

Kurzanleitung Scriptdeveloper V3.06.01 (19. 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 und ist nicht an die Nichtverwendung einer Raspberrymatic oder sonstiger Einschränkungen gebunden.

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 V3.06.01 (19. 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	9
1.6 Changelog	10
1.6.3 Changelog 06.06.01 LZL	10
1.6.2 Changelog 03.05.06 LZL	10
1.6.3 Changelog 03.05.01 LZL	10
1.6.4 Changelog 03.04.01 LZL	10
1.6.5 Changelog 03.03.01 LZL	10
2. Oberfläche	11
3 Scripteditor	12
3.1 Voreinstellungen Editor	13
3.2 Vervollständigen Funktion	14
4 Inspektor	15
4.1 Selektionswahl: DomScan	16
4.2 Selektionskriterium Types	19
4.3 Zusätzliche Selektionsbedingungen	20
4.4 Daten aus Inspektor in Editor übernehmen	26
4.4.1 Mehrfachauswahl als Enum String	28
4.5 Selektion von Selektion	30
4.6 Objekte löschen	31
4.7 Anwenderdefinierte Sichten	33
5 Backups	34
5.1 Räume	34
5.2 Gewerke	34
5.3 Systemvariablen	35
5.4 Devices und Kanäle	35
6 Kleine Helfer im Alltag	36
6.1 Umbenennen von Kanälen von Geräten	36
6.2 Paramset Master	37
6.3	37
7 Diagnosebild	39

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 mitgelieferten INI alle Schlüssel ab
[CFG_ChanView1_Methods]
Address=True....
Und in die bestehende INI Datei anzuhängen. Erspart eine Menge manueller Hakensetzen.

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

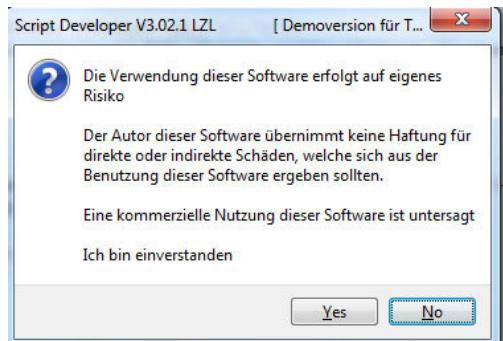
1.1 Lizenzyierung

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 ?

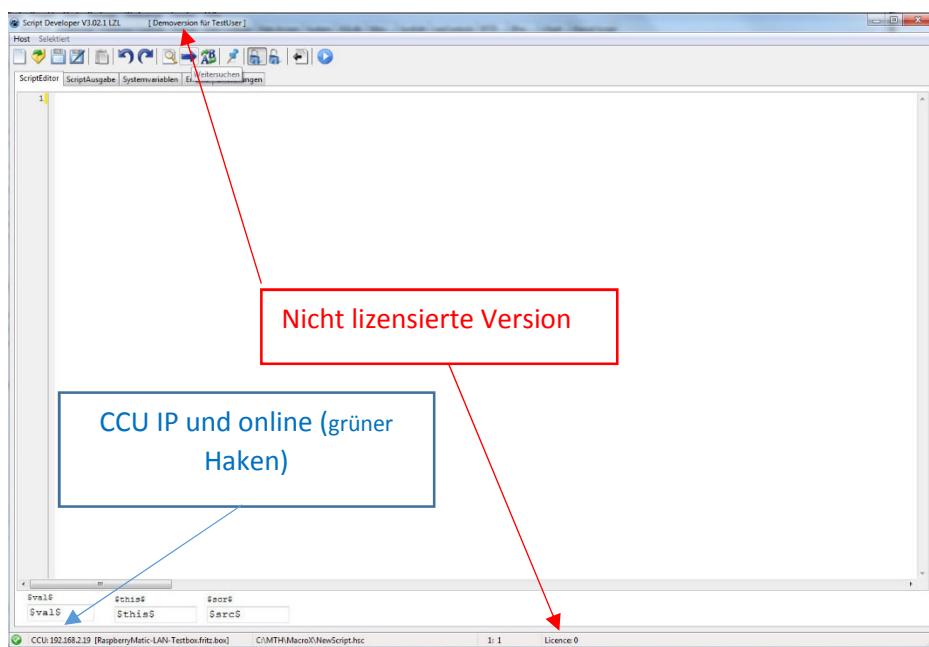
Der SDV telefoniert nicht nach Hause. Um eine Lizenz anzufragen ist folgender Weg einzuschlagen.

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

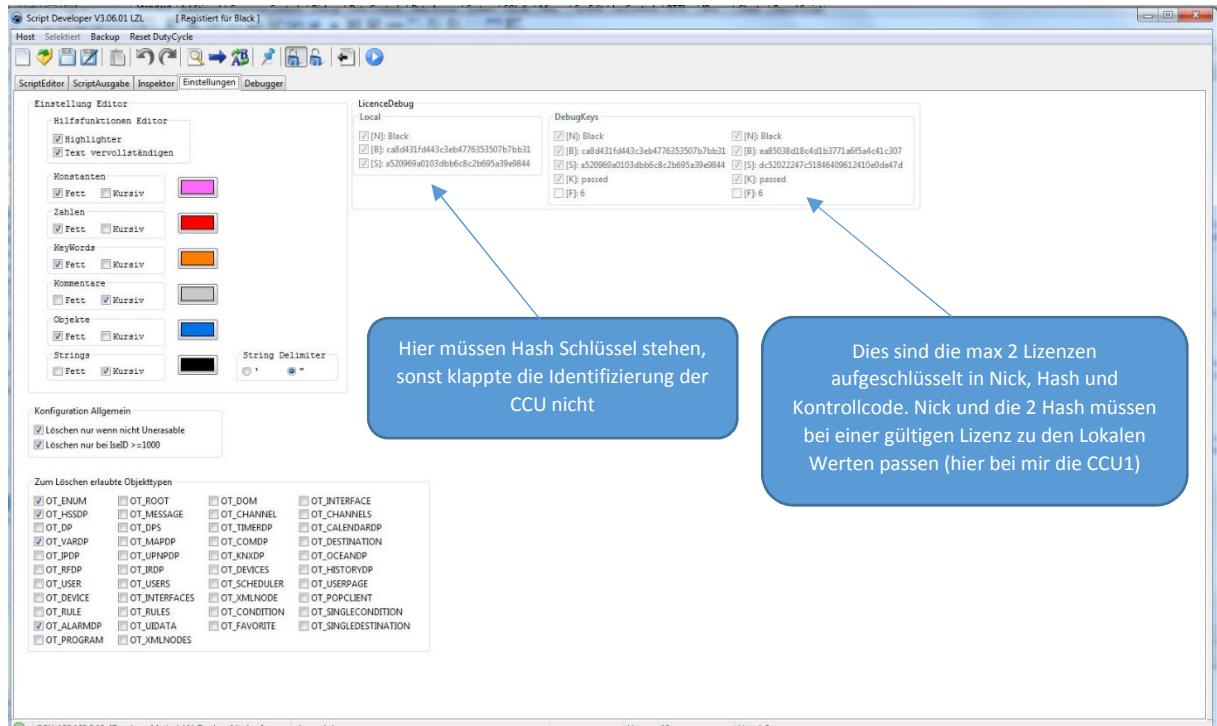


Bei allerersten Start muss dieses Fenster mit yes bestätigt werden. No führt so einem sofortigen Programmabbruch

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



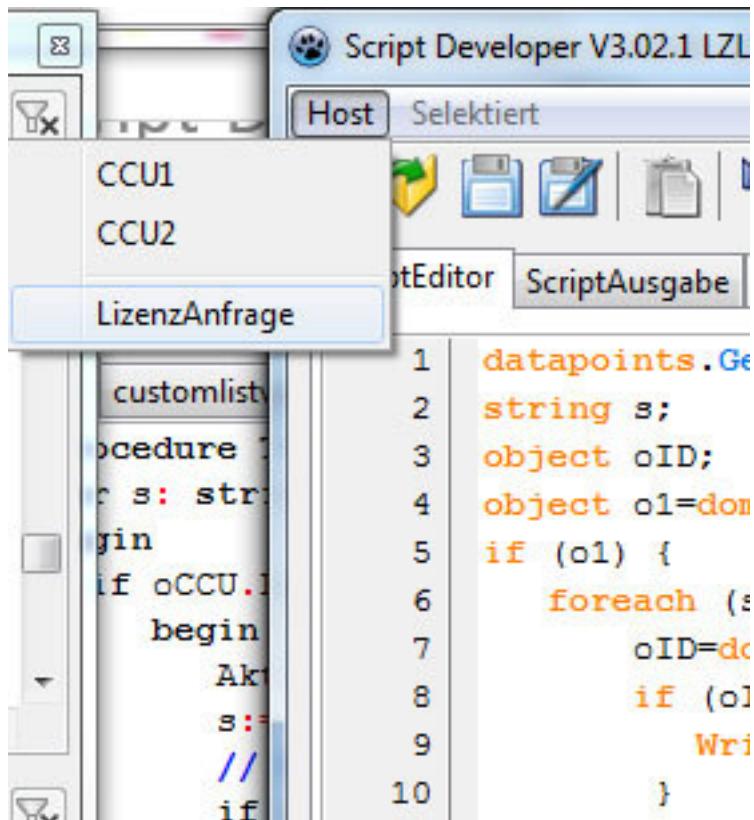
Wenn die CCU, für die der Schlüssel angefragt werden soll, als grün angezeigt wird, bitte vorher einmal unter dem Reiter Einstellungen kontrollieren



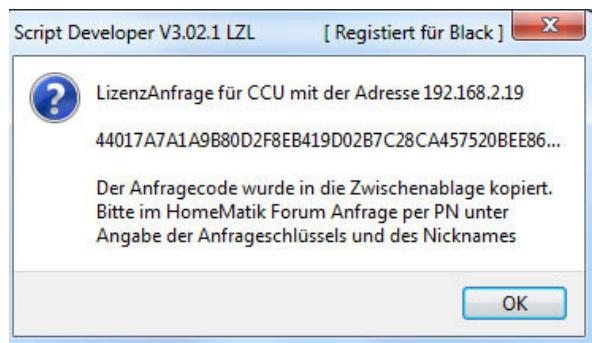
Aus einer Anfrage ohne unter local sinnige Einträge zu sehen lässt sich kein gültiger key generieren.

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

1.4 Bekannte Einschränkungen / Bugs

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

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

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

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

1.6 Changelog

1.6.3 Changelog 06.06.01 LZL

BugFix Backup Sysvars (Alarme kann halt der Status nicht herstellt 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)

1.6.2 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 beigefügt (*flt Dateien)
Umbenennen von DeviceKanälen nach dem Namen des übergeordneten Devices
Backup und Restore von Devicekanälen
Temporäre Lizenzen nun möglich

1.6.3 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

1.6.4 Changelog 03.04.01 LZL

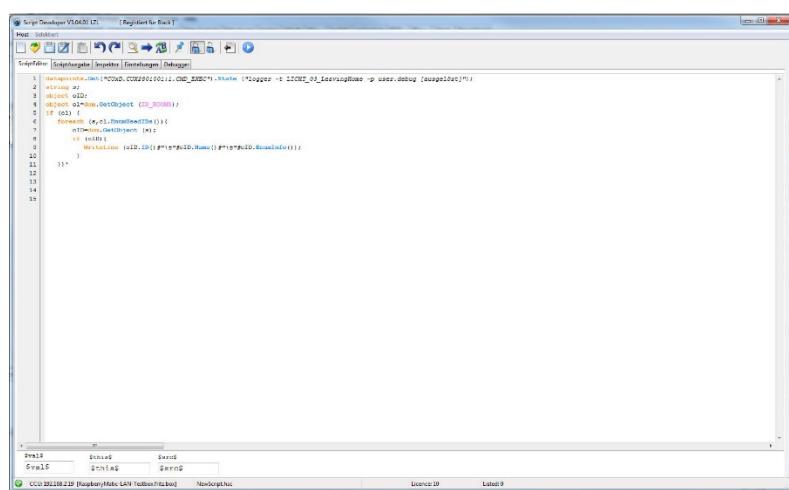
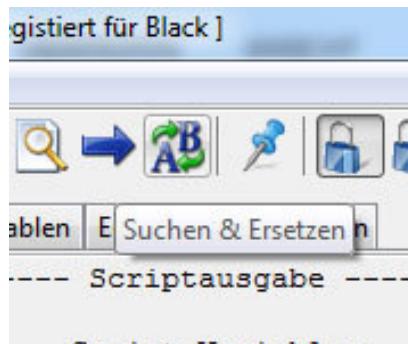
Einige Programmfehler beseitigt.
Variable Fenstergössen im Inspektor
Anzeige zur CCU Nummer die Hinterlegte IP
Scriptnamen Anzeige ohne Pfade (Länge der Anzeige)
Löschen von Objekten
Selektionen aus Selektionen
Sichern und Zurückholen von Selektionsfeldern (PIN)
Merken von Selektionen und Übernahme als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

1.6.5 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 Scriptes  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

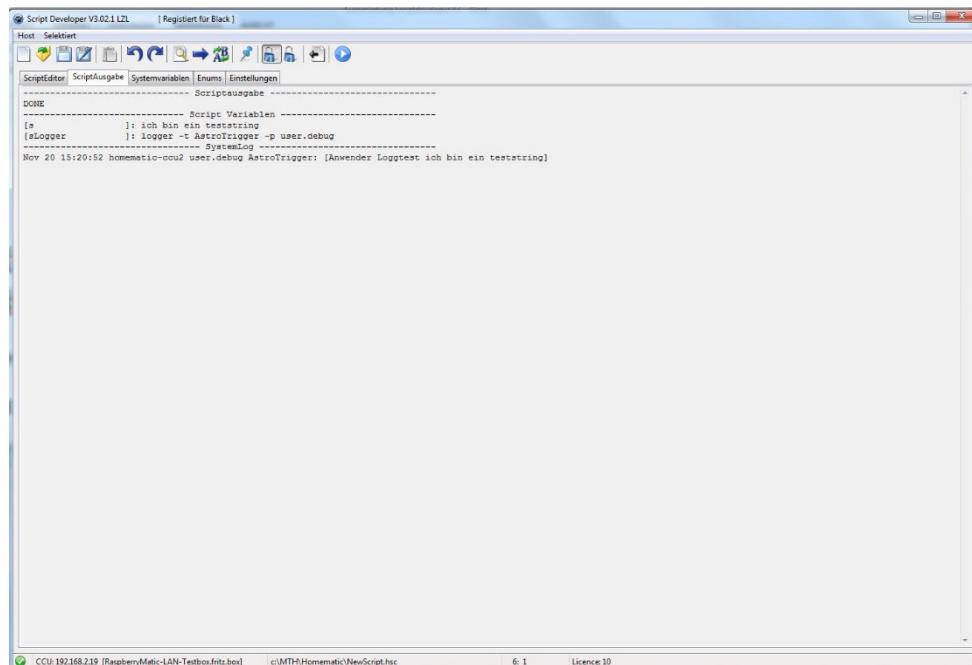
Im Scripteditor werden die Scripte geschrieben oder geladen, die mittels Run Script oder  an die CCU zum Ausführen gesendet werden. Das Scriptergebnis wird dann im Reiter Ausgabe angezeigt. Dieses kleine TestScript zum Beispiel:

```
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,

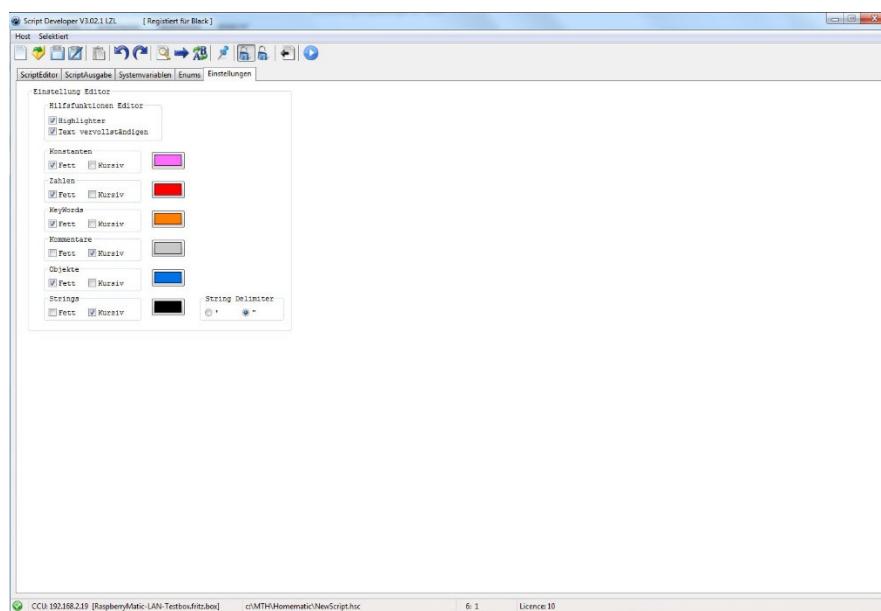
Unter lokale Script variablen stehen die Variablen welche im Script definiert wurden mit ihren Namen. In dem Fall hier sind das die Beiden String Variablen s und sLogger.

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



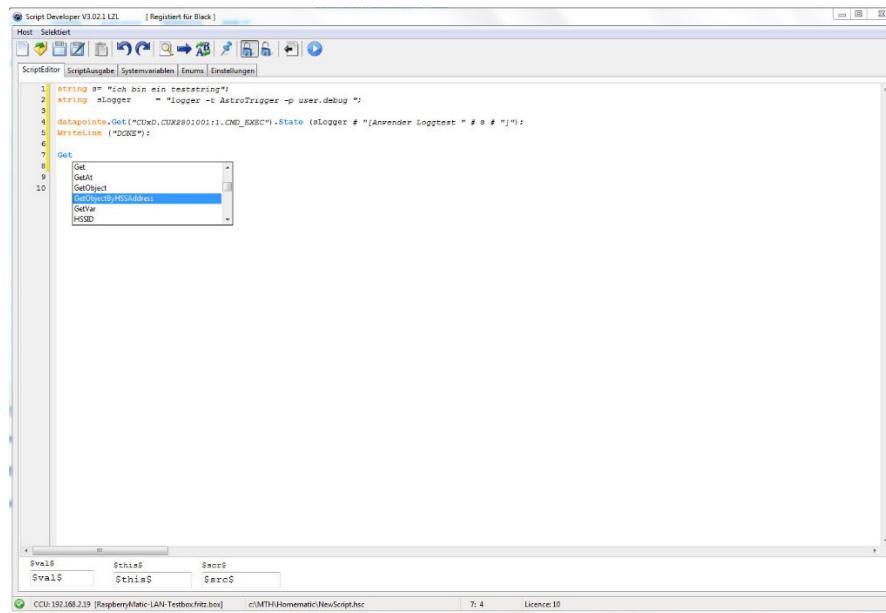
Hier kann nach Vorliebe der Highlighter konfiguriert oder auch ausgeschaltet werden

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.

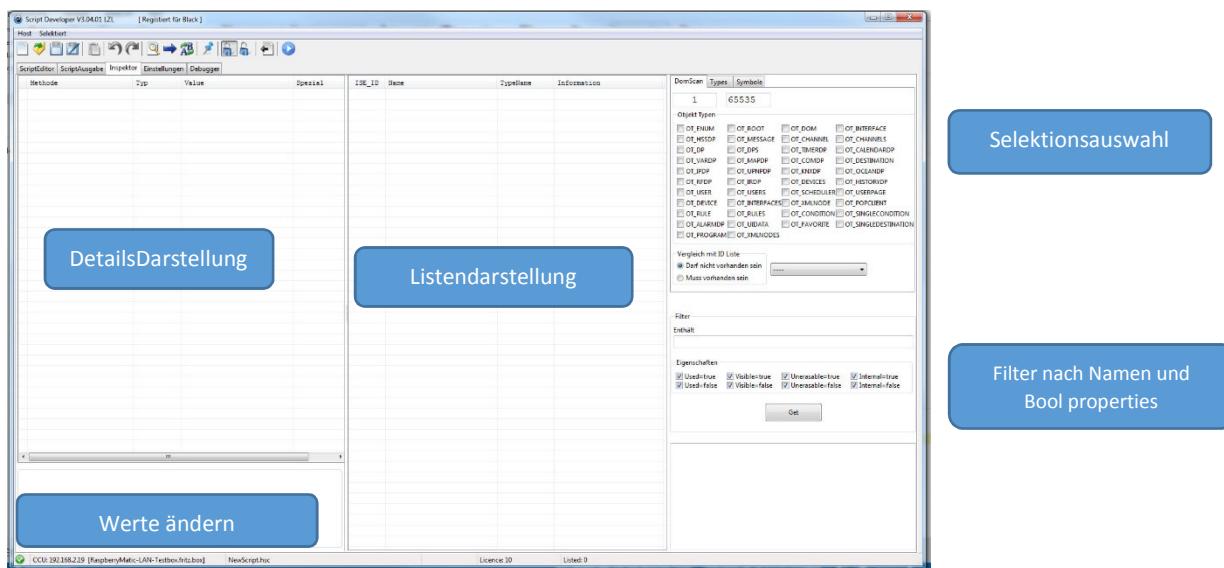


Nach Druck von Enter erscheint das Wort im Editor.

4 Inspektor

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

Es existieren verschiedene Selektionskriterien.



Filteroptionen:

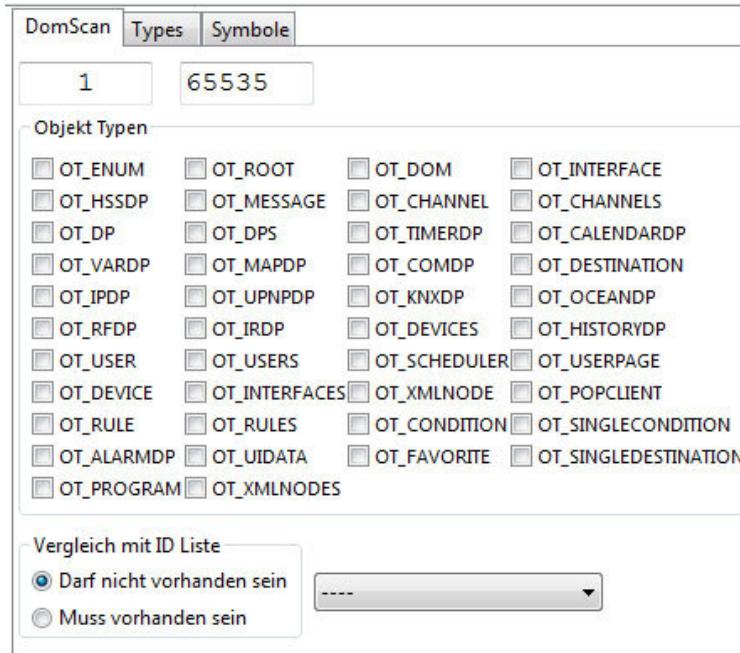
Auswahl der Aufzählungen (Räume, Gewerke, Favoriten, Interfaces , Systemvariablen sind bisher implementiert)

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 interface. On the left is a table of search results with columns: Method, ID, Name, TypeName, and Information. The table lists various objects, many of which are of type 'DEVICE'. On the right is the DomScan dialog, which is identical to the one shown in the first screenshot but with the '1-65535' range selected. The bottom of the dialog shows a 'Get' button and several checkboxes for filtering properties like 'UsedInScript', 'VisibleTrue', etc.

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 with the value 'VARDP' in the 'Value' column. The sidebar on the right lists various object types under 'Object Types'.

Metadate	Type	Value	Special	ID_ID	Name	TypeName	Information
*	integer	1	ActiveState	1704	ActiveState	VARDP	true
* Name	string			1781	ActiveState	VARDP	false
Type	integer	1003		1781	ActiveState	VARDP	true
TypeName	string	RASDP		2077	=ActiveState	VARDP	false
IsRoot	boolean	false		2214	=ActiveState	VARDP	false
Internal	boolean	false		2217	=ActiveState	VARDP	true
Visible	boolean	true		3401	=ActiveState	VARDP	true
Unselectable	boolean	false		3402	=ActiveState	VARDP	true
EncryptedData	string			3626	=ActiveState	VARDP	true
IPInfo	boolean	true		3723	=ActiveState	VARDP	true
IsLeaf	boolean	false		3920	=ActiveState	VARDP	true
* ValueName	string			4204	=ActiveState	VARDP	true
ValueIndex	integer	1	isvState	4205	=ActiveState	VARDP	true
ValueIndex0	integer	0		4206	=ActiveState	VARDP	true
ValueIndex1	integer	1		7141	=ActiveState	VARDP	true
ValueIndex2	integer	2		7398	New program=ActiveState	VARDP	false
ValueIndex3	integer	3		7399	New program=ActiveState	VARDP	false
ValueIndex4	integer	4		7377	New program=ActiveState	VARDP	false
Timestamp	time	2010-10-24 11:41:09		4870	=ActiveState	VARDP	true
LastTimestamp	time	1970-01-01 01:09:00		5852	=ActiveState	VARDP	true
ValueIndex5	string			6797	=ActiveState	VARDP	true
ValueIndex6	string	0		6930	=ActiveState	VARDP	true
ValueIndex7	string	1		7142	=ActiveState	VARDP	true
ValueIndex8	string	2		7398	New program=ActiveState	VARDP	false
ValueIndex9	string	3		7399	New program=ActiveState	VARDP	false
ValueIndex10	string	4		7377	New program=ActiveState	VARDP	false
D:\DomScan\Programme	string			6511	=ActiveState	VARDP	true

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

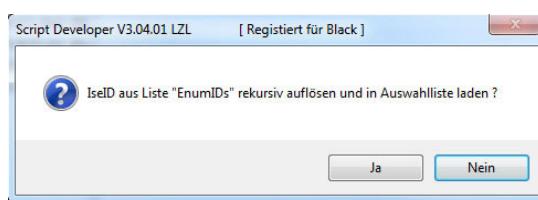
Weiterhin ist auch rekursives Arbeiten nun möglich

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

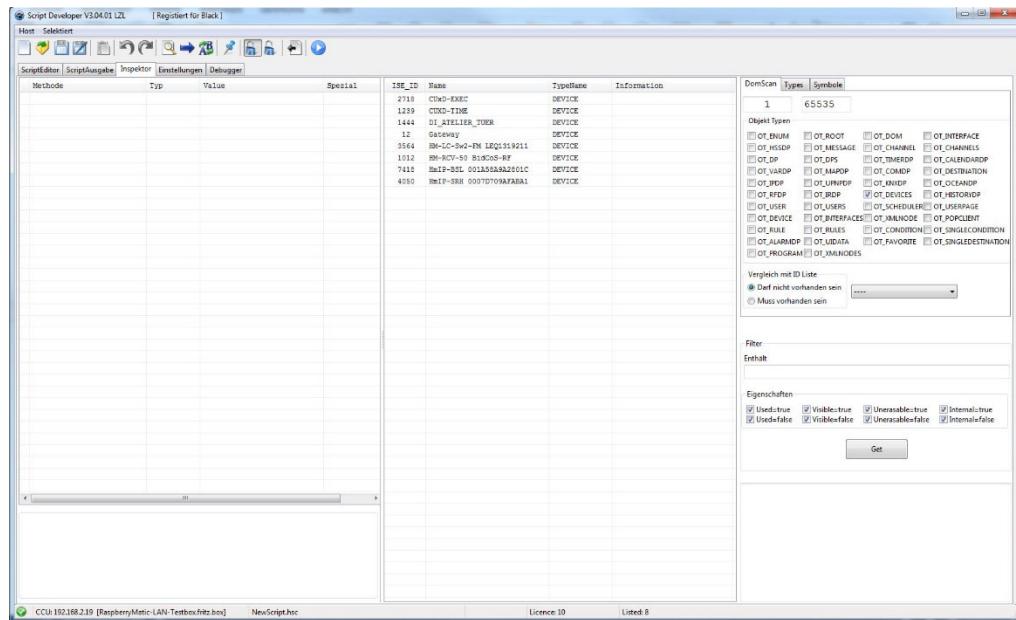
The screenshot shows the Script Developer interface with a search results table and a sidebar. The table has columns: Metadate, Type, Value, Special, ID_ID, Name, TypeName, and Information. One row is selected with the value 'Root Device' in the 'Value' column. The sidebar on the right lists various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

Metadate	Type	Value	Special	ID_ID	Name	TypeName	Information
*	integer	3	Root devices	19	Root devices	DEVICE	
* Name	string			1229	DEVICE	CLOUD-TIME	
Type	integer	19		1444	DEVICE	DL_FILTERED_T	
TypeName	string	DEVICES		12	DEVICE		
IsRoot	boolean	true		3166	DEVICE	EN-IC-2w-7W	
Internal	boolean	false		1612	DEVICE	EN-BCV-80-Bi	
Visible	boolean	true		7419	DEVICE	EN13-B2B_901	
Unselectable	boolean	false		6533	DEVICE	EN13-SR8_901	
EncryptedData	string						
Comments	string						

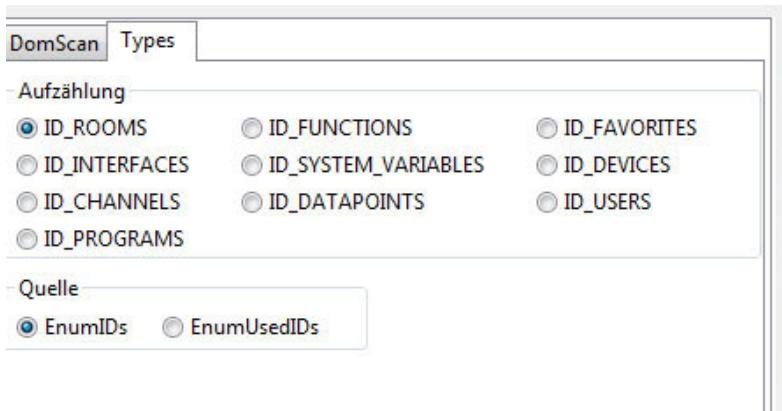
Das „R“ in der ersten Spalte zeigt an, dass ein Rekursiver Aufruf möglich ist. Doppelklicken auf die Zeile öffnet eine Sicherheitsabfrage



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



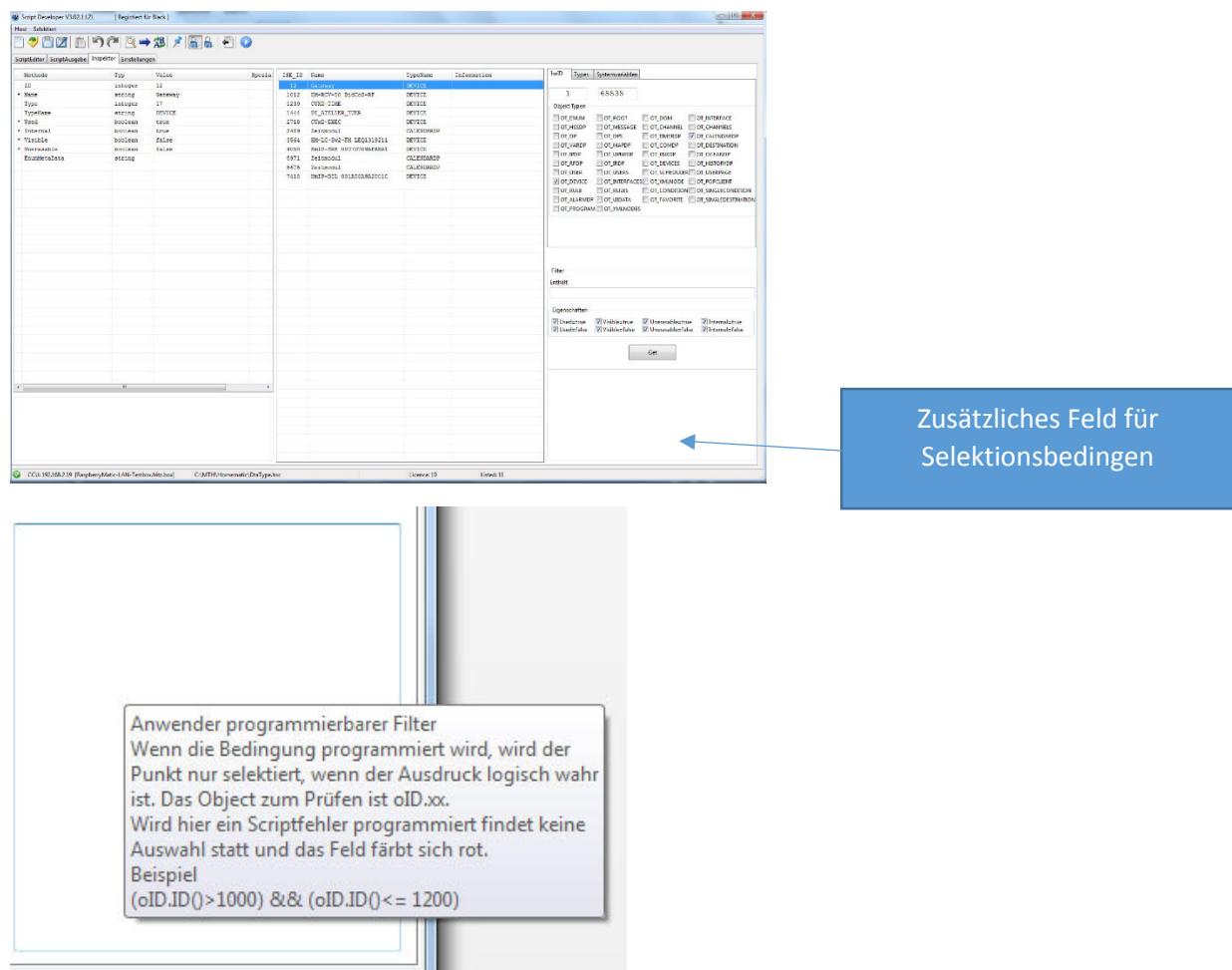
4.2 Selektionskriterium Types



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)



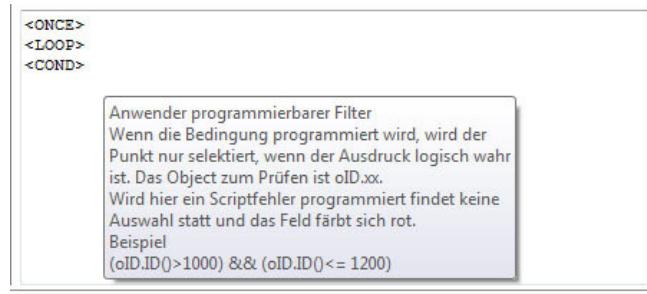
Filter sind ein mächtiges Werkzeug zum komplexen Eingrenzen und für Komplexe Abfragen.

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:

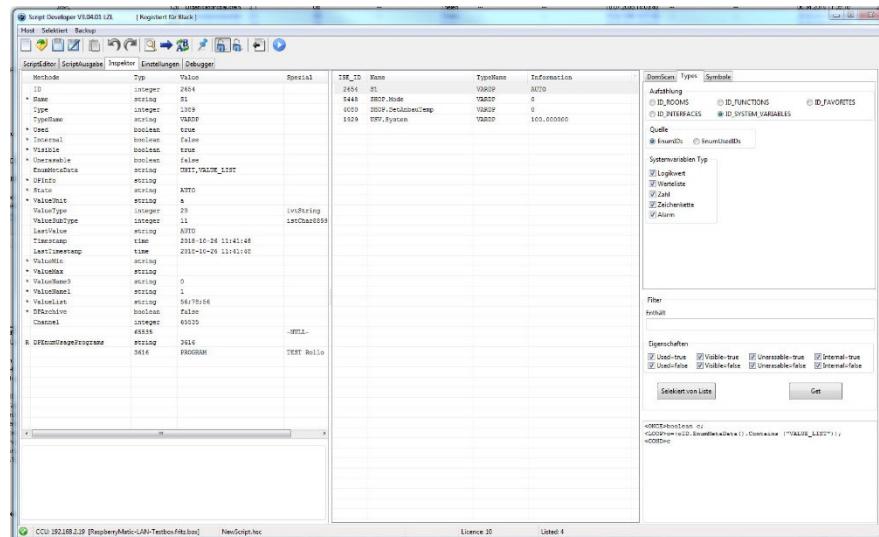
```
<ONCE>boolean c;
<LOOP>c=(oID.EnumMetaData().Contains ("VALUE_LIST"));
<COND>c
```

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

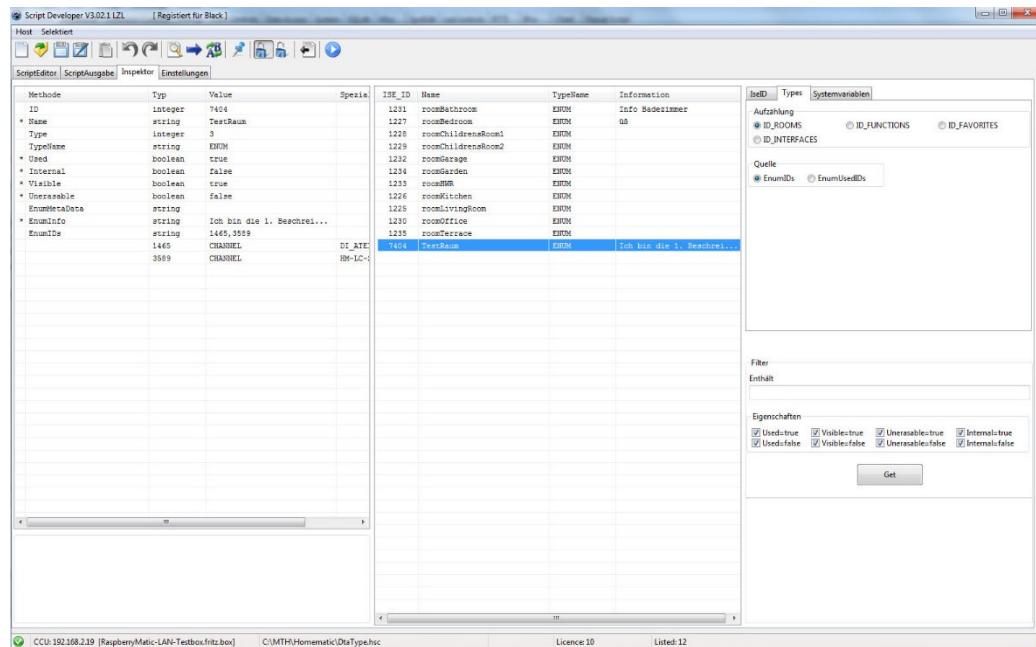
SYSVAR_VerwaisterChannel – Filter um Systemvariablen zu finden, deren Channel verweis in Nirvana zeigt

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.



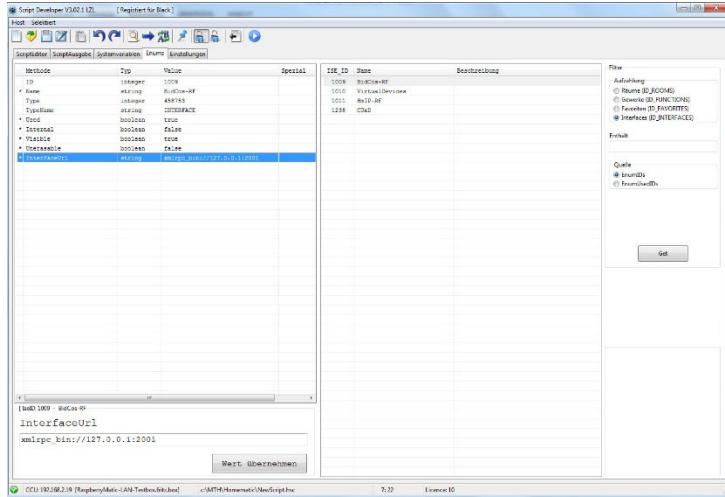
Die Spaltenbreite kann sowohl in Normalansicht als auch in Maximize separat eingestellt werden
(Das Programm sollte sich die Breiten merken und je nach Darstellungsart automatisch wieder einstellen, sollte...)

Dargestellt werden die Methode, der Vartype und die Property.

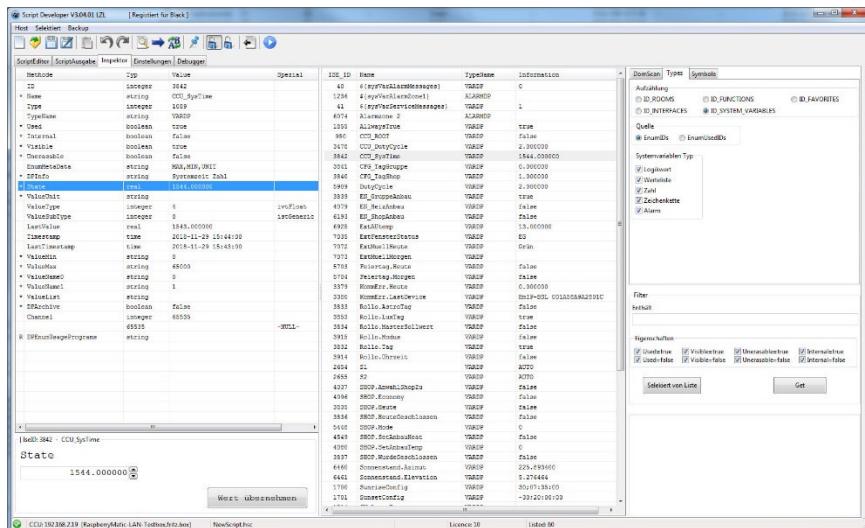
Bei den Aufzählungen wird jeweils eine Rekursionsstufe aufgelöst, um an die Detailinformationen zu kommen. Hier die Liste der Channels, die diesen Raum verwenden, aufgelöst in die ID, der Typ (hier Channels und der Name des Channels)

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

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.

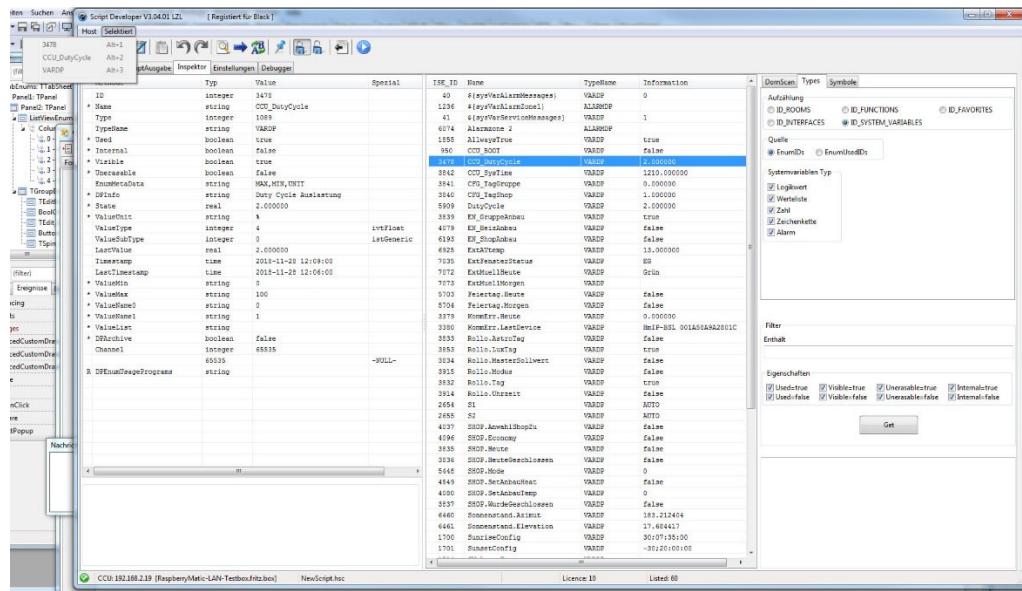


4.4 Daten aus Inspektor in Editor übernehmen

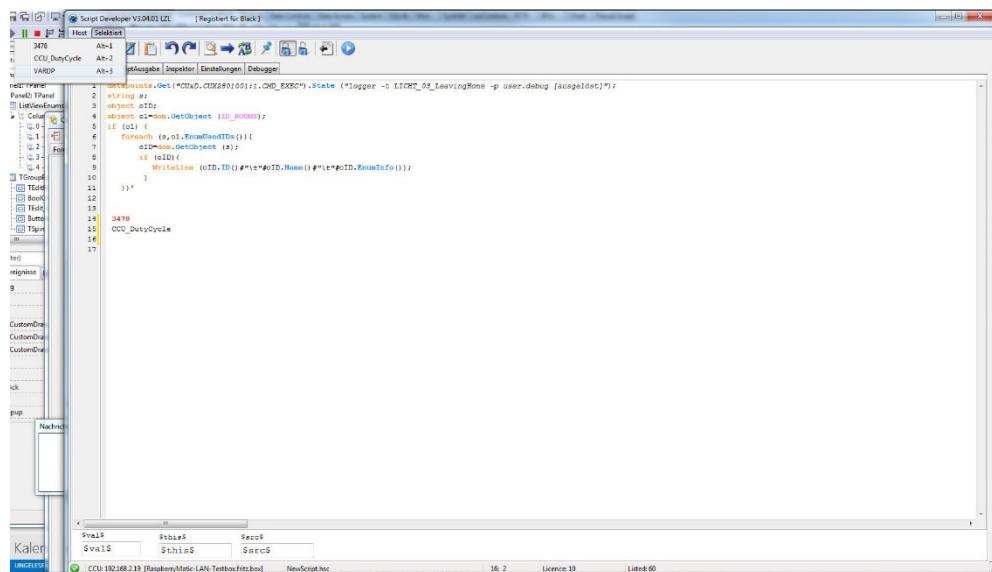
Zur schnelleren und auch möglichst fehlerfreien Bearbeitung besteht die Möglichkeit, Daten aus dem Inspektor direkt in den Editor zu übernehmen.

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

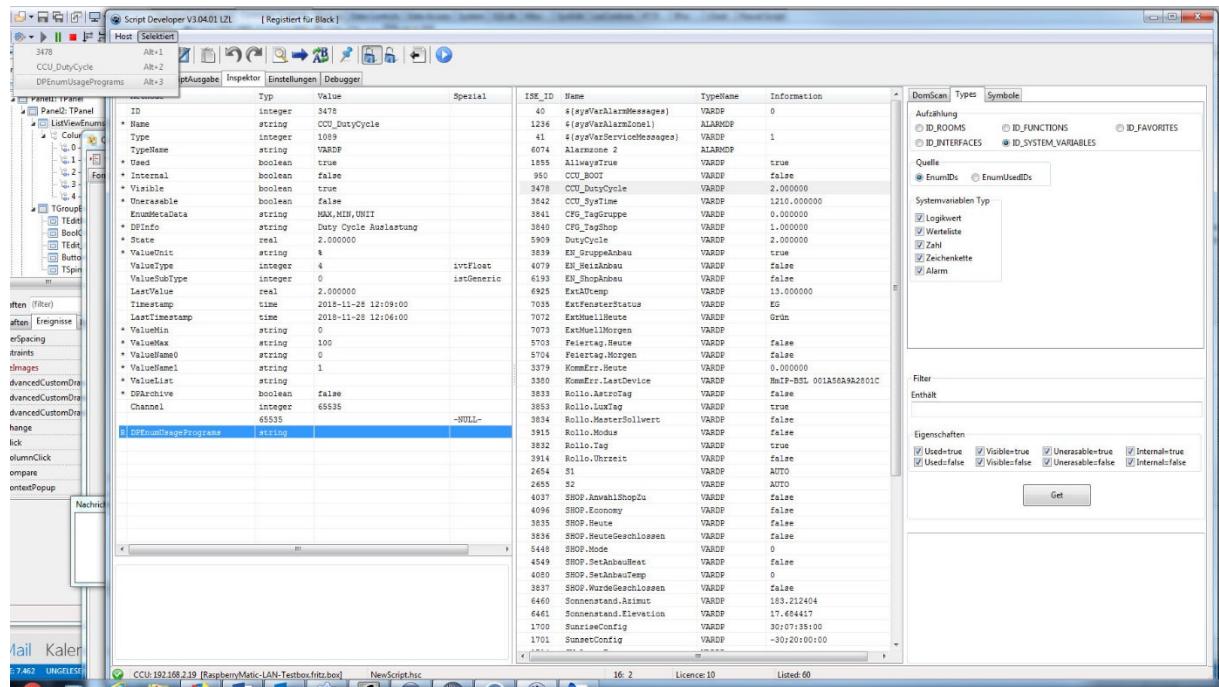
Mit den Tasten lassen sich folgende Spalteninhalte, welche vorher im Detailfeld selektiert worden sind, in den Editor übernehmen

Alt+M: Inhalt von Methode

Alt+T: Inhalt von typ

Alt+V: Inhalt von Value

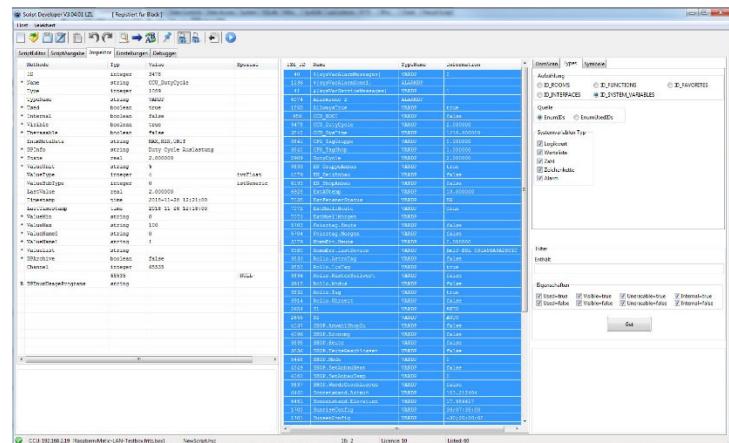
Alt+S: Inhalt von Spezial



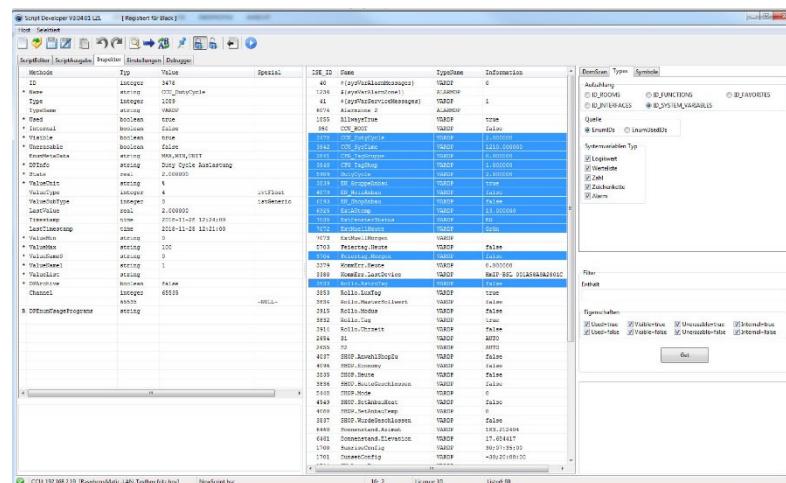
4.4.1 Mehrfachauswahl als Enum String

Es lassen sich im Hauptauswahlfeld Mehrfachselektionen vornehmen.

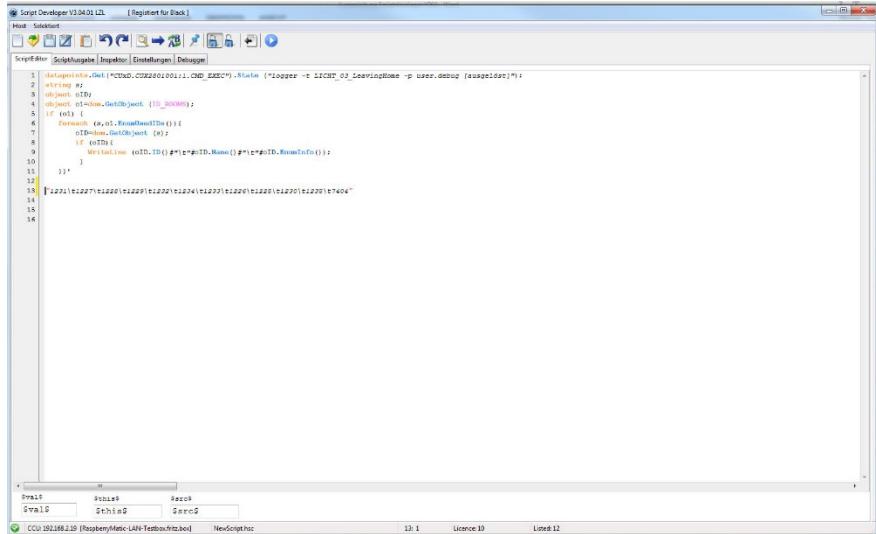
Ctrl-A : alle auswählen.



Oder die Übliche Mausbedienung:



Mit der  Taste 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)



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,

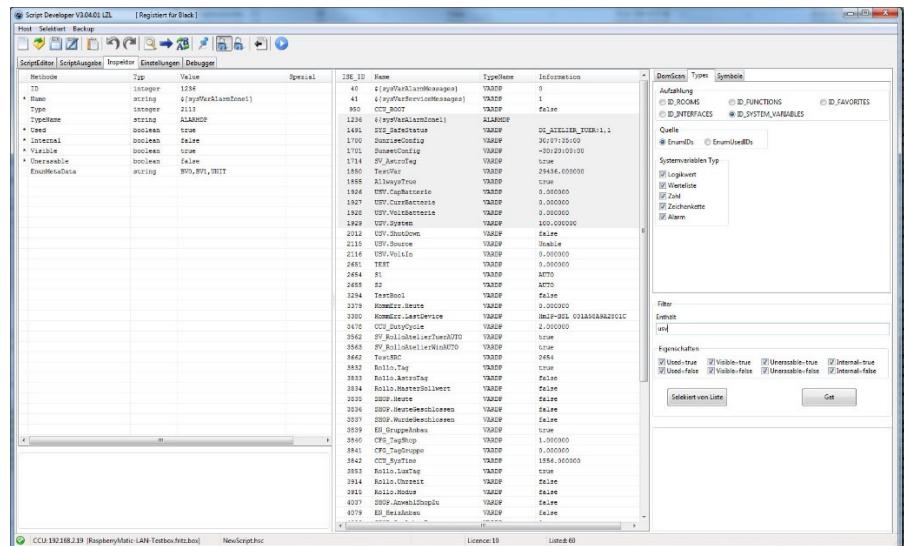


dann befinden sich die Selektierten Elemente wieder im Mittleren Feld.

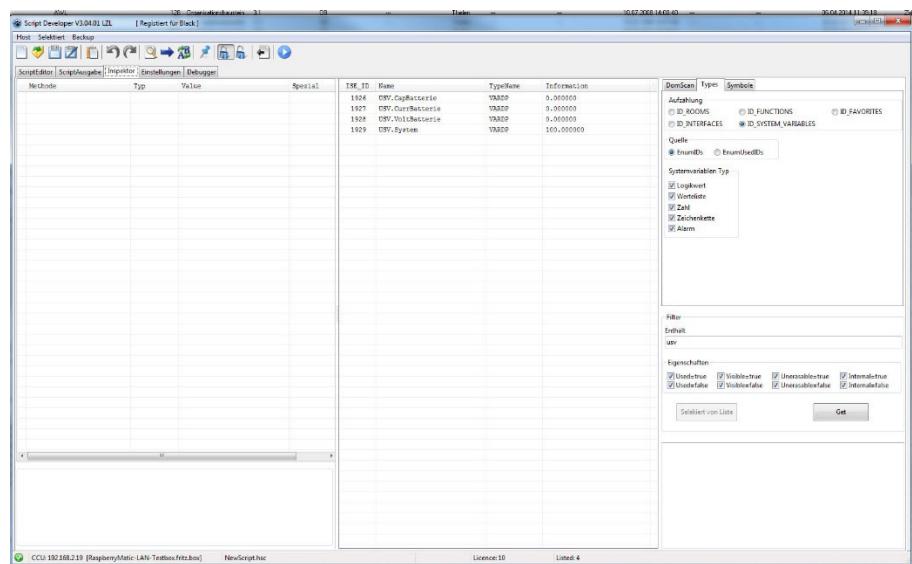
4.5 Selektion von Selektion

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

Hier Beispiel



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

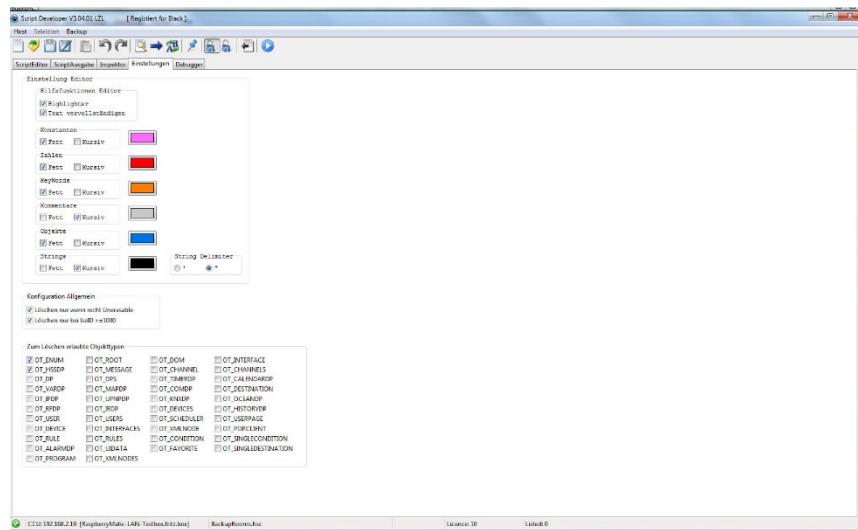


4.6 Objekte löschen

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

Redo geht nur über restore !

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



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

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

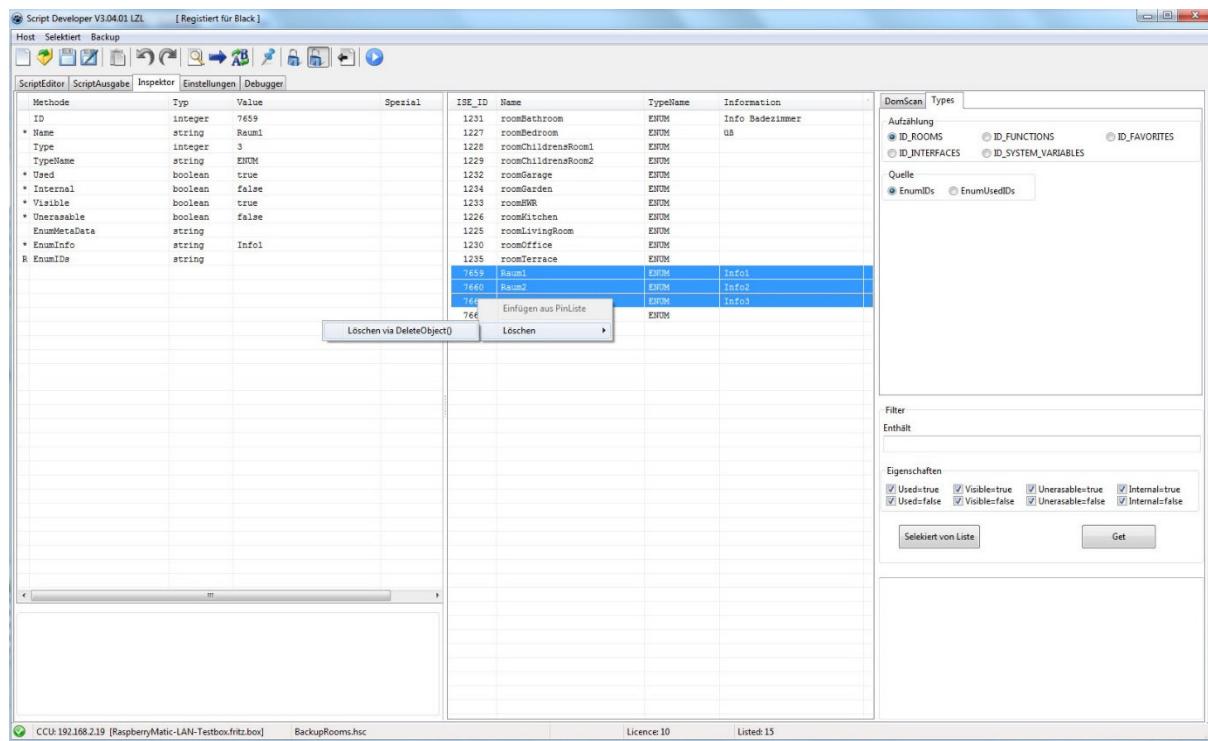
Im Inspektor unter Detailsview die Property entfernen (gilt nur für das Objekt), oder hier den Haken wegmachen (gilt für alle)

Löschen nur wenn ID>=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.

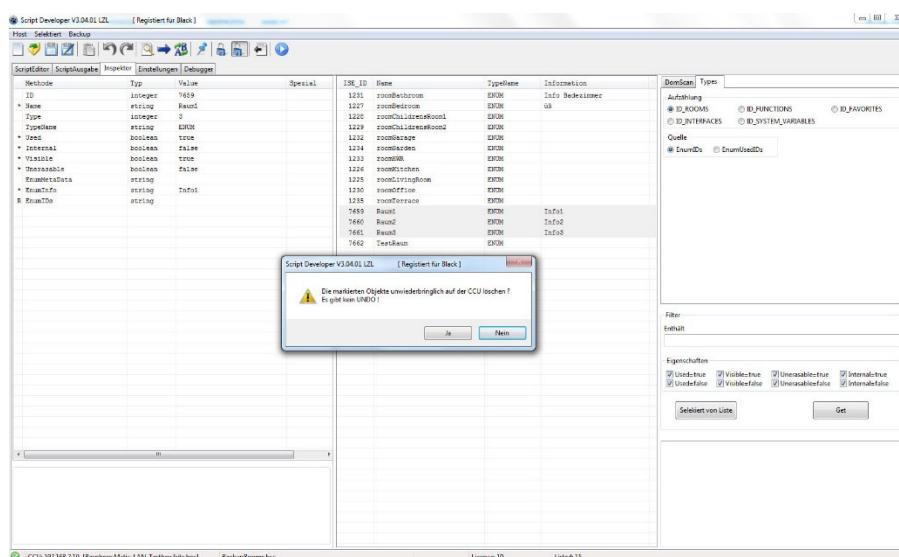
Löschbare Objekttypen. Die Letzte Sicherheit: ein zu lösches Objekt muss einen hier gechecked Objekttyp haben, sonst wird es nicht gelöscht.

Löschen läuft so ab:



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

Mehrfachselektion ist möglich

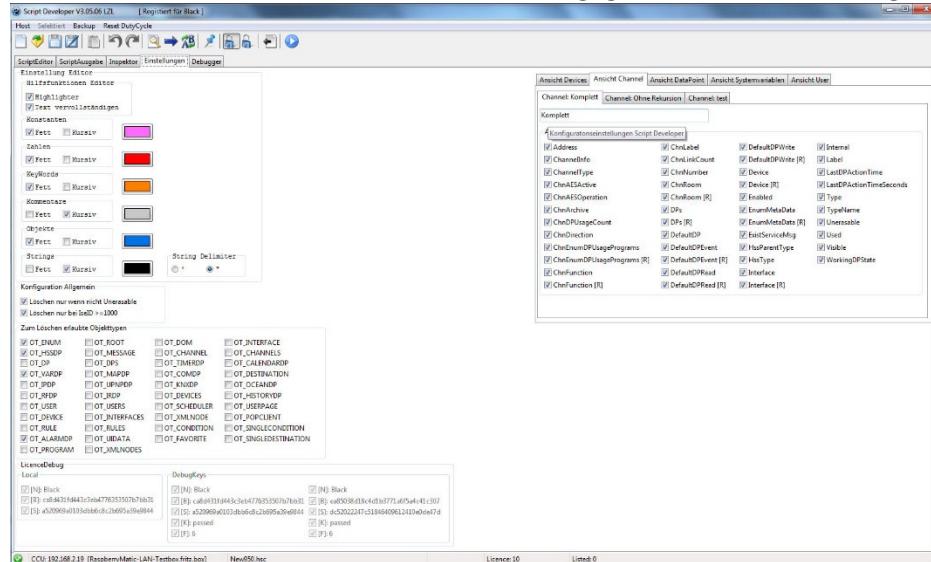


Nach dieser Sicherheitsabfrage sind die Objekte dann weg.. Zurück geht's dann nur mit restore.

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.

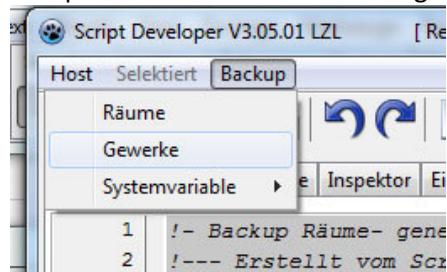
Die Entstellungen werden beim Verlassen gespeichert. Die jeweils geöffnete Sicht wird dann für das gefundene Objekt angewendet.

5 Backups

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

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

Den passenden Lizenzlevel vorausgesetzt, findet sich die Backups hier:



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

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

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)

Methoden	Typ	Value	Spezial	ISE_ID	Name	TypeName
	HSSDP		HmIP-RF.001A58A	2782	Hm-RC-19 CTIX2801001:3	CHANNEL
ChnDPUseageCount	integer	0		2807	Hm-RC-19 CTIX2801001:4	CHANNEL
ChnLinkCount	integer	0		2832	Hm-RC-19 CTIX2801001:5	CHANNEL
R ChnRoom	string			2857	Hm-RC-19 CTIX2801001:6	CHANNEL
R ChnFunction	string			2882	Hm-RC-19 CTIX2801001:7	CHANNEL
R DPs	idarray	7557,7558,7559,7560,756...	HmIP-RF.001A58A	2907	Hm-RC-19 CTIX2801001:8	CHANNEL
	HSSDP			2932	Hm-RC-19 CTIX2801001:9	CHANNEL
	HSSDP		HmIP-RF.001A58A	2957	Hm-RC-19 CTIX2801001:10	CHANNEL
	HSSDP		HmIP-RF.001A58A	2982	Hm-RC-19 CTIX2801001:11	CHANNEL
	HSSDP		HmIP-RF.001A58A	3007	Hm-RC-19 CTIX2801001:12	CHANNEL
	HSSDP		HmIP-RF.001A58A	3032	Hm-RC-19 CTIX2801001:13	CHANNEL
	HSSDP		HmIP-RF.001A58A	3057	Hm-RC-19 CTIX2801001:14	CHANNEL
	HSSDP		HmIP-RF.001A58A	3082	Hm-RC-19 CTIX2801001:15	CHANNEL
	HSSDP		HmIP-RF.001A58A	3107	Hm-RC-19 CTIX2801001:16	CHANNEL
	HSSDP		HmIP-RF.001A58A	3565	SB_Schalter:r0	CHANNEL
	HSSDP		HmIP-RF.001A58A	3589	Hm-IC->o2-FM LEQ1319211:1	CHANNEL
	HSSDP		HmIP-RF.001A58A	3595	Hm-IC->o2-FM LEQ1319211:2	CHANNEL
	HSSDP		HmIP-RF.001A58A	4051	HmIP-SR1 0007D709AFAB1:0	CHANNEL
	HSSDP		HmIP-RF.001A58A	4077	HmIP-SR1 0007D709AFAB1:1	CHANNEL
	HSSDP		HmIP-BST_001A58A2801C1	7419	HmIP-BST_001A58A2801C1	CHANNEL
	HSSDP		HmIP-BST_001A58A2801C1	7444	HmIP-BST_001A58A2801C1:1	CHANNEL
	HSSDP		HmIP-BST_001A58A2801C1	7447	HmIP-BST_001A58A2801C1:2	CHANNEL
	HSSDP		HmIP-BST_001A58A2801C1	7450	HmIP-BST_001A58A2801C1:3	CHANNEL
	HSSDP		HmIP-BST_001A58A2801C1	7455	HmIP-BST_001A58A2801C1:4	CHANNEL
	HSSDP		HmIP-BST_001A58A2801C1	7461	HmIP-BST_001A58A2801C1:5	CHANNEL
	HSSDP		HmIP-BST_001A58A2801C1	7467	HmIP-BST_001A58A2801C1:6	CHANNEL
	HSSDP		HmIP-BST_001A58A2801C1	7473	HmIP-BST_001A58A2801C1:7	CHANNEL
	HSSDP		HmIP-BST_001A58A2801C1	7482	HmIP-BST_001A58A2801C1:8	CHANNEL
	HSSDP		HmIP-BST_001A58A2801C1	7495	HmIP-BST_001A58A2801C1:9	CHANNEL
	HSSDP		HmIP-BST_001A58A2801C1	7508	HmIP-BST_001A58A2801C1:10	CHANNEL
	HSSDP		HmIP-BST_001A58A2801C1	7521	HmIP-BST_001A58A2801C1:11	CHANNEL
	HSSDP		HmIP-BST_001A58A2801C1	7530	HmIP-BST_001A58A2801C1:12	CHANNEL
	HSSDP		HmIP-BST_001A58A2801C1	7543	HmIP-BST_001A58A2801C1:13	CHANNEL
				7554	HmIP-BST_001A58A2801C1:14	CHANNEL
				7569	HmIP-BST_001A58A2801C1:15	CHANNEL
				7587	HmIP-SPI_00018A89B94F02:0	CHANNEL
				7615	HmIP-SPI_00018A89B94F02:1	CHANNEL
				7623	HmIP-SI0_00058A9FA04:2	CHANNEL
				7625	HmIP-SI0_00058A9FA04:0	CHANNEL
				7648	HmIP-SI0_00058A9FA04:1	CHANNEL
				7657	HmIP-SI0_00058A9FA04:2	CHANNEL
				7658	HmIP-SI0_00058A9FA04:3	CHANNEL

Die Werte lassen sich dann über PutParamset verändern in einem Script

6.3

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

Absicht Devices | Absicht Channel | Absicht DataPoint | Absicht Systemvariablen | Absicht User

Datapoint: Komplett | Datapoint: Ohne Rekursion | Datapoint: Normal

Ohne Rekursion Konfigurationseinstellungen Script Developer

Absicht Methoden

- Channel
- Channel [R]
- DPEnumUsagePrograms
- DPEnumUsagePrograms [R]
- DPInfo
- DPUsageCount
- Enabled
- EnumMetaData
- EnumSpecialIDs
- HSSAddress
- HSSID
- HssType
- Internal
- LastTimestamp
- LastTimestampSeconds
- LastTimestampSeconds
- LastName
- LastTriggerTime
- LastTriggerTimeSeconds
- LastValue
- Operations
- Timestamp
- TimestampSeconds
- Type
- TypeName
- Used
- Unerasable
- Value
- Visible

Konfiguration Datapoints

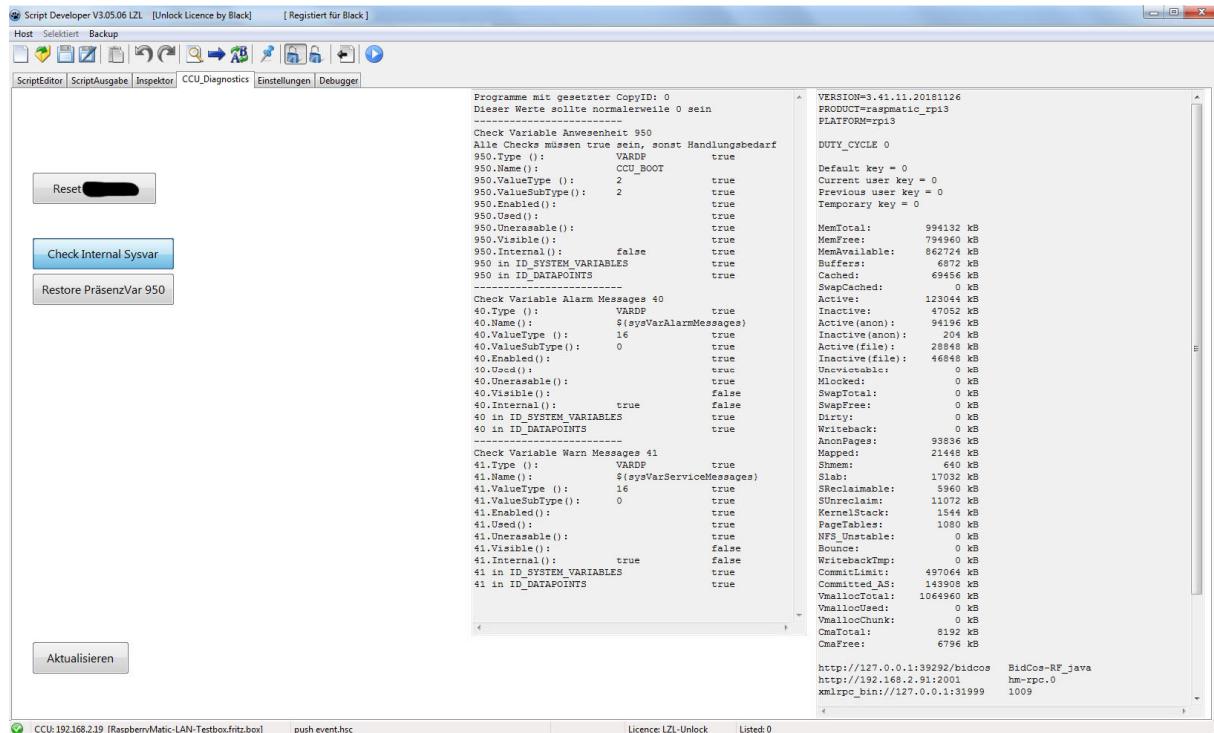
Send to DP via xmlrpc http://127.0.0.1:1999/ event

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

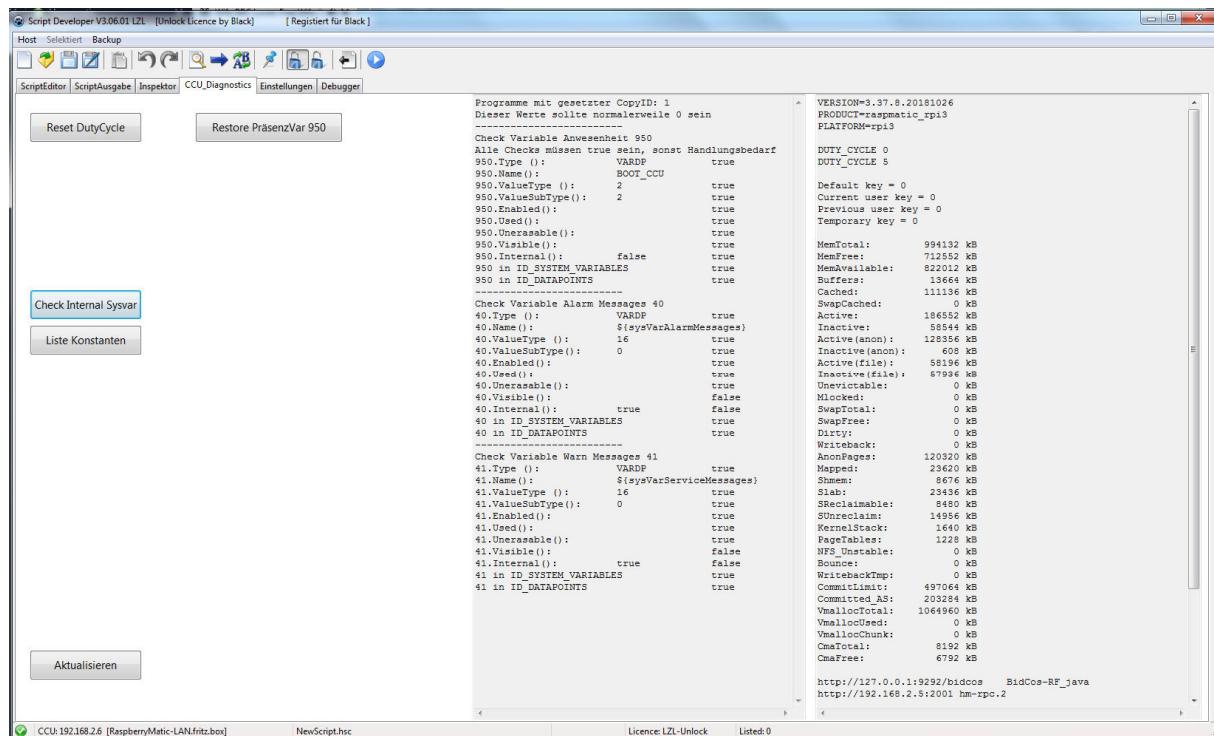
Hat der DP die Eigenschaft Write, wird ein ganz normales State (xxx) ausgeführt. Hat er diese nicht, macht der SDV bei Wertänderung von sich aus ein Rega Event auf den DP.

(Wenn die Freigabebedingungen passen)

7 Diagnosebild



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



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

Alphabetische Auflistung aller bisher bekannten Systemkonstanten mit ihren Werten

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

- ScriptDeveloper V3.06.01 L2L [Unlock Licence by Black] [Registiert für Black]**
- Host Selektiert Backup**
- ScriptEditor ScriptAusgabe Inspektor CCU_Diagnostics Einstellungen Debugger**
- Buttons:** Reset DutyCycle, Restore PräsenzVar 950, Check Internal Sysvar, Liste Konstanten, Aktualisieren.
- Text Area (List of Constants):**

```
ID_ALARM_MAPS : 700
ID_ALARMS : 701
ID_CALENDARPS : 34
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 : 850
ID_EOCLEAN_BEGIN : 851
ID_EOCLEAN_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_DP_GW : 39
ID_LINKS : 301
ID_MESSAGES : 31
ID_PRESENCE_SIMULATION : 19
ID_PRESENT : 150
ID_PROGRAMS : 15
ID_RF : 900
ID_RF_BEGIN : 901
ID_RF_LASTREC : 942
ID_ROUTES : 101
ID_SOT : 2
ID_RULES : 33
ID_RUNTIMECONFIG : 21
ID_SCENES : 401
ID_SCONDITIONS : 36
ID_SDESTINATIONS : 38
ID_SFUSER_DF : 945
ID_SERVICES : 28
ID_STRUCTURE : 6
ID_SYSTEM_VARIABLES : 27
ID_UL_DATAS : 32
ID_UNP : 750
```
- Text Area (Memory Statistics):**

```
VERSION=3.37.8.20181026
PRODUCT=raspberry_rp13
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: 710532 kB
MemAvailable: 652004 kB
Buffers: 13644 kB
Cached: 111136 kB
SwapCached: 0 kB
Active: 186552 kB
Inactive: 53444 kB
Active(anon): 128936 kB
Inactive(anon): 608 kB
Active(file): 58196 kB
Inactive(file): 67934 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: 8676 kB
Slab: 25416 kB
ShReclaimable: 5480 kB
ShUnreclaim: 14956 kB
KernelStack: 1640 kB
PageTables: 1228 kB
NFS_Unstable: 0 kB
Buddy: 0 kB
WritebackTmp: 0 kB
CommitLimit: 497064 kB
Committed_AS: 203284 kB
Vmalloc_CTotal: 1064960 kB
Vmalloc_CUsed: 0 kB
Vmalloc_CChunk: 0 kB
ComFree: 8192 kB
OmaFree: 6792 kB

http://127.0.0.1:9292/bidcos BidCos-RF_java
http://192.168.2.8:2001 hm-rpc.2
```
- Bottom Status Bar:** CCU:192.168.2.8 [RaspberryMatic-LAN.fritz.box] NewScript.hsc Licence:L2L-Unlock Listed:0