

Kurzanleitung Scriptdeveloper V3.07.11 (22. März 2019)

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

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

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

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

Dies ist eine BetaTestversion.

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

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

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

Inhalt

Kurzanleitung Scriptdeveloper V3.07.11 (22. März 2019)	1
1. Installation.....	4
1.1 Lizensierung.....	6
1.2 Donate	11
1.3 Systemvoraussetzungen.....	12
1.4 Was tut's bis jetzt	12
1.5 Bekannte Einschränkungen / Bugs.....	13
1.6 Geänderte Anforderung an Auflösung	13
1.7 Changelog	14
1.7.1 Changelog 03.07.11 LZL.....	14
1.7.2 Changelog 03.07.10 LZL.....	14
1.7.3 Changelog 03.07.07 LZL.....	14
1.7.4 Changelog 03.07.05 LZL.....	14
1.7.5 Changelog 03.07.04 LZL.....	14
1.7.6 Changelog 03.07.02 LZL.....	14
1.7.7 Changelog 03.07.01 LZL.....	15
1.7.8 Changelog 03.06.06 LZL.....	15
1.7.9 Changelog 03.06.04 LZL.....	15
1.7.10 Changelog 03.06.01 LZL.....	15
1.7.11 Changelog 03.05.06 LZL.....	15
1.7.12 Changelog 03.05.01 LZL.....	15
1.7.13 Changelog 03.04.01 LZL.....	16
1.7.14 Changelog 03.03.01 LZL.....	16
1.8 Neue Authentifizierungsmechanismen Raspberrymatik	17
1.8.1 SSL.....	17
1.8.2 Authentifizierung über Nutzername / Passwort.....	17
2. Oberfläche	19
2.1 Word Markup (seit 03.07.02).....	20
3 Scripteditor	21
3.1 Voreinstellungen Editor.....	23
3.2 Vervollständigen Funktion	24
3.3 Methodenhilfe.....	25
4 Inspektor.....	26
4.1 Selektionswahl: DomScan	27
4.2 Selektionskriterium Types	30

4.3 Zusätzliche Selektionsbedingungen	31
4.4 Daten aus Inspektor in Editor übernehmen.....	38
4.4.1 Mehrfachauswahl als Enum String	40
4.4.2 Übernahme von einem Script aus einem Programm direkt in den Editor	42
4.5 Selektion von Selektion	45
4.6 Objekte löschen.....	46
4.7 Anwenderdefinierte Sichten	48
4.8 Browsing durch Rekursionsebenen.....	49
4.9 Auflösen komplexer Objekte.....	52
4.9.1 Auflösen von Devices	52
4.9.2 Auflösen von Programmen.....	53
4.10 Zeitmodule	54
4.11 Suchen in Scripten nach Variablen, Devices etc.....	55
4.12 Volltextsuche in Scripten.....	58
4.13 SingleDestinations in ihrer Reihenfolge ändern.....	60
5 Backups.....	63
5.1 Räume.....	63
5.2 Gewerke	63
5.3 Systemvariablen	64
5.4 Devices und Kanäle.....	64
6 Kleine Helfer im Alltag	65
6.1 Umbenennen von Kanälen von Geräten	65
6.2 Paramset Master	66
6.3 Rega Push auf Datenpunkte via Rega event.....	66
7 Diagnosebild	68
7.2 Programmanalyse.....	70
7.3 Systemvariablen Metadatenkonsistenz	74
7.4 Systemvariablen Test auf NirvanaChannelBezüge	76
7.5 Check Internal Sysvar	78
8 License Disclaimer	79

1. Installation

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

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

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

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

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

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

[SECURITY]
HTTPREGASCRIPt=8181
HTTPREGAXMLRPC=1999
HTTPSREGASCRIPt=48181
HTTPSREGAXMLRPC=41999

[ENUM_NORM]                                ← Ab hier kommen dann interne Werte, Finger Weg
C1=65
C2=200
C3=293
C4=65

[ENUM_MAX]
C1=65
C2=200
C3=293
C4=65

Hinweis für CCU2 Nutzer oder Nutzer älterer Firmware: Wenn sich kein Anfrageschlüssel generieren lässt (keine Hashs erkannt), in der INI Datei prüfen, ob dort der entsprechenden
```

CCU der Eintrag USEHTTPS=True steht. Damit klappt es nicht. In diesem Fall muss der Eintrag in USEHTTPS=False geändert werden, dann klappt auch. Alte Firmwares mögen es auch nicht, wenn über Nutzernname und PW ein authentifizierter Zugriff über Name:PW@Host versucht wird. Bei älteren Firmware oder CCU keinen Nutzernamen oder Passwort eintragen bei CCU1 / CCU2 CCUHOST

Bei CCU2:

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

Bei CCU3:

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

BestandsNutzer:

Die Ini Datei hat sich ziemlich vergrößert. Hilfreich ist, in der mitgelieferten INI alle Schlüssel ab [CFG_ChView1_Methods]

Address=True.....

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

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

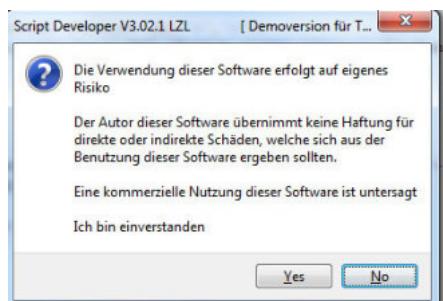
1.1 Lizenierung

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

Wie arbeitet das ?

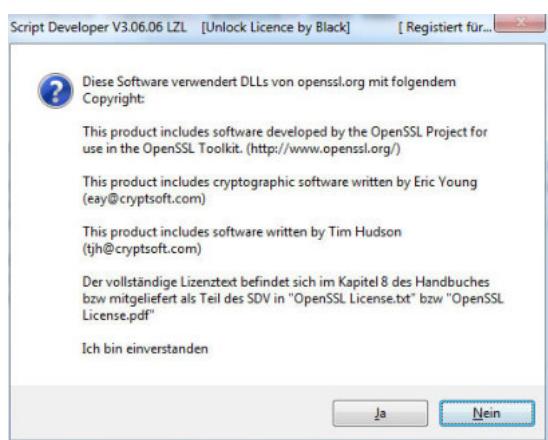
Der SDV telefoniert nicht nach Hause. Um eine Lizenz anzufragen ist folgender Weg einzuschlagen.

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



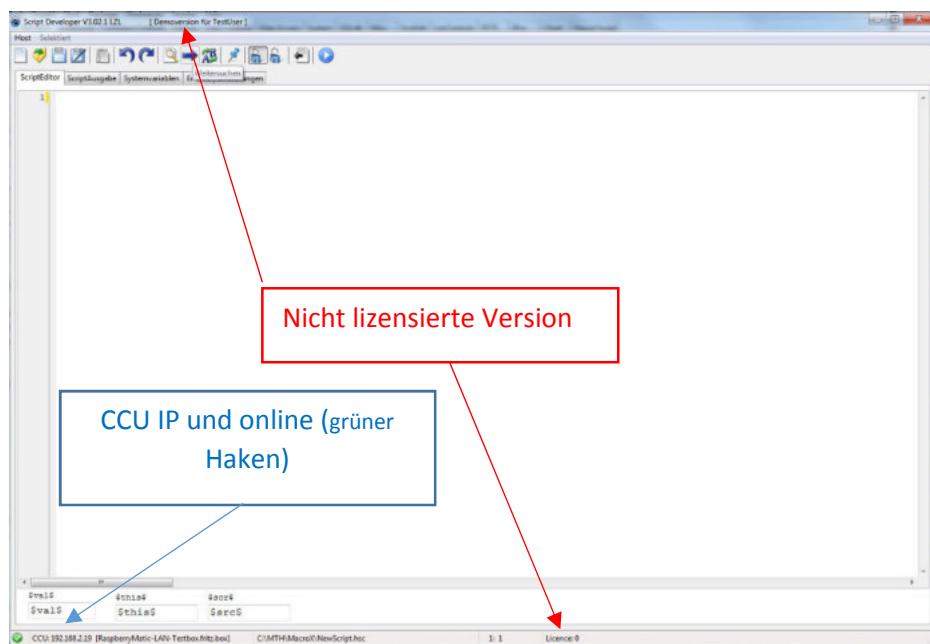
Bei allerersten Start muss dieses Fenster mit yes bestätigt werden. No führt so einem sofortigen Programmabbruch

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

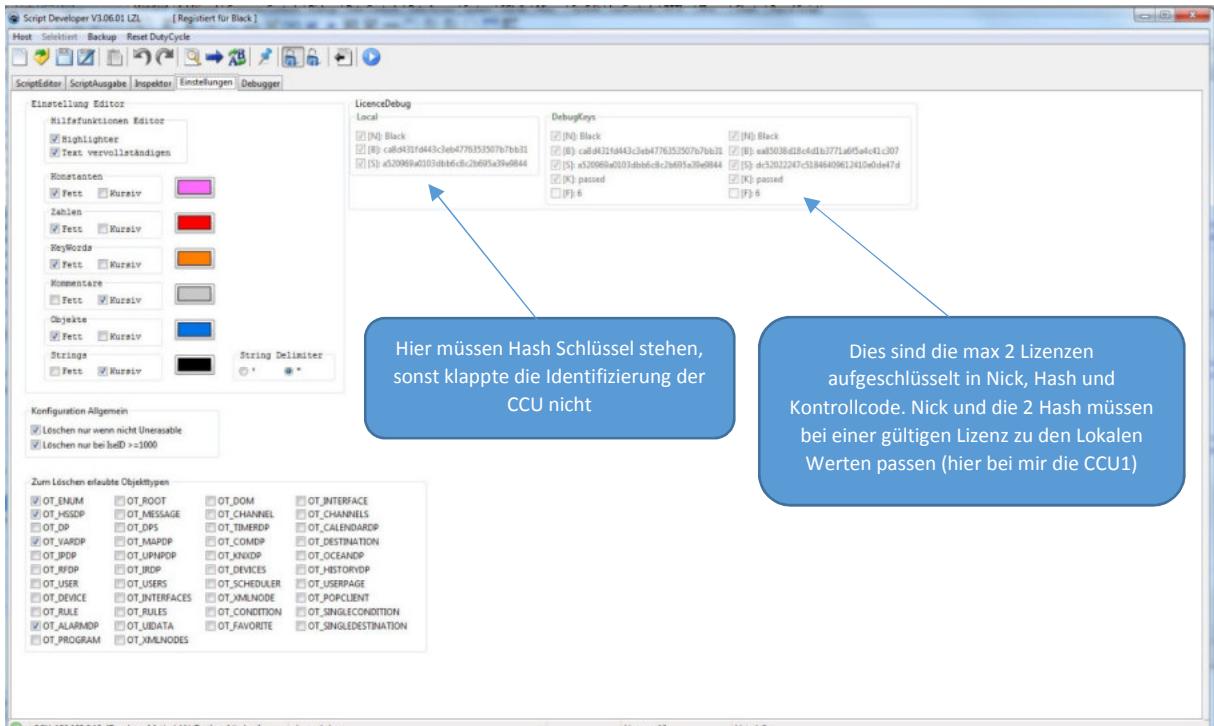


„Nein“ führt auch hier zu einem sofortigen Programmabbruch

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



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

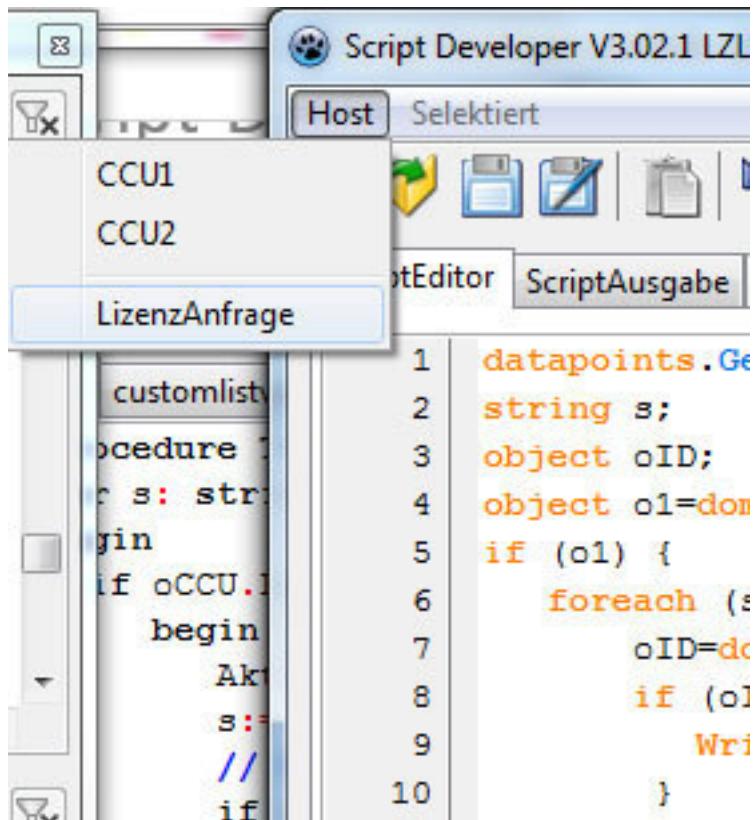


Aus einer Anfrage ohne unter local sinnige Einträge zu sehen lässt sich kein gültiger key generieren.

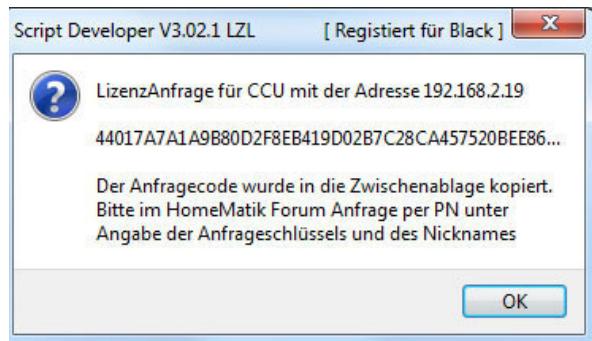
Aufgrund einer Programmungsgenauigkeit bei der Ausformulierung einer RegEx ist es leider sehr wahrscheinlich, dass alte Lizenzschlüssel vor 3.5.2 als ungültig erkannt werden. Sorry dafür, in dem Fall bitte die Anfrageschlüssel neu erstellen (wenn die Hashwerte gültig sind) und mir schicken. Der SDV 3.5.0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

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

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



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



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

Was enthält dieser Hexstring ?

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

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

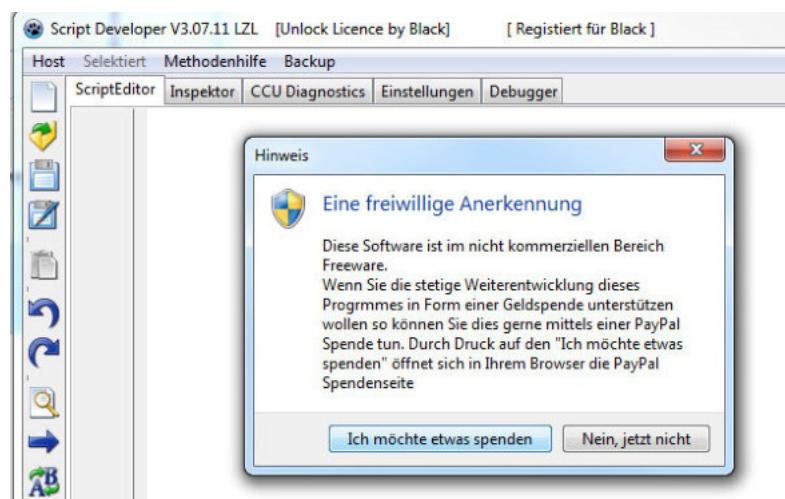
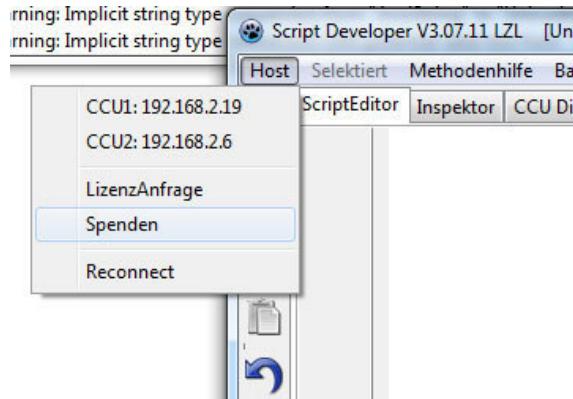
Wer mit diesem Verfahren nicht einverstanden ist, möge bitte an dieser Stelle die PDF Datei schließen und kann die Dateien beruhigt löschen.

Geplant hab ich folgende Lizenzabstufungen

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

1.2 Donate

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



1.3 Systemvoraussetzungen

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

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

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

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

1.4 Was tuts bis jetzt

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

Undo / Redo arbeiten

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

Enums und Sysvars arbeiten auch schon inkl Detaildaten und Editermöglichkeiten.

Darstellbarkeit zumindest der Grundmethoden aller Objekte

DomScan

Devices

Aufschlüsseln der MetaDaten

Datenpunkte

Kanäle

Programme

Favoriten

User

Paramset Master

Script Bearbeitung aus Programmen heraus

Verändert von Verzögert um von Scripten

Verändern von Retriggern

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

Destination und SingleDestination

MarkupUp im Editor

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

Suchen in Scripten nach Seriennummer von Devices und Channels

VollTextsuche in Scripten auch mit Regex Ausdrücken

1.5 Bekannte Einschränkungen / Bugs

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

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

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

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

1.6 Geänderte Anforderung an Auflösung

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

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

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

1.7 Changelog

1.7.1 Changelog 03.07.11 LZL

Kleine Bugfixes

Tab Scriptausgabe entfernt, dafür sitzt die Scriptausgabe nun in einem Fenster unter dem Editor (Grösse anpassbar)

Neue Keys eingeführt, um Script Run und Script Test auch über FKeys zu starten

Mit dieser Version kennt der SDV 620 Rega-Methoden und Konstanten

Methodenhilfe eingeführt

1.7.2 Changelog 03.07.10 LZL

Kleine Bugfixes

Properties von Zeitmodulen und Singledestination / Singleconditions über Dropboxes änderbar

Systemvariablen Metadaten Check und Reparatur eingeführt

Systemvariablen NirvanaChannel Check und Reparatur eingeführt

1.7.3 Changelog 03.07.07 LZL

Kleine Bugfixes

Bugfix: Alt-S aus dem Editor brachte nicht die Special Spalte: geändert in Alt-p

Ein Paar Methoden hinzugefügt

Reiter Scriptsuche fertiggestellt (ab level 6)

Damit nun möglich: Volltextsuche in allen Scripten

Selektierbar, auch suchen in GeisterDestinations möglich

Suchen auch als Regular Expression möglich

Beschleunigte Suche in Ram möglich

ProgrammObjektanalyse

1.7.4 Changelog 03.07.05 LZL

Kleine Bugfixes

Bugfix bei Programme Testen („\n“ in einem Kommentar führte zu einem Fehler) Bugfix tut nur auf RM, auf der die Rega bezüglich Doublequote gefixt wurde.

Im Editor folgende Tastenkürzel:

Ctrl F – Suchen Dialog, Ctrl R Replace Dialog, F3 – Weitersuchen

Regaports werden nun Automatisch ermittelt

Alternative Endungen für Scripte in INI hinzugefügt

Alternatives Scriptverzeichnis in INI hinzugefügt

1.7.5 Changelog 03.07.04 LZL

Auflösung geändert und ein wenig das Design

Programm Testen hinzugefügt

Suchen in Scripten nach Seriennummern von Devices / Kanälen

Ein paar weiterer Methoden eingepflegt

1.7.6 Changelog 03.07.02 LZL

Bugfix (Bei aufgelösten Programmen funktionierte das Rückschreiben des Scriptes in das Programm nicht)

BugFix Beim Ändern von Strings in der Detailansicht wurde der Ursprungswert nicht als Vorschlag genommen

Umstellen der Dialoge auf TTaskDialog und Rauswerfen der Unit Windows, damit ist die Tür in Richtung Linus wieder spaltbreit auf

Class TSynEditMarkupHighlightAllCaret ein wenig modifiziert, damit gehen nun Markups,

Setup für MarkUps und Anwählbar / Abwählbar

Suchen innerhalb von Scripten nach Namen von Systemvariablen, Geräten, Kanäle, Räumen und Gewerken

Reihenfolge von Singledestinations sind nun verschiebbar

1.7.7 Changelog 03.07.01 LZL

Einige kleine Bugfixes

Vervollständigen von Zeitmodulen (CalendarDP)

Aufschlüsseln von Programmen in Rules, Subrules, Condition, SingleCondition, Destination und Singledestinations

Detailansichten von Conditions, Single Conditions, Destination und Singledestinations inkl rekursiver Auflösung

Erweitern der Methoden im Highlighter

Klartextübersetzung der Regakonstanten in der Detailansicht

Verfeinern des Undo Stacks (Browsing durch Programme)

Hinzufügen von Methoden beim User Object (Level 7)

Eine interne Änderungen zur Vorbereitung von Backup / Restore von Programmen

1.7.8 Changelog 03.06.06 LZL

Ab dieser Version Unterstützt der SDV die Neuen Eigenschaften der Raspberrymatik / CCU3: SSL Verbindungen und Authentifizierung über Nutzername-Password. Dazu mehr in einem separaten Kapitel

Verbesserung des Undo Stacks im Inspektor

Kleinere interne Fixes und Veränderungen als Vorgriff auf die Kommenden Versionen

1.7.9 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

1.7.10 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)

1.7.11 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

1.7.12 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

1.7.13 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

1.7.14 Changelog 03.03.01 LZL

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

1.8 Neue Authentifizierungsmechanismen Raspberrymatik

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

1.8.1 SSL

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

Quelle: https://sourceforge.net/p/lazarus-ccr/s...en_ssl.pas.

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

http://packages.lazarus-ide.org/openssl-1.0.2j-x64_86-win64.zip für die 64 Bit Version
und
<http://packages.lazarus-ide.org/openssl-1.0.2j-i386-win32.zip> für die 32 Bit Version

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

1.8.2 Authentifizierung über Nutzername / Passwort.

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

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

Beispielhafter Aufbau der neuen INI

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

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

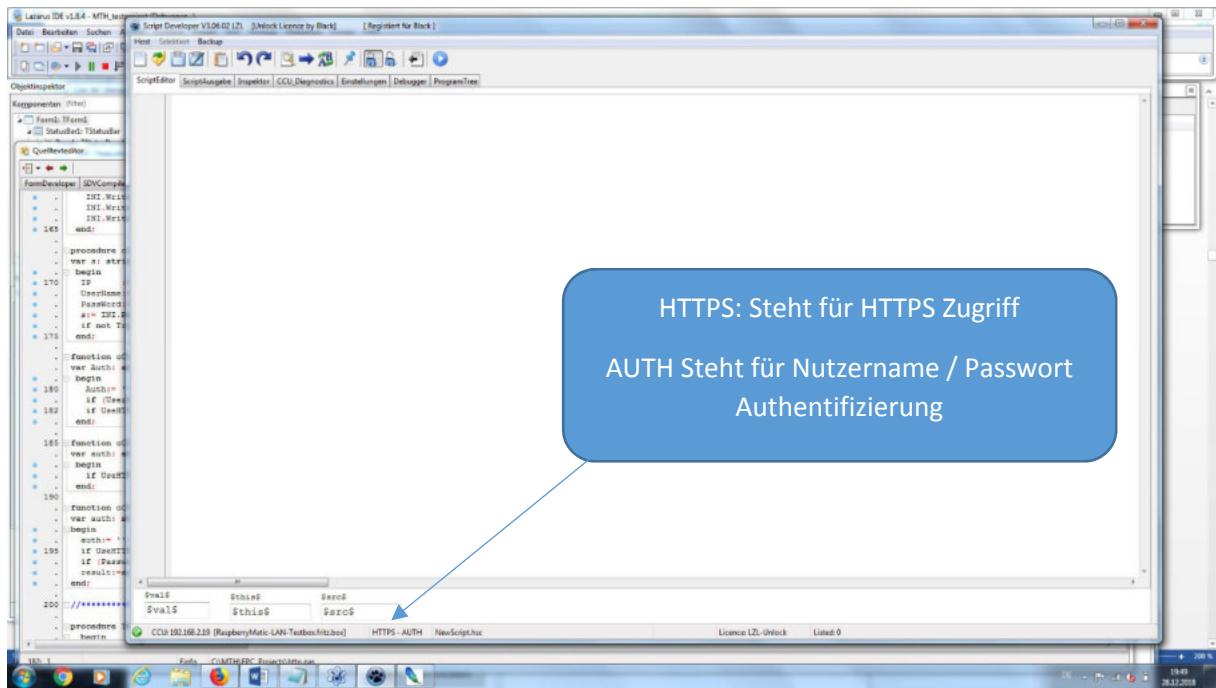
[CCU1]                                     Neuer Schlüssel
IP=192.168.2.19                           Ip Von CCU1
USERNAME=Admin                             Nutzernname auf der CCU1
PASSWORD=XXXXXXXXXXXX                      Passwort der Nutzers auf der CCU1
USEHTTPS=true                             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

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

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

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

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

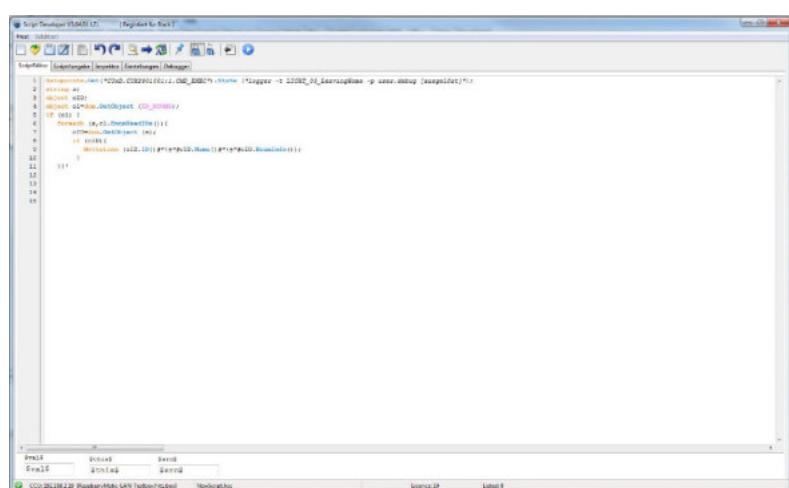
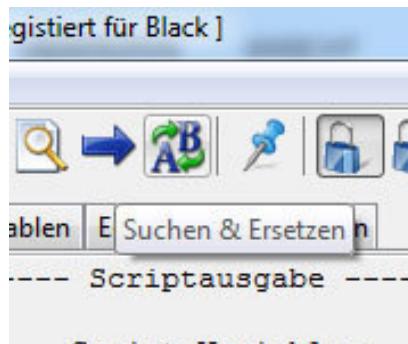


Ein Debugger Breakpoint im SDV zeigt hier den zusammengesetzten HostString, der bei HTTPS und Authentifizierung benutzt wird



2. Oberfläche

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



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

In der Statuszeile finden sich Information über:

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

Folgende Shortcuts wurden implementiert:

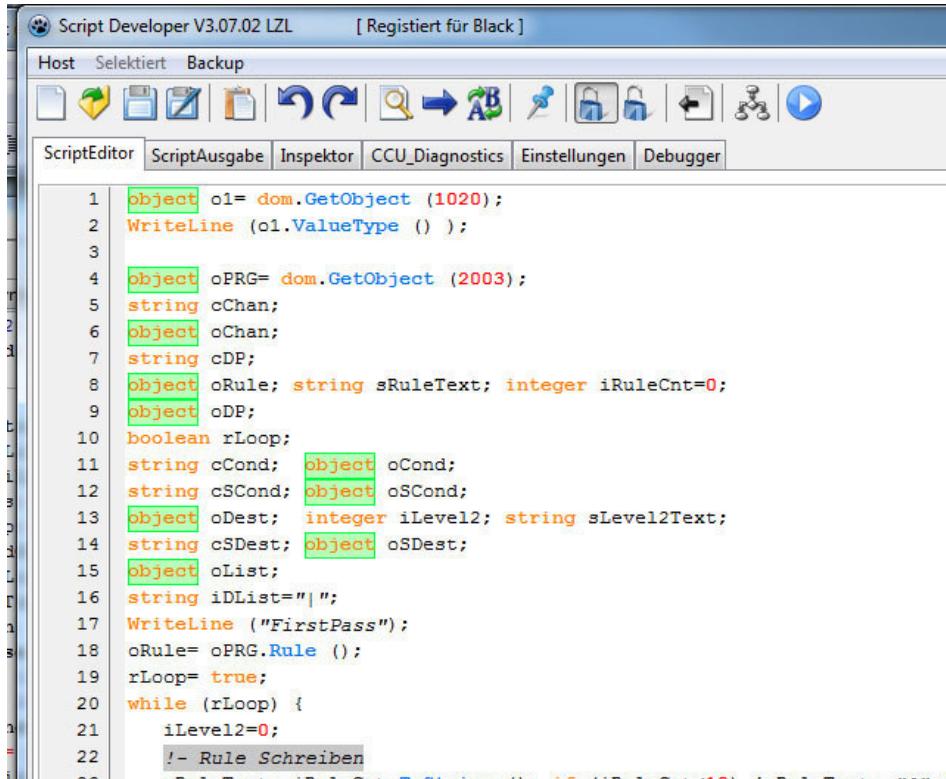
Ctrl-F: Suchen Dialog

Ctrl-R: Ersetzen Dialog

F3: Weitersuchen

2.1 Word Markup (seit 03.07.02)

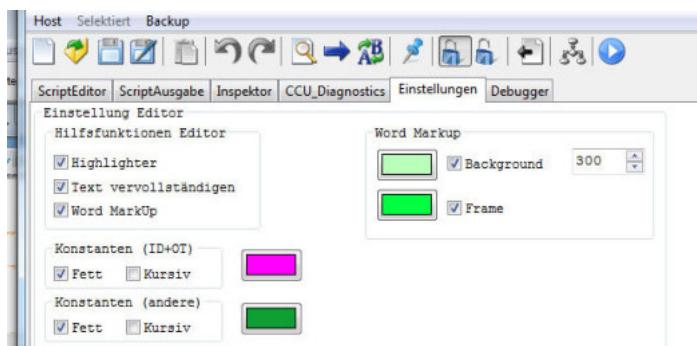
Der Editor beherrscht nun auch Wordmarkup. Bei einem Word, welches angeklickt wurde, oder geschrieben wurde (auf dem sich der Focus befindet ^^), werden sämtliche Vorkommen im Text farblich hervorgehoben



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

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

Die Funktion ist konfigurierbar unter Einstellungen



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

3 Scripteditor

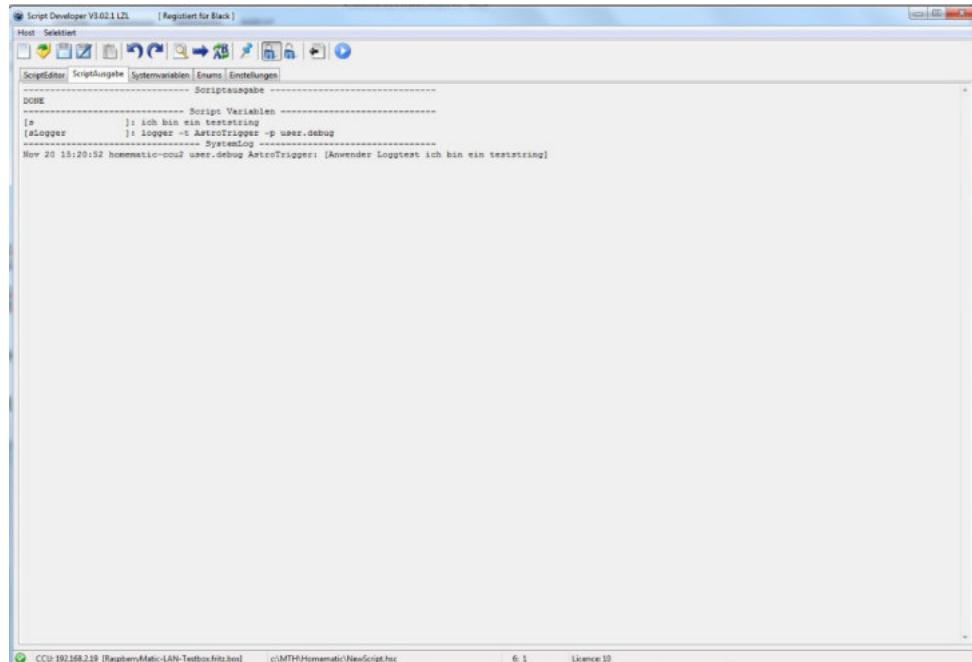
Im Scripteditor werden die Scripte geschrieben oder geladen, die mittels Run Script oder  an die CCU zum Ausführen gesendet werden. Das Scriptergebnis wird dann im Reiter Ausgabe angezeigt. Dieses kleine TestScript zum Beispiel:

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

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

Hier testweile aus State

Erzeugt folgende Ausgabe:



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

```
----- Script Variablen -----
[sLogger ]: ich bin ein teststring
[sLogger ]: logger -t AstroTrigger -p user.debug
----- SystemLog -----
Now 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]
```

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

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

Test Script erfolgreich

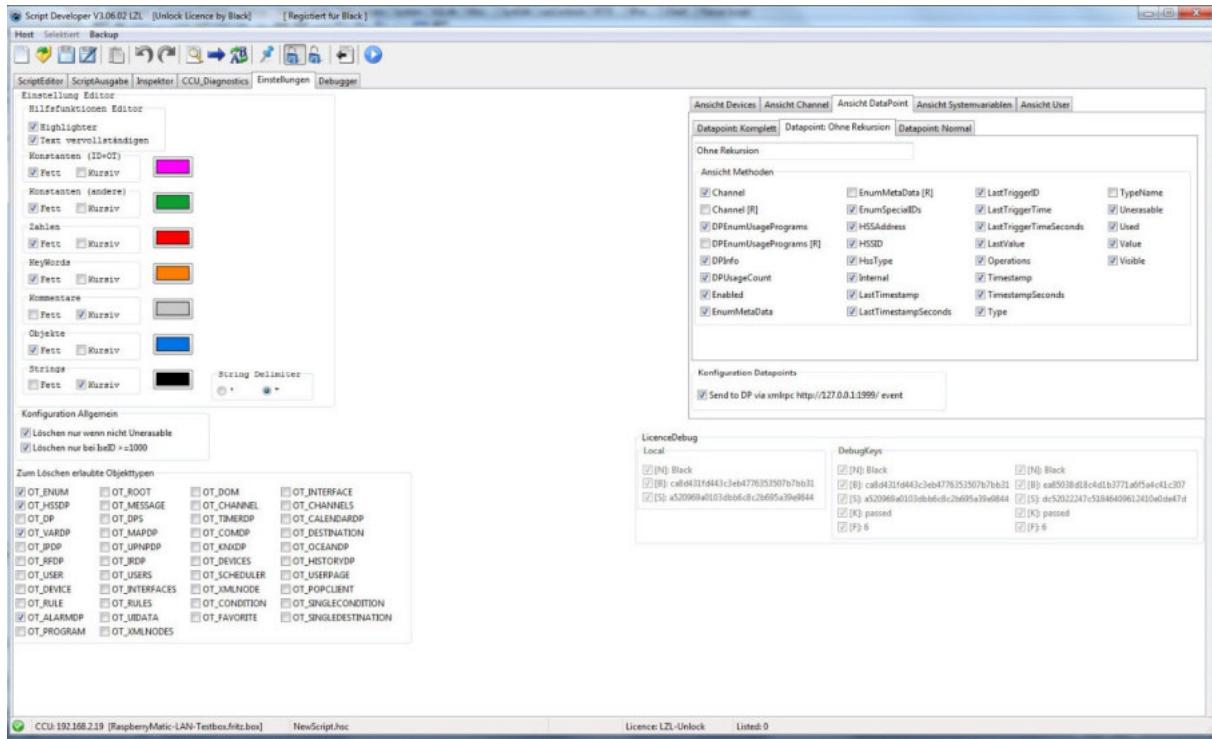
Trat ein Fehler auf, so erfolgt Sprung auf den Reiter Scriptausgabe mit Anzeige der Fehlermeldung

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

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

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

3.1 Voreinstellungen Editor



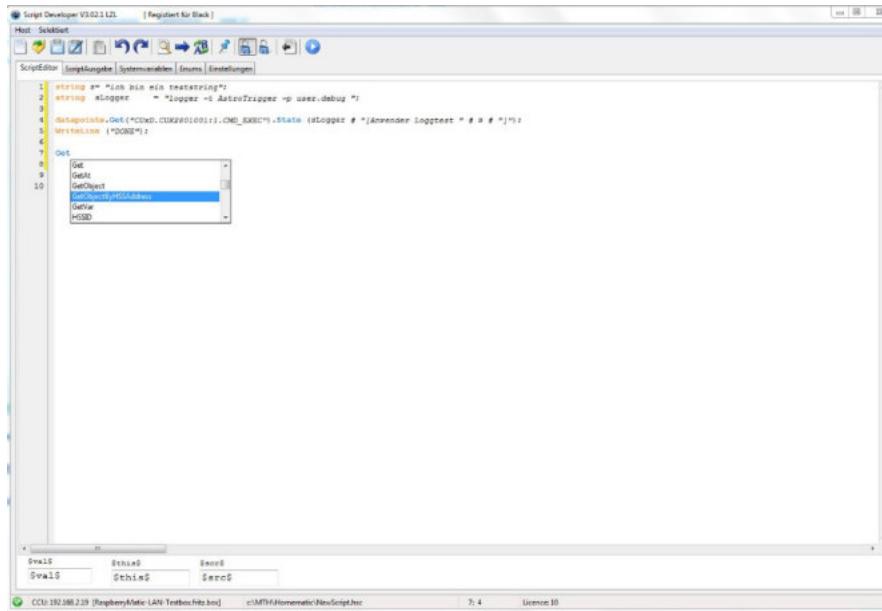
Hier kann nach Vorliebe der Highlighter konfiguriert oder auch ausgeschaltet werden

Ebenso lässt sich die Vervollständigen Funktion an oder abwählen.

Vorbedingung natürlich: Lizenzlevel muss vorhanden sein.

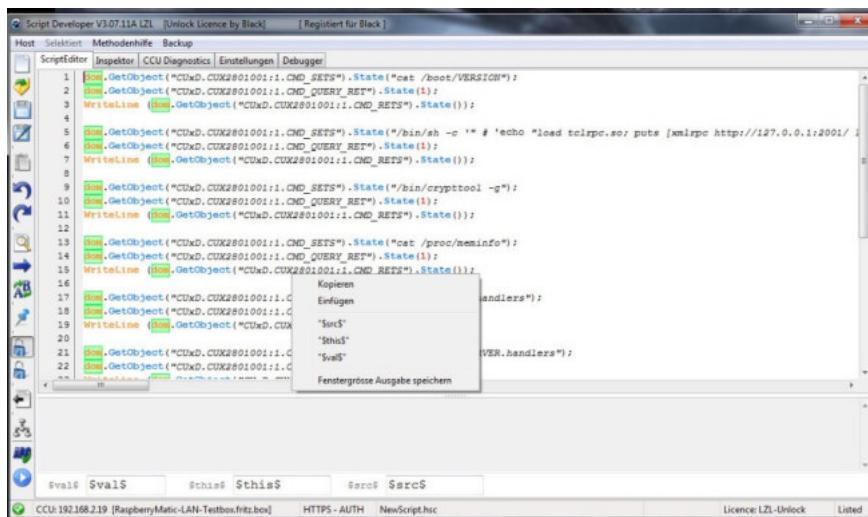
3.2 Vervollständigen Funktion

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



Nach Druck von Enter erscheint das Wort im Editor.

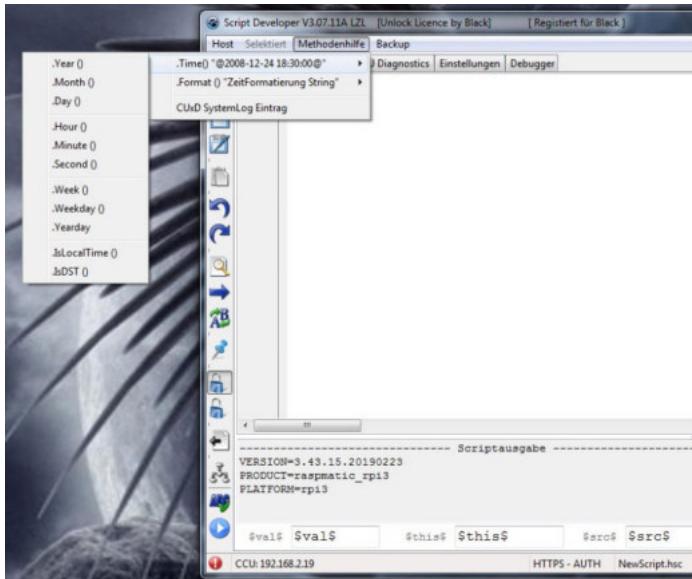
Die Grösse des Ausgabefensters kann abgespeichert werden.
Rechte Maustaste in den Editor und Fenstergrösse Ausgabe speichern



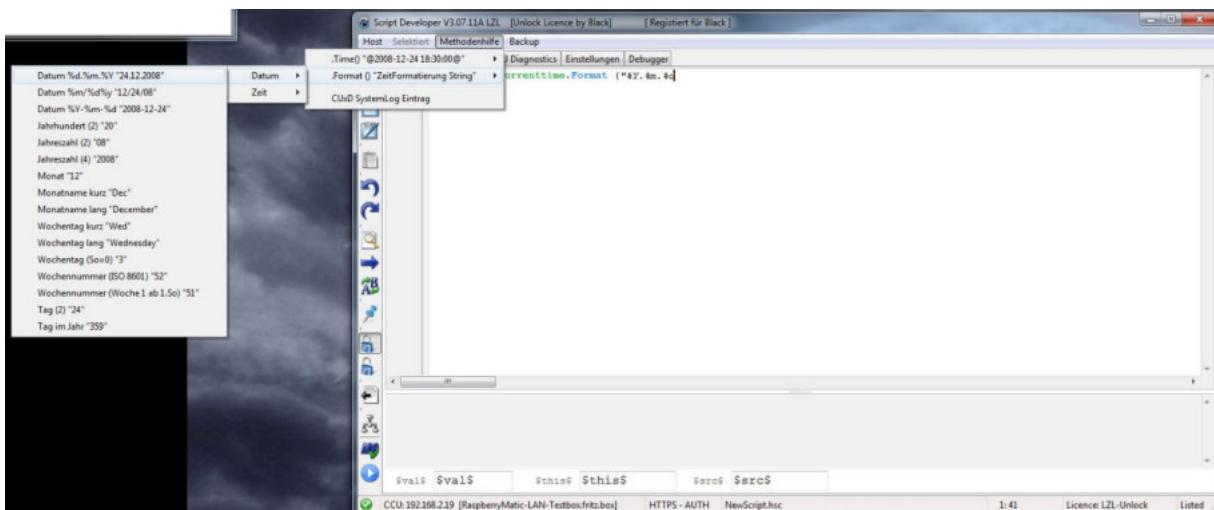
3.3 Methodenhilfe

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

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



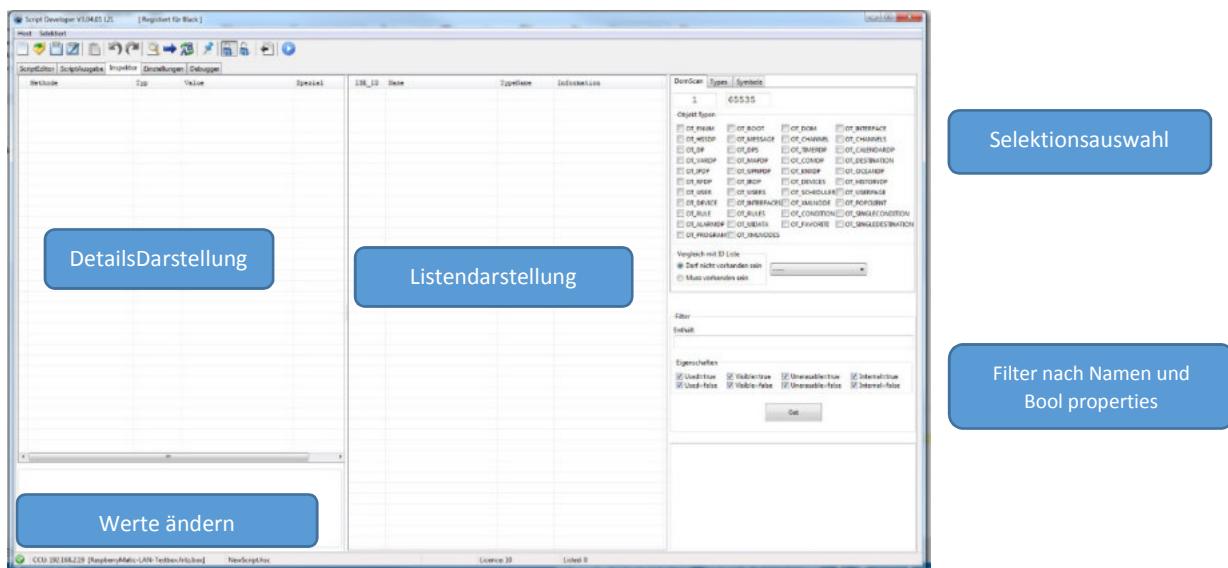
Durch Druck wird die Passende Funktion ausgewählt. Dito bei Zeitformatierung



4 Inspektor

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

Es existieren verschiedene Selektionskriterien.



Filteroptionen:

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

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

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

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

4.1 Selektionswahl: DomScan

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

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

Achtung

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

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

Mehrfachangaben sind möglich

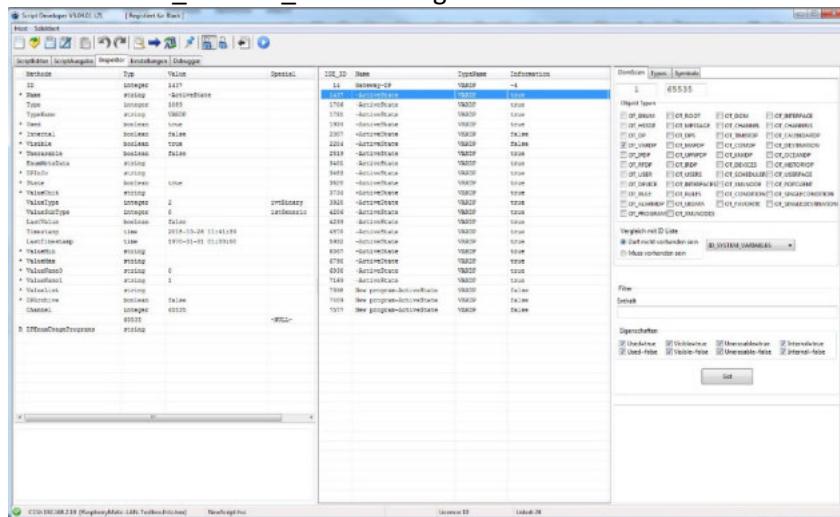
Beispiel für Suchen aller Objekte mit der Eigenschaft OT_DEVICE im Bereich der ISE_Nummern 1-65535

The screenshot shows the Script Developer V3.07.1 interface. On the left, there is a tree view of project files: 'Script-Developer V3.07.1.lzh' (selected), 'Raspberry-Main-L009-Testbox.lhs', 'CCU 182300.219 [Raspberry-Main-L009-Testbox.lhs]', and 'C:\MTH\NumericalData\lsh'. In the center, there is a table titled 'DomScan' with columns: 'Name', 'Value', 'Type', and 'Systemvariable'. The table lists several objects, including '1012 IP-PORT-10-Bridge-AF', '1239 CCU-ETH', '2144 CCU-ETH-USER', '2758 CCU-ETH', '2429 Zeitsymbol', '3568 Rm-12-Sv-Ph-1Q01110111', '4921 Zeitsymbol 0012121212121212', '4935 Zeitsymbol', '4974 Zeitsymbol', and '7418 Heiz-Reg. 031A8A86281010'. To the right of the table is the 'DomScan' dialog box. It has tabs for 'DomScan', 'Types', and 'Symbole'. The 'Types' tab is selected, showing checkboxes for various object types: OT_ENUM, OT_ROOT, OT_MESSAGE, OT_DOM, OT_INTERFACE, OT_HSSDP, OT_DP, OT_VARDP, OT_IPDP, OT_RFDP, OT_USER, OT_DEVICE, OT_RULE, OT_ALARMDP, OT_PROGRAM, OT_MESSAGE, OT_CHANNEL, OT_TIMERDP, OT_COMDP, OT_DESTINATION, OT_OCEANDP, OT_DEVICES, OT_SCHEDULER, OT_USERPAGE, OT_INTERFACES, OT_XMLNODE, OT_POPCLIENT, OT_CONDITION, OT_SINGLECONDITION, OT_FAVORITE, and OT_SINGLEDESTINATION. Below the checkboxes are dropdown menus for 'Vergleich mit ID Liste' (with options 'Darf nicht vorhanden sein' and 'Muss vorhanden sein') and a 'Filter' field. At the bottom of the dialog are 'Get' and 'Cancel' buttons.

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.

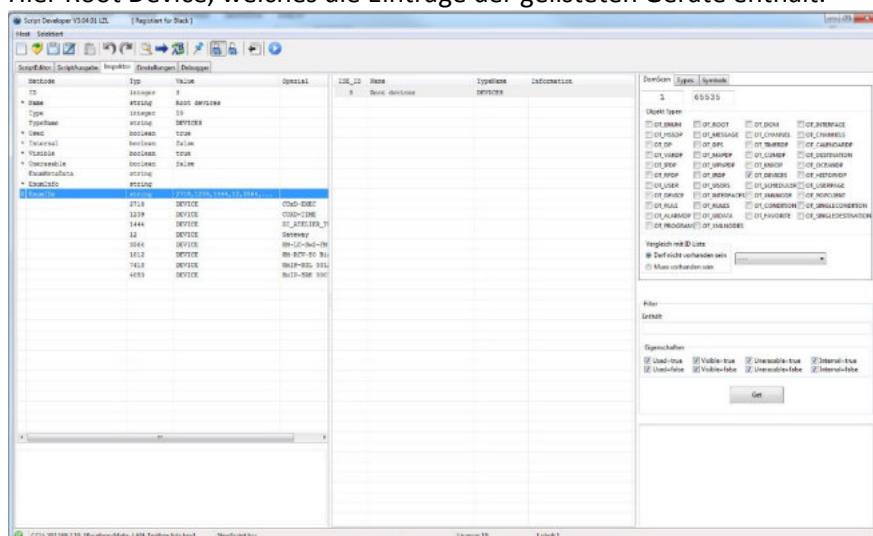
Hier Beispieldeweise: Scanlauf über alle Objekte aus DOM mit der Eigenschaft VARDP, die aber nicht unter ID_SYSTEM_VARIABLES gelistet sind:



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

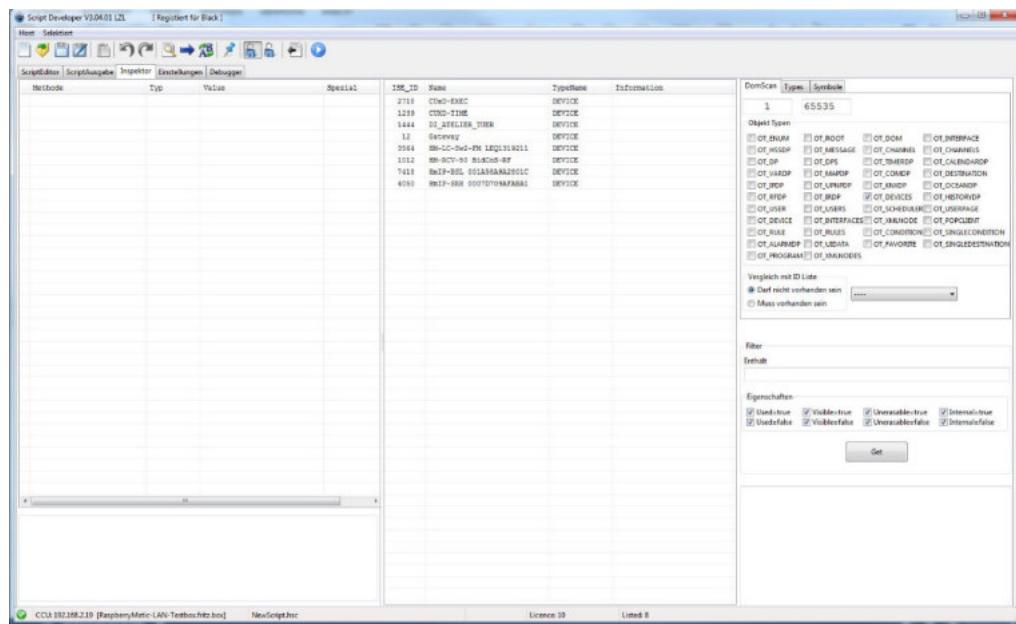
Weiterhin ist auch rekursives Arbeiten nun möglich

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

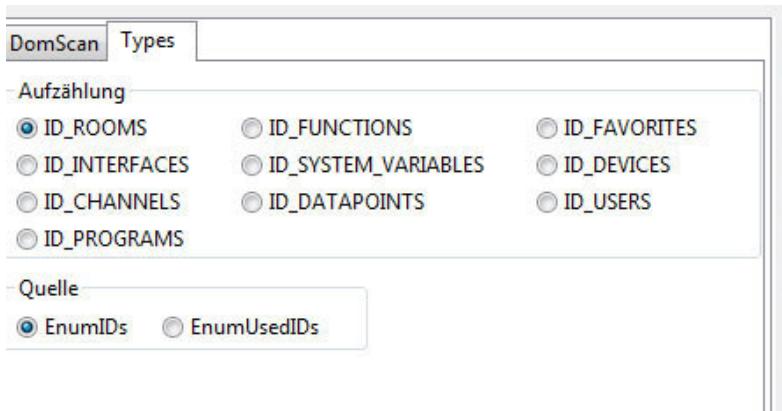


Das „R“ in der ersten Spalte zeigt an, dass ein Rekursiver Aufruf möglich ist. Doppelklicken löst und die Liste auf und lädt diese in die Listenauswahl. Ist es nur ein Wert, so wird dieser Wert auch direkt in der Detailansicht geöffnet.

Die neue Auswahllist steht nun zur weiteren Bearbeitung bereit.



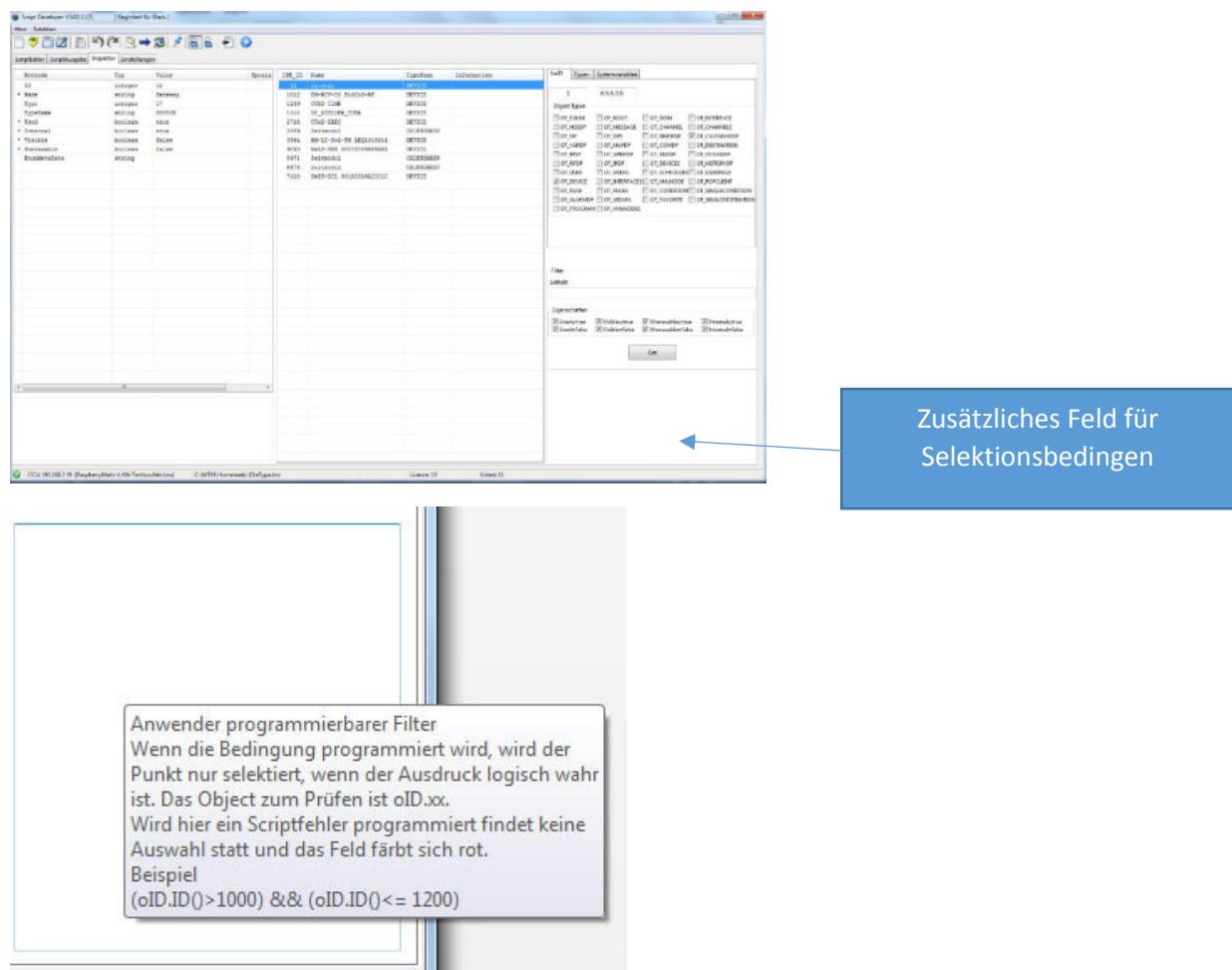
4.2 Selektionskriterium Types



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

4.3 Zusätzliche Selektionsbedingungen

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



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

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



Filter löschen entfernt sämtliche Filterbedingungen

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



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

Ein Filter besteht aus den 3 Abschnitten:

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

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

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

Vereinfachter Ablauf: so sieht vereinfacht das Listenselektionsprogramm aus:

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

Ein Anwenderdefinierter Filter wird dann in diese Grundschleife so eingebaut:

```
object oID;
string s;
ONCETEXT;

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

An diesem Kleinen Filter mal verdeutlicht:

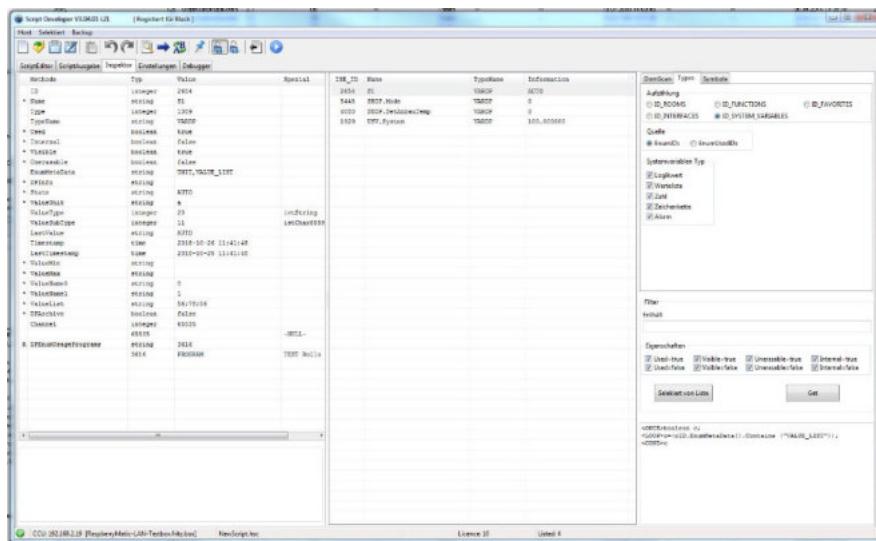
```
<ONCE>boolean c;  
<LOOP>c=(oID.EnumMetaData().Contains ("VALUE_LIST"));  
<COND>c
```

Daraus generiert der Scriptdeveloper folgende interne Filterabfrage:

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

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

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



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

object **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: Ein

boolean h: interner Filder, auch Finger weg

string done; auch interne Benutzung, auch

String done. auch interne Benutzung, auch Finger weg

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

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

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

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

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

PROGRAM_GeisterProg_CopyID - Filter um Geisterprogramme mit gesetzter CopyID zu finden

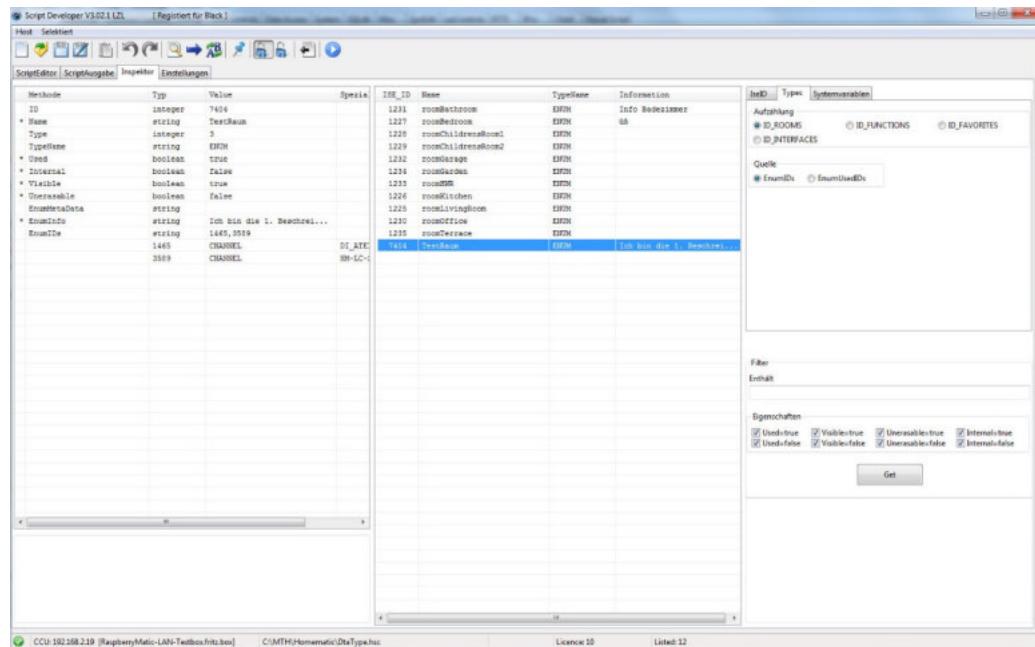
SYSVAR_VerwaisterChannel – Filter um Systemvariablen zu finden, deren Channel verweis in Nirvana zeigt

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

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

Changelog V3.03xx

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



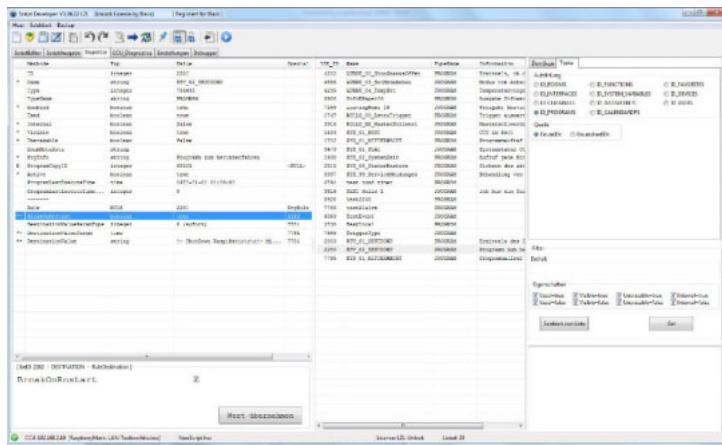
Die Spaltenbreite kann sowohl in Normalansicht als auch in Maximize separat eingestellt werden
(Das Programm sollte sich die Breiten merken und je nach Darstellungsart automatisch wieder einstellen, sollte...)

Dargestellt werden die Methode, der Vartype und die Property.

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

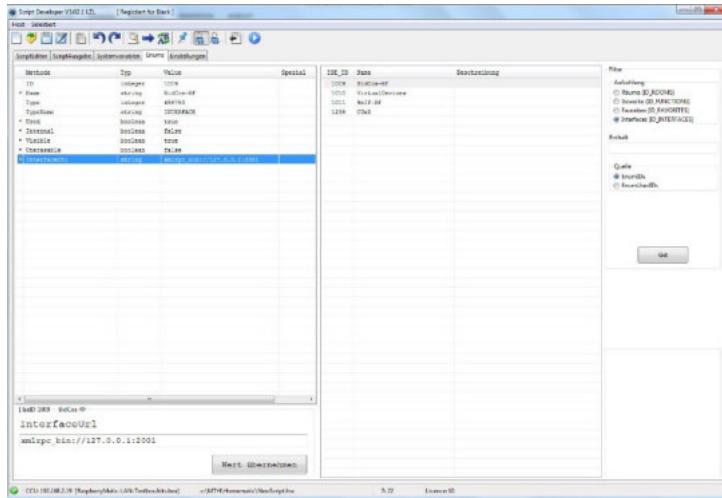
Properties, die in der ersten Zeile mit einem Stern (*) gekennzeichnet sind, können in ihrem Wert geändert werden.

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

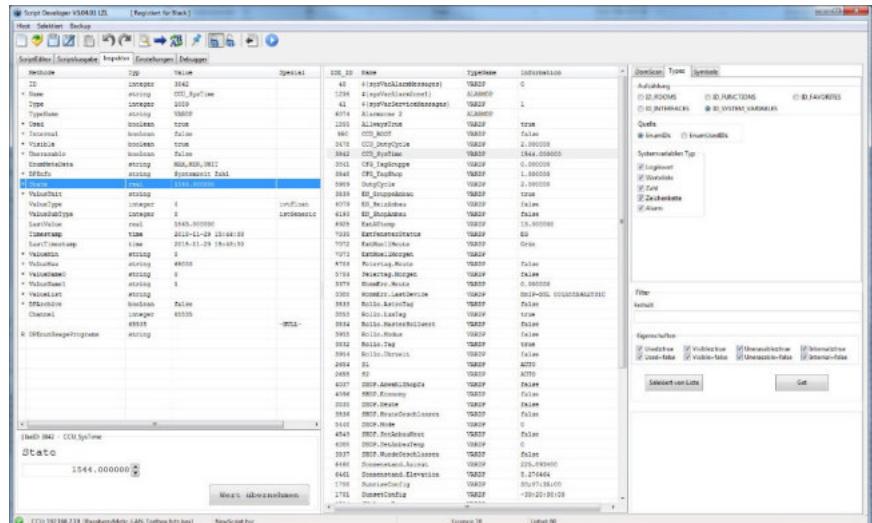


BreakOnRestart ist hier nicht Member von OT_PROGRAM sondern rekursiv von der entsprechenden Rule bzw SubRule.

Dazu auf die Zeile klicken



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

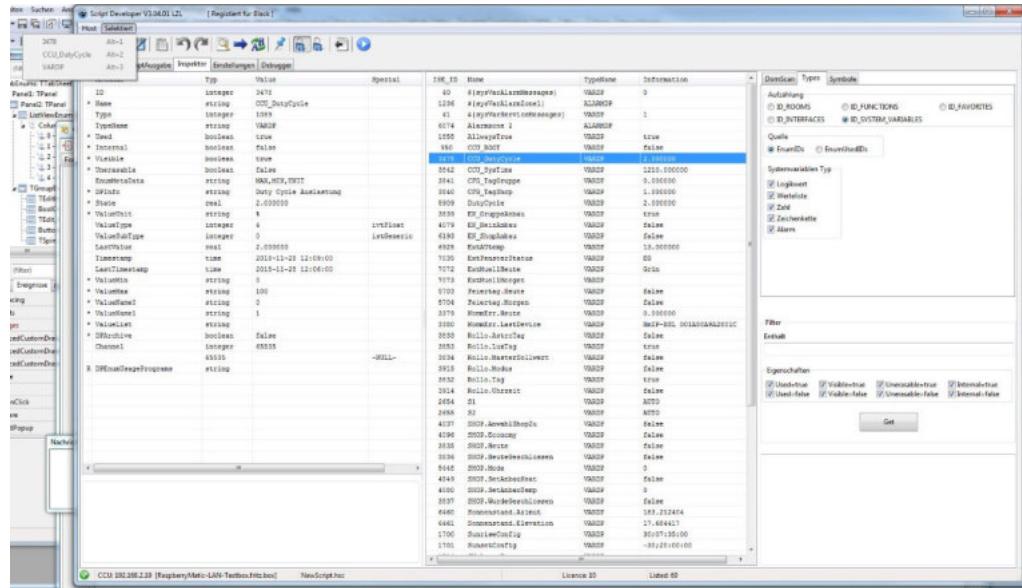


4.4 Daten aus Inspektor in Editor übernehmen

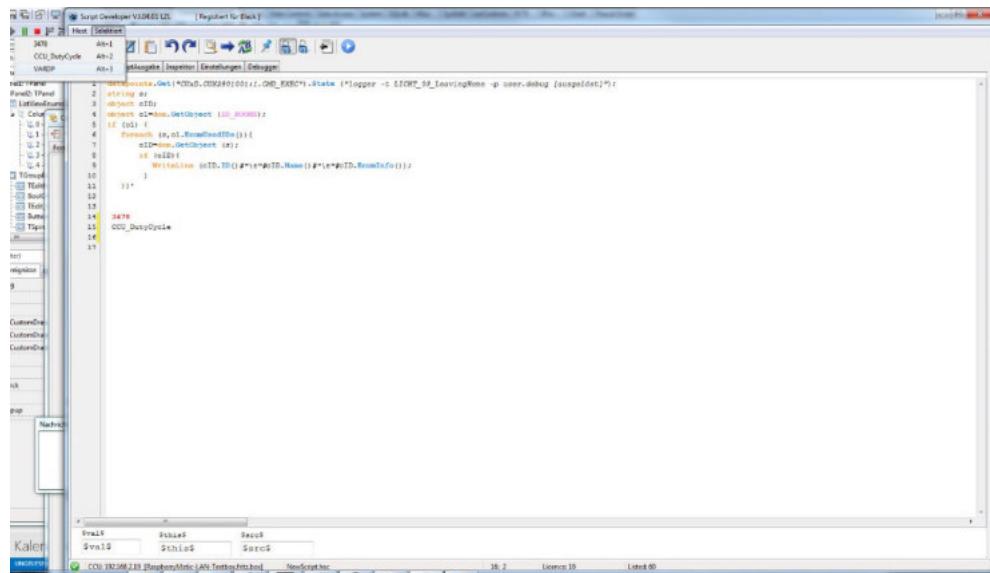
Zur schnelleren und auch möglichst fehlerfreien Bearbeitung besteht die Möglichkeit, Daten aus dem Inspektor direkt in den Editor zu übernehmen.

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

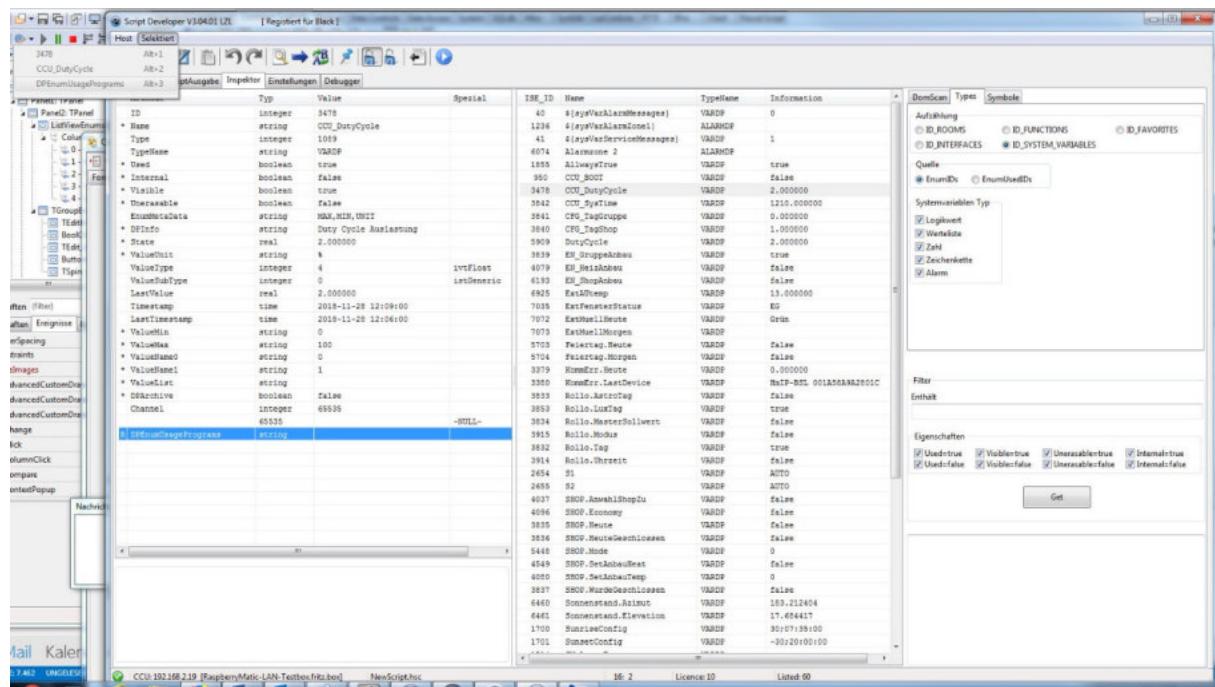
Mit den Tasten lassen sich folgende Spalteninhalte, welche vorher im Detailfeld selektiert worden sind, in den Editor übernehmen

Alt+m: Inhalt von Methode

Alt+t: Inhalt von typ

Alt+v: Inhalt von Value

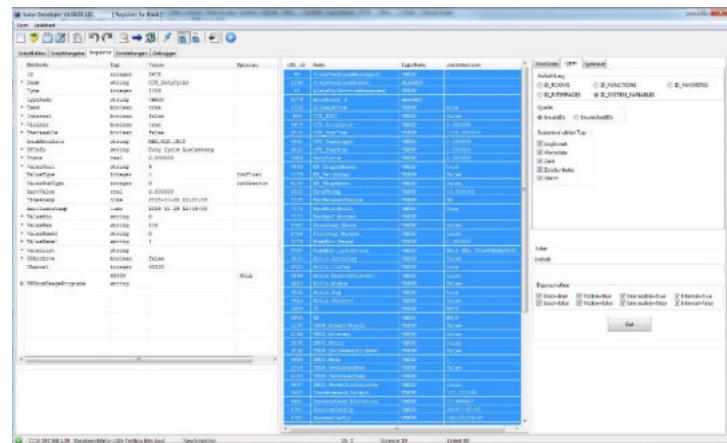
Alt+p: Inhalt von Spezial



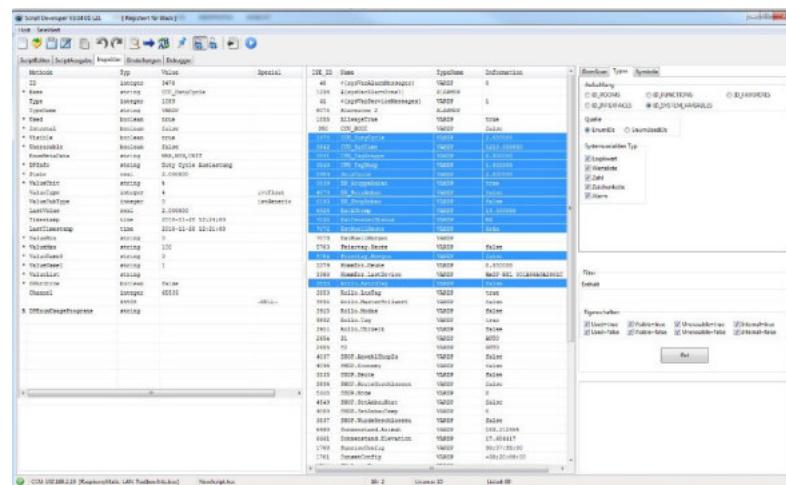
4.4.1 Mehrfachauswahl als Enum String

Es lassen sich im Hauptauswahlfeld Mehrfachselektionen vornehmen.

Ctrl-A : alle auswählen.



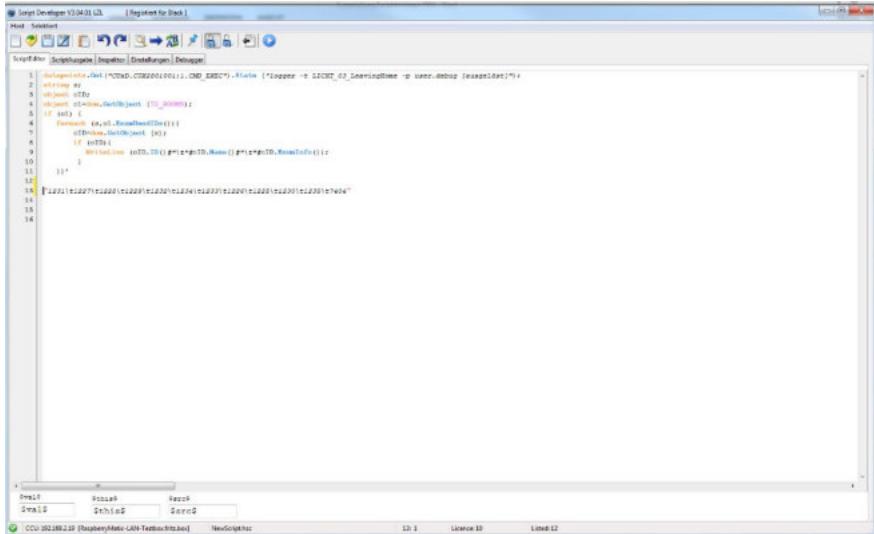
Oder die Übliche Mausbedienung:



Ab Version 3.07.07.

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

Mit der  Taste 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:

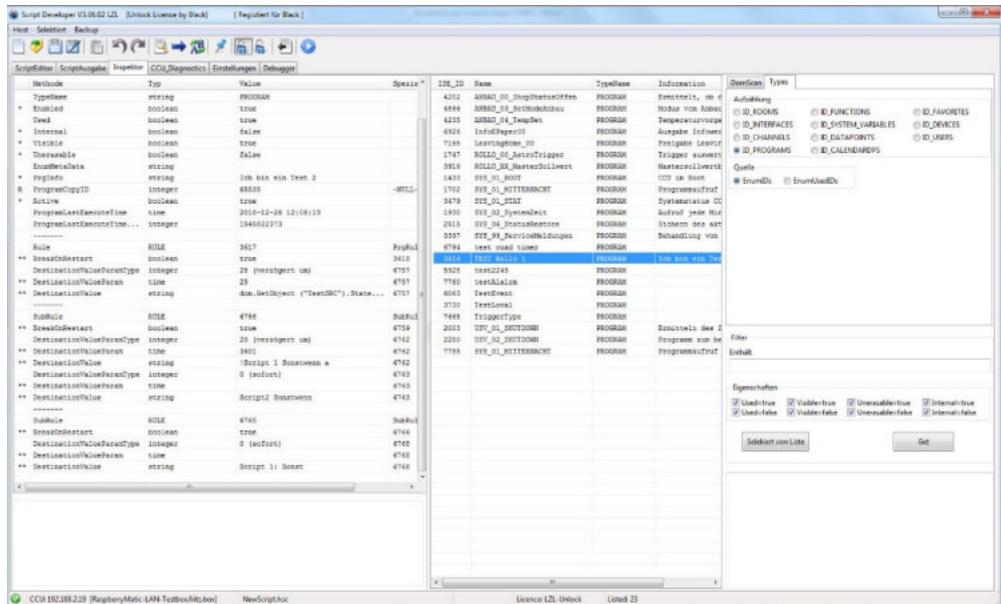
Mit rechter Maustaste im Mittleren Feld die Funktion „Einfügen aus Pinliste“ anwählen und die Sicherheitsabfrage bestätigen,



dann befinden sich die Selektierten Elemente wieder im mittleren Feld.

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

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



Dies entspricht der tabellarischen Darstellung dieses Programmes

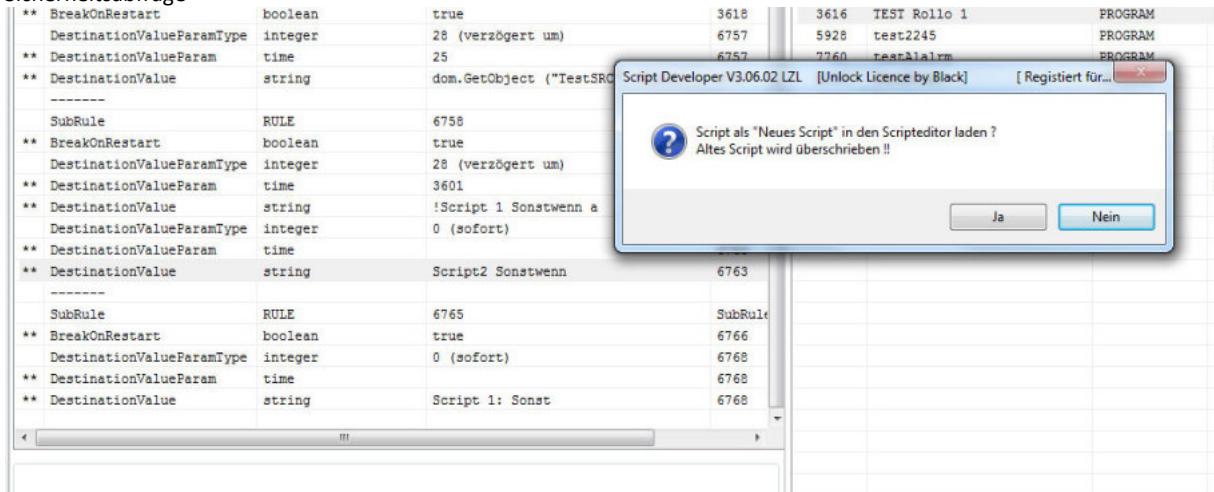
Name	Beschreibung	Bedingung (Wenn...)	Aktion (Dann... Sonst...)	Aktion
TEST Rule 1	Ich bin ein Test 2	Kanalzustand: DL_ATELIER_TUER:1 bei Fensterzustand: verriegelt bei Aktualisierung auslösen	Systemzustand: S1 sofort auf AUTO setzen	<input checked="" type="checkbox"/> systemieren

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

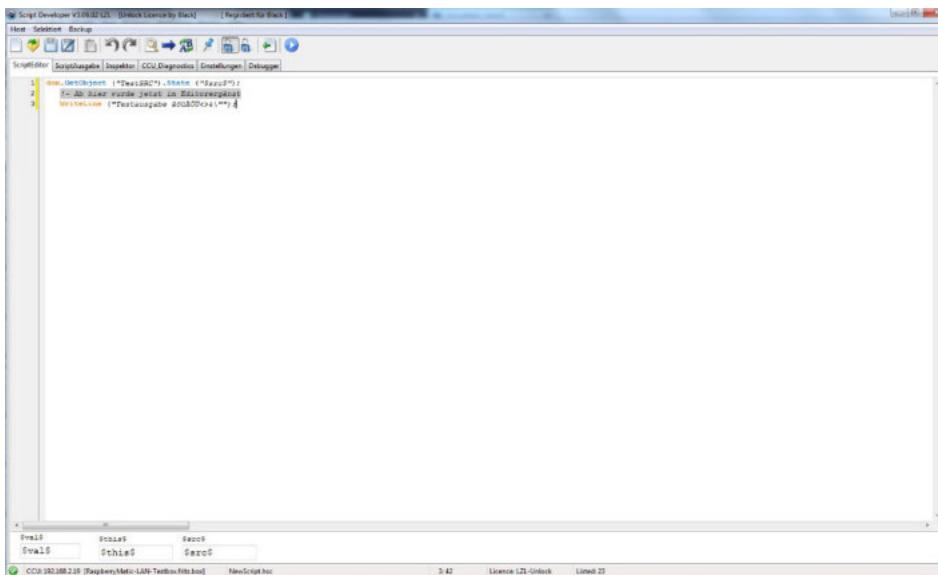
Die erste Subrule enthält dann die beiden Scripte aus der SonstWenn Bedingung.

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

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



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

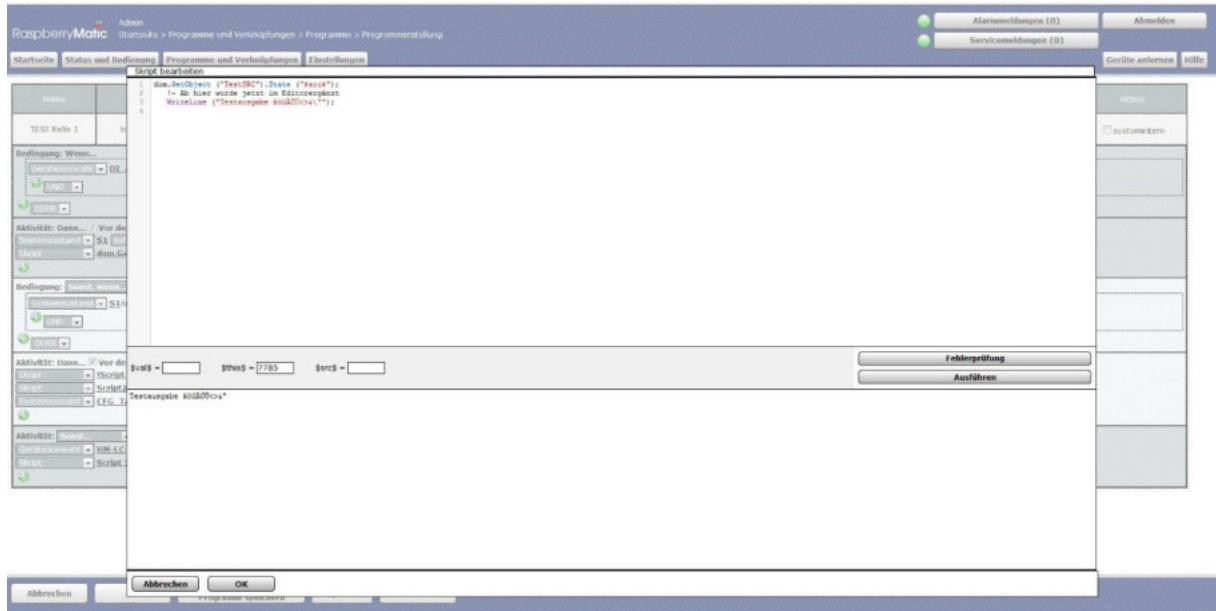


Danach kann durch Drücken der nun nicht mehr grauen Taste



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

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

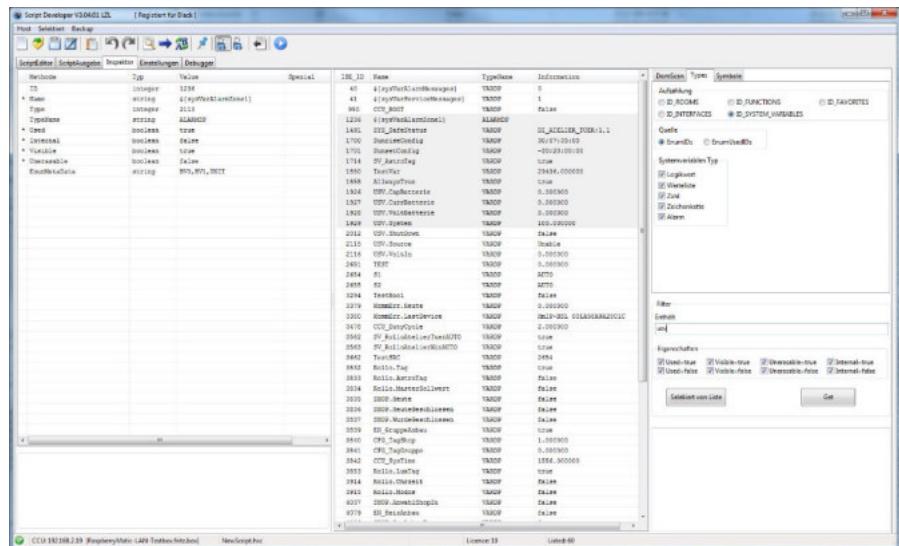


Das Programm würde nun das neue, geänderte Script bei Triggerung ausführen.

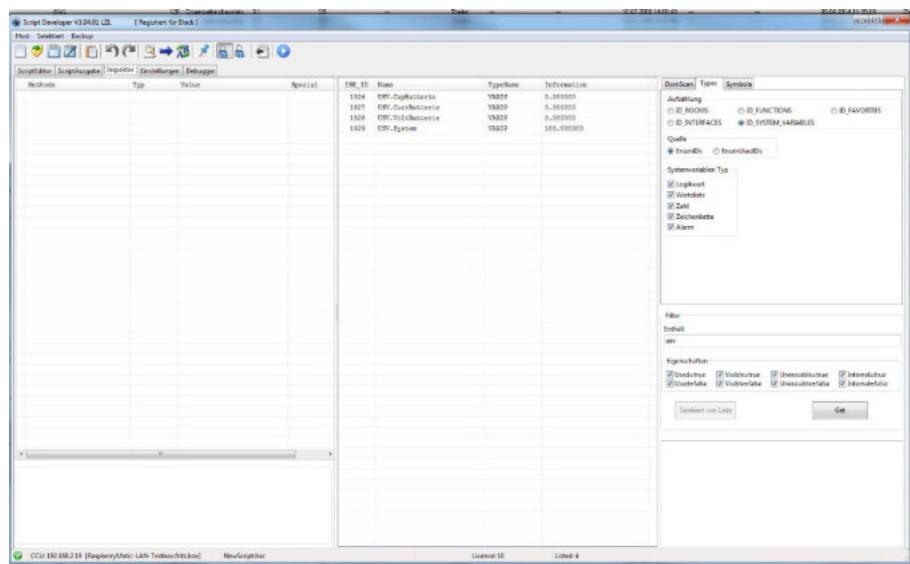
4.5 Selektion von Selektion

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

Hier Beispiel



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

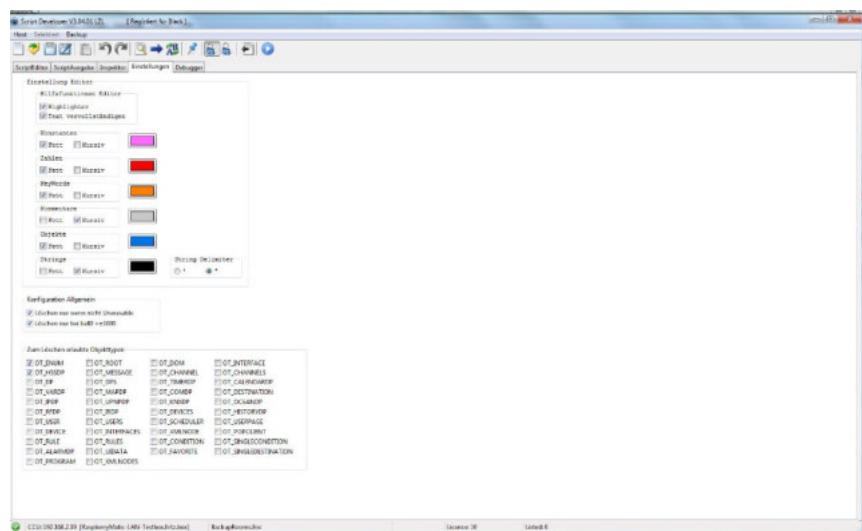


4.6 Objekte löschen

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

Redo geht nur über restore !

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



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

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

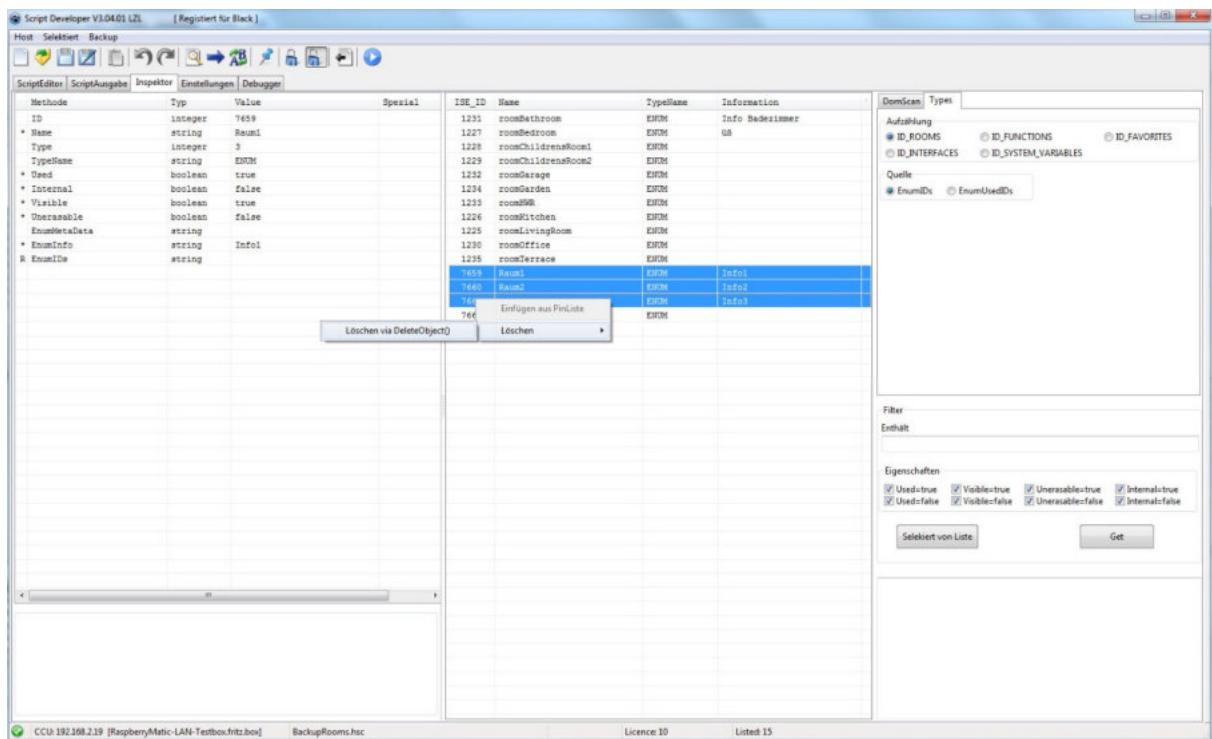
Im Inspektor unter Detailsview die Property entfernen (gilt nur für das Objekt), oder hier den Haken wegmachen (gilt für alle)

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

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

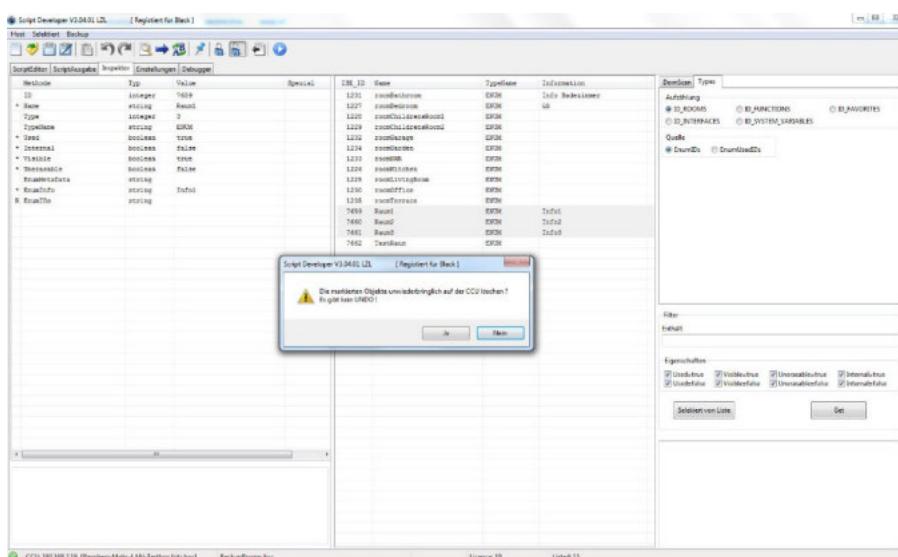
Löschbare Objekttypen. Die Letzte Sicherheit: ein zu lösches Objekt muss einen hier gecheckten Objekttyp haben, sonst wird es nicht gelöscht.

Löschen läuft so ab:



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

Mehrfachselektion ist möglich

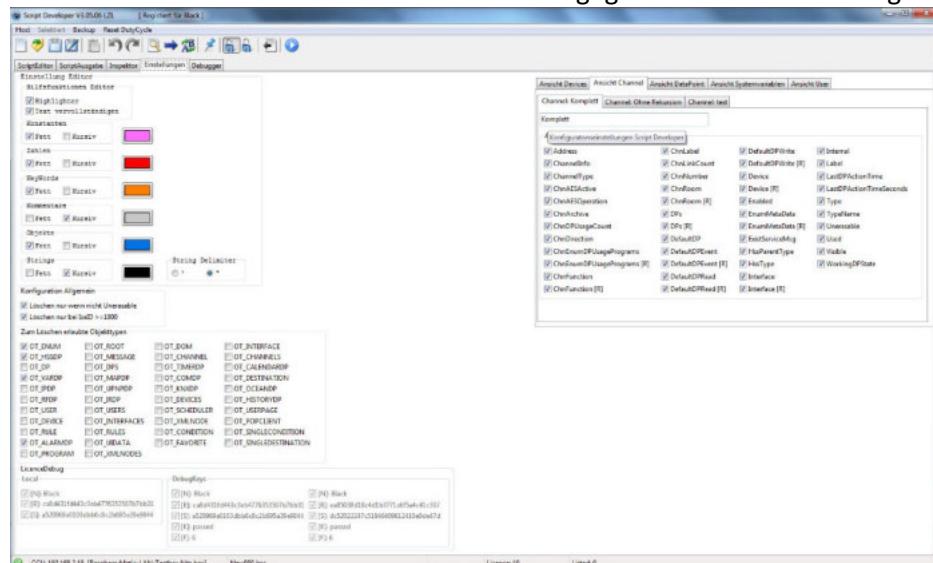


Nach dieser Sicherheitsabfrage sind die Objekte dann weg.. Zurück geht's dann nur mit Restore.

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

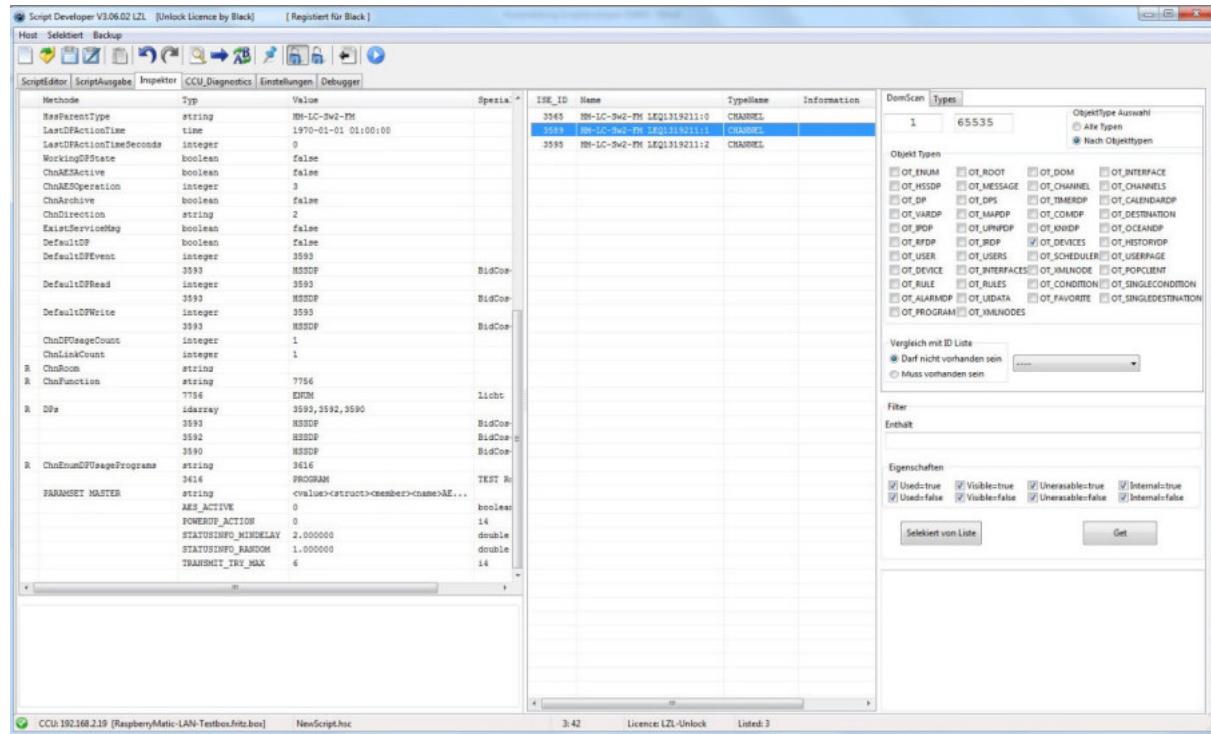
Hier Beispiel Einsprung über RootDevices

Methode	Typ	Value	Spezial	IS_E_ID	Name	TypeName	Information
ID	integer	3		2718	CUDB-EXEC	CUDB	
Name	string	Root devices		1239	CUDB-TIME	CUDB	
Type	integer	19		1444	DL_ATELIER	---	
TypeName	string	DEVICES		12	Gateway	---	
Enabled	boolean	true		3564	HM-LC-Sw1	---	
Used	boolean	true		1012	HM-RCV-S6	---	
Internal	boolean	false		7418	HMIP-BSL	---	
Visible	boolean	true		7624	HMIP-SLO	---	
Unreachable	boolean	false		7586	HMIP-SPI	---	
EnumMetaData	string			4050	HMIP-SRR	---	
EnumInfo	string						
R_EnumIds	string	2718,1239,1444,12,3564,1012,741...,					

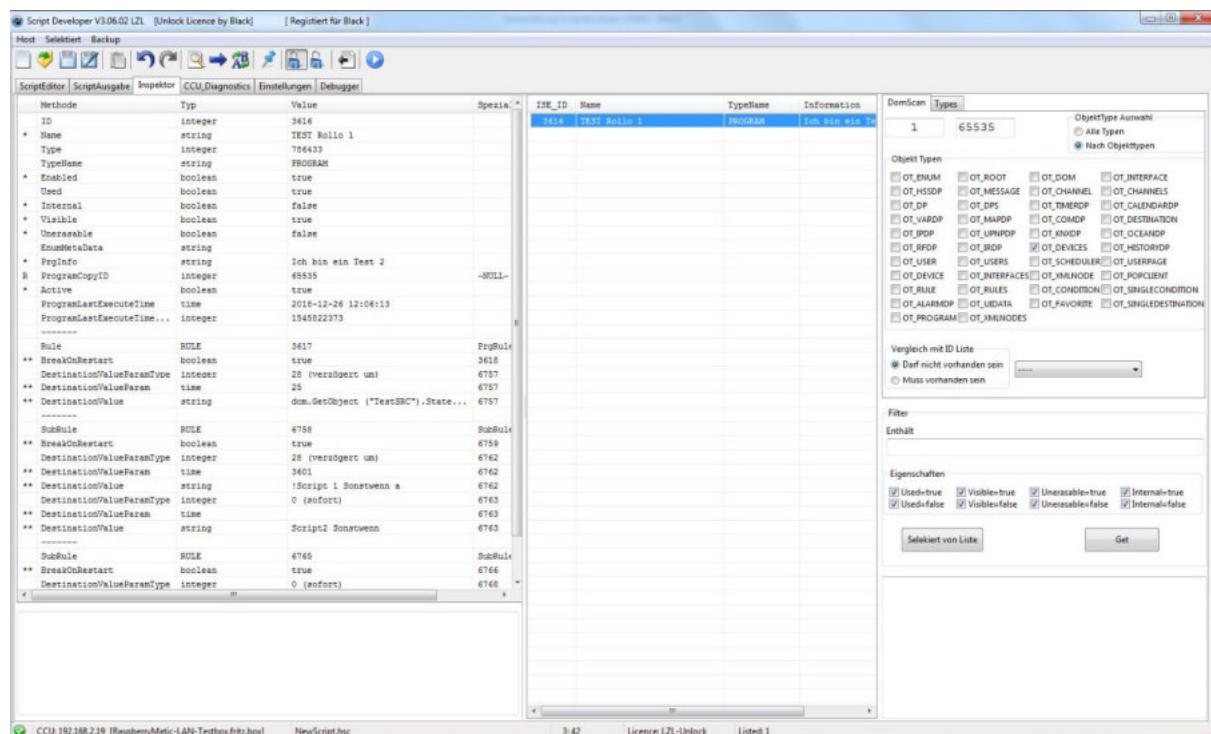
Rekursiv Aufgelöst die EnumIDs und Anwahl des 2 Fach Schaltaktors (Hier zu sehen, Undo ist schon anwählbar geworden)

Methode	Typ	Value	Spezial	IS_E_ID	Name	TypeName	Information
ID	integer	3564		2718	CUDB-EXEC	CUDB	
Name	string	HM-LC-Sw2-FM LEQ1319211		1239	CUDB-TIME	CUDB	
Type	integer	17		1444	DL_ATELIER_TOER	---	
TypeName	string	DEVICE		12	Gateway	---	
Enabled	boolean	true		3564	HM-LC-Sw2-FM LEQ1319211	DEVICE	BlidBox-RF
Used	boolean	true		1012	HM-RCV-S6 BlidBox-RF	DEVICE	BlidBox-RF
Internal	boolean	false		7418	HMIP-BSL 001A5EAA2B01C	DEVICE	BlidBox-RF
Visible	boolean	true		7624	HMIP-SLO 000D5A8A87FDA4	DEVICE	BlidBox-RF
Unreachable	boolean	false		7586	HMIP-SPI 000C1A8A8B9402	DEVICE	BlidBox-RF
EnumMetaData	string	AUTOCONF, DEVDESC, MASTERDESC, FAR...		4050	HMIP-SRR 0007D708AFA8A81	DEVICE	BlidBox-RF
DeviceInfo	string						
Interface	integer	1009	string				
Address	string	LEQ1319211					
Label	string	HM-LC-Sw2-FM					
NextType	string	HM-LC-Sw2-FM					
Channels().Count	integer	3					
R_Channels	idarray	3569,3595	CHANNEL				
		3595	CHANNEL				
CreatingCompleted	boolean	true					
LastTestCompletedTime	time	1970-01-01 01:00:00					
LastTestCompletedTimeS...	integer	0					

Und rekursiv weiter über die Channels des Devices

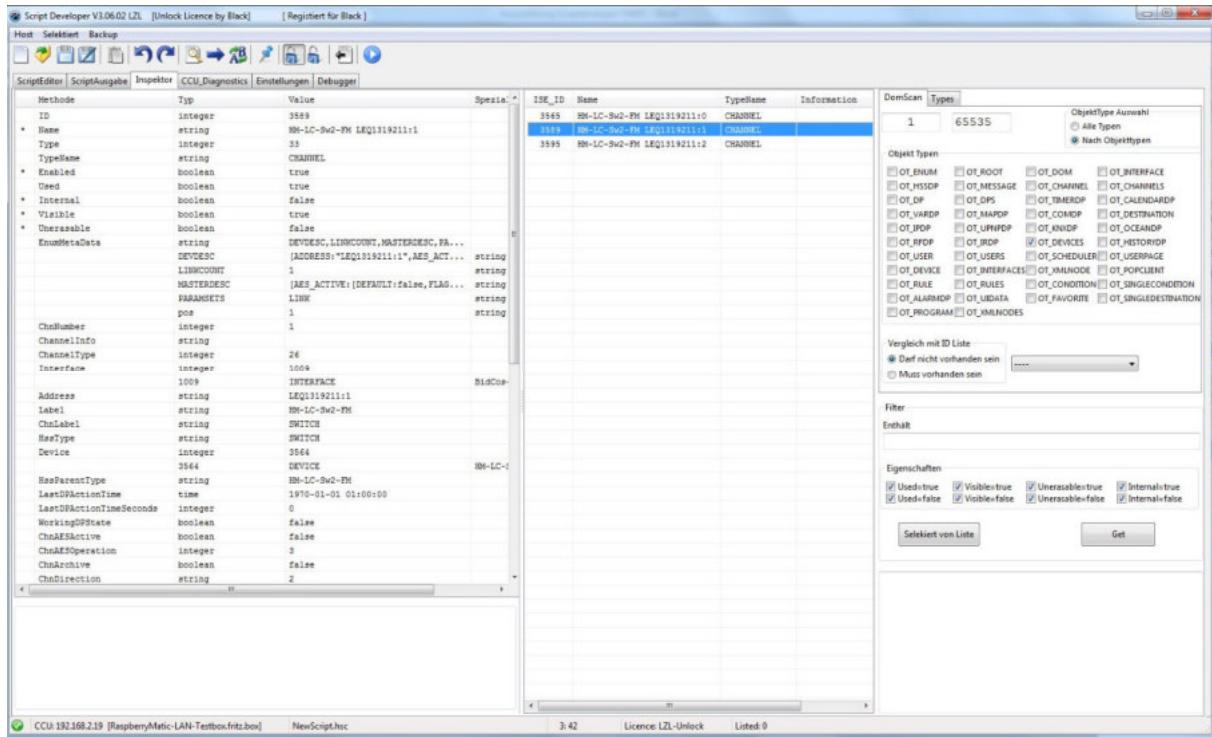


Mal in die Verwendung in dem Programm schauen über ChnEnumDPUsePrograms



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

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



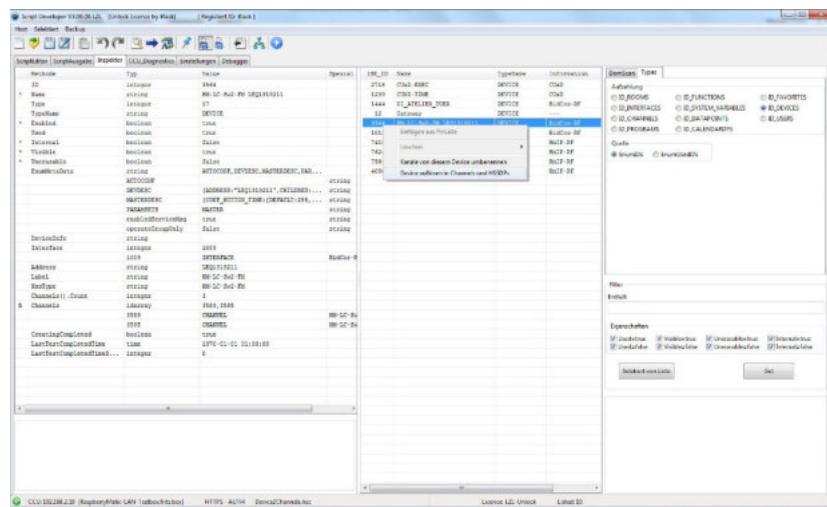
Ab Version 3.06.06 merkt sich der SDV zusätzlich zum Inhalt des Selektionsfeldes auch noch das zuletzt angezeigte Objekt in der Details view und stellt diese Ansicht auch wieder her (So das Objekt noch existent ist)

4.9 Auflösen komplexer Objekte

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

4.9.1 Auflösen von Devices

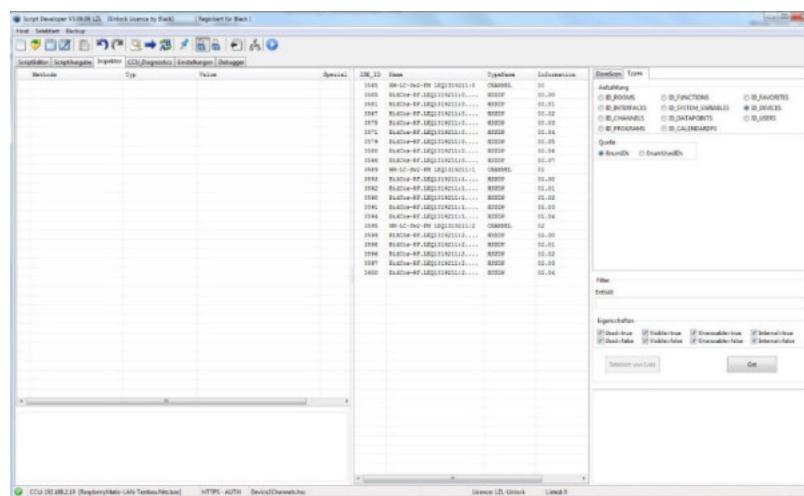
Ein Device besteht ja aus Channels und die Channels wiederum aus Datenpunkten



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

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

Als Ergebnis erhält man :

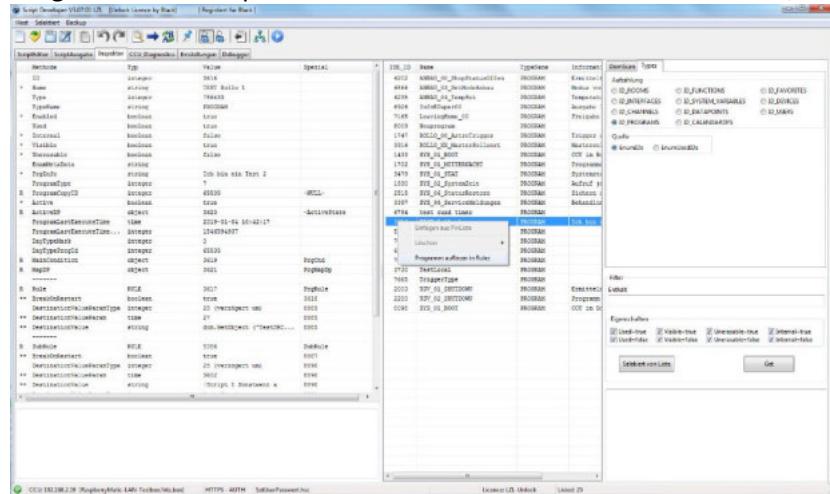


Diese Liste lässt sich dann weiter untersuchen mit den schon beschriebenen Arbeitsweisen (Auch Undo /Redo)

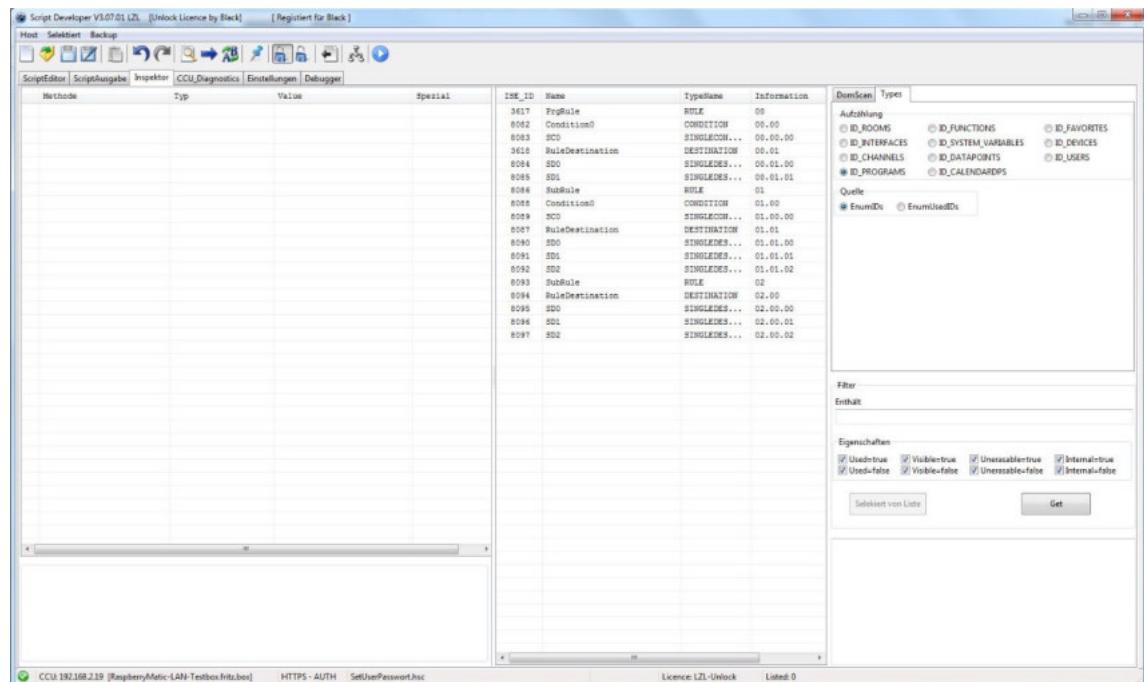
4.9.2 Auflösen von Programmen

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

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



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



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

Script Developer V3.07.01 L2L [Unlock Licence by Block] | Registriert für Block |

Host: Selektiert Backup

ScriptEditor ScriptAusgabe Inspector CCU_Diagnostics Einstellungen Debugger

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

- Left Pane (List View):**
 - Methoden: ID, Name, Type, Value, Spezial
 - ID: integer, Value: 8090
 - Name: string, Value: SD0
 - Type: integer, Value: 6291457
 - TypeName: SINGLEDESTINATION
 - Enabled: boolean, Value: true
 - Used: boolean, Value: true
 - Internal: boolean, Value: false
 - Visible: boolean, Value: true
 - Unusable: boolean, Value: false
 - EnumMetaData: string, Value: 'Script 1 Sonnena...
 - DestinationChannel: integer, Value: 65535, Special: -NULL-
 - DestinationDF: integer, Value: 65535, Special: -NULL-
 - DestinationParam: integer, Value: 20, Special: ivtString
 - DestinationParent: integer, Value: 8097, Special: DESTINATION
 - DestinationProgram: integer, Value: 3616, Special: PROGRAM
 - DestinationValue: integer, Value: 20, Special: ivtString
 - DestinationValueParamType: integer, Value: 20
 - DestinationValueParam: time, Value: 2019-01-04 01:00
 - DestinationValueParamType: integer, Value: 20
 - DestinationValueParam: time, Value: 2019-01-04 01:00
- Right Pane (Details View):**
 - ISID: integer, Value: 3417, Name: FrgRule, TypeName: RULE, Information: 00
 - ISID: integer, Value: 8082, Name: Condition0, TypeName: CONDITION, Information: 00.00
 - ISID: integer, Value: 8083, Name: SCO, TypeName: SINGLECOM..., Information: 00.00.00
 - ISID: integer, Value: 3618, Name: RuleDestination, TypeName: DESTINATION, Information: 00.01
 - ISID: integer, Value: 8084, Name: SD0, TypeName: SINGLEDES..., Information: 00.01.00
 - ISID: integer, Value: 8085, Name: SD1, TypeName: SINGLEDES..., Information: 00.01.01
 - ISID: integer, Value: 8086, Name: SubRule, TypeName: RULE, Information: 01
 - ISID: integer, Value: 8089, Name: Condition0, TypeName: CONDITION, Information: 01.00
 - ISID: integer, Value: 8090, Name: SCO, TypeName: SINGLECOM..., Information: 01.00.00
 - ISID: integer, Value: 8087, Name: RuleDestination, TypeName: DESTINATION, Information: 01.01
 - ISID: integer, Value: 8090, Name: SD0, TypeName: SINGLEDES..., Information: 01.01.00
 - ISID: integer, Value: 8091, Name: SD1, TypeName: SINGLEDES..., Information: 01.01.01
 - ISID: integer, Value: 8092, Name: SD2, TypeName: SINGLEDES..., Information: 01.01.02
 - ISID: integer, Value: 8093, Name: SubRule, TypeName: RULE, Information: 02
 - ISID: integer, Value: 8094, Name: RuleDestination, TypeName: DESTINATION, Information: 02.00
 - ISID: integer, Value: 8095, Name: SD0, TypeName: SINGLEDES..., Information: 02.00.00
 - ISID: integer, Value: 8096, Name: SD1, TypeName: SINGLEDES..., Information: 02.00.01
 - ISID: integer, Value: 8097, Name: SD2, TypeName: SINGLEDES..., Information: 02.00.02

4.10 Zeitmodule

Zeitmodule lassen sich nun auch in Detailansicht darstellen

Script Developer V3.07.01 L2L [Unlock Licence by Block] | Registriert für Block |

Host: Selektiert Backup

ScriptEditor ScriptAusgabe Inspector CCU_Diagnostics Einstellungen Debugger

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

- Left Pane (List View):**
 - Methoden: Internal, Visible, Unusable, EnumMetaData, State, Time, TimeSeconds, TimerType, Calculation, CalRepeatTime, CalRepeatTimeSeconds, Weekdays, Period, Begin, End, CalRepetitionCount, SunOffsetType, Next, Holidays, VacationCheck, VacationCheckType, BeginSeconds, EndSeconds, DPInfo, DPUusageCount, DPNumUsagePrograms, Timestamp, LastValue, LastTimestamp, LastTimestampSeconds, LastTriggererID, LastTriggerTime
 - Internal: boolean, Value: false
 - Visible: boolean, Value: true
 - Unusable: boolean, Value: false
 - EnumMetaData: string, Value: 'Script 1 Sonnena...
 - State: boolean, Value: true
 - Time: time, Value: 1970-01-01 01:00:00
 - TimeSeconds: integer, Value: 0
 - TimerType: integer, Value: 4, Special: ttPeriodic
 - Calculation: integer, Value: 0
 - CalRepeatTime: time, Value: 1970-01-01 01:00:00
 - CalRepeatTimeSeconds: integer, Value: 0
 - Weekdays: integer, Value: 0
 - Period: integer, Value: 60
 - Begin: time, Value: 2018-01-19 00:00:00
 - End: time, Value: 1970-01-01 01:00:00
 - CalRepetitionCount: integer, Value: 0
 - SunOffsetType: integer, Value: 0, Special: setNone
 - Next: time, Value: 2019-01-04 22:23:00
 - Holidays: boolean, Value: false
 - VacationCheck: boolean, Value: false
 - VacationCheckType: boolean, Value: false
 - BeginSeconds: integer, Value: 1516316400
 - EndSeconds: integer, Value: 0
 - DPInfo: string
 - DPUusageCount: integer, Value: 1
 - DPNumUsagePrograms: string, Value: 1930
 - Timestamp: time, Value: 2019-01-04 22:22:00
 - LastValue: boolean, Value: true
 - LastTimestamp: time, Value: 2019-01-04 22:21:00
 - LastTimestampSeconds: integer, Value: 1544688460
 - LastTriggererID: integer, Value: 65535
 - LastTriggerTime: time, Value: 1970-01-01 01:00:00
- Right Pane (Details View):**
 - ISID: integer, Value: 3489, Name: Zeitmodul, TypeName: CALENDAROP
 - ISID: integer, Value: 5971, Name: Zeitmodul, TypeName: CALENDAROP
 - ISID: integer, Value: 4678, Name: Zeitmodul, TypeName: CALENDAROP

4.11 Suchen in Scripten nach Variablen, Devices etc

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

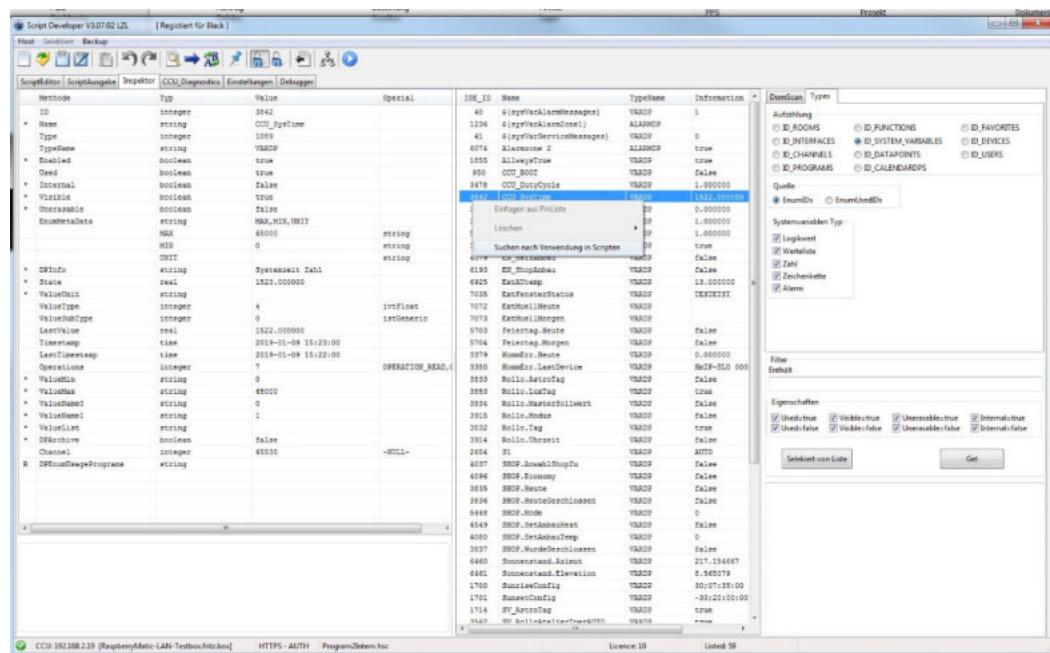
Dazu werden sämtliche SingleConditions, die als Property ein Script enthalten, gesucht und dann via Stringvergleich nach dem oder den Vorkommenden Namen abgesucht.

Vorgehensweise:

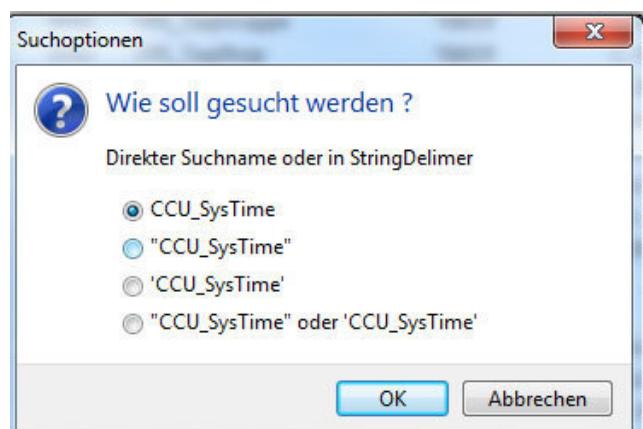
Markieren der oder auch mehrere Suchvariablen (Können Systemvariablen, Devices, Kanäle, Räume und Gewerke sein) Rechte Maustaste und suchen nach Verwendung in Scripten

Es kann hierbei gesucht werden:

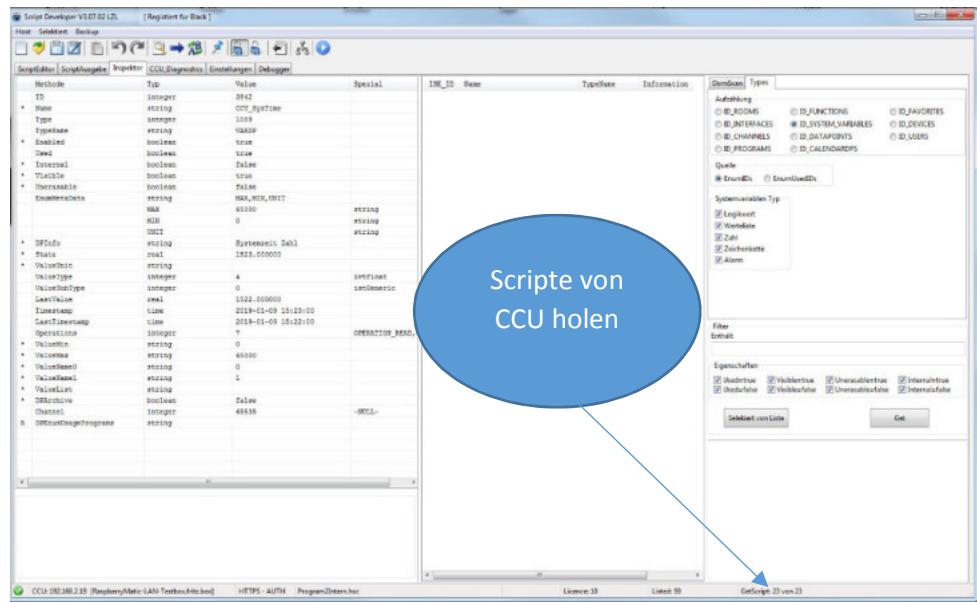
- Suchen "Name" nach Verwendung in Scripten : Suche nach dem Namen (SV, Alarme, Device, Chans)
- Suchen "Address" nach Verwendung in Scripten : Sucht nach der Seriennummer (Address) von Devices und Channels



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

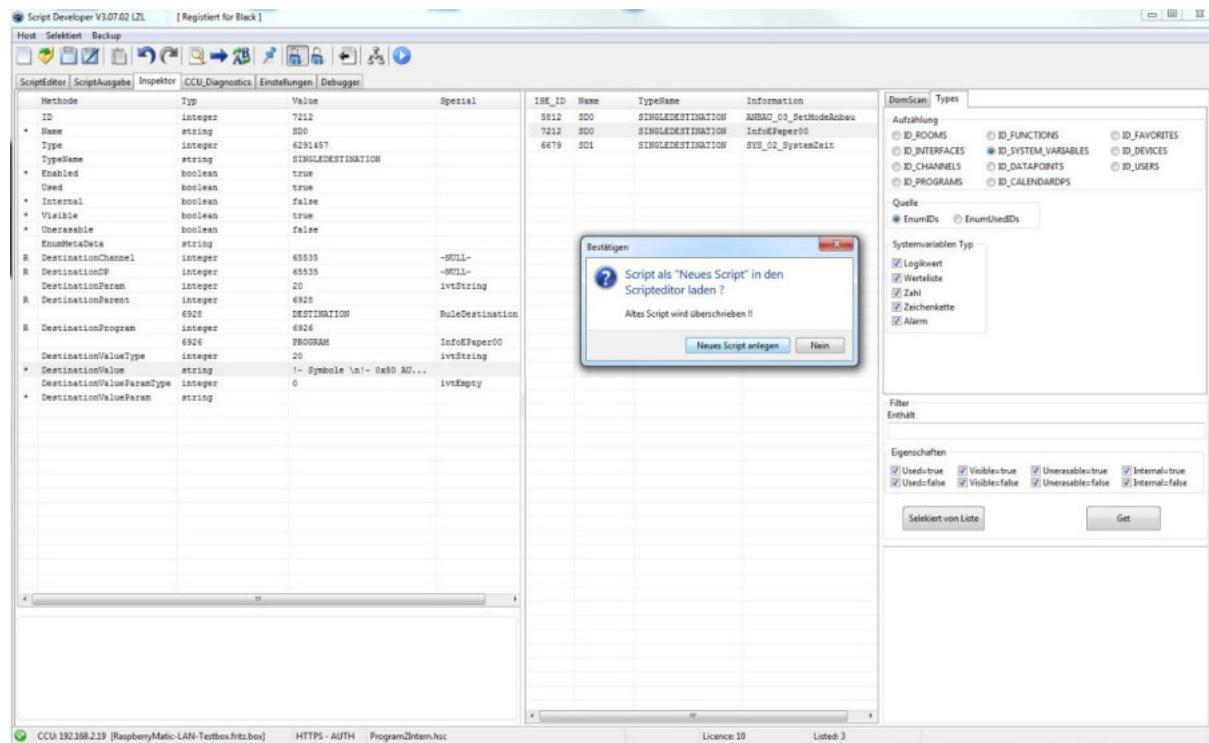


Daraufhin werden erstmal alle Rule in ihre Destinations aufgedröselt und eine Liste angelegt, wie viele Scripte es in diesen SingleDestinations dann gibt. Diese werden in den PC geladen. Das geht recht Fix, um der CCU zwischenzeitlich Zeit für Ihre Aufgaben zu lassen, wird direkt nach dem Empfang PC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



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

Das Script lässt sich dann nach Rückfrage öffnen:



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

The screenshot shows the Script Developer V3.07.02 L2L interface. The main window displays a script in the 'ScriptEditor' tab. The script contains various comments and variable assignments. At the bottom of the editor, there are four input fields: \$val1\$, \$this\$, \$src\$, and \$val2\$. Below these fields, the status bar shows the path 'CCU 192.168.2.19 [RaspberryPi-LAN-Testbox.fritzbox]', the connection 'HTTPS - AUTH', the file 'NewScript.hsc', 'Licence 10', and 'Listed: 3'.

```

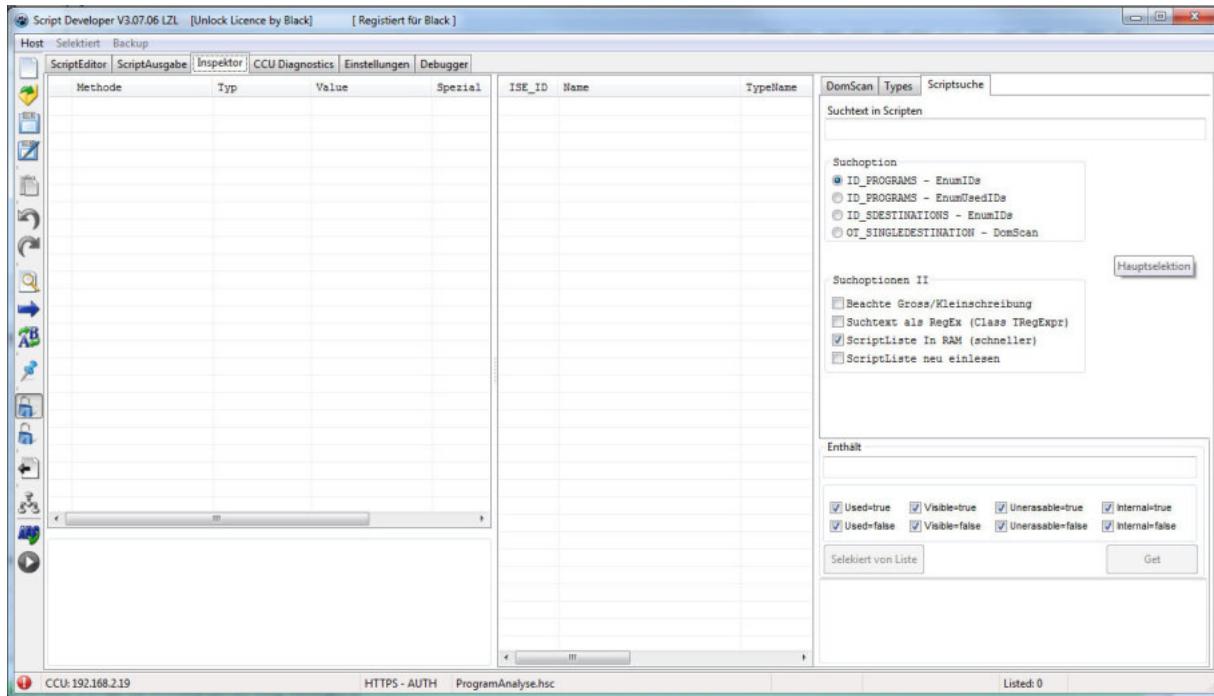
29 /*- 0xC1 LANG LANG
30 /*- 0xC2 LANG KURZ
31 /*- 0xC3 LANG_MORZ KURZ
32 /*- 0xC4 KURZ
33 /*- 0xC5 KURZ_KURZ
34 /*- 0xC6 LANG
35
36 /*- Signale
37 /*- 0xF0 AUS
38 /*- 0xF1 Rotes Blitzen
39 /*- 0xF2 Grünes Blitzen
40 /*- 0xF3 Orangenes Blitzen
41
42 /*- Seriennummer des HV-Disp-EP-WMS,
43 string seriennummer = "NEQ1443971";
44 object cSM= dom.GetObject("ID_SYSTEM_VARIABLES").Get("SVSubmit_Di_EP_Flur");
45
46 /*-variablen für die Texte und Icon
47 /*-Nicht verwendete Variablen nicht löschen, sondern auf "" setzen!
48 /*-definierte Zeichen können 0x00-0x99 für vordefinierte Texte oder bis zu 12 Zeichen Freitext sein. (längere Texte werden abgeschnitten)
49 object oObj= dom.GetObject("ID_SYSTEM_VARIABLES").Get("ExtFensterStatus");
50 if (oObj.State != "") {
51     zeile2= "";
52     icon2= "0x02";
53 } else {
54     zeile2= oObj.State ();
55     icon2= "0x02";
56 }
57
58 string zeile3 = dom.GetObject("ID_SYSTEM_VARIABLES").Get("ExtAUtemp").State ().ToString (1) # * C #;
59 string icon3 = "";
60
61 /*- Müllstatus nach der Uhrzeit darstellen
62 /*- Wenn heute Müll und Uhrzeit zwischen 0-8 Uhr Heute Müll darstellen
63 /*- Andernfalls
64 /*- Wenn Morgen Müll und Uhrzeit zwischen 18 und 24 Uhr.. Morgen Müll darstellen
65 integer lSysTime= dom.GetObject("ID_SYSTEM_VARIABLES").Get("CCU_System").State ().ToInt32 ();
66 object oSel= dom.GetObject("ID_SYSTEM_VARIABLES").Get("ExtSel");
67 string icon4 = "";
68 oObj= dom.GetObject("ID_SYSTEM_VARIABLES").Get("ExtNullibeute");
69 if ((oObj.State != "") && (lSysTime>90) && (lSysTime<330)) {
70     zeile4= oObj.State ();
71     icon4= "0xd4";
72 }

```

\$val1\$	\$this\$	\$src\$
\$val2\$	\$this\$	\$src\$

4.12 Volltextsuche in Scripten

Ab Version 03.07.07 existiert eine Volltextsuche (ab Level 6)



Suchtext ist selbsterklärend.

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

ID_Programs – EnumIDs

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

ID_Programs – EnumUsedIDs

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

ID_SDESTINATIONS – EnumIDs

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

ID_SINGLEDESTINATION – DomScan

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

Beachte Gross/Kleinschreibung

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

Suchtext als Regex

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

https://en.wikipedia.org/wiki/Regular_expression#POSIX%20Basic%20Regular%20Expressions

Scriptliste in Ram

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

Scriptliste neu einlesen

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

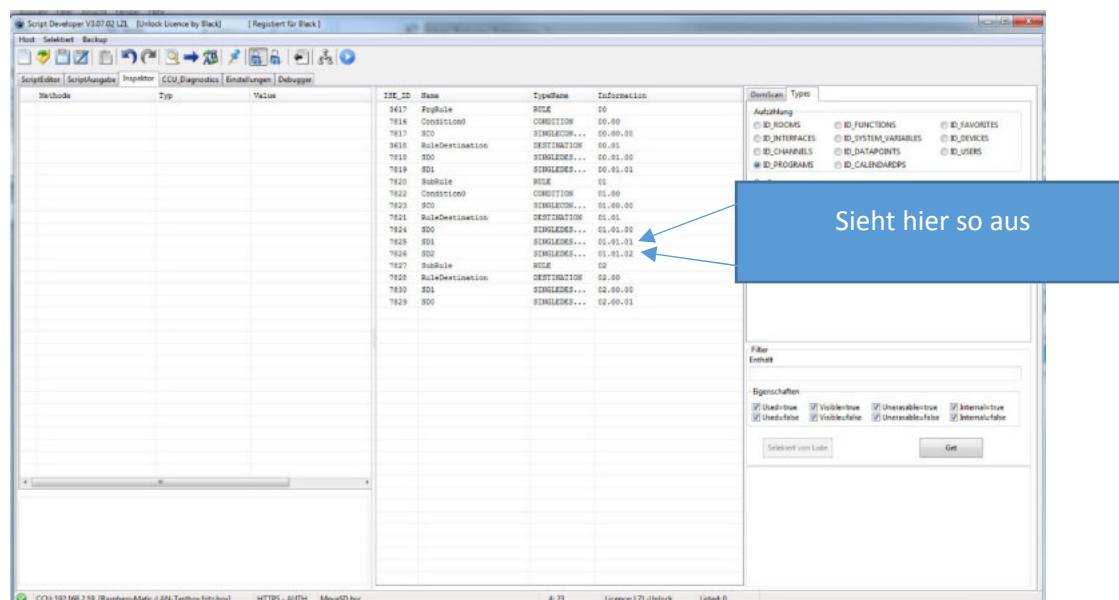
4.13 SingleDestinations in ihrer Reihenfolge ändern

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

Vorher:

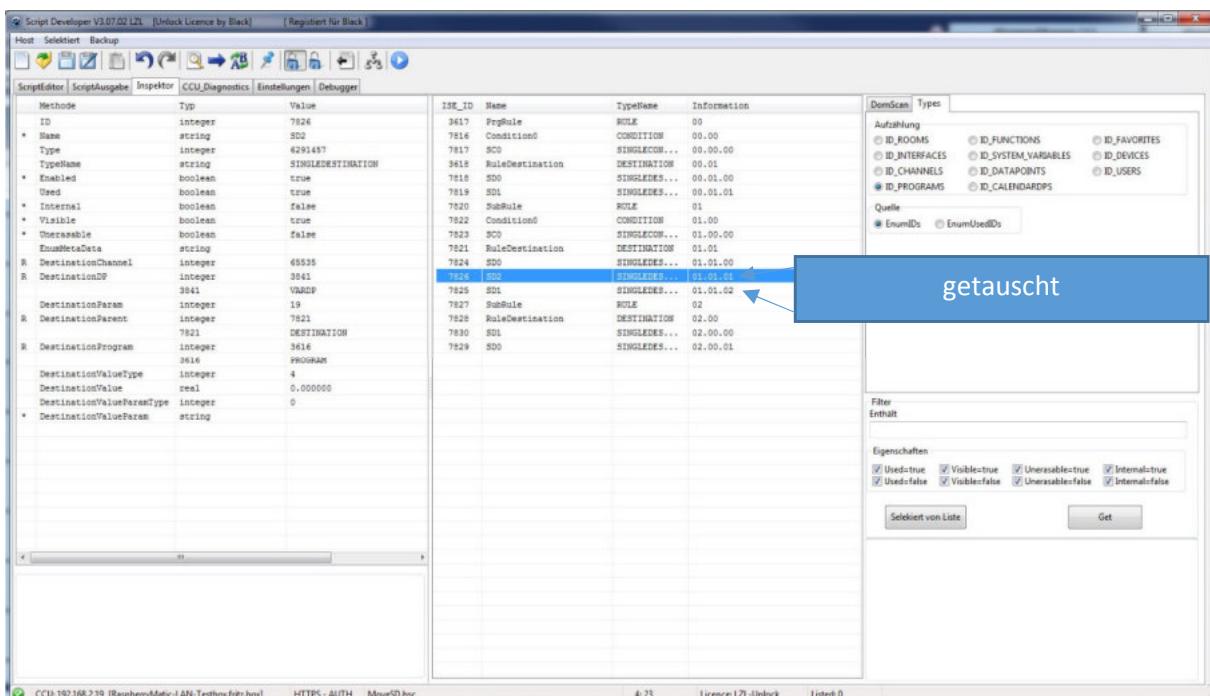
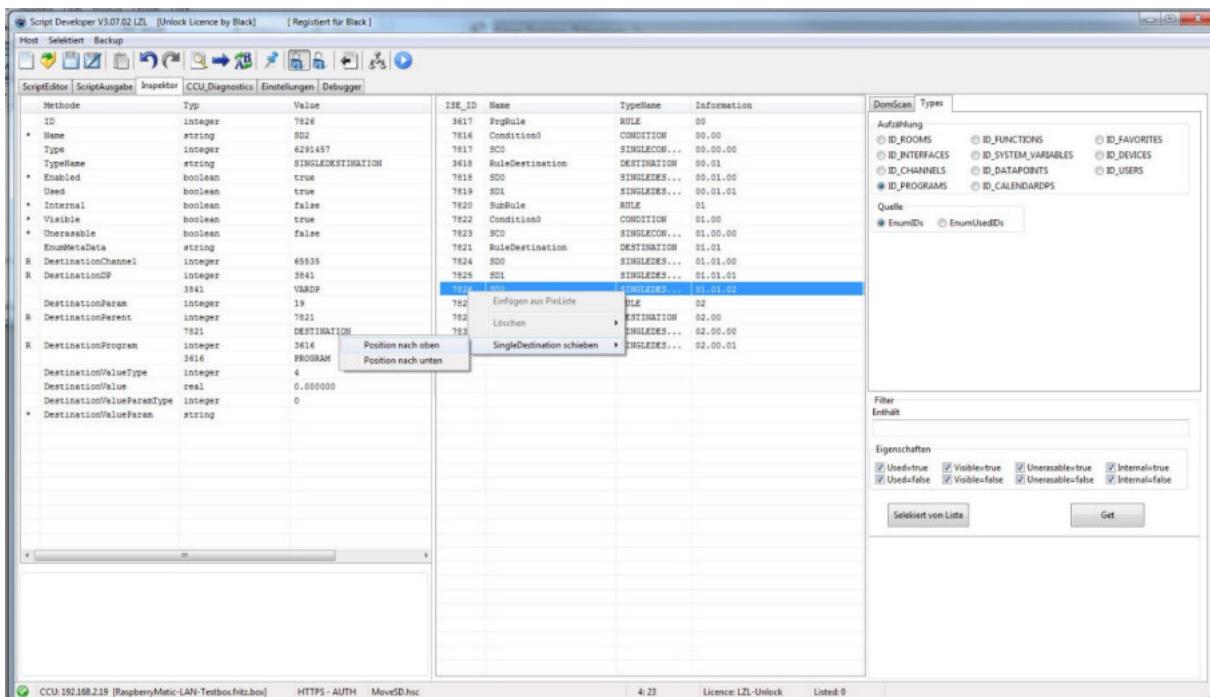
Name	Beschreibung	Bedingung (Wenn...)
TEST Rollo 1	Ich bin ein Test 2	Kanalzustand: DI_ATELIER_TUER:1 bei Fensterzustand: verriegelt bei Aktualisierung auslös
Bedingung: Wenn... <input checked="" type="checkbox"/> Gerätelauswahl <input type="checkbox"/> DI_ATELIER_TUER:1 bei Fensterzustand: verriegelt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

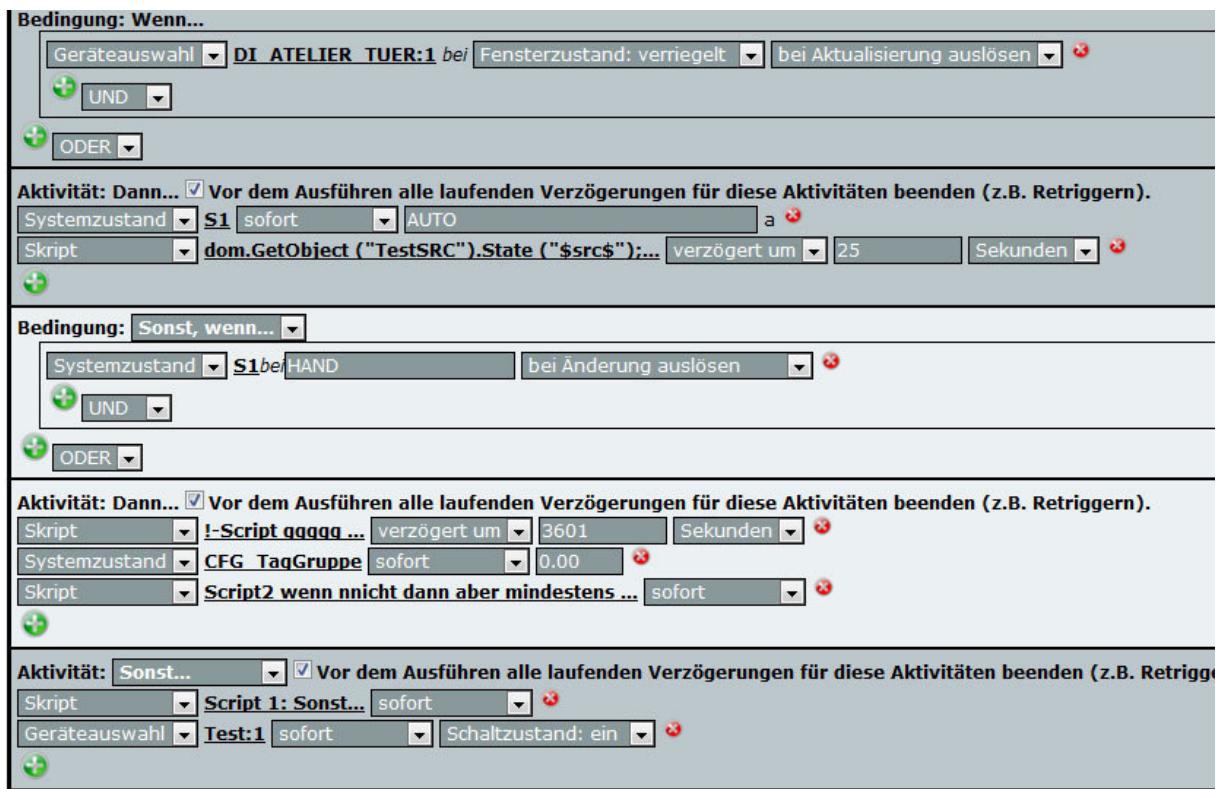
Das Aussehen im SDV



The screenshot shows the Script Developer V3.07.02 L2L interface. On the left, there's a tree view labeled "Selbst & Backup". The main area is a table with columns: ID, Name, Type/Name, and Information. The table contains several entries, including Rule, Condition, SCU, RuleDestination, and RuleSubRule. To the right of the table is a sidebar with sections for "Bereichen" (Regions), "Aufgaben" (Tasks), and "Filter" (Filter). A blue callout box points to the "Aufgaben" section with the text "Sieht hier so aus".

ID	Name	Type/Name	Information
7817	ProgRule	RULE	00
7818	Condition0	CONDITION	00-00
7817	SCU	SINGLEREG...	00-00-00
3418	RuleDestination	DESTINATION	00-01
7818	SCU	SINGLEREG...	00-01-00
7818	SCU	SINGLEREG...	00-01-01
7820	SubRule	RULE	01
7822	Condition0	CONDITION	01-00
7823	SCU	SINGLEREG...	01-00-00
7821	RuleDestination	DESTINATION	01-01
7824	SCU	SINGLEREG...	01-01-00
7825	SCU	SINGLEREG...	01-01-01
7827	SubRule	RULE	02
7820	RuleDestination	DESTINATION	02-00
7830	SCU	SINGLEREG...	02-00-00
7829	SCU	SINGLEREG...	02-00-01





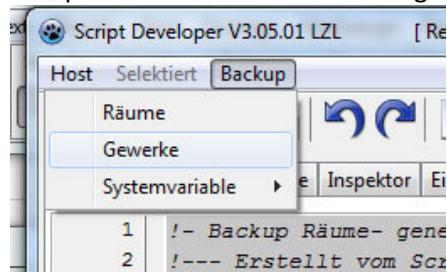
Und hier im WebUI Script

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
40	Kanäle von diesem Device umbenennen		HmIP-RF

Rückfrage mit Ja bestätigen und die Kanäle werden so benannt:

Device: DeviceName

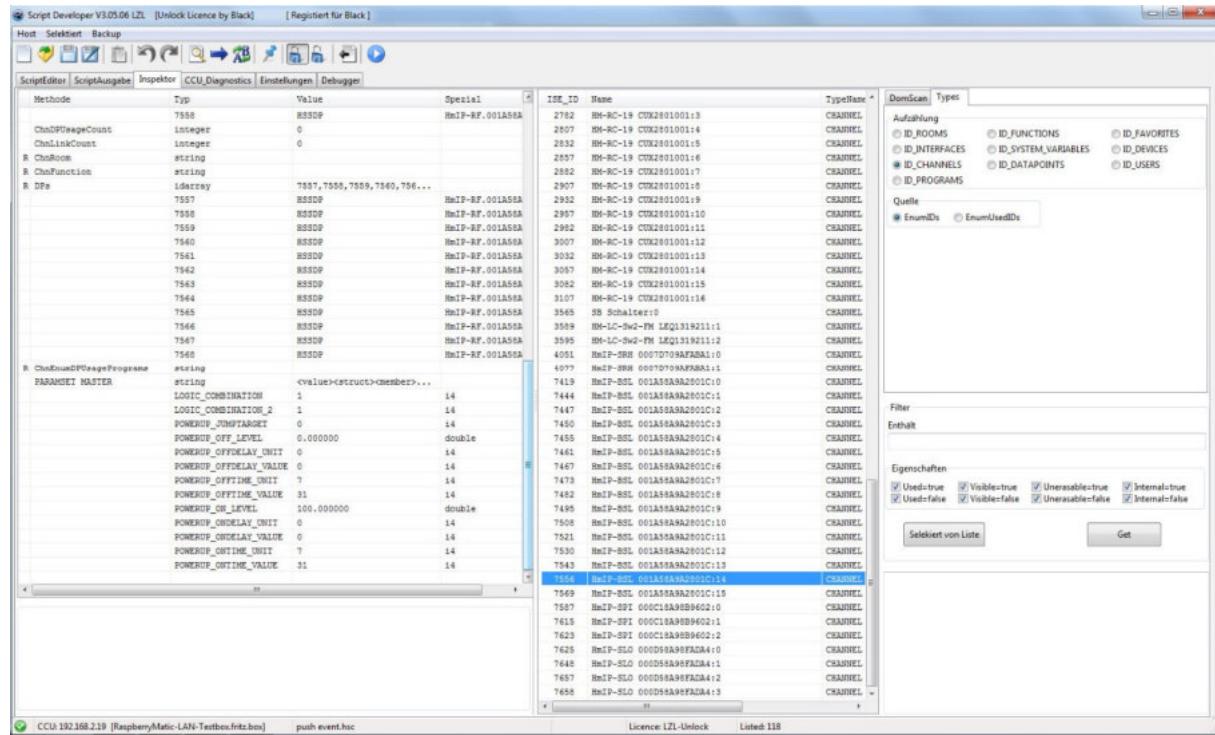
Kanal0 : DeviceName:0

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

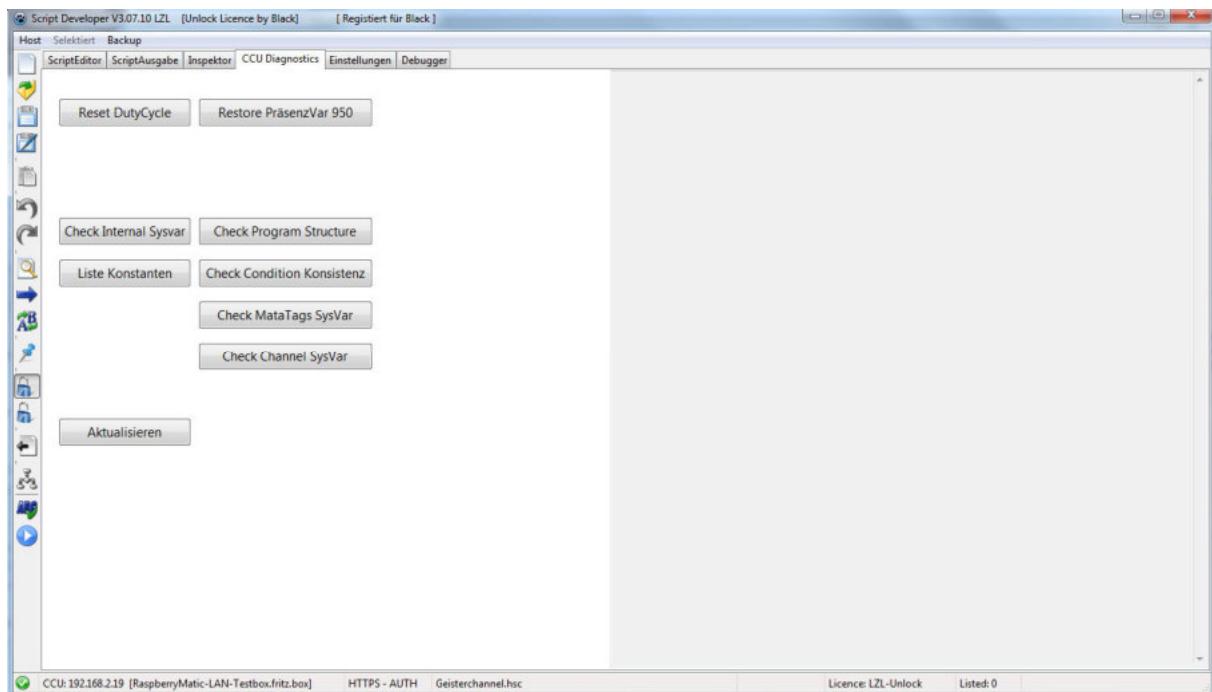
Rega Push: Damit lassen sich Datenpunkte innerhalb der Rega verändern, die sich normalerweise nicht verändern lassen. Z.B. Batteriefehler eines Netzaktors ^^. Setzt voraus, dass dieser Haken im der Konfiguration Datenpunkte gesetzt wurde:

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



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

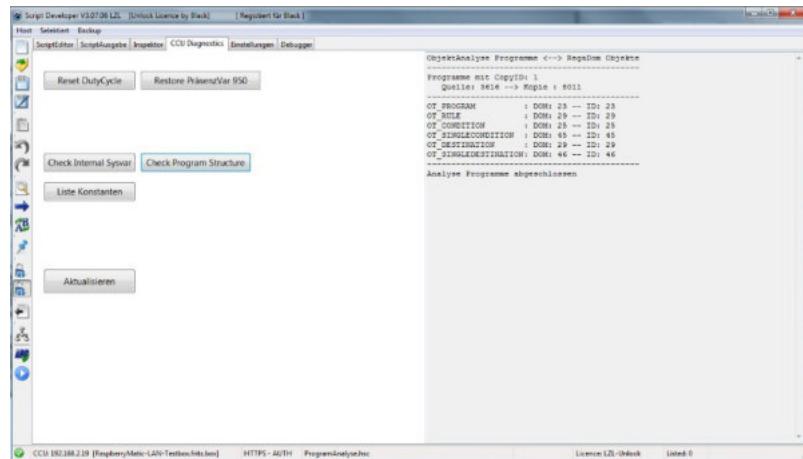
Alphabetische Auflistung aller bisher bekannten Systemkonstanten mit ihren Werten

The screenshot shows the Script Developer V3.07.10 LZL interface. The main window title is "Script Developer V3.07.10 LZL [Unlock Licence by Black]". The menu bar includes "Host", "Selektiert", "Backup", "ScriptEditor", "ScriptAusgabe", "Inspektor", "CCU Diagnostics", "Einstellungen", and "Debugger". Below the menu is a toolbar with icons for file operations like Open, Save, Print, and a search function. A central panel contains several buttons: "Reset DutyCycle", "Restore PräsenzVar 950", "Check Internal Sysvar", "Check Program Structure", "Liste Konstanten" (which is highlighted in blue), "Check Condition Konsistenz", "Check MataTags SysVar", and "Check Channel SysVar". Below these buttons is an "Aktualisieren" button. To the right of the buttons is a large list of system constants with their values. At the bottom of the window, there is status information: "CCU: 192.168.2.19 [RaspberryMatic-LAN-Testbox.fritz.box]", "HTTPS - AUTH", "Geisterchannel.hsc", "Licence: LZL-Unlock", and "Listed: 0".

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

7.2 Programmanalyse

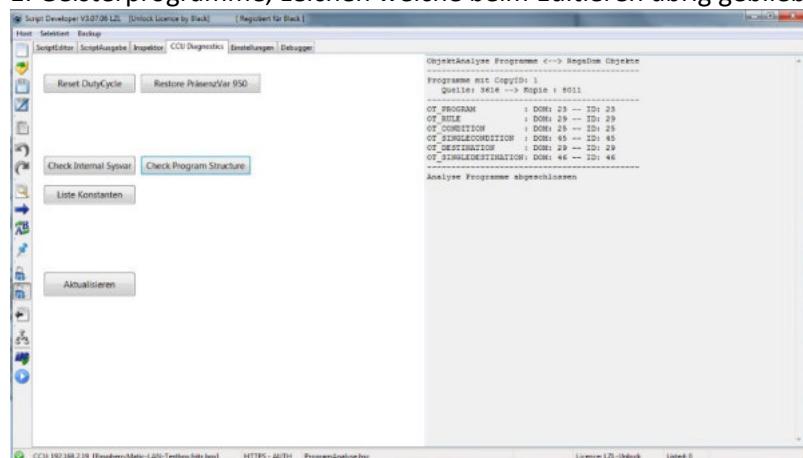
Mittels der Programmanalyse kann die Konsistenz der Programmobjekte überprüft werden.
So sieht der Aufruf auf, wenn die Prüfroutine erfolgreich über die Programme gelaufen ist.



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

Mögliche Fehlerbilder:

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



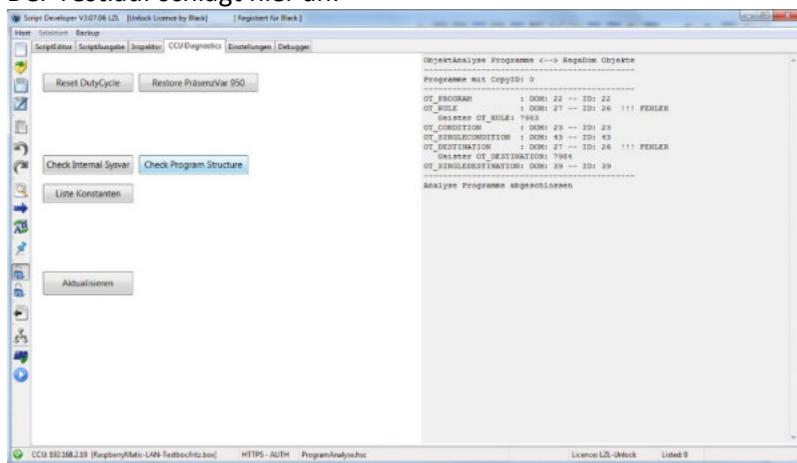
Hier wird das ursprüngliche Programm und die (Geisterkopie) daraus angezeigt. Der erste Versuch wäre: einen Reboot zu versuchen und zu überprüfen, ist der Eintrag des Geisterprogrammes weg oder nicht. Wenn nicht, kann manuell versucht werden, die Objekte der Kopie (nicht des Quellprogrammes) zu löschen

2. Programmfragmentreste

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

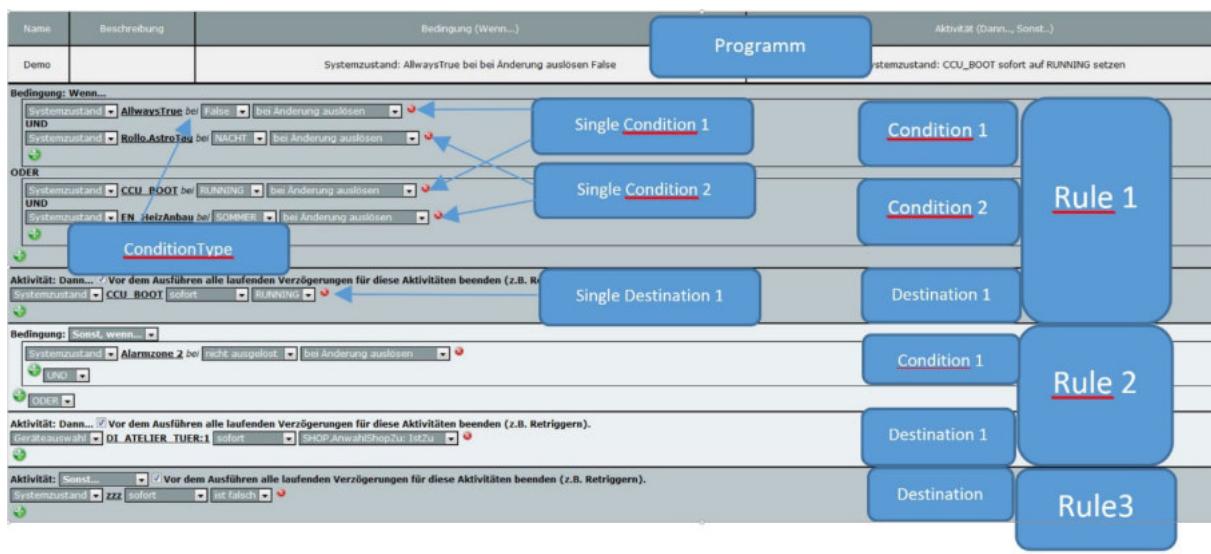
Provoziert habe ich es hier in dem Testfall, in dem ich ein RULE Object mal händisch angelegt habe.

Der Testlauf schlägt hier an:

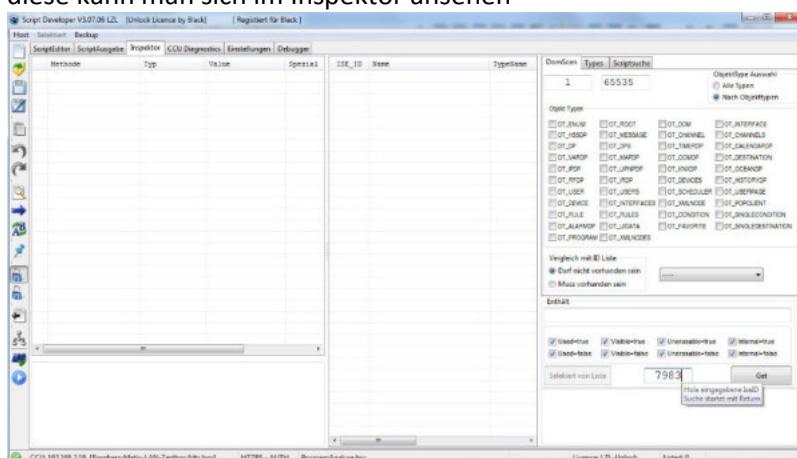


Es wird unter den Objecten OT_RULE und OT_DESTINATION angezeigt, dass sich in der Rega 2 Objekte tummeln, welche keinen Bezug zu dem Inhalt eines Programmobjektes haben. (2 deshalb, weil ein dom.CreateObject (OT_RULE) auch gleichzeitig das zu der Rule gehörende OT_DESTINATION Object erzeugt)

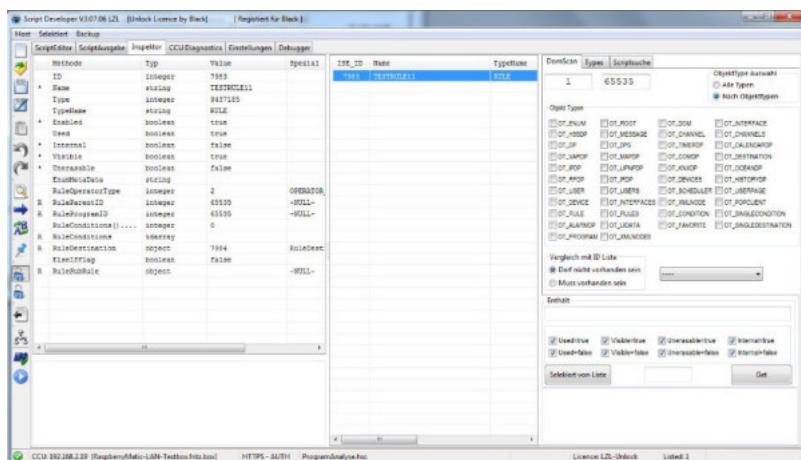
Ein Programm ist auf der CCU so Aufgebaut:



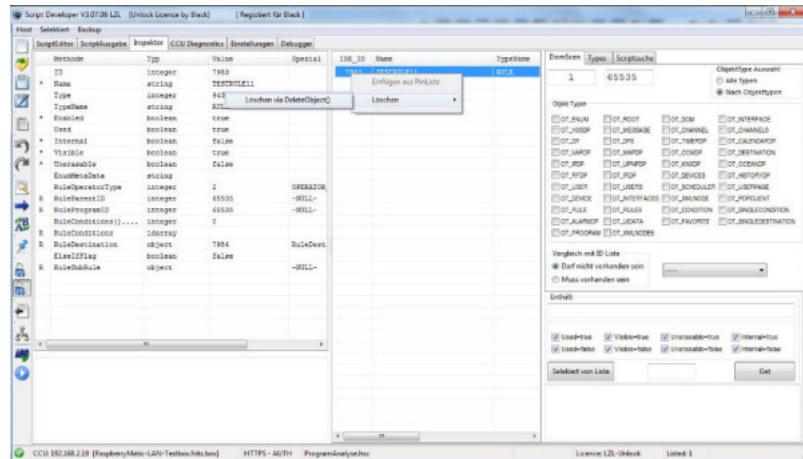
Die isID des gefundenen Objektes wird ausgegeben
diese kann man sich im Inspektor ansehen



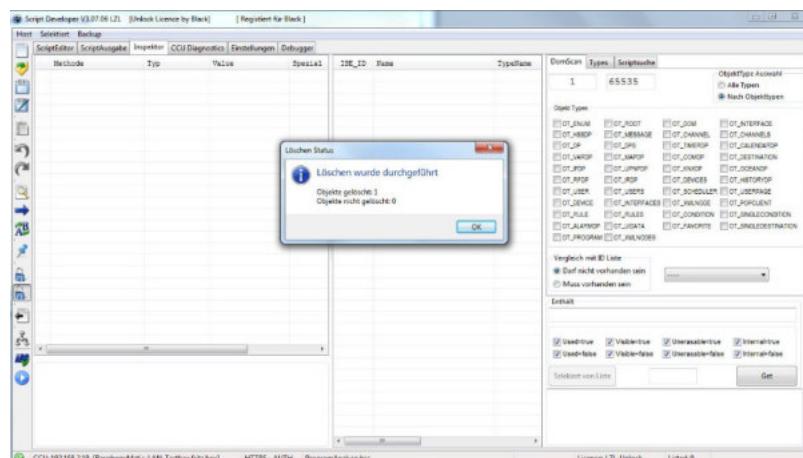
Dazu die gefundene ID in die Suchmaske eintragen und Enter drücken



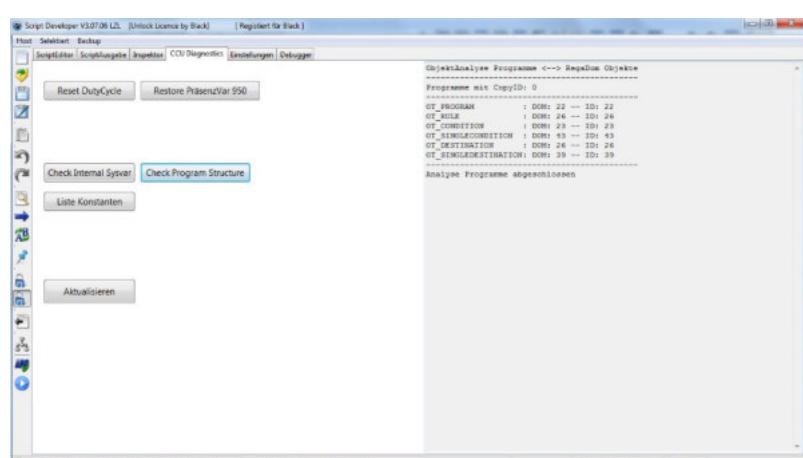
Man sieht, ein Object ohne Bezüge. es kann also entfernt werden. Löschen Freigeben, Programmunterobjekte müssen auch freigeben sein zum löschen, dann



es wurde gekillt



Die abschliessende Kontrolle zeigt:



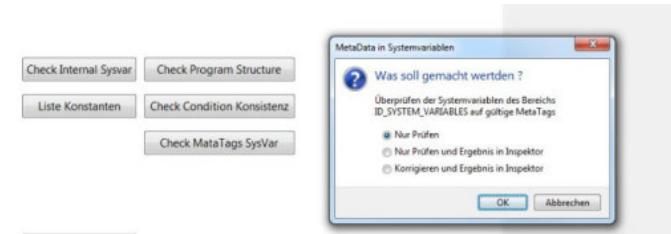
Es ist weg.

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

7.3 Systemvariablen Metadatenkonsistenz

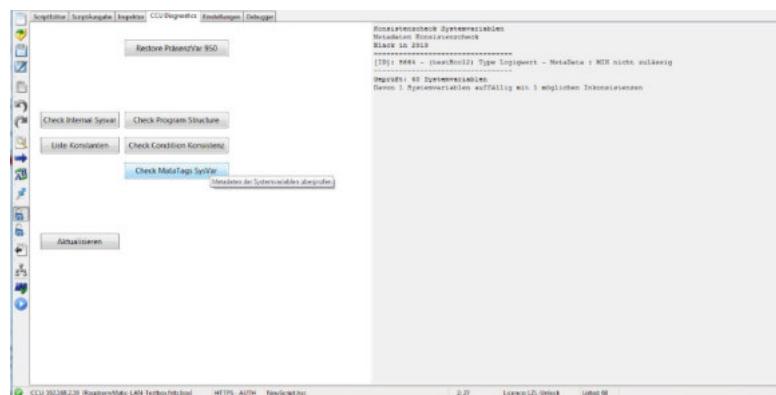
Dieser Menüpunkt überprüft die Systemdaten auf Konsistenz von dem Typ, welcher durch ValueType und ValueSubtype spezifiziert ist und den Einträgen unter EnumMetadata.

Druck erzeugt folgendes Menü:



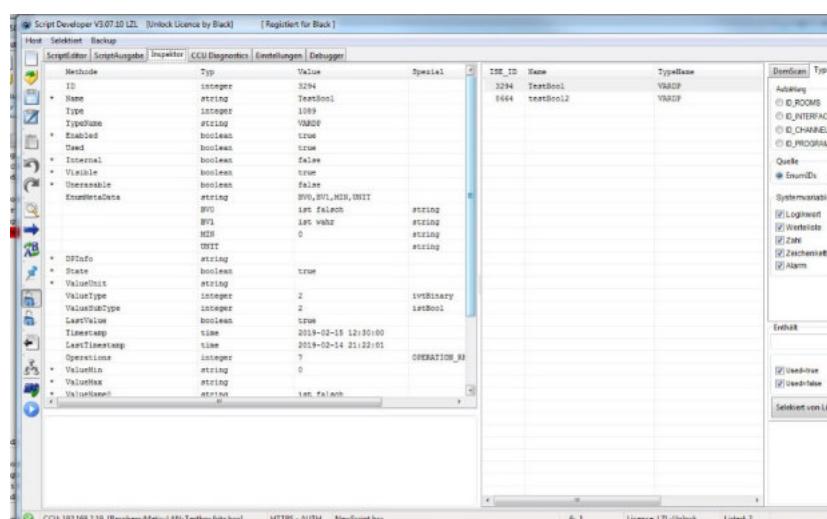
Punkt 1: nur Prüfen

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



Punkt 2: Nur Prüfen und Ergebnis in Explorer

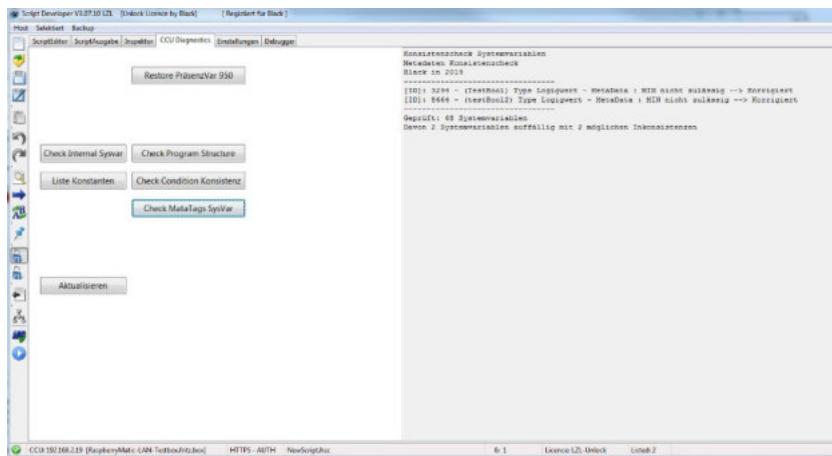
Genau wie Punkt 1, es wird eine Überprüfung gemacht, zusätzlich wird das Ergebnis noch in die Auswahlliste des Inspektors geladen. Dort kann überprüft und gegenbenenfalls auch manuell korrigiert werden.



Punkt 3: Korrigieren und Ergebnis in den Inspektor

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

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

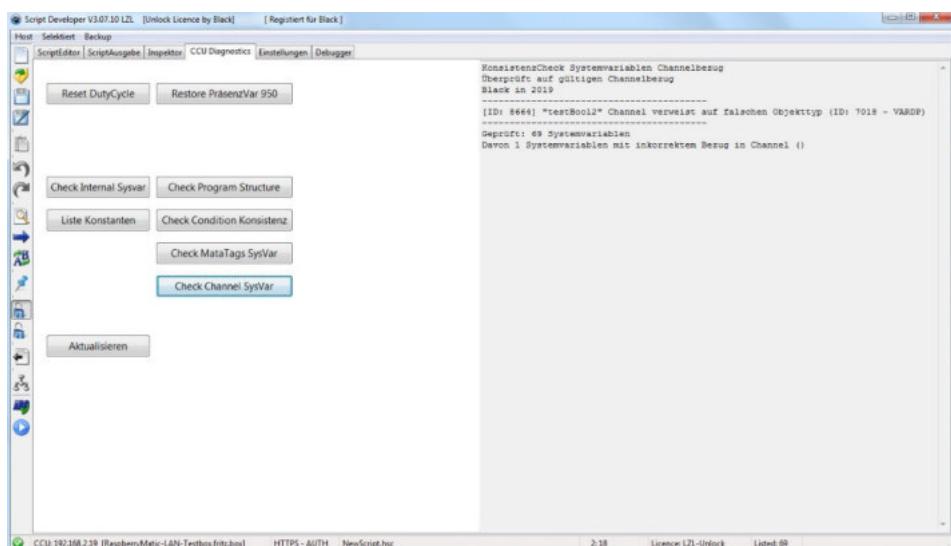
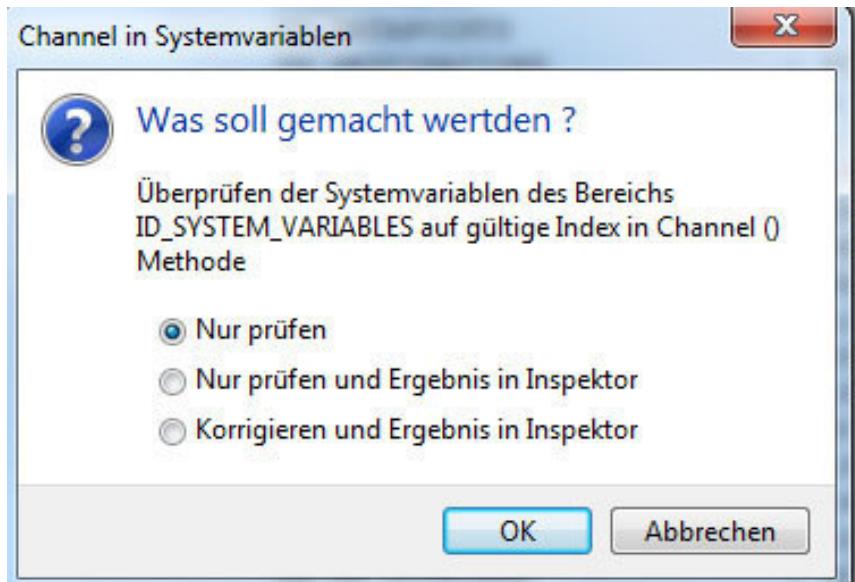


Ein nochmaliger Durchlauf bringt dann die Fehlerfrei-Nachricht.

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

7.4 Systemvariablen Test auf NirvanaChannelBezüge

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

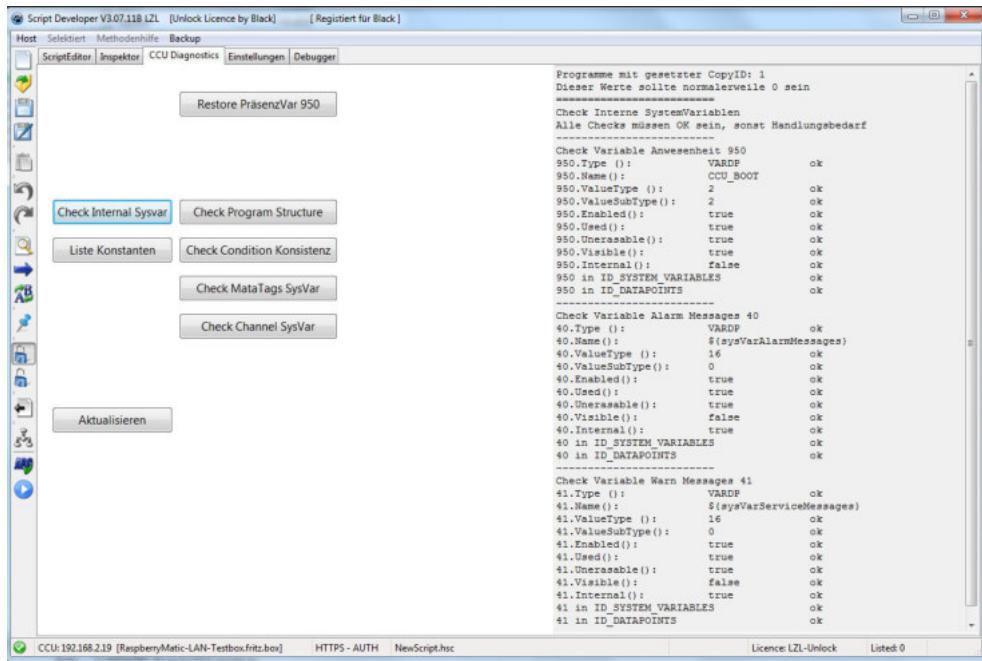


Die Auswahl Korrigieren bereinigt dann diese automatisch und lädt die veränderten Systemvariablen in den Inspektor.

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

7.5 Check Internal Sysvar

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



Alles Tests sollten OK melden, bei Fehler besteht Handlungsbedarf.

8 License Disclaimer

Mit diesem Programm werden 2 DLLs des OpenSSL Project mitgeliefert

Original Lizenztext:

LICENSE ISSUES

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

OpenSSL License

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

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

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

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

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

* must display the following acknowledgement:

* "This product includes cryptographic software written by

* Eric Young (eay@cryptsoft.com)"

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

* being used are not cryptographic related :-).

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

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

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

*

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

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

* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

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

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

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

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

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

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

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

* SUCH DAMAGE.

*

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

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

* copied and put under another distribution licence

* [including the GNU Public Licence.]

*/