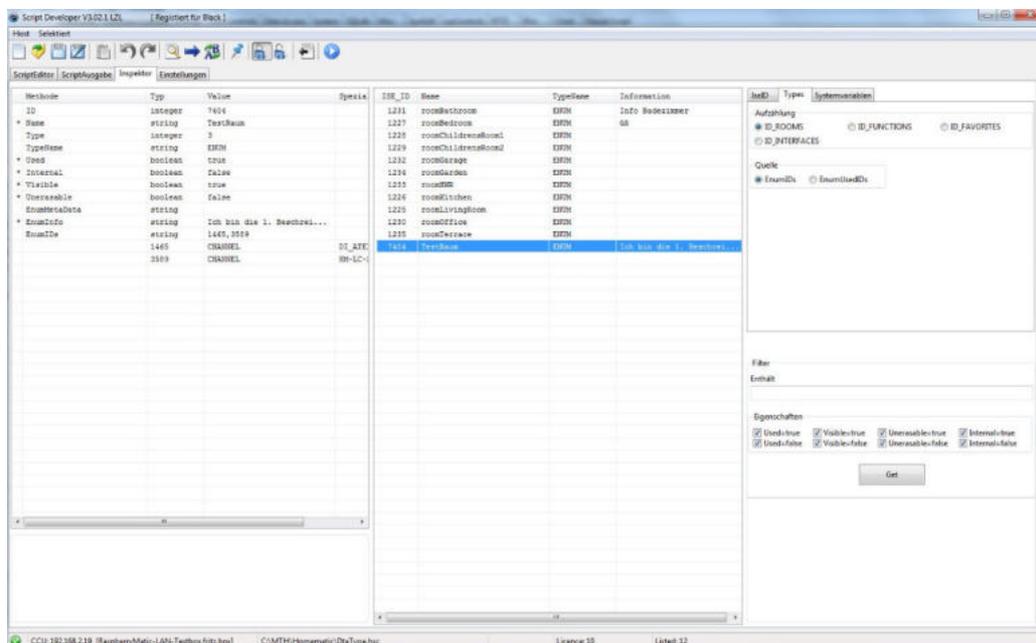
SYSVAR_VerwaisterChannel – Filter um Systemvariablen zu finden, deren Channel verweis in Nirvana zeigt

Durch Click auf die Beschreibungszeile IseID 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 Sortialgorithmen suboptimal arbeiteten, hat das ListView Object neue selektive Sortialgorithmen bekommen. IseID sortiert nun wie man erwartet nach Integer aufsteigend, Name sortiert alphabetisch aufsteigend, TypeName sortiert alphabetisch, sind die Typenamen gleich, wird innerhalb gleicher Typenamen nach IseID numerisch sortiert. (Ab Version 3.08.08 verbesserter Sortialgorithmus, hier wird durch Klick auf die Spalte zwischen Sortieren aufsteigend und Sortieren absteigend getoggelt.)



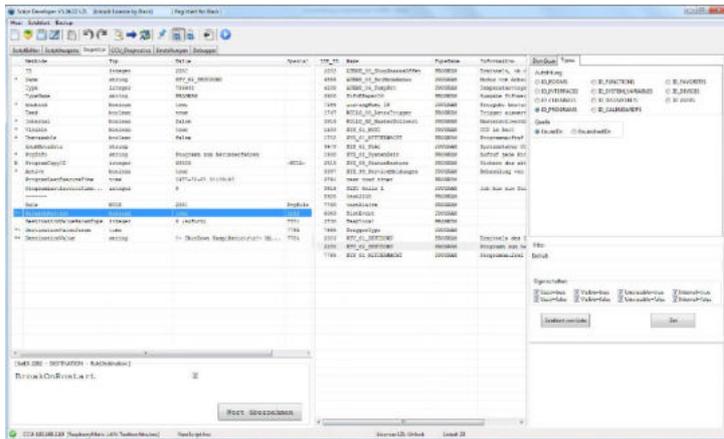
Die Spaltenbreite kann sowohl in Normalansicht als auch in Maximize separat eingestellt werden (Das Programm sollte sich die Breiten merken und je nach Darstellungsart automatisch wieder einstellen, sollte...)

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)

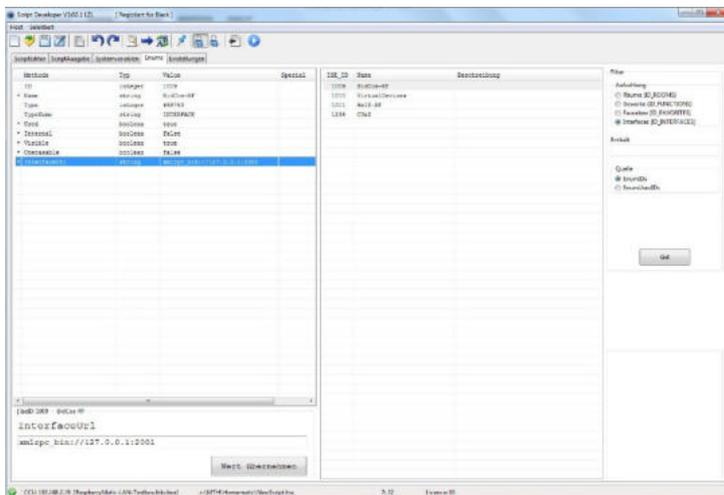
Properties, die in der ersten Zeile mit einem Stern (*) gekennzeichnet sind, können in ihrem Wert geändert werden.

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



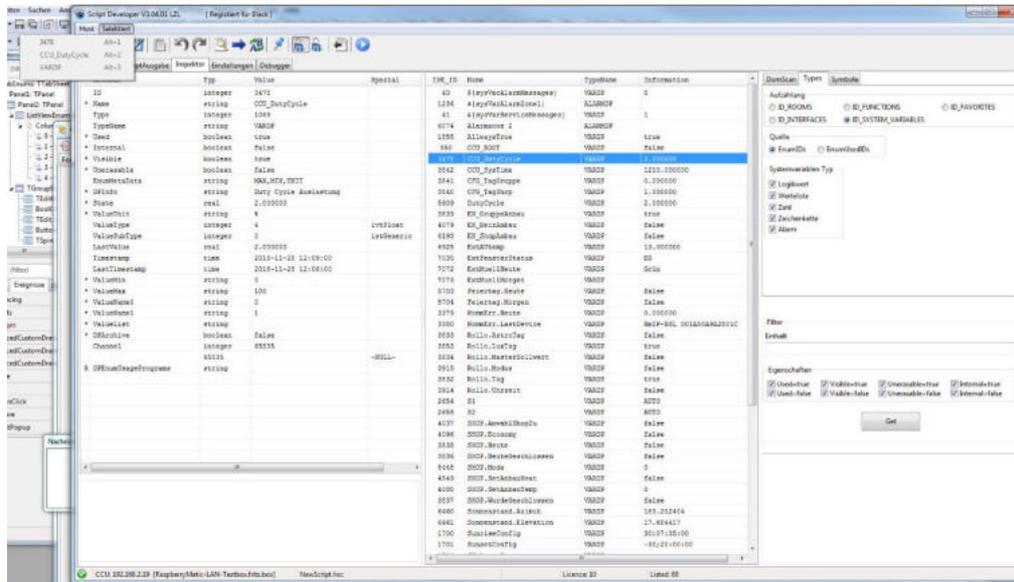
Nach Click auf Wert übernehmen wird der Wert in der CCU geändert. Also Vorsichtig mit dieser Funktion umgehen, hier gibt es kein redo.

4.4 Daten aus Inspektor in Editor übernehmen

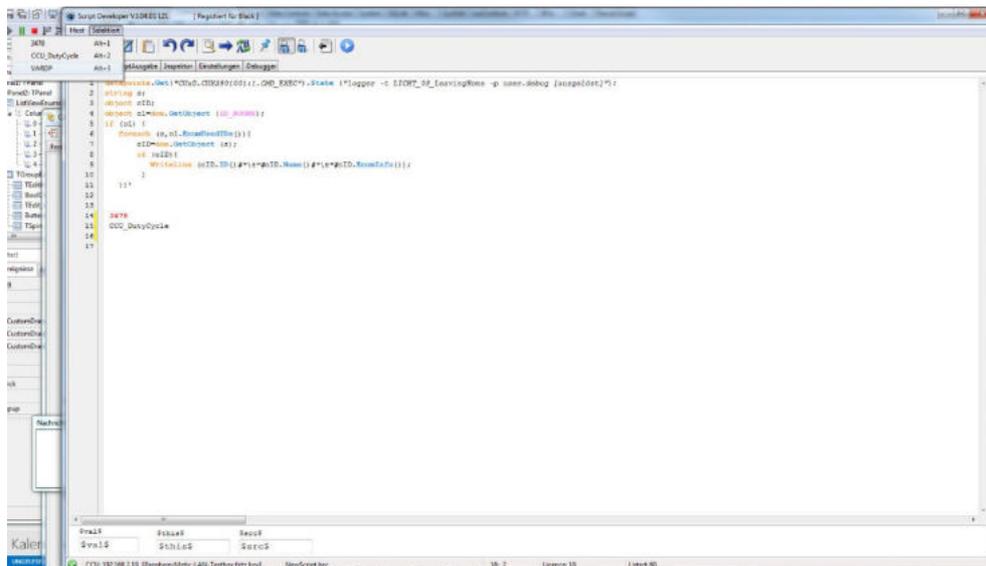
Zur schnelleren und auch möglichst fehlerfreien Bearbeitung besteht die Möglichkeit, Daten aus dem Inspektor direkt in den Editor zu übernehmen.

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

Hier Klick auf die Systemvariable



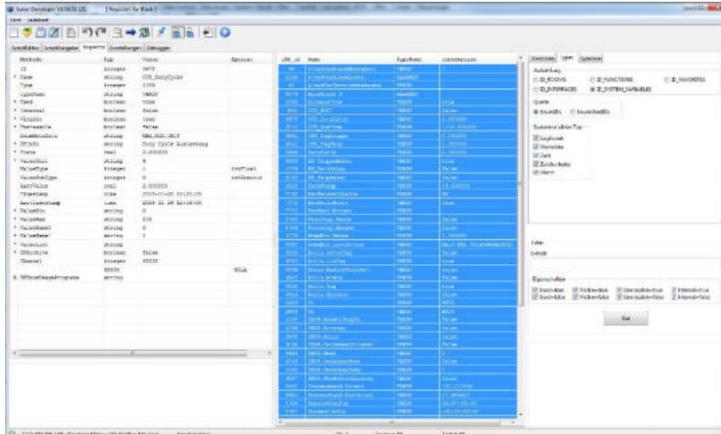
Unter selektiert sind die Eigenschaften herausgefiltert worden und lassen sich im Editor entweder durch das Menü selektiert oder durch die Kurztasten Alt-1: ID, Akt-2: Name und Alt 3: Eigenschaft bzw. Methode einfügen.



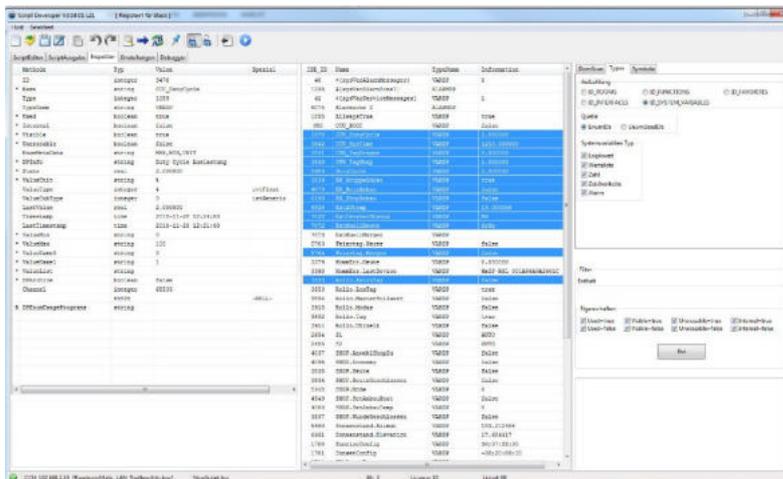
4.4.1 Mehrfachauswahl als Enum String

Es lassen sich im Hauptauswahlfeld Mehrfachselektionen vornehmen.

Ctrl-A : alle auswählen.



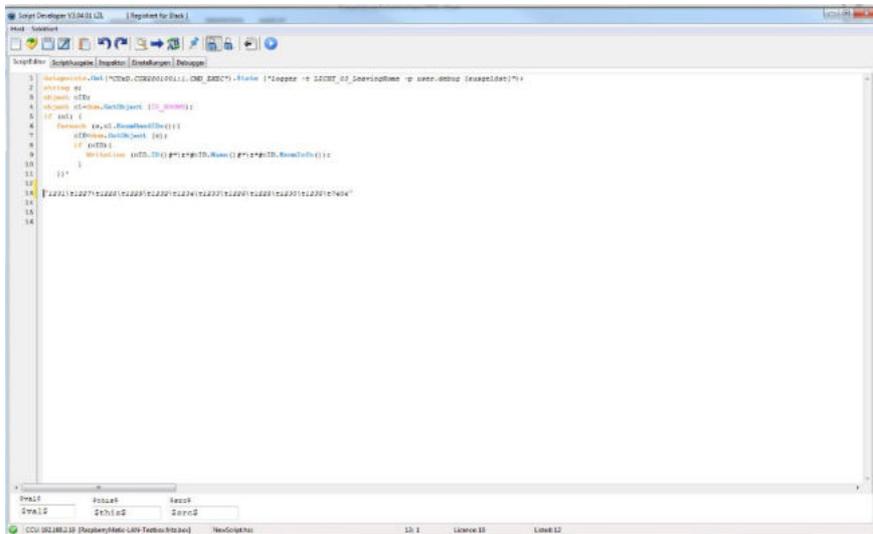
Oder die Übliche Mausbedienung:



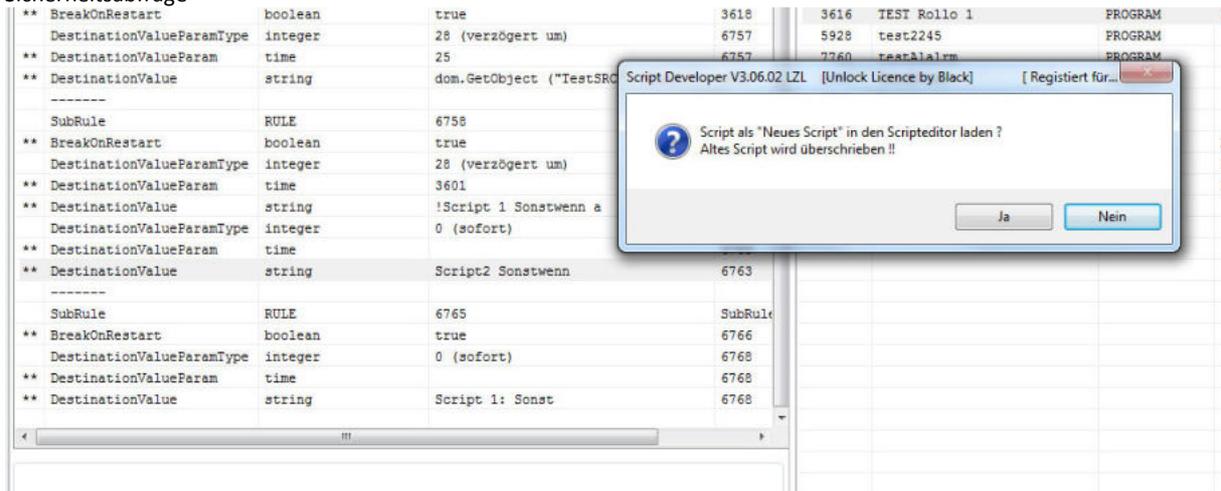
Ab Version 3.07.07.

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

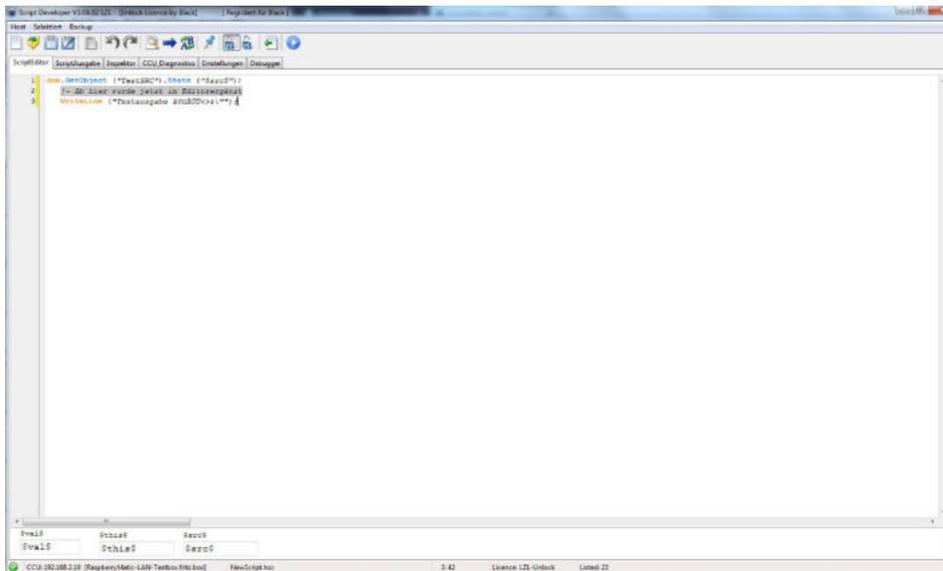
Mit der  Taste merkt sich der Inspektor die Auswahl, welche sich dann Script Konform im Editor Als ID-Enum durch die  Einfügen Clipboard Taste, welche nach dem Pin Druck nicht mehr grau ist, lassen sie die selektierten ID,s im Editor einfügen (z.B. zur Verarbeitung in einem Script als foreach)



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



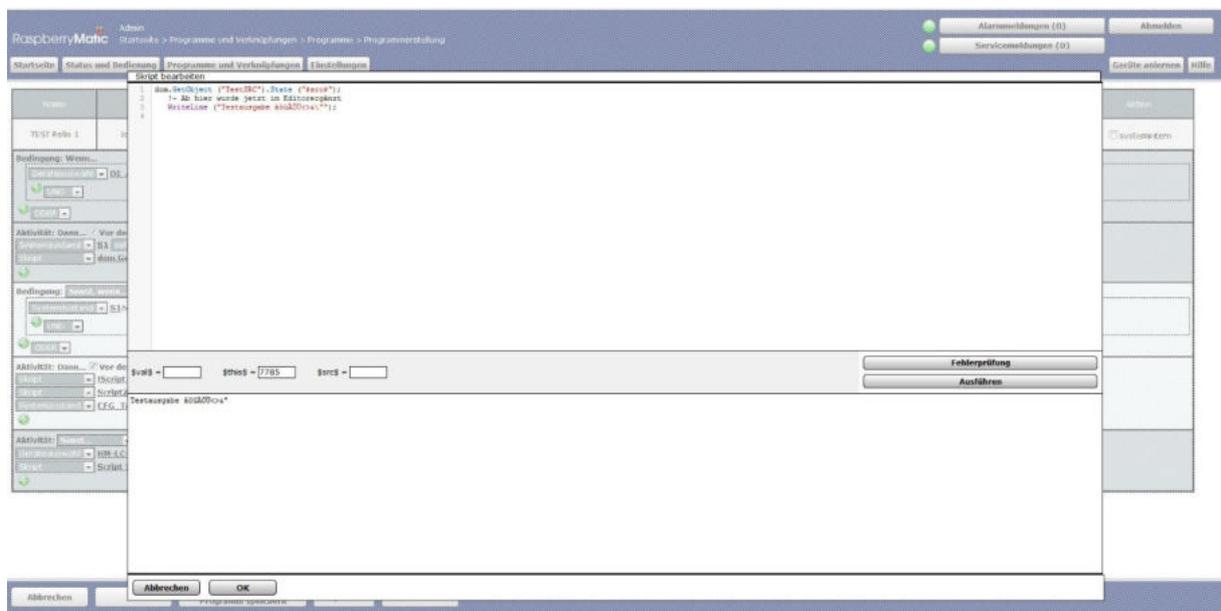
Das Programm aus dem Editor direkt in das Programm der CCU geladen werden. Im Hint wird als Gedächtnisstütze der Name des Programmes mit angezeigt.

Neu ab 3.10.01A: Der SingleDestinationbezug wird gemerkt. Es kann also der Inspektor ganz normal weiter benutzt werden. Hostwechsel, neues Script oder Script laden löscht diesen gemerkten Bezug. Aus Sicherheitsgründen wird vor dem Rücksichern in das Rega-Objekt nochmal ein Konsistenztest gemacht, ob es sich wirklich um das Quellobjekt handelt. Wenn nicht erfolgt ein Hinweis:



Der Editortext befindet sich dann in der Windows-Zwischenablage und muss manuell über die WebUI in das Programm geladen werden, oder aber das Skript muss nochmal aus dem Inspektor extrahiert werden und durch den Inhalt der Zwischenablage ersetzt werden.

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

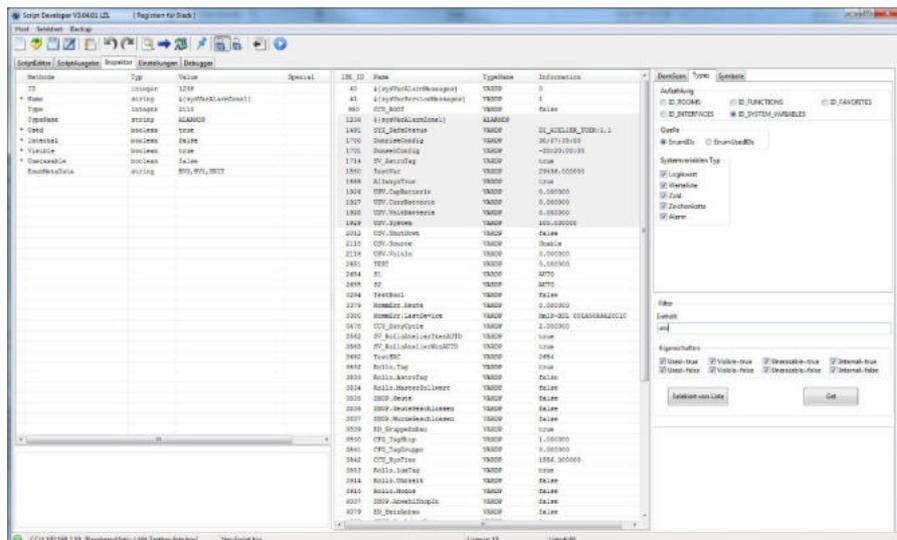


Das Programm würde nun das neue, geänderte Script bei Triggerung ausführen.

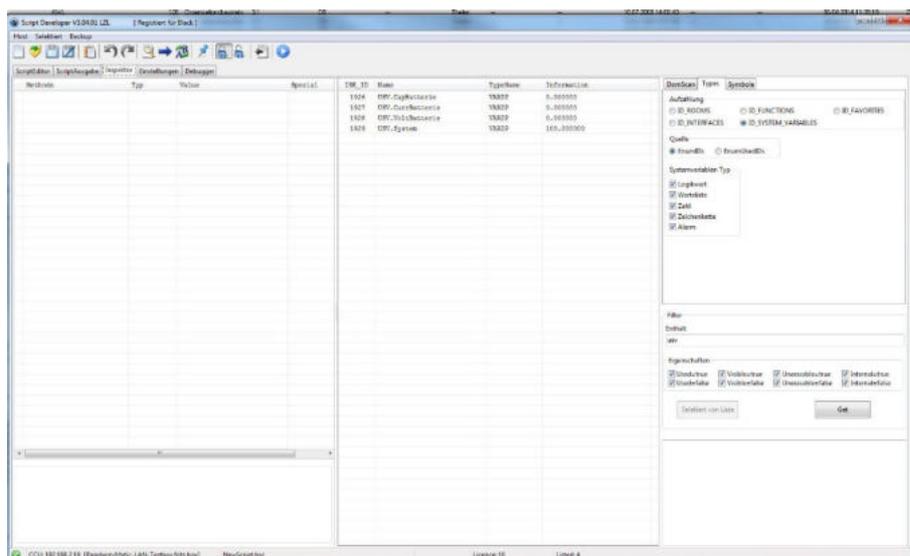
4.5 Selektion von Selektion

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

Hier Beispiel



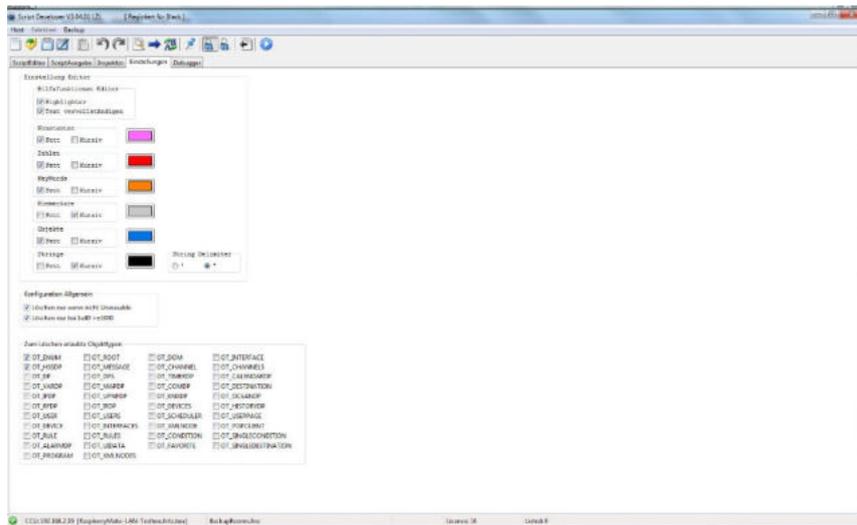
Selektierter Bereich von Systemvariablen, die hier darauf gefiltert werden sollen, dass der Name den String „usr“ enthält. Es müssen 4 Sysvars gefunden werden, die IDS 1012,2115,2116 werden hier nicht berücksichtigt, da diese sind selektiert sind. Bei <Druck auf: Selektiert von Liste: ergibt sich dann



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 löschbare Elemente zu löschen entweder:

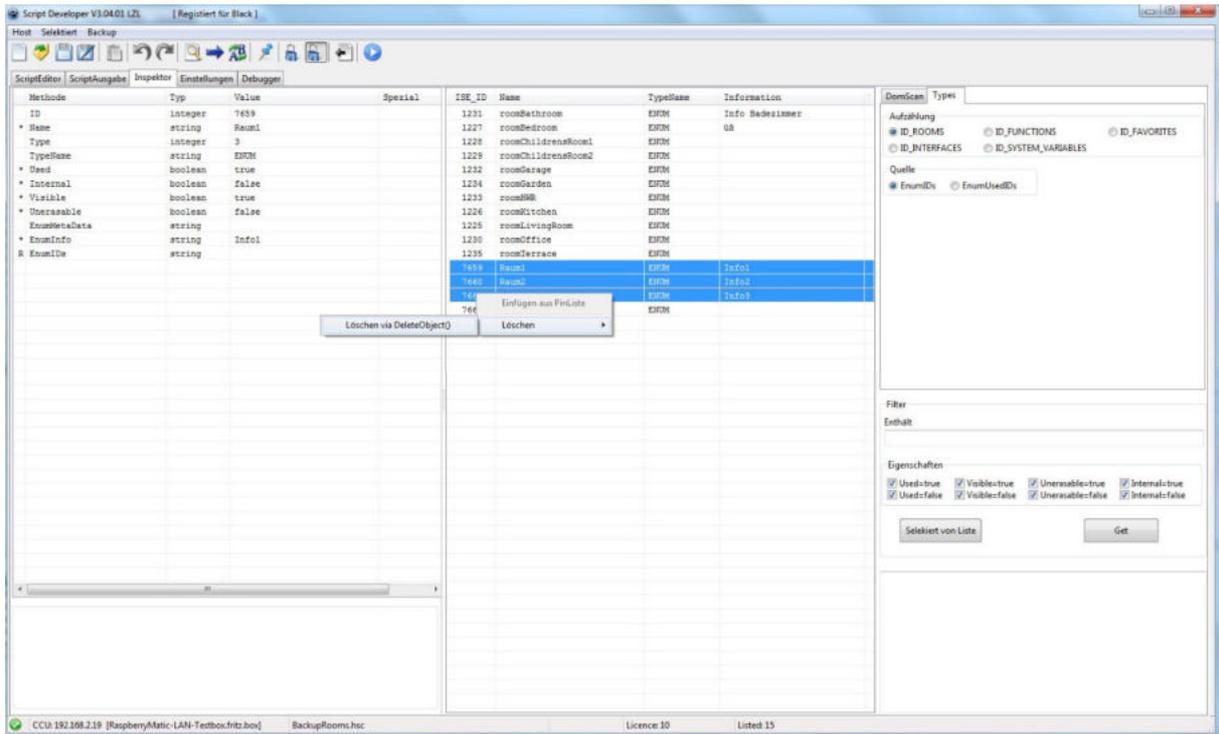
Im Inspektor unter Detailsview die Property entfernen (gilt nur für das Objekt), oder hier den Haken wegmachen (gilt für alle)

Löschen nur wenn ID >= 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.

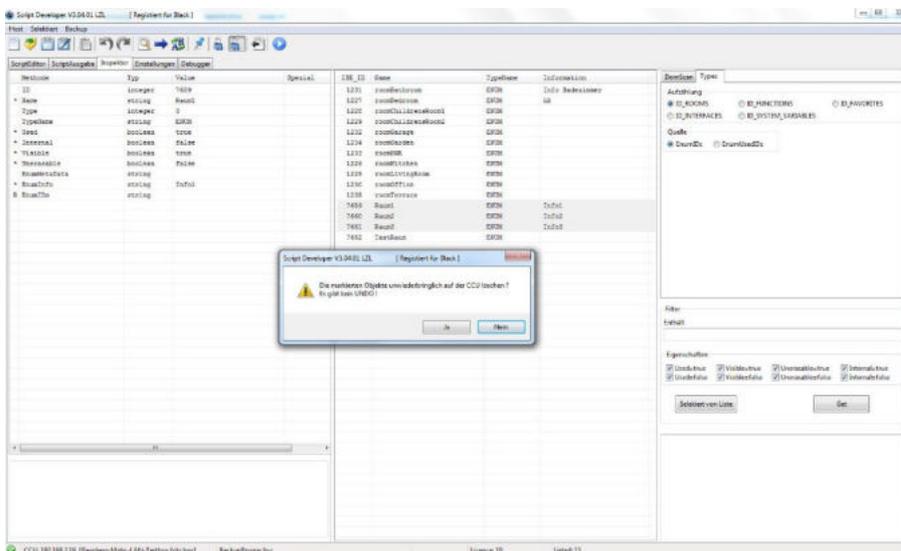
Löschbare Objekttypen. Die Letzte Sicherheit: ein zu löschendes Objekt muss einen hier gechecked Objekttyp haben, sonst wird es nicht gelöscht.

Löschen läuft so ab:



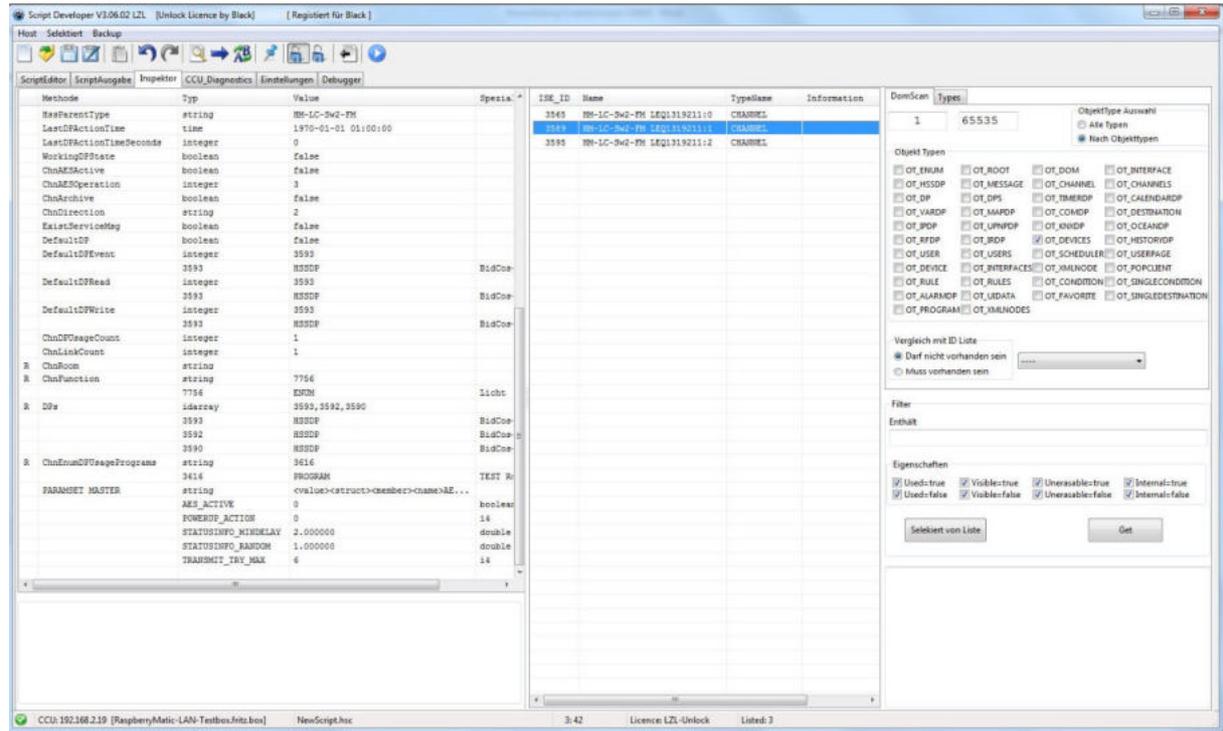
Objekte filtern und markieren, rechte Maustaste, Löschen, Löschen via DeleteObject ()

Mehrfachselektion ist möglich

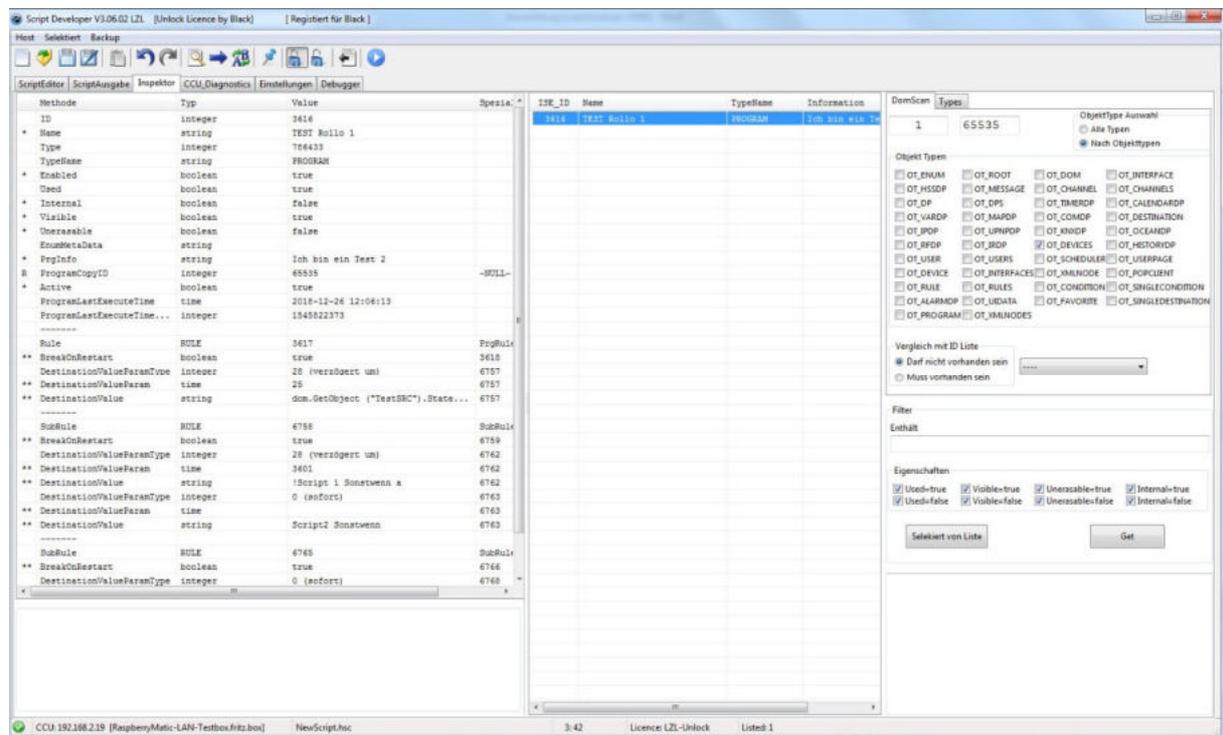


Nach dieser Sicherheitsabfrage sind die Objekte dann weg.. Zurück geht's dann nur mit Restore. Bei einem Device Objekt gibt es noch die zusätzliche Möglichkeit Löschen über xmlrpc.deleteDevices ().

Und rekursiv weiter über die Channels des Devices

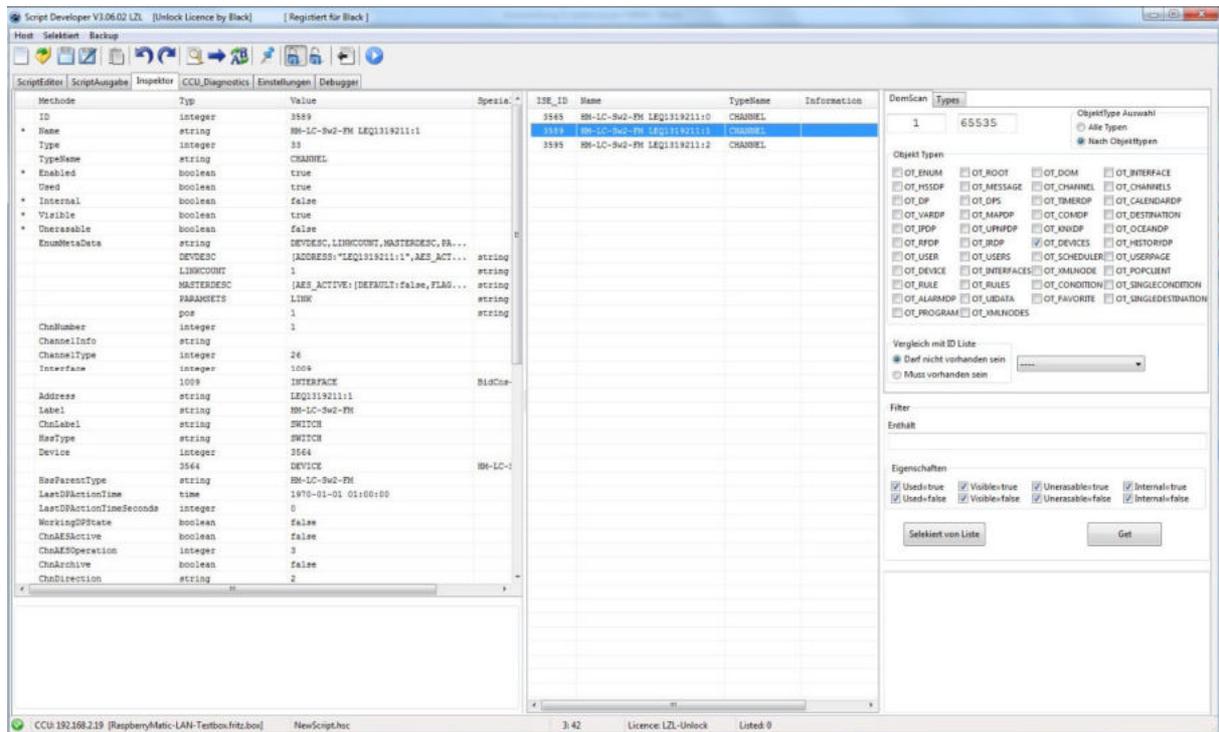


Mal in die Verwendung in dem Programm schauen über ChnEnumDPUsagePrograms



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



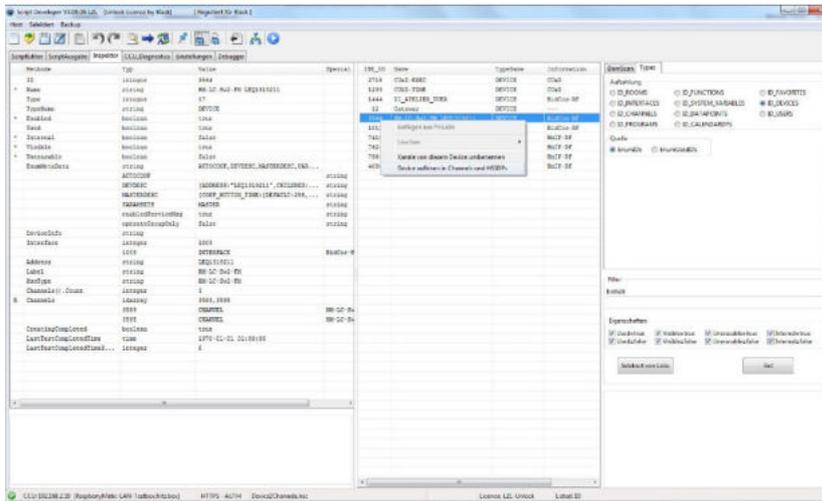
Ab Version 3.06.06 merkt sich der SDV zusätzlich zum Inhalt des Selektionsfeldes auch noch das zuletzt angezeigte Objekt in der Details view und stellt diese Ansicht auch wieder her (So das Objekt noch existent ist)

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

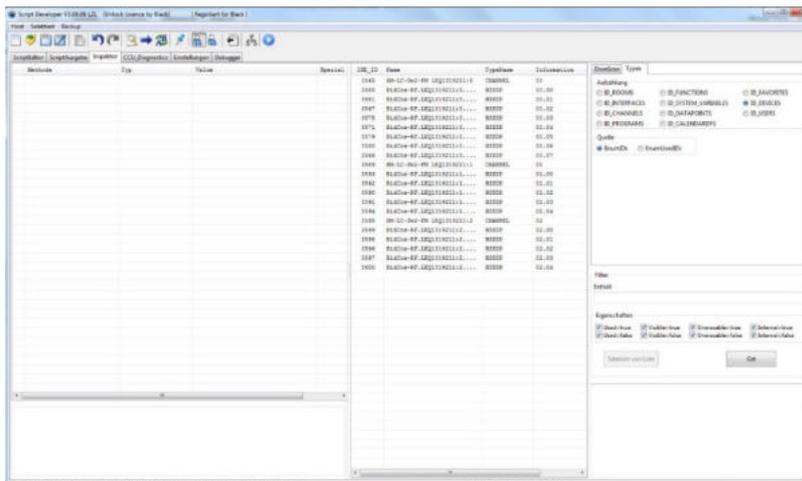
Ein Device besteht ja aus Channels und die Channels wiederum aus Datenpunkten



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 HSDPs.

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

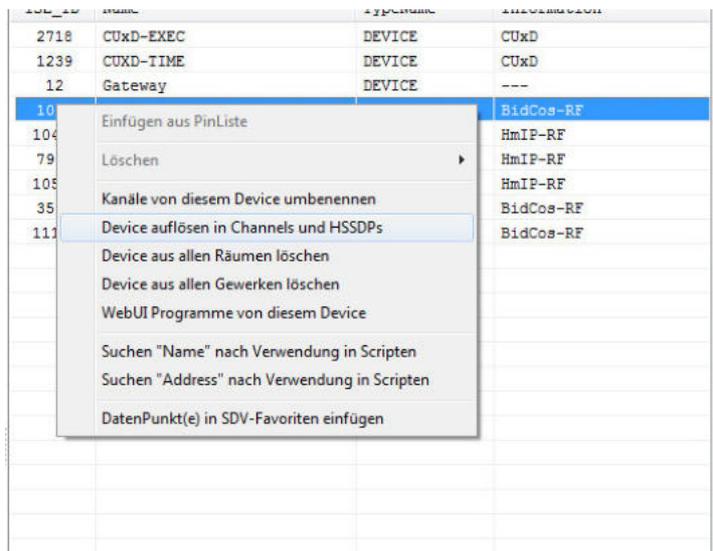
Als Ergebnis erhält man :



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

Wenn ein Device selektiert wurde, stehen mit Klick rechte Maustaste einige Bearbeitungsmenüs zur Verfügung

OBJ_ID	Name	AppTyp	ANWENDUNG
2718	CUxD-EXEC	DEVICE	CUxD
1239	CUxD-TIME	DEVICE	CUxD
12	Gateway	DEVICE	---
10			BidCos-RF
104			HmIP-RF
79			HmIP-RF
108			HmIP-RF
35			BidCos-RF
111			BidCos-RF



Das dargestellte Menü ist abhängig von dem selektierten Objekttyp.

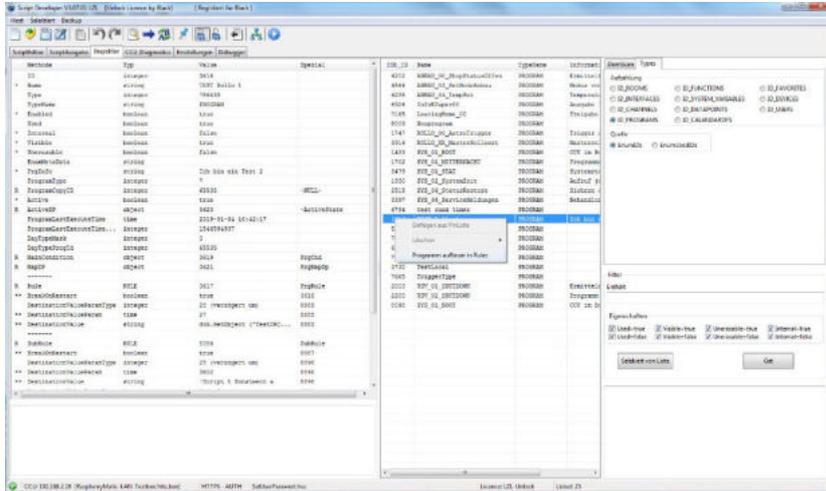
Bei Devices z.B.

Lassen sich alle untergeordneten Kanäle automatisch aus allen Räumen oder aus allen Gewerken entfernen. Dies war früher immer gerne eine Tipporgie.

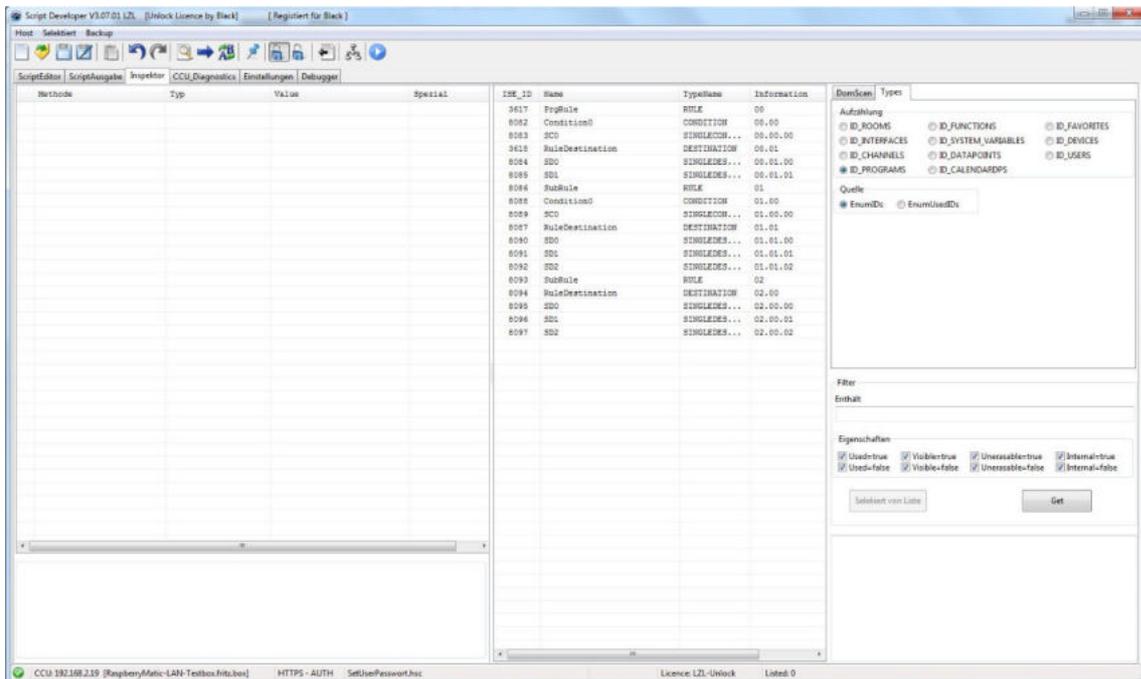
4.9.2 Auflösen von Programmen

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

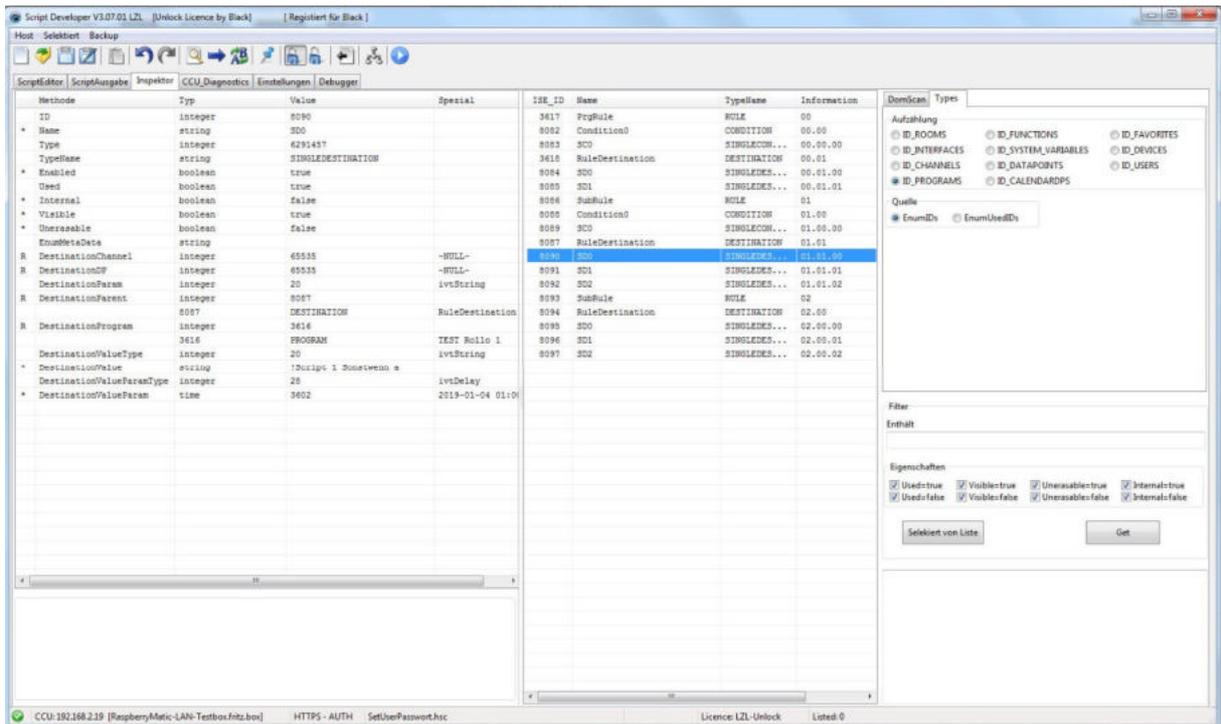
Aufgelöst wird adäquat zu den Devices: Auswahl über selektieren, dann rechte Maustaste und



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

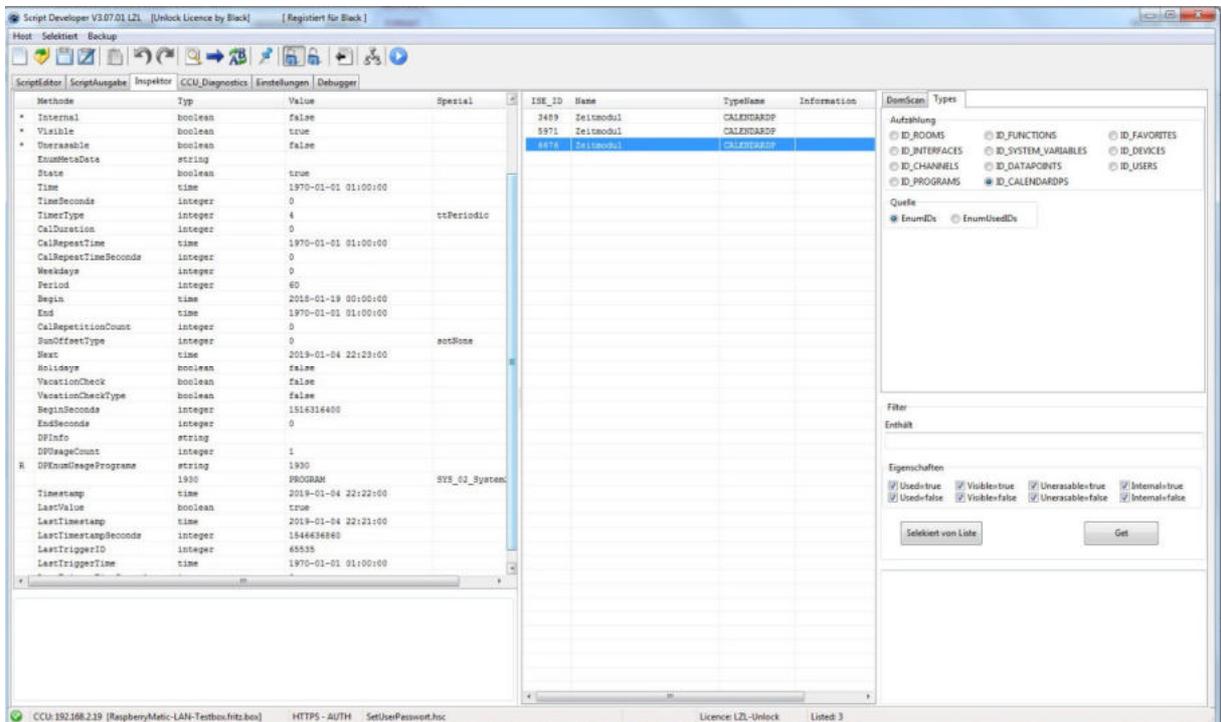


Die Sortierung über Information erlaubt eine chronologische Sortierung nach Auftreten der Objekte in einem Programm. Diese Liste lässt sich nun in der Detailansicht weiter untersuchen



4.10 Zeitmodule

Zeitmodule lassen sich nun auch in Detailansicht darstellen



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.

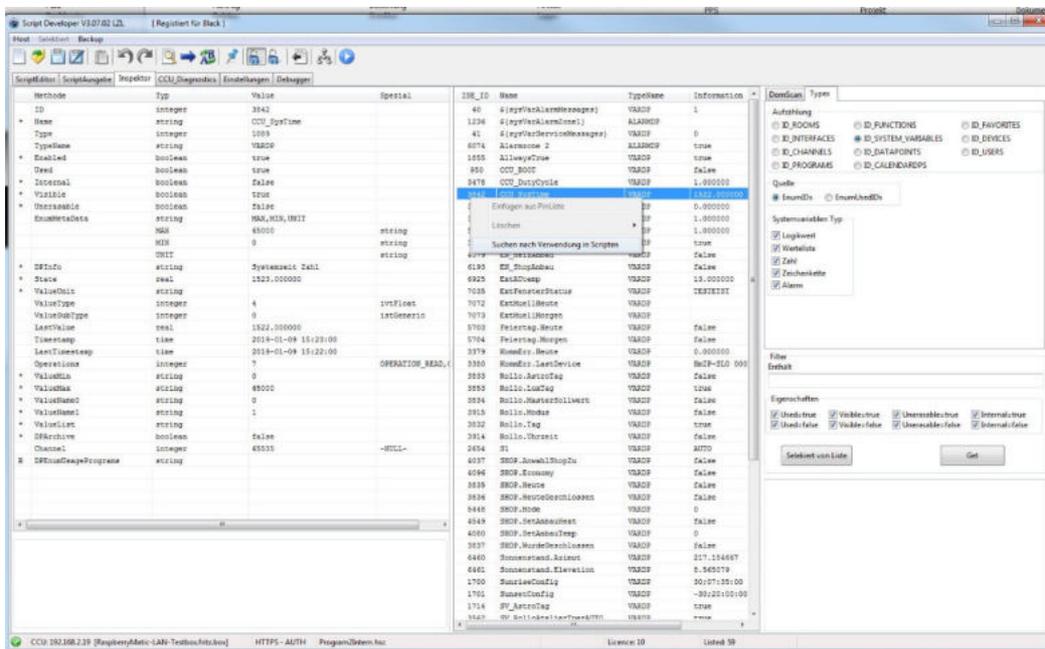
Dazu werden sämtliche SingleConditions, die als Property ein Script enthalten, gesucht und dann via Stringvergleich nach dem oder den Vorkommenden Namen abgesucht.

Vorgehensweise:

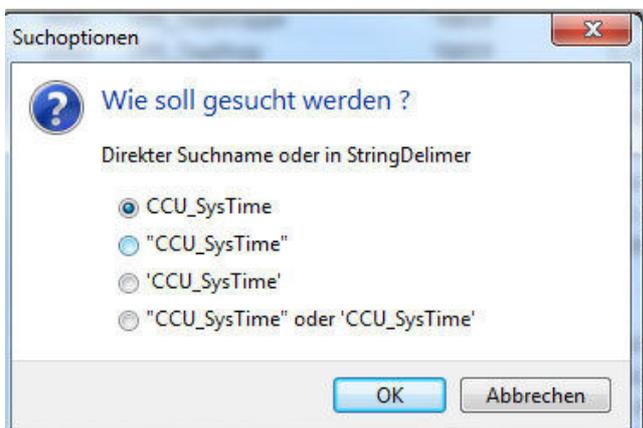
Markieren der oder auch mehrere Suchvariablen (Können Systemvariablen, Devices, Kanäle, Räume und Gewerke sein) Rechte Maustaste und suchen nach Verwendung in Scripten

Es kann hierbei gesucht werden:

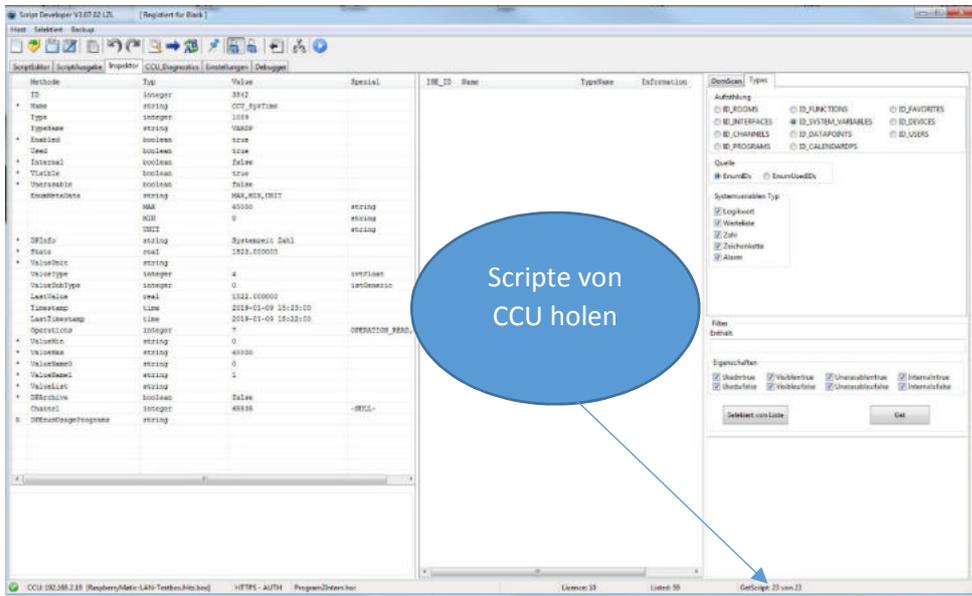
- Suchen "Name" nach Verwendung in Scripten : Suche nach dem Namen (SV, Alarme, Device, Chans)
- Suchen "Address" nach Verwendung in Scripten : Sucht nach der Seriennummer (Address) von Devices und Channels



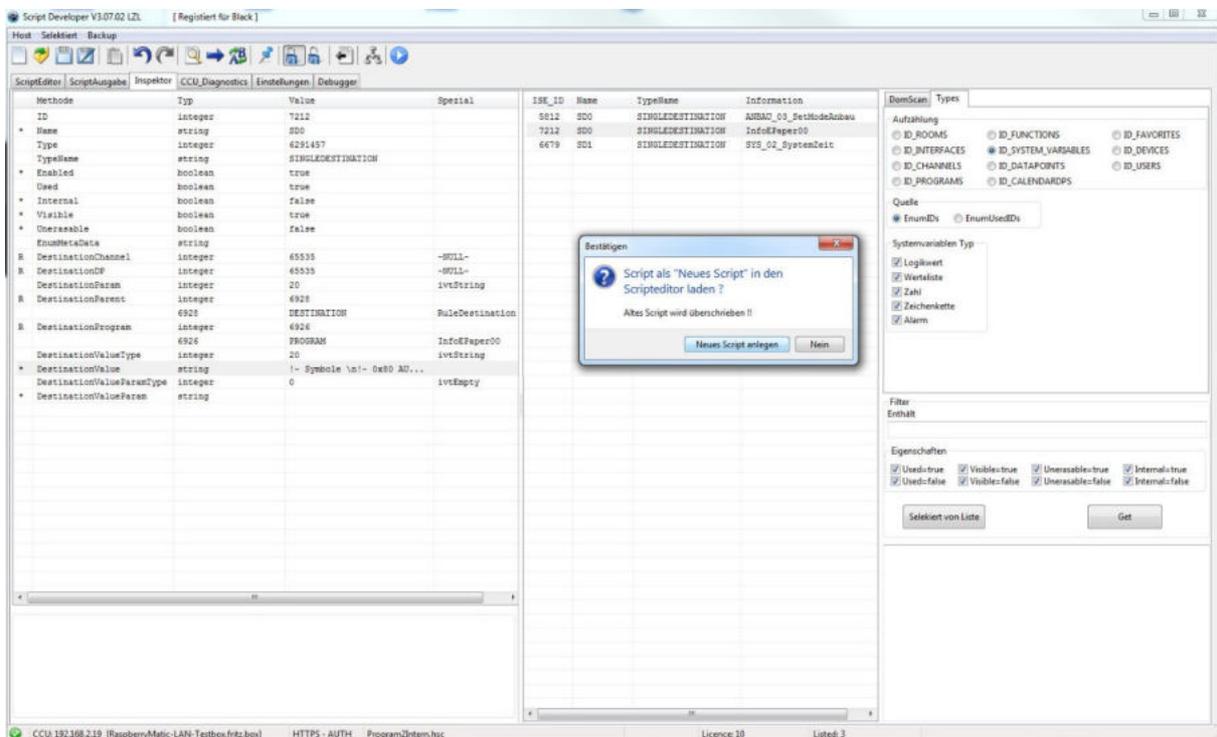
Im darauf sich öffnenden Dialog festlegen ob der reine Text oder der Text in StringDelimitern gesucht werden soll



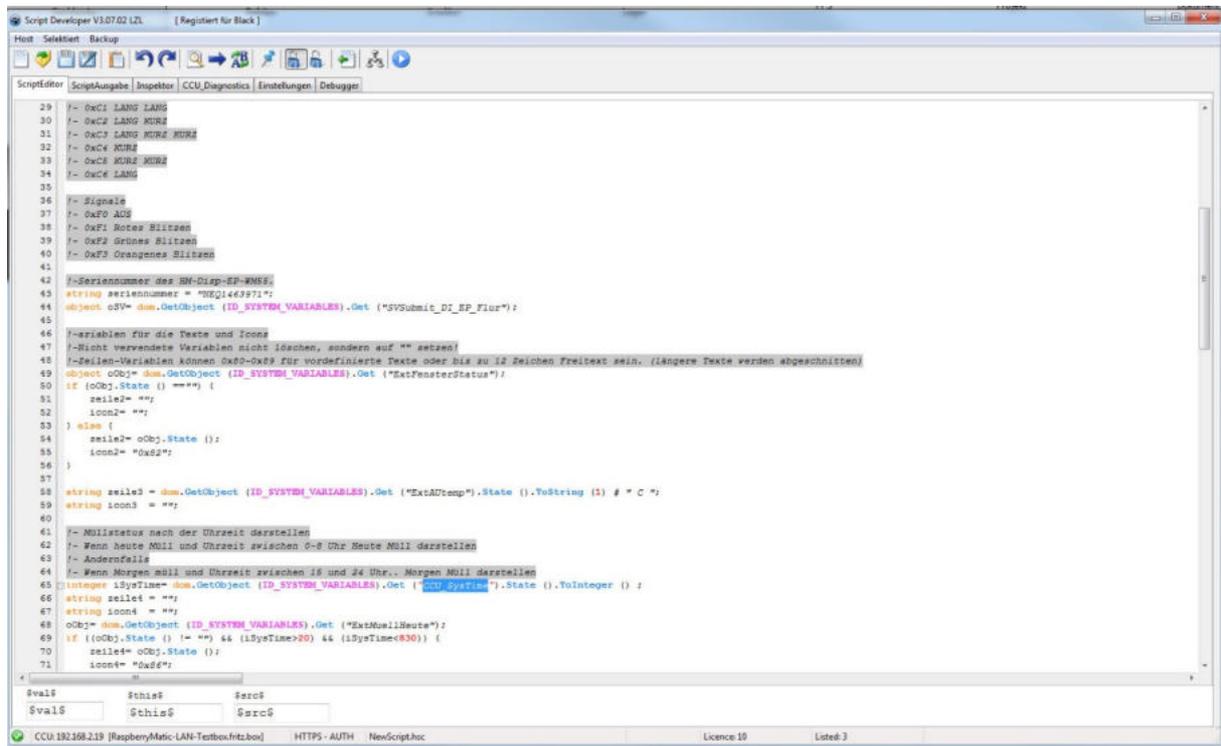
Daraufhin werden erstmal alle Rule in ihre Destinations aufgedrösel und eine Liste angelegt, wie viele Scripte es in diesen SingleDestinations dann gibt. Diese werden in den PC geladen. Das geht recht Fix, um der CCU zwischenzeitlich Zeit für Ihre Aufgaben zu lassen, wird direkt nach dem Empfang PC seitig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

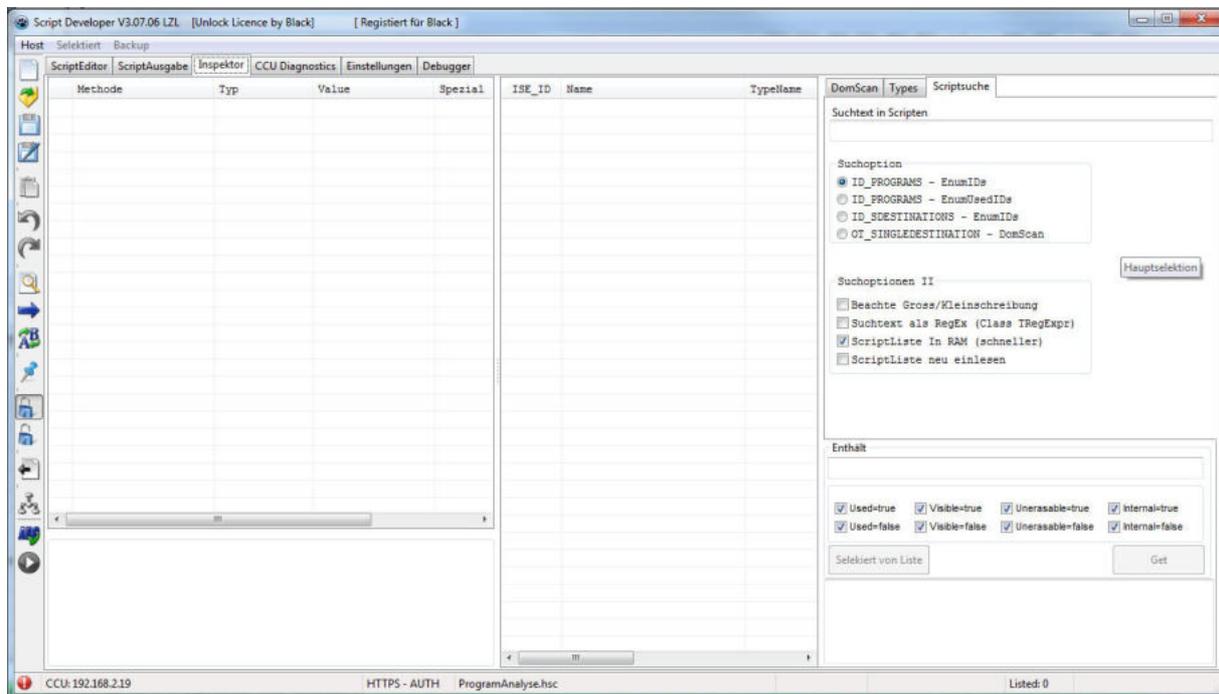


Der Editor öffnet sich und es wird direkt zum ersten Vorkommen des zu suchenden Namens gesprungen. bei mehrfachem Vorkommen sind alle Stellen gemäß den Markup Einstellungen hervorgehoben. Mit dem Pfeil für Weitersuchen lässt sich dann auch durch der Text nach dem Namen durchsuchen unabhängig von der 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



4.12 Volltextsuche in Skripten

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

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 Suchen Verwendungsstelle in WEB-UI Programmen (ab 3.09.04)

Aus dem Inspektor heraus kann nun komfortable in WebUI Programmen gesucht werden nach Verwendungen:
es kann gesucht werden:

1. über ein Device: Das Device wird über die Channels in seine Datenpunkte aufgelöst und mit dieser Datenpunktliste läuft dann die Programmanalyse. Ein Device führt also zu einem ID-Bündel
2. Über einen Channel: der Channel in seine Datenpunkte aufgelöst und mit dieser Datenpunktliste läuft dann die Programmanalyse. Ein Channel führt also zu einem ID-Bündel
3. HSSDP: dieser einzelne HSSDP wird gesucht
4. Sysvar: diese einzelne Sysvar wird gesucht

Diese ID liste (entweder eine oder auch mehrere IDs läuft dann durch die Programmanalyse). dabei wird geprüft:

eine dieser IDs ist ein Trigger des Programms -> Ergebnis in Inspektor

eine dieser IDs ist "nur prüfen" des Programms -> Ergebnis in Inspektor

eine dieser IDs ist Im Bedingungsteil des Programms (Trigger oder nur prüfen)-> Ergebnis in Inspektor

eine dieser IDs ist im Zuweisungsteil des Programms - Ergebnis in den Inspektor

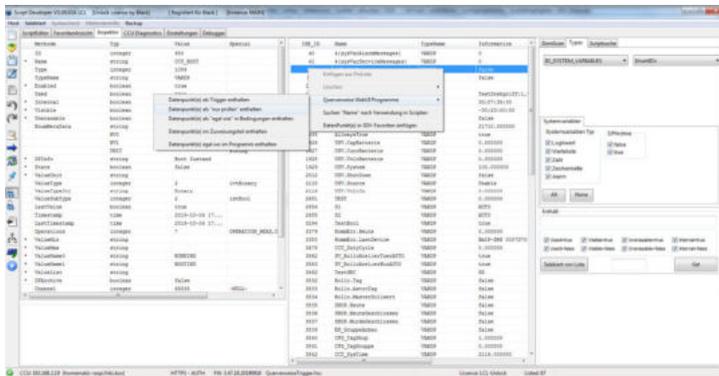
Eine dieser IDs kommt im Bedingungs oder Anweisungsteil vor --> Ergebnis in den Inspektor

(Scriptsuche ist davon unabhängig, der SDV konnte auch schon vorher volltextsuchen in Scripten)

Arbeitsweise:

Ist das selektierte Objekt ein Device, Channel, HSSDP oder Sysvar, gibt es auf der rechten Maustaste den Menüpunkt: Querverweise WebUI Programme.

Dabei kann nun eine der Suchvergänge ausgewählt werden

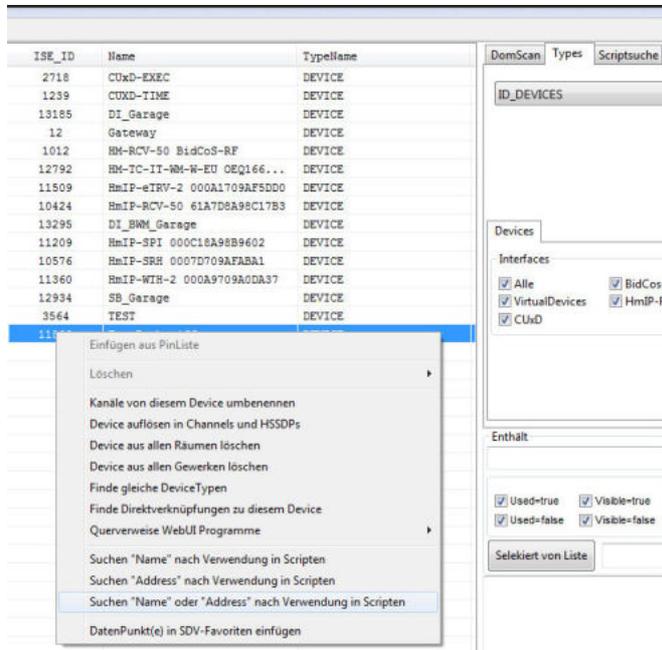


Ergebnis wäre hier die List von Programmen, wo in der WebUI eine Zuweisung auf die Sysvar Anwesenheit gemacht wird

ISE_ID	Name	TypeName	Information	DomScan	Types
8632	Demo	PROGRAM	DemoTest		
1433	SYS_01_ROOT	PROGRAM	CCU In Boot		ID_SYSTEM_VAR
					Systemvariablen

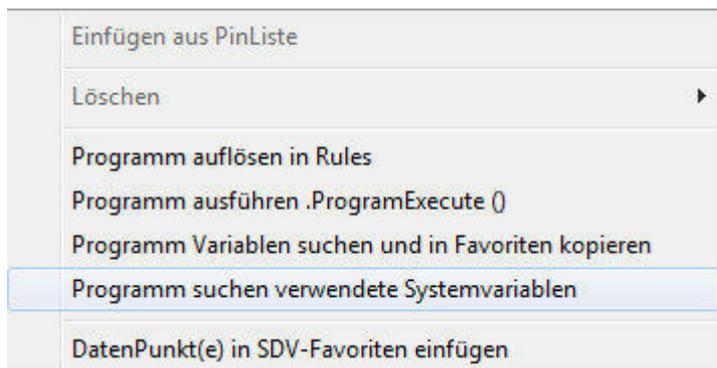
4.14 Suchen „Name“ oder „Adress“ nach Verwendung in Scripten

Das angeklickte Device im Inspektor wird nach Vorkommen auf „Name“ bzw. der „Adresse“ (Seriennummer) in Scripten gesucht.



4.15 Programm suchen verwendete Systemvariablen

Bei einem selektierten Programm besteht die Möglichkeit, sich die im WebUI Programm verwendeten Systemvariablen anzeigen zu lassen.



4.16 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

Bedingung: Wenn...

Geräteauswahl **DI_ATELIER_TUER:1** bei Fensterzustand: verriegelt bei Aktualisierung auslösen

UND

ODER

Aktivität: Dann... Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).

Systemzustand **S1** sofort AUTO

Skript **dom.GetObject ("TestSRC").State ("\$src\$")...** verzögert um **25** Sekunden

Bedingung: Sonst, wenn...

Systemzustand **S1** bei HAND bei Änderung auslösen

UND

ODER

Aktivität: Dann... Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).

Skript **!-Script qaaaa...** verzögert um **3601** Sekunden

Skript **Script2 wenn nnicht dann aber mindestens...** sofort

Systemzustand **CFG_TagGruppe** sofort **0,00**

Aktivität: Sonst... Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).

Skript **Script 1: Sonst...** sofort

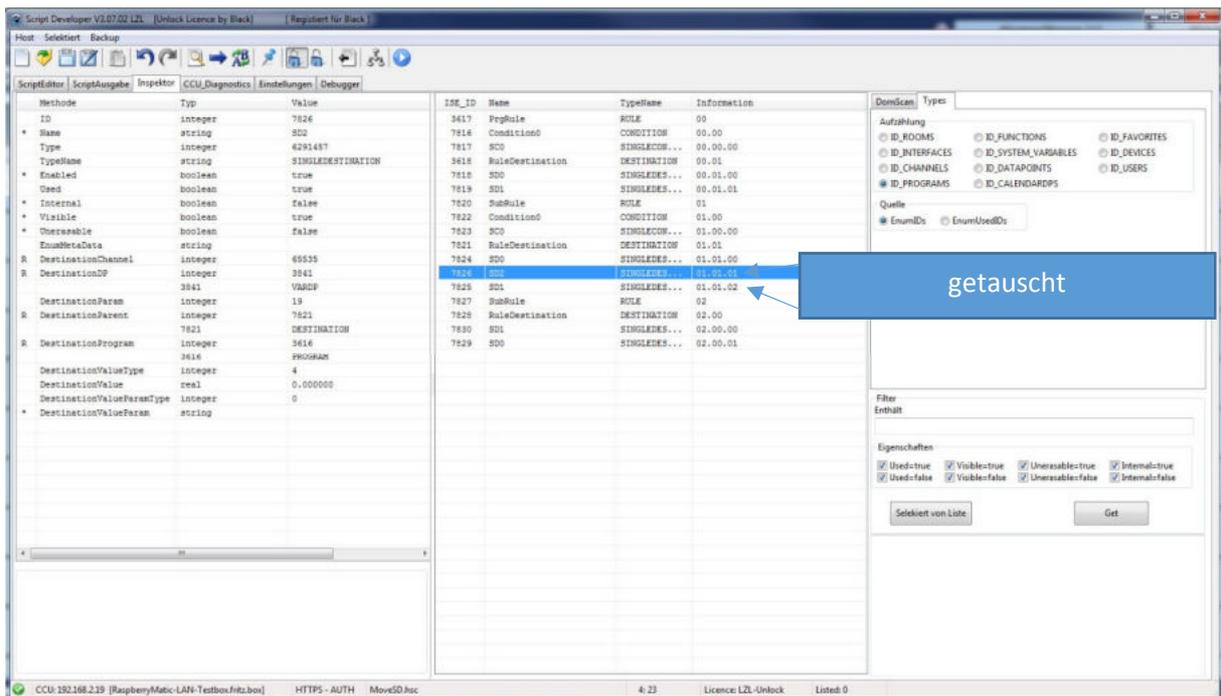
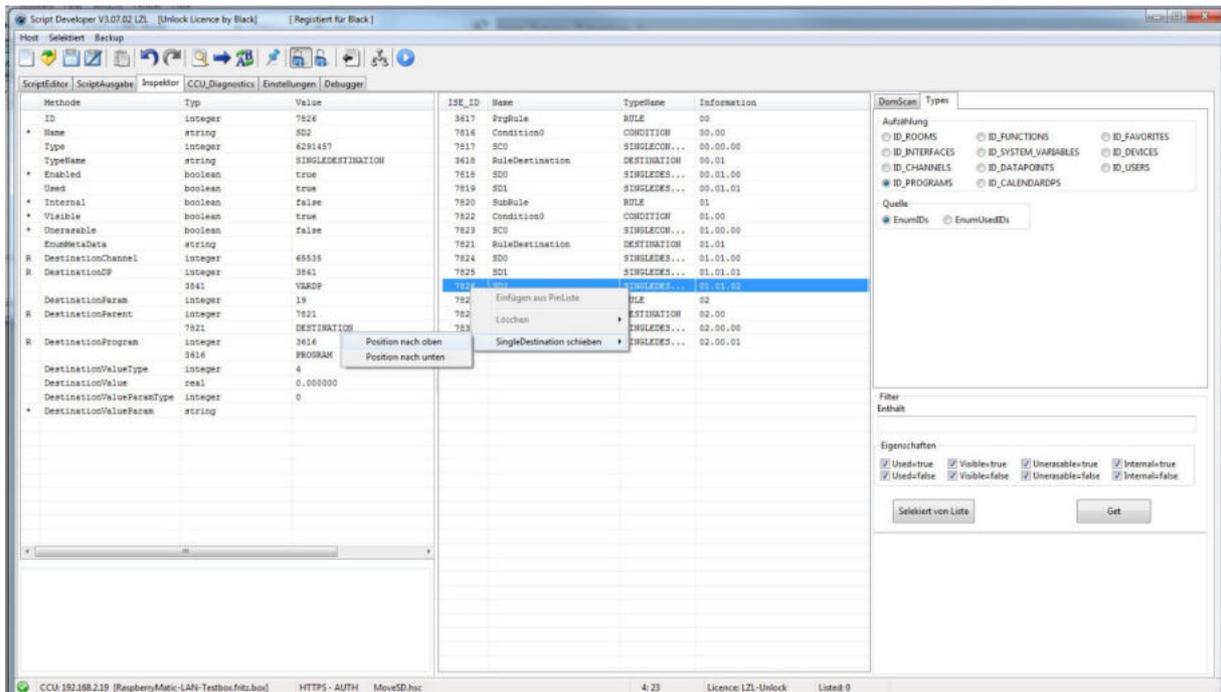
Geräteauswahl **Test:1** sofort Schaltzustand: ein

Das UrsprungScript

Das Aussehen im SDV

ID	Name	TypeName	Information
7817	Physikale	RULE	00
7818	Condition	CONDITION	00:00
7817	SD	SINGLEDEST...	00:00:00
7818	RuleDestination	DESTINATION	00:01
7818	SD	SINGLEDEST...	00:01:00
7819	SD	SINGLEDEST...	00:01:01
7820	SubRule	RULE	01
7822	Condition	CONDITION	01:00
7823	SD	SINGLEDEST...	01:00:00
7821	RuleDestination	DESTINATION	01:01
7824	SD	SINGLEDEST...	01:01:00
7829	SD	SINGLEDEST...	01:01:01
7826	SD	SINGLEDEST...	01:01:02
7827	SubRule	RULE	02
7825	RuleDestination	DESTINATION	02:00
7830	SD	SINGLEDEST...	02:00:00
7829	SD	SINGLEDEST...	02:00:01

Sieht hier so aus



Bedingung: Wenn...

Geräteauswahl bei Fensterzustand: verriegelt bei Aktualisierung auslösen

Aktivität: Dann... Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retrigger).

Systemzustand sofort AUTO a

Skript verzögert um Sekunden

Bedingung: Sonst, wenn...

Systemzustand bei HAND bei Änderung auslösen

Aktivität: Dann... Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retrigger).

Skript verzögert um Sekunden

Systemzustand sofort

Skript sofort

Aktivität: Sonst... Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retrigger).

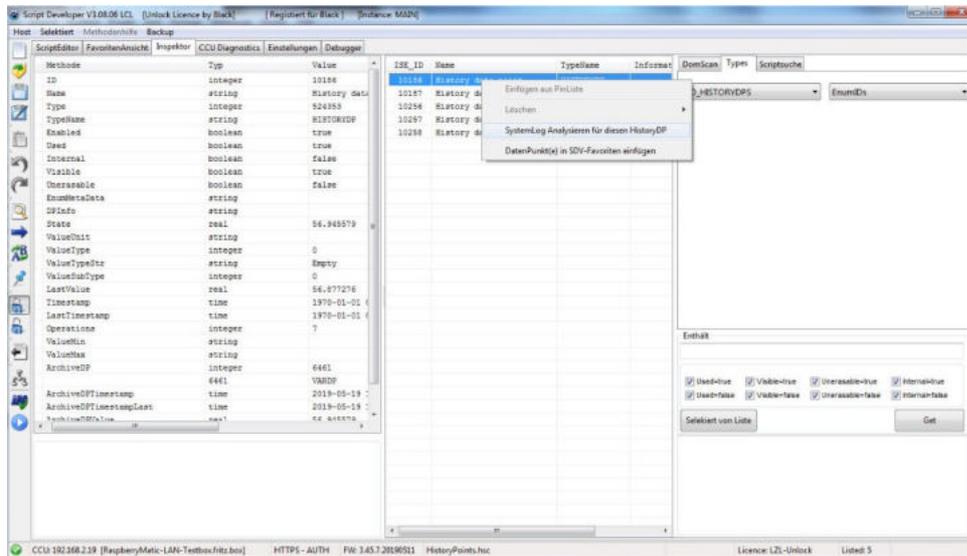
Skript sofort

Geräteauswahl sofort Schaltzustand: ein

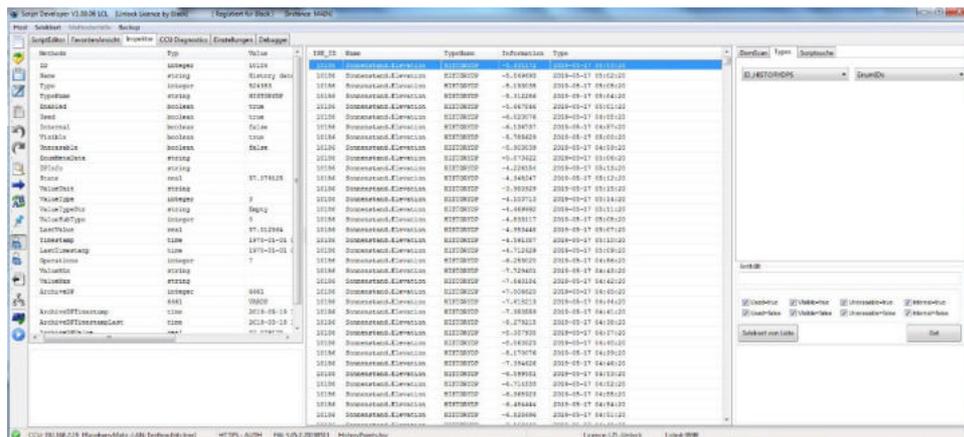
Und hier im WebUI Skript

4.17 History Data Points und Systemprotokoll

Es besteht die Möglichkeit, History Datapoints im Systemprotokoll zu Selektieren

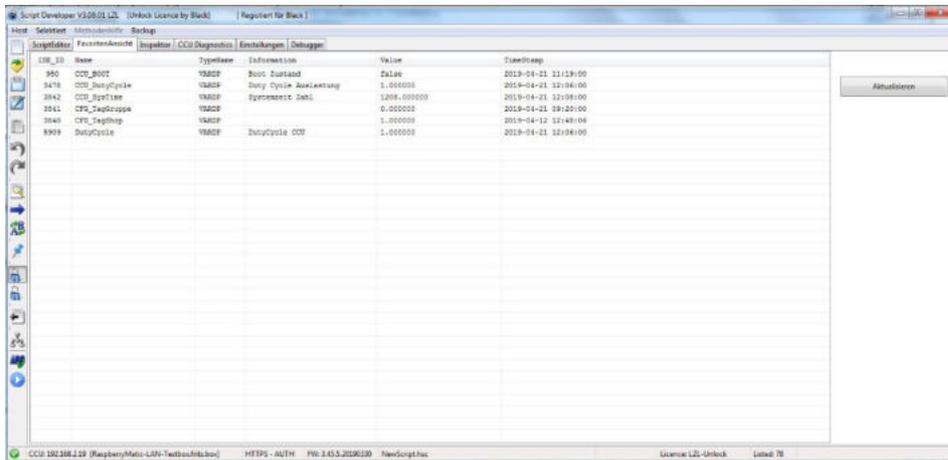


Testweise hat der punkt hier bei mir fast 10000 Einträge, dauert dann ein paar Sekunden, bis die Werte geholt sind



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, Alarme, Datenpunkte und Programme übernommen werden.

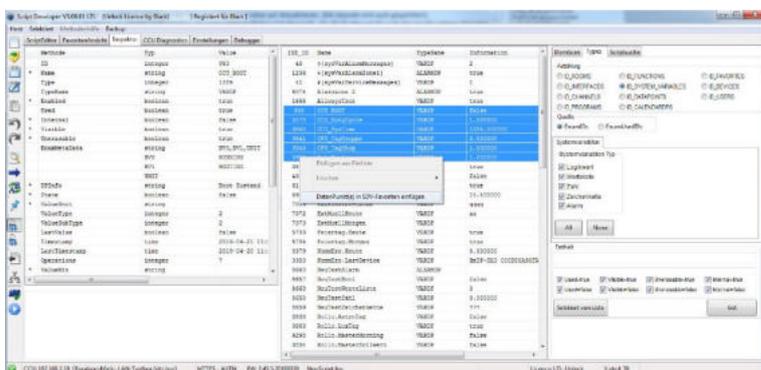
Will man einen Datenpunkt mal eben ändern, einen Doppelklick auf die Zeile der Favoriten, daraufhin wird unter Berücksichtigung des Undo / redo Stacks die Favoriten in die Listenansicht geladen und das angeklickte Objekt zur Bearbeitung in der Detailsicht geöffnet.

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

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

5.1 Hinzufügen von Objekten in die Favoriten

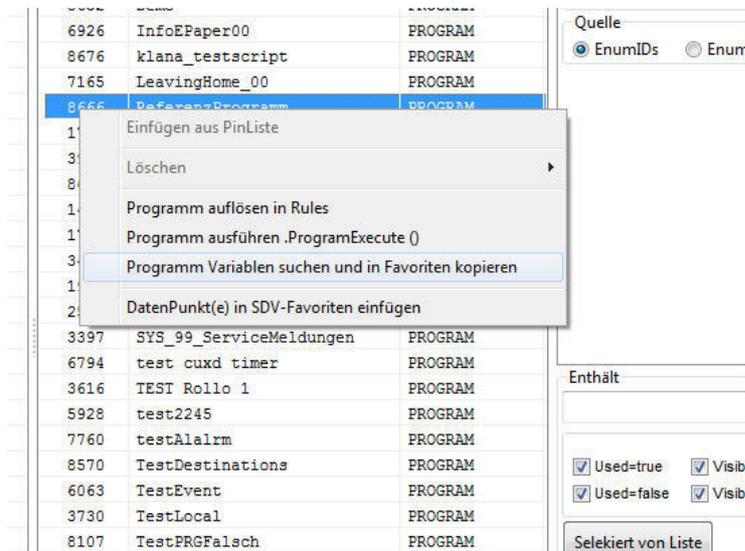
Objekte können aus der Listenansicht des Inspektors in die Favoriten geladen werden. Dazu Objekte Selektieren rechte Maustaste und



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 !!!

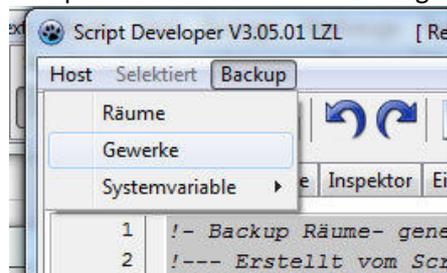
Vielmehr dienen diese im Falle eines Umzuges von einem alt System auf ein Neusystem als Hilfestellung, wenn man das alte Systembackup nicht benutzen will (Loswerden von in den Jahren angesammelten Leichen), oder aber ein inkonsistentes System.

Ab Version 3.09.03 besteht die Möglichkeit, jeder CCU ein separates Backup Verzeichnis zuzuordnen. Damit werden Verwechslungen vermieden, wenn man mit 2 CCU arbeitet. (Beide CCUs haben ein jeweils eigenes Backup-Verzeichnis)

z.B.

```
[CCU1]
IP=192.168.2.19
USERNAME=ScriptAdmin
PASSWORD=xxxxxxxxxxxxxxxxxxxxxx
USEHTTPS=true
SSHUSERNAME=root
SSHPW=xxxxxxxxxxxxxxxxxxxxxx
BACKUPDIR=C:\MTH\Homematic\Backups\CCU_19\
```

Den passenden Lizenzlevel vorausgesetzt, finden 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 angelegt, 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 angelegt, 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 Alarme immer scharf, auch wenn dieser Alarm vorher im Alt System über AlArm (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:

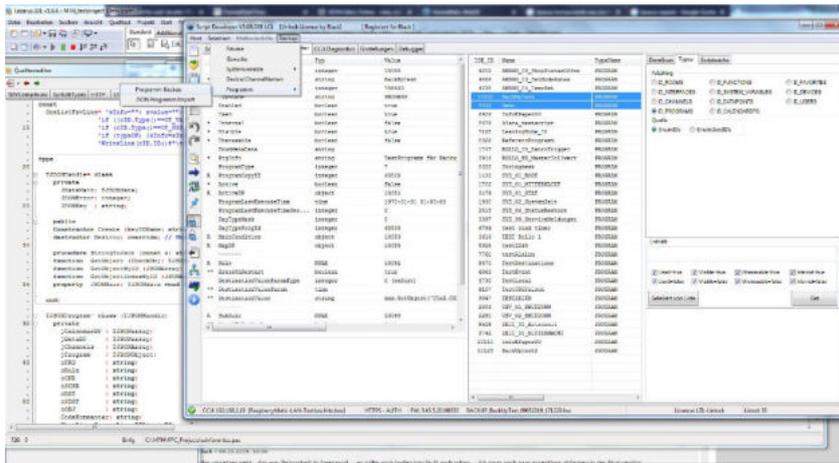
Für die Programme gibt es die Möglichkeit, den Backupnamen in der INI festzulegen.

```
[LAST]
(...)
BACKUPNAME=$NAME_BACKUP_$FW_$DATE_$TIME
```

Bedeutung der Abkürzungen

- \$NAME: der Name des Programmes wird eingesetzt
- \$FW: Firmwareversion
- \$DATE: Datum in der Form JJJJMMTT
- \$TIME: Zeit in der Form HHMMSS

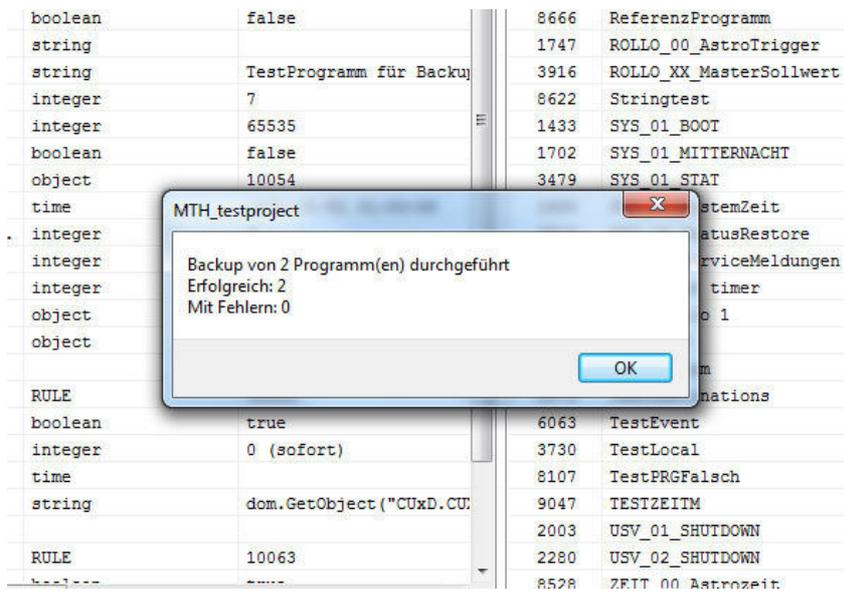
Programme in der Listenauswahl selektieren, eins oder mehrere...



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

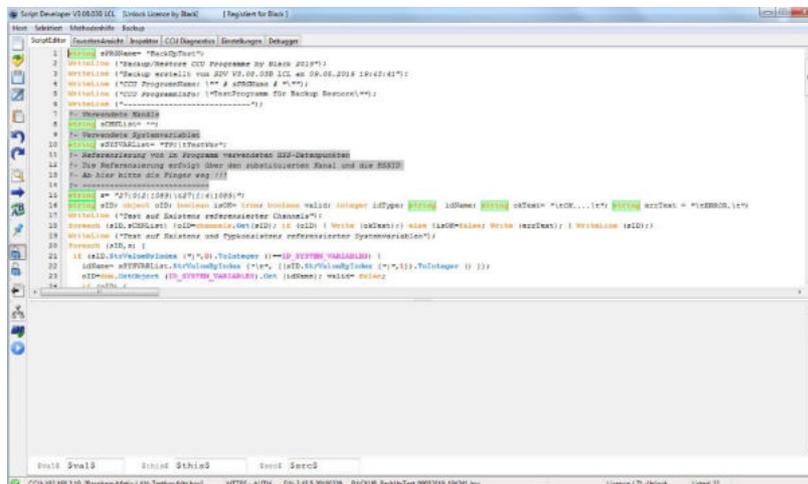
Der Programmname heisst 'BACKUP_ + ProgrammName + _ + TTMMYYYY_HHMMSS.hsc bzw .json danach werden die Backups der ausgewählten Programme angelegt. werden Objekte selektiert, die keine Programme sind, so werden diese natürlich ignoriert.

So sollte es aussehen, wenn alles durchlief:



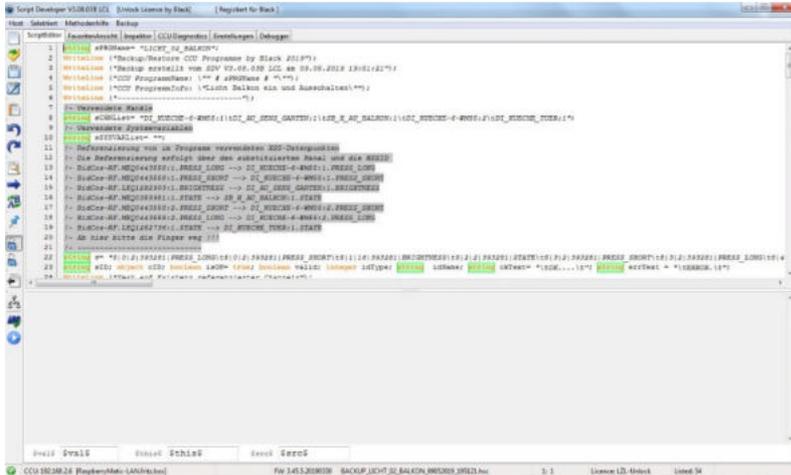
ab V8.03.04 wird auch noch die Laufzeit für das Backup in Sekunden angezeigt. In der unteren Infozeile erfolgt Anzeige ganz rechts, welches Programm von wie vielen gerade im Backupprozess ist. Danach liegen im Verzeichnis 2 Files.

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 zusammenbastel 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.

```
1 #Scripter
2 #Metadaten
3 #Backup/Restore
4 #Metadaten
5 #Metadaten
6 #Metadaten
7 #Metadaten
8 #Metadaten
9 #Metadaten
10 #Metadaten
11 #Metadaten
12 #Metadaten
13 #Metadaten
14 #Metadaten
15 #Metadaten
16 #Metadaten
17 #Metadaten
18 #Metadaten
19 #Metadaten
20 #Metadaten
21 #Metadaten
22 #Metadaten
23 #Metadaten
24 #Metadaten
25 #Metadaten
26 #Metadaten
27 #Metadaten
28 #Metadaten
29 #Metadaten
30 #Metadaten
31 #Metadaten
32 #Metadaten
33 #Metadaten
34 #Metadaten
35 #Metadaten
36 #Metadaten
37 #Metadaten
38 #Metadaten
39 #Metadaten
40 #Metadaten
41 #Metadaten
42 #Metadaten
43 #Metadaten
44 #Metadaten
45 #Metadaten
46 #Metadaten
47 #Metadaten
48 #Metadaten
49 #Metadaten
50 #Metadaten
51 #Metadaten
52 #Metadaten
53 #Metadaten
54 #Metadaten
55 #Metadaten
56 #Metadaten
57 #Metadaten
58 #Metadaten
59 #Metadaten
60 #Metadaten
61 #Metadaten
62 #Metadaten
63 #Metadaten
64 #Metadaten
65 #Metadaten
66 #Metadaten
67 #Metadaten
68 #Metadaten
69 #Metadaten
70 #Metadaten
71 #Metadaten
72 #Metadaten
73 #Metadaten
74 #Metadaten
75 #Metadaten
76 #Metadaten
77 #Metadaten
78 #Metadaten
79 #Metadaten
80 #Metadaten
81 #Metadaten
82 #Metadaten
83 #Metadaten
84 #Metadaten
85 #Metadaten
86 #Metadaten
87 #Metadaten
88 #Metadaten
89 #Metadaten
90 #Metadaten
91 #Metadaten
92 #Metadaten
93 #Metadaten
94 #Metadaten
95 #Metadaten
96 #Metadaten
97 #Metadaten
98 #Metadaten
99 #Metadaten
100 #Metadaten
```

```
1 #Scripter
2 #Metadaten
3 #Backup/Restore
4 #Metadaten
5 #Metadaten
6 #Metadaten
7 #Metadaten
8 #Metadaten
9 #Metadaten
10 #Metadaten
11 #Metadaten
12 #Metadaten
13 #Metadaten
14 #Metadaten
15 #Metadaten
16 #Metadaten
17 #Metadaten
18 #Metadaten
19 #Metadaten
20 #Metadaten
21 #Metadaten
22 #Metadaten
23 #Metadaten
24 #Metadaten
25 #Metadaten
26 #Metadaten
27 #Metadaten
28 #Metadaten
29 #Metadaten
30 #Metadaten
31 #Metadaten
32 #Metadaten
33 #Metadaten
34 #Metadaten
35 #Metadaten
36 #Metadaten
37 #Metadaten
38 #Metadaten
39 #Metadaten
40 #Metadaten
41 #Metadaten
42 #Metadaten
43 #Metadaten
44 #Metadaten
45 #Metadaten
46 #Metadaten
47 #Metadaten
48 #Metadaten
49 #Metadaten
50 #Metadaten
51 #Metadaten
52 #Metadaten
53 #Metadaten
54 #Metadaten
55 #Metadaten
56 #Metadaten
57 #Metadaten
58 #Metadaten
59 #Metadaten
60 #Metadaten
61 #Metadaten
62 #Metadaten
63 #Metadaten
64 #Metadaten
65 #Metadaten
66 #Metadaten
67 #Metadaten
68 #Metadaten
69 #Metadaten
70 #Metadaten
71 #Metadaten
72 #Metadaten
73 #Metadaten
74 #Metadaten
75 #Metadaten
76 #Metadaten
77 #Metadaten
78 #Metadaten
79 #Metadaten
80 #Metadaten
81 #Metadaten
82 #Metadaten
83 #Metadaten
84 #Metadaten
85 #Metadaten
86 #Metadaten
87 #Metadaten
88 #Metadaten
89 #Metadaten
90 #Metadaten
91 #Metadaten
92 #Metadaten
93 #Metadaten
94 #Metadaten
95 #Metadaten
96 #Metadaten
97 #Metadaten
98 #Metadaten
99 #Metadaten
100 #Metadaten
```

Bei einem Erfolgreichen Durchlauf sieht die Scriptausgabe so aus:

```
Backup erstellt vom SDV V3.08.03B LCL 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.

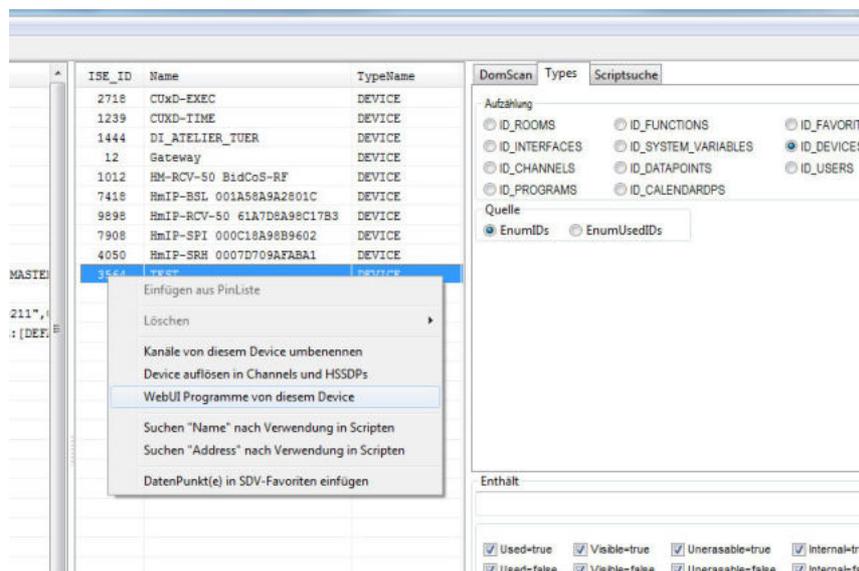
Funktion ist unspektakulär kann die WebUI auch (habs nur programmiert, weil ein Device keine idarray hat, wo die verwendeten Programme drinstehen)

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..)

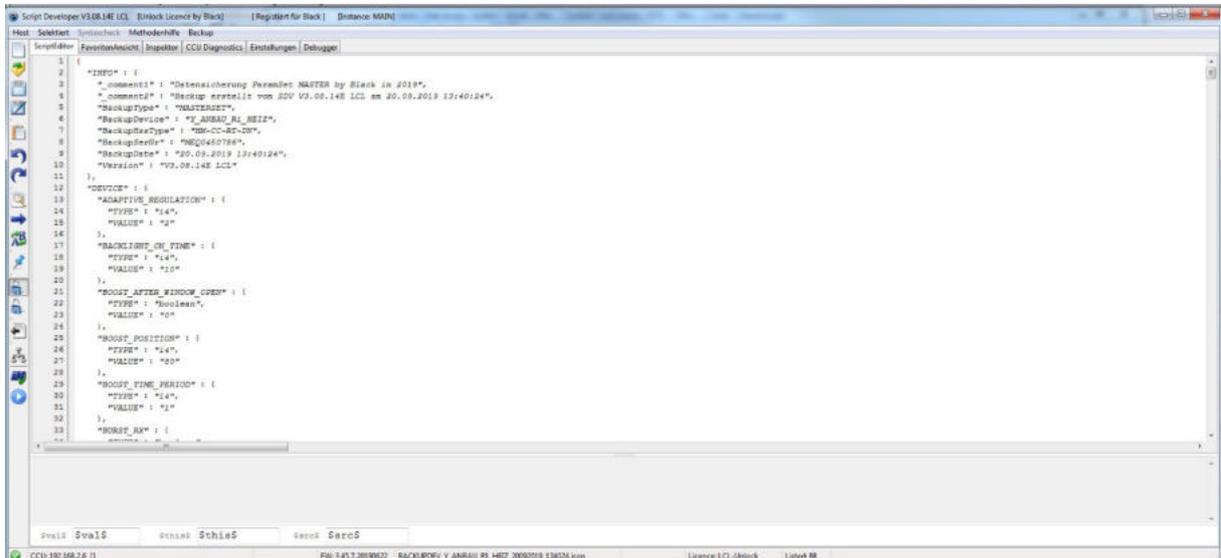
natürlich müssen die neuen Kanal Bezeichner richtig eingegeben worden sein, sonst gibt das Shit in - Shit out Prinzip



6.1.6 Backup Masterset

Es ist möglich, von einem Gerät (oder von mehreren Selektierten) ein Backup zu erstellen. Dazu werden das oder die Geräte Selektiert, dann Backup – Paramset Master von Device

Dabei wird von allen selektieren Geräten eine JSON Datei von den Geräteeinstellungen angelegt. Dieses JSON kann in den Editor geladen und dann ausgeführt werden.



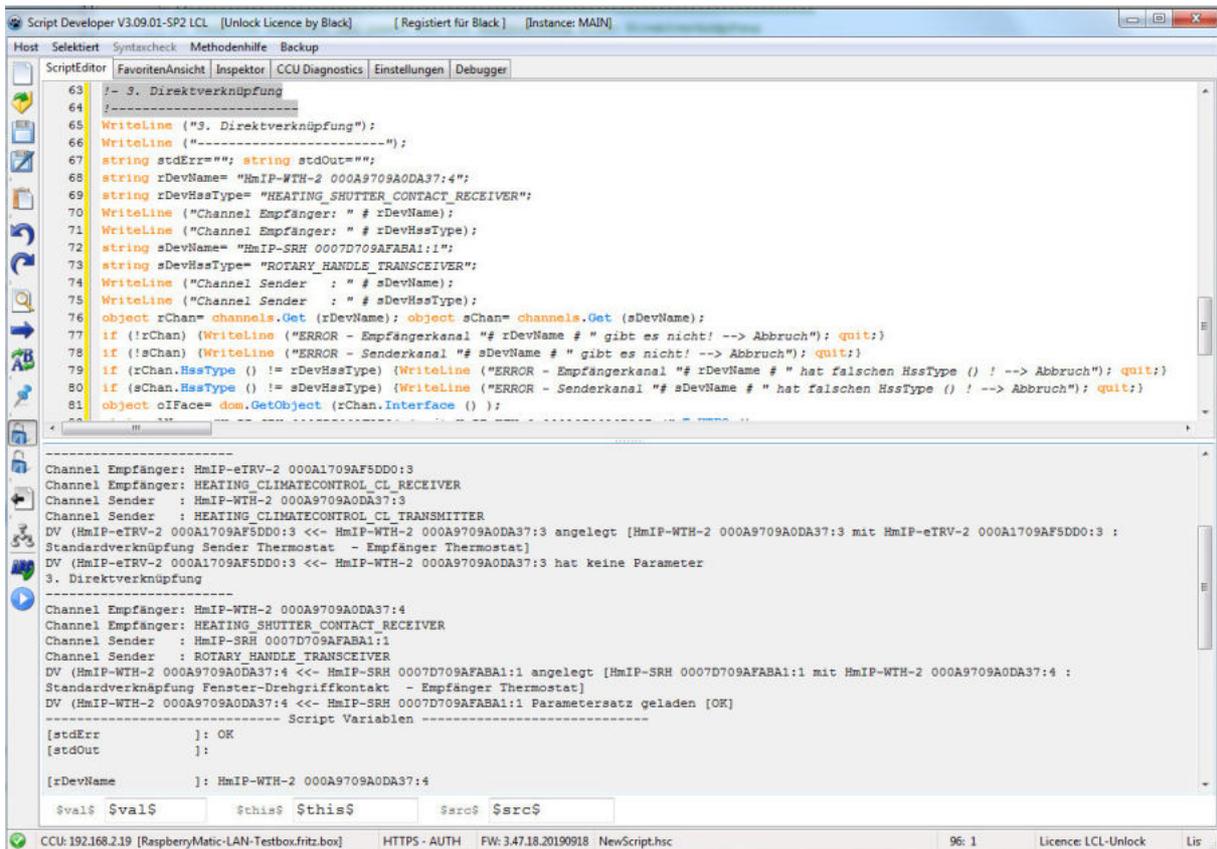
```
1  "INFO" : {
2    "_comment1" : "Datenicherung Paramset MASTER by Black im 2019",
3    "_comment2" : "Backup erstellt von SDV V3.08.148 LCL am 20.09.2019 13:40:24",
4    "BackupType" : "MASTERSET",
5    "BackupDevice" : "T_ANBAU_RL_HEIZ",
6    "BackupSrcType" : "DW-CC-BT-DW",
7    "BackupSerial" : "MEQ040788",
8    "BackupDate" : "20.09.2019 13:40:24",
9    "Version" : "V3.08.148 LCL"
10  },
11
12  "DEVICE" : {
13    "ADAPTIVE_REGULATION" : {
14      "TYPE" : "14",
15      "VALUE" : "2"
16    },
17    "BACKLIGHT_ON_TIME" : {
18      "TYPE" : "14",
19      "VALUE" : "10"
20    },
21    "BOOST_AFTER_WINDOW_OPEN" : {
22      "TYPE" : "Boolean",
23      "VALUE" : "0"
24    },
25    "BOOST_POSITION" : {
26      "TYPE" : "14",
27      "VALUE" : "20"
28    },
29    "BOOST_FINE_PERIOD" : {
30      "TYPE" : "14",
31      "VALUE" : "1"
32    },
33    "BURST_RX" : {
```

Dabei werden die Einstellungen des JSON an das Gerät übertragen. Wenn man weiss, was man tut, kann man einzelne Werte auch manuell verändern.

6.1.7 Backup Linkset

Es ist möglich, von einem Gerät (oder von mehreren Selektierten) ein Backup der zu dem Gerät definierten Direktverknüpfungen zu erstellen. Dazu werden das oder die Geräte Selektiert, dann Backup – Paramset Link von Device

Dabei wird von allen selektieren Geräten eine JSON Datei von den zu den Geräten definierten Direktverknüpfungen angelegt. Dieses JSON kann in den Editor geladen und dann ausgeführt werden. Dabei werden die Direktverknüpfungen angelegt über AddLink und anschließend die Werte der Expertenparameter, so vorhanden, geladen



```
Script Developer V3.09.01-SP2 LCL [Unlock Licence by Black] [Registriert für Black] [Instance: MAIN]
Host Selektiert Syntaxcheck Methodenhilfe Backup
ScriptEditor Favoritenansicht Inspektor CCU Diagnostics Einstellungen Debugger

63 1- 3. Direktverknüpfung
64
65 Writeline ("3. Direktverknüpfung");
66 Writeline ("-----");
67 string stdErr=""; string stdOut="";
68 string rDevName= "HmIP-WTH-2 000A9709A0DA37:4";
69 string rDevHssType= "HEATING_SHUTTER_CONTACT_RECEIVER";
70 Writeline ("Channel Empfänger: " # rDevName);
71 Writeline ("Channel Empfänger: " # rDevHssType);
72 string sDevName= "HmIP-SRH 0007D709AFAB1:1";
73 string sDevHssType= "ROTARY_HANDLE_TRANSCEIVER";
74 Writeline ("Channel Sender : " # sDevName);
75 Writeline ("Channel Sender : " # sDevHssType);
76 object rChan= channels.Get (rDevName); object sChan= channels.Get (sDevName);
77 if (!rChan) (Writeline ("ERROR - Empfängerkanal "# rDevName # " gibt es nicht! --> Abbruch"); quit:);
78 if (!sChan) (Writeline ("ERROR - Senderkanal "# sDevName # " gibt es nicht! --> Abbruch"); quit:);
79 if (rChan.HssType () != rDevHssType) (Writeline ("ERROR - Empfängerkanal "# rDevName # " hat falschen HssType () ! --> Abbruch"); quit:);
80 if (sChan.HssType () != sDevHssType) (Writeline ("ERROR - Senderkanal "# sDevName # " hat falschen HssType () ! --> Abbruch"); quit:);
81 object oIFace= dom.GetObject (rChan.Interface () );

Channel Empfänger: HmIP-eTRV-2 000A1709AF5DD0:3
Channel Empfänger: HEATING_CLIMATECONTROL_CL_RECEIVER
Channel Sender : HmIP-WTH-2 000A9709A0DA37:3
Channel Sender : HEATING_CLIMATECONTROL_CL_TRANSMITTER
DV (HmIP-eTRV-2 000A1709AF5DD0:3 <<- HmIP-WTH-2 000A9709A0DA37:3 angelegt [HmIP-WTH-2 000A9709A0DA37:3 mit HmIP-eTRV-2 000A1709AF5DD0:3 :
Standardverknüpfung Sender Thermostat - Empfänger Thermostat]
DV (HmIP-eTRV-2 000A1709AF5DD0:3 <<- HmIP-WTH-2 000A9709A0DA37:3 hat keine Parameter
3. Direktverknüpfung
-----
Channel Empfänger: HmIP-WTH-2 000A9709A0DA37:4
Channel Empfänger: HEATING_SHUTTER_CONTACT_RECEIVER
Channel Sender : HmIP-SRH 0007D709AFAB1:1
Channel Sender : ROTARY_HANDLE_TRANSCEIVER
DV (HmIP-WTH-2 000A9709A0DA37:4 <<- HmIP-SRH 0007D709AFAB1:1 angelegt [HmIP-SRH 0007D709AFAB1:1 mit HmIP-WTH-2 000A9709A0DA37:4 :
Standardverknüpfung Fenster-Drehgriffkontakt - Empfänger Thermostat]
DV (HmIP-WTH-2 000A9709A0DA37:4 <<- HmIP-SRH 0007D709AFAB1:1 Parametersatz geladen [OK]
----- Script Variablen -----
[stdErr      ]: OK
[stdOut      ]:
[rDevName    ]: HmIP-WTH-2 000A9709A0DA37:4
$val$ $val$      $this$ $this$      $src$ $src$
CCU: 192.168.2.19 [RaspberryMatic-LAN-Testbox.fritz.box] HTTPS - AUTH FW: 3.47.18.20190918 NewScript.hsc 96: 1 Licence LCL-Unlock Lis
```

6.1.8 Backup Gerät von der WebUI

Beim Entfernen eines Gerätes werden ja alle WebUI Bezüge getilt. Nach dem Neuanlernen sind ja alle Einträge in Räume, Gewerke, Favoriten und sehr ärgerlich, in den Programmen weg. Mit dem neuen Menüpunkt besteht die Möglichkeit, die WebUI Einträge, die zu dem gerät gehörten wieder vollständig zu rekonstruieren.

Master und Linksets werden hierbei nicht angepackt, dazu gibt's die separaten Backupmöglichkeiten.

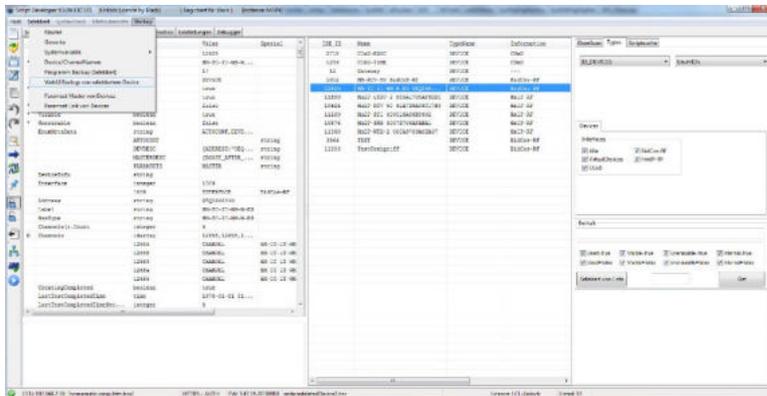
Es ist hierbei sogar möglich, der WebUI ein Baugleiches IP gerät unterzuschieben und zu rekonstruieren, dafür muss nach dem anlernen des anderen Gerätes nur der Geräte Name auf den alten Namen geändert werden.

Ansonsten ist der Identifiziermechanismus

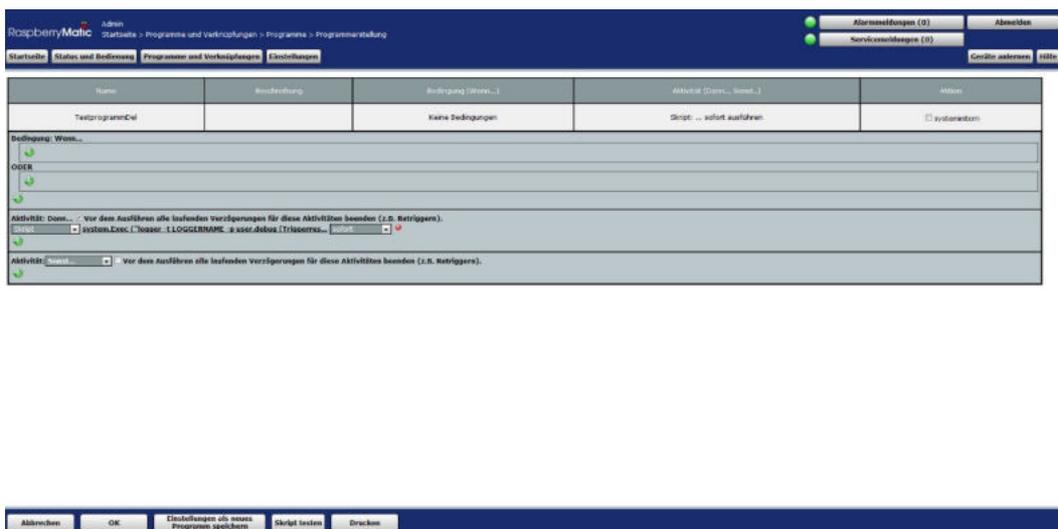
Seriennummer, wenn nicht gefunden --> Gerätename, wenn auch nicht gefunden Error

Vorgehensweise:

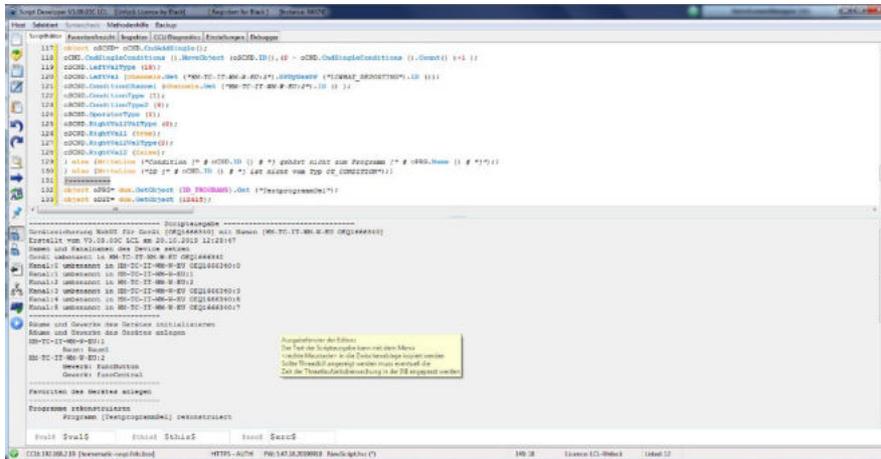
Gerät selektieren



Im folgenden Dialog das Backup Verzeichnis festlegen. Nach Durchführung liegt das generierte Restoreprogramm als Datei vor und liegt ebenfalls in der Windows Zwischenablage. Nach dem Löschen des Gerätes sieht ein Programm, welches das Gerät beinhaltet, ein wenig zerrupft aus:



Das generierte Programm kann nun auf der CCU unter Script ausführen oder im SDV in den Scripteditor geladen und dort ausgeführt werden
 Die Statusausgabe verrät auch was alles angepasst und geändert wurde



- Das Gerät wird anhand seiner Seriennummer identifiziert, wenn das fehlschlägt, anhand des Gerätenamens
- Der Gerätename und die Namen aller Kanäle des Gerätes werden auf den Namen aus der Sicherung geändert.
- Alle Räume und Gewerke, denen es beim Anlernen eventuell zugewiesen wurde werden entfernt
- die Kanäle werden allen Räumen und Gewerken gemäß der Sicherung zugeordnet.
- In den Favoriten werden die Kanäle des Geräte wieder richtig (an die ursprüngliche Stelle) eingefügt
- In allen Programmen, in denen das Gerät bzw Datenpunkte der Kanäle des Geräte verwendet wurden und wo diese Bezüge (SingleConditions/SingleDestinations) des Gerätes beim Löschen des Gerätes auch gelöscht wurden, werden diese auch wieder an der richtigen Stelle rekonstruiert. (Zum Einsatz kommen hier auch Mechanismen aus dem schon lange laufenden Programme-Backup)

Auch das Programm schaut anschließend wieder so aus wie vor dem Löschen des Gerätes

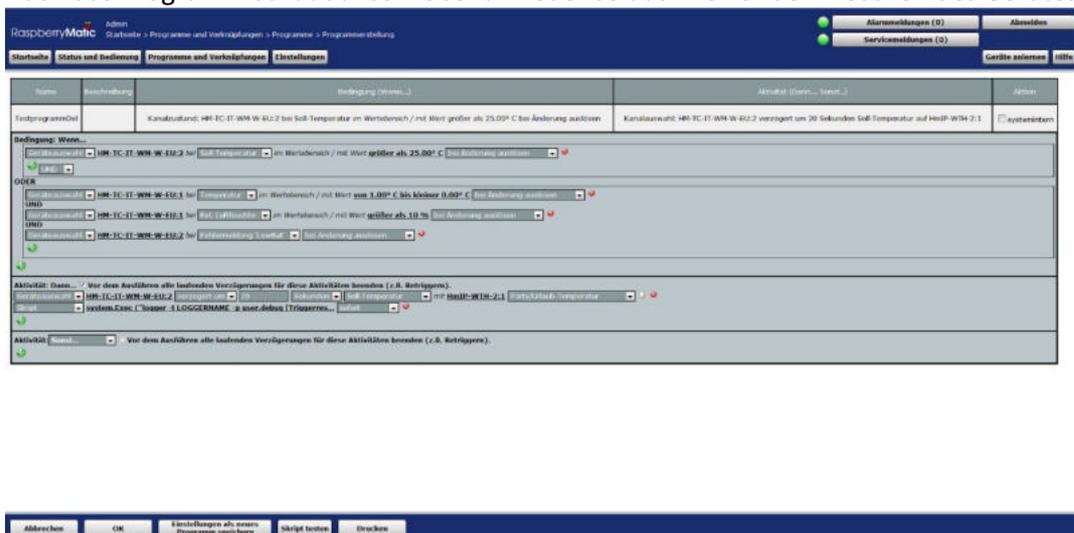


Diagramme kann er nicht rekonstruieren und wird er auch nicht können, da ich die Diagrammfunktion der CCU gerade gesagt selber suboptimal finde, ich diese selber nicht nutze (bei mir läuft IOBroker), ich mir also auch die Zeit schenke, dafür etwas zu programmieren.

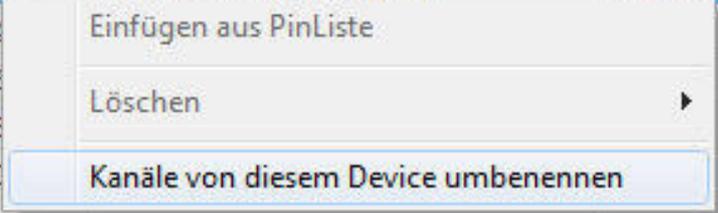
6.2 Umbenennen von Kanälen von Geräten

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 Namen 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			HmIP-RF
76			HmIP-RF
75			HmIP-RF
40			HmIP-RF



A context menu is displayed over the row with ISE_ID 1012. The menu items are: 'Einfügen aus PinListe', 'Löschen', and 'Kanäle von diesem Device umbenennen'. The 'Kanäle von diesem Device umbenennen' option is highlighted in blue.

Rückfrage mit Ja bestätigen und die Kanäle werden so benannt:

Device: DeviceName

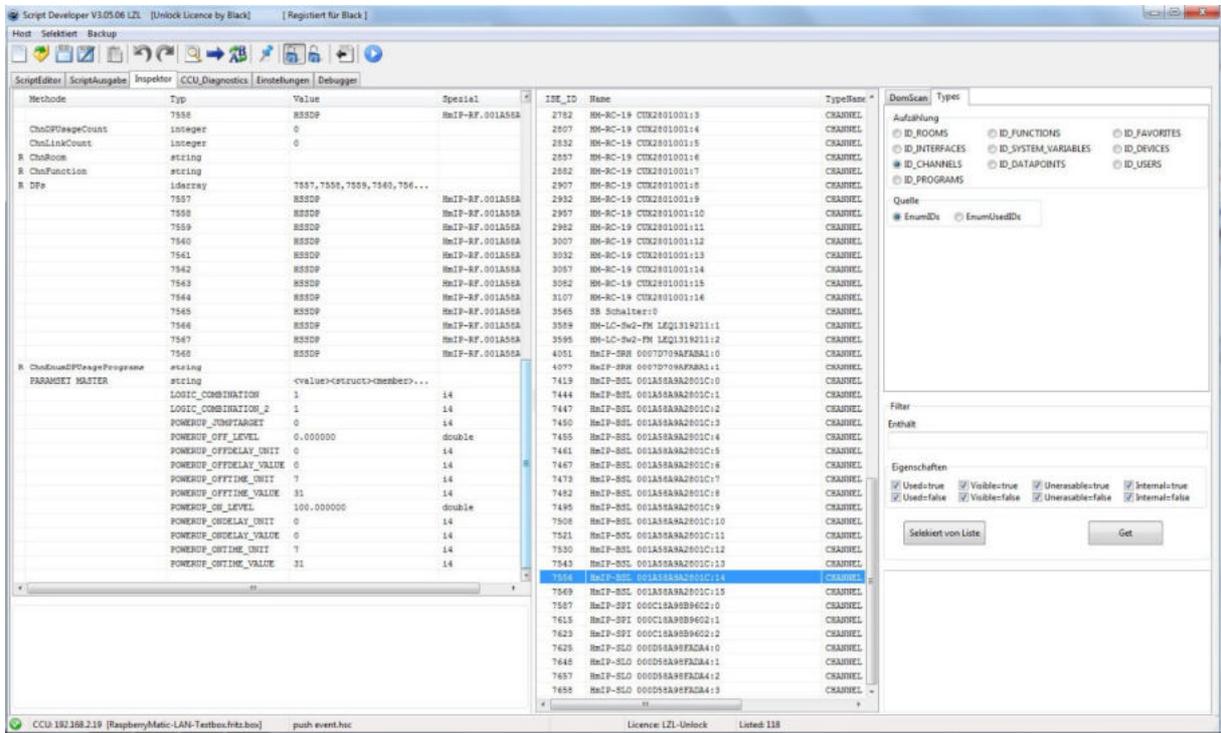
Kanal0 : DeviceName:0

Kanal1 : Devicename:1

Etc...

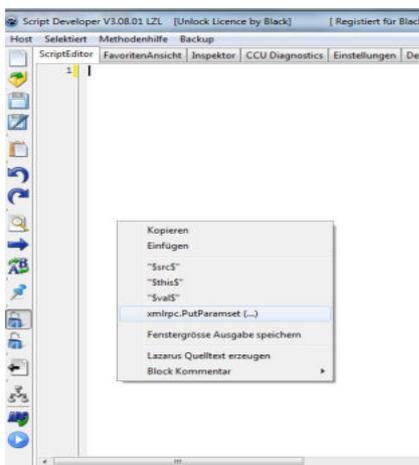
6.3 Paramset Master

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

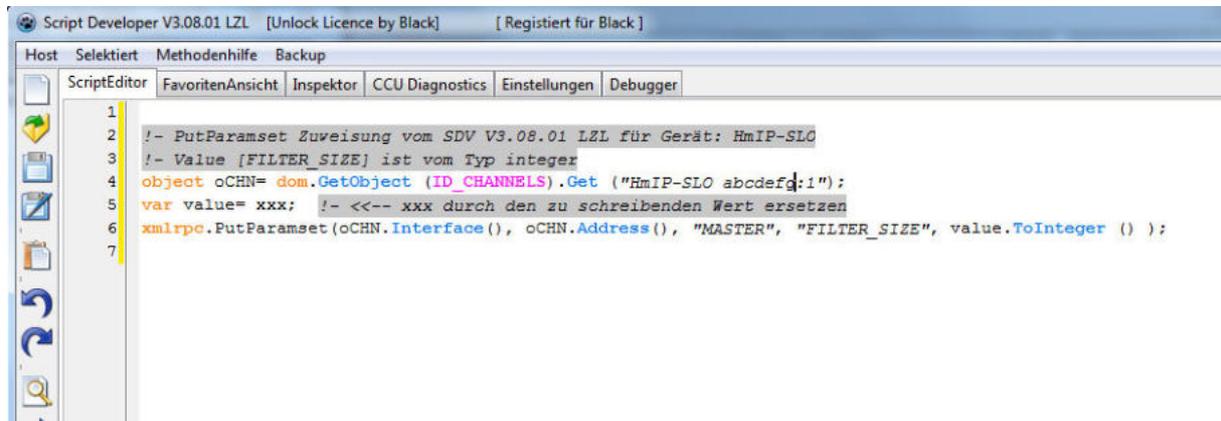


Die Werte lassen sich dann über PutParamset verändern in einem Script

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



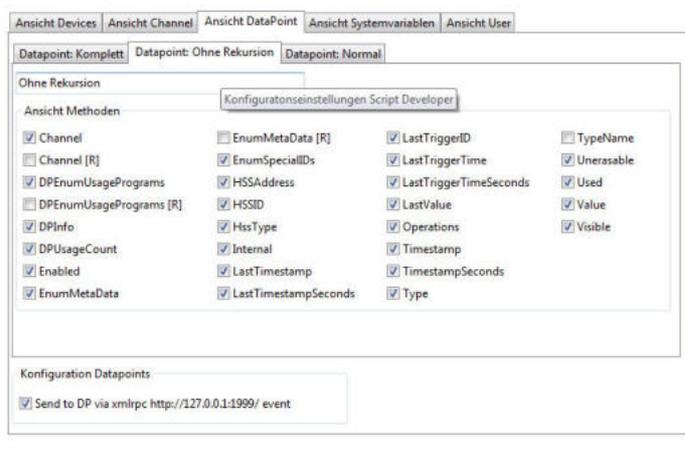
Dies erzeugt nun automatisiert den richtigen Code, in Abhängigkeit von: welches Gerät ist es (HMIP, HM, CUXD o.ä) und dem erwarteten Datentyp.



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


6.5 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:



Des weiteren braucht es in der Sicht Freigabe auf TypeName (SDV prüft auf HSSDP) und anklicken von Value. (Level 6 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)

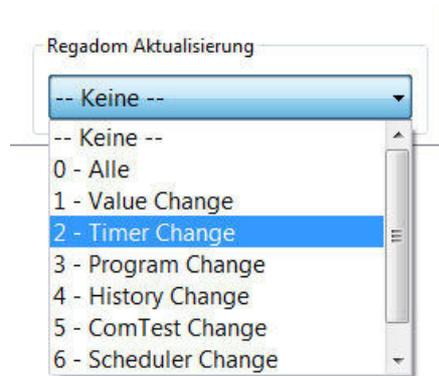
Dieses Feature ist extrem nützlich aus eigener Erfahrung zum Testen von Programmen, wenn man z.B. Testtrigger und ähnliches von Sensoren prüfen will. Man muss hier nicht z.B mit dem Fön auf einen Thermostaten blasen , damit dieser über 35 Grad geht, einfach den DP anklicken, Value auswählen und die Zahl ändern. Wert übernehmen.

7.1.4 CacheInfo

Darstellung des Cacheinhaltes der CCU

7.1.5 Regadom Aktualisierung

Anstossen von dom.RTUpdate (). Auswahl für die ComboBox



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

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.

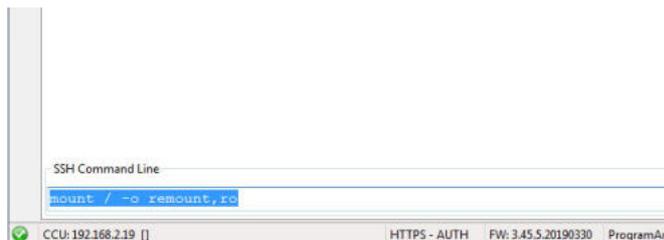
```
[CCU1]
IP=192.168.2.xx
USERNAME=xxxx
PASSWORD=yyyy
USEHTTPS=true
SSHUSERNAME=root Bestandsnutzer: diese schlüssel hinzufügen
SSHPW=sshpwort
```

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

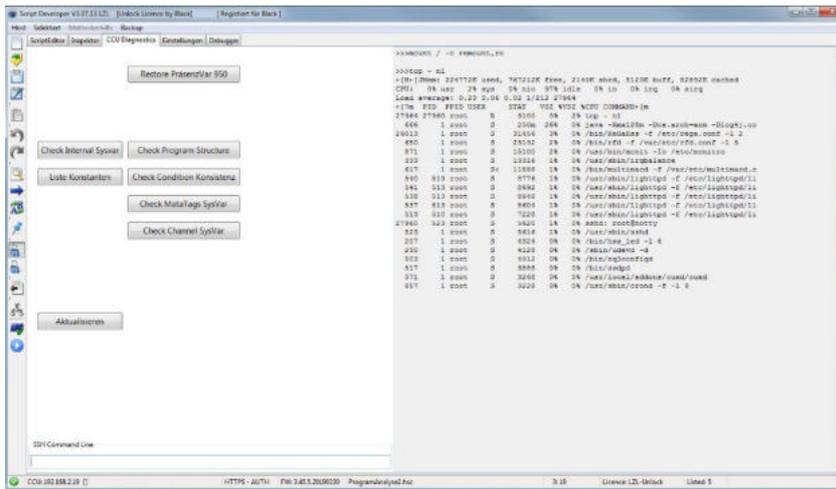
```
[HOSTCCU]
IP=192.168.2.xx
USERNAME=
PASSWORD=
USEHTTPS=False
SSHUSERNAME=root
SSHPW=
```

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

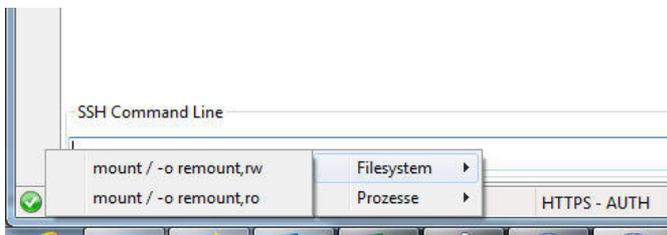
In dieser Zeile lassen sich nun einfache SSH Kommandos ausführen.



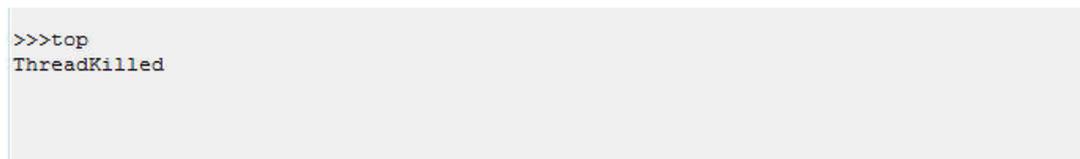
Dies soll und ist aber kein Ersatz für eine putty Konsole, eher gedacht für ich muss mal eben schnell einen Befehl ausführen oder man hat nicht die Befehlsfolgen im Kopf sondern benutzt die Hilfe. Programme dort zu starten oder z.b. top zusehen geht nicht und ist auch nicht geplant, das kann putty ganz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



7.1.1 SSH Realisierung (CUxD ab 2.3.1 oder plink.exe)

INI Schlüssel SSHPLINK=false

bei false versucht er den Zugriff über die CUXD Seite. Wenn der Zugriff nicht Freigeschaltet ist über

USERACCESS=1+

USERLOGIN=

Wichtig ist das + !!!!!!!!!!!!!

rennt er dabei gegen Access denied.

Das Kleingedruckte: USERACCESS=1+ ist Sicherheit wie alte Version, also keine.

Wer das nicht möchte, kann den damals beschriebenen alten Weg gehen, putty suite runterladen, plink exe ins SDV Verzeichnis kopieren, cmd unter Windows öffnen und einmal im SDV Verzeichnis ausführen:

```
plink -ssh root@IPderCCU -pw SSHPASSWORT top - n1
```

Wenn es der erste Aufruf war, die Frage nach dem Zertifikat bejahen. danach geht der Aufruf auch headless mit dem SDV.

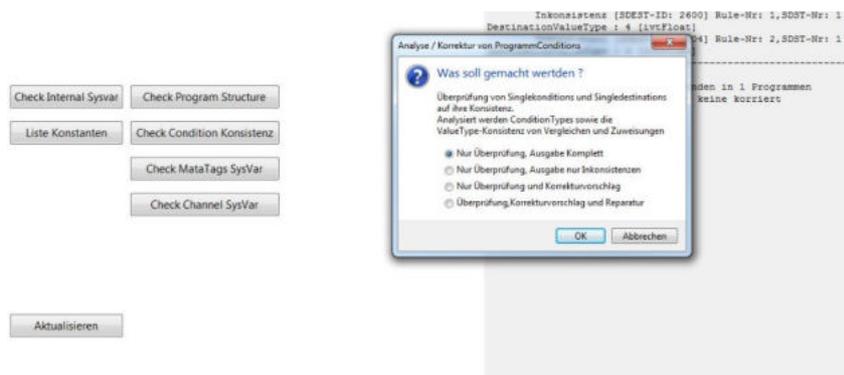
damit ist aber wieder Windows im Spiel.

Wer weder CuXD den Access setzen möchte noch plink benutzen will, der kann den SDV nutzen bis auf die SSH Fnkionalität.

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, wens 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 [ivtBinary]
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 korriert
```

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 in 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

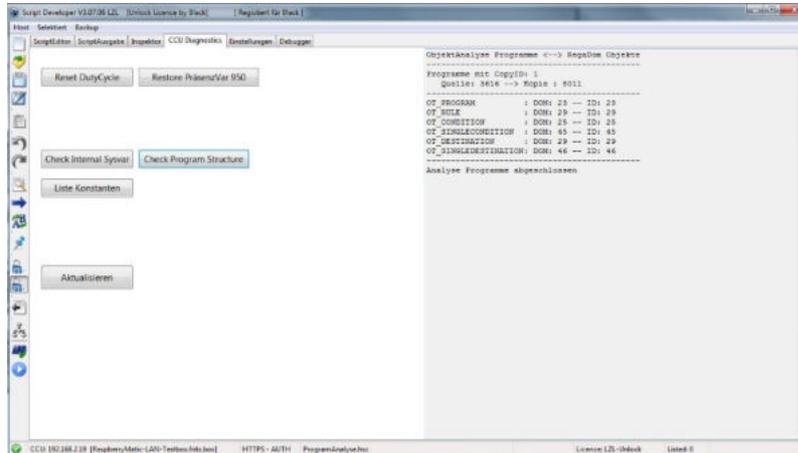
```
ProgrammAnalyse Konsistenz SingleConditions & SingleDestinations
Black in April 2019
-----
Analyse Program: Alarmtest [ID: 8528]
Inkonsistenz [SCND-ID: 8567] Rule-Nr: 1,CND-Nr: 1, SCND-Nr: 1 -- LeftVal: 2 [ivtBinary], RightVal: 4 [ivtFloat]
Ursache Boolvergleich ivtBinary Links mit nicht ivtFloat Wert rechts
Umrechnen von rechts Wert 15.000000 [ivtFloat] in true [ivtBinary]
Kann vom SDV korrigiert werden
Korrigiert durch RightVallValType (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 RightVallValType (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 korriert
```

7.3 Programmanalyse

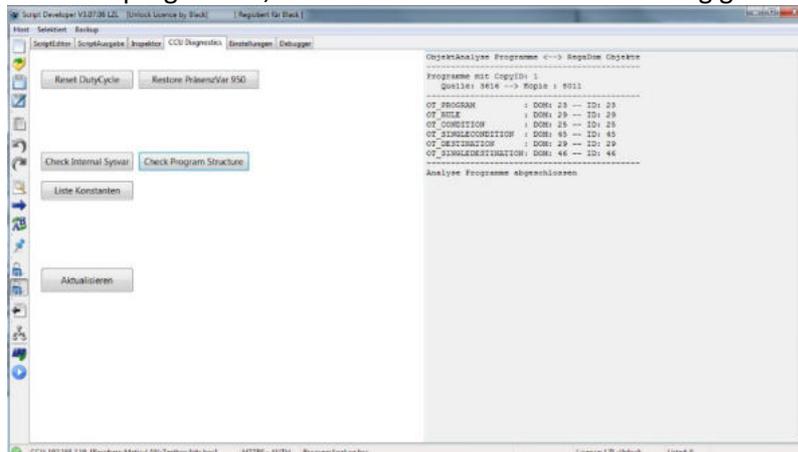
Mittels der Programmanalyse kann die Konsistenz der Programmobjekte überprüft werden. So sieht der Aufruf auf, wenn die Prüfroutine erfolgreich über die Programme gelaufen ist.



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.



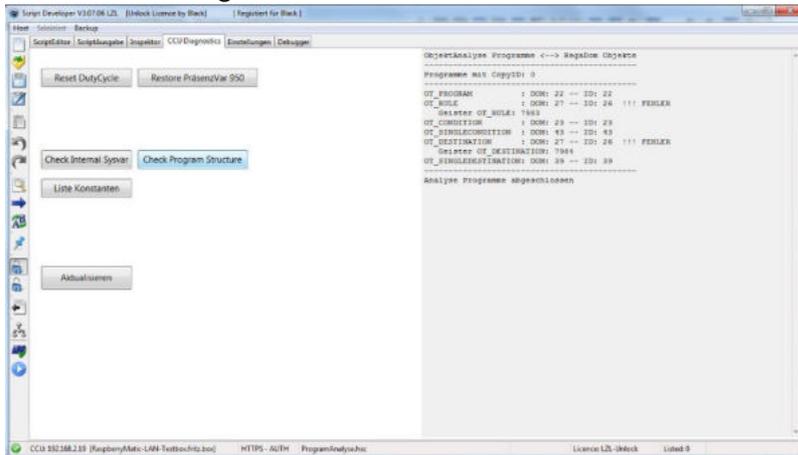
Hier wird das ursprüngliche Programm und die (Geisterkopie) daraus angezeigt. Der erste Versuche wäre: einen Reboot zu versuchen und zu überprüfen, ist der Eintrag des Geisterprogrammes weg oder nicht. wenn nicht, kann manuell versucht werden, die Objecte der Kopie (nicht des Quellprogrammes) zu löschen

2. Programmfragmentreste

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

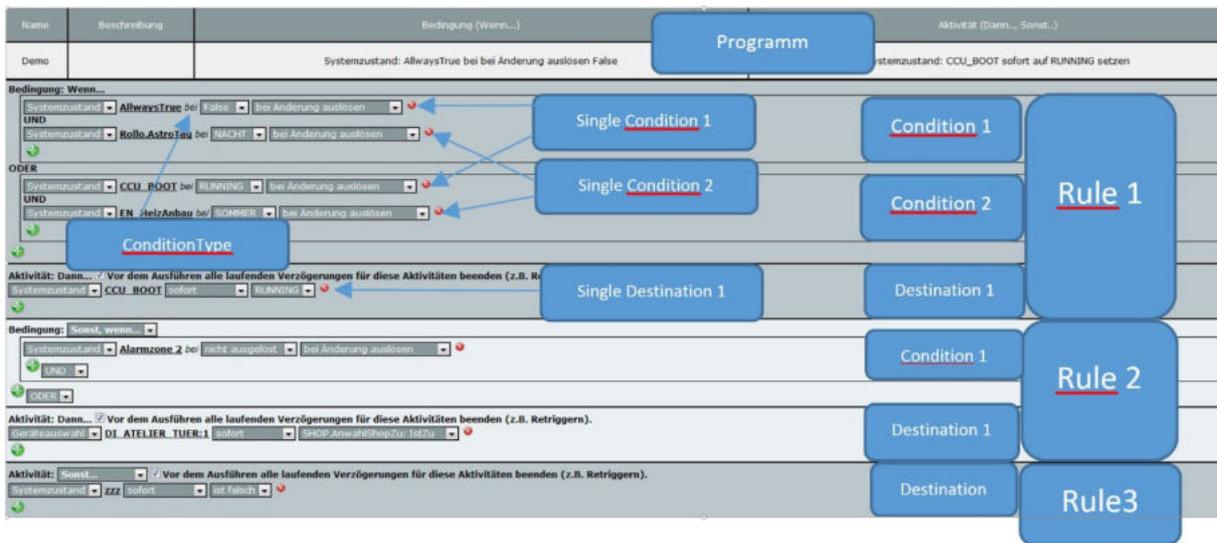
Provoziert habe ich es hier in dem Testfall, in dem ich ein RULE Object mal händisch angelegt habe.

Der Testlauf schlägt hier an:

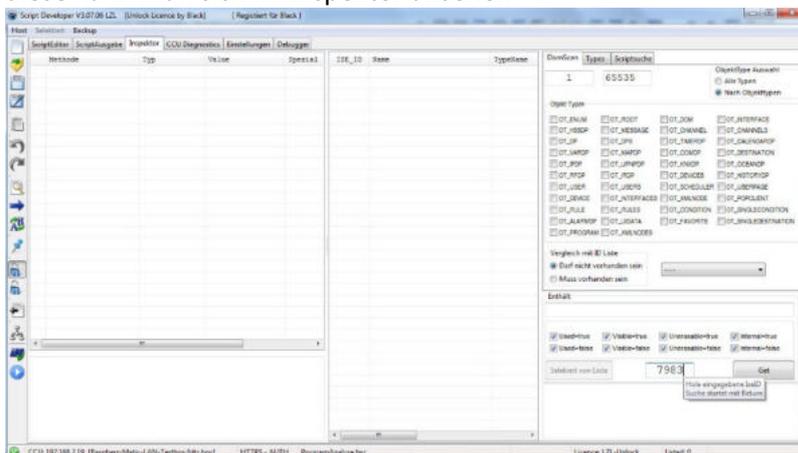


Es wird unter den Objecten OT_RULE und OT_DESTINATION angezeigt, dass sich in der Rega 2 Objekte tummeln, welche keinen Bezug zu dem Inhalt eines Programmobjektes haben. (2 deshalb, weil ein dom.CreateObject (OT_RULE) auch gleichzeitig das zu der Rule gehörende OT_DESTINATION Object erzeugt)

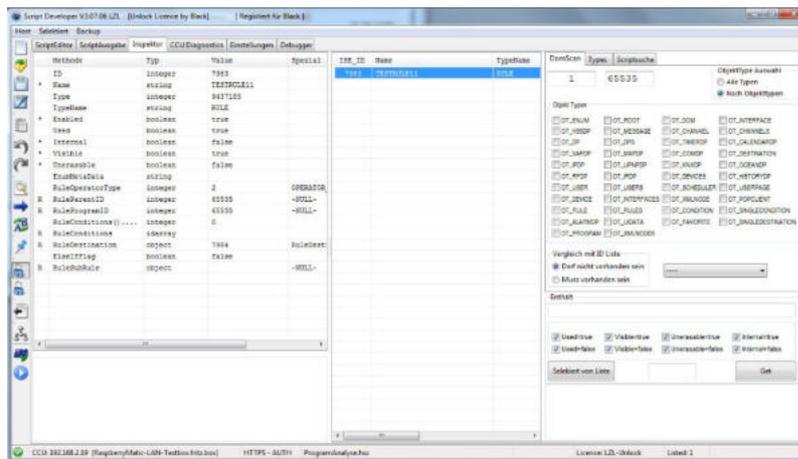
Ein Programm ist auf der CCU so aufgebaut:



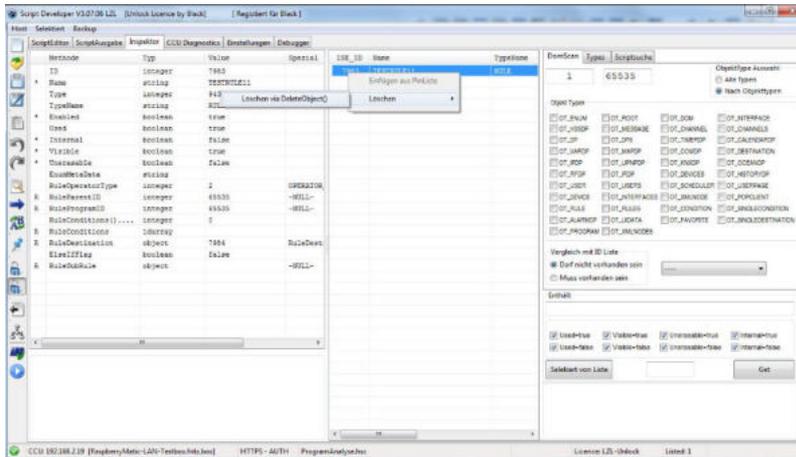
Die iselD des gefundenen Objektes wird ausgegeben diese kann man sich im Inspektor ansehen



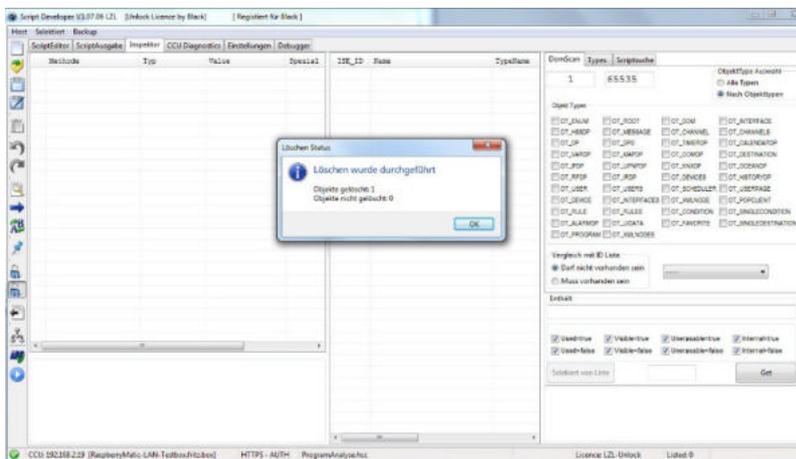
Dazu die gefundene ID in die Suchmaske eintragen und Enter drücken



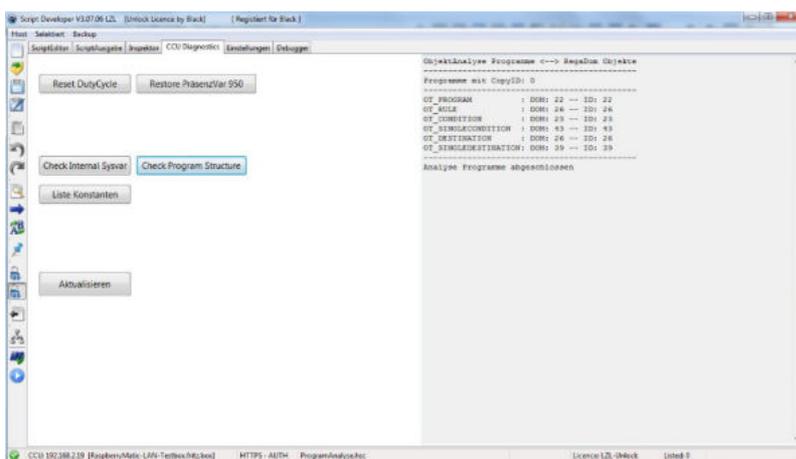
Man sieht, ein Object ohne Bezüge. es kann also entfernt werden. Löschen Freigeben, Programmunterobjekte müssen auch freigeben sein zum löschen, dann



es wurde geklickt



Die abschliessende Kontrolle zeigt:



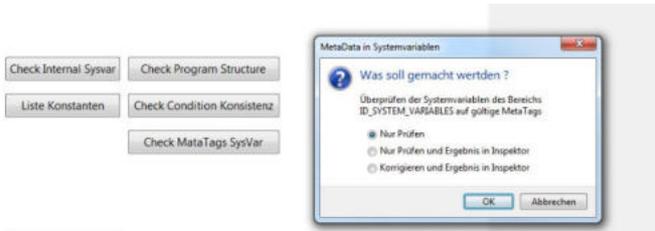
Es ist weg.

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:



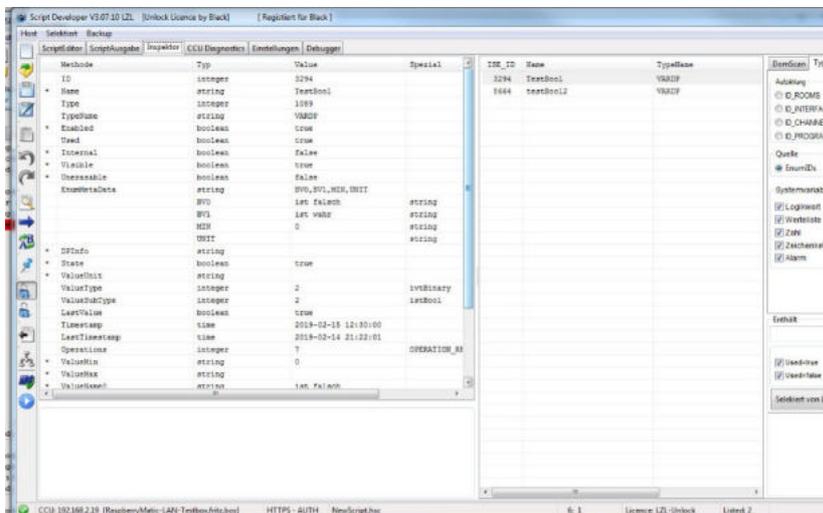
Punkt 1: nur Prüfen

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

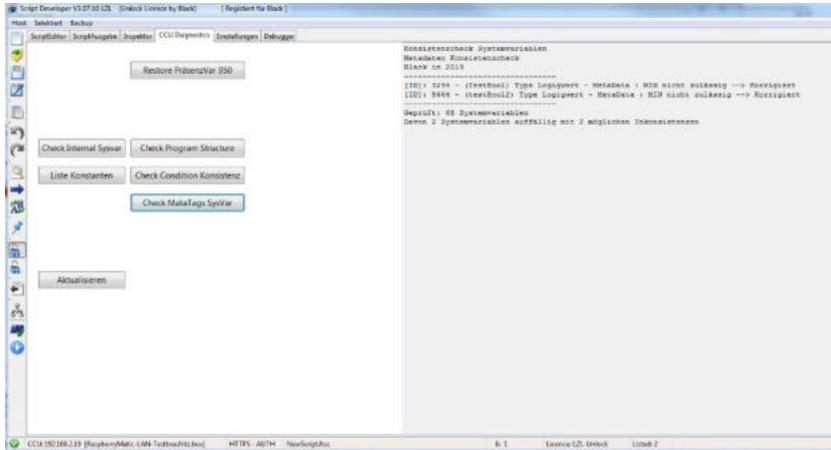
Genau wie Punkt 1, es wird eine Überprüfung gemacht, zusätzlich wird das Ergebnis noch in die Auswahlliste des Inspektors geladen. Dort kann überprüft und gegebenenfalls auch manuell korrigiert werden.



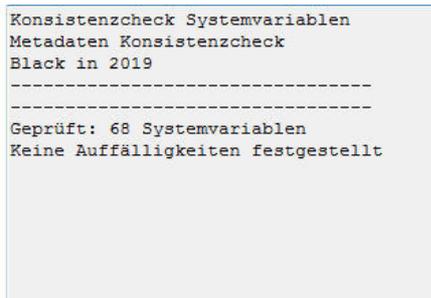
Punkt 3: Korrigieren und Ergebnis in den Inspektor

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

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

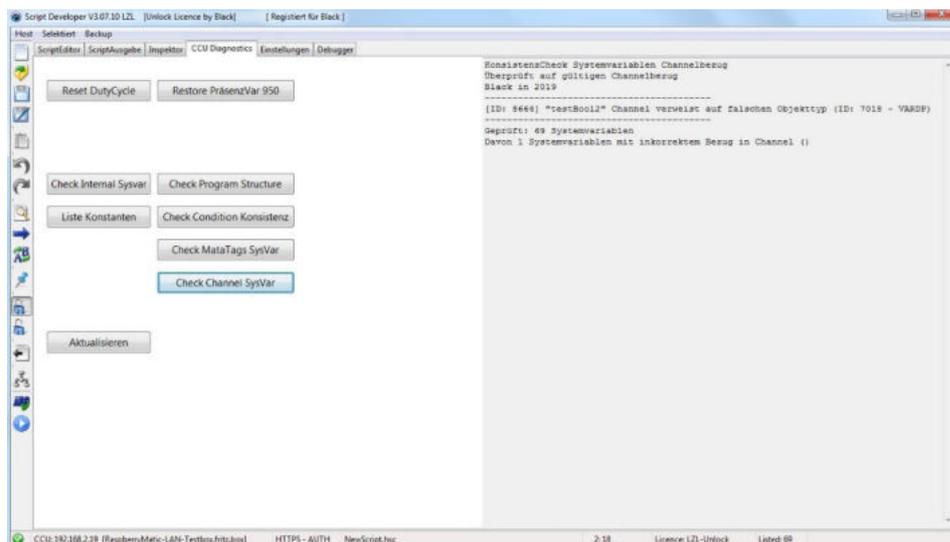
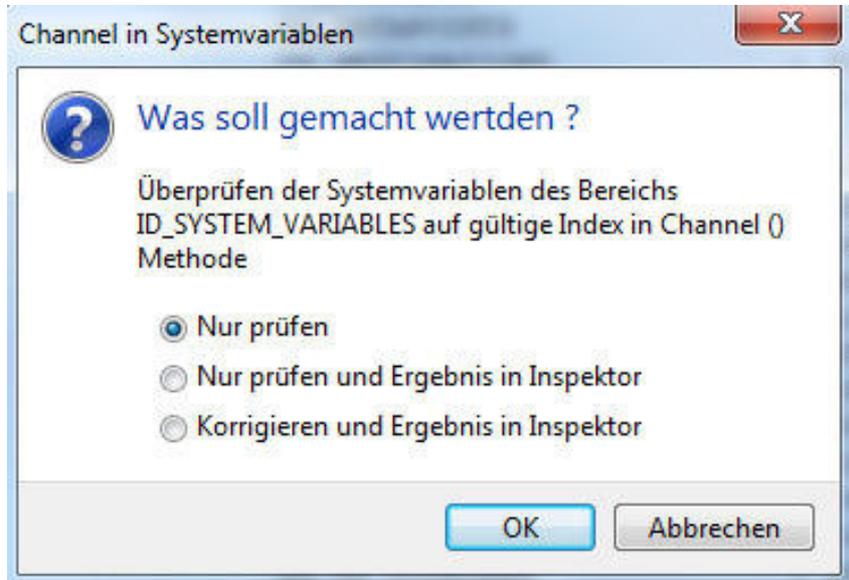


Ein nochmaliger Durchlauf bringt dann die Fehlerfrei-Nachricht.



7.5 Systemvariablen Test auf NirvanaChannelBezüge

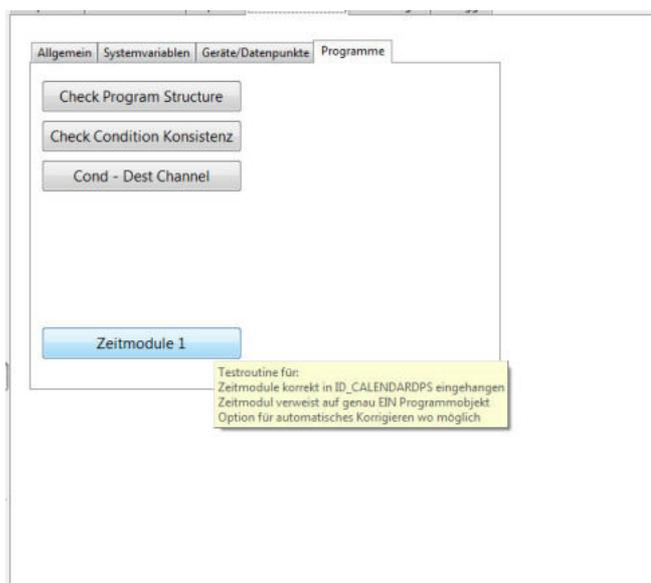
Es wird überprüft, ob die Methode Channel () entweder Leer ist (ID_ERROR) oder auf einen gültiges Channel Objekt verweist. Wenn nicht, handelt es sich um einen NirvanaChannelBezug. Hier auch: Auswahl was gemacht werden soll



Die Auswahl Korrigieren bereinigt dann diese automatisch und läd 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 inkorrektem Bezug in Channel ()
```

7.6 Test und Korrektur Zeitmodule



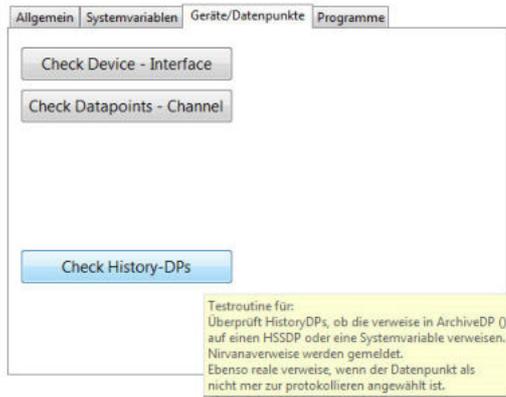
Seit Version 3.08.10 ist das Menü CCU Diagnostics in Reitern strukturiert. Unter Programme findet sich nun die Testoption Zeitmodule 1 . Hierbei wird geprüft auf korrektes Einhängen des Zeitmodules in ID_CALENDARPS, ob jedem Zeitmodul genau ein Programmobjekt zugeordnet ist. Korrektur kann da erfolgen wo möglich. Auswahl erfolgt nach anklicken des Buttons „Zeitmodul 1“



7.7 Test und Korrektur HistoryDPs.

HistoryDPs sind eine wunderbare Gelegenheit um immensen Datenmüll in der RegaDOM zu hinterlassen. So wird beispielsweise für bei Anwahl eines Channels auf protokolliert für jeden einzelnen Datenpunkt des Channels ein HistoryDP angelegt. Wird das gerät irgendwann mal gelöscht oder die Aufzeichnung deaktiviert, so bleibt – richtig gedacht – der HistoryDP erhalten. Kann man so machen, muss man aber nicht.

Mit dem Punkt



Lassen sich verwaiste oder auch falsche HistoryDPs aufspüren. Die Art der Bearbeitung wird beim Anklicken festgelegt.



Nur Anzeige:

Listet die gefundenen „Inkonsistenzen“ rechts im Ausgabefenster auf. Es wird aber noch nix korrigiert oder verändert

Anzeige und List im Inspektor

Listet die gefundenen „Inkonsistenzen“ rechts im Ausgabefenster auf. Zusätzlich werden die gefundenen IDs in Listauswahl des Inspektors geladen und können dort mit dem Inspektor untersucht werden. Es wird aber noch nix korrigiert oder verändert.

Anzeige und Löschen

Wie unter Punkt 1 werden die gefundenen Inkonsistenzen rechts im Ausgabefenster gelistet, gleichzeitig wird korrigiert (nicht referenzierte DPs werden gelöscht).

Lauf alles in Ordnung:

```
Analyse History DPs
Black in August 2019
-----
Es wurden 11 HistoryDPs überprüft
Dabei wurden keine Auffälligkeiten festgestellt
```

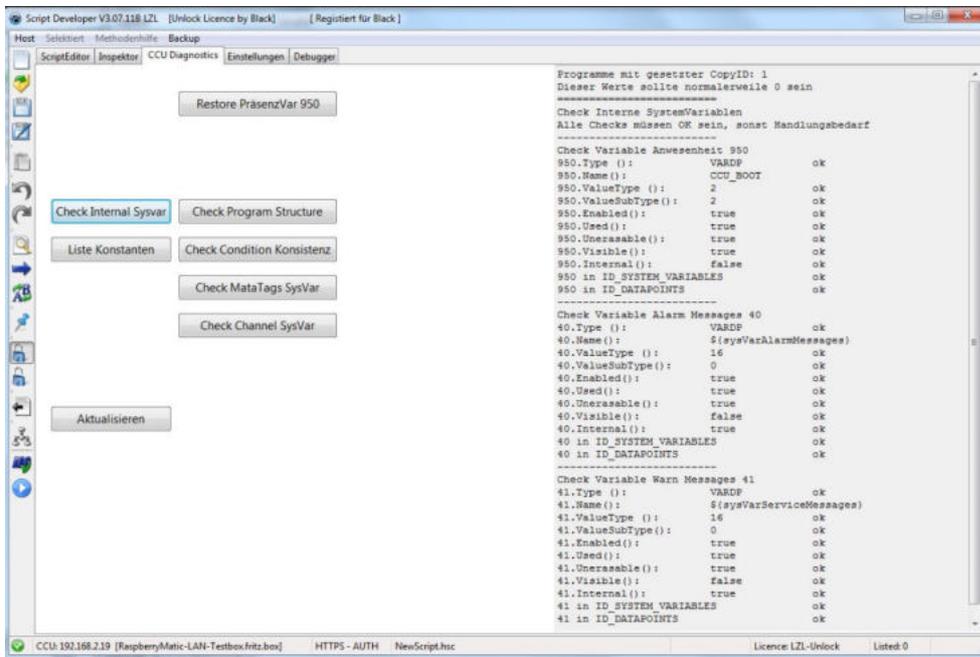
Ein HistoryDP verweist auf eine Systemvariable, bei der Protokollierung allerdings abgewählt wurde:

```
Analyse History DPs
Black in August 2019
-----
HistoryDP 11208 verweist in ArchiveDP [6461] auf Systemvariable mit DPArchive ()=false. Empfehlung löschen
-----
Es wurden 12 HistoryDPs überprüft
Dabei wurden 1 Inkonsistenz(en) festgestellt
Davon wurden 0 Inkonsistenz(en) korrigiert
```

Wenn Korrektur angewählt würde, würde der HistoryDP gelöscht werden vom SDV.

7.8 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.

7.8 Check Objects

Testroutine für:

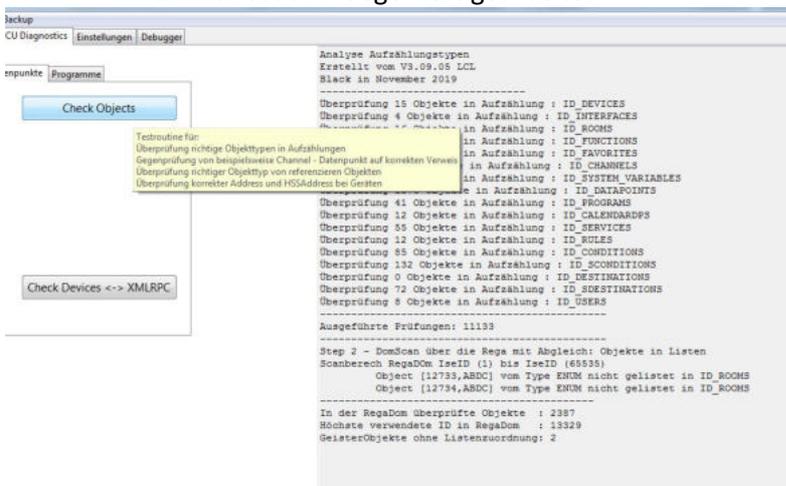
Überprüfung richtige Objekttypen in Aufzählungen

Gegenprüfung von beispielsweise Channel - Datenpunkt auf korrekten Verweis

Überprüfung richtiger Objekttyp von referenzieren Objekten

Überprüfung korrekter Address und HSSAddress bei Geräten

Es wird immer mit Test und Gegentest gearbeitet



7.9 Check Devices <-> XMLRPC

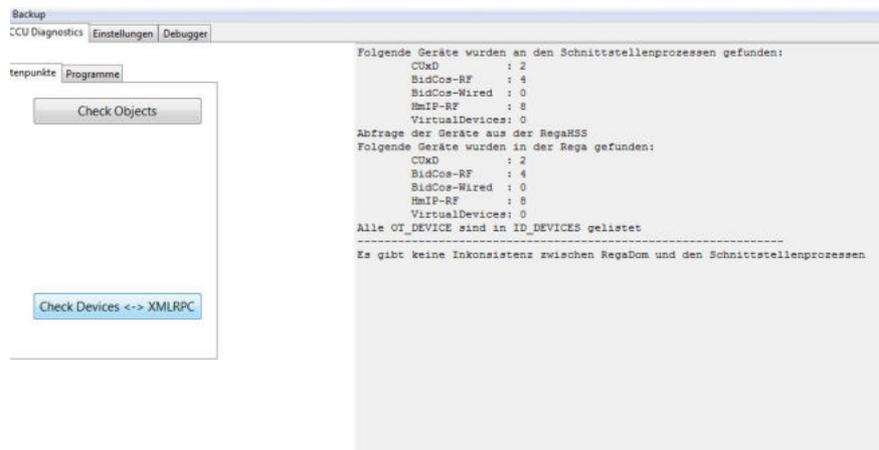
Folgendes wird überprüft:

Alle Geräte mit gültigem Interface müssen auch in ID-Devices gelistet sein.
Jedes dieser Geräte muss sich auch im Schnittstellenprozess wiederfinden.

Aufgelistet werden:

- Geräte, die Im Schnittstellenprozess vorhanden sind aber nicht in der Rega
- Geräte, die in der Rega vorhanden sind aber nicht im Schnittstellenprozess
- Geräte die nicht in ID-Devices auftauchen

Check Devices <-> xml neu eingeführt. Dieser Lauf führt einen Vergleich der Rega gegen die Schnittstellenprozesse durch. Damit lässt dich auch ein Gerät finden, dass nicht mehr in der Rega ist aber noch in den Schnittstellenprozessen hängt. Eine Reparatur wird nicht automatisch gemacht, aber der Weg vorgeschlagen (der beginnt immer mit einem VollBackup). Das vor einiger Zeit mal CuxD GeisterDevice liess sich damit auch finden.



7.10 Check Devices <-> XMLRPC

Testroutine für:

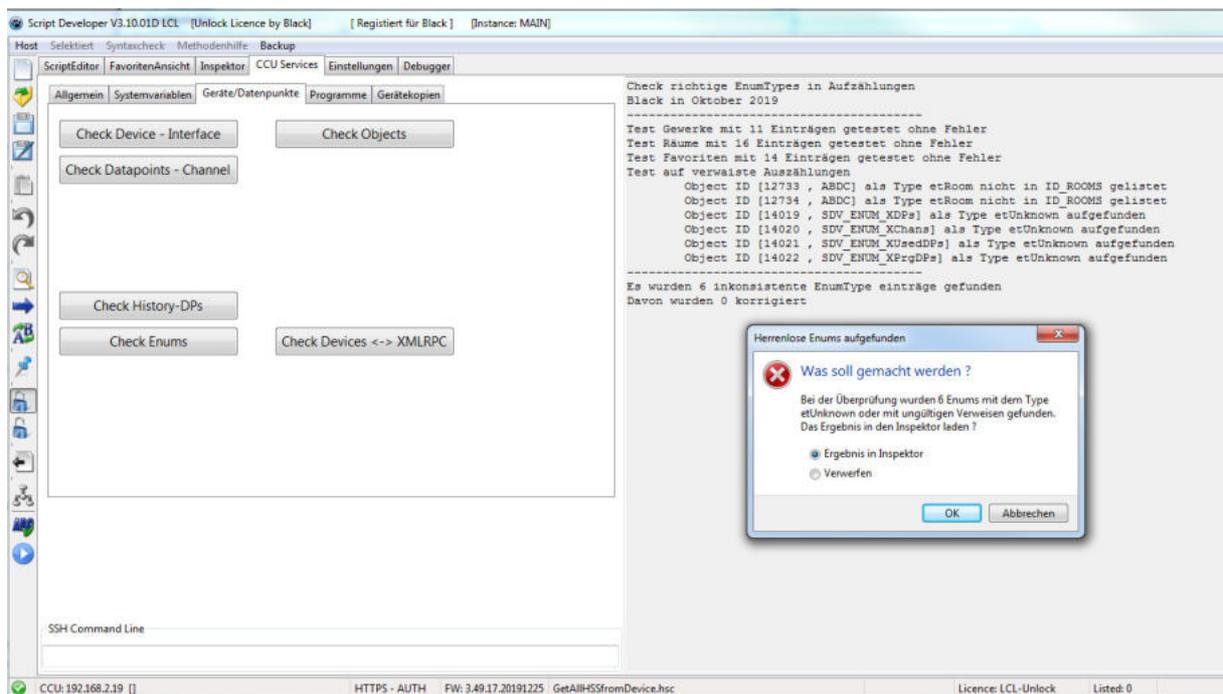
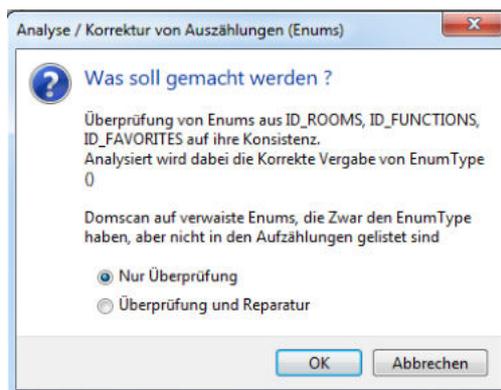
Überprüft Enums auf folgende Konsistenz:

a:) EnumTypes ()

1. ID_ROOMS müssen Type etRoom
 2. ID_FUNCTIONS müssen Type etFunction
 3. ID_FAVORITE müssen Type etFavorite haben. Korrektur ist möglich
- b:) DomScan nach Enums gefundene Enums
1. etRoom müssen in ID_ROOMS
 2. etFunction müssen in ID_FUNCTIONS
 3. etFavorite müssen in ID_FAVORITE gelistet sein

Die Liste hierbei gefundener Inkonsistenter Enums wird in das Listenfeld des Inspektors geladen

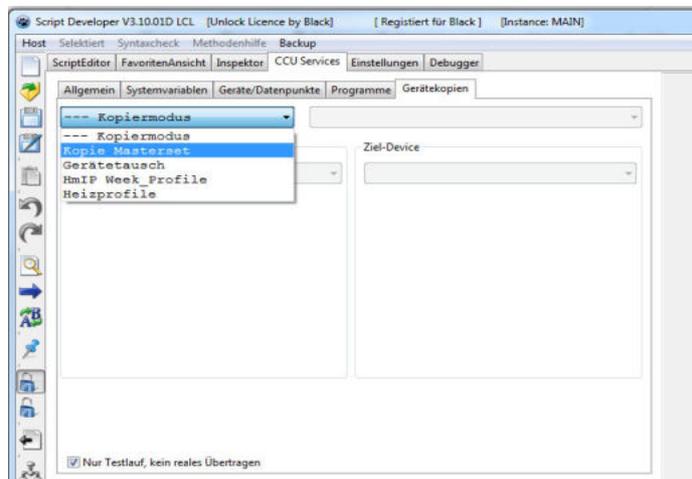
Auch hierbei wieder das Auswahlfeld zur Bestätigung was durchgeführt werden soll:



Es folgt die Abfrage, ob das Ergebnis in den Inspektor geladen werden soll zur Überprüfung / Reparatur oder ob das Ergebnis verworfen werden soll

8 Gerätekopieren

Unter diesem Reiter in der Hauptkategorie CCU Services verbirgt sich ein mächtiges Tool, um Geräteeinstellungen von einem Gerät in ein anderes zu kopieren bzw Geräte auszutauschen. Im gegensatz zur originalen WebUI arbeitet der SDV dabei auch mit ähnlichen (nearEqual) Geräten und auch mit HmIP Geräten.



1. Kopie Masterset: kopier die Geräteeinstellungen eines Gerätes in ein Baugleiches oder ähnliches Gerät
2. Gerätetausch: noch nicht implementiert in 3.10.02, kommt noch
3. HmIP Week_Profile kopiert die Wochenprofile aus dem entsprechenden Kanal eines HmIP Aktors in einen anderen HmIP Aktor
4. Heizprofile: kopiert die Heizprofile von Hand / Heizkörperthermostaten bzw Heizgruppen selektiv untereinander, auch zwischen HMclassic und HmIP Geräten

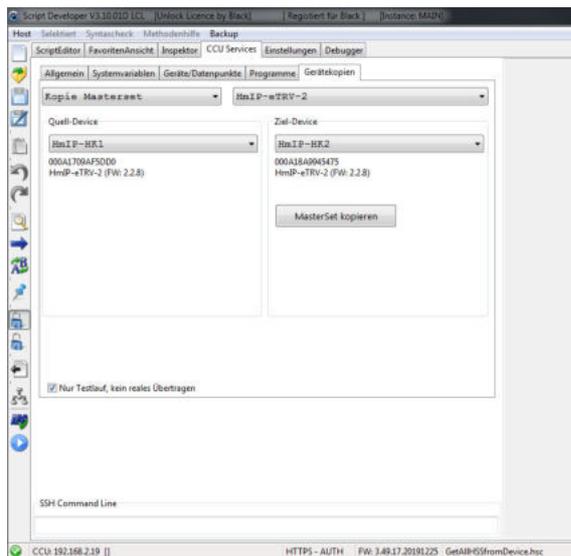
Ähnliche Geräte sind im Source Code Hardcoded, momentan sind folgende Gruppen angelegt:

```
NearEqualMASTER:= tNearEqualDevice.Create ();
NearEqualMASTER.Add ('HM-LC-B11-FM'#9'HM-LC-B11PBU-FM'#9'HM-LC-B11-SM');
NearEqualMASTER.Add ('HM-LC-Sw1PBU-FM'#9'HM-LC-Sw1-FM'#9'HM-LC-Sw1-DR');
NearEqualMASTER.Add ('HM-Sec-RHS'#9'HM-Sec-RHS-2');
NearEqualMASTER.Add ('HM-Sen-MDIR-O-3'#9'HM-Sen-MDIR-O-2'#9'HM-Sen-MDIR-O'#9'HM-Sen-MDIR-SM');
```

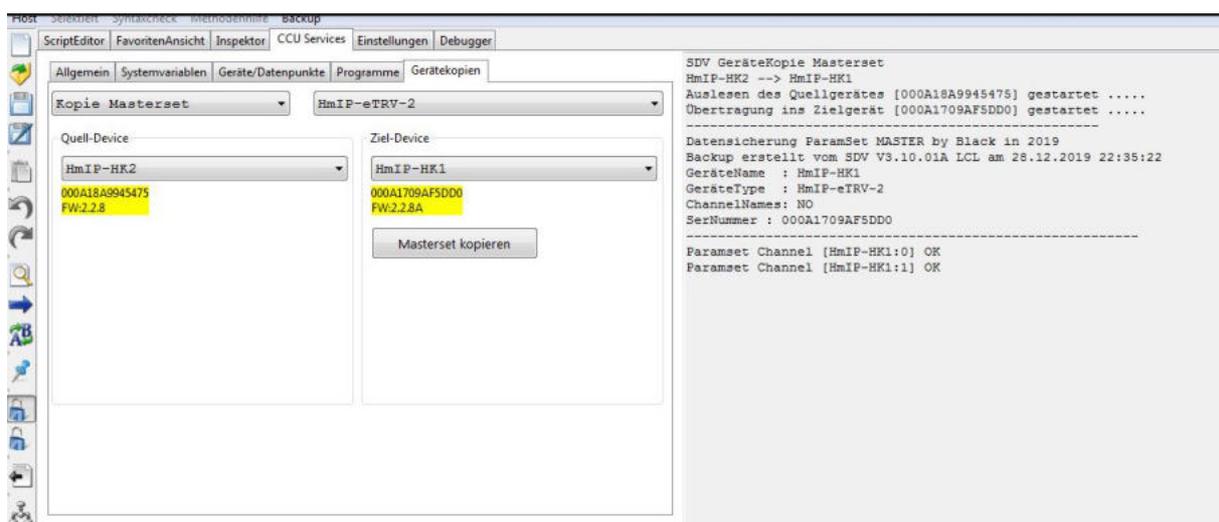
8.1 Kopieren Masterset

Als Erstes unter CCU Services - Gerätekopieren den Kopiermodus einstellen Dann im nächsten Fenster die Geräteart auswählen, unter der kopiert werden soll. Es wird vom SDV eine Auswahlliste generiert, wobei immer mindestens 2 gleichartige Geräte vorhanden sein müssen. (sonst wäre es auch schlecht mit kopieren). Dann das Quell und das Zielgerät selektieren. und Masterset Kopieren drücken. daraufhin werden die Einstellungen von dem Quellgerät in das Zielgerät kopiert.

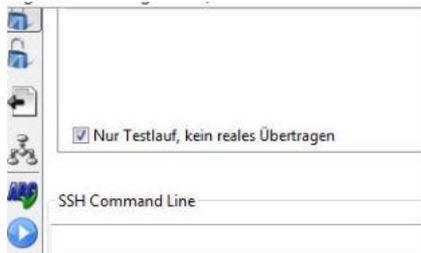
Unter Gerätetype werden die Gerätetypen vorgeschlagen, die mindestens 2 mal vorhanden sind, dann daraufhin das Ziem und das Quelldevice auswählen.



Sollten beide Geräte nicht identisch sein oder bei identischen Geräten eine unterschiedliche Firmware vorliegen, so wird das Infocfeld gelb hinterlegt. Letzte Entscheidungsinstanz ist dann der Mensch



Angehakt gibt es nur Testlauf, ohne Haken wird ausgeführt... Standardmäßig ist bei Programmstart der Haken gesetzt. Es werden bei der Generierung Quertests auf gegenseitige Existenz und Typgleichheit der einzelnen Keys laufen lassen. Existiert ein Key des Zieldevices nicht im Quelldevice oder ist ein Typ nicht gleich, so wird der Key samt Value des Zieldevices beibehalten und nicht verändert. Es erfolgt dann aber auch eine Meldung. ebenso wenn die Kanalnumerierung in beiden Geräten nicht passig ist. (sollte aber passen normalerweise, weil Geräte mit unterschiedlichen Lasse ich nicht als Paar zu , hardcoded)



Mögliche Fehlerbilder:

Hier existiert ein Key im Zieldevice, aber nicht im Quelldevice: Meldung

```
SDV GeräteKopie Masterset
HmIP-HK1 --> HmIP-HK2
Übernehmen 21 Values aus Quelldevice in HmIP-HK2:0
  Key ARR_TIMEOUT existiert nicht
Übernehmen 571 Values aus Quelldevice in HmIP-HK2:1
```

Ein Channel lässt sich nicht zuordnen:

```
SDV GeräteKopie Masterset
HmIP-HK1 --> HmIP-HK2
ChannelNummer zu (HmIP-HK2:0) existiert nicht im QuellDevice
ChannelNummer zu (HmIP-HK2:1) existiert nicht im QuellDevice
```

8.2 Gerätetausch

In Version 3.10.02 noch nicht implementiert

8.3 Kopieren Week Profiles

Im Menü IP WeekProfile stehen alle IP Geräte mit einem WEEK_Profile Channel zur Auswahl, hier lassen sich nun Quell und Zieldevice auswählen.

Sind beide Typ mäßig nicht gleich oder unterscheiden sich in der Kanalnummer der WEEK_Profile Channel oder haben unterschiedliche Anzahlen virtueller Kanäle so werden die Infopfelder gelb hinterlegt.. Letzte Instanz Mensch.

Wird trotzdem übertragen gedrückt läuft folgende, erweiterte Logig:

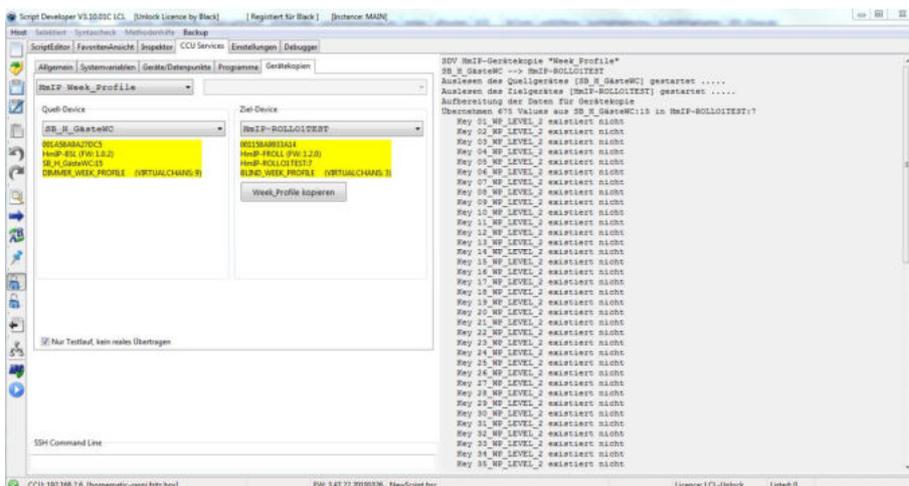
prüfen ob es den Key in dem Quelldevice gibt: wenn nein, Meldung und der alte Wert im Zieldevice bleibt erhalten. Hier in Bild zu sehen, ein BSL hat im Gegensatz zu einem FROLL keinen LEVEL_2 drum die Menge an Meldungen.

dann Level: Ein Switch kennt nur 0 oder 1 ein Dimmer 0 bis 1. ist das Ziel ein Switch, werden die Werte uminterpretiert: alles außer 0 wird im Switch zu einer 1. also Dimmer etwas an bedeutet beim Überspielen in einen Switch: Switch an.

Bei den Targetkanälen wird ausmaskiert : hier auch zu sehen, der BSL hätte 9 virtuelle Kanäle, der FROLL 3. es wird beim Übertragen runtermaskiert auf die Kanäle des Zielddevices..

Der SDV findet die Geräte selbstständig und schlägt diese dann im Auswahlmenü vor. ebenso wird der WEEK Profile Kanal automatisch gefunden sowie die Anzahl der VirtualChans für die mask ermittelt.

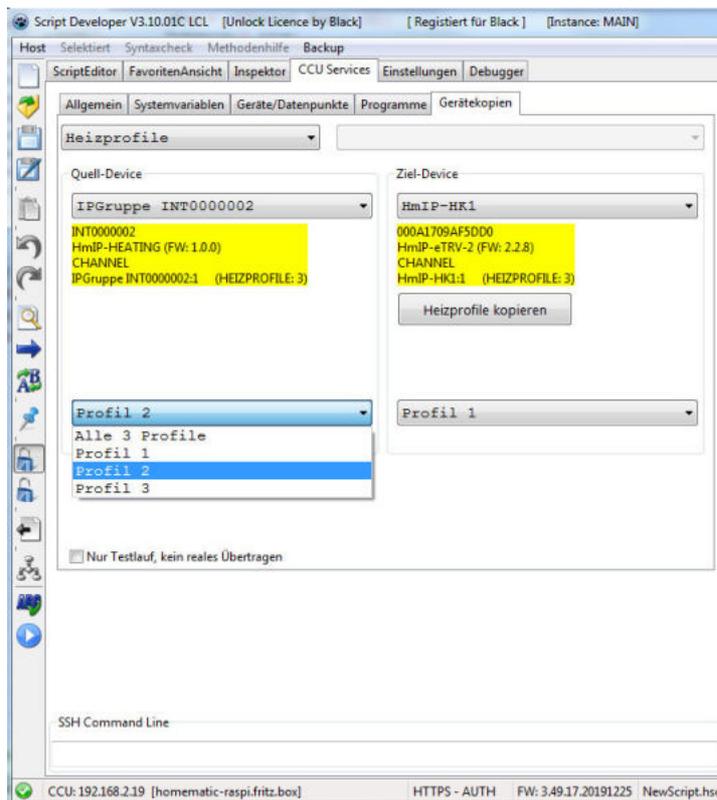
Es können dabei die Wochenprofile (mit den o.g. Einschränkungen) auch unter unterschiedlichen IP Geräten kopiert werden.



8.4 Kopieren Heizprofile

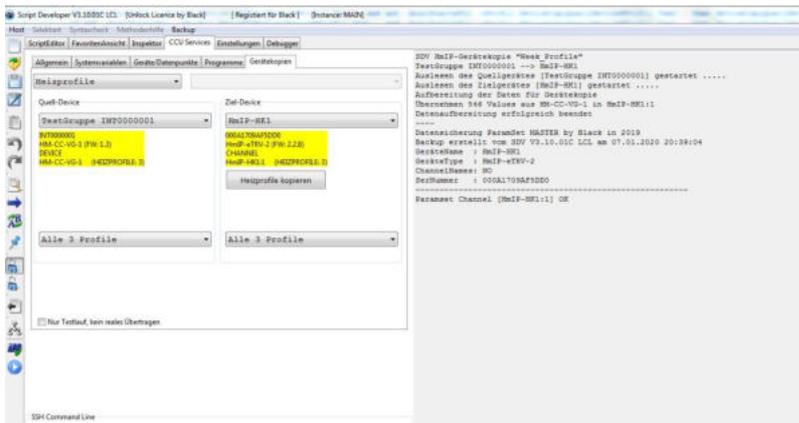
Es können Profile selektiv auch zwischen verschiedenen Gerätefamilien kopieren werden, also z.B. von IPgruppen in IP Thermostate, oder von IP Thermostate in Classic Thermostate, oder von einer Classic Gruppe in einen IP Thermostaten. Bei mehreren Profilen in einem gerät können alle Profile oder auch selektiv kopiert werden, es geht also auch: von Quellgerät Gruppe 1 in Zielgerät Gruppe 2

Bei Auswahl der Geräte wird ermittelt, welche und wieviele Profile vorhanden sind und diese dann in der Auswahl angeboten

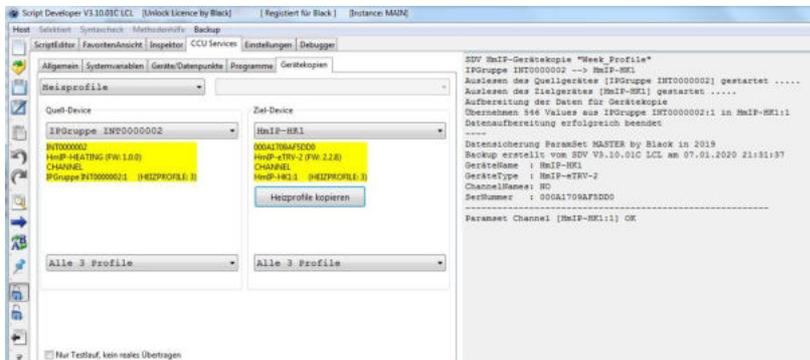


Zwischen allen Geräten classic HK Thermostat, classic WT Thermostat, classic Heizgruppe sowie zwischen IP HK Thermostaten verschiedener Ausführungen, dem IP Wandthermostaten und der IP Heizgruppe können nun wild die Heizprofile gerätegruppenübergreifend hin und her kopiert werden.

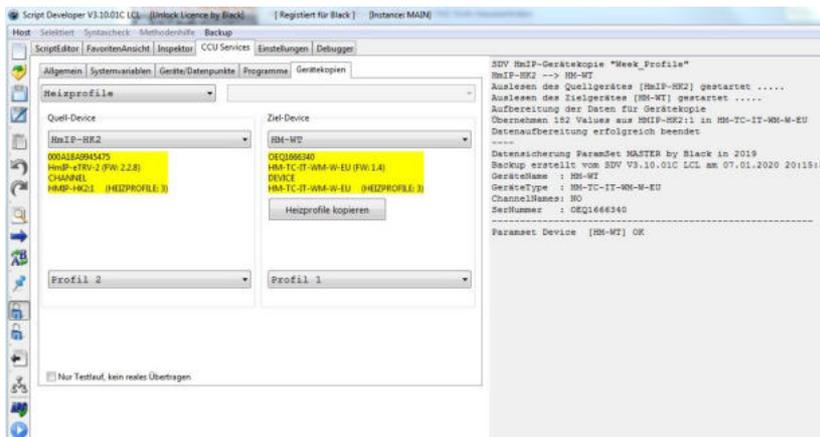
Beispiel: Kopieren aller 3 Profile einer classic Heizgruppe in einen IP HK-Thermostat



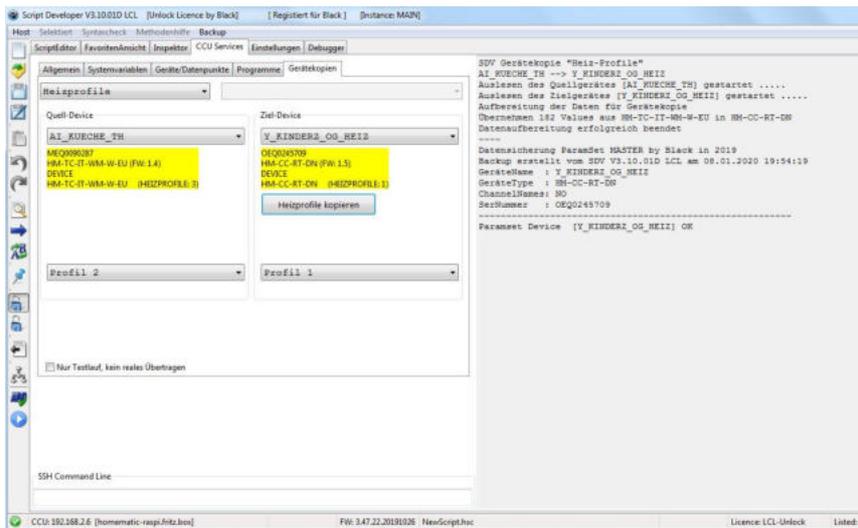
Beispiel: Kopieren alle Profile einer IP Heizgruppe in einen IP Wandthermostaten



Beispiel: Kopie Profil IP-HK Thermostat Profil 2 in classic Wandthermostat Profil 1



Beispiel: kopieren aus einem WT classic in deinen HK classic, der ja nur ein Profil hat



9 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,
 - * lhash, 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.]
- */

9.1 Compiler

Programmiert wurde der SDV unter Lazarus unter Win64

