

Kurzanleitung Scriptdeveloper V3.06.04 (27. Dezember 2018)

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

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

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

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

Dies ist eine BetaTestversion.

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

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

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

Inhalt

Kurzanleitung Scriptdeveloper V3.06.04 (27. Dezember 2018)	1
1. Installation	4
1.1 Lizensierung	5
1.2 Systemvoraussetzungen	9
1.3 Was tut's bis jetzt	9
1.4 Bekannte Einschränkungen / Bugs	10
1.6 Changelog	11
1.6.1 Changelog 03.06.04 LZL	11
1.6.2 Changelog 03.06.01 LZL	11
1.6.3 Changelog 03.05.06 LZL	11
1.6.4 Changelog 03.05.01 LZL	11
1.6.5 Changelog 03.04.01 LZL	11
1.6.6 Changelog 03.03.01 LZL	12
2. Oberfläche	13
3 Scripteditor	14
3.1 Voreinstellungen Editor	15
3.2 Vervollständigen Funktion	16
4 Inspektor	17
4.1 Selektionswahl: DomScan	18
4.2 Selektionskriterium Types	21
4.3 Zusätzliche Selektionsbedingungen	22
4.4 Daten aus Inspektor in Editor übernehmen	29
4.4.1 Mehrfachauswahl als Enum String	31
4.4.2 Übernahme von einem Script aus einem Programm direkt in den Editor	33
4.5 Selektion von Selektion	36
4.6 Objekte löschen	37
4.7 Anwenderdefinierte Sichten	39
5 Backups	43
5.1 Räume	43
5.2 Gewerke	43
5.3 Systemvariablen	44
5.4 Devices und Kanäle	44
6 Kleine Helfer im Alltag	45
6.1 Umbenennen von Kanälen von Geräten	45
6.2 Paramset Master	46

6.3.....	46
7 Diagnosebild	48

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

[HOST]
CCU=192.168.2.19
NICKNAME=Benutzername           ← anpassen Benutzername / Nick
CCU1=192.168.2.XX              ← IP Adresse der 1. CCU
CCU2=192.168.2.XX              ← Wenn vorhanden, IP der 2. CCU
CUXD=CUXD.CUX2801001:5         ← CUxd KANAL (ist nötig, dafür braucht es kein pscp mehr)
LICENCE1=                        ← Lizenzschlüssel für 1. CCU
LICENCE2=                        ← Lizenzschlüssel für 2. CCU

[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
```

BestandsNutzer:

Die Ini Datei hat sich ziemlich vergrößert. Hilfreich ist, in der mitgeliefertenINI alle Schlüssel ab
[CFG_ChanView1_Methods]

Address=True.....

Und in die bestehendeINI Datei anzuhängen. Erspart eine Menge manuellerHakensetzen.

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

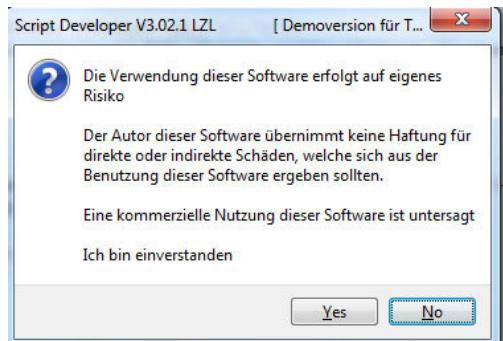
1.1 Lizenzyierung

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

Wie arbeitet das ?

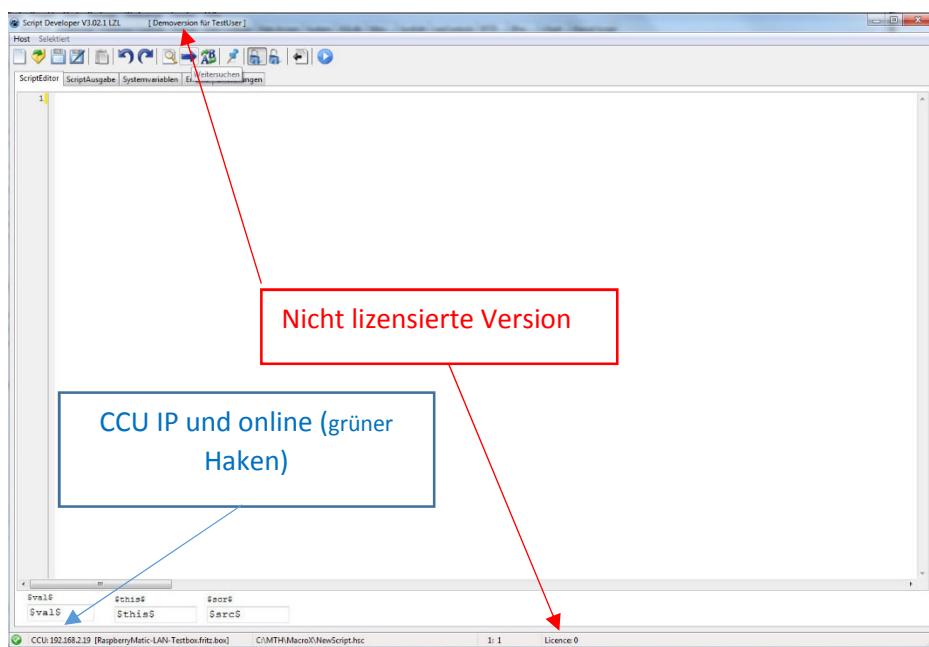
Der SDV telefoniert nicht nach Hause. Um eine Lizenz anzufragen ist folgender Weg einzuschlagen.

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

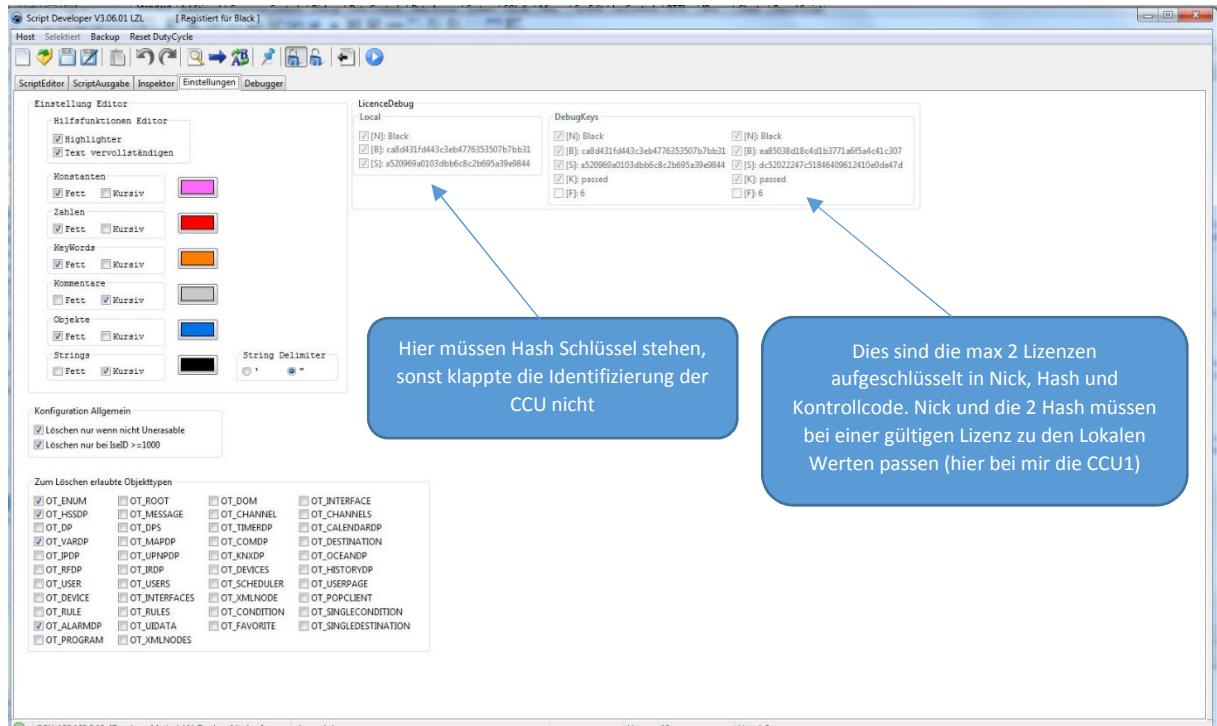


Bei allerersten Start muss dieses Fenster mit yes bestätigt werden. No führt so einem sofortigen Programmabbruch

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



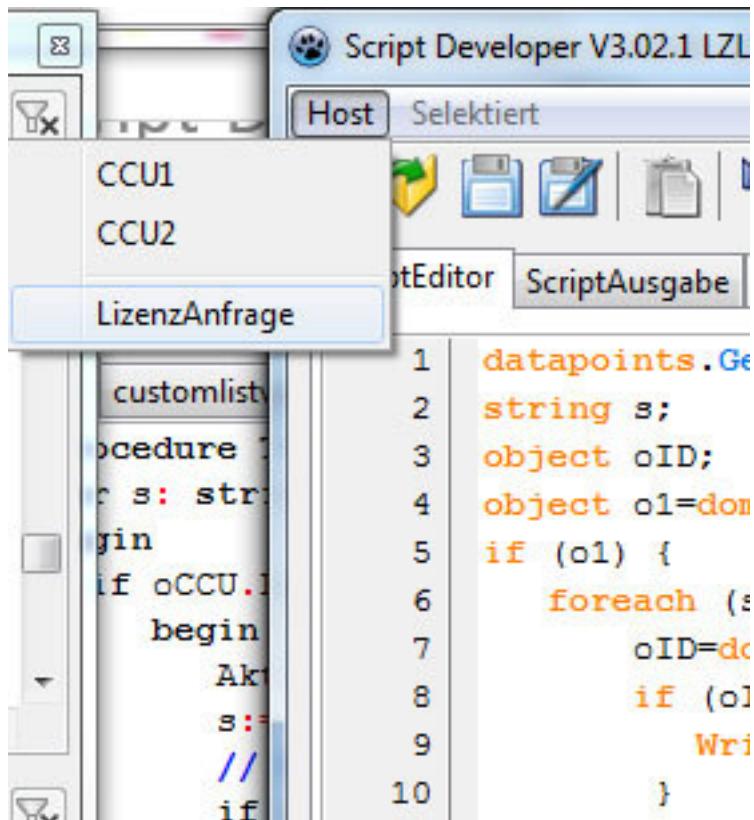
Wenn die CCU, für die der Schlüssel angefragt werden soll, als grün angezeigt wird, bitte vorher einmal unter dem Reiter Einstellungen kontrollieren



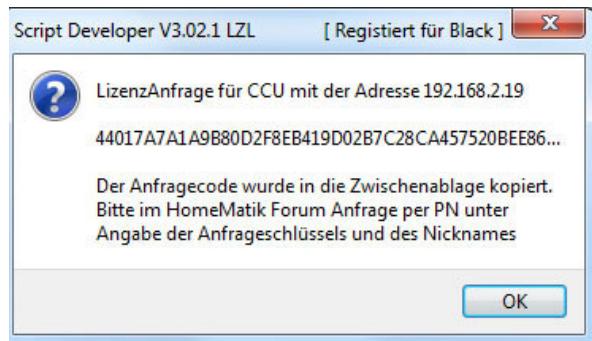
Aus einer Anfrage ohne unter local sinnige Einträge zu sehen lässt sich kein gültiger key generieren.

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

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 Systemvoraussetzungen

Der SDV lief bisher in Testinstallionen unter WIN 7 64/32 bit, und unter Win 10 64bit. Da unter recht konservativen Compilereinstellungen übersetzt wurde, sollte er eigentlich unter allen Windows Version laufen (ab Win 7)

Auf der Homematic-Seite wurde bei mir auf einer Raspberrymatic 3.37.8.20181026 und 3.41.11.20181126 getestet. Allerdings ist noch nicht der Authorisierungsmechanismus über Nutzername/Passwort implementiert. Der SDV braucht Zugriff über Port :8181

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

1.3 Was tuts bis jetzt

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

Undo / Redo arbeiten

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

Enums und Sysvars arbeiten auch schon inkl Detaildaten und Editermöglichkeiten.

Darstellbarkeit zumindest der Grundmethoden aller Objekte

DomScan

Devices

Aufschlüsseln der MetaDaten

Datenpunkte

Kanäle

Programme

Favoriten

User

Paramset Master

Script Bearbeitung aus Programmen heraus

Verändert von Verzögert um von Scripten

Verändern von Retriggern

1.4 Bekannte Einschränkungen / Bugs

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

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

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

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

1.6 Changelog

1.6.1 Changelog 03.06.04 LZL

2 Bugfix aufgrund Rega Fehlern (DoubleQuotes und wissenschaftliche Notation)
TSynHighlighterClass angepasst: 2. KontantenTable eingeführt
Undo Stack eingeführt für den Inspektor, damit ist jetzt quasi Browsing durch die Rekursionsebenen möglich
Wegen Undo Stack Sicherheitsabfrage Rekursives Auflösen entfernt (geht ja nun via Undo wieder zurück)
Programme lösen nun Ihre Rules und Subrules auf (retriggern veränderbar)
Bei einem Script als SingleDestination ist dieses Script direkt in den Editor ladbar, dort veränderbar und auch wieder ins CCU Programm hochladbar.
Verzögert um dieses Scripts veränderbar aus dem SDV heraus
Diverse weitere Methoden und Konstanten hinzugefügt

1.6.2 Changelog 03.06.01 LZL

BugFix Backup Sysvars (Alarme kann halt der Status nicht hergestellt werden und 950er kann nicht rekontruiert werden. Dafür
Sonderfunktionen
Bugfix Highlighter (in HM Script ist // kein kommentar)
Paramset Master Aufschlüsselung (ab Level 6)
Editor Zusatzfunktionen um Werte aus der Detailsicht in ein Script zu kopieren
DiagnoseBild
Restaurieren einer kaputten oder fehlenden 950er Präsenzvariable tuts (ab Level 7)
Rega Event Push auf DPs (ab level 7)

1.6.3 Changelog 03.05.06 LZL

Multithreading eingeführt für CCU Zugriff, Ping, etc (sollte keine Hänger mehr geben)
TSynHighlighterClass umgeschrieben, im Gegensatz zur originalen Version arbeitet meine nun CaseSensitive wie auch die CCU
Weitere Methoden eingefügt in Highlighter, Autocomplete und Detailansicht.
Detailansicht für Datenpunkte, Alarme, Systemvariablen, Devices und Channels komplettiert
Auf Sonderwunsch unseres Stammtischs Programme schon mal provisorisch mit ProgramCopyID Test eingefügt
Ein paar Standartfilter geschrieben und der Version im Rar beigelegt (*flt Dateien)
Umbenennen von DeviceKanälen nach dem Namen des übergeordneten Devices
Backup und Restore von Devicekanälen
Temporäre Lizenzen nun möglich

1.6.4 Changelog 03.05.01 LZL

Einige Programmfehler beseitigt.
Endlich die Codierung zwischen LZL und dem WebServer richtig in Griff bekommen
Device Objekt in Auswahl hinzu und Detailansicht
Systemvariablen Objekt vervollständigt
Alarm Objekt angelegt und vervollständigt
Metadaten Aufschlüsselung
Backup Methoden für Räume, Gewerke und Systemvariablen

1.6.5 Changelog 03.04.01 LZL

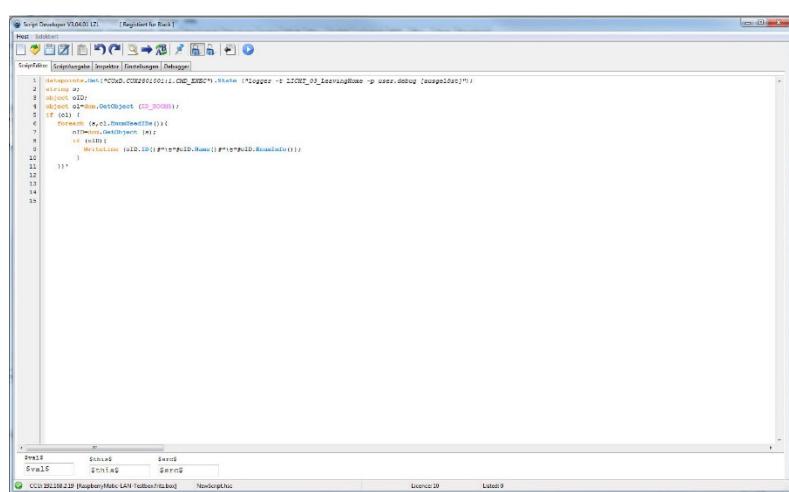
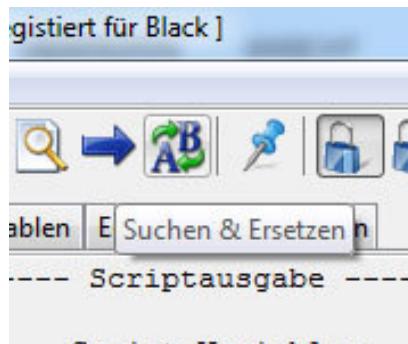
Einige Programmfehler beseitigt.
Variable Fenstergössen im Inspektor
Anzeige zur CCU Nummer die Hinterlegte IP
Scriptnamen Anzeige ohne Pfade (Länge der Anzeige)
Löschen von Objekten
Selektionen aus Selektionen
Sichern und Zurückholen von Selektionsfeldern (PIN)
Merken von Selektionen und Übernahme als Enum in den Scripteditor
Rekursives Auflösen der Detaildarstellung
Einige interne Änderungen, um die nächsten Steps der Roadmap effektiver zu ermöglichen

1.6.6 Changelog 03.03.01 LZL

Listendarstellung Sortieralgorithmus geändert
Kleine Programminkonsistenzen beseitigt.
Feld Objecttyp in Listendarstellung hinzu
Schreibfehle rbei \$src\$ beseitigt.
ObjectSelektion hinzugefügt

2. Oberfläche

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



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

In der Statuszeile finden sich Information über:

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

3 Scripteditor

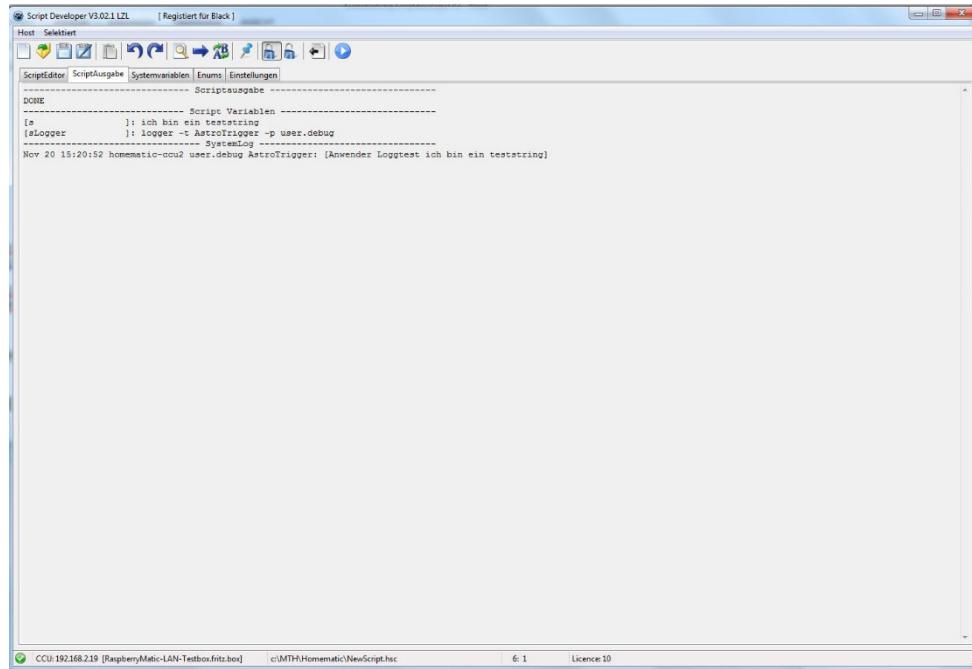
Im Scripteditor werden die Scripte geschrieben oder geladen, die mittels Run Script oder  an die CCU zum Ausführen gesendet werden. Das Scriptergebnis wird dann im Reiter Ausgabe angezeigt. Dieses kleine TestScript zum Beispiel:

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

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

Hier testweile aus State

Erzeugt folgende Ausgabe:



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

```
DONE
----- Script Variablen -----
[s           ]; ich bin ein teststring
[sLogger     ]; logger -t AstroTrigger -p user.debug
----- Systemlog -----
Nov 20 15:20:52 homematic-ccu2 user.debug AstroTrigger: [Anwender Loggtest ich bin ein teststring]
```

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

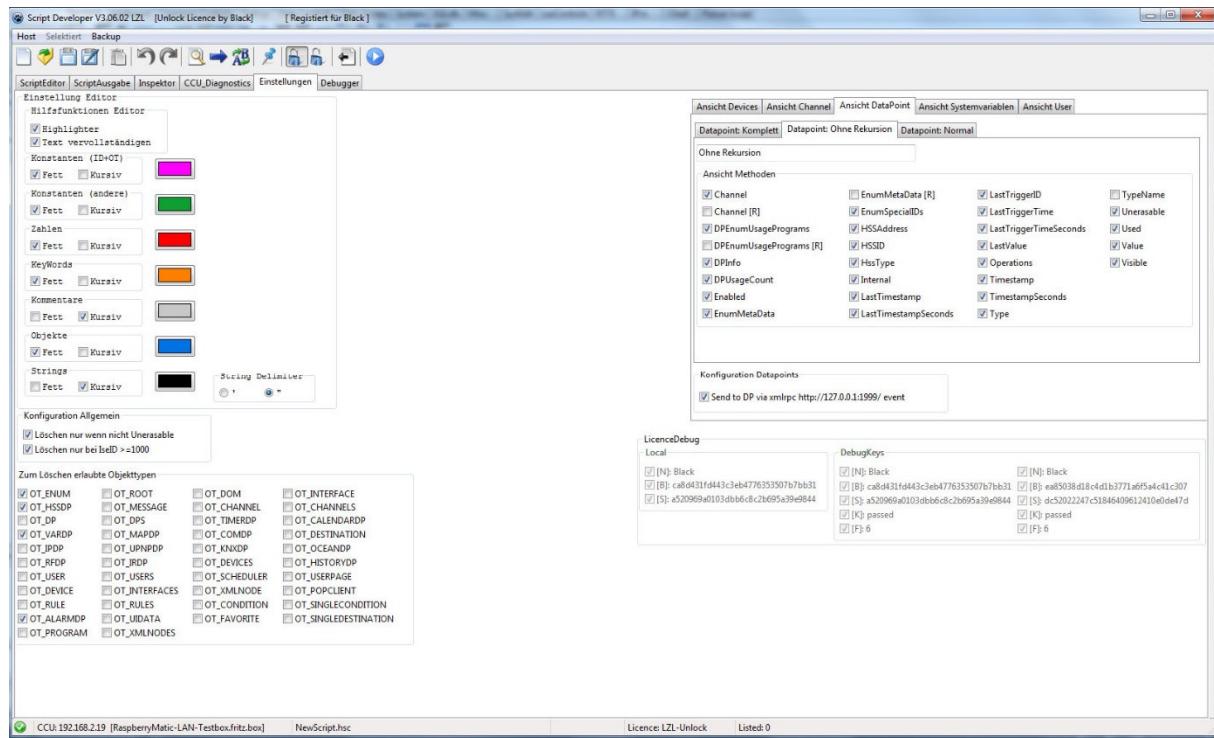
Unter lokale Script variablen stehen die Variablen welche im Script definiert wurden mit ihren Namen. In dem Fall hier sind das die Beiden String Variablen s und sLogger.

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

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

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

3.1 Voreinstellungen Editor



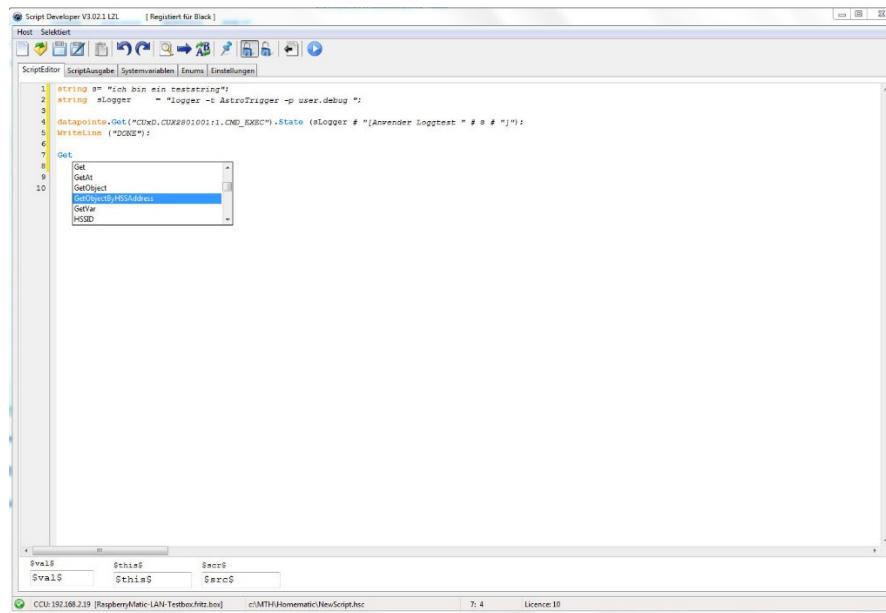
Hier kann nach Vorliebe der Highlighter konfiguriert oder auch ausgeschaltet werden

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 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.

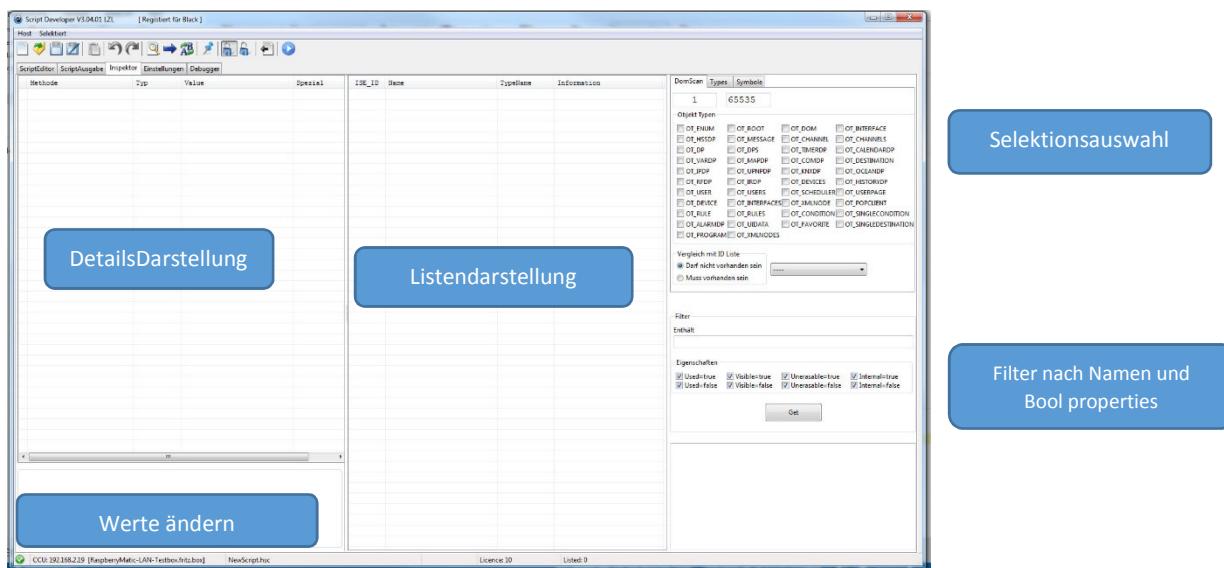


Nach Druck von Enter erscheint das Wort im Editor.

4 Inspektor

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

Es existieren verschiedene Selektionskriterien.



Filteroptionen:

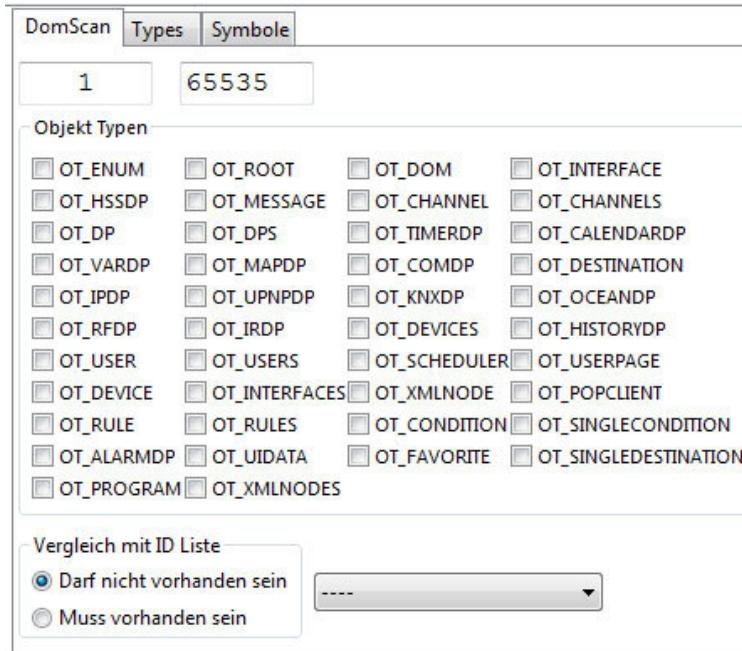
Auswahl der Aufzählungen (Räume, Gewerke, Favoriten, Interfaces , Systemvariablen sind bisher implementiert)

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

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

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

4.1 Selektionswahl: DomScan



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

Achtung

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

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

Mehrfachangaben sind möglich

Beispiel für Suchen aller Objekte mit der Eigenschaft OT_DEVICE im Bereich der ISE_Nummern 1-65535

The screenshot shows the 'Script-Developer V3.06.xx' interface with a search results table and the 'DomScan' dialog. The search results table lists objects with ISE_IDs ranging from 1012 to 7418, all categorized as 'DEVICE'. The 'DomScan' dialog is overlaid, showing the same search parameters and object type selection grid as the previous screenshot.

IseID	Name	TypeName	Information
1012	IP-RCON-10_BlaBlaDP-Ref	DEVICE	
1239	CCU-ETIME	DEVICE	
1414	CCU-ALERTUSER	DEVICE	
2718	CCU-EXCT	DEVICE	
3649	Zeitsymbol	CALENDARDP	
3864	Mo-1C-SvD-PH	CALENDARDP	
4029	Zeitsymbol	CALENDARDP	
5971	Zeitsymbol	CALENDARDP	
6676	Zeitsymbol	CALENDARDP	
7418	RmTP-BSL_001A5A9A628010	DEVICE	

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

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

Hier Beispielsweise: Scanlauf über alle Objekte aus DOM mit der Eigenschaft VARDP, die aber nicht unter ID_SYSTEM_VARIABLES gelistet sind:

The screenshot shows the Script Developer interface with a search results table and a sidebar. The table has columns: Metadate, Type, Value, Special, ID_ID, Name, TypeName, and Information. One row is selected, showing 'ActiveState' as the value for 'Name'. The sidebar on the right lists various object types under 'Object Types'.

Metadate	Type	Value	Special	ID_ID	Name	TypeName	Information
2	integer	3	ActiveState	1704	ActiveState	VARDP	true
* Name	string			1704	ActiveState	VARDP	true
Type	integer	1003		1704	ActiveState	VARDP	true
TypeName	string	RDSP		1704	ActiveState	VARDP	true
IsRoot	boolean	false		2077	=ActiveState	VARDP	false
Internal	boolean	false		2224	=ActiveState	VARDP	false
Visible	boolean	true		2225	=ActiveState	VARDP	true
Unlocked	boolean	false		3401	=ActiveState	VARDP	true
EncryptedData	string			3402	=ActiveState	VARDP	true
DPInfo	boolean	true		3626	=ActiveState	VARDP	true
IsLeaf	boolean	false		3723	=ActiveState	VARDP	true
* ValueName	string			3920	=ActiveState	VARDP	true
ValueType	integer	1	invState	4204	=ActiveState	VARDP	true
ValueSubType	integer	0	invState	4205	=ActiveState	VARDP	true
LastUpdate	boolean	false		4210	=ActiveState	VARDP	true
Timestamp	time	2010-10-24 11:41:09		4270	=ActiveState	VARDP	true
LastTimestamp	time	1970-01-01 01:09:00		5822	=ActiveState	VARDP	true
ValueIndex	string			6790	=ActiveState	VARDP	true
ValueIndex0	string	0		6930	=ActiveState	VARDP	true
ValueIndex1	string	1		7141	=ActiveState	VARDP	true
ValueIndex2	string	2		7358	New program=ActiveState	VARDP	false
ValueIndex3	string	3		7359	New program=ActiveState	VARDP	false
ValueIndex4	string	4		7377	New program=ActiveState	VARDP	false
Channel	integer	0020					
Global	string						
D:\DomScan\Programme	string						

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

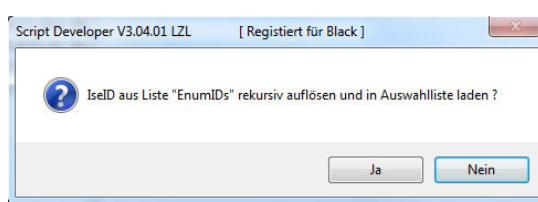
Weiterhin ist auch rekursives Arbeiten nun möglich

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

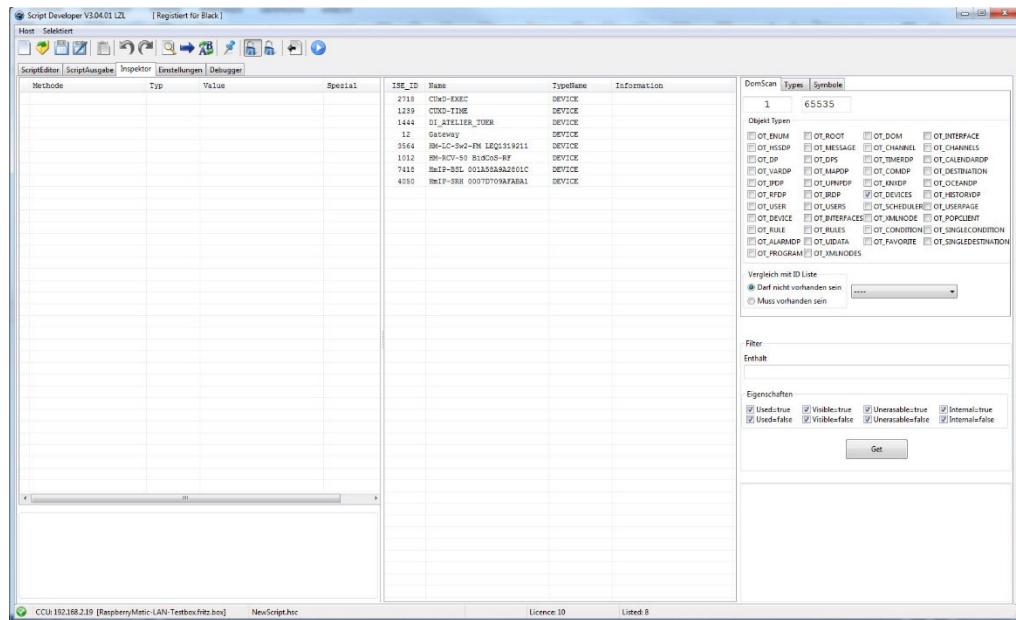
The screenshot shows the Script Developer interface with a search results table and a sidebar. The table has columns: Metadate, Type, Value, Special, ID_ID, Name, TypeName, and Information. One row is selected, showing 'Root device' as the value for 'Name'. The sidebar on the right lists various object types under 'Object Types'.

Metadate	Type	Value	Special	ID_ID	Name	TypeName	Information
2	integer	3	Root device	19	Root device	DEVICE	
* Name	string			2118	DEVICE	CLOUD-EIME	
Type	integer	19		1229	DEVICE	CLOUD-EIME	
TypeName	string	DEVICE		1444	DEVICE	DL_1000ITED_T	
IsRoot	boolean	true		12	DEVICE		
Internal	boolean	false		3166	DEVICE	EN-IC-2x2-7W	
Visible	boolean	true		1612	DEVICE	EN-BCV-80-Bi	
Unlocked	boolean	false		7429	DEVICE	EN13-024-901	
EncryptedData	string			4639	DEVICE	EN13-SR8-901	
DPInfo	string						
IsLeaf	string						

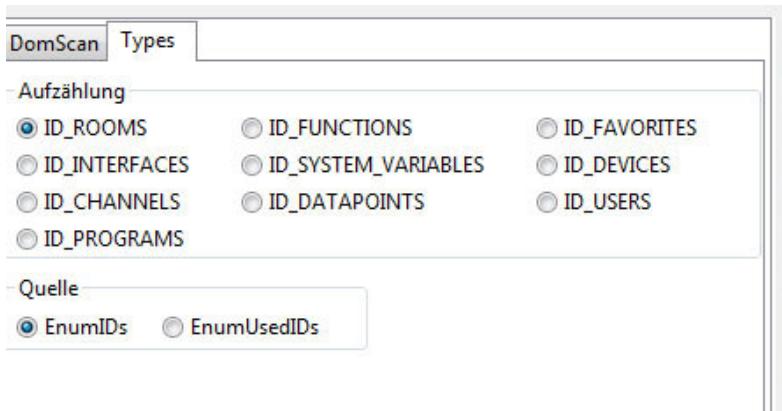
Das „R“ in der ersten Spalte zeigt an, dass ein Rekursiver Aufruf möglich ist. Doppelklicken auf die Zeile öffnet eine Sicherheitsabfrage



Wurde mit JA bestätigt, so wird die EnumList nun Aufgelöst und als neue Auswahlliste zur weiteren Bearbeitung zur Verfügung gestellt.



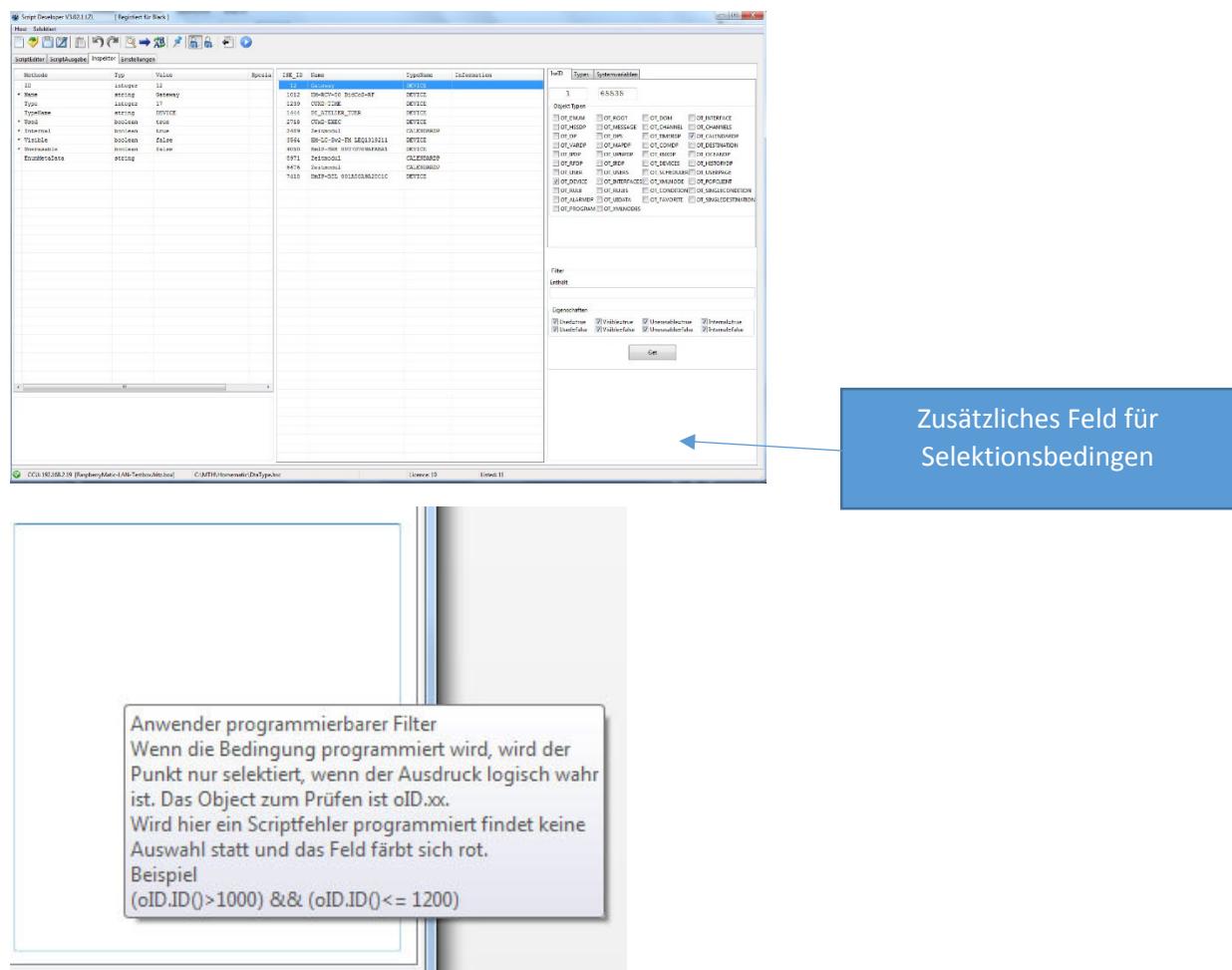
4.2 Selektionskriterium Types



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

4.3 Zusätzliche Selektionsbedingungen

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



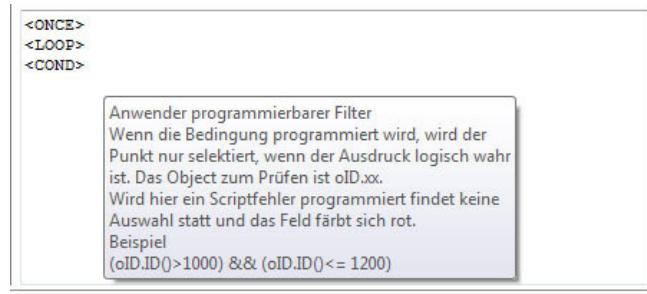
Filter sind ein mächtiges Werkzeug zum komplexen Eingrenzen und für Komplexe Abfragen.

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



Filter löschen entfernt sämtliche Filterbedingungen

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



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

Ein Filter besteht aus den 3 Abschnitten:

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

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

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

Vereinfachter Ablauf: so sieht vereinfacht das Listenselektionsprogramm aus:

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

Ein Anwenderdefinierter Filter wird dann in diese Grundschleife so eingebaut:

```
object oID;
string s;
ONCETEXT;

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

An diesem Kleinen Filter mal verdeutlicht:

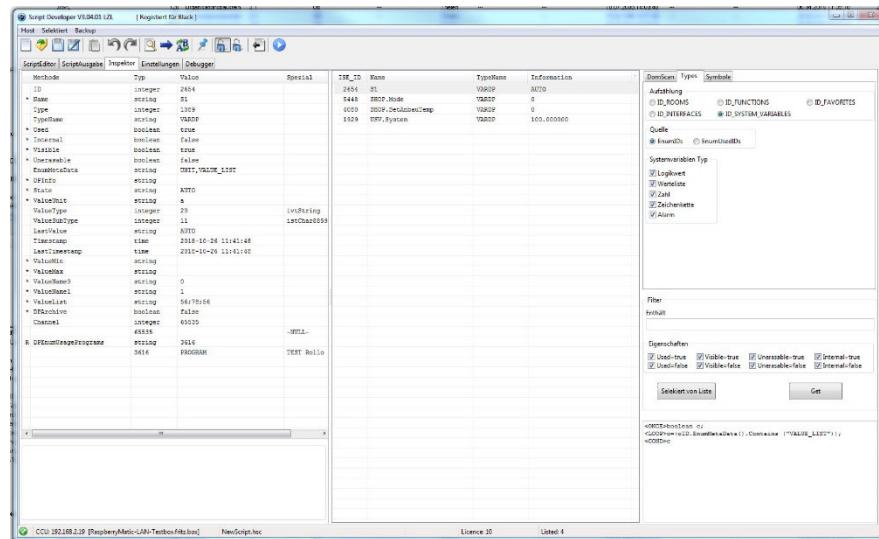
```
<ONCE>boolean c;
<LOOP>c=(oID.EnumMetaData().Contains ("VALUE_LIST"));
<COND>c
```

Daraus generiert der Scriptdeveloper folgende interne Filterabfrage:

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

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

Filtert aus der Gruppe der Systemvariablen alle, in deren Eigenschaft EnumMetadata das Wort VALUE_LIST vorkommt.



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

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

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

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

PROGRAM_GeisterProg_CopyID - Filter um Geisterprogramme mit gesetzter CopyID zu finden

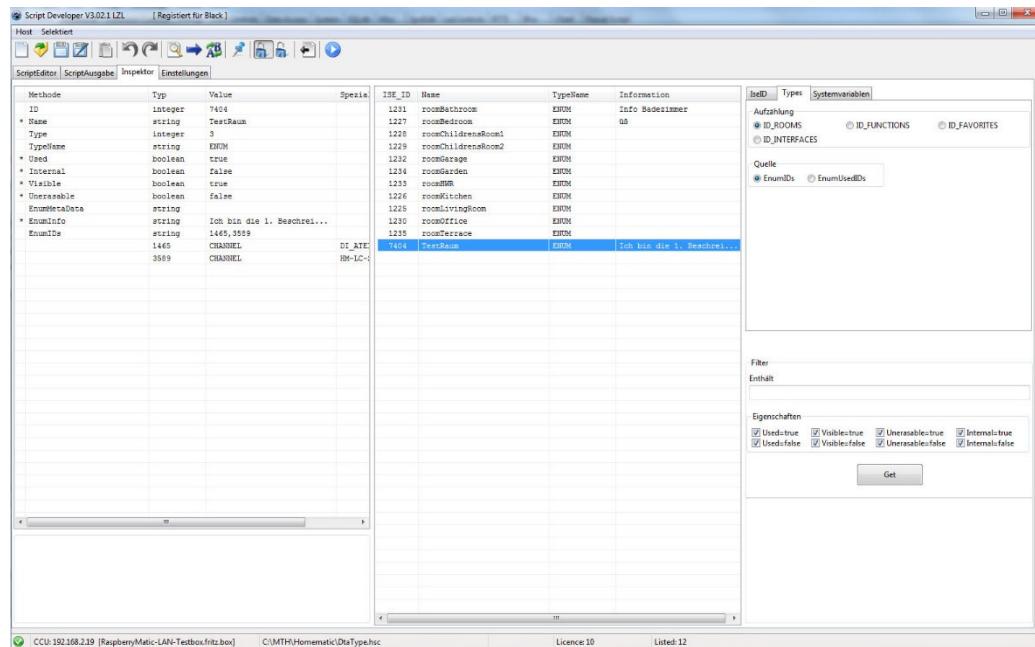
SYSVAR_VerwaisterChannel – Filter um Systemvariablen zu finden, deren Channel verweis in Nirvana zeigt

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

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

Changelog V3.03xx

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



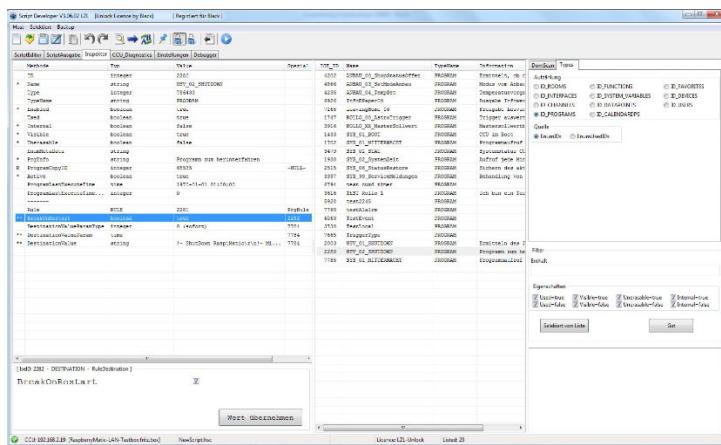
Die Spaltenbreite kann sowohl in Normalansicht als auch in Maximize separat eingestellt werden
(Das Programm sollte sich die Breiten merken und je nach Darstellungsart automatisch wieder einstellen, sollte...)

Dargestellt werden die Methode, der Vartype und die Property.

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

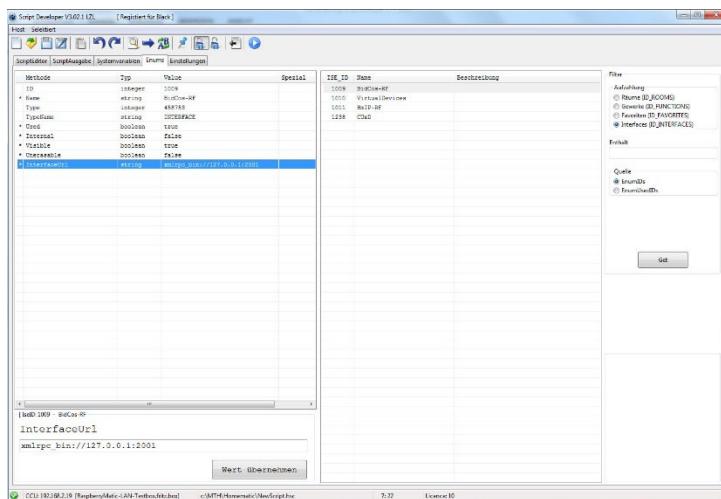
Properties, die in der ersten Zeile mit einem Stern (*) gekennzeichnet sind, können in ihrem Wert geändert werden.

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

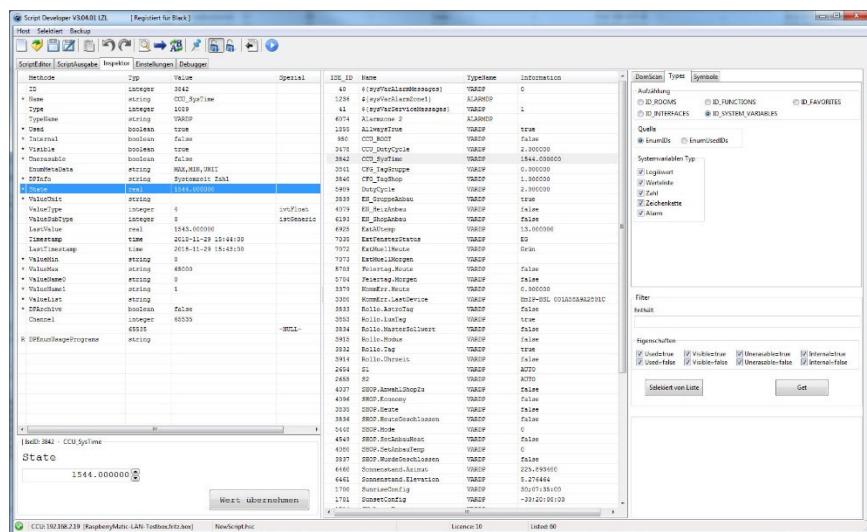


BreakOnRestart ist hier nicht Member von OT_PROGRAM sondern rekursiv von der entsprechenden Rule bzw SubRule.

Dazu auf die Zeile klicken



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

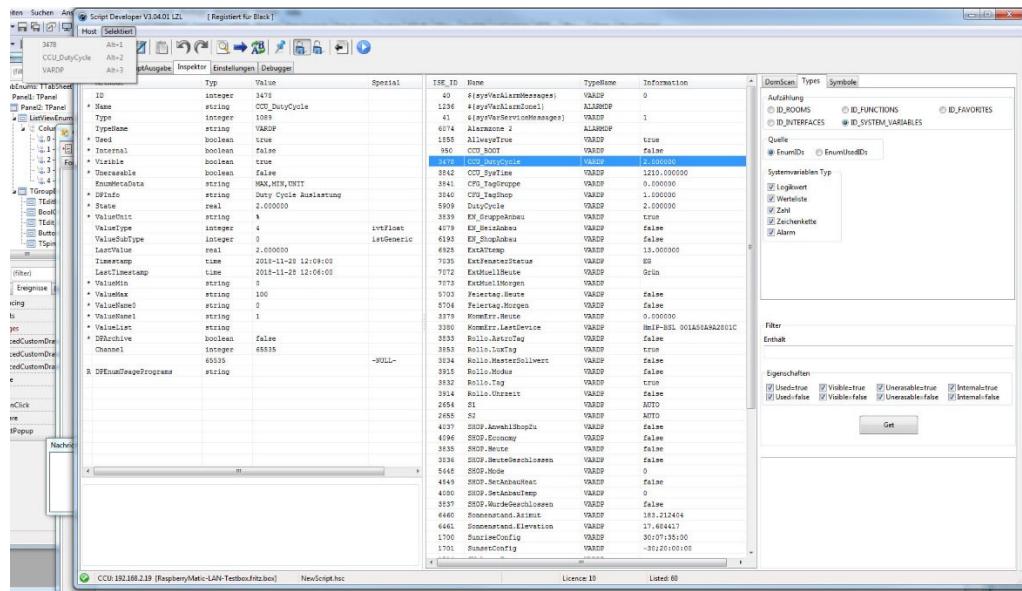


4.4 Daten aus Inspektor in Editor übernehmen

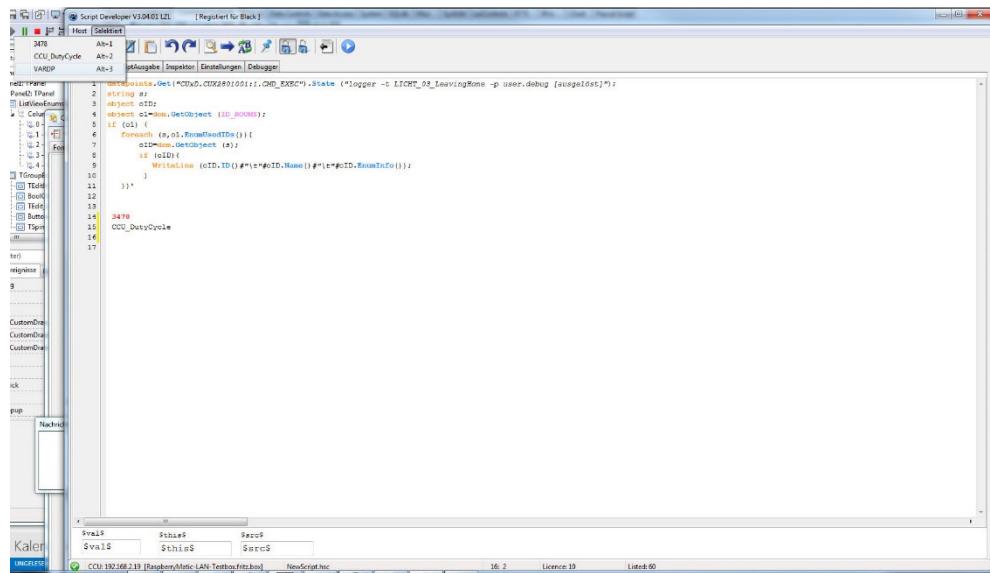
Zur schnelleren und auch möglichst fehlerfreien Bearbeitung besteht die Möglichkeit, Daten aus dem Inspektor direkt in den Editor zu übernehmen.

Immer wenn im Inspektor in den beiden Listviews auf eine Eigenschaft geklickt wurde, stehen diese Daten dann im Editor unter Selektiert zu 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, Alt-2: Name und Alt 3: Eigenschaft bzw. Methode einfügen.



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

Ab 3.06.01

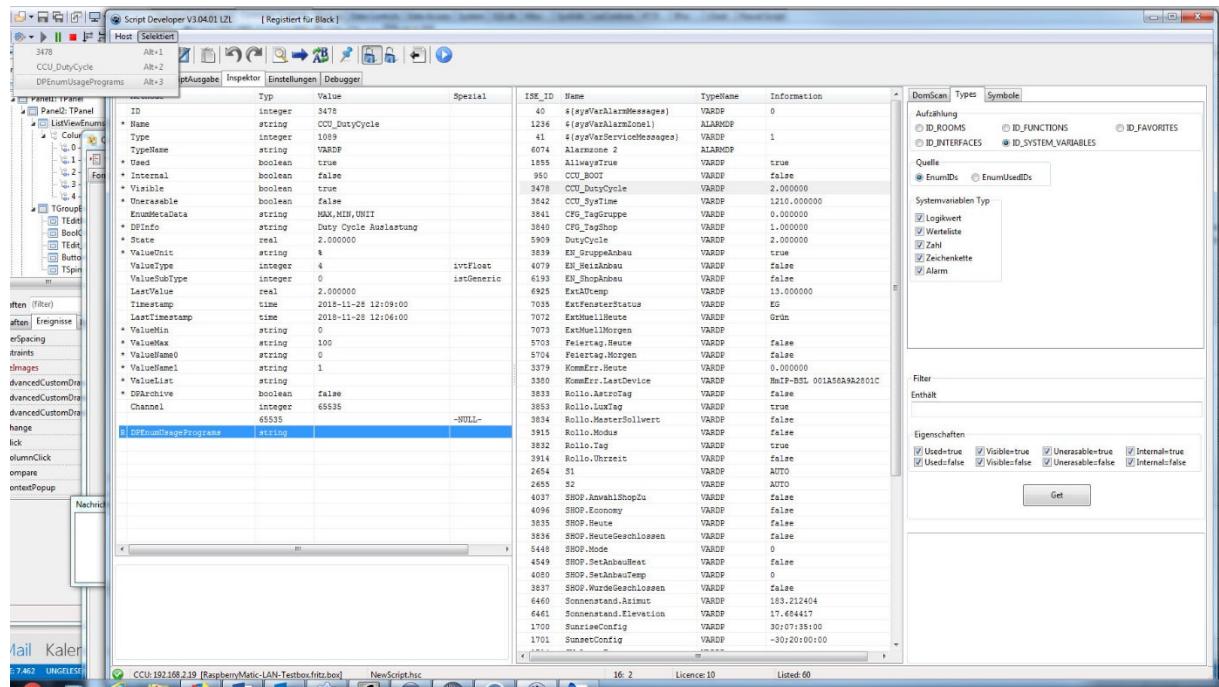
Mit den Tasten lassen sich folgende Spalteninhalte, welche vorher im Detailfeld selektiert worden sind, in den Editor übernehmen

Alt+M: Inhalt von Methode

Alt+T: Inhalt von typ

Alt+V: Inhalt von Value

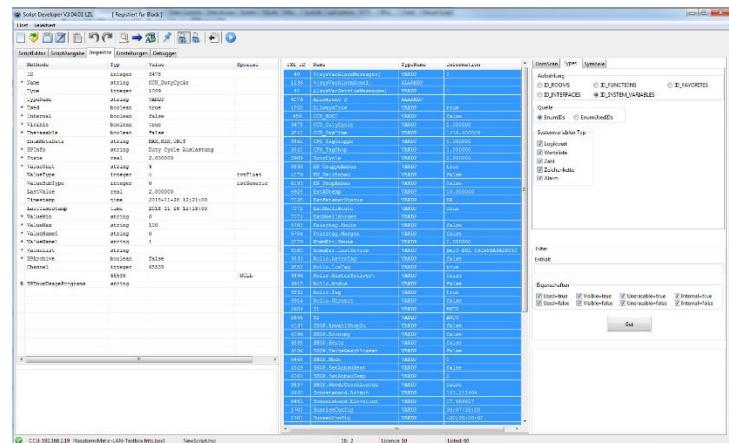
Alt+S: Inhalt von Spezial



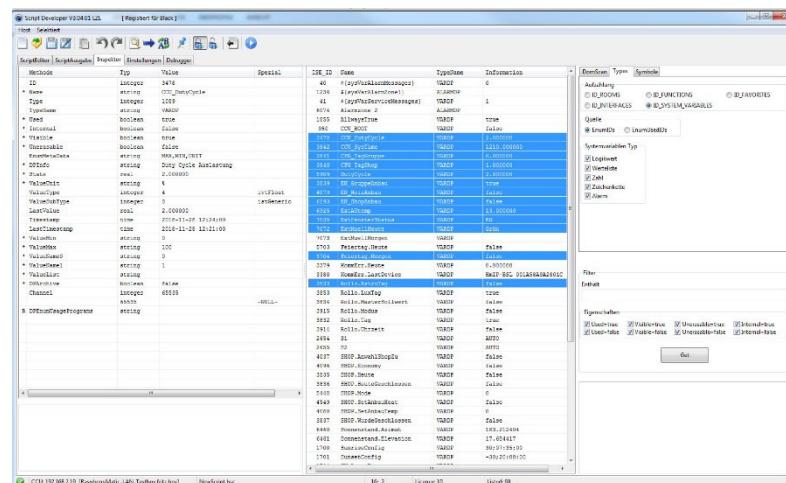
4.4.1 Mehrfachauswahl als Enum String

Es lassen sich im Hauptauswahlfeld Mehrfachselektionen vornehmen.

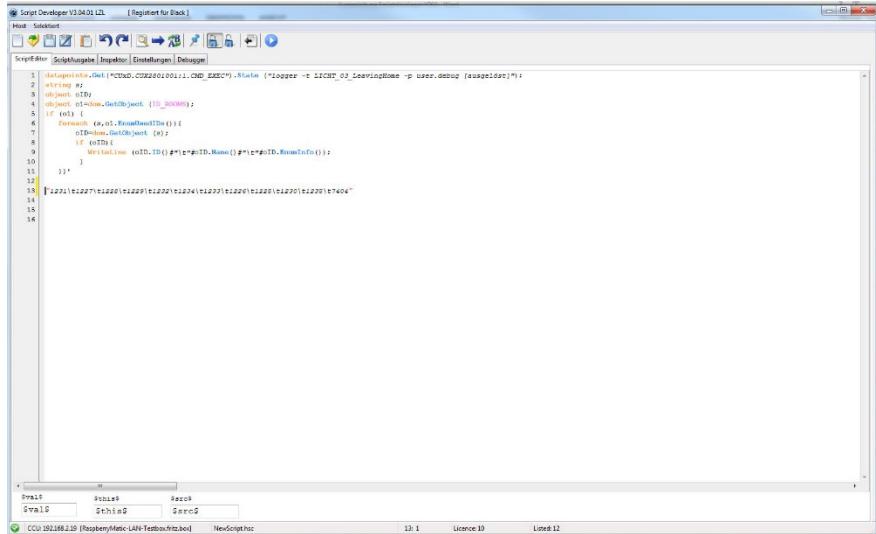
Ctrl-A : alle auswählen.



Oder die Übliche Mausbedienung:



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



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

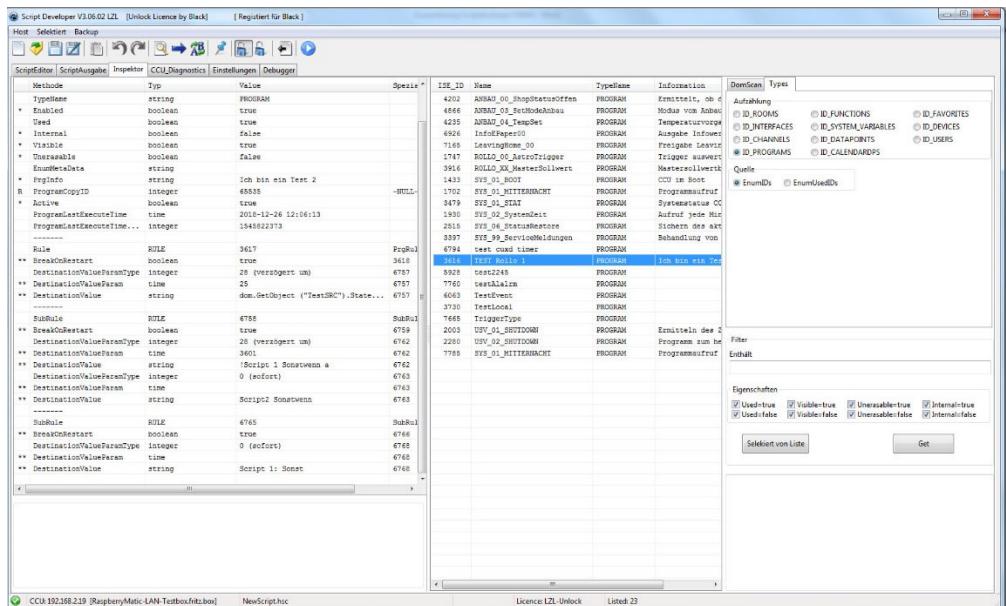
Mit rechter Maustaste im Mittleren Feld die Funktion „Einfügen aus Pinliste“ anwählen und die Sicherheitsabfrage bestätigen,



dann befinden sich die Selektierten Elemente wieder im Mittleren Feld.

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

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



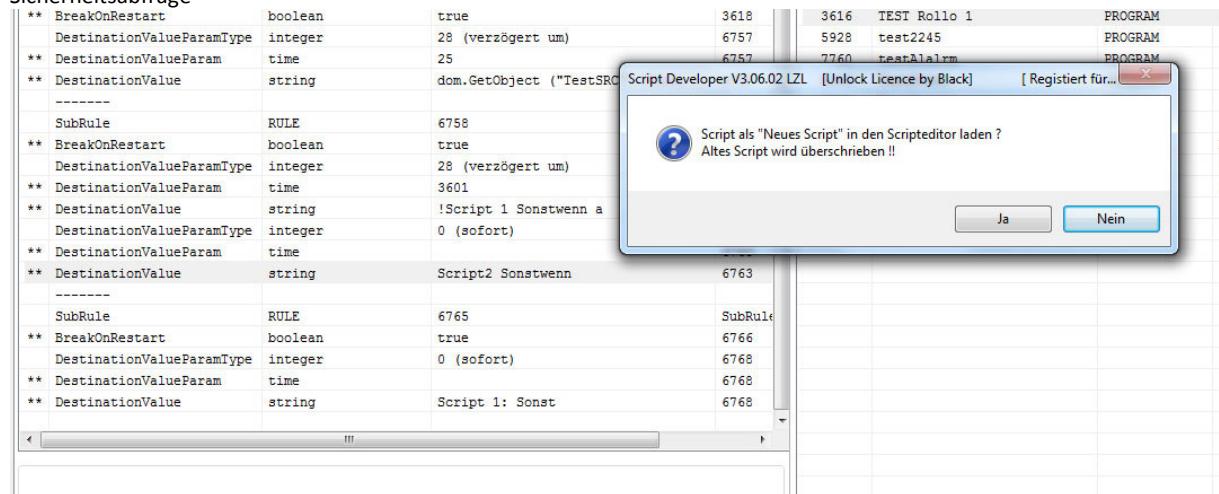
Dies entspricht der tabellarischen Darstellung dieses Programmes

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

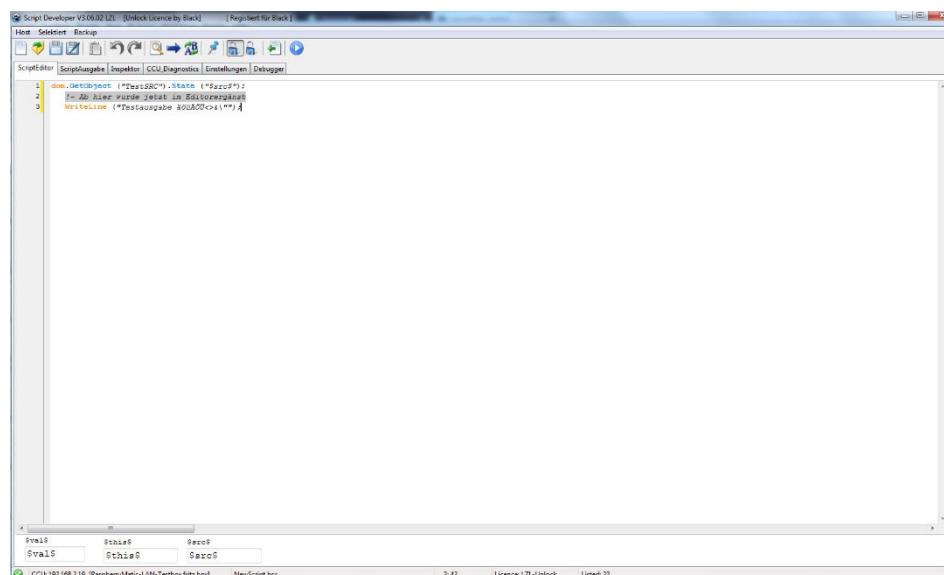
Die erste Subrule enthält dann die beiden Scripte aus der SonstWenn Bedingung.

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

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



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

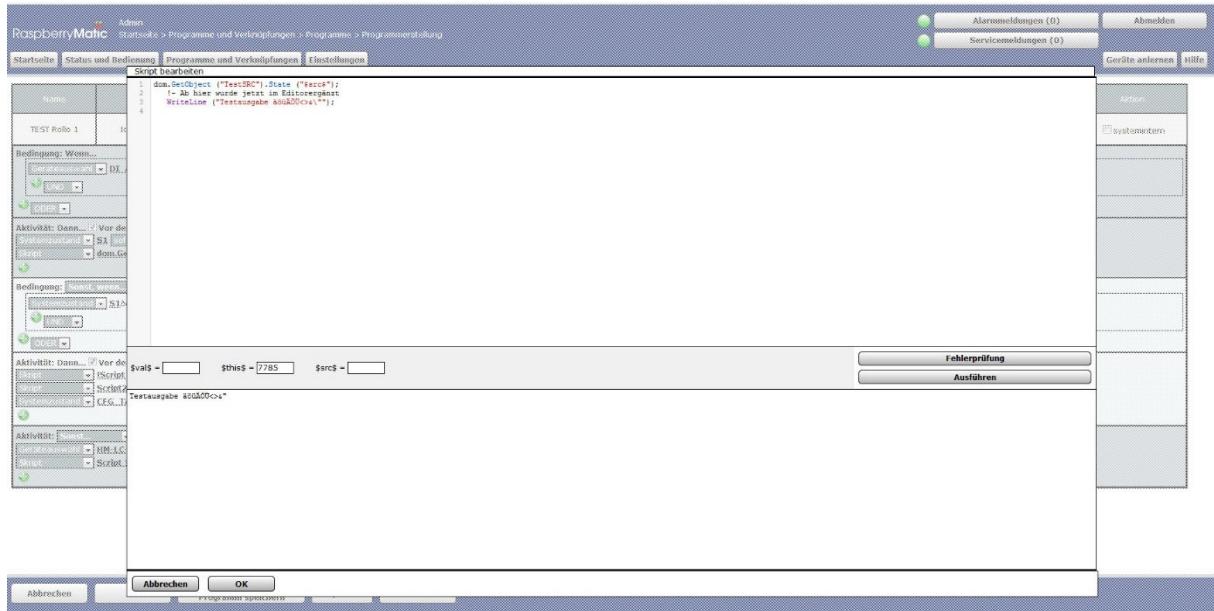


Danach kann durch Drücken der nun nicht mehr grauen Taste



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

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



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

4.5 Selektion von Selektion

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

Hier Beispiel

Methoden	Typ	Value	Spezial	ID	Name	TypeName	Information
TD	integer	1234		40	EV_MyVariableMessage	VARDI	0
* Name	string	\$(myVariableMessage)		41	EV_MyVariableMessage	VARDI	1
Type	integer	2111		950	CTC_ROOT	VARDI	false
System	string	Alles		1241	EV_MyVariableMessage	VARDI	1
* Read	boolean	true		1481	EV_PacketStatus	VARDI	00_AZILIEN_USER.1.1
* Internal	boolean	false		1700	SunriseCoding	VARDI	30/27/2010:00
* Visible	boolean	true		1701	SunsetCoding	VARDI	-50/20/0001:00
* Overridable	boolean	false		1711	EV_PacketLog	VARDI	0
EndeKnoten	string	RPT_RPT_INIT		1860	TestVar	VARDI	29416.00000
				1885	EVLogVarTrue	VARDI	true
				1924	EV_Tagservice	VARDI	0.00000
				1927	EV_Motorstart	VARDI	0.00000
				1928	EV_VoltmeterSteck	VARDI	0.00000
				1929	EV_UVSystem	VARDI	100.00000
				2012	EV_UvshutDown	VARDI	false
				2110	EV_UvWakeUp	VARDI	0.00000
				2116	EV_UvLogIn	VARDI	0.00000
				2451	TEST	VARDI	0.00000
				2474	EV_UvLogOut	VARDI	0
				3293	EV_UvLogOut	VARDI	0
				3294	EV_UvTool	VARDI	false
				3279	EV_Motorstart	VARDI	0.00000
				3290	EV_UvStartLastService	VARDI	0
				3475	EV_UvCycle	VARDI	2.00000
				3562	EV_BollardstielTernJDO	VARDI	true
				3563	EV_BollardstielRnchJDO	VARDI	true
				3483	EV_UvLogIn	VARDI	0.00000
				3832	Bollo-Tag	VARDI	true
				3833	Bollo-AcessTag	VARDI	false
				3834	Bollo-HarterSolvert	VARDI	false
				3835	EV_UvLogOut	VARDI	0
				3236	EV_UvWerkeschleissen	VARDI	false
				3237	EV_UvWurzelbeschleissen	VARDI	false
				3839	EV_Groupschlüssel	VARDI	true
				3840	EV_UvLogIn	VARDI	1.00000
				3841	EV_UvLogOut	VARDI	0.00000
				3842	CTC_System	VARDI	158.00000
				3851	EV_UvLogIn	VARDI	0
				3814	EV_UvTool	VARDI	false
				3915	EV_UvMode	VARDI	false
				4037	EV_ArmelSchlüssel	VARDI	false
				4079	EV_Beizschlüssel	VARDI	false
				4080	EV_UvLogIn	VARDI	0

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

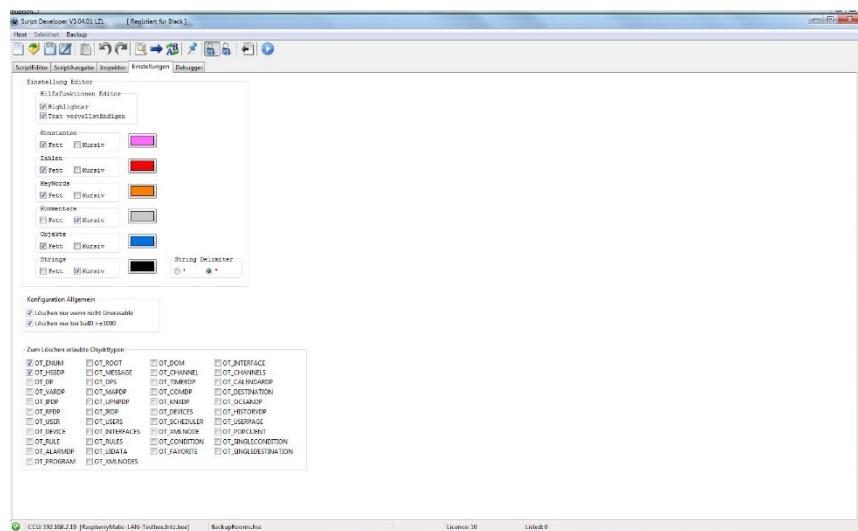
Methoden	Typ	Value	Spezial	ID	Name	TypeName	Information
				1924	EV_UvBatterie	VARDI	0.00000
				1937	EV_UvWerkeschleissen	VARDI	0.00000
				1928	EV_VoltmeterSteck	VARDI	0.00000
				1929	EV_System	VARDI	100.00000

4.6 Objekte löschen

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

Redo geht nur über restore !

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



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

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

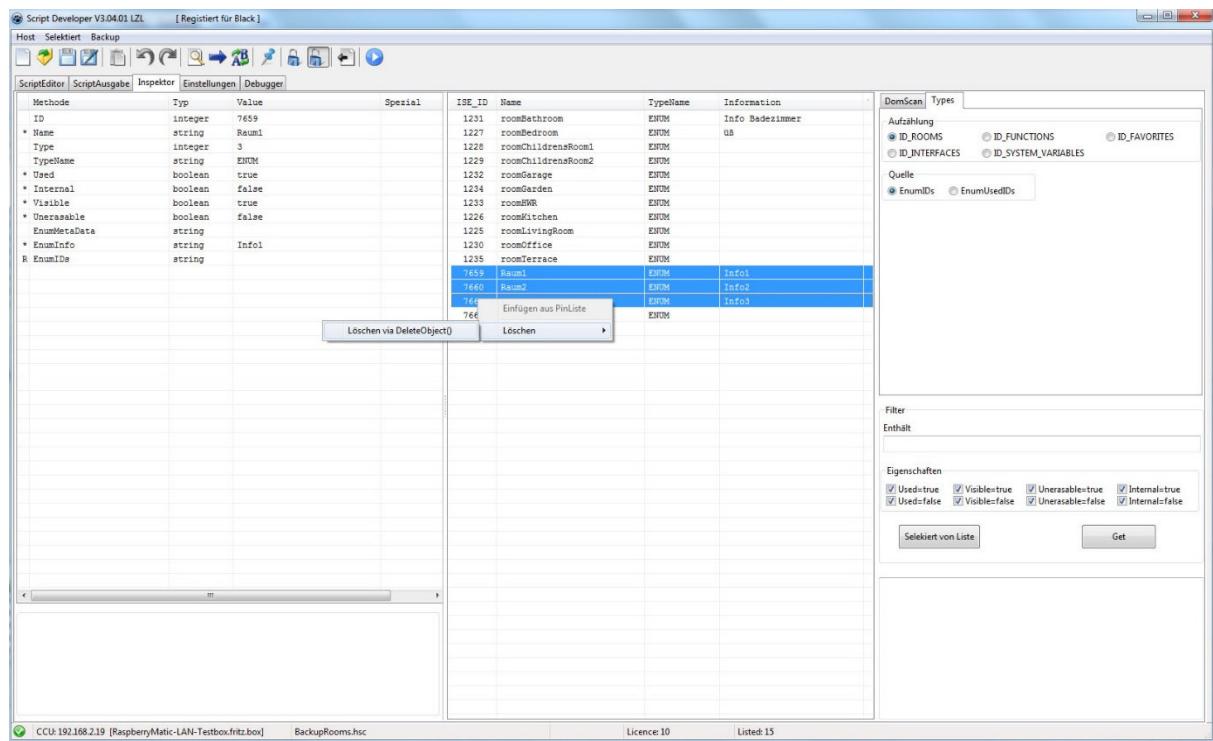
Im Inspektor unter Detailsview die Property entfernen (gilt nur für das Objekt), oder hier den Haken wegmachen (gilt für alle)

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

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

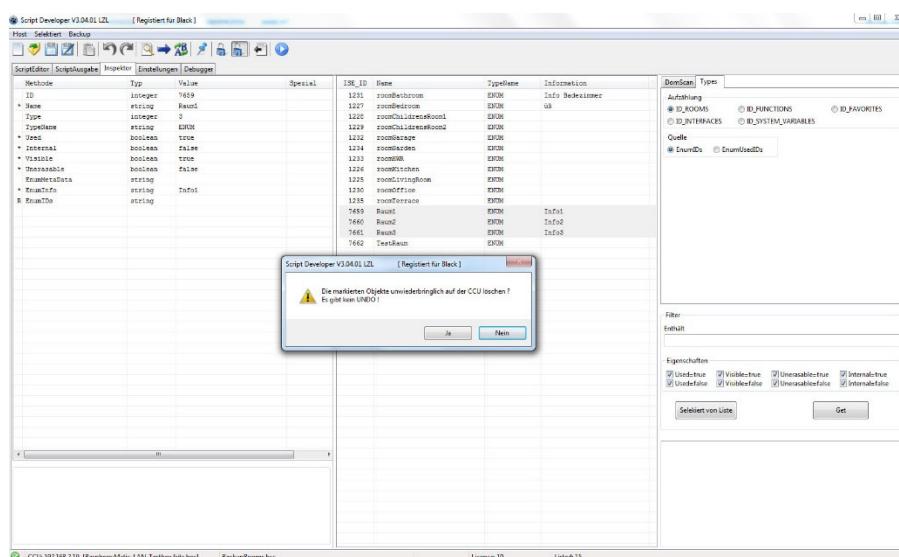
Löschbare Objekttypen. Die Letzte Sicherheit: ein zu lösches Objekt muss einen hier gecheckten Objekttyp haben, sonst wird es nicht gelöscht.

Löschen läuft so ab:



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

Mehrfachselektion ist möglich

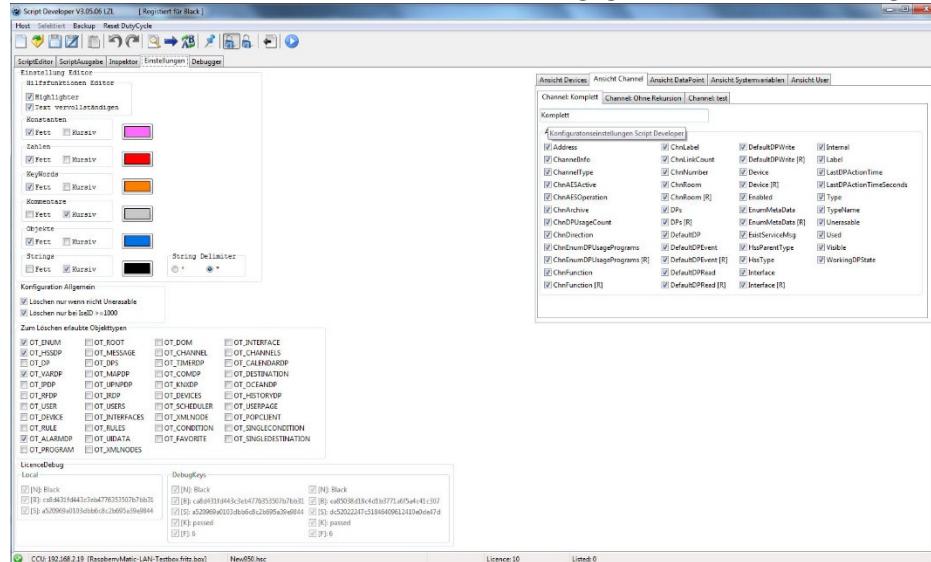


Nach dieser Sicherheitsabfrage sind die Objekte dann weg.. Zurück geht's dann nur mit restore.

4.7 Anwenderdefinierte Sichten

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

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



Sichten können mit eigenen Namen versehen werden.

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

4.8 Browsing durch Rekursionsebenen

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

Hier Beispiel Einsprung über RootDevices

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

Und rekursiv weiter über die Channes des Devices

Script Developer V3.06.02 L2L [Unlock Licence by Black] [Registert für Black]

Host Selektiert Backup

ScriptEditor ScriptAusgabe Inspector CCU_Diagnostics Einstellungen Debugger

Methode	Typ	Value	Spezif.*
LastDPActionTime	time	1970-01-01 01:00:00	
LastDPActionTimeSeconds	integer	0	
WorkingDPState	boolean	false	
ChnAESActive	boolean	false	
ChnAESOperation	integer	3	
ChnArchive	boolean	false	
ChnDirection	string	2	
ExistServiceMsg	boolean	false	
DefaultDP	boolean	false	
DefaultDPEvent	integer	3593	
DefaultDPRRead	integer	3593	
DefaultDPWrite	integer	3593	
ChnDFMessageCount	integer	1	
ChnLinkCount	integer	1	
R_ChnRoom	string	7756	
R_ChnFunction	string	ENUM	Licht
R_DPs	idarray	3593,3592,3590	
		3593	
		3592	
		3590	
R_ChnEnumDPUUsagePrograms	string	3616	PROGRAM
	PARAMSET_MASTER	string	<value><struct><member><name>AE...
			AES_ACTIVE 0
			POWERUP_ACTION 0
			STATUSINFO_MINDELAY 2.000000
			STATUSINFO_RANDOM 1.000000
			TRANSMIT_IRY_MAX 6

ISE_ID	Name	TypeName	Information
3565	HW-1C-Sw2-FW_IEQ1319211:0	CHANNEL	
3589	HW-1C-Sw2-FW_IEQ1319211:1	CHANNEL	
3595	HW-1C-Sw2-FW_IEQ1319211:2	CHANNEL	

DomScan Types

1 65535 ObjektType Auswahl
 Alle Typen
 Nach Objekttypen

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_UPNPDP OT_KNXPDP OT_OCEANDP
OT_RDP OT_JRDP OT_DEVICES OT_HISTORIDP
OT_USER OT_USERS OT_SCHEDULER OT_USERSPAGE
OT_DEVICE OT_INTERFACES OT_MLNODE OT_POCCLIENT
OT_RULE OT_RULES OT_CONDITION OT_SINGLECONDITION
OT_ALARMDP OT_UDATA OT_FAVORITE OT_SINGLEDESTINATION
OT_PROGRAM OT_MLNODES

Vergleich mit ID Liste
 Darf nicht vorhanden sein
 Muss vorhanden sein

Filter

Eigenschaften

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

Selektiert von Liste Get

Mal in die Verwendung in dem Programm schauen über ChnEnumDPUUsagePrograms

Script Developer V3.06.02 L2L [Unlock Licence by Black] [Registert für Black]

Host Selektiert Backup

ScriptEditor ScriptAusgabe Inspector CCU_Diagnostics Einstellungen Debugger

Methode	Typ	Value	Spezif.*
ID	integer	3616	
* Name	string	TEST Rollo 1	
Type	integer	786435	
TypeName	string	PROGRAM	
* Enabled	boolean	true	
Used	boolean	true	
* Internal	boolean	false	
* Visible	boolean	true	
* Unreadable	boolean	false	
EnumMetaData	string		
* ProjInfo	string	Ich bin ein Test 2	
R_ProgramCopyID	integer	65535	-NULL-
* Active	boolean	true	
ProgramLastExecuteTime	time	2016-12-26 12:06:13	
ProgramLastExecuteTime....	integer	1545822373	

Rule	RULE	3617	ProgRule
** BreakOnRestart	boolean	true	3618
DestinationValueParamType	integer	28 (verzögert um)	6757
** DestinationValueParam	time	25	6757
** DestinationValue	string	dom.GetObject ("TestSRC").State...	6757

SubRule	RULE	6758	SubRule
** BreakOnRestart	boolean	true	6759
DestinationValueParamType	integer	28 (verzögert um)	6762
** DestinationValueParam	time	3601	6762
** DestinationValue	string	!Script 1 Sonst wenn a	6762
DestinationValueParamType	integer	0 (sofort)	6763
** DestinationValueParam	time	6763	6763
** DestinationValue	string	Script2 Sonst wenn	6763

SubRule	RULE	6765	SubRule
** BreakOnRestart	boolean	true	6766
DestinationValueParamType	integer	0 (sofort)	6768

DomScan Types

1 65535 ObjektType Auswahl
 Alle Typen
 Nach Objekttypen

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_UPNPDP OT_KNXPDP OT_OCEANDP
OT_RDP OT_JRDP OT_DEVICES OT_HISTORIDP
OT_USER OT_USERS OT_SCHEDULER OT_USERSPAGE
OT_DEVICE OT_INTERFACES OT_MLNODE OT_POCCLIENT
OT_RULE OT_RULES OT_CONDITION OT_SINGLECONDITION
OT_ALARMDP OT_UDATA OT_FAVORITE OT_SINGLEDESTINATION
OT_PROGRAM OT_MLNODES

Vergleich mit ID Liste
 Darf nicht vorhanden sein
 Muss vorhanden sein

Filter

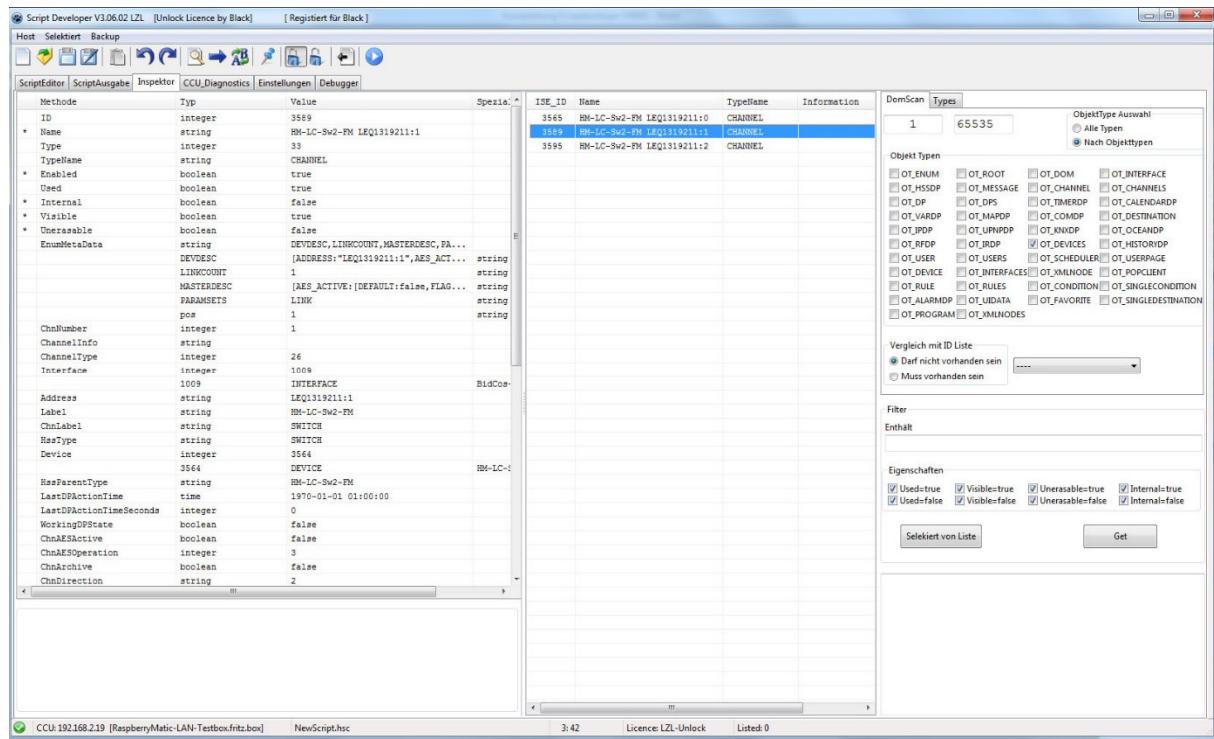
Eigenschaften

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

Selektiert von Liste Get

Will ich jetzt aber wieder in den Channel zurück, sowas in der alten Version 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

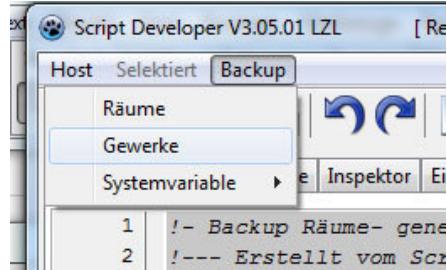


5 Backups

Von relevanten Objekten können Backups gemacht werden. Diese ersetzen KEIN richtiges SystemBackup an der CCU !!!

Vielmehr dienen diese im Falle einer Umzüge 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.

Den passenden Lizenzlevel vorausgesetzt, findet sich die Backups hier:



Devices müssen VORHER manuell umgezogen worden sein über ablernen und neu anlernen. Und die Geräte müssen, damit die Backups von Räumen und Gewerken sinnig arbeiten können, wieder ihre „alten“ Namen haben.

Siehe dazu auch die passende EQ3 Dokumentation. Der SDV legt keine neuen Devices oder Direktverbindungen an.

5.1 Räume

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

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

Dabei passiert folgendes:

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

5.2 Gewerke

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

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

Dabei passiert folgendes:

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

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

5.3 Systemvariablen

Der komplizierteste Part.

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

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

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

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

bcreate:
true: wenn die Systemvariable noch nicht existiert wird diese angelegt und in ID_SYSTEM_VARIABLES
eingehängt.
False: wenn die Systemvariable noch nicht existierte, wird auch nix gemacht.

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

barchive: (nur bei Neuanlage)
true: beim Restore wird die Archiv Option der Systemvariable aus dem Backup genommen.
False: es wird immer ohne Archiv Option angelegt beim Restore.
```

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

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

5.4 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 Kleine Helfer im Alltag

6.1 Umbenennen von Kanälen von Geräten

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

Das Device wird selektiert und der Name der Device geändert.

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

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

Rückfrage mit Ja bestätigen und die Kanäle werden so benannt:

Device: DeviceName

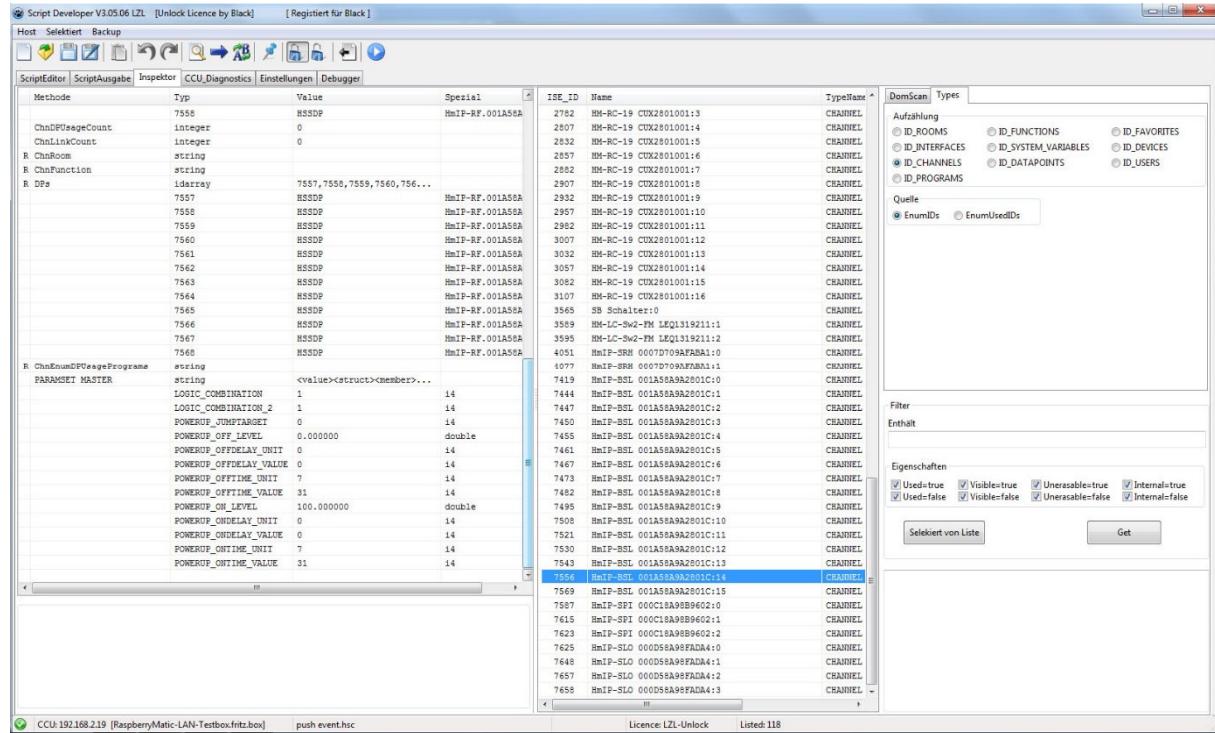
Kanal0 : DeviceName:0

Kanal1 : Devicename:1

Etc...

6.2 Paramset Master

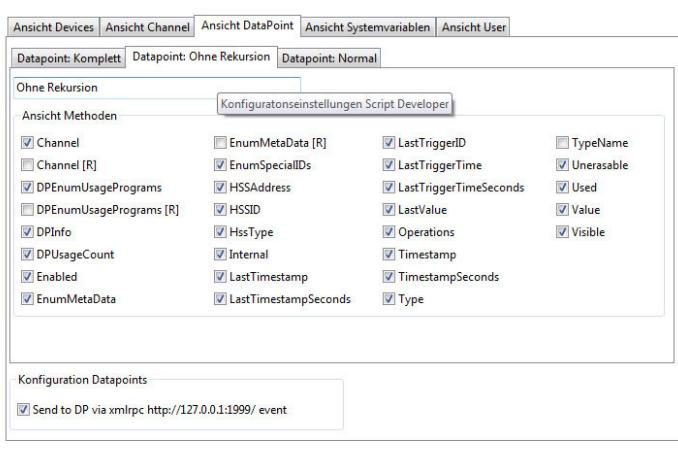
Bei Device, Kanälen, die einen Paramset Master Haben, wird dieser mit angezeigt (wenn in der Sicht konfiguriert und mind. Level 6)



Die Werte lassen sich dann über PutParamset verändern in einem Script

6.3 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 7 braucht es dafür auch)

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

(Wenn die Freigabebedingungen passen)

7 Diagnosebild

```

Programme mit gesetzter CopyID: 0
Dieser Wert sollte normalerweise 0 sein
-----
Check Variable Anwesenheit 950
Alle Checks müssen true sein, sonst Handlungsbedarf
950.Type () : VARDP true
950.ValueType () : 2 true
950.ValueSubType(): 2 true
950.Enabled(): true
950.Used(): true
950.Unerasable(): true
950.Internal(): false true
950 in ID_SYSTEM_VARIABLES true
950 in ID_DATAPOINTS true
-----
Check Variable Alarm Messages 40
40.Type () : VARDP true
40.Name () : $sysVarAlarmMessages
40.ValueType (): 16 true
40.ValueSubType(): 0 true
40.Enabled(): true
40.Used(): true
40.Unerasable(): true
40.Visible(): false true
40.Internal(): true false
40 in ID_SYSTEM_VARIABLES true
40 in ID_DATAPOINTS true
-----
Check Variable Warn Messages 41
41.Type () : VARDP true
41.Name () : $sysVarServiceMessages
41.ValueType (): 16 true
41.ValueSubType(): 0 true
41.Enabled(): true
41.Used(): true
41.Unerasable(): true
41.Visible(): false true
41.Internal(): true false
41 in ID_SYSTEM_VARIABLES true
41 in ID_DATAPOINTS true
-----
VERSION=3.41.11.20181126
PRODUCT=rasmatic_rpi3
PLATFORM=rpi3

DUTY CYCLE 0

Default key = 0
Current user key = 0
Previous user key = 0
Temporary key = 0

MemTotal: 994132 kB
MemFree: 794960 kB
MemAvailable: 862724 kB
Buffers: 68 kB
Cached: 69456 kB
SwapCached: 0 kB
Active: 123044 kB
Inactive: 47052 kB
Active(anon): 94196 kB
Inactive(anon): 204 kB
Active(file): 10448 kB
Inactive(file): 46048 kB
Unevictable: 0 kB
Blocked: 0 kB
SwapTotal: 0 kB
SwapFree: 0 kB
Dirty: 0 kB
Writeback: 0 kB
AnonPages: 93836 kB
Mapped: 21448 kB
Shmem: 640 kB
Slab: 17032 kB
ShReclaimable: 5598 kB
ShUnreclaimable: 1572 kB
KernelStack: 1544 kB
PageTables: 1080 kB
NFS_Unstable: 0 kB
Bounce: 0 kB
WritebackTmp: 0 kB
CommitLimit: 497064 kB
Committed_AS: 143908 kB
VmallocTotal: 1064960 kB
VmallocUsed: 0 kB
VmallocChunk: 0 kB
CmaTotal: 8192 kB
CmaFree: 6796 kB

http://127.0.0.1:9292/bidcos BidCos-RF_java
http://192.168.2.91:2001 hm-rpc.0
xmtrpc_bin://127.0.0.1:31999 1009

```

Check Internal Sysvar
Restore PräsenzVar 950
Aktualisieren

Recht selbsterklärend: rechts etwas status über die Online CCU, das Mittige Fenster ausgaben der jeweils angewählten Funktion:

```

Programme mit gesetzter CopyID: 1
Dieser Wert sollte normalerweise 0 sein
-----
Check Variable Anwesenheit 950
Alle Checks müssen true sein, sonst Handlungsbedarf
950.Type () : VARDP true
950.Name(): BOOT_CCU true
950.ValueType () : 2 true
950.ValueSubType(): 2 true
950.Enabled(): true
950.Used(): true
950.Unerasable(): true
950.Visible(): true
950.Internal(): false true
950 in ID_SYSTEM_VARIABLES true
950 in ID_DATAPOINTS true
-----
Check Variable Alarm Messages 40
40.Type () : VARDP true
40.Name () : $sysVarAlarmMessages
40.ValueType (): 16 true
40.ValueSubType(): 0 true
40.Enabled(): true
40.Used(): true
40.Unerasable(): true
40.Visible(): false true
40.Internal(): true false
40 in ID_SYSTEM_VARIABLES true
40 in ID_DATAPOINTS true
-----
Check Variable Warn Messages 41
41.Type () : VARDP true
41.Name () : $sysVarServiceMessages
41.ValueType (): 16 true
41.ValueSubType(): 0 true
41.Enabled(): true
41.Used(): true
41.Unerasable(): true
41.Visible(): false true
41.Internal(): true false
41 in ID_SYSTEM_VARIABLES true
41 in ID_DATAPOINTS true
-----
VERSION=3.37.8.20181026
PRODUCT=rasmatic_rpi3
PLATFORM=rpi3

DUTY CYCLE 0
DUTY_CYCLE 5

Default key = 0
Current user key = 0
Previous user key = 0
Temporary key = 0

MemTotal: 994132 kB
MemFree: 722552 kB
MemAvailable: 820208 kB
Buffers: 1064 kB
Cached: 111136 kB
SwapCached: 0 kB
Active: 186552 kB
Inactive: 58544 kB
Active(anon): 12592 kB
Inactive(anon): 608 kB
Active(file): 58196 kB
Inactive(file): 57036 kB
Unevictable: 0 kB
Blocked: 0 kB
SwapTotal: 0 kB
SwapFree: 0 kB
Dirty: 0 kB
Writeback: 0 kB
AnonPages: 120320 kB
Mapped: 23620 kB
Shmem: 8676 kB
Slab: 25356 kB
ShReclaimable: 8480 kB
ShUnreclaimable: 14956 kB
KernelStack: 1640 kB
PageTables: 1228 kB
NFS_Unstable: 0 kB
Bounce: 0 kB
WritebackTmp: 0 kB
CommitLimit: 497064 kB
Committed_AS: 203254 kB
VmallocTotal: 1064960 kB
VmallocUsed: 0 kB
VmallocChunk: 0 kB
CmaTotal: 8192 kB
CmaFree: 6792 kB

http://127.0.0.1:9292/bidcos BidCos-RF_java
http://192.168.2.8:2001 hm-rpc.2

```

Check Internal Sysvar
Liste Konstanten
Aktualisieren

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

Alphabetische Auflistung aller bisher bekannten Systemkonstanten mit ihren Werten

Script Developer V3.06.01 L2L [Unlock Licence by Block] [Registriert für Block]	
Host Selektiert Backup	
ScriptEditor ScriptAusgabe Inspektor CCU_Diagnostics Einstellungen Debugger	
Reset DutyCycle	Restore PräsenzVar 950
Check Internal Sysvar	Liste Konstanten
Aktualisieren	
<pre>ID_AZIMUTH_MAPS : 700 ID_CLOCKS : 701 ID_CALENDARPS : 94 ID_CHANNEL_COMMUNICATION : 24 ID_CHANNEL_STATE_VARIABLES : 23 ID_CHANNELS : 4 ID_CIRCUITS : 501 ID_CONTACTS : 35 ID_CONTACTS : 601 ID_DATAPOINTS : 5 ID_DESTINATIONS : 37 ID_DEVICES : 3 ID_DON : 1 ID_EOCLEAN : 880 ID_ENOCLEAN_BEGIN : 851 ID_ENOCLEAN_LASTREC : 892 ID_ERROR : 65535 ID_EVENTING : 11 ID_FAVORITES : 201 ID_FUNCTIONS : 151 ID_RUNTIMECONFIG : 20 ID_GW_CHANNEL : 13 ID_GW_DATAPOINT : 14 ID_GW_DEVICE : 12 ID_GW_SYSALARM : 40 ID_GW_SYSERVICE : 41 ID_INTERFACES : 16 ID_INTERNALCHANNEL : 9 ID_INTERNALCHANNEL : 100 ID_IP_DG_GW : 39 ID_LINKS : 301 ID_MESSAGES : 31 ID_PRESENCE_SIMULATION : 18 ID_PRESENT : 150 ID_PROGRAMS : 15 ID_RF : 900 ID_RF_BEGIN : 901 ID_RF_LASTREC : 942 ID_ROOMS : 101 ID_SOT : 2 ID_RULES : 33 ID_RUNTIMECONFIG : 21 ID_SCENES : 401 ID_SCNDITIONS : 36 ID_SDESTINATIONS : 38 ID_SFILTER_DF : 945 ID_SERVICES : 28 ID_STRUCTURE : 6 ID_SYSTEM_VARIABLES : 27 ID_UI_DATAS : 32 ID_UPNP : 750 VERSION=3.37.8.20181026 PRODUCT=pragmatic_rpi3 PLATFORM=rpi3 DUTY_CYCLE 0 DUTY_CYCLE 5 Default key = 0 Current user key = 0 Previous user key = 0 Temporary key = 0 MemTotal: 994132 kB MemFree: 71084 kB MemAvailable: 822012 kB Buffers: 13644 kB Cached: 111136 kB SwapCached: 0 kB Active: 186562 kB Inactive: 5544 kB Active(anon): 128912 kB Inactive(anon): 608 kB Active(file): 58196 kB Inactive(file): 57936 kB Unevictable: 0 kB Mlocked: 0 kB SwapTotal: 0 kB SwapFree: 0 kB Dirty: 0 kB Writeback: 0 kB Anonymous: 120320 kB Mapped: 23620 kB Shmem: 564 kB Slab: 23436 kB ShReclaimable: 5480 kB ShUnreclaim: 14956 kB KernelStack: 1640 kB PageTables: 1228 kB NFS_Unstable: 0 kB BuddyList: 0 kB WritebackTmp: 0 kB CommitLimit: 497064 kB Committed_AS: 203284 kB VmallocTotal: 1064960 kB VmallocUsed: 0 kB VmallocChunk: 0 kB Commal: 8192 kB CmaFree: 6792 kB http://127.0.0.1:9292/bidcos BidCos-RF_java http://192.168.2.5:2001 hm-rpc.2</pre>	