

Kurzanleitung Script-Developer SDV V5.03.01A (15. Oktober 2023)

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

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

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

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

Dies ist eine BetaTestversion.

Die Verwendung dieser Software erfolgt auf eigenes Risiko

Der Autor dieser Software übernimmt keine Haftung für direkte oder indirekte Schäden, welche sich aus der Benutzung dieser Software ergeben sollten.

Eine kommerzielle Nutzung dieser Software ist untersagt

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

Inhalt

Kurzanleitung Script-Developer SDV V5.03.01A (15. Oktober 2023).....	1
1 Installation	8
1.1 Historische Entwicklung	8
1.1.1 Installation des SDV.....	11
1.3 Lizenierung.....	13
1.4 Donate	18
1.5 Systemvoraussetzungen.....	19
1.6 Was tut es bis jetzt	20
1.7 Bekannte Einschränkungen / Bugs.....	21
1.7 Geänderte Anforderung an Auflösung	21
1.7.1 Mehrfacher Start.....	21
1.8 Changelog V5.x.x	22
1.8.1 Changelog V5.01.02C.....	22
1.8.2 Changelog V4.09.05C.....	22
1.8.3 Changelog V4.09.01.....	22
1.8.4 Changelog V4.07.03E-HF1	22
1.8.5 Changelog V4.05.03.....	23
1.8.6 Changelog V4.02.07G	23
1.8.7 Changelog 04.01.10.....	23
1.8.8 Changelog 04.01.01.....	23
1.9 Neue Authentifizierungsmechanismen Raspberrymatik	24
1.9.1 SSL.....	24
1.9.2 Authentifizierung über Nutzernname / Passwort.....	24
1.10 Ein Wort des Authors zum Thema „gewerbliche Nutzung“	26
2 Oberfläche	27
2.1 Hauptmenu (Mainmenü).....	27
2.2 Toolbar	27
2.3 Arbeitsflächen (Pages).....	27
3 Skripteditor	28
3.1 EditorTabs.....	28
3.1.1 Öffnen eines Scriptes.....	29
3.1.2 Öffnen eines Scripts aus dem Inspektor (Extrakt aus der CCU)	30
3.1.3 Öffnen eines Scriptes durch das Betriebssystem.....	30
3.1.4 Öffnen der Scripte bei SDV-Start.....	30
3.1.5 Öffnen eines Skriptes aus der CCU und Zurückspielen	33

3.2 Voreinstellungen Editor.....	35
3.3 Vervollständigen Funktion	36
3.3.1 Verfeinerte Vervollständiger Funktionen.....	36
3.3.2 Spezielle Vervollständiger Funktionen	37
3.3.3 Qualifizierter Objektzugriff.....	37
3.3.4 AutoComplete	40
3.3.5 Vervollständigen über „Intellisense“.....	41
3.3.6 Autopopup Skriptvariablen	43
3.3.7 Komfortables Beenden und Weiterschreiben.....	44
3.4 Editormenü (rechte Maustaste)	45
3.4.1 Dateiinhalt an Cursorposition einfügen	45
3.4.2 Drucken	45
3.4.3 Toggle Bookmarks	45
3.4.4 \$src\$, \$this\$ und \$val\$.....	46
3.4.5 Xmlrpc.PutParamset:.....	46
3.4.6 Xmlrpc.getParamset:.....	46
3.4.7 Fenstergrösse Ausgabe speichern:.....	46
3.4.8 Block Kommentar	46
3.4.9 Editor Tastaturkürzel Hilfe.....	47
3.4.10 Überflüssige Leerzeilen entfernen	48
3.5 Word Markup (seit 03.07.02).....	49
3.6 SyncroEdit (ab 3.08.10)	50
3.6.1 Schrittgrößen und Schriftart (ab 3.08.07)	51
3.7 Methodenhilfe.....	51
3.8 Symbolischer Zugriff auf selektierte Datenpunkte des Inspektors	54
3.9 Automatische Codeerzeugung für Read/Write von Master/Linksets	55
3.10 Nützliche Tastenkombinationen des Editors.....	56
3.11 Editor Hint-Funktionalitäten.....	59
3.12 Blockdarstellung	60
3.13 Umfassen eines Textblockes	61
3.14 Crossreferenzliste Skriptvariablen.....	63
4 Inspektor.....	65
4.1 Selektionswahl: DomScan	66
4.1.1 Strategien für den Scan der Rega.....	69
4.2 Selektionskriterium Types	71
4.3 Zusätzliche Selektionsbedingungen	72

4.3.1 Enthält Filter	72
4.3.2 Anwenderdefinierte Filter	73
4.3.3 Schnellausführung von Get	80
4.4 Daten aus Inspektor in Editor übernehmen.....	81
4.4.1 Mehrfachauswahl als Enum String	83
4.4.2 Übernahme von einem Skript aus einem Programm direkt in den Editor.....	85
4.4.3 Einzelne Spalten in Zwischenablage kopieren.....	88
4.5 Selektion von Selektion	88
4.6 Objekte löschen.....	90
4.7 Anwenderdefinierte Sichten	92
4.8 Browsing durch Rekursionsebenen.....	93
4.9 Auflösen komplexer Objekte.....	96
4.9.1 Auflösen von Devices	96
4.9.2 Auflösen von Programmen.....	98
4.10 Zeitmodule	99
4.10.1 Ändern eines Zeitmodules.....	100
4.11 Suchen in Skripten nach Variablen, Devices etc.....	101
4.12 Volltextsuche in Skripten.....	104
4.13 Suchen Verwendungsstelle in WEB-UI Programmen (ab 3.09.04).....	106
4.14 Suchen „Name“ oder „Adress“ nach Verwendung in Skripten	107
4.15 Programm suchen verwendete Systemvariablen	107
4.16 SingleDestinations in ihrer Reihenfolge ändern.....	108
4.17 History Data Points und Systemprotokoll	111
4.18 EchtZeit Hilfe Hints zu Objekten im Inspektor	112
4.19 Ghost Objekte (Bezüge in Aufzählungen auf Objektleichen).....	115
4.20 Master und Linksets	117
4.20.1 Linksets	119
5 Favoritenansicht.....	122
5.1 Hinzufügen von Objekten in die Favoriten.....	122
5.2 Verwendete Objekte eines Programmes in die Favoriten laden	124
5.3 Favoritenliste löschen.....	124
5.4 Einschränkungen	124
5.5 Dynamisches Auffrischen von Favoriten	125
6 Systemprotokoll	126
6.1 Undock und Dock des Protokollfensters.....	127
6.1.1 Undock	127

6.1.2 Dock.....	127
7 Kleine Helfer im Alltag.....	128
7.1 Debugging von WebUi Programmen.....	128
7.1.2 Automatisiertes Einfügen von DebugSkripten	131
7.2 Backups.....	136
7.2.1 Räume.....	137
7.2.2 Gewerke	137
7.2.3 Systemvariablen	138
7.2.4 Devices und Kanäle.....	139
7.2.5 Backup Programme	139
7.2.6 WebUi Backup von selektiertem Device	144
7.2.7 Komplett Backup von Device.....	146
7.2.8 Backup Masterset.....	148
7.2.9 Backup Linkset.....	149
7.3 Umbenennen von Kanälen von Geräten.....	150
7.4 Paramset Master	151
7.5 Linkset eines Gerätes.....	153
7.5.1 Kopieren von Teilen eines Linksets einer DV in eine andere DV.....	154
7.5.2 Einzelne Parameter markieren und in Link oder Masterparameter einfügen	156
7.6 Rega Push auf Datenpunkte via Rega event.....	159
7.7 Querverweise zu Datenpunkten in WebUI-Programmen	160
7.8 Suche Referenzbezüge in Regadom	161
8 Diagnosebild	163
8.1 Allgemein.....	163
8.1.1 Liste Konstanten	163
8.1.2 Regadom.....	163
8.1.3 Systemübersicht	164
8.1.4 CacheInfo.....	164
8.1.5 Regadom Aktualisierung.....	164
8.1 SSH Funktionalität	165
8.1.1 SSH Realisierung (CUxD ab 2.3.1 oder plink.exe)	167
8.2 Diagnosen Systemvariablen	168
8.2.2 Check Internal Sysvar	168
8.2.3 Check metaTags Sysvar	168
8.2.3 Check Channel Sysvar.....	171
8.2.4 Check Sysvar Typkonsistenz	171

8.2.5 Restore Präsenzvar 950.....	171
8.3 Geräte Datenpunkte.....	172
8.3.1 Check Device-Interface.....	172
8.3.2 Check Datapoints Channel.....	172
8.3.3 Check Channel Linkcounts	172
8.3.4 Check History DPs.....	173
8.3.5 Check Enums	174
8.3.6 Check Objects	175
8.3.7 Check Devices < -- > XMLRPC	176
8.3.8 Check Virtual HmIP Keys.....	176
8.4 Programme.....	178
8.4.1 Check Program Structure	178
8.4.1 Check Condition Konsistenz	182
8.4.3 Cond- Destination Channel.....	184
8.4.4 Test und Korrektur Zeitmodule	184
9 Gerätekopieren	185
9.1 Kopieren Masterset.....	186
9.2 Gerätetausch	188
9.3 Kopieren Week Profiles	188
9.4 Kopieren Heizprofile.....	189
10 Direkte Verknüpfungen DVs.....	192
10.1 Löschen von Direkten Verknüpfungen	193
10.2 Sichern von Direktverknüpfungen.....	194
11. SDV Programmeditor.....	195
11.1 Speichermanagement	195
11.1.1 Allgemeine Definitionen.....	195
11.1.2 Speichermanagement der WebUI/ der Rega	197
11.1.3 Speichermanagement des SDV Programmeditors	198
11.1.4 Speicherstrategie des SDV-Programmeditors	198
11.2 Der Programmeditor	200
11.2.1 Positionsangaben beim Einfügen / Generieren	202
11.2.2 Menüleiste.....	203
11.2.3 Spalten des Programmeditors.....	205
11.2.4 Programmgruppen	207
11.3 Drag – Drop	212
11.4 Undo Redo.....	213

11.5 Exemplarische Aufgabenstellungen	214
11.5.1 Löschen einer Regel inmitten eines Programmes.....	214
11.5.2 Einfügen einer neuen Regel vor dem Wenn.....	217
11.5.3 Vertauschen des Wenn und des 1. Sonst-Wenn Regelblocks	218
11.6 Anzeigen bzw. Ändern von Programmen oder Programmunterobjekten	221
11.6.1 Programmobjekt (PRG).....	221
11.6.2 Programmhauptbedingung (Main-Condition).....	222
11.6.3 Regelblock	222
11.6.4 Bedingungsblock.....	223
11.6.5 Die Einzel-Bedingung.....	223
11.6.6 Anweisungsblock.....	226
11.6.6 Die einzelne Anweisung	226
11.7 Die Maincondition	230
12.1 OpenSSL.....	232
12.2 Synapse Ararat	235
12.3 Compiler und RAD	235

1 Installation

1.1 Historische Entwicklung

Bekannterweise wurden im Jahre 2017 sämtliche Programmversionen des damals verfügbaren Editors / Analyseprogrammes durch den Entwickler entfernt und waren mit der Argumentation, „(...) das Forum hätte es nicht verdient (...)\", nur noch einem elitären Kreis zugänglich. Einige User und ich hatten allerdings die Möglichkeiten erkannt, die sich durch eine gute Aufbereitung und vor allem Darstellung der Regadom Informationen ergaben. Da ich nicht unbedingt der Anhänger von monopolisiertem Wissen bin, begannen die Arbeiten an Alternativen, Ende des Jahres 2017 war ein ProofOfConcept erfolgreich gelaufen. Dieser SDV V1 bestand nur aus einem Editor, und einem primitiven Inspektor als reinem Textausgabefeld schnell an einem Abend coded in Scite. Die Schwächen dieses Inspektors zeigten sich recht schnell, keine Flexibilität. Es folgte der zweite Ansatz, der SDV V2, erstmals zu Jahresbeginn 2018 auf dem West-Stammtisch. Diese Frühform besaß schon einen Editor, und einen 2 spaltigen Inspektor für die Listen und die Detailansicht, ein Konzept, dass sich bis heute bewährt hat. Auch frühe Versionen dieses V2 konnten schon in der Anfangszeit Standartelemente der Rega analysieren sowie Skripte aus Programmen extrahieren und im Editor bearbeitbar machen. Intern war das Konzept, dass die RegaHss durch Skripte nur Rohdaten lieferte, die von dem SDV dann in Hochsprache aufbereitet und visualisiert wurden. Schnell wuchsen die Möglichkeiten, man lernte aus den Ergebnissen, die man im Inspektor sah und untersuchte. Allerdings wuchsen auch die Wünsche und Anforderungen, zum einem die persönlichen, zum anderen aus dem Tester und Stammtischkreis und aus dem Forum. Zudem hatte die V2 auch einen entscheidenden Haken, eigentlich als Testkonzept unter der Skriptsprache AutoIT in Scite entwickelt, erreichte das Konzept schnell seine Grenzen, die Geschwindigkeit einer reinen nicht objektorientierten Skriptsprache, viele Virensanner klassifizieren die ExecEngine von AutoIT als potentiell gefährlich (man kann damit auch wirklich Viren schreiben und böse auf dem TCPIP Stack rummachen wenn man's kann). Jedenfalls stand hier ein Cut an.

Auswahl musste eine Hochsprache sein, mit der sich GUIs programmieren lassen, aber ebenso Server und Clients umsetzen lassen, die Multithreading beherrscht sowie einen extrem guten Inline Debugger mitbringt, die ebenso eine entsprechende fertige Komponentenbibliothek hat, hierbei fiel auch ein Augenmerk auf eine gute Editorkomponente. Und die Programmiersprache sollte ich schon beherrschen, Babylon erfinden für so ein Projekt wollte ich auch nicht. Also fiel VisualBasic raus. Blieben noch C und Delphi. Aufgrund seit 1983 kontinuierlicher Programmiererfahrung unter Borlands TurboPascal Versionen und dem Nachfolger Delphi fiel dann die Wahl auf Delphi, im Hinblick auf die Lizenzproblematik von Embarcadero aber nicht auf Delphi selber, sondern auf den Opensource Ableger Free Pascal mit Lazarus als „Rapid Application Development“. Damit stand dann das Konzept fest. An einem Wochenende schon im Sommer 2018 entstand der noch sehr rudimentäre Grundzug des SDV 3 mit Editor, Inspektor, Debugger und Setup. Hier spielte die gewählte RAD ihre Vorteile aus, in der Komponentenbibliothek gab es einen vernünftigen Editor mit Highlighter und Completion Funktionallität und auch die ersten Testversionen fanden im Stammtischkreis und auch im Forum grossen Anklang. Um der Rega ihre Methoden zu entlocken wurde die RegaHss unter dem IDA pro Decompiler von Hexware einmal genauer betrachtet. Daraufhin existierten dann auch bei mir die vollständigen Methoden und Konstantenlisten. Es machte noch etwas Arbeit, diese zu sortieren und mit Versuch mach klug hinsichtlich ihrer Parametrierung zu untersuchen. Ab diesem Zeitpunkt wuchsen dann die Möglichkeiten des SDV exponentiell, neue Erkenntnisse flossen ins Programm ein, deren dargestellte Ergebnisse dann wieder zu neuen Erkenntnissen führen, quasi das Moore'sche Prinzip. Und mit jeder Neuerung wurde letztlich das Arbeiten mit der CCU einfacher.

Im Sommer 2020 kam dann bedingt durch die von mir hochbejubelten Braking Changes der Rega der SDV in der Version 4. Das reine codebasiere Konzept kam da allerdings beim Editor an seine Grenzen, Hilfestellungs-Hints oder gar Pre-Syntaxanalyse waren so nicht allgemein umsetzbar. Dafür bräuchte es bekannterweise eine schnelle Codebasierte Datenbank und so wie es FPC in seinem Editor umsetzt, einen Analyse Background Task, der die syntaktische Vor-Analyse macht. Hinzu kamen noch ein paar Unbekannte aus dem Syneditor selber, welcher ja nicht unbedingt ausführlich dokumentiert ist. Die Hürde mit der Dokumentation löste sich mit der Übersetzung der „Biblia de la Synedit“ aus dem spanischen mit Hilfe eines spanisch sprechenden Arbeitskollegen. Ab da ging es dann drastisch schneller, es folgten Foldings, Sprungmarken etc. Den letzten Ansporn, den Editor und den Highlighter auf Datenbankbasis umzustellen bekam ich durch die vollmundige Behauptung, es gäbe nur einen „vernünftigen“ Editor, und der SDV Editor sei eh Schrott- nun denn, im Spätsommer verbrachte ich berufsbedingt so einige Zeit in den Terminals verschiedener Flughäfen und so entstand die Grundlage der Datenbank basierten Version während der Wartezeit auf den Flughäfen Wien, Danzig und Bukarest. Damit ließen sich dann das Highlighting und die Syntaxprüfung realisieren und quasi als Abfallprodukt, da die Datenbank ja nun eh schon da war, gab es noch die ultraceheime Skriptdoku. So geheim ist die nun wirklich nicht, und die Anzeigeaufbereitung aus der Datenbank brauchte maximal 200 Programmzeilen.

Einen weiteren größeren Schritt im Editor gab es im Februar 2022 mit der Einführung des LL1 Parsers/Lexers erstmal nur zur genaueren Methodenvorhersage bei der IntelliSense Funktionalität. Diese auch als hoch geheim und unendliche schwierig eingestufte Funktionalität, das übliche Gepolter hält, diese Algorithmen sind so ultraceheim, die wurden schon zu meiner Studienzeit Anfang der 1990er Jahre in den Vorlesungen zum Thema „Compilerbau“ gelehrt. Und die unendliche Schwierigkeit bei der Programmierung war eigentlich nur – meine alten Vorlesungs-Skripte auf dem Dachboden zu finden.

Es folgten bis Juni 2023 nur partielle Verbesserungen / Erweiterungen begründet darin, dass ich eigentlich Möglichkeiten zur Erweiterung der WebUI schon im Jahre 2023 aufgezeigt hatte. Mich störte es immens, dass die WebUI kein Drag& Drop hat, dass sich Regeln mitten in einem Programm weder einfügen, verschieben noch löschen lassen, ohne dass dieses einem Neuschreiben des Programmes gleichkommt. Sicher, es gab Möglichkeiten dieses umständlich über Skripting im Zusammenspiel mit dem Inspektor zu machen, aber Normal-Anwendertauglich ist etwas Anderes. Ansätze dazu hatte ich. Damit es aber auch für die WebUI tauglich würde, hätte es ein paar zusätzliche Methoden bedurft, worauf ich schon 2020 ein Issue mit jährlicher Wiedervorlage im entsprechenden Github eingereicht hatte. Es tat sich in der Richtung allerdings nicht, null, niente.

Also ist mir irgendwann mal der Kragen geplatzt, an einem mal wieder Warte Aufenthalt an einem Flughafenterminal (das scheint sehr produktiv für die Entwicklung zu sein), habe ich angefangen den Programmedit des SDV umzusetzen. Da ja kein Interesse bestand, dieses auch WebUI tauglich auch in der CCU direkt umzusetzen, standen mir ja alle Wege offen und ich konnte Programm und Komponententechnisch alle Register ziehen ohne Rücksicht auf Strukturen und Gegebenheiten der WebUI nehmen zu müssen. Er ist noch nicht ganz fertig, erlaubt aber jetzt schon eine Vielfalt an Funktionen die es so in der WebUI nicht gibt (und höchstwahrscheinlich bei gegebener Struktur auch nicht geben wird)

Welche Schwerpunkte ich beim SDV setze lässt sich grob aus der Anzahl der Programmzeilen für die einzelnen Funktionalitäten ableiten. Die Version 5.02.01 hat etwas über 81000 Programmzeilen (81000 von mir geschriebene Codezeilen, ohne Kommentare und die von Lazarus schon in der RAD implementieren Komponenten (Syneditor, SynEditHighlighterFoldBase als Highlighter, alle visuellen Komponenten, crypto units und Netzwerktools nicht mit eingerechnet) auf die 81000 Zeilen verteilen sich funktional (aufgeteilt, stellenweise überschneiden sich auch cases). Mit der Version 5.02.xx ergibt sich folgende Aufteilung von Programmzeilen zu Funktionen.

Ca. 500 Zeilen - Allgemeines (Load / Save von Einstellungen, Fenster Splits)
ca. 1000 Zeilen - CCU Kommunikation und Auswertung, Thread und Exception handling
ca. 600 Zeilen - Crypto und Licensing
ca. 1400 Zeilen - Child der Synedit Class incl. der Completion (benutzt die Datenbank)
ca. 850 Zeilen - Child der SynEditHighlighterFoldBase
ca. 518 Zeilen - Multi Tab Editor
ca. 4800 Zeilen - Code Datenbank und deren Initialisierung (Methoden, konstanten, Syntax, Beschreibung)
Ca 1800 Zeilen – der Tokenizer und der LL1 Parser
Ca 1000 Zeilen – Die RegaAbstractionUnit
ca. 24000 Zeilen - Inspektor (Childs von TListView) mit allem was da so zugehört, wie Paramssets, Direktverknüpfungen
ca. 400 Zeilen - Favoriten
ca. 200 Zeilen - Systemprotokoll
ca. 400 Zeilen - Setup
ca. 200 Zeilen - Die Skriptdoku (ist ein Abfallprodukt der Code Datenbank)
ca. 6000 Zeilen - CCU Services
ca. 20000 Zeilen - Die Verschiedenen unscheinbaren Backups, die der SDV ermöglicht (Programme, Geräte, WebUi etc.)
ca. 13000 Zeilen – Der SDV Programmeditor (in der Form Stand August 2023)
ca. 5500 Zeilen - Der große Punkt Gerätekopieren (Heizprofile zwischen HM und HmIP, Wochenprogramme, Gerätetausch) nur die separaten Zeilen, setzt auf classes der Backups auf)
ca. 5000 Zeilen (nicht genau zuordbar, allgemeines, grundlegende Classes und Deklarations, JSON Interpreter, die von vielen Punkten gebraucht werden, interner Debugger)

1.1 Installation des SDV

Das *.rar File in ein beliebiges Verzeichnis entpacken. Ein Installer ist nicht notwendig. In diesem Verzeichnis befindet sich auch das Konfigurationsfile SDV4.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
SSHLINK=true
THREADKILL=15000
THREADKILLXML=20000

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

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

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

[SECURITY]                                    Die Ports die der SDV benutzt
HTTPREGASCIPT=8181
HTTPREGAXMLRPC=1999
HTTPSREGASCIPT=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

Wie bekommt man das ganze nun in Funktion

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

Und auch wichtig, dieser Fehler kommt auch oft vor: Der Nutzernname darf nicht Testuser sein !

Bitte dann in der INI ändern, die INI speichern, SDV neu starten und neuen Anfragecode generieren.

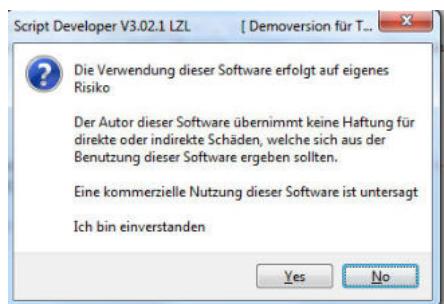
1.3 Lizenierung

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 ?

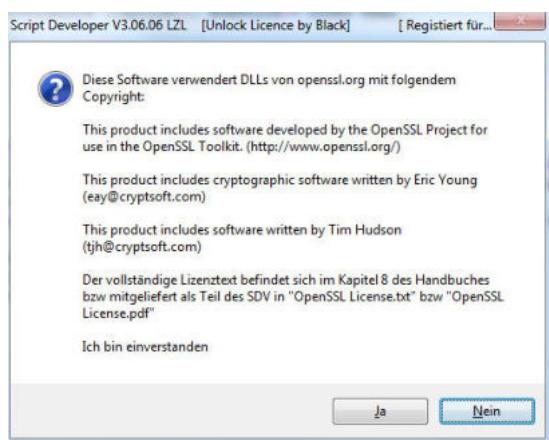
Der SDV telefoniert nicht nach Hause. Um eine Lizenz anzufragen ist folgender Weg einzuschlagen.

1. Die Konfigurationsdatei SDV4.INI mit einem Editor öffnen.
2. Nickname anpassen
3. IP der CCU 1 eintragen
4. IP der CCU 2 eintragen
5. CUXD Kanal eintragen
6. Konfiguration abspeichern
7. Script-Developer starten



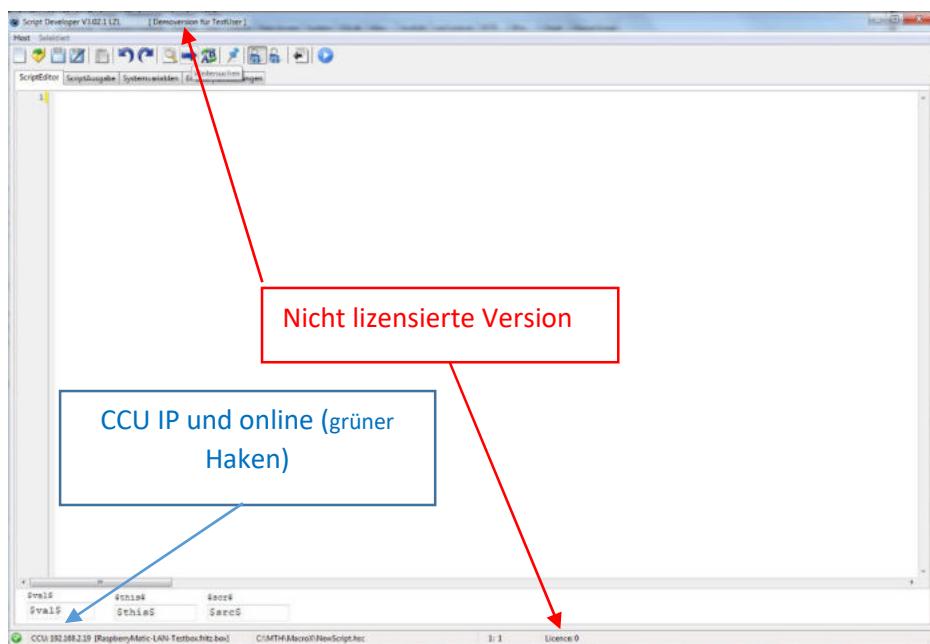
Bei allerersten Start muss dieses Fenster mit yes bestätigt werden. No führt so einem sofortigen Programmabbruch

Ab der Version 3.06.06 befinden sich in dem Package des SDV 2 DLL's von OpenSSL.org, welche für den HTTPS Zugang zur CCU benötigt werden. Hierbei ist einmalig ebenfalls eine Zustimmung über die Verwendung der Software nötig. Der komplette Lizenztext kann im Kapitel 9 nachgelesen werden bzw. ist als Datei mit in dem Package enthalten.

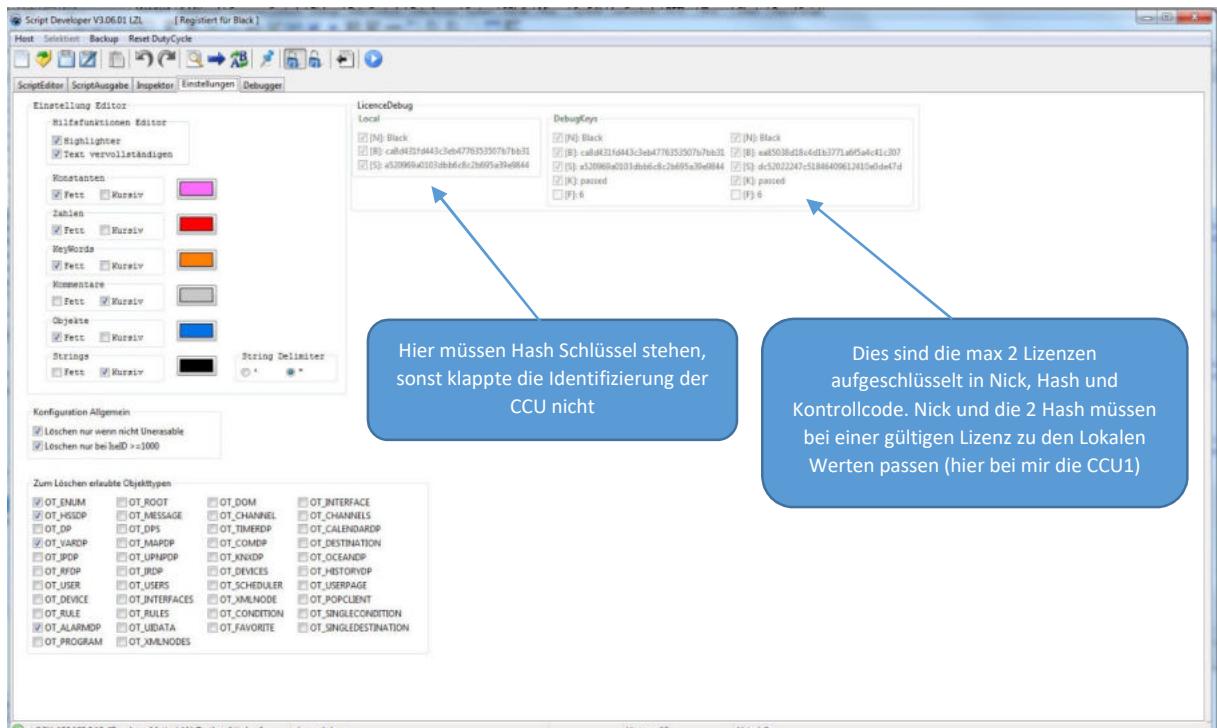


„Nein“ führt auch hier zu einem sofortigen Programmabbruch

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



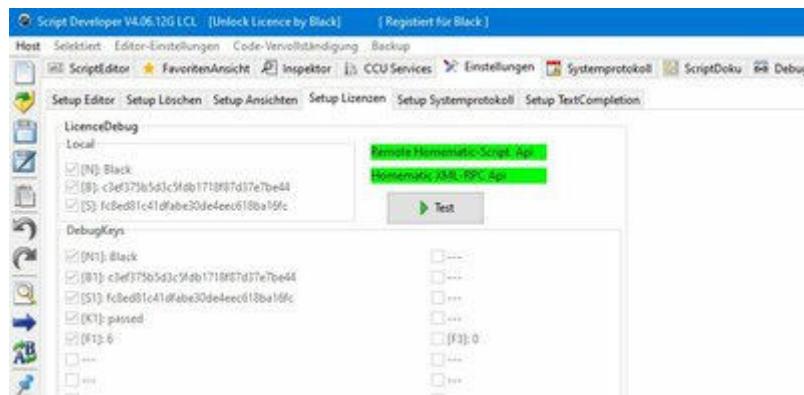
Wenn die CCU, für die der Schlüssel angefragt werden soll, als grün angezeigt wird, bitte vorher einmal unter dem Reiter Einstellungen kontrollieren



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

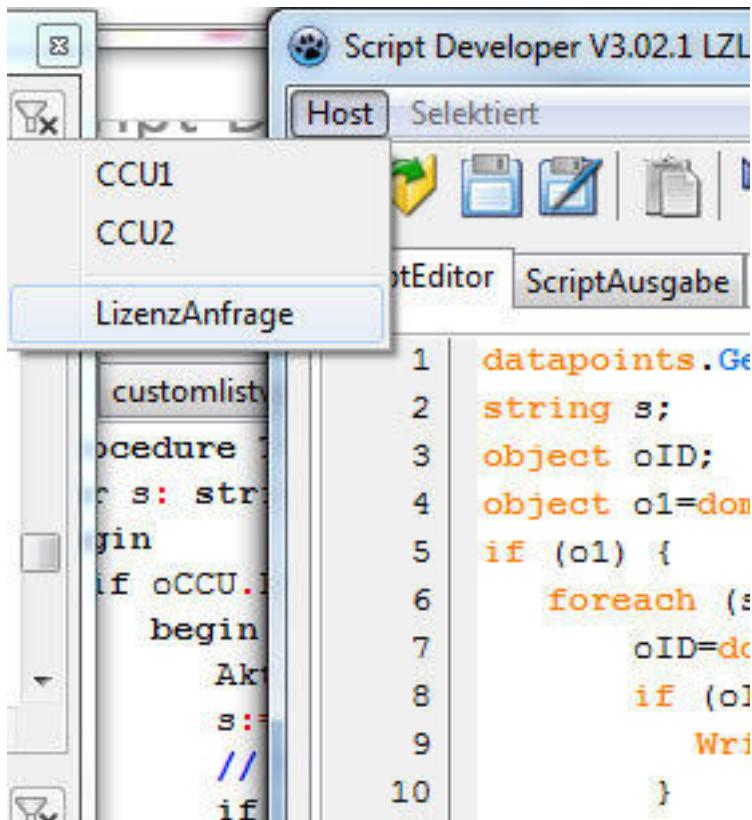
Im Auslieferungszustand ist unter NICKNAME= noch TestUser eingetragen. Mit TestUser lässt sich kein Key generieren. Hierbei dann bitte die INI nochmal anpassen und den SDV neu starten-

Im Service unter Debug Lizenzen Möglichkeit zum Test der Verbindung programmiert. Also bitte erst Anfragecode generieren, wenn der test 2 mal grün ergibt. wenn nicht, stimmt etwas in der Konfiguration der Firewall bzw. wenn Authentifizierung vorgegeben ist, in der Authentifizierung nicht. Das sehe ich dann spätestens im Licencer von mir

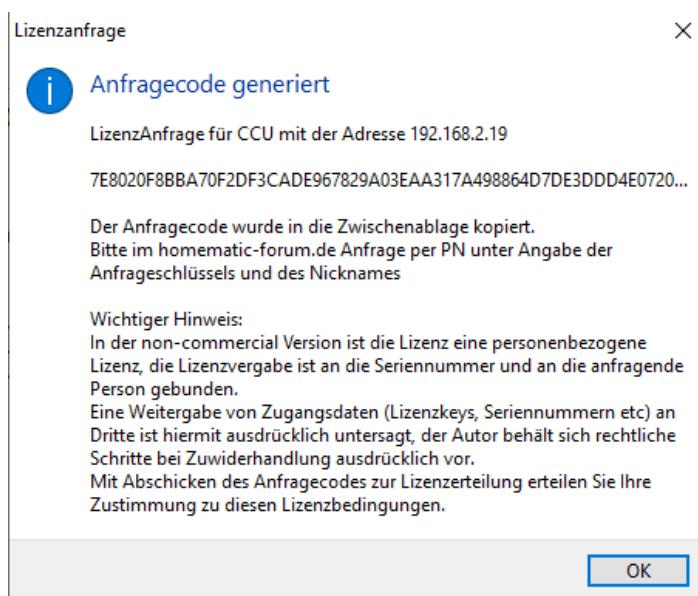


Den Punkt gabs zwar schon seit ein paar Versionen. Gelb bedeutet: Test läuft grade, rot: war ein Griff ins Klo, grün: tut's

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.

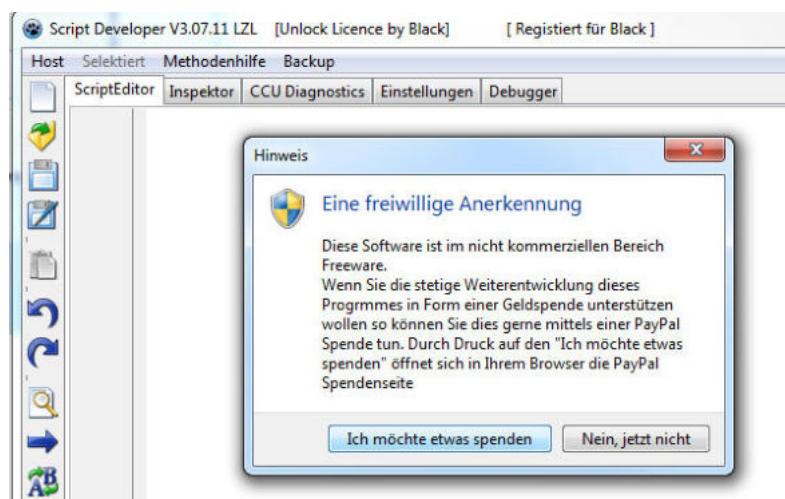
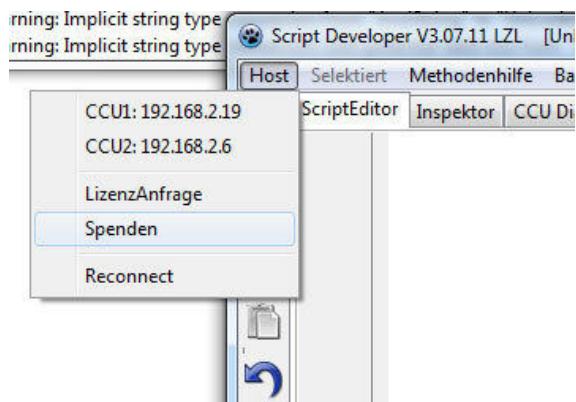
Geplant hab ich folgende Lizenzabstufungen

Level	Editor	Skript 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

Ach so, noch ein Tipp an denjenigen, der versucht, den Code für meine Unlock-Licence in dem Code zu finden.... Eher findest du den Heiligen Gral, das ist eine Compiler Direktive, die beim Kompilieren an x Stellen anderen Code erzeugt bzw. manche Punkte und Funktionen vom kompilieren ausnimmt und keine einfache Variablenabfrage zur Laufzeit... so als Tipp halt ^^

1.4 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.



1.5 Systemvoraussetzungen

Mittlerweile wird der SDV V4 unter einer 64bit Umgebung kompliert. Getest und Lauffähig ist die Anwendung unter Win10/64bit, Win7/64bit sowie unter entsprechenden VMs unter Linus bzw Apple Umgebungen

Auf der Homematic-Seite wurde bei mir auf einer Raspberrymatic 3.65.11 getestet.

Aufgrund der breaking Changes der 3.51.6.20200621 gilt Folgendes:

Ab Rega version der 3.51.20200621 arbeitet der SDV V4 komplett, in älteren Versionen sind die Backup Funktionalitäten, die Gerätekopien und alles, was mit den Schnittstellenprozessen direkt arbeitet, disabled bzw ausgeblendet. Wird dieses benötigt, so muss die Last Legacy Edition der SDV V3 eingesetzt werden.

Eventuell werde ich im Laufe des Jahres 2021 mal schauen, ob ich es nativ unter Debian auch kompiliert und zum Laufen bekomme. Wahrscheinlich wird dieses Projekt aber zurückgestellt, da leider das Threadhandling unter Linux sich völlig anders wie Windows verhält.

Windows11:

Es gibt Stand November 2022 43 User, bei denen der aktuelle SDV problemlos läuft und einer, wo wohl massive Probleme auftraten. Ursachen und Hintergründe kenne ich noch nicht, da ich noch nicht Windows11 einsetze.

1.6 Was tut es bis jetzt

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

Undo / Redo arbeiten

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

Enums und Sysvars arbeiten auch schon inkl Detaildaten und Editiermöglichkeiten.

Darstellbarkeit zumindest der Grundmethoden aller Objekte

DomScan

Devices

Aufschlüsseln der MetaDaten

Datenpunkte

Kanäle

Programme

Favoriten

User

Paramset Master

Paramset Links

Skript Bearbeitung aus Programmen heraus

Verändert von Verzögert um von Skripten

Editor mit bis zu 32 SkriptTabs (Früher 16, ab 5.02 32 Tabs)

Verändern von Retriggern

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

Destination und SingleDestination

MarkupUp im Editor

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

Suchen in Skripten nach Seriennummer von Devices und Channels

VollTextsuche in Skripten auch mit Regex Ausdrücken

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

Backup von Mastersets

Backup von LinkSets

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

Diverse Konsistenstests

Auflösen von Direktverbindungen

Backup von WEBUi Programmen und Einstellungen eines Gerätes

Devicekopien auch unter „ähnlichen“ Geräten

Selektives Kopieren von Linkparametern zwischen Direktverknüpfungen

1.7 Bekannte Einschränkungen / Bugs

~~Auswahl dialoge sind auf Englisch. Weiß ich, zurzeit benutze ich die in der Laufzeitumgebung integrierten Dialoge, und die sind leider trotz Landeseinstellung englisch.~~ (Edit: mittlerweile geändert)

~~Folding im Editor arbeitet noch nicht. Wenn der Rest läuft gucke ich da mal nach.~~ (Edit: mittlerweile umgesetzt)

~~Kommentare im Skript 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~~ (Edit: mittlerweile umgesetzt)

Aufgrund dessen, dass als Middleware bei mir IOBroker läuft und ich die Diagramm und die History Funktion der CCU nicht nutze, werde ich diese im SDV auch nicht ausprogrammieren. (Edit: History ist mittlerweile programmiert, Diagramm gibt es aber nicht und wird es auch nicht geben)

Linkset Restore von Heizungsgruppen arbeitet noch nicht sauber und ist deshalb erstmal disabled.

~~Completion greift bei der Erkennung schon mal gerne ins Leere~~ Ist Stand Nov. 2022 auch gefixt

Der Programmeditor ist noch nicht abschließend fertig, bei der Entwicklung habe ich darauf Wert gelegt, zuerst die Funktionen zu implementieren, die die WebUI nicht hat. Was jetzt im SDV Editor noch fehlt, kann die WebUI. Bis zur finalen Version müssen da Programm Editor und WebUI zusammenarbeiten

1.7 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.

1.7.1 Mehrfacher Start

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

Die frühere Möglichkeit des mehrfachen Starts ist mittlerweile wieder abgeschafft worden, durch die Einführung des Multi Tab Editors sowie der Interprocess-Communikation wird ein durch Doppelklick auf ein Skript 2 fach gestarteter SDV keine zweite Instanz öffnen, sondern seine Startparameter via IPC an den Server des Hauptprozesses übertragen und dann wieder beenden. Die gewünschte Datei wird dann im EditorTab des 1. SDV geöffnet und kann dort bearbeitet werden

Zu empfehlen ist auch, zB. die Dateiendung .hsc mit dem SDV zu verknüpfen, so öffnet ein Doppelklick auf eine Skriptdatei dann automatisch den SDV mit dem geladenen Skript

1.8 Changelog V5.x.x

1.8.1 Changelog V5.01.02C

Lazarus RAD aktualisiert auf V2.2.4
Programmeditor begonnen
Central Links lassen sich löschen
Anzeigen von Ghosts auch in der Listenansicht
Im Editor Möglichkeiten nicht druckbare Zeichen anzuzeigen
Automatischer Vorschlag der Skriptvariablen beim Schreiben nach einstellbarer Zeichenanzahl
Reparaturmöglichkeit für verlorene gegangene Channels bei den internen Systemtasten
Einige Verbesserungen und Bugfixes
Änderung des Scan-Algorithmus für DomScan und interne Reparaturläufe
Vergleichslauf Paramset in Paramset Description

1.8.2 Changelog V4.09.05C

Lazarus RAD aktualisiert auf V2.2.0
Markierte Listen in der Detailansicht des Inspektors lassen sich mit CTRL-C kopieren und z.b. in Excel einfügen
Filtermöglichkeiten im Inspector erweitert (Maincondition etc) #
Zeitmodule lassen sich nun komfortabel in einem eigenen Menü ändern und aktivieren
Systemprotokoll lässt sich im SDV nun ausdocken und z.b. Auf einen zweiten Bildschirm legen
This,src und value nun für jeden Skripttab Separat
Completion von Skriptvariable wesentlich verbessert

1.8.3 Changelog V4.09.01

Verbesserte Algorithmus zur Intellisense Methodenvorhersage.
Verschiedene ZusatzMenüs wie:
Crossreferenzliste Skriptvariablen
Auswahlmenü mit Automatischer Erzeugung von vollqualifizierten Zugriffen
Verbesserungen im Highlighter und der Syntaxerkennung
Verbesserungen im Inspector (zusätzliche Menüs und Veränderbarkeit von Objekten)

1.8.4 Changelog V4.07.03E-HF1

Verbesserungen am Editor (Block EIn/Ausrücken)
Umfassen Funktionalität im Editor
darstellbarkeit im Editor von {} und () Bereichen
Programme reparieren/Kopieren Funktionalität
Verbesserungen im Inspector zur Werteänderung an normalen Datenpunkten / eigentlich nicht beschreibbaren Datenpunkten
Inspector Möglichkeit Master,LinkSets zu verändern
Kommentarmöglichkeiten für alle Objekte eingefügt
Direktes Editieren von Metadaten
Komfortables Editieren von Metadaten
Verschieben von Link Paramsets in andere Direktverknüpfungen
Verbesserte Löschalgorithmen
Wesentliche Verbesserungen von Hintinformationen von Master / Linksets
Einfache Testmöglichkeit, ob die Anfrageeinstellungen in Ordnung sind (2 mal grüne Ausgabe)
Inspector verbessert, damit dieser auch mit Ghost Objekten klarkommt und diese darstellen kann
idarray: Reihenfolge eingehängter Objekte einfach verschiebbar
Skriptdoku weiter verbessert
Inspector Hint Informationen wesentlich erweitert
Zyklische Abfrage des Systemprotokolls implementiert
Inspector wesentlich schneller gemacht: Optimierung Classen und Umstellung auf direkten Schnittstellenprozesszugriff

1.8.5 Changelog V4.05.03

Ab dieser Version Umstellung des Highlighters auf eine Datenbank. Damit bildet der Highlighter nun die vollständige Regasyntax ab

Codebasierte Methodendatenbank
vollständige Regasyntaxunterstützung des Highlighters
Echtzeit Syntaxprüfung im Editor (Warn und fehler)
Überarbeitetes Code Completion
Komfortables Auto-Complete (Einfügen completer Code Schnipsel)

1.8.6 Changelog V4.02.07G

MultiTab Editor
diverse Bug Fixes

1.8.7 Changelog 04.01.10

Bugfixes
Erweiterte Hints
Bookmarks
Einführung des MultiTab Editors
Drag/Drop zum Kopieren von Parameters von Linksets in einen gleichartigen anderen Linksets
Erweiterung des Highlogthers
Eingeschränkt lauffähig auch auf alten Regas, hier aber ohne Backup und Gerätekopie

1.8.8 Changelog 04.01.01

Die Breaking Changes version, als v4 nur noch Lauffähig auf der neuen Rega.

Die alten V3 Versionen betrachte ich als obsolet, diese sollten auch nicht mehr eingesetzt werden.

1.9 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.

1.9.1 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.

1.9.2 Authentifizierung über Nutzername / Passwort.

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

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

Beispielhafter Aufbau der neuen INI

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

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

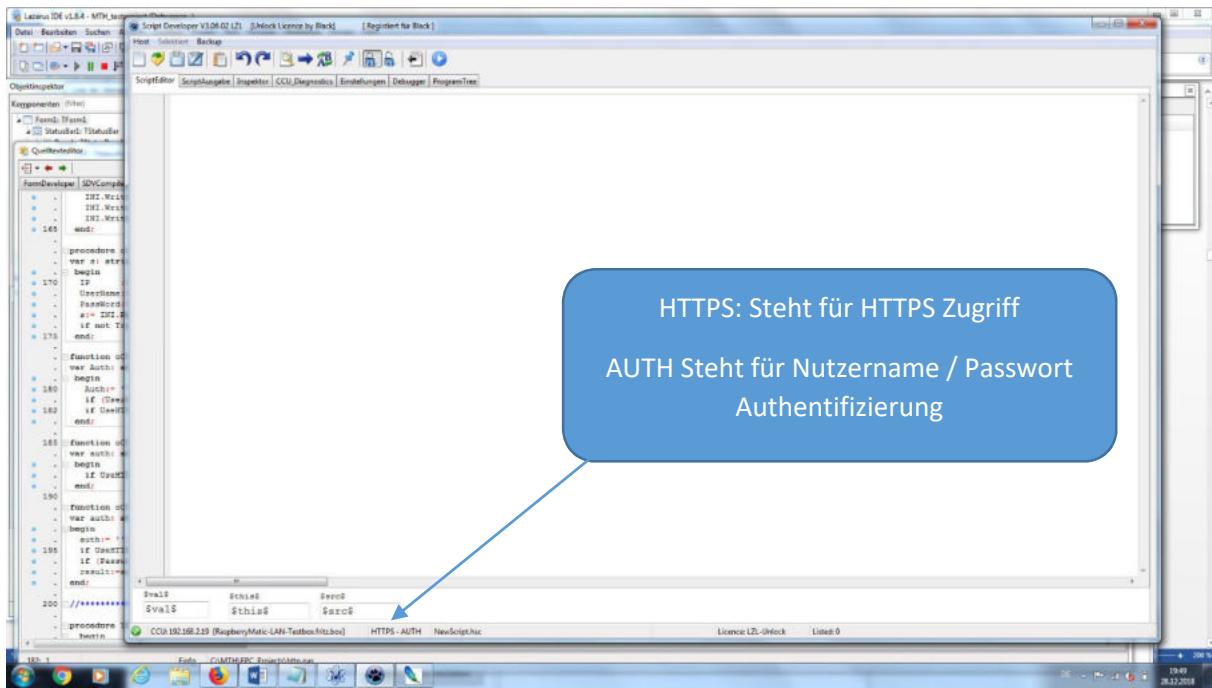
[CCU1]                                     Neuer Schlüssel
IP=192.168.2.19                           Ip Von CCU1
USERNAME=Admin                             Nutzernname auf der CCU1
PASSWORD=XXXXXXXXXXXX                      Passwort der Nutzers auf der CCU1
USEHTTPS=true                             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

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

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

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

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



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

```
function oCCUNetwork.CCURootString: string;
var Auth: string;
begin
  Auth:= '';
  if (UserName<>'') and (PassWord<>'') then auth:=UserName+':'+PassWord+'$';
  if (not HTTPS) then result:= 'http://'+IP+Auth else result:= 'https://'+IP+Auth+':'+Port+Auth;
end;

function oCCUNetwork.varResult: string
var auth: string;
begin
  if UseHTTPS then result:= 'https://'+IP else result:= 'http://'+IP;
end;
```

1.10 Ein Wort des Authors zum Thema „gewerbliche Nutzung“

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

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

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

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

Ich hoffe der Schritt von mir ist nachvollziehbar

2 Oberfläche

Die Bedienoberfläche des SDV besteht aus den folgenden Elementen: Zu fast allen Elementen bzw die wo es möglich ist, sind Hints programmiert

2.1 Hauptmenü (Mainmenü)



Host: Wechsel zwischen den verschiedenen CCUs, Möglichkeit zur Lizentianfrage und Donate

Selektiert: Markierte und Selektierte Objekte aus dem Inspektor lassen sich im Editor abrufen.

Syntaxcheck: Noch in Entwicklung

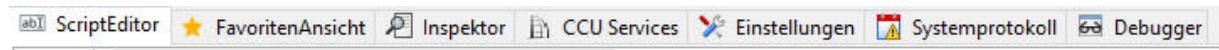
Methodenhilfe: Tief geschachteltes Menü mit typischen Hilfen für den Skriptprogrammieralltag

Backup: Menü mit den Möglichkeiten gezielte Backups vom System zu ziehen.

2.2 Toolbar

Die links seitliche Toolbar bietet Schnellzugriff auf die üblichen Editorfunktionen (undo, redo, Copy, paste, Suchen und Ersetzen etc. Die Buttons sind mit Hilfe-Hints ausgestattet

2.3 Arbeitsflächen (Pages)



SkriptEditor: s. Kapitel 3

Favoritenansicht s. Kapitel 5

Inspektor : s. Kapitel 4

CCU Services: Hilfe und Reparaturfunktionen sowie Gerätekopien

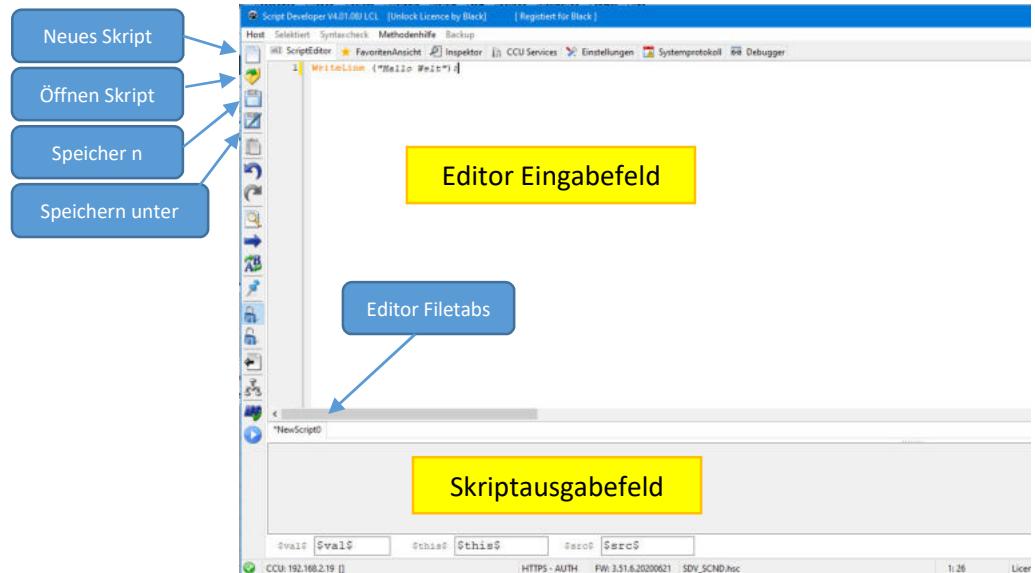
Einstellungen: Setup Möglichkeiten

Systemprotokoll: s. Kapitel 6 Ausgelesenes Systemprotokoll mit vielen Filtermöglichkeiten

Debugger: Testbereich für mich für die Entwicklung

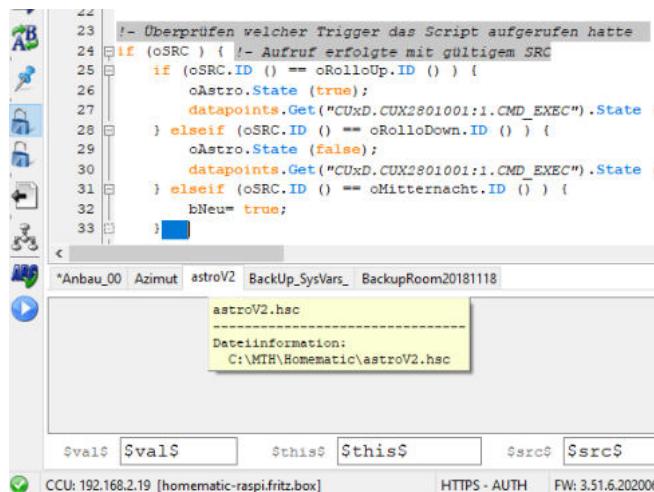
3 Skripteditor

Im Skripteditor werden die Skripte geschrieben oder geladen, die mittels Run Skript oder  an die CCU zum Ausführen mit mit  zum Testen an die CCU gesendet werden. Das Skriptergebnis wird dann unter Skriptausgabe dargestellt.

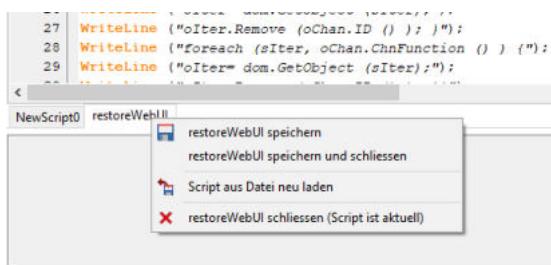


3.1 EditorTabs

Bei Mouse über einen Tab öffnet sich der Hint mit Informationen über die Datei.



Rechte Mousetaste auf einen EditorTab öffnet PopupMenü:



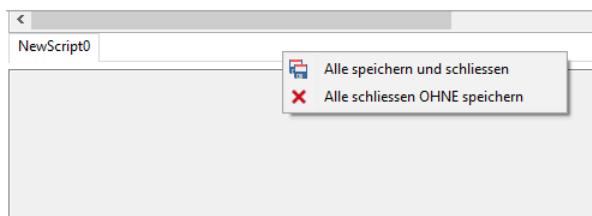
Speichern: schreibt den Inhalt des Editors bzw des Puffers in die Datei

Speichern und schliessen: schreibt den Inhalt des Editors bzw des Puffers in die Datei und schliesst den Tag

Script aus Datei neu laden: Wenn die Datei schon existiert, dann der Inhalt neu geladen werden und damit der Ext im Editor überschrieben werden.

Schliessen (mit Hinweis ist aktuell oder wurde geändert) schliesst den aktiven Editor Tab.

Klick mit rechte Mousetaste in freies Feld ausserhalb der Tabs öffnet Menü:



Alle Speichern und schliessen : alle geänderten Dateien werden gespeichert und die Tabs geschlossen

Alle Schliessen ohne Speichern: Alle Tabs werden ohne zu schliessen geschlossen. Es erfolgt noch eine Sicherheitsrückfrage:



1.1.1 Öffnen eines Scriptes

Es können bis zu 16 Scripte gleichzeitig geöffnet werden. Beim Versuch das 17. Script zu öffnen kommt eine Meldung, dass dieses nicht geht. Als Neues Script wird ein Script immer mit dem Namen NewScript mit fortlaufender Nummer angelegt.

Der Öffnen Mechanismus ist immer der folgende:

Existiert schon ein Tab mit dem Scriptnamen ?

Nein: Weniger als 16 Scripte offen ?

Ja : Fehlermeldung und Abbruch

Nein: Script öffnen

Ja: Steht das Script als Modified eingetragen ?

Ja: Rückfrage ob Script überschrieben werden soll ?

Nein: Script öffnen

3.1.2 Öffnen eines Scripts aus dem Inspektor (Extrakt aus der CCU)

Durch Klicken auf ** (Sternspecial) im Inspektor kann bekannter Weise direkt ein Script aus dem Inspektor exportiert werden und im Editor geöffnet werden.

Hier gilt folgender Mechanismus:

1. Der Scriptname ist Programmname_IDdesSingleDestinationObjektes
- 1.1 Es existiert schon ein Tab mit diesem Namen ?
2. Nein: Weniger als 16 Scripte offen ?
3. Ja : Fehlermeldung und Abbruch
4. Nein: Script öffnen
5. Ja: Steht das Script als Modified eingetragen ?
6. Ja: Rückfrage ob Script überschrieben werden soll ?
7. Nein: Script öffnen

In dem Tab wird die Herkunft des Scriptes vermerkt. Wird ein Tab geöffnet mit Kennung eines exportierten Scriptes, so wird der Rückladebutton farbig dargestellt und das Script bei Push auf den Rückladebutton wieder in das Singledestinationobject in der CCU wieder hochgeladen.

3.1.3 Öffnen eines Scriptes durch das Betriebssystem.

Ein oder auch mehrere Scripte können durch das Betriebssystem (Doppelklick auf eine Datei bei gesetzter Bedingung Öffnen mit) oder Drag and Drop abgelegt werden. Sollten dabei mehrere Prozesse durch das OS geöffnet werden, sprechen dies die SDV Instanzen über Inter-process-Communication ab, so dass nur die Hauptinstanz dann die Dateien öffnen wird und die weiteren Instanzen sich selber beenden.

Der Öffnen Mechanismus geht dann folgendermaßen:

1. Ist die Datei ein gültiges Script ?
2. Ja: Mechanismus gemäß 3.1.1

3.1.4 Öffnen der Scripte bei SDV-Start

Der SDV merkt sich die Namen der geöffneten Scripte beim SDV Programmende. Bei Start des SDV wird diese Liste wieder aus der INI geladen und nach folgendem Schema abgearbeitet:

1. Datei existiert ?
2. Ja: Inhalt laden, Datei öffnen
3. Nein: Die Datei existiert nicht, also nix machen

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 in the Script Developer interface. The output window displays the following text:

```

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

```

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

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

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

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

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

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

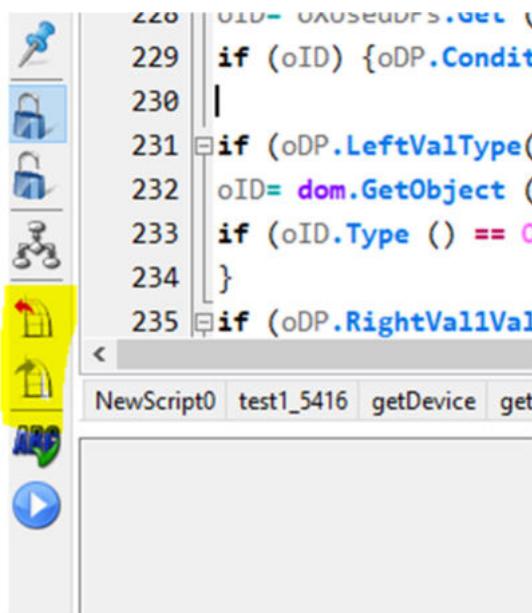
Bestandsnutzer können den folgenden Schlüssel in die SDV.INI schreiben:

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

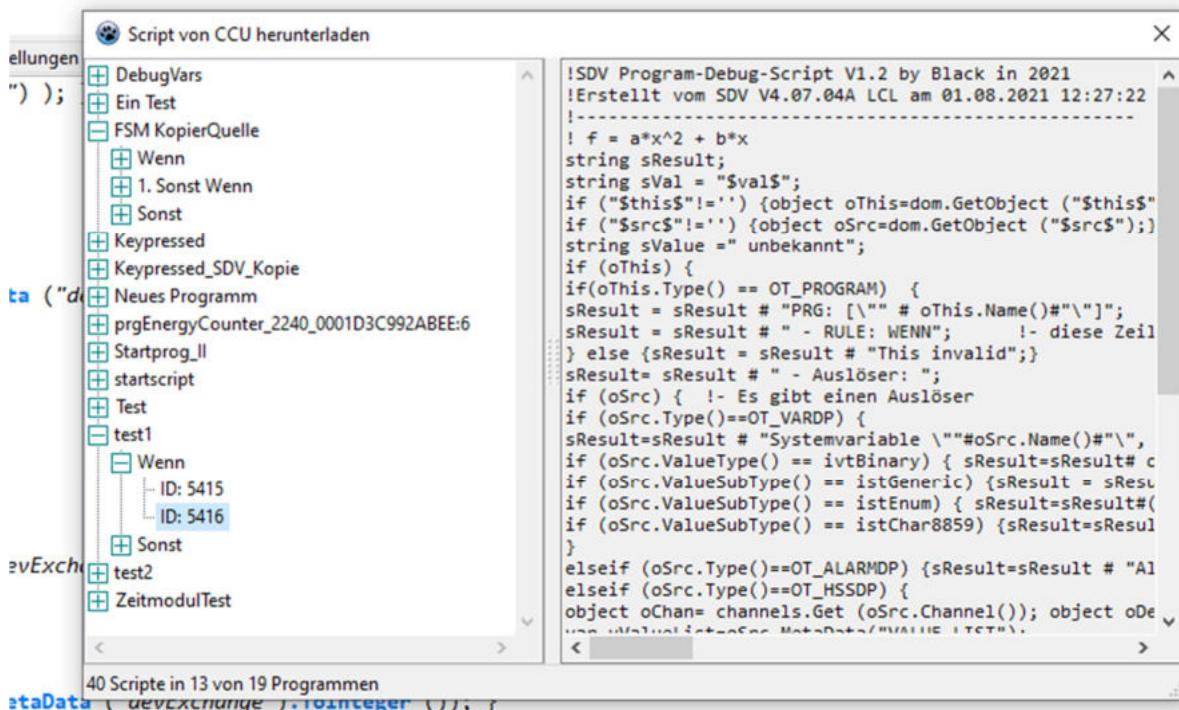
Die legt Script RUN auf F1 und Script testen auf F12. Kann selber angepasst werden. F3 ist reserviert !!! für suche erneut

3.1.5 Öffnen eines Skriptes aus der CCU und Zurückspielen

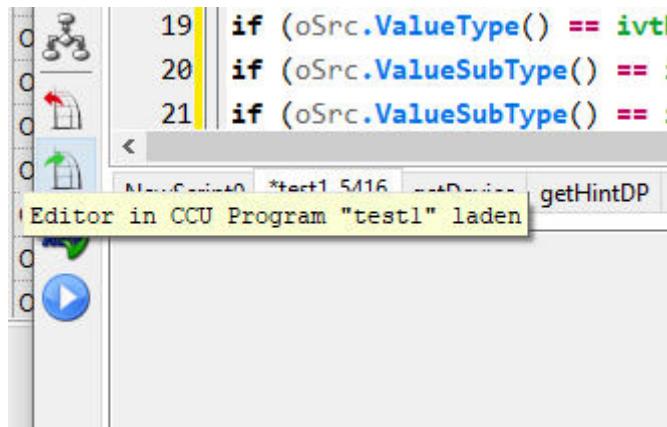
Mit der neuen Version ab 4.7.5 ist es komfortabler Möglich, direkt ein Script über die Menüleisten aus der CCU herunterzuladen bzw. auf diese zurückzuspielen



Roter Pfeil öffnet ein Auswahlmenü, mit dem sich unter allen Scripten aus der CCU eins auswählen lässt:



Links im Treeview sind die Programme und die darin enthaltenen Scripte inkl des Verweises, in welchem Teil des Programmes das entsprechende Script steht. Eine Schnellansicht des Scripte wird im rechten Teil des Fensters dargestellt. Doppelklick oder Enter übernimmt das Skript gemäß der unter 3.1.2 beschriebenen Bedingungen im Editor.



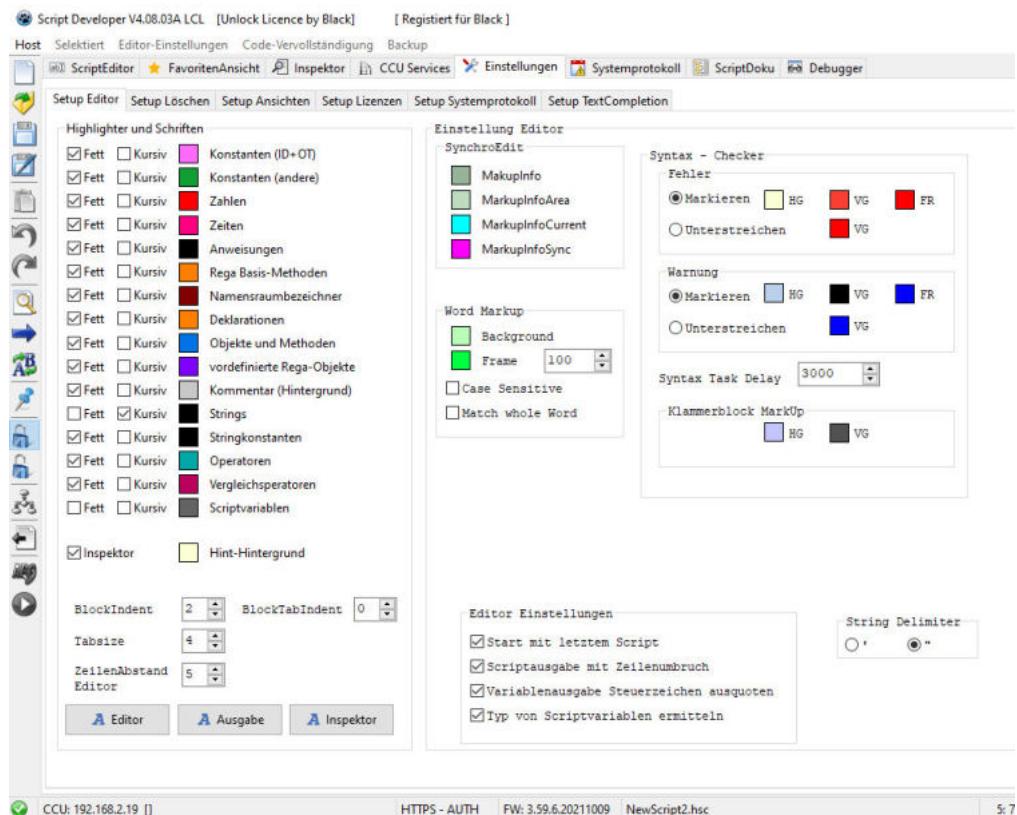
The screenshot shows a software interface for script development. On the left is a tree view with icons representing programs and scripts. The main area is a code editor with the following code:

```
19 if (oSrc.ValueType() == ivtI
20 if (oSrc.ValueSubType() == i
21 if (oSrc.ValueSubType() == i
```

The status bar at the bottom displays the message "Editor in CCU Program "test1" laden".

Mit der grünen Taste wird das Programm aus dem Editor dann wieder in die richtige Stelle in der CCU zurückgespielt.

3.2 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.

Editor Einstellungen:

Start mit letztem Script:

Enabled: Beim Starten des SDV werden die beim Schliessen gemerkten Tabs wieder versucht zu öffnen.

Disabled: Der SDV startet mit einem leeren Neuen NewScript0

Scriptausgabe mit Zeilenumbruch:

Enabled: bei der Darstellung der Scriptausgabe wird bei langen Zeilen umgebrochen

Disabled: lange zeilen werden auch nur als eine Zeile dargestellt mit Scrollbar

Variablenausgabe Steuerzeichen ausquoten

Enabled: XML Codierte Ausgabe des Variableninhaltes wird menschenlesbar wieder nochmal umgequotet

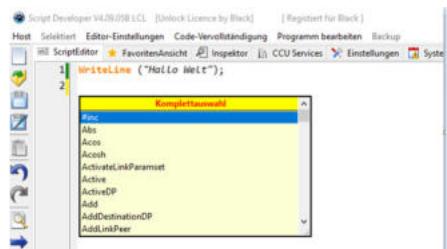
Disabled: Ausgabe bleibt in Originalzustand

Typ von Scriptvariablen ermitteln.

Hier wird vom SDV automatisch ZusatzCode an das zu testende Script angefügt und anschliessend das Ergebnis ausgewertet. So sind Aussagen über den Variablentyp möglich und ob sich der Typ der Variable geändert hatte und wenn ja, in welchen Typ.

3.3 Vervollständigen Funktion

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



Nach Druck von Enter erscheint das Wort im Editor.

Diese recht grobe Gesamtvorgabe gibt es noch etwas feinmodularer:

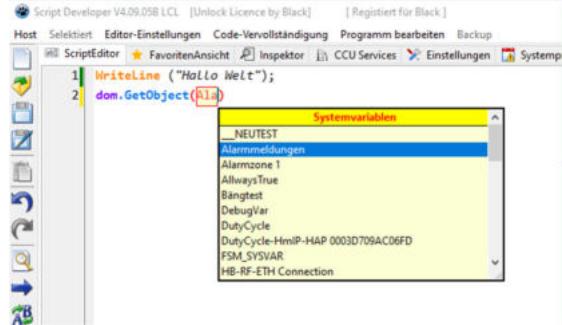
3.3.1 Verfeinerte Vervollständiger Funktionen

- CTRL SHIFT M : Alle Methoden
- CTRL SHIFT S : Alle String Methoden
- CTRL SHIFT C : Alle Mathematic Methoden
- CTRL SHIFT N : Alle ID_Konstanten
- CRTL SHIFT O : Alle OT_Konstanten

3.3.2 Spezielle Vervollständiger Funktionen

CTRL SHIFT D : Echtzeit Skriptanalyse: Alle im Skript deklarierten Variablen werden vorgeschlagen

CTRL ALT S : Echtzeitanalyse auf der CCU: Im Vervollständiger werden alle Systemvariablen angeboten



CTRL ALT G : Echtzeitanalyse auf der CCU: Im Vervollständiger werden alle Gewerke angeboten

CTRL ALT R : Echtzeitanalyse auf der CCU: Im Vervollständiger werden alle Raum angeboten

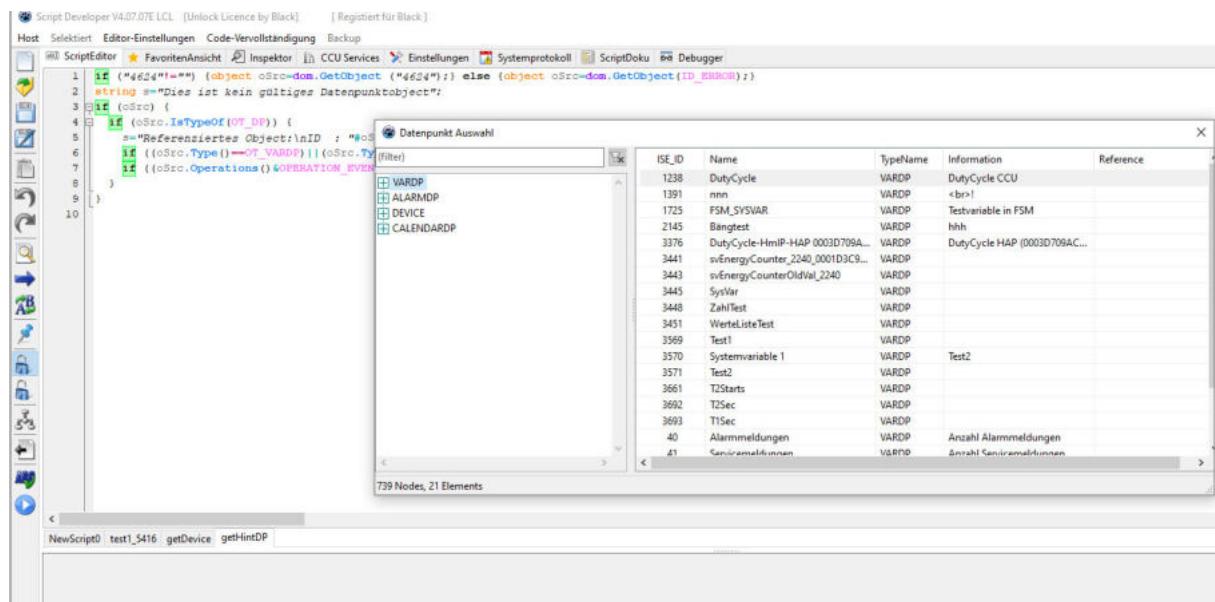
CTRL ALT D : Echtzeitanalyse auf der CCU: Im Vervollständiger werden alle Devices angeboten

CTRL ALT P : Echtzeitanalyse auf der CCU: Im Vervollständiger werden alle Programme angeboten

3.3.3 Qualifizierter Objektzugriff

Anwendung findet dieses erstmal an zwei Stellen im Editor. Erstmal über \$src\$ und Doppelclick öffnet sich der Auswahl Dialog:

An der Stelle sind zum Programmtriggern: Systemvariablen, Datenpunkte und Zeitmodule erlaubt. Grösse des Dialogs, der Anteil zwischen dem linken VirtualTreeview und dem rechten Listview sowie die Spaltenbreiten des Listviews lassen sich verändern und werden beim Programmende persistiert und beim Programmstart geladen.

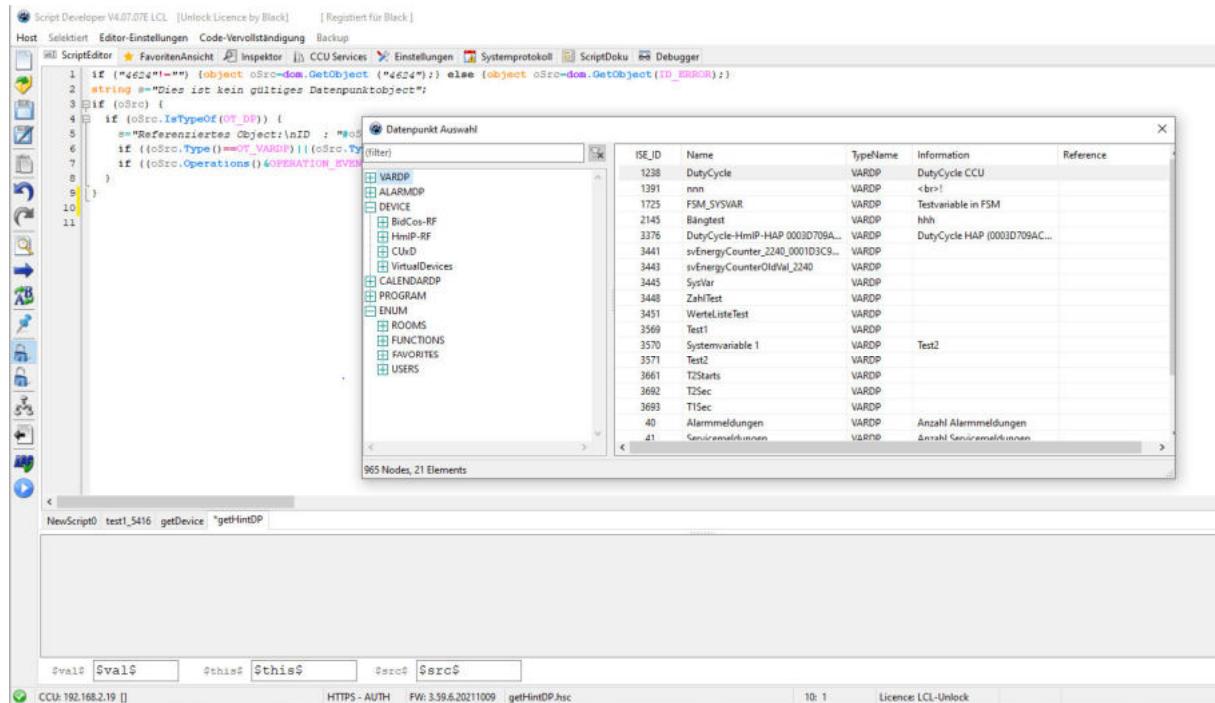


Über den Treeview wird die Grobauswahl gemacht (Endauswahl mit Doppelklick oder auch Enter geht auch) in der Listview die Endauswahl gemacht (Auch hier Doppelklick oder Enter). Zwischen dem Baum und dem Listenfenster lässt sich mit ctrl rechts bzw ctrl links öffnen. im Baum klappt Pfeil rechts einen Node auf, Pfeil links klappt den Node zu. Die Filtereingabe funktioniert so wie auch in der Skriptdoku, damit wir der gesamte Baum nach dem eingegebenen Suchbegriff gefiltert und die passenden Nodes aufgeklappt.

ISE_ID	Name	TypeName	Information	Reference
4816	BidCos-RF.LEQ1248855:0.UNREACH	HSSDP	UNREACH	TestDrehSensor:0
4812	BidCos-RF.LEQ1248855:0.STICKY_...	HSSDP	STICKY_UNREACH	TestDrehSensor:0
4802	BidCos-RF.LEQ1248855:0.CONFIG...	HSSDP	CONFIG_PENDING	TestDrehSensor:0
4806	BidCos-RF.LEQ1248855:0.LOWBAT	HSSDP	LOWBAT	TestDrehSensor:0
4810	BidCos-RF.LEQ1248855:0.RSSI_DE...	HSSDP	RSSI_DEVICE	TestDrehSensor:0
4811	BidCos-RF.LEQ1248855:0.RSSI_PEER	HSSDP	RSSI_PEER	TestDrehSensor:0

Die Funktion wurde nun auch voll in den Editor integriert. Es gab im SDV Editor noch keine vernünftige Auswahl direkt im Editor von Kanälen und Datenpunkten, die verwendete Completion Componente wäre für eine sinnige Baumstruktur darstellen schlicht ungeeignet. Der Umweg ging bisher über den Inspektor, auswählen des Punktes und dann mit Alt-O den vollqualifizierten Zugriff im Editor generieren lassen. Dies geht nun komfortabler:

Die alten Auswahlmethoden über die Completion wurden beibehalten, hinzu kam nun der Menupunkt "Qualifizierter Objektzugriff". Anwahl entweder über Menüpunkt oder über Ctrl-Alt-Space. Die gleiche Class wie eben auch, aber mit mehr Auswahlmöglichkeiten:



Hier lässt sich wieder in der Baumstruktur das Gewünschte Object grob (auch fein) auswählen, in der Listview dann Feinauswählen. Wie eben auch: Doppelklick oder Return. Je nachdem, welches Objekt man ausgewählt hat, wird daraufhin im Editor der Vollqualifizierte Zugriff eingetragen. z.B. Bei einem Enum Raum wird ein dom.GetObject (ID_ROOMS).Get("DerGewählte Raumname") erzeugt, für alle anderen entsprechenden Objekte auch der dementsprechende qualifizierte Zugriff. Interfaces, Devices und Channels wählt man in der Baumstruktur aus (Doppelklick oder Enter) Damit lassen sich nun komfortabel direkt im Editor Datenpunkte auf der CCU auswählen und sich auch automatisch der korrekte qualifizierte Zugriff im Editor generieren, ohne dass man über den Inspektor gehen muss.

Wenn das letzte Zeichen vor dem Cursor ein . ist und der Auswahldialog aufgerufen wird, dann werden nur die Devices eingeblendet werden und bei Auswahl nur noch der Part DPByHssDP übernommen. Grund wäre, wenn z.B. ein ChannelObject die Basis bildet und man schnell auf mehrere verschiedene Datenpunkte zugreifen möchte in der Art oChannel. <- dann der Aufruf, da braucht man dann nicht mehr den kompletten vollqualifizierten Zugriff, sondern nur noch den Part DPByHssDP ("SelektierterDatenpunkt")

3.3.4 AutoComplete

Unter Autocomplete lassen sich häufig verwendete Skriptsegmente ablegen und unter Stichworten abrufen.

Beispielhafter Aufbau:

```
SystemZeit_allgemein  
= system.Date(" | ");  
  
Beschreibung des Skriptsegmentes  
= beschreibt den dann einzufügenden Text  
Leerzeile trennt nächstes Skriptsegment
```

```
SystemZeit_Stunde_Minute  
= system.Date("%H:%M");
```

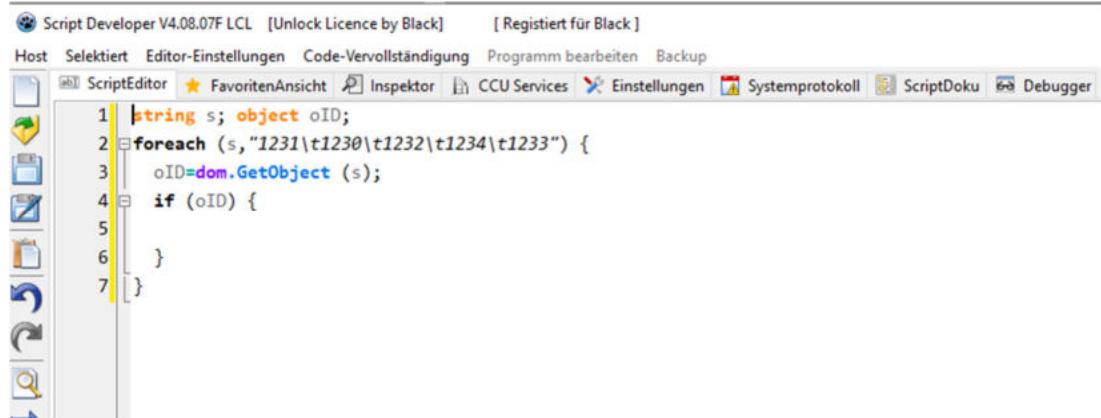
```
foreach  
=foreach (s,o)  
={  
=  
=}
```

Es gibt noch folgende Sonderform:

Der Placeholder \$\$\$ID\$\$\$ kann durch gespeicherte ID-Listen ersetzt werden kann. Konkretes Anwendungsgebiet: Bestimmte Objekte, die im Inspektor markierte wurden, sollen im Skripteditor iteriert werden um dort Aktionen durchzuführen

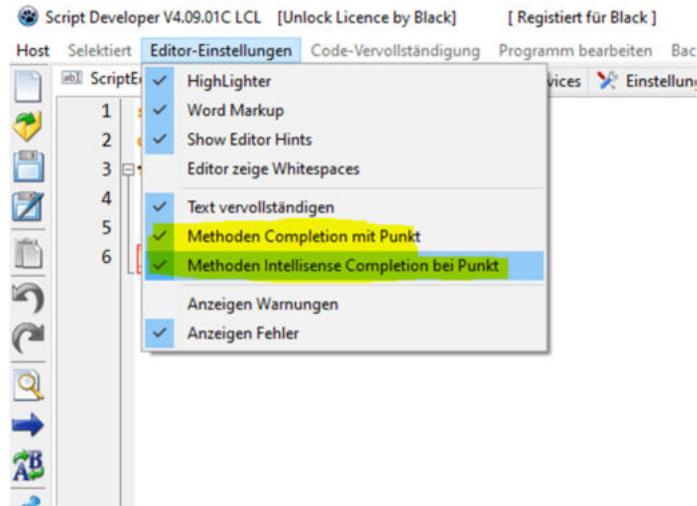
```
foreach_selected  
=string s; object oID;  
=foreach (s,$$$ID$$$) {  
= oID=dom.GetObject (s);  
= if (oID) {  
=XXX  
= }  
=}
```

Ergibt dann im Editor:



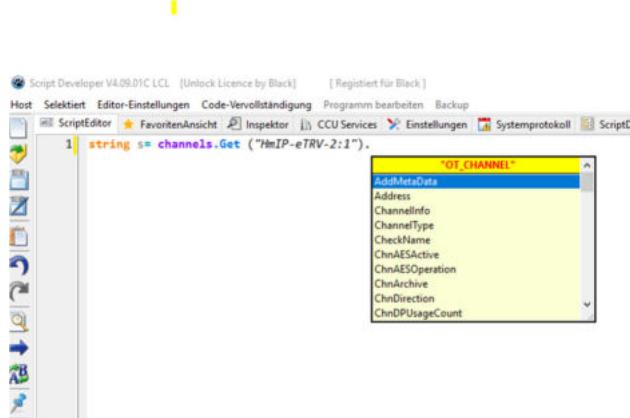
3.3.5 Vervollständigen über „Intellisense“

Der Editor ist ab der Version 4.09.xx mit einer verbesserten IntelliSense Funktionalität ausgestattet.
Zum Aktivieren müssen beide Haken gesetzt werden



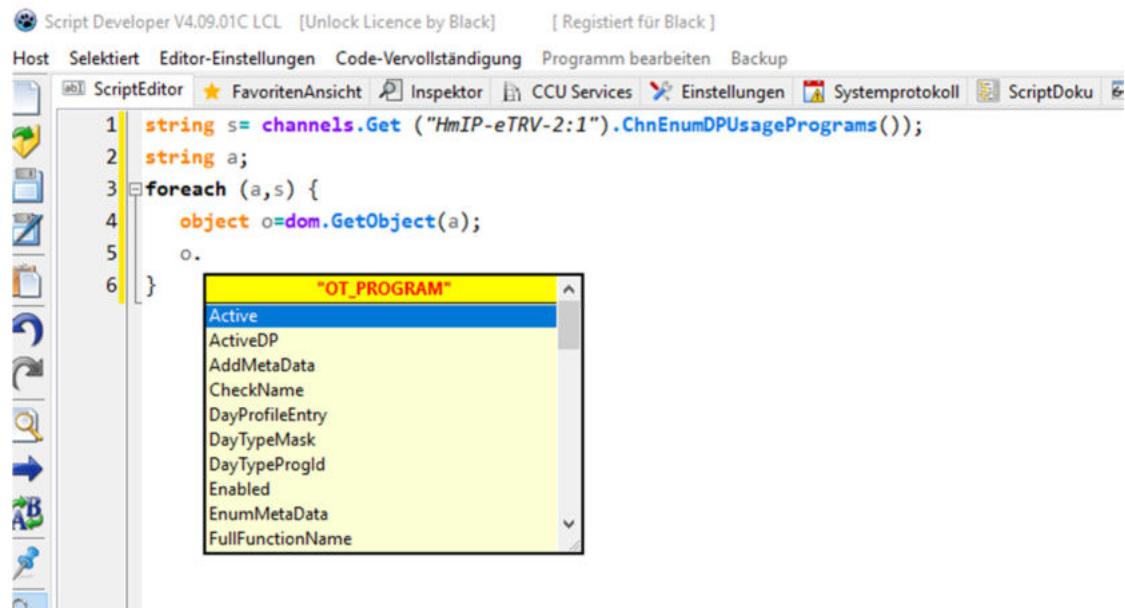
Wird nun beim Schreiben im Editor der . gedrückt und an dieser Stelle eine Methode möglich ist, so bestimmt der SDV nun mithilfe seines Echtzeit LL1 Parsers und unter Zuhilfenahme der RegaAbstraction Unit, welche Methoden an dieser Stelle wahrscheinlich möglich sind und schlägt diese nun vor.

An dieser Stelle hier sind die Methoden eines Objektes von Type OT_CHANNEL möglich



Es wird auch zur Info der Objekttyp angezeigt (oder Variablenotyp bzw. Namensraum).

Durch die Rega Abstraction Unit gehen auch Analytische Vorhersagen, wie in dem Beispiel hier wird auf ein OT_PROGRAM gerechnet



The screenshot shows the Script Developer interface. The menu bar includes "Host", "Selektiert", "Editor-Einstellungen", "Code-Vervollständigung", "Programm bearbeiten", and "Backup". The toolbar on the left contains icons for file operations like Open, Save, Print, and Find. The main window has a "ScriptEditor" tab selected. The code editor displays the following script:

```
1 string s = channels.Get ("HmIP-eTRV-2:1").ChnEnumDPUUsagePrograms();
2 string a;
3 foreach (a,s) {
4     object o=dom.GetObject(a);
5     o.
6 }
```

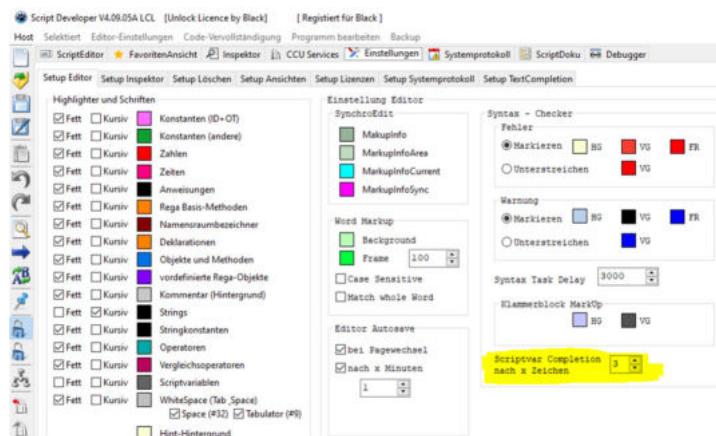
A code completion dropdown menu is open at the end of the line "o.". The title of the dropdown is "OT_PROGRAM". It lists the following properties and methods:

- Active
- ActiveDP
- AddMetaData
- CheckName
- DayProfileEntry
- DayTypeMask
- DayTypeProgId
- Enabled
- EnumMetaData
- FullFunctionName

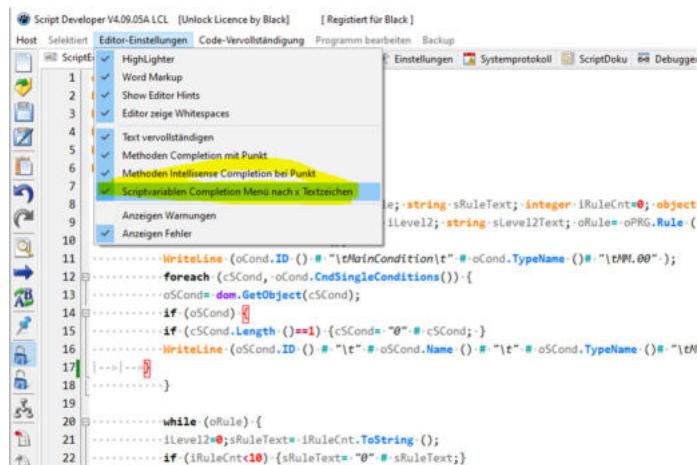
3.3.6 Autopopup Skriptvariablen

Der SDV Editor kann so eingestellt werden, dass während des Schreibens die Wahrscheinlichkeit der Verwendung einer Skriptvariablen erkannt wird und bei entsprechender Einstellung sich automatisch das Completion Fenster für die Skriptvariablen öffnet.

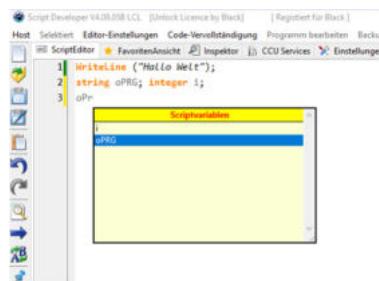
Die Prüfung findet nach einer einstellbaren Mindestwortlänge statt



In die diesem Beispiel müssen mindestens 3 Zeichen geschrieben worden sein.
Zusätzlich muss die Funktion generell eingeschaltet worden sein



Das Wort muss beginnen mit Zeichen für einen gültigen Variablennamen, es darf kein String oder kein Kommentar sein, die Mindestlänge muss erreicht sein, die Funktion muss angewählt sein und die getippte Buchstabenfolge muss schon dem Anfang mindestens einer vorhanden Skriptvariablen entsprechen, dann öffnet sich automatisch das Completion PopUp.



3.3.7 Komfortables Beenden und Weiterschreiben

Die Completion Menüs lassen sich alle auch mit „nicht Identifier keys“ beenden, heißt z.B. .,+-* / etc. Wenn die Auswahl zB. mit dem Punkt beendet wird, so wird z.B. die selektierte Skriptvariable übernommen und es öffnet sich Automatisch das Completion Menü, wo bei aktiviertem IntelliSense die zu der vorherigen Auswahl möglichen Methoden ausgewählt werden können.

Bei Methoden bewirkt das Beenden mit dem „.“ folgendes:

Es werden automatisch Klammern gesetzt

Ist ein leerer Parameter „()“ möglich, so wird der Punkt nach den Klammern gesetzt und automatisch wieder das IntelliSense Menü für die nächste Methodenauswahl aufgerufen

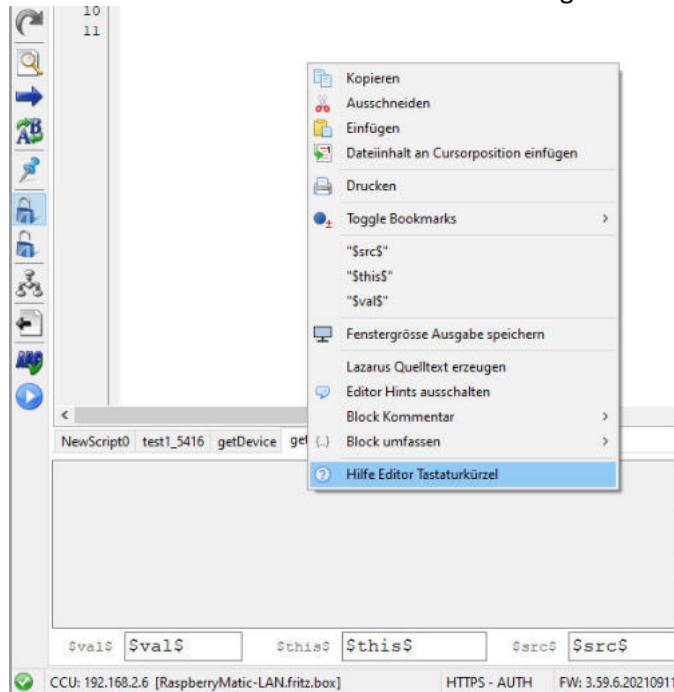
Ist kein leere Parameter möglich, so wird der Cursor innerhalb der Klammern positioniert

Auswahl mit Doppelklick oder Return

Hat die Methode nur leeren Parameter „()“ so wird der Cursor nach der „)“ positioniert, existiert mindestens ein Parameterwert, wird der Cursor in die Klammer gesetzt.

3.4 Editormenü (rechte Maustaste)

Rechte Maustaste auf das Editorfeld öffnet folgendes Menü:



Kopieren , Ausschneiden und Einfügen sind die Standardfunktionen unter CTRL C, CTRL X und CTRL V

3.4.1 Dateiinhalt an Cursorposition einfügen

Das Dateiauswahlmenü öffnet sich. Der Inhalt der ausgewählten Datei wird an der Cursorposition eingefügt. Dabei wird Der Undo Buffer berücksichtigt, nach dem Laden kann der Vorgang mit Undo rückgängig gemacht werden

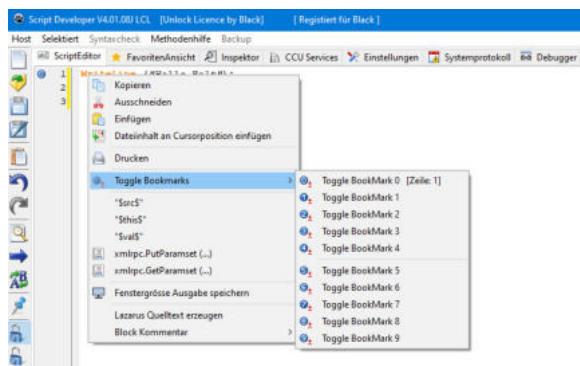
3.4.2 Drucken

Der Text des Editors samt Highlighting wird an den Standardbrowser geschickt zum Drucken.

3.4.3 Toggle Bookmarks

Pro Skript können bis zu 10 Bookmarks gesetzt werden (Möglichkeiten des Schnellsprunges)

Diese werden auch mit gesichert, wenn der SDV verlassen wird.



Natürlich sind die Bookmarks dynamisch, wenn die Zeilen verschoben werden.

3.4.4 \$src\$, \$this\$ und \$val\$

fügen den Text an der Cursorposition ein. Diese Felder können verschiedene Werte in den unterschiedlichen Skriptabs haben

3.4.5 Xmlrpc.PutParamset:

Wurde im Inspektor ein oder mehrere Master oder Linkparameter (bis zu 5 Stück) selektiert, so erzeugt dieser Menüpunkt automatisch den richtigen Code, um diese Parameter zu beschreiben. S.a. Kapitel 3.8

3.4.6 Xmlrpc.getParamset:

Wurde im Inspektor ein oder mehrere Master oder Linkparameter (bis zu 5 Stück) selektiert, so erzeugt dieser Menüpunkt automatisch den richtigen Code, um diese Parameter auszulesen. S.a. Kapitel 3.8

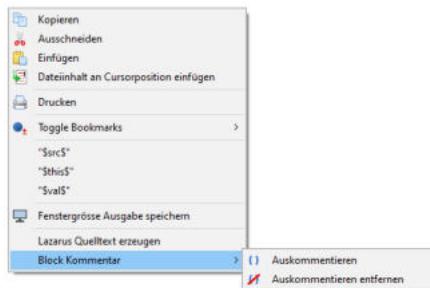
3.4.7 Fenstergrösse Ausgabe speichern:

Die Grösse des Ausgabefensters kann abgespeichert werden.

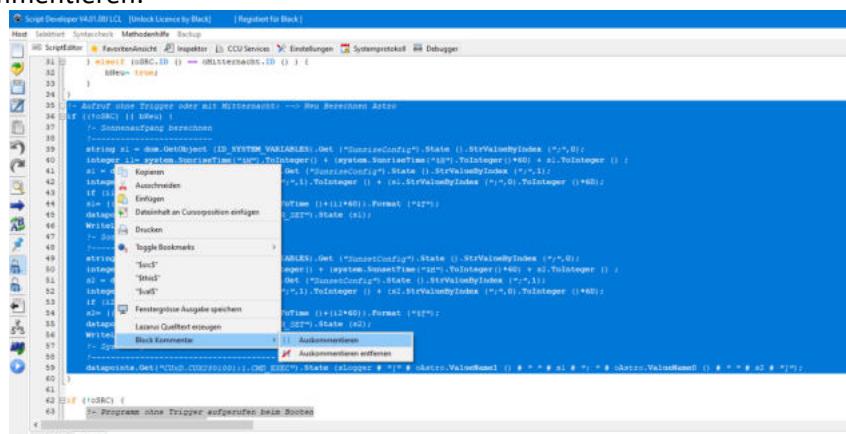
Rechte Maustaste in den Editor und Fenstergrösse Ausgabe speichern

3.4.8 Block Kommentar

Mit dieser Funktion können im Editor selektierte Zeilen aus bzw auskommentiert werden



Auskommentieren:



Wird zu:

```

34  }
35  /*- !- Aufruf ohne Trigger oder mit Mitternacht: --> Neu Berechnen Astro
36  /*- if ((!oSRC) || bNeu) {
37  /*- !- Sonnenaufgang berechnen
38  /*-
39  /*- string s1 = dom.GetObject (ID_SYSTEM_VARIABLES).Get ("SunriseConfig").State ().StrValueByIndex (";",0);
40  /*- integer i1= system.SunriseTime ("MN").ToInteger () + (system.SunriseTime ("H").ToInteger () *60) + s1.ToInteger ();
41  /*- s1 = dom.GetObject (ID_SYSTEM_VARIABLES).Get ("SunriseConfig").State ().StrValueByIndex (";",1);
42  /*- integer iZeitlimit=s1.StrValueByIndex (":",1).ToInteger () + (s1.StrValueByIndex (":",0).ToInteger () *60);
43  /*- if (i1<iZeitlimit) (i1= iZeitlimit);
44  /*- s1= ((system.Date ("TF") # " 00:00:00").ToTime ()+(i1*60)).Format ("iT");
45  /*- datapoints.Get ("CUxD.CUX2800001:10.TIMER_SET").State (s1);
46  /*- WriteLine (s1);
47  /*- !- Sonnenuntergang berechnen
48  /*-
49  /*- string s2 = dom.GetObject (ID_SYSTEM_VARIABLES).Get ("SunsetConfig").State ().StrValueByIndex (";",0);
50  /*- integer i2= system.SunsetTime ("MN").ToInteger () + (system.SunsetTime ("H").ToInteger () *60) + s2.ToInteger ();
51  /*- s2 = dom.GetObject (ID_SYSTEM_VARIABLES).Get ("SunsetConfig").State ().StrValueByIndex (";",1);
52  /*- integer iZeitlimit=s2.StrValueByIndex (":",1).ToInteger () + (s2.StrValueByIndex (":",0).ToInteger () *60);
53  /*- if (i2>iZeitlimit) (i2= iZeitlimit);
54  /*- s2= ((system.Date ("TF") # " 00:00:00").ToTime ()+(i2*60)).Format ("iT");
55  /*- datapoints.Get ("CUxD.CUX2800001:11.TIMER_SET").State (s2);
56  /*- WriteLine (s2);
57  /*- !- Systemlog Neuberechnet Astrozeiten
58  /*-
59  /*- datapoints.Get ("CUxD.CUX2801001:1.CMD_EXEC").State (oLogger # "[" # oAstro.ValueName1 () # " " # s1 # ":" # oAstro.ValueName0 () # " " # s2 # "]");
60  /*-
61  */

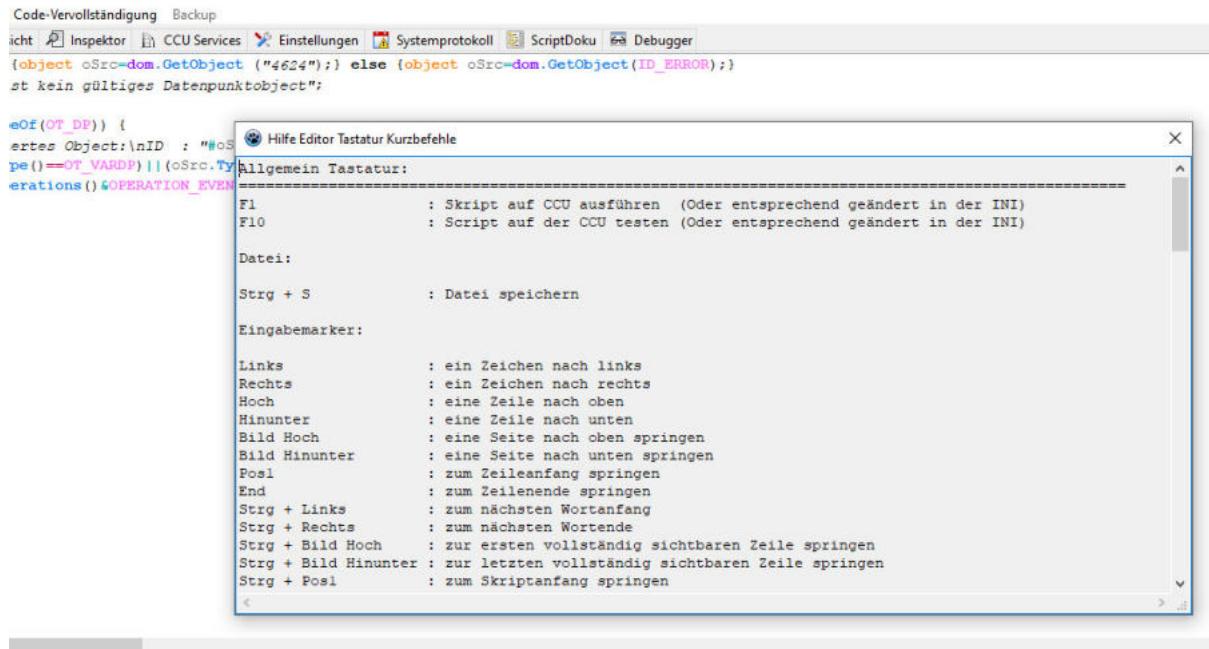
```

Eine nützliche Funktion, um beim Skripttesten mal eben schnell einen Block totzulegen ohne ihn löschen zu müssen oder mühsam von Hand auszukommentieren.

Auskommentieren entfernt entfernt von dem Selektierten Block die beginnende !- Kommentierung

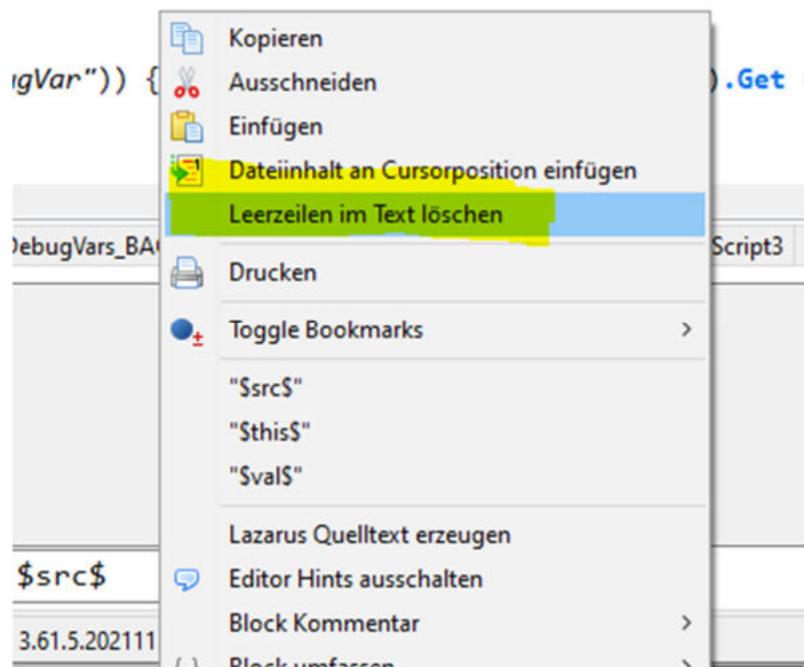
3.4.9 Editor Tastaturkürzel Hilfe

Kurzdarstellung der vom Editor verwendeten Tastaturkürzel



3.4.10 Überflüssige Leerzeilen entfernen

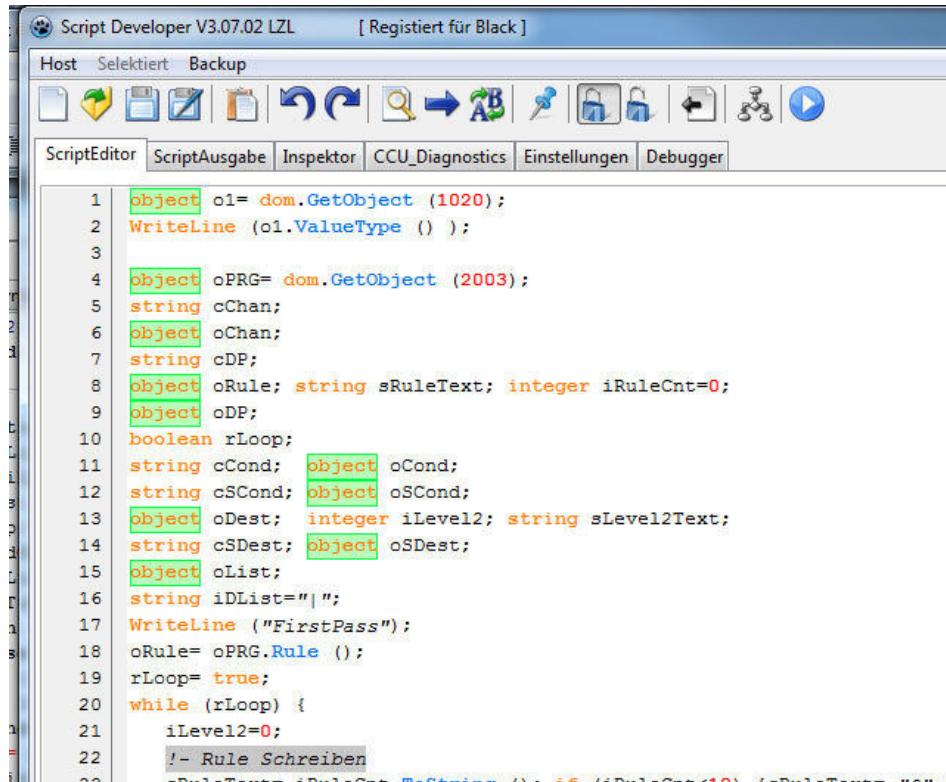
Sollte eine Editordatei aus irgendwelchen Gründen mal kaputt sein (sinnlos viele Zeilenvorschübe etc) kann dieses Menü helfen



Leerzeilen innerhalb Strings erkennt der SDV und löscht diese nicht

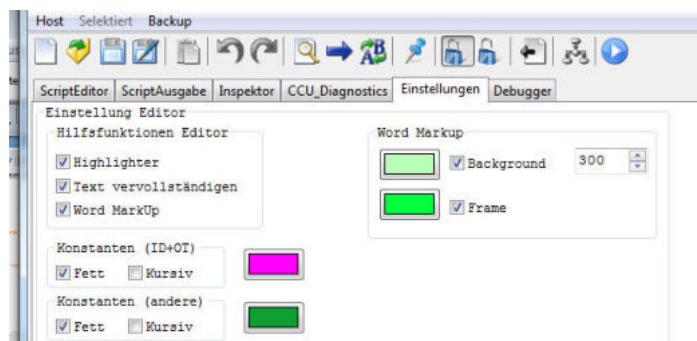
3.5 Word Markup (seit 03.07.02)

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



```
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
```

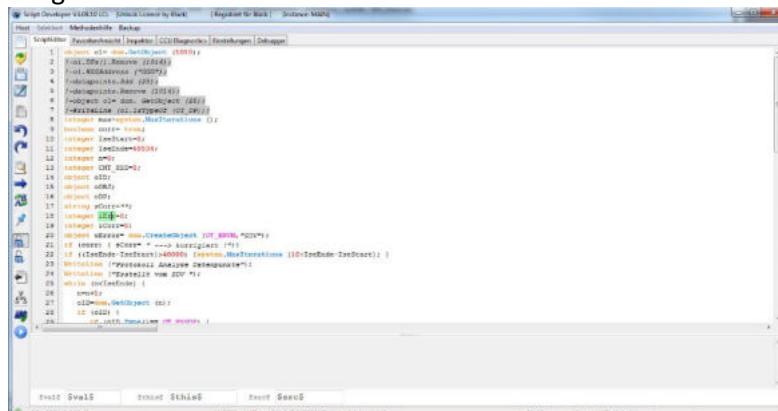
Die Funktion ist konfigurierbar unter Einstellungen



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

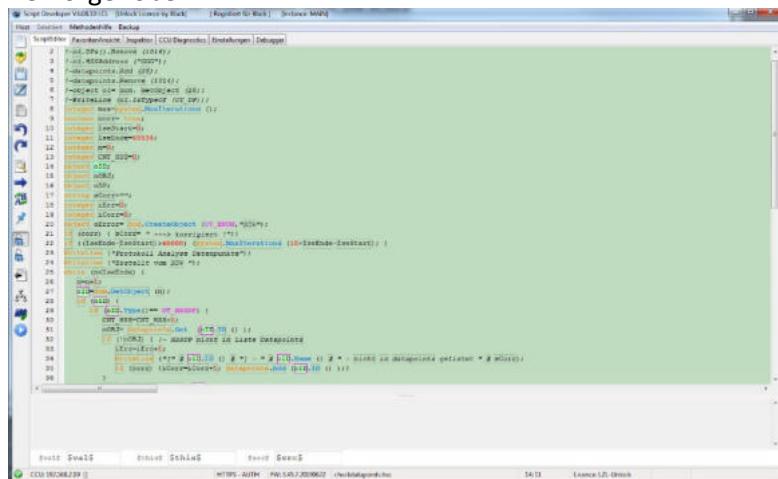
3.6 SyncroEdit (ab 3.08.10)

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



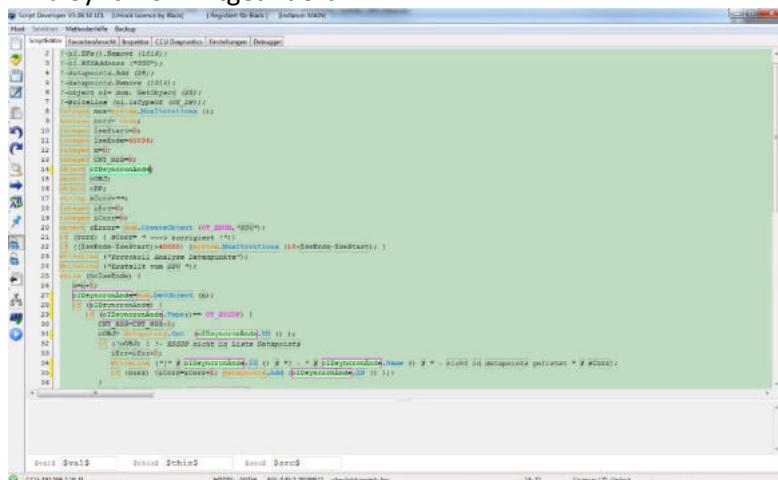
```
1 <!--> $!<!--> $!<!--> $!
2 <!--> $!<!--> $!<!--> $!
3 <!--> $!<!--> $!<!--> $!
4 <!--> $!<!--> $!<!--> $!
5 <!--> $!<!--> $!<!--> $!
6 <!--> $!<!--> $!<!--> $!
7 <!--> $!<!--> $!<!--> $!
8 <!--> $!<!--> $!<!--> $!
9 <!--> $!<!--> $!<!--> $!
10 <!--> $!<!--> $!<!--> $!
11 <!--> $!<!--> $!<!--> $!
12 <!--> $!<!--> $!<!--> $!
13 <!--> $!<!--> $!<!--> $!
14 <!--> $!<!--> $!<!--> $!
15 <!--> $!<!--> $!<!--> $!
16 <!--> $!<!--> $!<!--> $!
17 <!--> $!<!--> $!<!--> $!
18 <!--> $!<!--> $!<!--> $!
19 <!--> $!<!--> $!<!--> $!
20 <!--> $!<!--> $!<!--> $!
21 <!--> $!<!--> $!<!--> $!
22 <!--> $!<!--> $!<!--> $!
23 <!--> $!<!--> $!<!--> $!
```

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



```
2 <!--> $!<!--> $!<!--> $!
3 <!--> $!<!--> $!<!--> $!
4 <!--> $!<!--> $!<!--> $!
5 <!--> $!<!--> $!<!--> $!
6 <!--> $!<!--> $!<!--> $!
7 <!--> $!<!--> $!<!--> $!
8 <!--> $!<!--> $!<!--> $!
9 <!--> $!<!--> $!<!--> $!
10 <!--> $!<!--> $!<!--> $!
11 <!--> $!<!--> $!<!--> $!
12 <!--> $!<!--> $!<!--> $!
13 <!--> $!<!--> $!<!--> $!
14 <!--> $!<!--> $!<!--> $!
15 <!--> $!<!--> $!<!--> $!
16 <!--> $!<!--> $!<!--> $!
17 <!--> $!<!--> $!<!--> $!
18 <!--> $!<!--> $!<!--> $!
19 <!--> $!<!--> $!<!--> $!
20 <!--> $!<!--> $!<!--> $!
21 <!--> $!<!--> $!<!--> $!
22 <!--> $!<!--> $!<!--> $!
23 <!--> $!<!--> $!<!--> $!
```

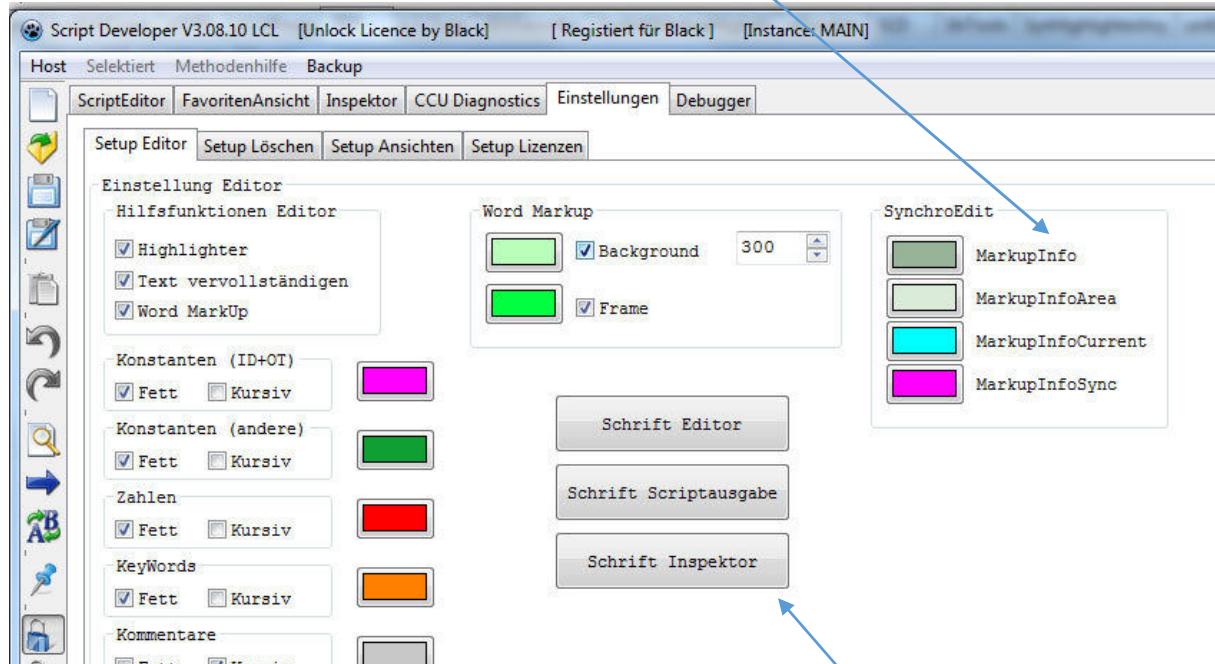
Jetzt kann man schreiben und an allen hellgrün hervorgehobenen Stellen im Mindgrünen Bereich wird Synchron mitgeändert.



```
2 <!--> $!<!--> $!<!--> $!
3 <!--> $!<!--> $!<!--> $!
4 <!--> $!<!--> $!<!--> $!
5 <!--> $!<!--> $!<!--> $!
6 <!--> $!<!--> $!<!--> $!
7 <!--> $!<!--> $!<!--> $!
8 <!--> $!<!--> $!<!--> $!
9 <!--> $!<!--> $!<!--> $!
10 <!--> $!<!--> $!<!--> $!
11 <!--> $!<!--> $!<!--> $!
12 <!--> $!<!--> $!<!--> $!
13 <!--> $!<!--> $!<!--> $!
14 <!--> $!<!--> $!<!--> $!
15 <!--> $!<!--> $!<!--> $!
16 <!--> $!<!--> $!<!--> $!
17 <!--> $!<!--> $!<!--> $!
18 <!--> $!<!--> $!<!--> $!
19 <!--> $!<!--> $!<!--> $!
20 <!--> $!<!--> $!<!--> $!
21 <!--> $!<!--> $!<!--> $!
22 <!--> $!<!--> $!<!--> $!
23 <!--> $!<!--> $!<!--> $!
```

Die Markierung aufheben geht dann mit Escape.

Die Farben lassen sich natürlich in Setup Menü einstellen



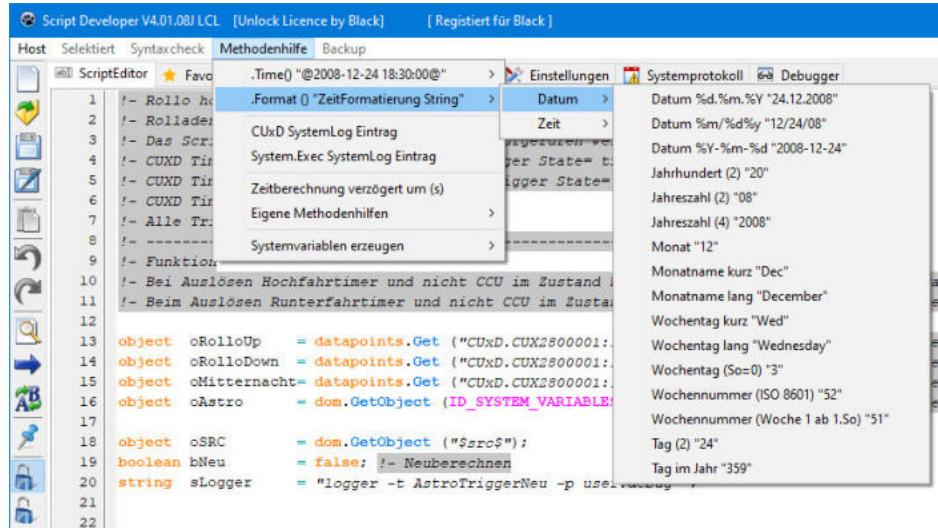
3.6.1 Schriftgrößen und Schriftart (ab 3.08.07)

Einstellungen für Schriftwart Editor, dem Editor Ausgabefenster und im Inspektor

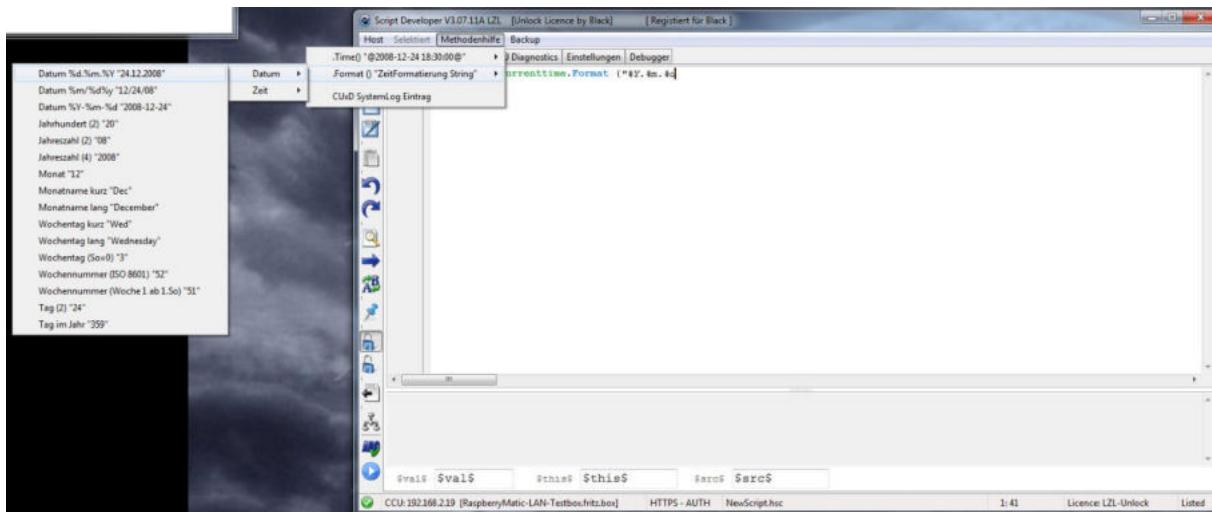
3.7 Methodenhilfe

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

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



Durch Druck wird die Passende Funktion ausgewählt. Dito bei Zeitformatierung



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

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

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


[METHODHELP2]
NAME=
TEXT=


[METHODHELP3]
NAME=
TEXT=


[METHODHELP4]
NAME=
TEXT=


[METHODHELP5]
NAME=
TEXT=


[METHODHELP6]
NAME=
TEXT=


[METHODHELP7]
NAME=
TEXT=


[METHODHELP8]
NAME=
TEXT=


[METHODHELP9]
NAME=
```

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

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

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

dabei bitte auf die Falle aufpassen !

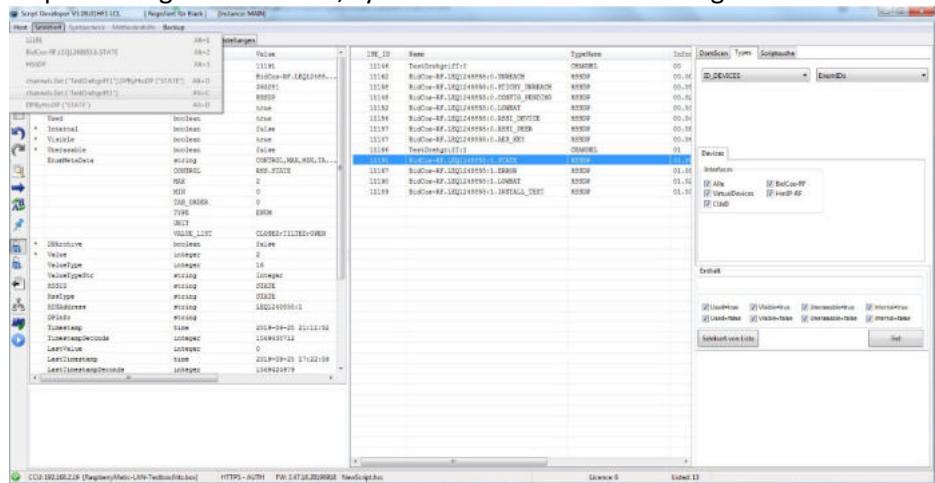
ich hab extra die Kommentare da mit reingesetzt, sie dort aufgeföhrt Sonderzeichen müssen gequotet sein rechts des Gleichheitszeichens.

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

Zweizeilig geht auch, dann muss \n benutz werden.

3.8 Symbolischer Zugriff auf selektierte Datenpunkte des Inspektors

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



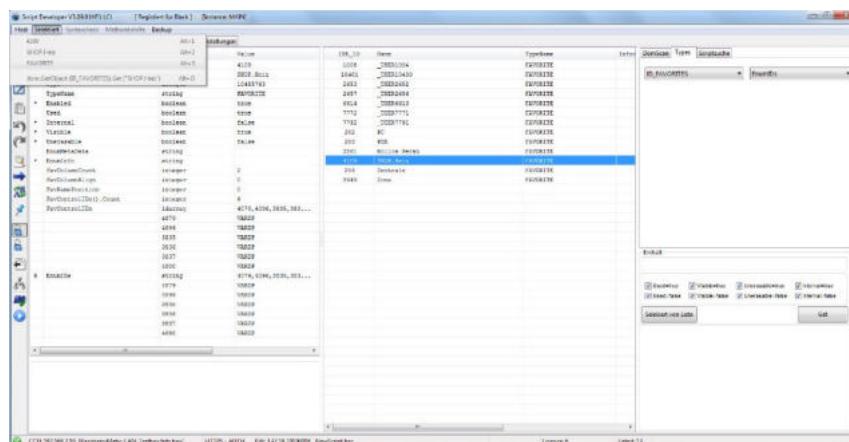
Bei einem HSSDP werden 3 Möglichkeiten angeboten (Menüpunkt selektiert)

Vollzugriff über Kanal.HSSDP

Nur Kanal

Nur HSSDP

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

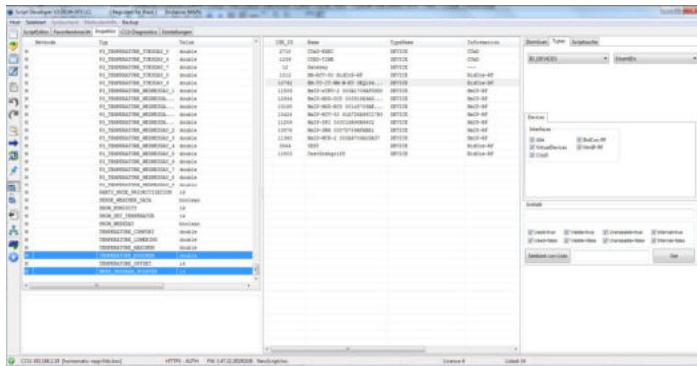


Eine einfache, schnelle und effektive Methode um unnötige Schreibfehler zu vermeiden.

3.9 Automatische Codeerzeugung für Read/Write von Master/Linksets

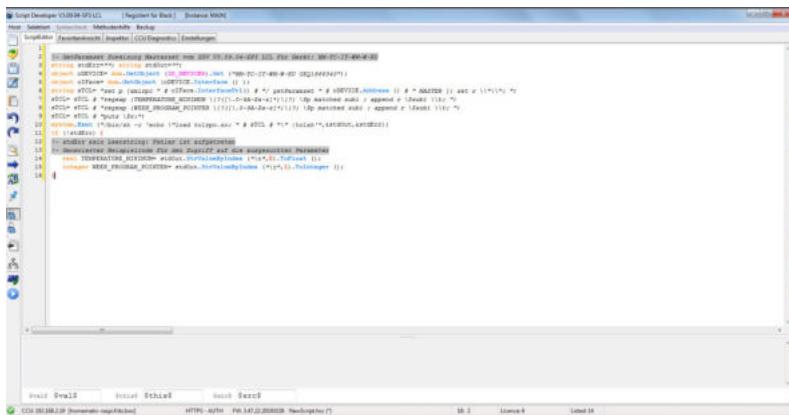
Im Editor kann automatisch Code erzeugt werden für das Schreiben von Master/Linksets (putParamset) oder für das Lesen von Master/Linksets (getParamset).

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



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

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



Natürlich funktioniert die automatische Codeerzeugung für Paramset Master, also die Geräteparameter als auch für die Daten von Direktverknüpfungen

3.10 Nützliche Tastenkombinationen des Editors

Alt – Pfeil rechts: es wird zu dem nächsten folgenden, gleichen Wort gesprungen, welches sich gerade unter dem Cursor befindet

Alt – Pfeil Links: es wird zu dem vorherigen gleichen Wort gesprungen, welches sich grade unter dem Cursor befindet

Allgemein Tastatur:

F1 : Skript auf CCU ausführen (Oder entsprechend geändert in der INI)
F10 : Skript auf der CCU testen (Oder entsprechend geändert in der INI)

Datei:

Strg + S : Datei speichern

Eingabemarker:

Links : ein Zeichen nach links
Rechts : ein Zeichen nach rechts
Hoch : eine Zeile nach oben
Runter : eine Zeile nach unten
Bild Hoch : eine Seite nach oben springen
Bild Runter : eine Seite nach unten springen
Pos1 : zum Zeilenanfang springen
End : zum Zeilenende springen
Strg + Links : zum nächsten Wortanfang
Strg + Rechts : zum nächsten Wortende
Strg + Bild Hoch : zur ersten vollständig sichtbaren Zeile springen
Strg + Bild Runter : zur letzten vollständig sichtbaren Zeile springen
Strg + Pos1 : zum Skriptanfang springen
Strg + End : zum Skriptende springen

Scollen: (Cursor verbleibt immer im sichtbaren Bereich)

Strg + Hoch : Ansicht nach oben scrollen
Strg + Hinunter : Ansicht nach unten scrollen

Bearbeiten:

Strg + A : alles auswählen
Strg + C : kopieren (Strg + Einfügen)
Strg + V : einfügen (Shift + Einfügen)
Strg + X : ausschneiden (Shift + Entfernen)
Strg + J : synchrones mehrzeiliges Editieren aktivieren (ESC zum Beenden des Modus)
Strg + Backspace : bis zum Wortanfang löschen
Enter : Zeile umbrechen
Strg + T : bis zum Wortende löschen
Strg + Y : Zeile löschen
Shift + Strg + Y : bis zum Zeilenende löschen
Strg + Z : Rückgängig (Undo)
Shift + Strg + Z : Rückgängig zurücknehmen (Redo)

```

Shift + Strg + I      : Block- / Einrücken
Shift + Strg + U      : Block- / Ausrücken

Suchen und Ersetzen
-----
Strg + F              : Suchen
Strg + R              : Ersetzen
F3                  : weitersuchen
Shift + F3            : entgegengesetzt weitersuchen
Alt + links           : zum vorherigen Vorkommen des Wortes unterhalb des Cursors springen
Alt + rechts          : zum nächsten Vorkommen des Wortes unterhalb des Cursors springen

Lesezeichen
-----
Strg + 0              : gehe zum Lesezeichen 0
Strg + 1              : gehe zum Lesezeichen 1
Strg + 2              : gehe zum Lesezeichen 2
Strg + 3              : gehe zum Lesezeichen 3
Strg + 4              : gehe zum Lesezeichen 4
Strg + 5              : gehe zum Lesezeichen 5
Strg + 6              : gehe zum Lesezeichen 6
Strg + 7              : gehe zum Lesezeichen 7
Strg + 8              : gehe zum Lesezeichen 8
Strg + 9              : gehe zum Lesezeichen 9

Spezielle Kombinationen des SDV
=====
Spezielle Anzeigefunktionen
-----
Strg + H              : Hint Funktionalität im Editor ein bzw ausschalten

Codevervollständigung
-----
Strg + Shift + S      : Completion Auswahl Methoden für Strings
Strl + Alt + I         : Completion Auswahl alle ID_xxx Konstanten
Strg + Alt + O         : Completion Auswahl alle OT_xxx Konstanten
Strg + Alt + M         : Completion Auswahl alle Methoden
Strg + Alt + V         : Completion Auswahl: verwendete Skriptvariablen
Strg + Alt + S         : Completion Auswahl: verwendete Systemvariablen
Strg + Alt + R         : Completion Auswahl: verwendete Räume
Strg + Alt + G         : Completion Auswahl: verwendete Gewerke
Strg + Alt + D         : Completion Auswahl: verwendete Devices
Strg + Alt + P         : Completion Auswahl: verwendete Programme
Strg + Alt + Space     : Auswahldialog für das Generieren des qualifizierten Zugriff
Strg + Shift + Space   : Autocompletion Auswahldialog
Strg + Shift + F1       : Autocompletion 1. frei definierter, einzufügender Textbaustein
Strg + Shift + F2       : Autocompletion 2. frei definierter, einzufügender Textbaustein
Strg + Shift + F3       : Autocompletion 3. frei definierter, einzufügender Textbaustein
Strg + Shift + F4       : Autocompletion 4. frei definierter, einzufügender Textbaustein
Strg + Shift + F5       : Autocompletion 5. frei definierter, einzufügender Textbaustein
Strg + Shift + F6       : Autocompletion 6. frei definierter, einzufügender Textbaustein
Strg + Shift + F7       : Autocompletion 7. frei definierter, einzufügender Textbaustein
Strg + Shift + F8       : Autocompletion 8. frei definierter, einzufügender Textbaustein
Strg + Shift + F9       : Autocompletion 9. frei definierter, einzufügender Textbaustein
Strg + Shift + F10      : Autocompletion 10. frei definierter, einzufügender Textbaustein

Einfügeoptionen aus dem Inspektor (bei angewähltem Objekt im Inspektor verfügbar)
-----
Alt + 0               : Einfügen der ID aus der Listenauswahl
Alt + 1               : Einfügen des Namens aus der Listenauswahl
Alt + 3               : Einfügen des Objekttypes aus der Listenauswahl
Alt + O               : Einfügen des qualifizierten Zugriffs des Objekttypes aus der
Listenauswahl
Alt + M               : Einfügen des Methodennamens aus der selektierten Zeile des Detailview
Alt + T               : Einfügen des Types aus der selektierten Zeile des Detailview
Alt + V               : Einfügen des Values aus der selektierten Zeile des Detailview
Alt + S               : Einfügen des Specials aus der selektierten Zeile des Detailview

```

=====

Mausbedienung

Links Einfachclick : Cursor positionieren
Links Doppelclick : Wort markieren
Links Dreifachklick : Zeile markieren
Links Vierfachklick : Block auswählen, der zwischen Leerzeilen eingeschlossen ist

SDV Maus Sonderfunktionen

Alt + rechte Maustaste : der zwischen { } liegende Block wird farblich markiert
Ctrl + Alt + rechte Maustaste: der zwischen () liegende Block wird farblich markiert

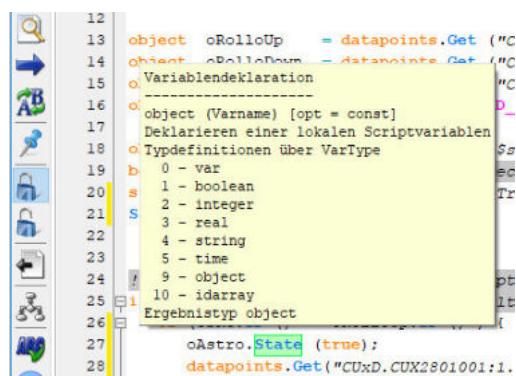
3.11 Editor Hint-Funktionalitäten

Durch die Hint Funktionalität werden Kontextbezogene Hilfestellungen zu dem Wort unterhalb der Maus dargestellt:



```
16 object oAstro = dom.GetObject (ID_SYSTEM_VARIABLES).Get ("Rollo.AstroTag"); // Master Variable
17
18 object oSRC = dom.GetObject ("Jsrc2");
19 boolean bNeu = false; // Neuberechnen
20 string sLogger = "logger -t AstroTriggerNeu -p user.debug ";
21 string strValuebyIndex;
22
23     .StrValuebyIndex ();
24 //-- Obergriff, string=(string).StrValuebyIndex(string separator, integer index)
25 if (oSRC) { // Der Parameter ist eine Liste, deren Elemente durch Separatoren voneinander getrennt sind.
26     // oSRC Die Methode "StrValuebyIndex" liefert das durch den Index spezifizierte Listenelement.
27     // obacht: Die Zahlung für "index" beginnt bei 0.
28     datapoints.Get("CUxD.CUX2801001:1.CMD_EXEC").State (sLogger + "[Master Trigger " + oAstro.ID + "]");
29 } elseif (oSRC.ID () == oRollUp.ID ()) {
30     oAstro.State (false);
31     datapoints.Get("CUxD.CUX2801001:1.CMD_EXEC").State (sLogger + "[Master Trigger " + oAstro.ID + "]");
32 } elseif (oSRC.ID () == oMitternacht.ID ()) {
33     bNeu= true;
```

Bzw



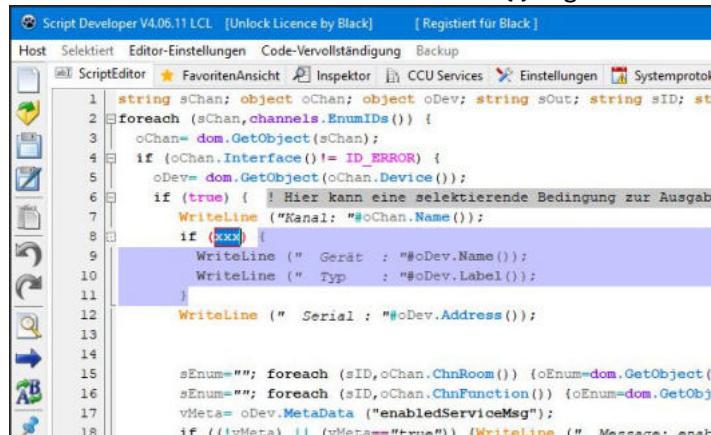
```
12
13 object oRollUp = datapoints.Get ("CUxD.CUX2801001:1.CMD_EXEC");
14 object oRollDown = datapoints.Get ("CUxD.CUX2801001:1.CMD_EXEC");
15 Variablendeklaration
16 o_
17     object (Varname) [opt = const]
18         Deklarieren einer lokalen Scriptvariablen
19         Typdefinitionen über VarType
20         b_ 0 - var
21         s_ 1 - boolean
22         i_ 2 - integer
23         r_ 3 - real
24         st_ 4 - string
25         t_ 5 - time
26         ob_ 6 - object
27         ar_ 7 - idarray
28         Ergebnistype object
29             oAstro.State (true);
30             datapoints.Get("CUxD.CUX2801001:1.
```

Hints werden kontinuierlich erweitert

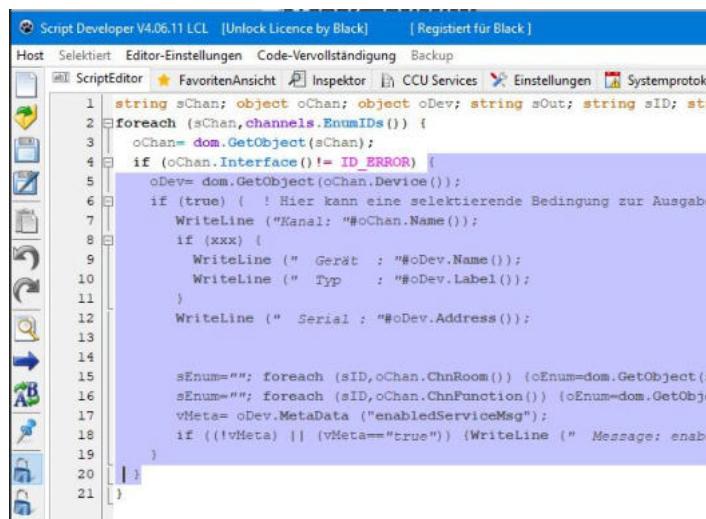
3.12 Blockdarstellung

Mit den folgenden Tasten/Mauskombinationen kann an der aktuellen Cursorposition der aktuell wirksame {} Block (mit alt+rechte Maustaste bzw der aktuelle wirksame runde Klammensblock eingefärbt werden.

Alt + rechte Maustaste : der zwischen { } liegende Block wird farblich markiert



```
1 string sChan; object oChan; object oDev; string sOut; string sID; st
2 foreach (sChan,channels.EnumIDs()) {
3     oChan= dom.GetObject(sChan);
4     if (oChan.Interface()!= ID_ERROR) {
5         oDev= dom.GetObject(oChan.Device());
6         if (true) ( ! Hier kann eine selektierende Bedingung zur Ausgab
7             WriteLine ("Kanal: "#oChan.Name());
8         if (xxx) {
9             WriteLine (" Gerät : "#oDev.Name());
10            WriteLine (" Typ : "#oDev.Label());
11        }
12        WriteLine (" Serial : "#oDev.Address());
13
14
15        sEnum=""; foreach (sID,oChan.ChnRoom()) (oEnum=dom.GetObject(
16        sEnum=""; foreach (sID,oChan.ChnFunction()) (oEnum=dom.GetObject
vMeta= oDev.MetaData ("enabledServiceMsg");
if (!vMeta) || (vMeta=="true")) (WriteLine (" Message: enab
```



```
1 string sChan; object oChan; object oDev; string sOut; string sID; st
2 foreach (sChan,channels.EnumIDs()) {
3     oChan= dom.GetObject(sChan);
4     if (oChan.Interface()!= ID_ERROR) {
5         oDev= dom.GetObject(oChan.Device());
6         if (true) ( ! Hier kann eine selektierende Bedingung zur Ausgab
7             WriteLine ("Kanal: "#oChan.Name());
8         if (xxx) {
9             WriteLine (" Gerät : "#oDev.Name());
10            WriteLine (" Typ : "#oDev.Label());
11        }
12        WriteLine (" Serial : "#oDev.Address());
13
14
15        sEnum=""; foreach (sID,oChan.ChnRoom()) (oEnum=dom.GetObject(
16        sEnum=""; foreach (sID,oChan.ChnFunction()) (oEnum=dom.GetObject
vMeta= oDev.MetaData ("enabledServiceMsg");
if (!vMeta) || (vMeta=="true")) (WriteLine (" Message: enab
```

Ctrl + Alt + rechte Maustaste: der zwischen () liegende Block wird farblich markiert

```
Enum.Name()#"|";} WriteLine (" Raum : "#sEnum.Trim("|"));
um#oEnum.Name()#"|";} WriteLine (" Gewerk : "#sEnum.Trim("|"));
iteLine (" Message: disabled\n"); }
```

Die Art der Einfärbung kann im Setup Menü Editor eingestellt werden.



3.13 Umfassen eines Textblocks

Praktisches Beispiel: gegeben sei dieses beispielhafte Programm:

A screenshot of the Script Developer V4.06.10 LCL interface. The title bar says 'Script Developer V4.06.10 LCL [Unlock Licence by Black]'. The main window shows a script editor with the following code:

```
1 string sChan; object oChan; object oDev; string sOut; string sID; string sEnum; o
2 foreach (sChan,channels.Enumerate()) {
3     oChan=dom.GetObject(sChan);
4     if (oChan.Interface() != ID_ERROR) {
5         oDev=dom.GetObject(oChan.Device());
6         if (true) { // Hier kann eine selektierende Bedingung zur Ausgabebeschränkung
7             Writeln ("Kanal: "#+oChan.Name());
8             Writeln (" Gerät : "#+oDev.Name());
9             Writeln (" Typ : "#+oDev.Label());
10            Writeln (" Serial : "#+oDev.Address());
11            sEnum=""; foreach (sID,oChan.ChrRoom()) {oEnum=dom.GetObject(sID); sEnum+=o
12            sEnum=""; foreach (sID,oChan.ChrFunction()) {oEnum=dom.GetObject(sID); sEn
13            vMeta=oDev.MetaData ("enabledServiceMsg");
14            if ((vMeta) || (vMeta=="true")) {Writeln (" Message: enabled\n"); } el
15        }
16    }
17 }
```

The code block from line 8 to line 10 is highlighted with a blue selection.

Die Blau markierten Blöcke möchte ich beispielsweise in eine neu zu erstellende if Bedingung einkapseln. das geht nun schnell und komfortabel mit der Umfassen Funktionalität. Dazu rechte Maustaste, Editorpopup Menü öffnet sich und unten auf Umfassen gehen und auswählen "Umfassen mit if"

The screenshot shows the Script Developer interface with a context menu open over a selected block of code. The menu path 'Block-umfassen' is highlighted, and a submenu is displayed with the following options:

- Umfassen mit if (...){} (selected)
- Umfassen mit else {}
- Umfassen mit elseif (...){} (disabled)
- Umfassen mit forsch (...){} (disabled)
- Umfassen mit while (...){} (disabled)

vor den Block wird nun eine if Zeile mit { eingefügt, der Block selber wird nach den eingestellten Werten recht eingerückt und am Ende des Blockes wird formatiert auch wieder eine } hinzugefügt. Der Cursor befindet sich dann direkt in der if Bedingung, so dass diese direkt angepasst werden kann.

The screenshot shows the modified script in the Script Developer editor. The original code has been wrapped in an if block:

```

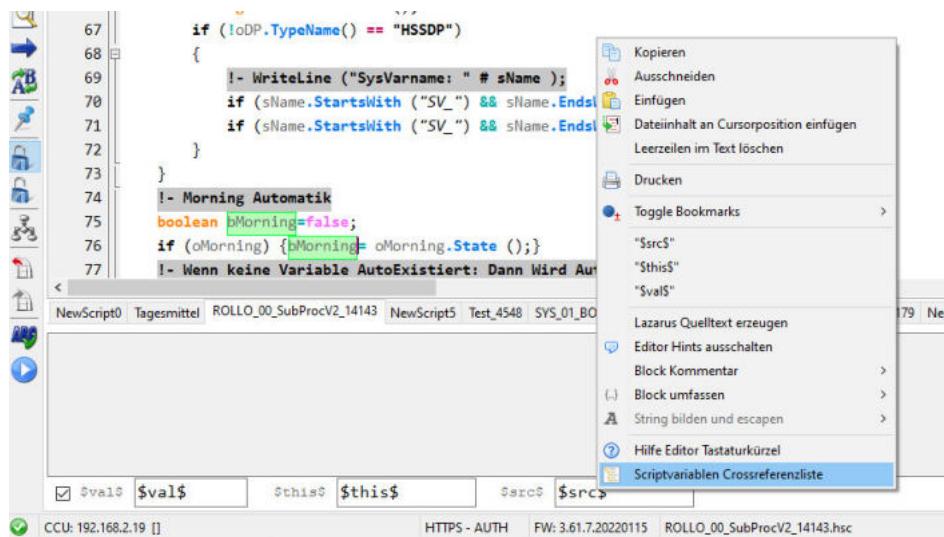
1 string schatz object <Chanc> object oDevz string abuz string aliz string schatz object <Chanc> var idobjz
2 foreach (oChanc,channele,channele) {
3   oChanc.dom.GetObject(schatz)
4   if (oChanc.Interface() != 10_IDBLOB) {
5     oChanc.dom.GetObject(oChanc.Devices())
6     if (true) { // Hier kann eine selektierende Bedingung zur Ausgabebeschränkung raus
7       Writeline ("* Gerät : "+oChanc.Name())
8       Writeline ("* Typ : ")
9       Writeline ("* Serial : ")
10      idObjz="" foreach (idz,oChanc,channele) {
11        idObjz+=oChanc.Name()
12      }
13      idObjz+="<br>" foreach (idz,oChanc,channele) {
14        idObjz+=oChanc.Name()
15      }
16      idObjz+=oDevz.Metadata("enabledServiceMsg")
17      if ((true)) || (connect=="true") {Writeline (* "Mezzager enabledin"+idObjz) } else {Writeline (* "Mezzager disabledin"+idObjz) }
18    }
19  }

```

Eine schnelle Methode, um nachträgliche Änderungen und Erweiterungen in einem bestehenden Skript hinzuzufügen.

3.14 Crossreferenzliste Skriptvariablen

Im Editormenü erzeugt der Menüpunkt „Skriptvariablen Crossreferenzliste“ die Verwendungsliste der Skriptvariablen.



Es wird dabei eine Liste angelegt von allen Stellen, wo Deklarierend oder Schreibend auf die entsprechende Variable zugegriffen wird.

Gelistet werden alle vorkommenden Stellen als

Deklaration [D]

Zuweisung [Z]

mit der Angabe der Textstelle. Click auf die Position markiert die Stelle im Editor

Angezeigt wird auch, als was der SDV den Variabtentyp an der Stelle kalkuliert.

Der SDV braucht für seine Methodenvorhersage schon ein paar mehr Informationen als nur die 9 Variablenarten, wenn ein Ergebnis ein bestimmtes Objekt beinhaltet, so wird dieses auch in den Variablen berücksichtigt, siehe in dem Beispiel.

Hier wird die Zuweisung als eine Variable von Typ Object erkannt, weitergehend aber ergibt sich durch die AST Auswertung und der Abstraction Unit, das hierbei dies nicht nur ein object ist, sondern dies noch feiner gradual betrachtet werden kann, es wird ein OT_CHANNEL sein.

The screenshot shows the Script Developer V4.09.01E LCL interface. The 'Scriptvariablen Referenzliste' (Variable Reference List) window is open, displaying a tree view of variable usages and declarations. A search filter is applied to show 'OT_CHANNEL' objects. The code editor on the left shows a script with several if statements and variable assignments. One assignment is highlighted with a yellow background: 'oChar = dom.GetObject(oSensor.Channel());'. The reference list shows this assignment under 'oChar' with a count of 690 occurrences.

Die Auswahl lässt sich filtern nach Variablenname (anwählbar ob "Name muss so Anfangen" oder "Suchbegriff irgendwo im Namen" und Variablenarten:

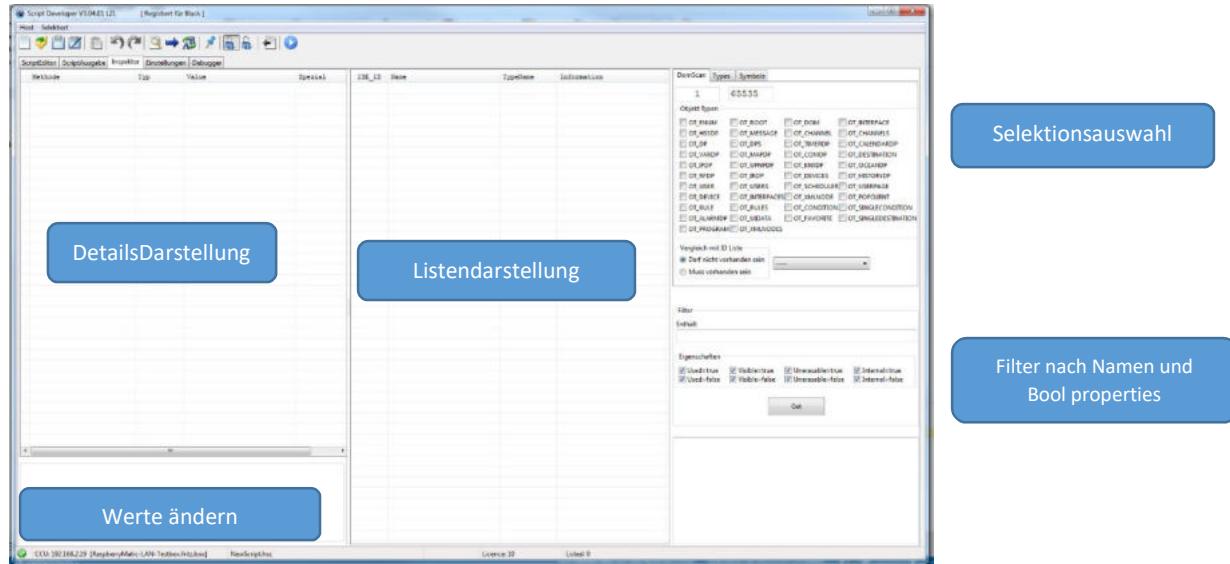
Eine Variable wird angezeigt, wenn eine !! ihrer Zuweisungen/Deklarationen in der dem Checkboxfeld vorkommt.

Die Größe des Fensters sollte persistiert werden beim Beenden.

4 Inspektor

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

Es existieren verschiedene Selektionskriterien.



Filteroptionen:

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

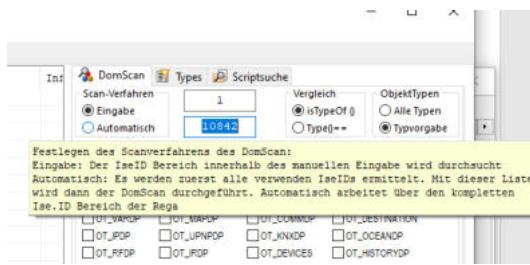
Enthält: leerer Eintrag = es wird nicht nach enthaltener Buchstabensequenz selektiert
Eingegebener Text. Die Systemvariable muss im Namen die Buchstabensequenz enthalten.

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

Am Beispiel used:

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

4.1 Selektionswahl: DomScan



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

Doppelklick auf das Eingabefeld der oberen Grenze ermittelt automatisch die höchste verwendete Ise-ID. Beschreibung der Strategien unter 4.1.1 bzw. unter Setup Inspektor

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

ObjectType Auswahl:

- Alle Typen: Es wird im Zahlenbereich gescannt ohne Objecttypen zu berücksichtigen
- Nach Objecttypen: Das Objekt muss dem oder den angewählten Typen entsprechen

Vergleich

- IsTypeOf () : Der Vergleich findet mit der Methode TypeOf statt. OT_Object findet dann z.B. ALLE Objecte, weil alle Objecte vom Type OT_OBJECTS abgeleitet sind
- Type() == : Das Object muss genau dieser Typ sein. Beispielsweise findet die Suche nach OT_OBJECTS auch nur genau die Objecte, die vom Type OT_OBJECTS sind.

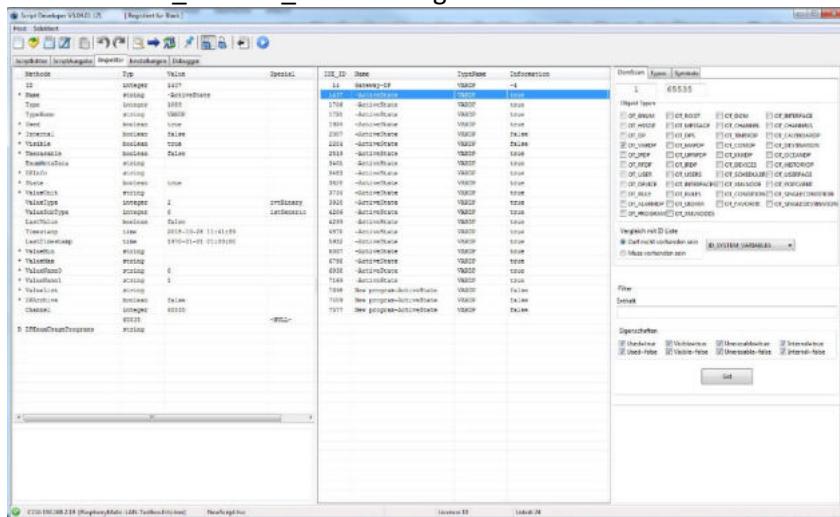
Beispiel für Suchen aller Objekte mit der Eigenschaft OT_DEVICE im Bereich der ISE_Nummern 1-65535

IsID	Name	Type	Description
1012	BR-BCT-10_3L4003-AZ	DEVICE	
1238	CORE-TIME	DEVICE	
3444	UL_ALLIANCE_USER	DEVICE	
3471	UL_CLOUD	DEVICE	
3489	Zeitreal	CALLERIDMAP	
3564	BB-CC-Netz-FM-1EGLX1110113	DEVICE	
6201	Back-Office 10010101000000000000	DEVICE	
6973	Zeitreal	CALLERIDMAP	
7418	Multi-XML_00100A0A0020000C	DEVICE	

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

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

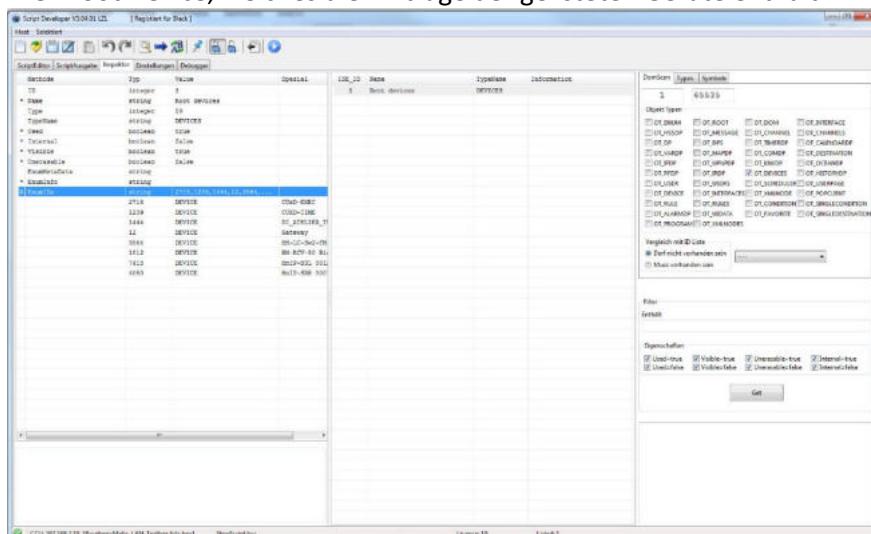
Hier Beispieldeweise: Scanlauf über alle Objekte aus DOM mit der Eigenschaft VARDP, die aber nicht unter ID_SYSTEM_VARIABLES gelistet sind:



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

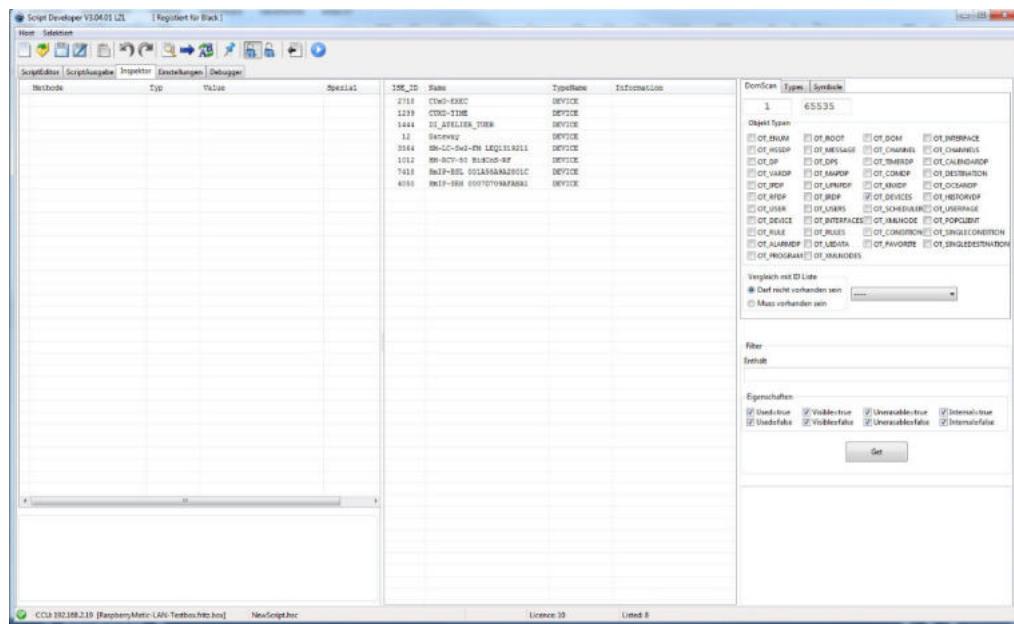
Weiterhin ist auch rekursives Arbeiten nun möglich

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.



4.1.1 Strategien für den Scan der Rega

Ermittlungsverfahren Iteration über die IselIDs

nötig geworden durch die Änderungen in der Strategie über die Vergabe neuer IselIDs
Die Einstellung hier mit Bedacht vornehmen, sie gelten SDV Weit.

RegaDom - Maxlse

Hier wird über eine Linux Befehlsfolge die Maximale IselID in der Datei Regadom gesucht. Nachteil: es muss persistiert werden, das dom.Save() dauert aber. Es wird in Testscheiben über alle Nummern iteriert, auch Lücken müssen untersucht werden

Regadom - Used Blocks

Hier wird durch eine Linux Befehlsfolge (Danke an Jerome für die Hilfe seinerzeit) die Blockbelegung der Rega analysiert. Nachteil: es muss persistiert werden, das dom.Save() dauert aber. Es wird dann nur über die gefundenen IDs iteriert.

Dom.CreateObject - Maxlse

Achtung- Nur mit aktuellen Regas zu verwenden, wo der Allready in MAP Error durch die neue Strategie gefixt wurde. Es wird dann in Testscheiben über alle Nummern iteriert, auch Lücken müssen untersucht werden.

Empfehlung:

=====

Neue Rega: CreateObject

Alte Rega: Auf keinen Fall CreateObject

Neue Rega und riesig aufgebläht so dass die Analyse der Lücken zeit kostet:

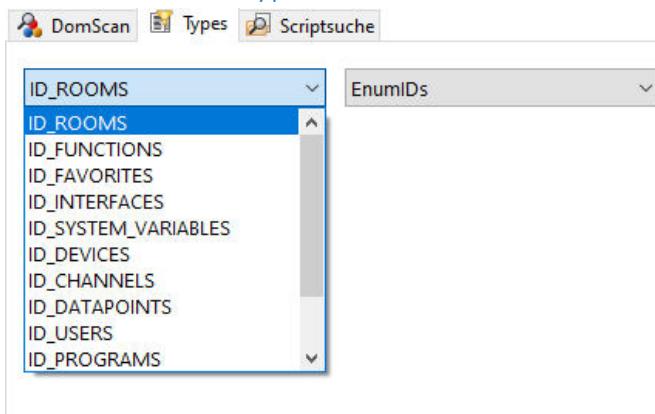
Used Blocks

Alte Rega oder CCU2:

Used Blocks

RegaDom Maxlse ist aus Kompatibilitätsgründen noch drin, wurde früher vom SDV verwendet.

4.2 Selektionskriterium Types

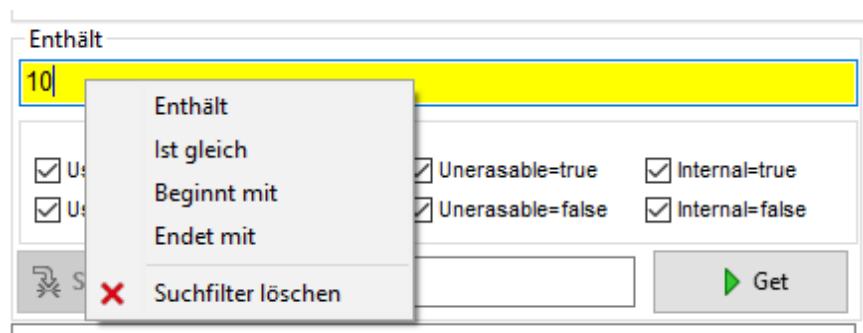
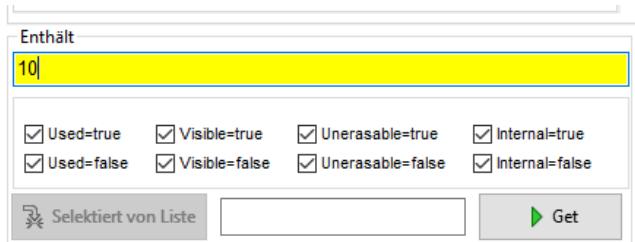


Hierbei wird wie schon in der Version 3.2 in festen Bereichen gesucht und selektiert. Schneller und einfacher zu handeln als die Objekt Selektion, dafür nicht so umfangreich.

4.3 Zusätzliche Selektionsbedingungen

4.3.1 Enthält Filter

Über den Enthält-Filter lässt sich bei der Auswahl über die Property NAME der gefundenen Objekte selektieren. Wenn ein Enthält Kriterium Selektiert ist wird diese Zeile gelb dargestellt, damit dieses auch sichtbar ins Auge fällt.



Enthält: Die Zeichenfolge muss irgendwo in NAME enthalten sein

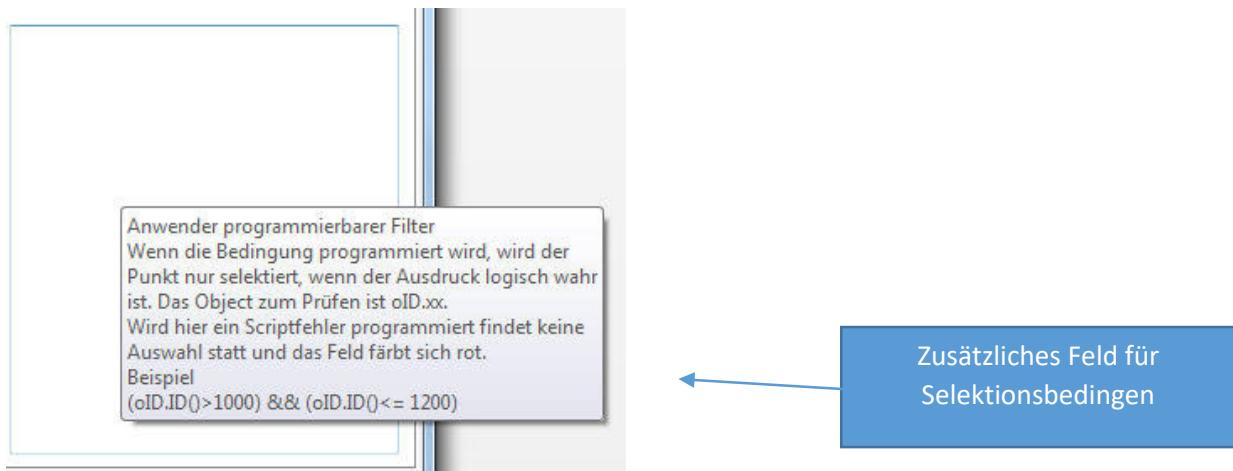
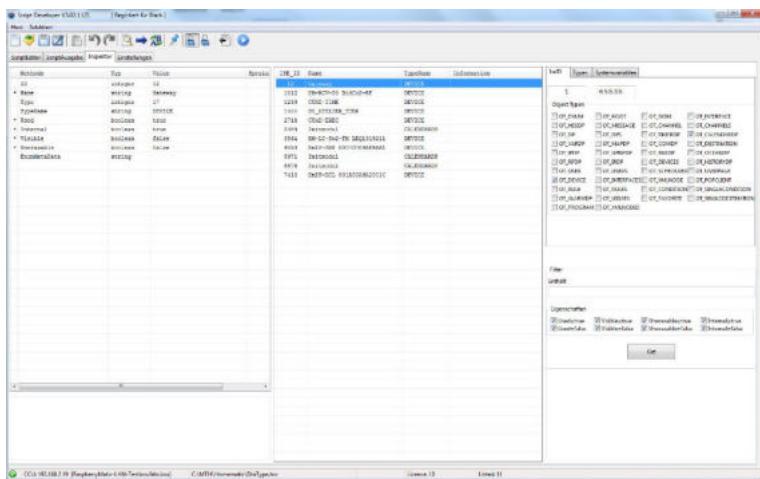
Ist gleich: NAME muss der Zeichenfolge entsprechen

Beginnt mit: NAME muss mit der Zeichenfolge beginnen

Endet mit: NAME muss mit der Zeichenfolge enden

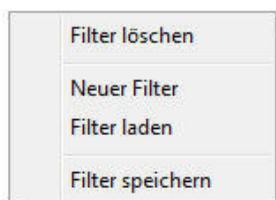
Durch Druck auf Get wird die Liste gemäß Selektion von der CCU angefordert, aufbereitet und dargestellt. (Lizenzlevel vorausgesetzt)

4.3.2 Anwenderdefinierte Filter



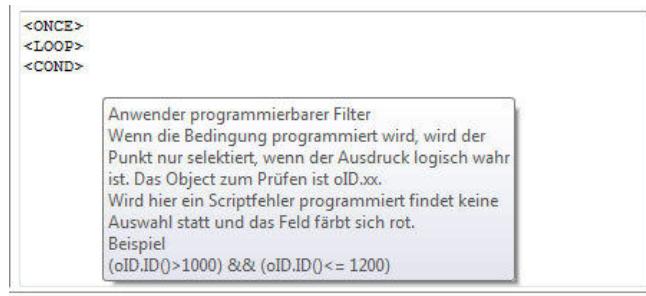
Filter sind ein mächtiges Werkzeug zum komplexen Eingrenzen und für Komplexe Abfragen.

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 Abfrageskriptes quasi im einmaligen Durchlauf eingefügt. Normalerweise stehen hier Definitionen, welche nicht bei jedem Durchlauf aktualisiert werden müssen

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

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

Vereinfachter Ablauf: so sieht vereinfacht das Listenselektionsprogramm aus:

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

Ein Anwenderdefinierter Filter wird dann in diese Grundschleife so eingebaut:

```
object oID;
string s;
ONCETEXT;

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

An diesem Kleinen Filter mal verdeutlicht:

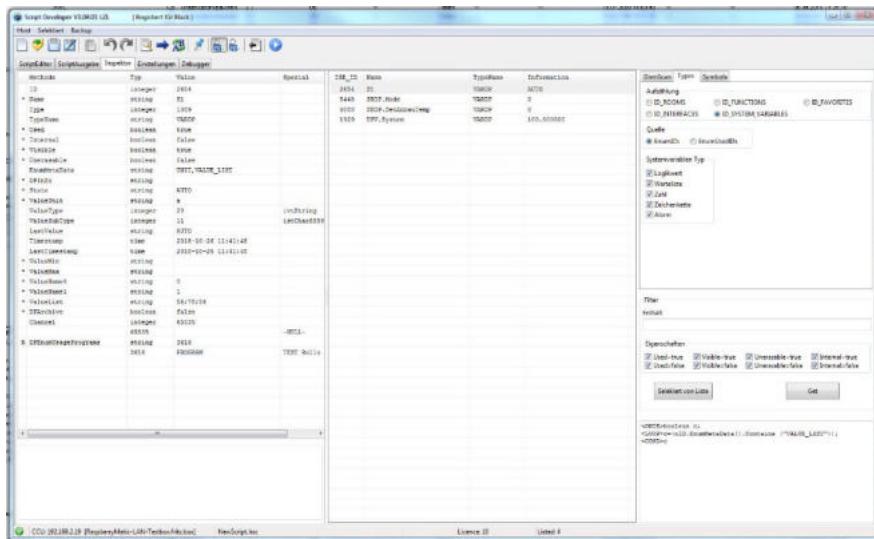
```
<ONCE>boolean c;
<LOOP>c=(oID.EnumMetaData().Contains ("VALUE_LIST"));
<COND>c
```

Daraus generiert der Scriptdeveloper folgende interne Filterabfrage:

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

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

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



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

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

var v: intern benutzt zur Typerkennung: Fingers weg

string sInfo: intern benutzt zur Listengenerierung: Fingers weg

boolean b: interner Filter, auch Finger weg

string done: auch interne Benutzung, auch Finger weg

Die Filterbedingung wird in HM Skript 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 IselD 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. IselD sortiert nun wie man erwartet nach Integer aufsteigend, Name sortiert alphabetisch aufsteigend, TypeName sortiert alphabetisch, sind die Typenames gleich, wird innerhalb gleicher Typenames nach IselD numerisch sortiert. (Ab Version 3.08.08 verbesserter Sortieralgorithmus, hier wird durch Klick auf die Spalte zwischen Sortieren aufsteigend und Sortieren absteigend getoggelt.)

Methode	Type	Value	Syntax	IselD	Name	TypeName	Information
ID	integer	7404			room@bedroom	ERIN	Info Badzimmer
* Name	string	TestRaum			root@bedroom	ERIN	BB
Type	integer	3			root@children@root	ERIN	
TypeName	string	ERIN			root@children@root	ERIN	
* Used	boolean	TRUE			root@bedroom	ERIN	
* Internal	boolean	FALSE			root@children@root	ERIN	
* Visible	boolean	TRUE			root@children@root	ERIN	
* Unavailable	boolean	FALSE			root@children@root	ERIN	
EnumableData	string				root@children@root	ERIN	
* EnumIDs	string	Ich bin die 1. Bezeichn...			root@children@root	ERIN	
EnumIDe	string	1445,2019			root@children@root	ERIN	
1445	CHANNEL	SI_AFE	value TestRaum		root@children@root	ERIN	Ich bin die 1. Bezeichnung
2019	CHANNEL	SI-UC-<					

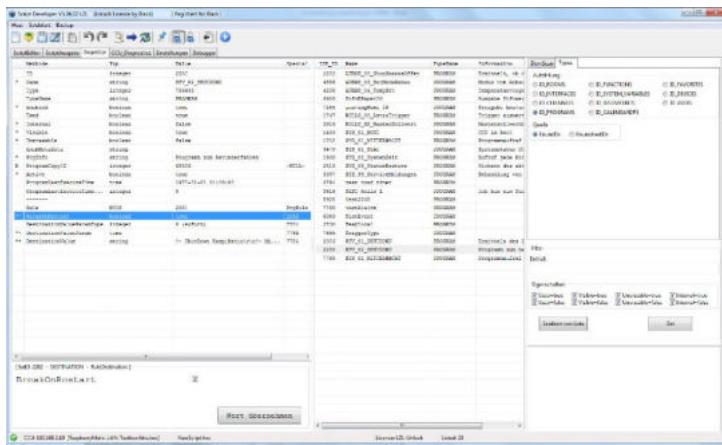
Die Spaltenbreite kann sowohl in Normalansicht als auch in Maximize separat eingestellt werden (Das Programm sollte sich die Breiten merken und je nach Darstellungsart automatisch wieder einstellen, sollte...)

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)

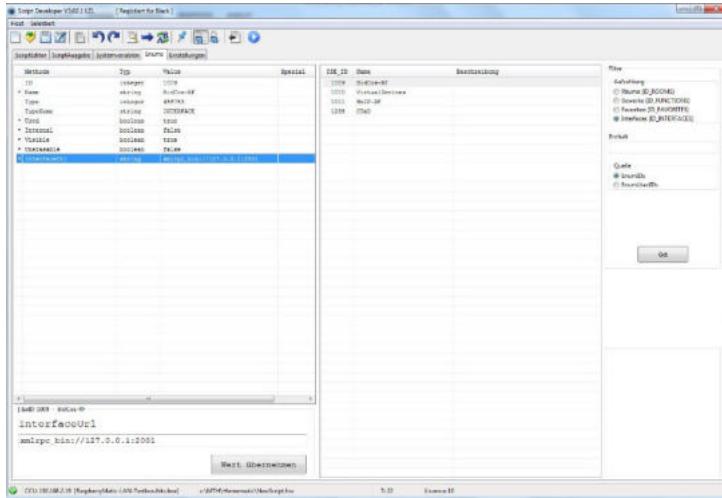
Properties, die in der ersten Zeile mit einem Stern (*) gekennzeichnet sind, können in ihrem Wert geändert werden.

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.

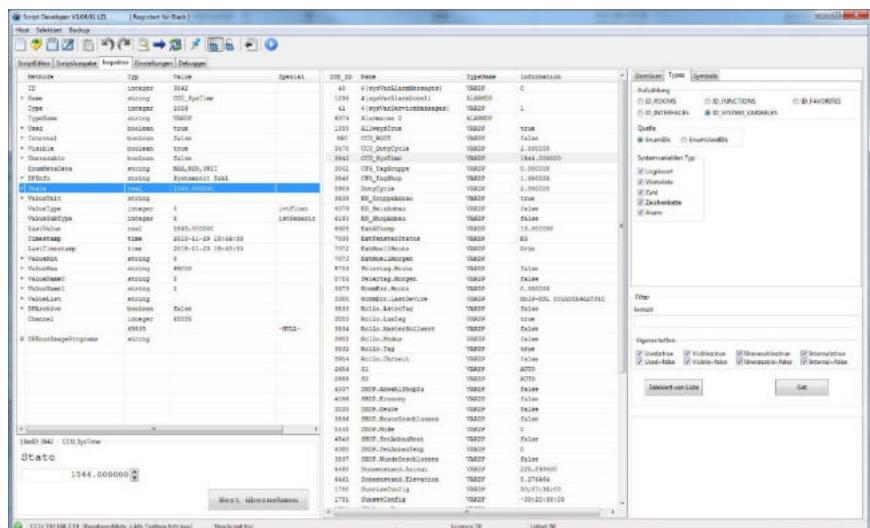


BreakOnRestart ist hier nicht Member von OT_PROGRAM sondern rekursiv von der entsprechenden Rule bzw SubRule.

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.



4.3.3 Schnellausführung von Get

Get kann auf folgenden Wegen ausgeführt werden:

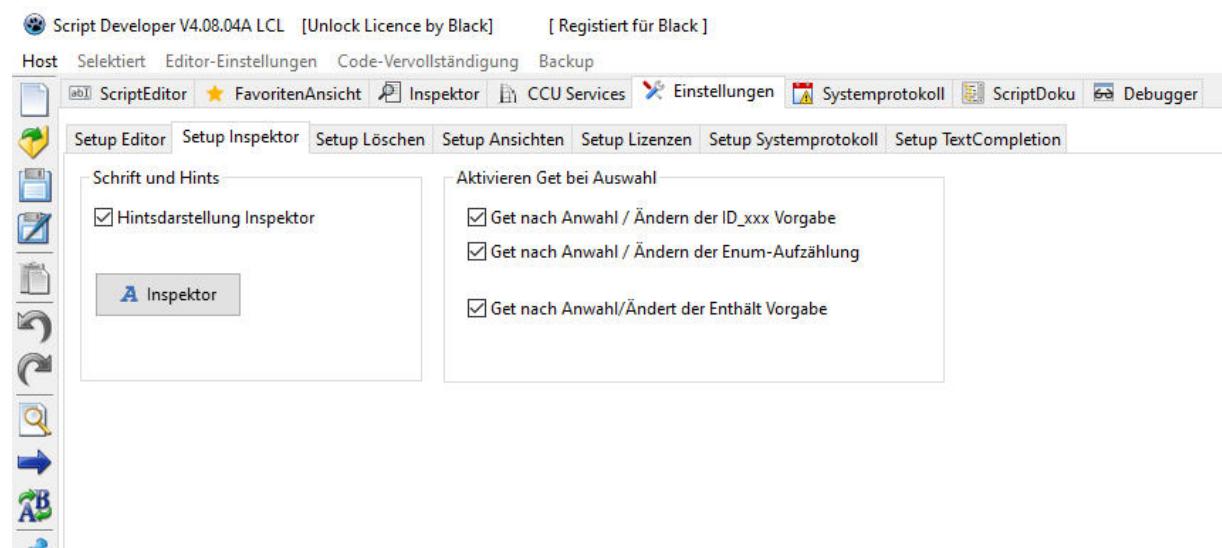
Druck auf den Button

Druck auf F1 im Inspektor (bzw der Taste, die für Skript ausführen definiert wurde)

Ändern der IDxx Eigenschaft (Wenn im Setup eingestellt)

Ändern der Enum Suchkriterien (Wenn im Setup eingestellt)

Ändern des Enthält Textes bzw der Bedingung (Beginnt, Endet, enthält) Eingestellt wird dies hier:

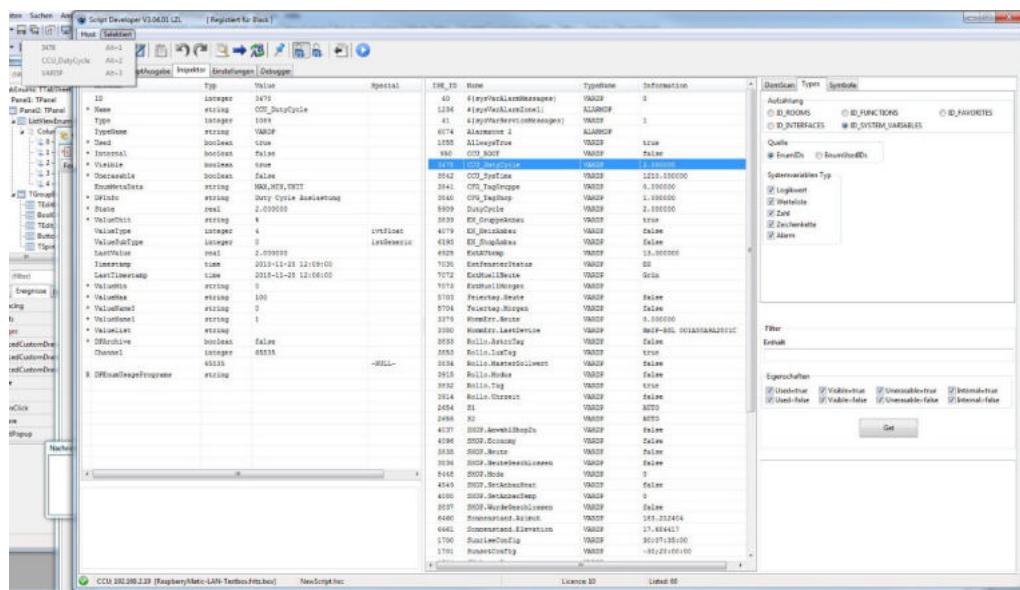


4.4 Daten aus Inspektor in Editor übernehmen

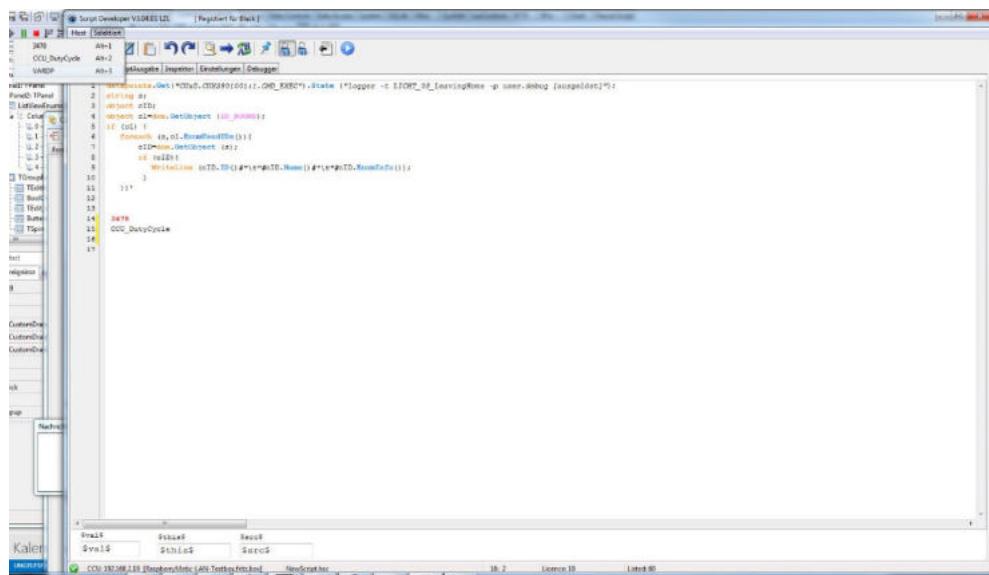
Zur schnelleren und auch möglichst fehlerfreien Bearbeitung besteht die Möglichkeit, Daten aus dem Inspektor direkt in den Editor zu übernehmen.

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

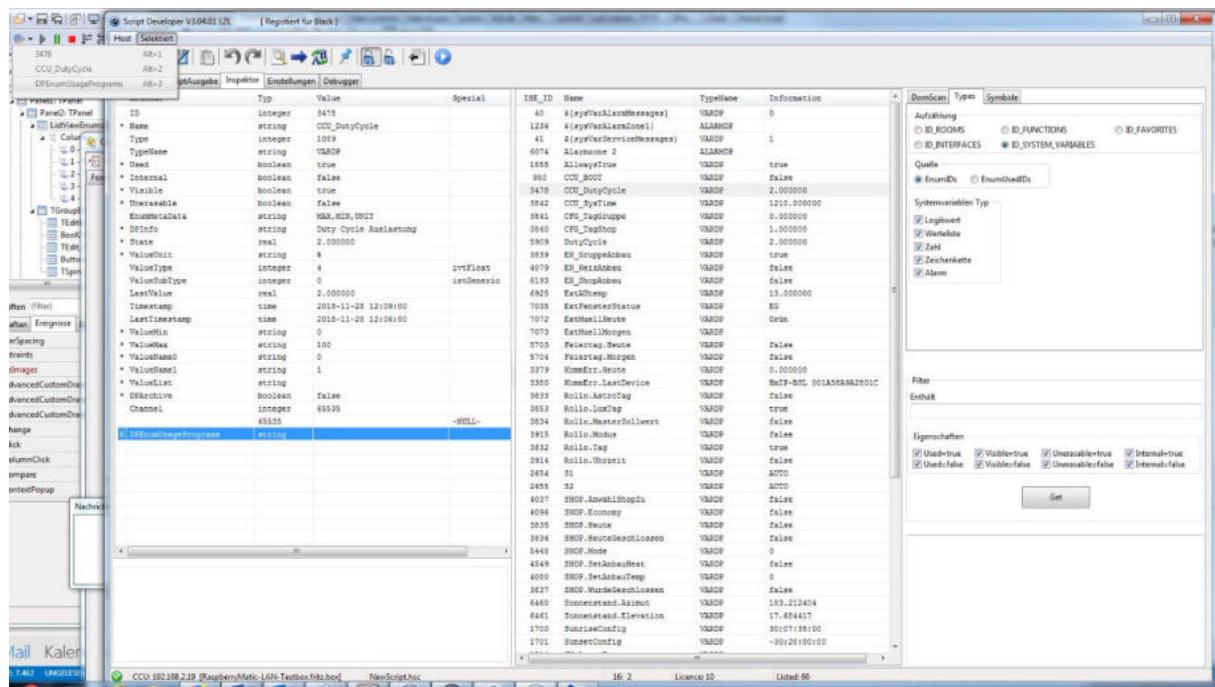
Mit den Tasten lassen sich folgende Spalteninhalte, welche vorher im Detailfeld selektiert worden sind, in den Editor übernehmen

Alt+m: Inhalt von Methode

Alt+t: Inhalt von typ

Alt+v: Inhalt von Value

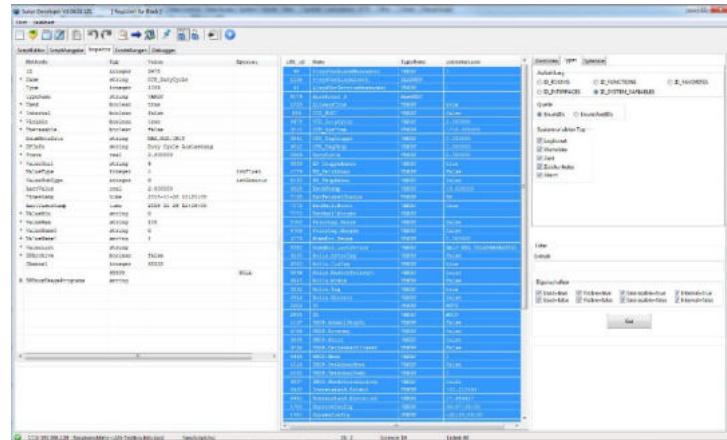
Alt+p: Inhalt von Spezial



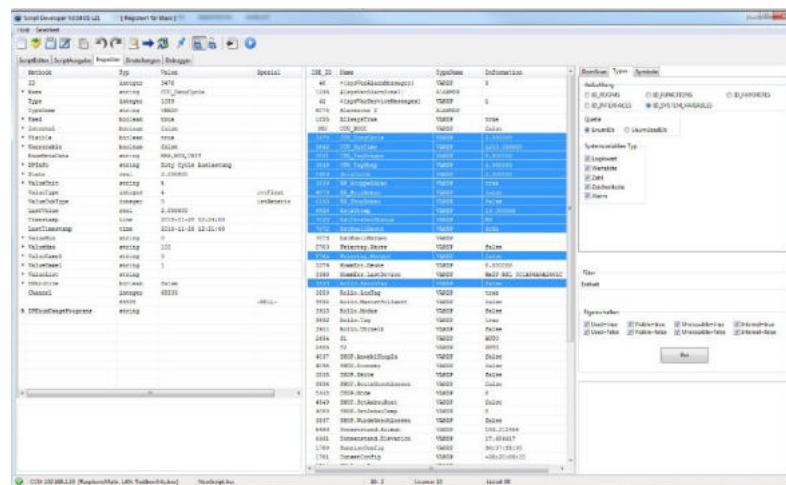
4.4.1 Mehrfachauswahl als Enum String

Es lassen sich im Hauptauswahlfeld Mehrfachselektionen vornehmen.

Ctrl-A : alle auswählen.



Oder die Übliche Mausbedienung:



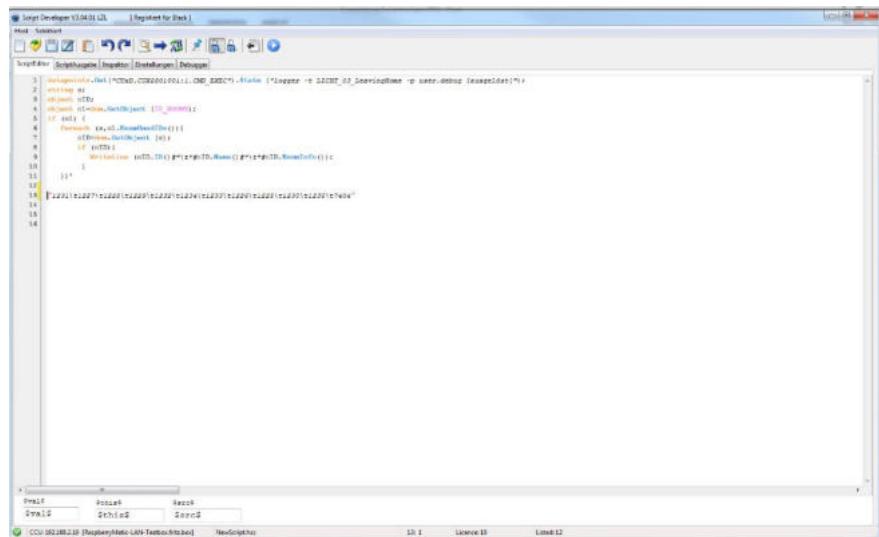
Ab Version 3.07.07.

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



Mit der **Pin** Taste merkt sich der Inspektor die Auswahl, welche sich dann Skript 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 Skript 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:

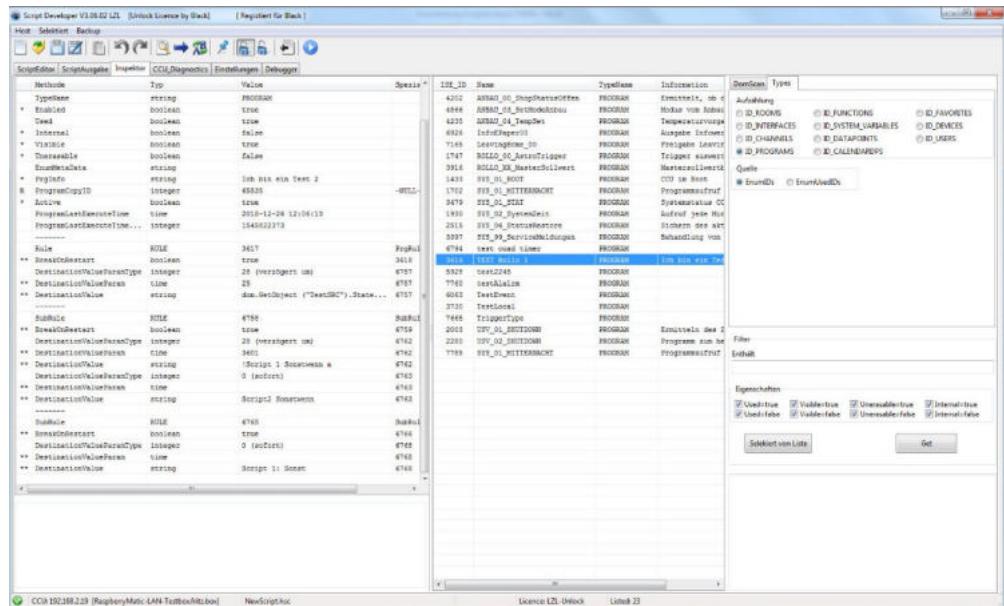
Mit rechter Maustaste im Mittleren Feld die Funktion „Einfügen aus Pinliste“ anwählen und die Sicherheitsabfrage bestätigen,



dann befinden sich die Selektierten Elemente wieder im mittleren Feld.

4.4.2 Übernahme von einem Skript 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 Skript enthalten



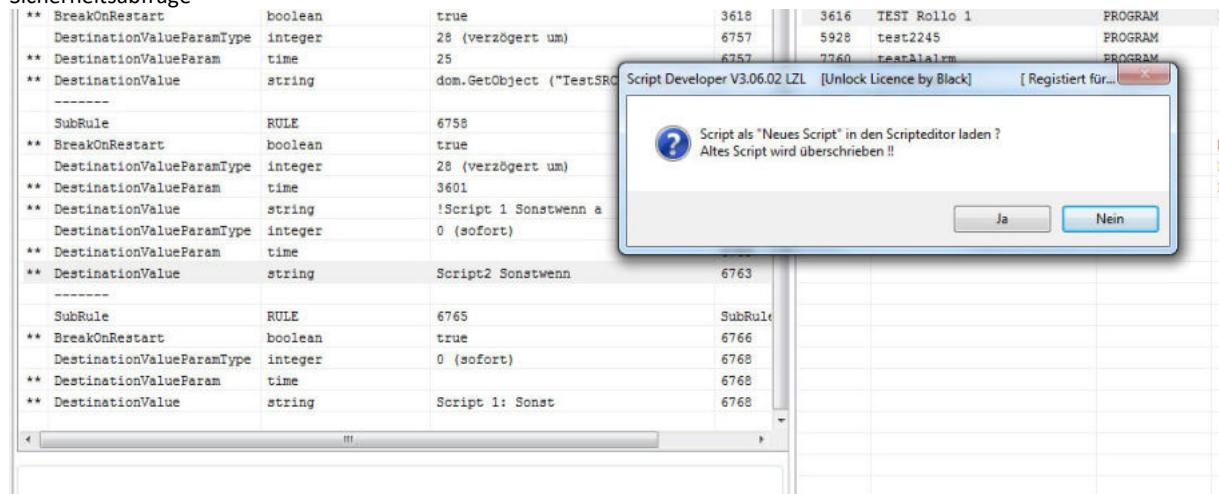
Dies entspricht der tabellarischen Darstellung dieses Programmes

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

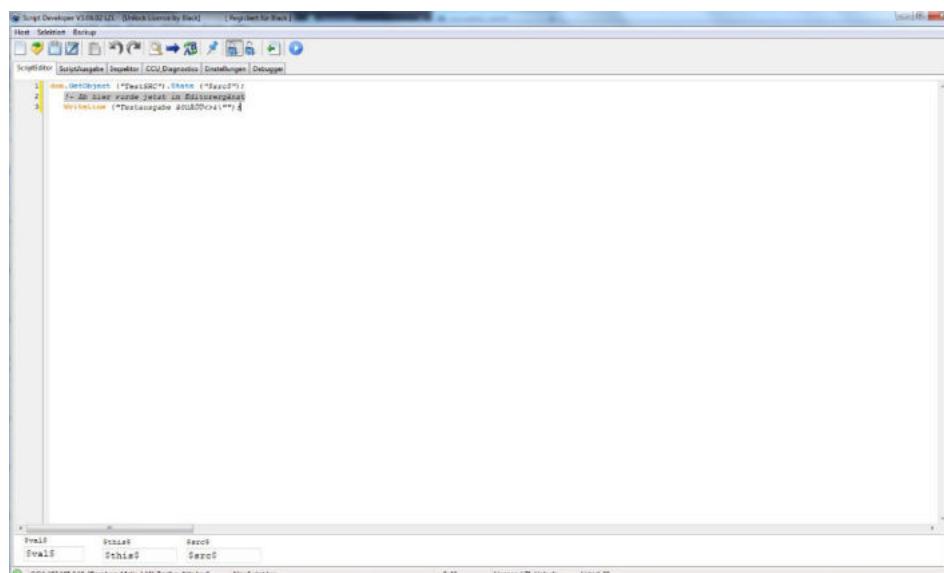
Die erste Subrule enthält dann die beiden Skripte aus der SonstWenn Bedingung.

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

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



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



Danach kann durch Drücken der nun nicht mehr grauen Taste



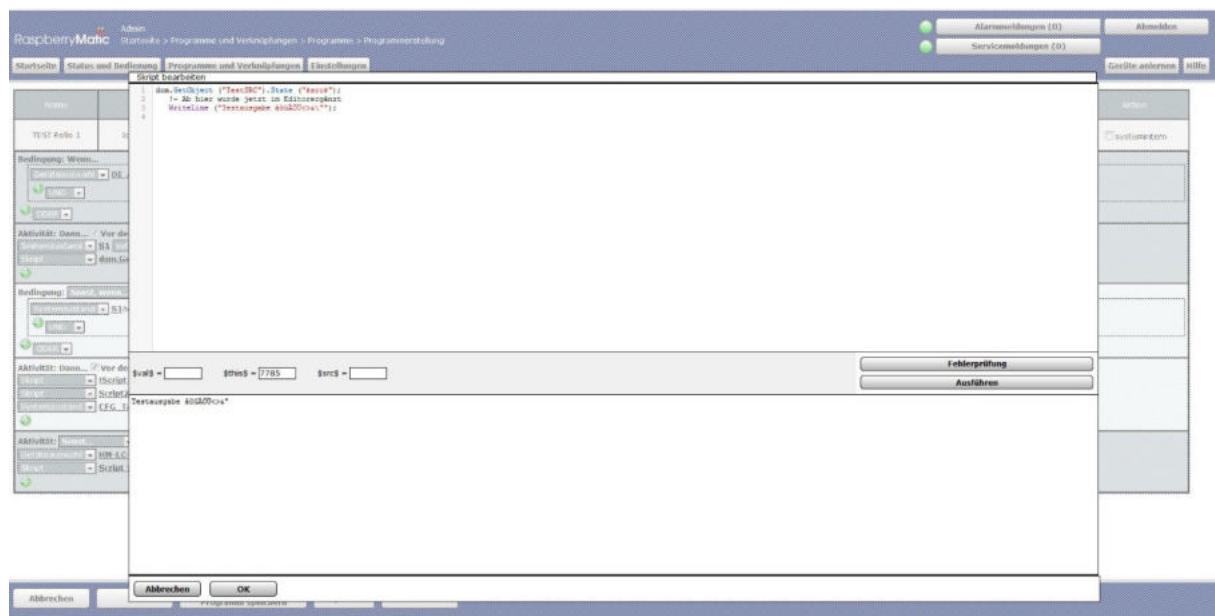
Das Programm aus dem Editor direkt in das Programm der CCU geladen werden. Im Hint wird als Gedächtnisstütze der Name des Programmes mit angezeigt.

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



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

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



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

4.4.3 Einzelne Spalten in Zwischenablage kopieren

Einzelne Elemente aus den beiden Listenansichten lassen sich in die Zwischenablage kopieren.

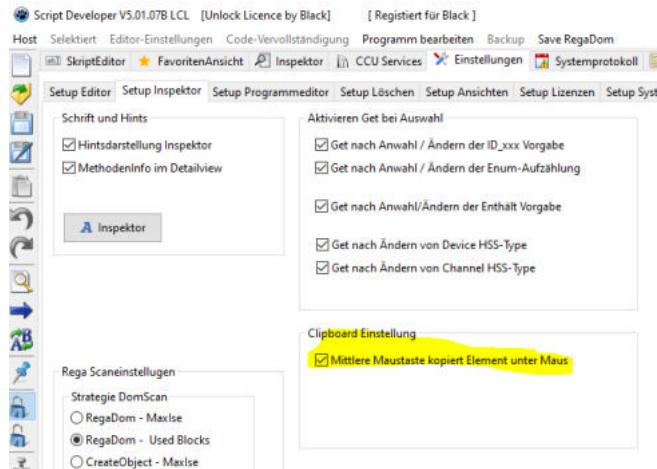
1. Möglichkeit

Das entsprechende Listen-Control muss aktiv sein. Maus über das zu kopierende Element bringen. Mit **Ctrl+Shift+C** wird das entsprechende Element in die Zwischenablage kopiert

2. Möglichkeit

Diese Option muss aktiviert sein:

Dann kopiert die mittlere Maustaste das Element, welches unter dem Mauszeiger liegt, in die Zwischenablage:



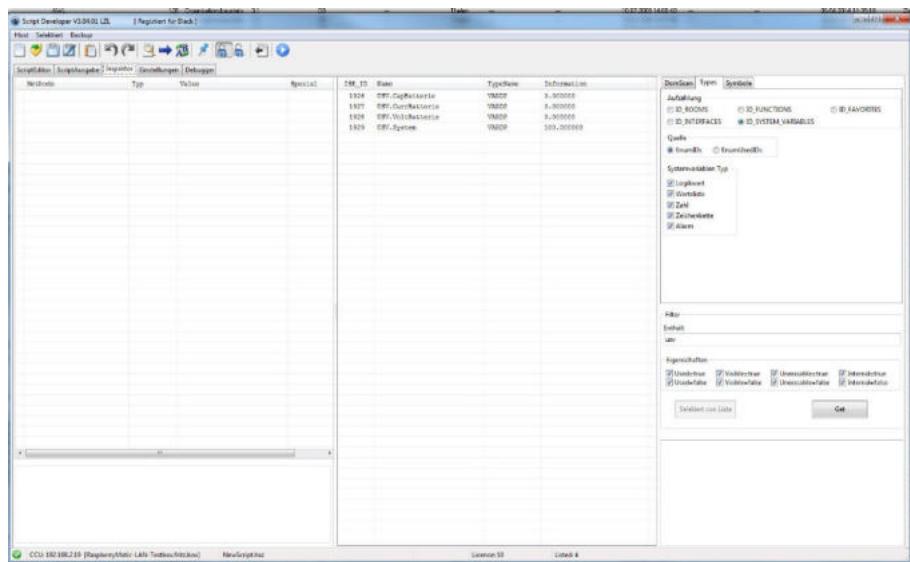
4.5 Selektion von Selektion

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

Hier Beispiel

The screenshot shows a list view in Script Developer with a specific range selected. The selection is highlighted with a blue border. To the right of the list view, there is a context menu with various filter and selection options. The 'Selektiert von Liste' button is visible at the bottom of the menu.

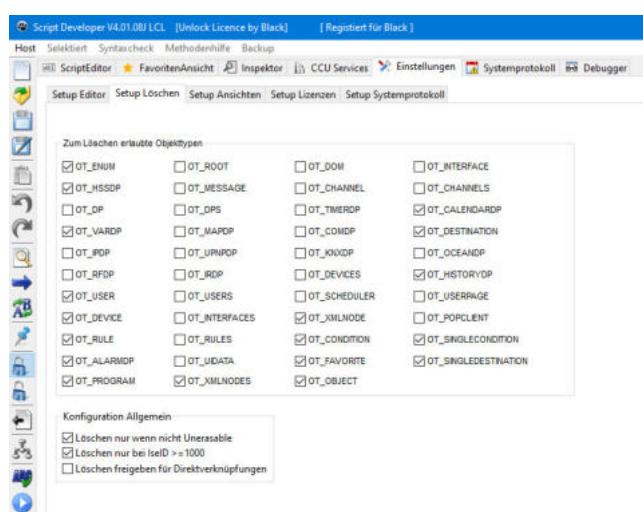
Selektierter Bereich von Systemvariablen, die hier darauf gefiltert werden sollen, dass der Name den String „usv“ enthält. Es müssen 4 Sysvars gefunden werden, die IDS 2012,2115,2116 werden hier nicht berücksichtigt, da diese sind selektiert sind. Bei <Druck auf: Selektiert von Liste: ergibt sich dann



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 löschbare Elemente zu löschen entweder:

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

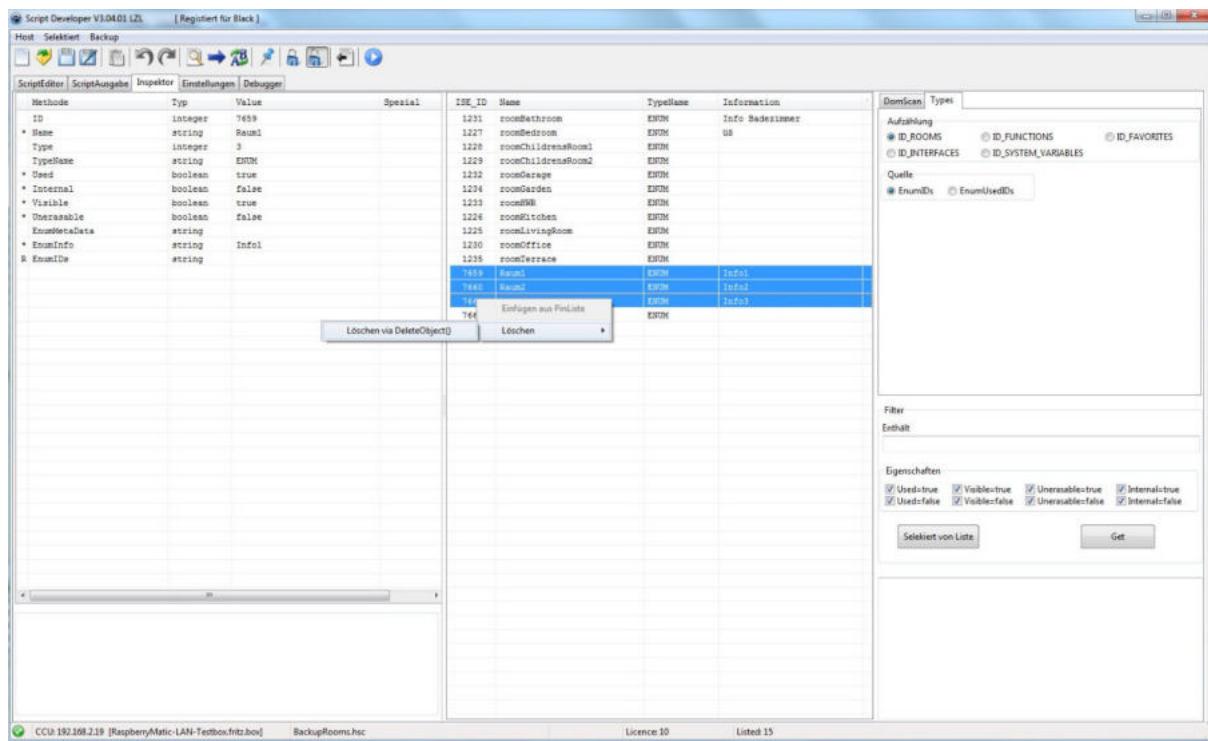
Löschen nur wenn ID>=1000. Dieser haken verhindert, dass man versehentlich Interne IDs der CCU (normalerweise unter kleiner 1000 angelegt) löscht. Will man in dem Bereich löschen, muss der hier explizit manuell unchecked werden.

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

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

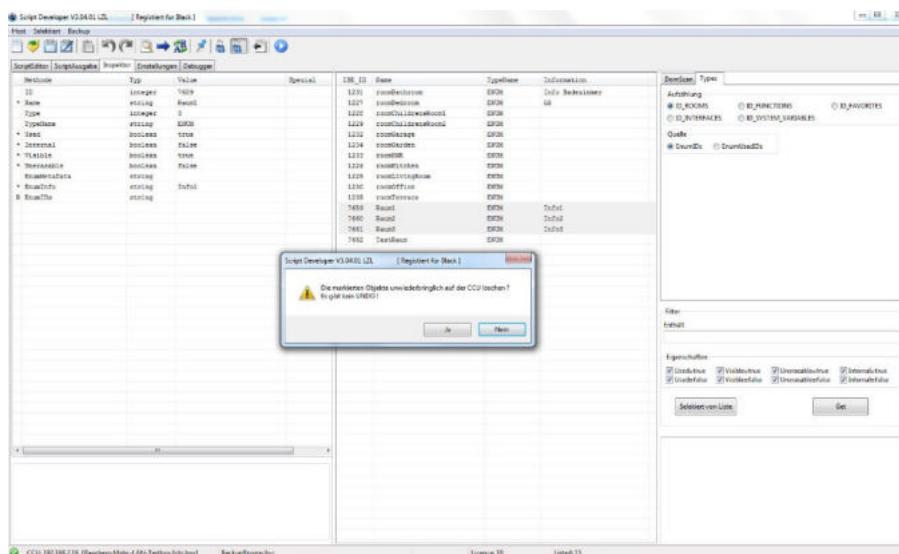
Löschen Freigeben von Direktverknüpfungen: Erlaubt, das Direktverbindungen ebenfalls gelöscht werden können

Löschen läuft so ab:



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

Mehrfachselektion ist möglich

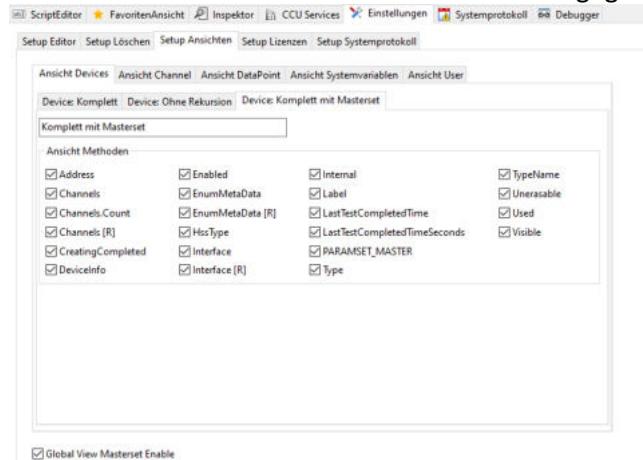


Nach dieser Sicherheitsabfrage sind die Objekte dann weg.. Zurück geht's dann nur mit Restore. Bei einem Device Objekt gibt es noch die zusätzliche Möglichkeit Löschen über xmlrpc.deleteDevices ().

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

ID	Name	Type	Special	ISK_ID	Name	TypeName	Information
2718	CUD-B-EXEC	DEVICE		2718	Root-devices	DEVICES	
1239	CUD-TIME	DEVICE		1239			
1444	DI_ATELIER	DEVICE		1444			
12	Gateway	DEVICE		12			
3564	HM-LC-SwG	DEVICE		3564			
1012	HM-RCV-S6	DEVICE		1012			
7418	HaIP-SNL	DEVICE		7418			
7624	HaIP-SLO	DEVICE		7624			
7586	HaIP-SPI	DEVICE		7586			
4050	HaIP-SRH	DEVICE		4050			

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

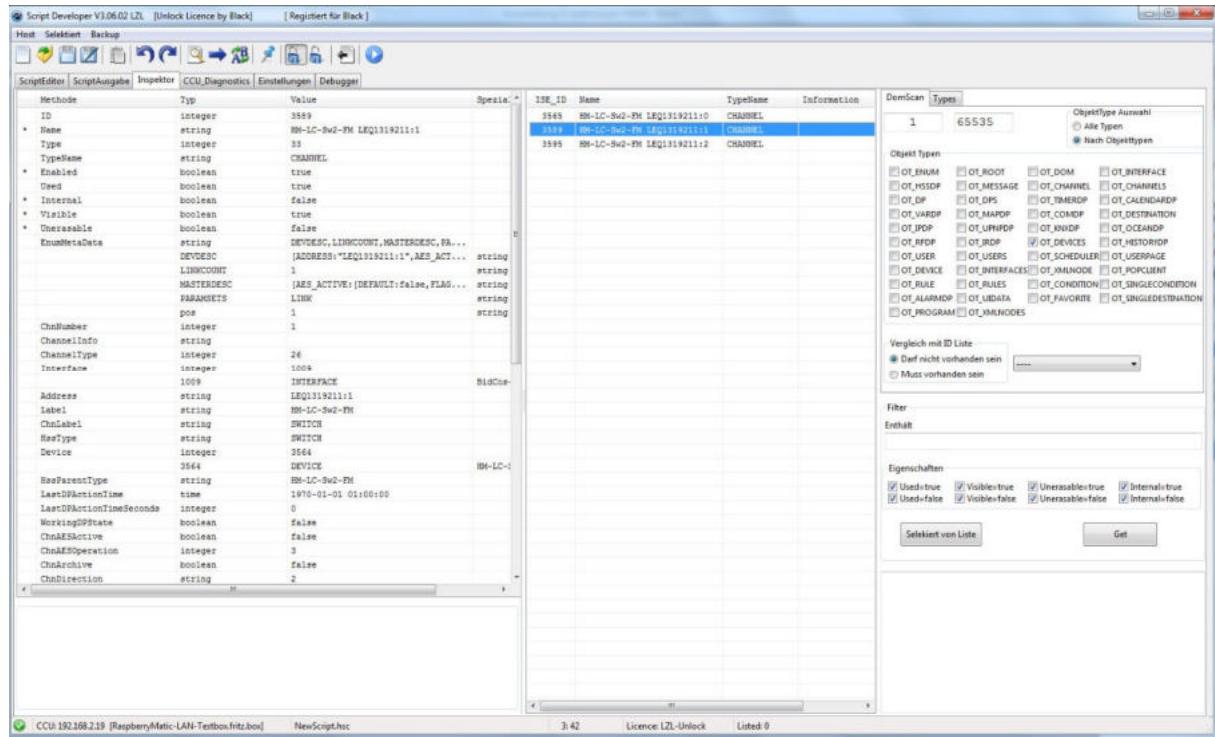
ID	Name	Type	Special	ISK_ID	Name	TypeName	Information
3564	HM-LC-SwG-PM	DEVICE	BisCoe-RF	2718	CUD-B-EXEC	DEVICE	BisCoe-RF
1012	HM-RCV-S6	DEVICE	BisCoe-RF	1239	CUD-TIME	DEVICE	CUD
7418	HaIP-SNL	DEVICE	BisIP-RF	1444	DI_ATELIER_TOER	DEVICE	BisCoe-RF
7624	HaIP-SLO	DEVICE	BisIP-RF	12	Gateway	DEVICE	---
7586	HaIP-SPI	DEVICE	BisIP-RF	3564	HM-LC-SwG-PM	DEVICE	BisCoe-RF
4050	HaIP-SRH	DEVICE	BisIP-RF	1012	HM-RCV-S6	DEVICE	BisCoe-RF

Und rekursiv weiter über die Channels des Devices

Mal in die Verwendung in dem Programm schauen über ChnEnumDPUsagePrograms

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

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



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

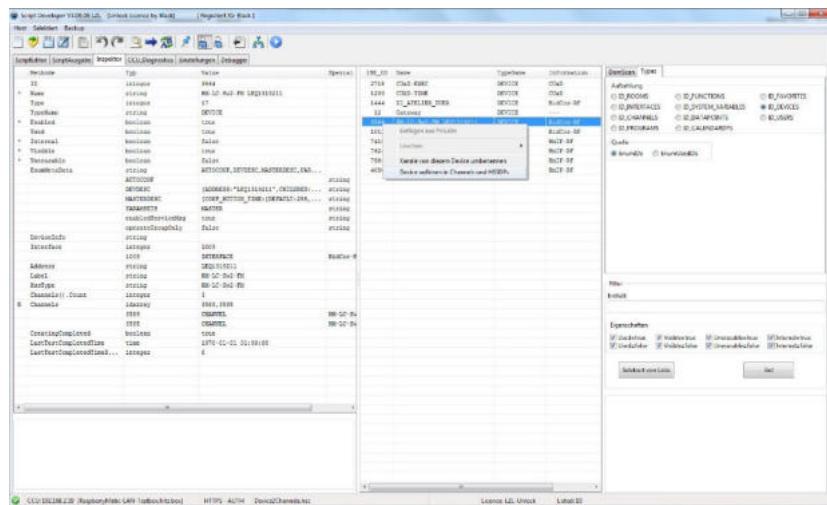
4.9 Auflösen komplexer Objekte

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

Beispielsweise Devices und auch Programme.

4.9.1 Auflösen von Devices

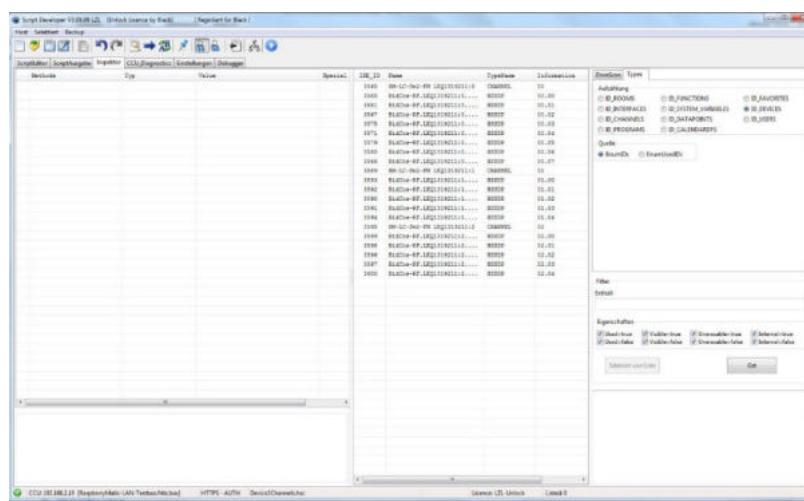
Ein Device besteht ja aus Channels und die Channels wiederum aus Datenpunkten



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

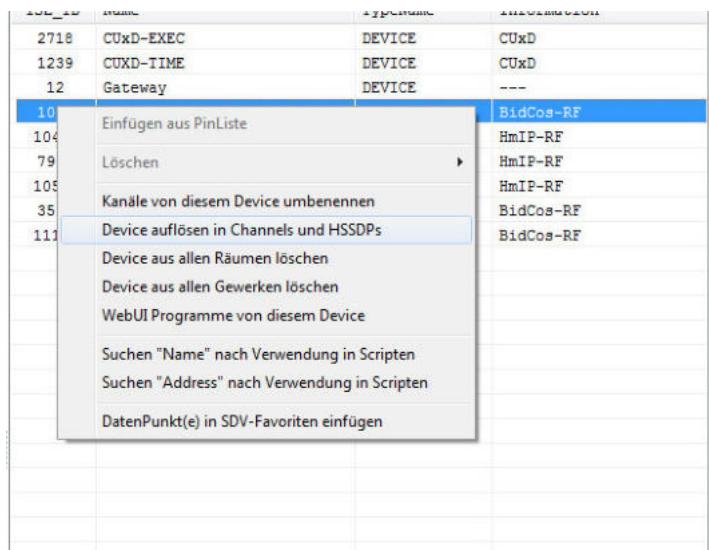
Alternativ über den neuen Menüpunkt . Dieser ist nicht ausgegraut, wenn sich ein Objekt auflösen lässt.

Als Ergebnis erhält man :



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

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



Das dargestellte Menü ist abhängig von dem selektierten Objekttyp.

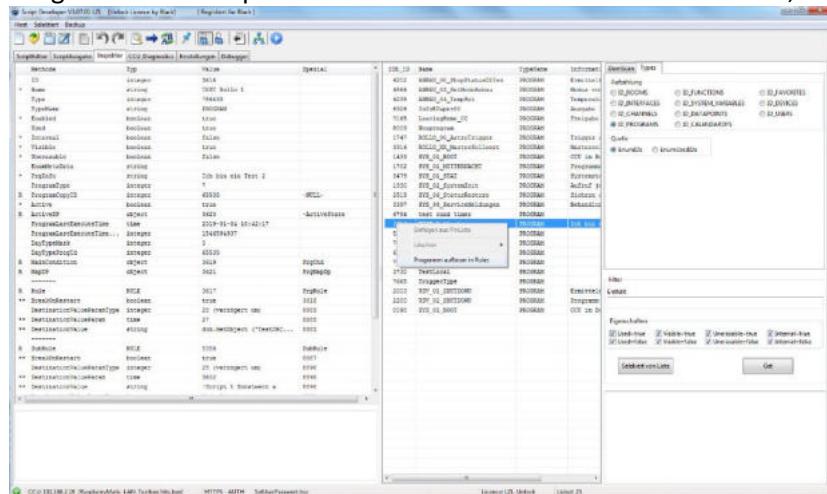
Bei Devices z.b.

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

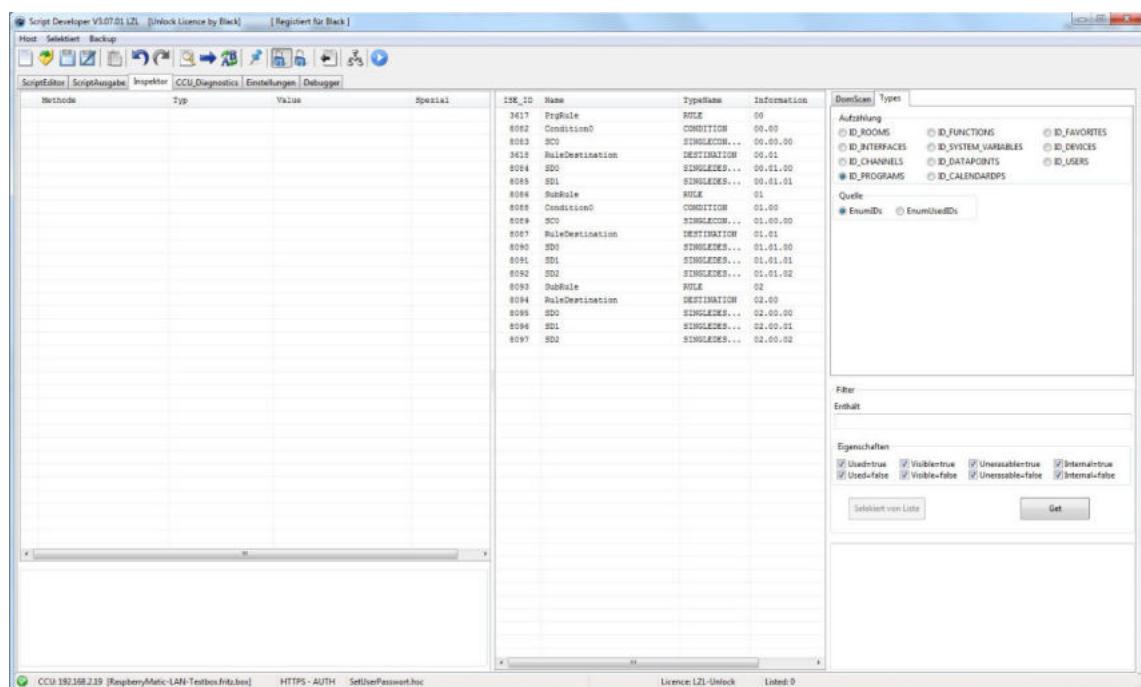
4.9.2 Auflösen von Programmen

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

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



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



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

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

- Left Pane (List View):** Shows a table of objects with columns: Methode, Typ, Value, Spezial, ISE_ID, Name, Typename, Information.
- Right Pane (Detail View):** Shows a table of objects with columns: ISE_ID, Name, Typename, Information.
- Toolbar:** Includes icons for ScriptEditor, ScriptAusgabe, Inspektor, CCU_Diagnostics, Einstellungen, and Debugger.
- Filter Bar:** Contains "Aufzählung" (Enumeration) and "DomScan" sections with various filter options like ID_ROOMS, ID_FUNCTIONS, etc.
- Properties Panel:** Shows checkboxes for properties like Used=true, Visible=true, etc.
- Status Bar:** Displays "CCU:192.168.2.19 [RaspberryMatic-LAN-Testbox.fritz.box] HTTPS - AUTH SetUserPassword.hsc" and "Licence: L2L-Unlock Listed: 0".

4.10 Zeitmodule

Zeitmodule lassen sich nun auch in Detailansicht darstellen

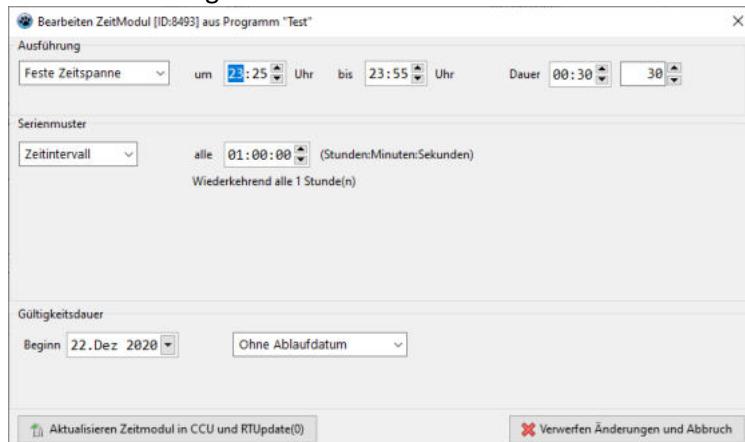
The screenshot shows the Script Developer interface with the following details:

- Left Pane (List View):** Shows a table of time module objects with columns: Methode, Typ, Value, Spezial, ISE_ID, Name, Typename, Information.
- Right Pane (Detail View):** Shows a table of time module objects with columns: ISE_ID, Name, Typename, Information.
- Toolbar:** Includes icons for ScriptEditor, ScriptAusgabe, Inspektor, CCU_Diagnostics, Einstellungen, and Debugger.
- Filter Bar:** Contains "Aufzählung" (Enumeration) and "DomScan" sections with various filter options like ID_ROOMS, ID_FUNCTIONS, etc.
- Properties Panel:** Shows checkboxes for properties like Used=true, Visible=true, etc.
- Status Bar:** Displays "CCU:192.168.2.19 [RaspberryMatic-LAN-Testbox.fritz.box] HTTPS - AUTH SetUserPassword.hsc" and "Licence: L2L-Unlock Listed: 3".

4.10.1 Ändern eines Zeitmodules

Ein Zeitmodul kann komfortabel geändert werden. Dazu entweder in der Listenansicht doppelklicken auf das Objekt, oder in der Detailansicht, wenn das Objekt ein Zeitmodul ist.

Es öffnet sich folgendes Menü:



Hier kann Intuitiv die Werte des Zeitmodules geändert werden. Aktualisieren ändert das CalendarDP direkt und führt ebenso ein dom.RTUpdate(0) durch, damit die Rega diese Änderungen übernimmt. Verwerfen beendet das Menü ohne Datenübernahme.

4.11 Suchen in Skripten nach Variablen, Devices etc

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

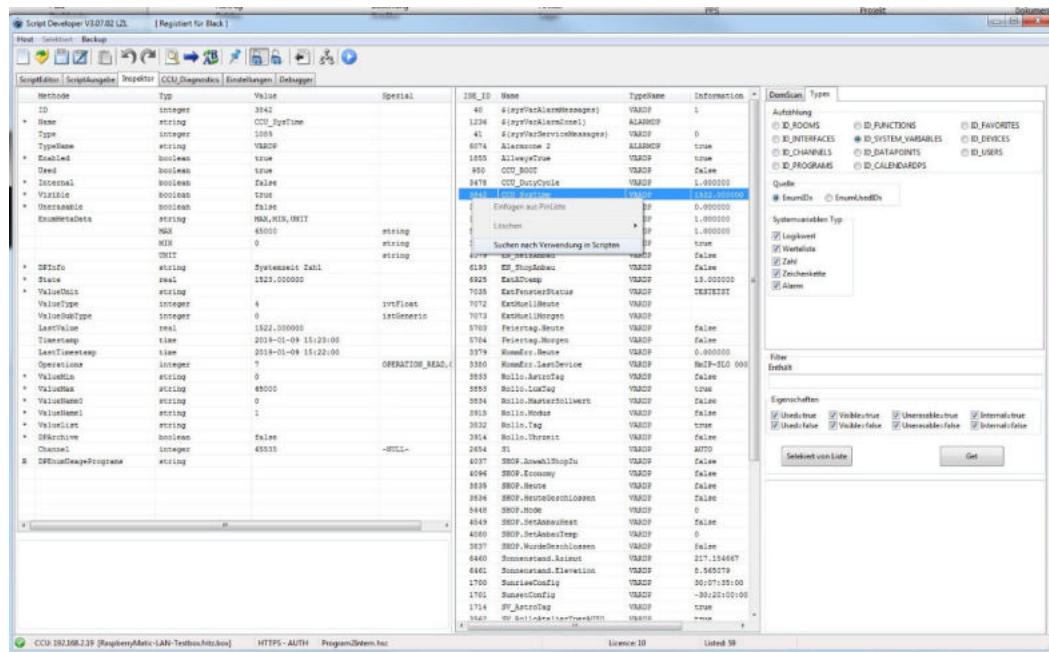
Dazu werden sämtliche SingleConditions, die als Property ein Script enthalten, gesucht und dann via Stringvergleich nach dem oder den Vorkommenden Namen abgesucht.

Vorgehensweise:

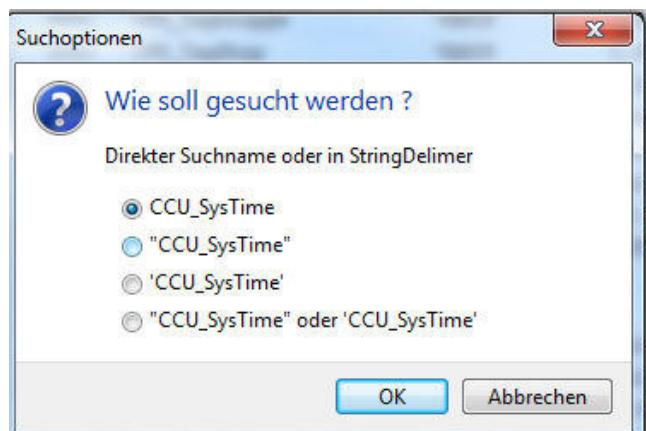
Markieren der oder auch mehrere Suchvariablen (Können Systemvariablen, Devices, Kanäle, Räume und Gewerke sein) Rechte Maustaste und suchen nach Verwendung in Skripten

Es kann hierbei gesucht werden:

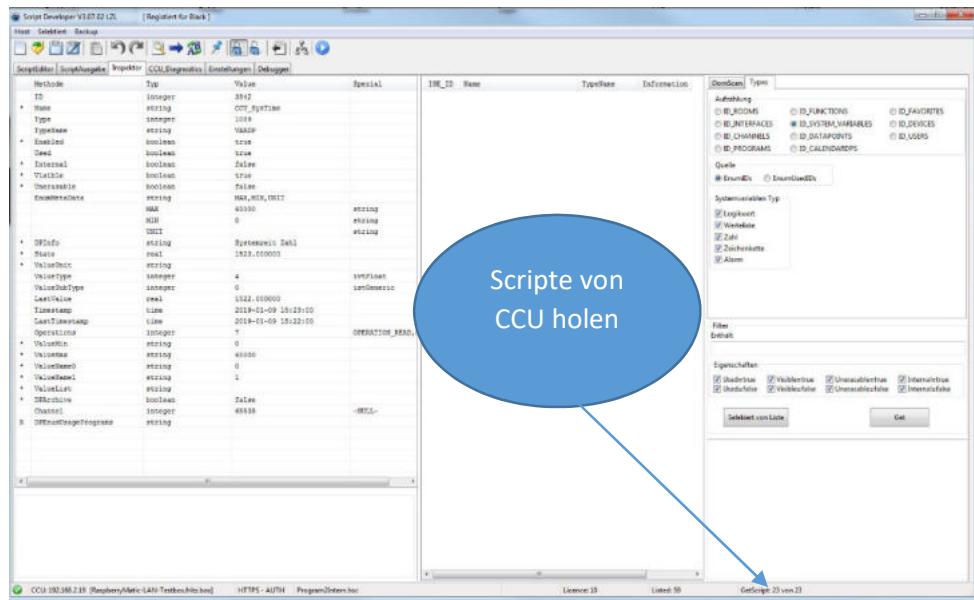
- Suchen "Name" nach Verwendung in Skripten : Suche nach dem Namen (SV, Alarme, Device, Chans)
- Suchen "Address" nach Verwendung in Skripten : Sucht nach der Seriennummer (Address) von Devices und Channels



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

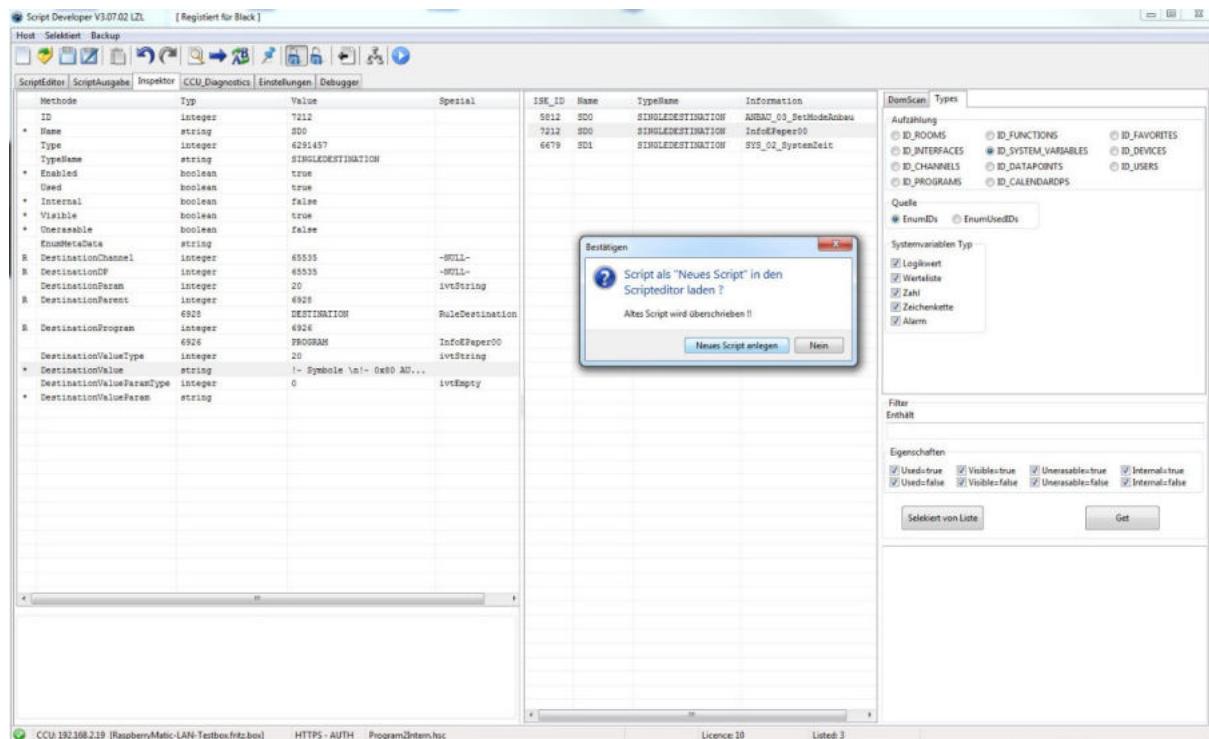


Daraufhin werden erstmal alle Rule in ihre Destinations aufgedröselt und eine Liste angelegt, wie viele Skripte es in diesen SingleDestinations dann gibt. Diese werden in den PC geladen. Das geht recht Fix, um der CCU zwischenzeitlich Zeit für Ihre Aufgaben zu lassen, wird direkt nach dem Empfang PC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



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

Das Skript 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

```

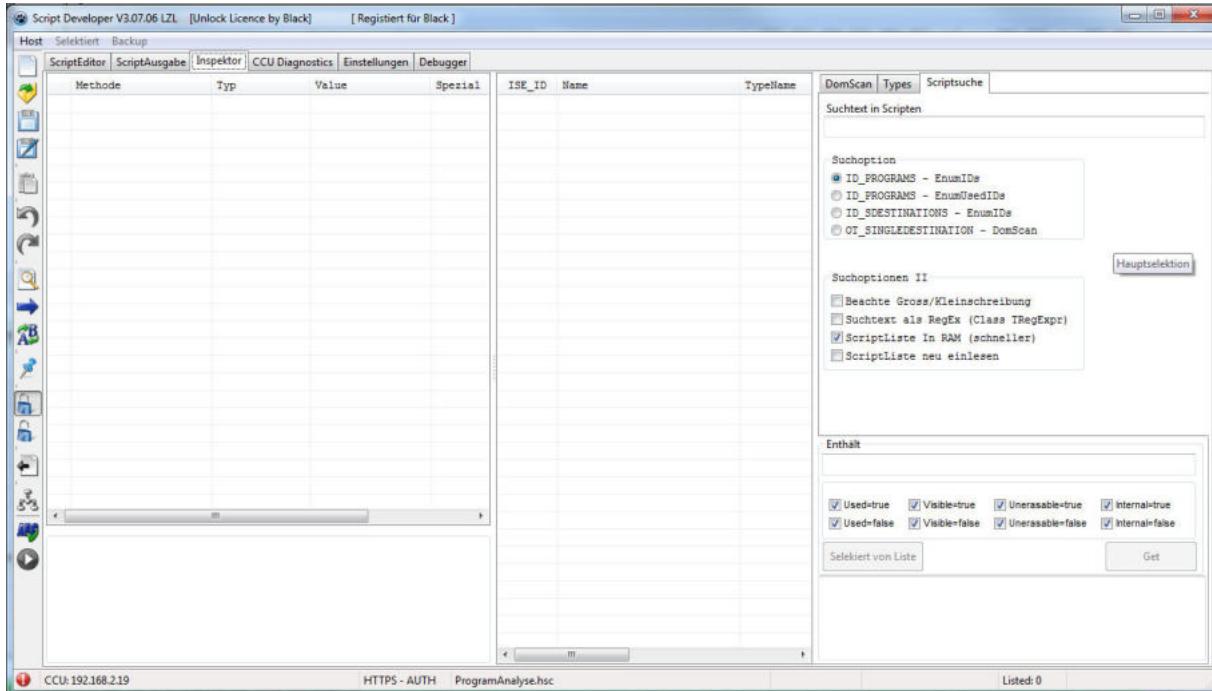
29 /* 0x01 LANG LANG
30 /* 0x02 LANG KURZ
31 /* 0x03 LANG MÜLL
32 /* 0x04 KURZ
33 /* 0x05 KURZ MÜLL
34 /* 0x06 LANG
35
36 /* Signale
37 /* 0xF0 AUS
38 /* 0xF1 Rotes Blitzen
39 /* 0xF2 Grünes Blitzen
40 /* 0xF3 Orangenes Blitzen
41
42 /* Seriennummer des BH-Disp-EP-WMS,
43 string seriennummer = "DEQ1443971";
44 object oSV= dom.GetObject("ID_SYSTEM_VARIABLES").Get("SVSubmit_D1_EP_Flur");
45
46 /*-arbeiten für die Texte und Icons
47 /*-Nicht verwendete Variablen nicht löschen, sondern auf "" setzen!
48 /*-Definierte Variablen können 0x00-0x89 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= "";
53 } else {
54     zeile2= oObj.State;
55     icon2= "0x02";
56 }
57
58 string zeile3 = dom.GetObject("ID_SYSTEM_VARIABLES").Get("ExtAUtemp").State.ToString();
59 string icon3 = "";
60
61 /* Nullstatus nach der Uhrzeit darstellen
62 /* Wenn heute Null und Uhrzeit zwischen 6-8 Uhr Heute Null darstellen
63 /* Andernfalls
64 /* Wenn Morgen null und Uhrzeit zwischen 18 und 24 Uhr.. Morgen Null darstellen
65 int iSysTime= dom.GetObject("ID_SYSTEM_VARIABLES").Get("CCU_System").State.ToIntInteger();
66
67 string zeile4 = "";
68
69 object oObj= dom.GetObject("ID_SYSTEM_VARIABLES").Get("ExtMailHeute");
70 if ((oObj.State != "") && (iSysTime>20) && (iSysTime<830)) {
71     zeile4= oObj.State;
72     icon4= "0x04";
73 }
74
75
76 $val$ $this$ $src$  

77 $val$ $chis$ $src$
```

CCU:192.168.2.19 [RaspberryMatic-LAN-Testbox.fritz.box] HTTPS - AUTH NewScript.hsc Licence: 10 Listed: 3

4.12 Volltextsuche in Skripten

Ab Version 03.07.07 existiert eine Volltextsuche (ab Level 6)



Suchtext ist selbsterklärend.

Entweder wird nach dem Vorkommen des Textes in dem Skript 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 Skripte die das Wort Sonst, aber auch z.B. Sonat enthalten. Syntax nachzulesen unter

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

Skriptliste 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.

Skriptliste neu einlesen

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

4.13 Suchen Verwendungsstelle in WEB-UI Programmen (ab 3.09.04)

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

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

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

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

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

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

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

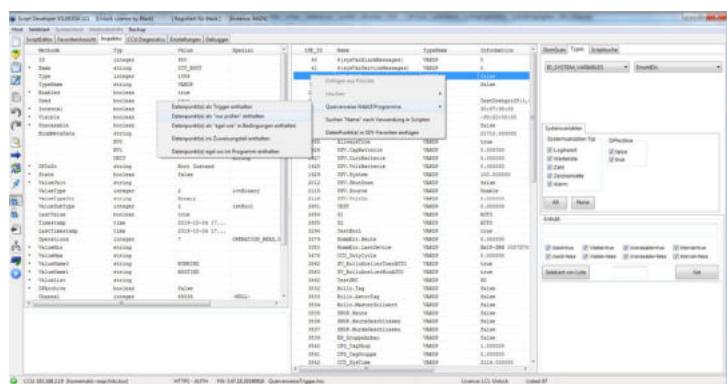
Eine dieser IDs kommt im Bedingungs oder Anweisungsteil vor --> Ergebnis in den Inspektor

(Skriptsuche ist davon unabhängig, der SDV konnte auch schon vorher Volltextsuchen in Skripten)

Arbeitsweise:

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

Dabei kann nun eine der Suchvergänge ausgewählt werden

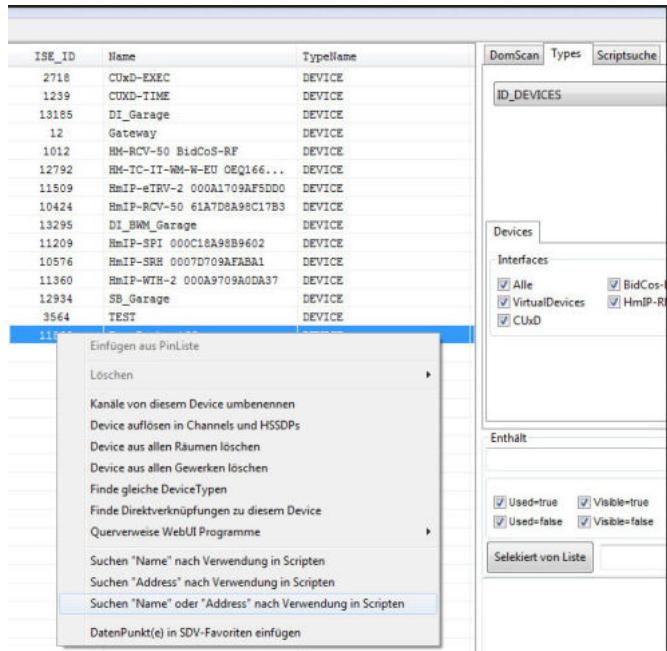


Ergebnis wäre hier die List von Programmen, wo in der WebUI eine Zuweisung auf die Sysvar Anwesenheit gemacht wird

ISE_ID	Name	TypeName	Information	DomScan	Types
8632	Demo	PROGRAM	DemoTest		ID_SYSTEM_VAR
1433	SYS_01_BOOT	PROGRAM	CCU im Boot		

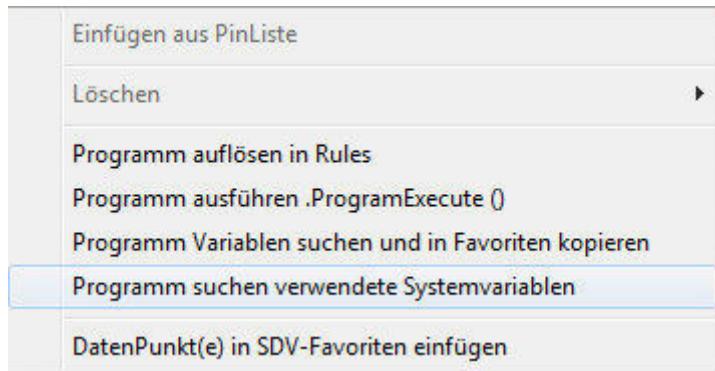
4.14 Suchen „Name“ oder „Adress“ nach Verwendung in Skripten

Das angeklickte Device im Inspektor wird nach vorkommen auf „Name“ bzw der „Adress“ (Seriennummer) in Skripten gesucht.



4.15 Programm suchen verwendete Systemvariablen

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



4.16 SingleDestinations in ihrer Reihenfolge ändern

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

Vorher:

Name	Beschreibung	Bedingung (Wenn...)
TEST Rollo 1	Ich bin ein Test 2	Kanalzustand: DI_ATELIER_TUER:1 bei Fensterzustand: verriegelt bei Aktualisierung auslös

Bedingung: Wenn...

Geräteauswahl ▼ DI_ATELIER_TUER:1 bei Fensterzustand: verriegelt ▼ bei Aktualisierung auslösen □

+ UND □

+ ODER □

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

Systemzustand □ S1 sofort □ AUTO □ a □

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

+

Bedingung: Sonst, wenn...

Systemzustand □ S1 bei HAND □ bei Änderung auslösen □

+ UND □

+ ODER □

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

Skript □ !-Script qqqq... verzögert um □ 3601 □ Sekunden □

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

Systemzustand □ CFG_TagGruppe sofort □ 0.00 □

+ Das UrsprungScript

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

Skript □ Script 1: Sonst... sofort □

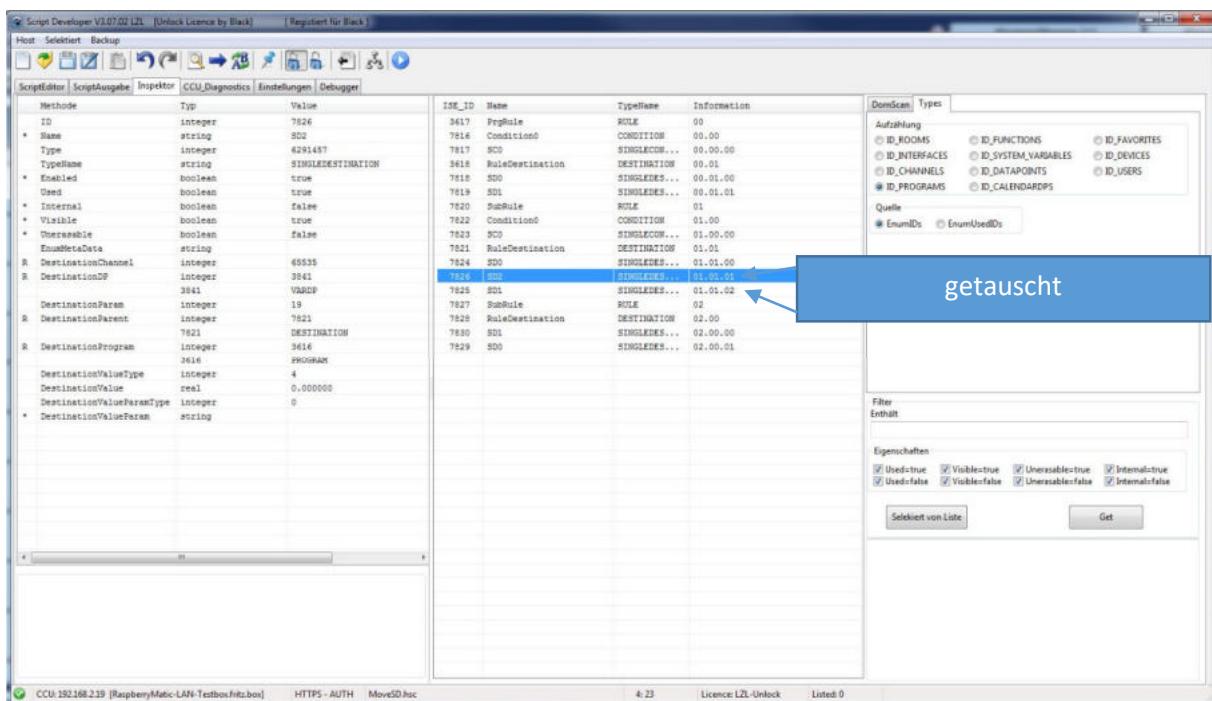
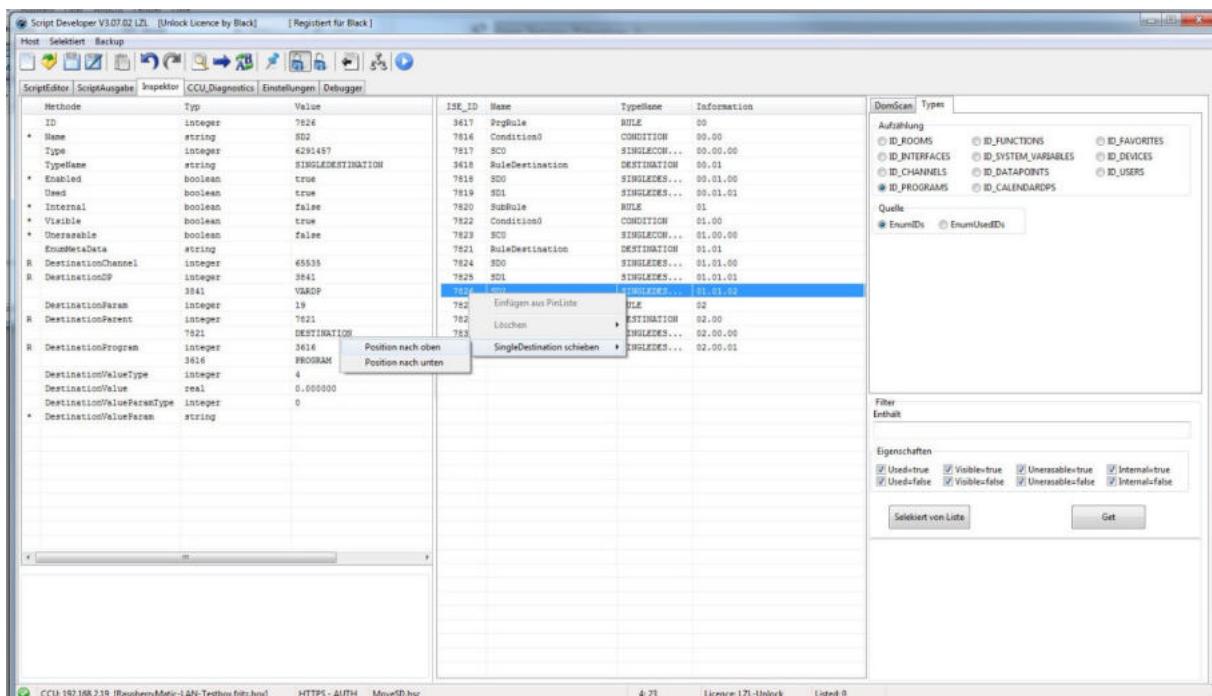
Geräteauswahl □ Test:1 sofort □ Schaltzustand: ein □

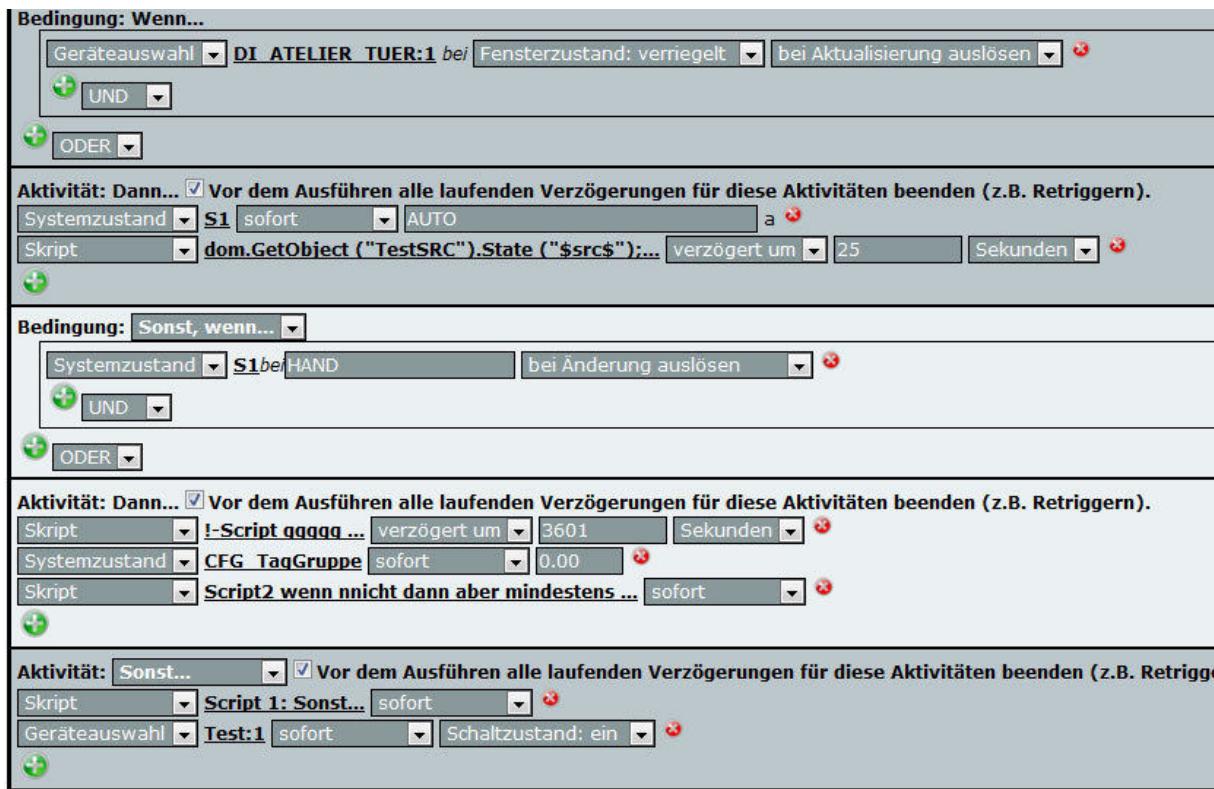
+

Das Aussehen im SDV

The screenshot shows the Script Developer V5.03 interface with a list of objects in the main pane. A blue callout box labeled "Sieht hier so aus" points to the "SingleDestinations..." row in the table. The table columns include ID, Name, Type, and Information. The "Information" column for the "SingleDestinations..." row shows the value "01,01,01".

ID	Name	Type	Information
7817	ProgRule	RULE	00
7818	Condition0	CONDITION	00,00
7819	SDU	SINGLECOM...	00,00,00
7820	RuleDestination	DESTINATION	00,01
7821	SDU	SINGLECOM...	00,01,00
7822	SDU	SINGLECOM...	00,01,01
7823	Condition0	CONDITION	01,00
7824	SDU	SINGLECOM...	01,00,00
7825	RuleDestination	DESTINATION	01,01
7826	SDU	SINGLECOM...	01,01,00
7827	SubRule	RULE	01,01,01
7828	SDU	SINGLECOM...	01,01,01
7829	SDU	SINGLECOM...	01,01,01
7830	RuleDestination	DESTINATION	02,00
7831	SDU	SINGLECOM...	02,00,00
7820	SDU	SINGLECOM...	02,00,01

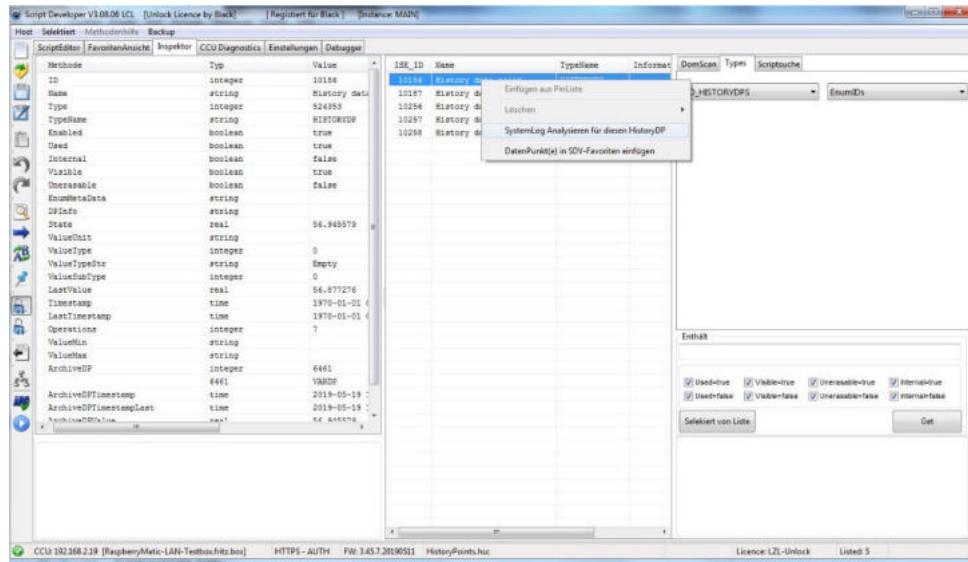




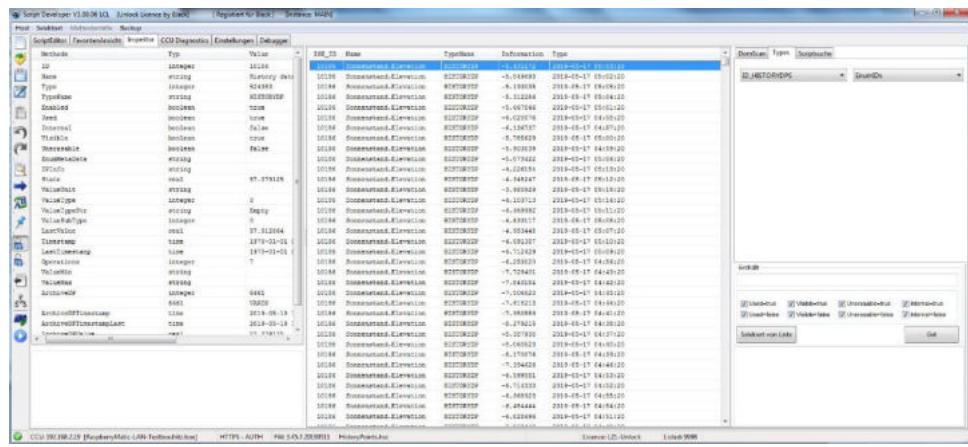
Und hier im WebUI Skript

4.17 History Data Points und Systemprotokoll

Es besteht die Möglichkeit, History Datapoints im Systemprotokoll zu Selektieren



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



4.18 EchtZeit Hilfe Hints zu Objekten im Inspektor

Um effektives Arbeiten und Untersuchen von Objekten zu vereinfachen, verfügt der Inspektor über Hilfehints. Diese werden in Echtzeit Abhängig von jeweiligen Objekt dargestellt:

Beispielsweise in der Listendarstellung:

Hilfe bei einer Systemvariablen:

tempotokoll Debugger			
ISE_ID	Name	TypeName	Information
40	\$(sysVarAlarmMessages)	VARDP	0
41	\$(sysVarServiceMessages)	VARDP	10
19252	-Umlauteäöüß test	VARDP	false
1236	Alarmzone 1	ALARMDP	false
19253	Alarmzone 2	ALARMDP	
1855	All Systemvariable - Typ: Alarm		true
13725	Anwe-----		true
951	Name : Alarmzone 1		false
13726	Anwe-----		true
13722	Anwe Beschreibung: S.USV Power Loss		false
13723	Anwe Wert: 2 (nicht ausgelöst)		false
13723	Anwe 23.07.2020 15:18:27		1
13721	Anwesenheit.Nutzer	VARDP	true
13724	Anwesenheit.string	VARDP	Black
13727	Anwesenheit.Susa	VARDP	false
9704	AstroTag	VARDP	true

Oder bei einem Zeitmodul:

tempotokoll Debugger				
ISE_ID	Name	TypeName	Information	Type
15882	Zeitmodul	CALENDARDP		2...
16802	Zeitmodul	CALENDARDP	ZEIT_00_MITTERNACHT	2...
17504	Zeitmodul	CALENDARDP	Astrotest	2...
19022	Zeitmodul	CALENDARDP	Astrozeit 3	2...
19037	Zeit Zeitmodul aus Programm "Astrozeit 3"			2...
19093	Zeit -----			2...
19130	Zeit Auslösezeitpunkt		- Zeitspanne von 10 Minuten nach Sonnenaufgang für 60 Minuten	2...
19145	Zeit		- Aktuell: 06:06:00 bis 07:06:00	2...
19231	Zeit Serienmuster			2...
19233	Zeit		- täglich, jeden Tag	
19235	Zeit		Nächster Auslösezeitpunkt	2...
19269	Zeit		- 27.07.2020 06:06:00	

Schnelle Information zu einer Singledestination

Zuweisung, Ausführung und zugehöriges Programm

tempotokoll Debugger			
ISE_ID	Name	TypeName	Information
19718	PrgRule	RULE	00
20119	Condition0	CONDITION	00.00
20255	SC1	SINGLECONDITION	00.00.00
20121	SC1	SINGLECONDITION	00.00.01
19719	RuleDestination	DESTINATION	00.01
20256	SDO	SINGLEDESTINATION	00.01.00
SingleDestination zu Programm "Baxxytestprogramm"			
Ausführung sofort			
Gerät "TEST" := Gerät "TestWTH-3:1.SET_POINT_TEMPERATURE"			

Hilfe zu Typaufschlüsselungen:

Script Developer V4.01.08J LCL [Unlock Licence by Black] [Registriert für Black]			
Selektiert Syntaxcheck Methodenhilfe Backup			
Methode	Typ	Value	Spezial
ID	integer	20256	ISE_ID 19718 PrgRule
Name	string	SDO	20119 Condition0
Type	integer	6291457	20255 SC1
TypeName	string	SIN Objekt : SDO	
Enabled	boolean	tru	trtypenname: SINGLEDESTINATION
Used	boolean	tru	trt-----
Internal	boolean	fal	OT_OBJECT 0000 0000 0000 0000 0000 0000 0000 0001
Visible	boolean	true	OT_SINGLEDESTINATION 0000 0110 0000 0000 0000 0000 0000 0001
Unerasable	boolean	false	
EnumMetaData	string		
R DestinationC...	integer	20225	TestWTH-3:1
		20225 CHANNEL	
R DestinationDP	integer	2651	TEST
		2651 VARDP	

Bestimmung von Binärmasken bei virtuellen Kanälen

Zu bearbeitende Kanäle markieren und dann mit der Maus über einen der Dimmer Virtual gibt einem:

Den Binärwert des Kanals über dem die Mouse steht (Mouseover, hier der 8. virtuelle Kana) Bzw die Binärmasken zu den markierten Kanälen (der SDV berücksichtigt hier aber auch nur die virtuellen Kanäle)

Systemprotokoll Debugger					
ISE_ID	Name	TypeName	Information	Type	Inter
16780	SB_H_GästeWC:0	CHANNEL	HmIP-BSL	MAINTENANCE	
16805	SB_H_GästeWC:1	CHANNEL	HmIP-BSL	KEY_TRANSCEIVER	
16808	SB_H_GästeWC:2	CHANNEL	HmIP-BSL	KEY_TRANSCEIVER	
16811	SB_H_GästeWC:3	CHANNEL	HmIP-BSL	SWITCH_TRANSMITTER	
16816	SB_H_GästeWC:4	CHANNEL	HmIP-BSL	SWITCH_VIRTUAL_...	
16822	SB_H_GästeWC:5	CHANNEL	HmIP-BSL	SWITCH_VIRTUAL_...	
16828	SB_H_GästeWC:6	CHANNEL	HmIP-BSL	SWITCH_VIRTUAL_...	
16834	SB_H_GästeWC:7	CHANNEL	HmIP-BSL	DIMMER_TRANSMITTER	
16843	SB_H_GästeWC:8	CHANNEL	HmIP-BSL	DIMMER_VIRTUAL_...	
16856	SB_H_GästeWC:9	CHANNEL	HmIP-BSL	DIMMER_VIRTUAL_...	
16869	SB_H_GästeWC:10	CHANNEL	HmIP-BSL	DIMMER_VIRTUAL_...	
16882	SB_H_GästeWC:11	CHANNEL	HmIP-BSL	DIMMER_TRANSMITTER	
16891	SB_H_GästeWC:12	CHANNEL	HmIP-BSL	DIMMER_VIRTUAL_...	
16904	SB_H_GästeWC:13	CHANNEL	HmIP-BSL	DIMMER_VIRTUAL_...	
16917	SB_H_GästeWC:14	CHANNEL	HmIP-BSL	DIMMER_VIRTUAL_...	
16930	SB_H_GästeWC:15	CHANNEL	HmIP-BSL	DIMMER_Virtuelle Kanalmasken	

MouseOver : 128 (8. virtueller Kanal)
 Selektiert: 200

Devices
 Interfaces
 Alle BidCos-F
 VirtualDevices HmIP-RF
 CUxD

Enthält

Einstellbar ist auch, ob in der Detailview Hilfetexte zu den Methoden eingeblendet werden sollten

The screenshot shows a software interface with a toolbar on the left containing various icons. A table lists properties with their types and values. A specific row for the 'AlType' method is highlighted with a yellow background, showing its documentation. The bottom status bar indicates the CCU IP, authentication type, and FW version.

EnumMetaData	string	BV0, BV1, UNIT
	BV0	nicht ausgelöst
	BV1	ausgelöst
	UNIT	
*	DPInfo	string
	State	string
	AIState	integer
	AICounter	integer
*	AIType	integer
R	ALT-----	integer
R	A1D -----	65535
R	ALT Zugehörigkeit zu Objectclasses:	65535
	A1E OT_ALARMDP	
A1A -----		0
ALT	Zugriffsart: Read	0
	.AIType () --> integer	0
ALR	Zugriffsart: Write	0
	.AIType (AValue: integer) --> boolean	

	Alarmtyp (Konstanten atxxx)	
	- 0: atGeneric	
	- 1: atEmergency	
	- 2: atFire	
	- 3: atBurglary	
	- 4: atSystem	
	- 5: atService	

CCU: 192.168.2.19 [] HTTPS - AUTH FW: 3.61.5.

4.19 Ghost Objekte (Bezüge in Aufzählungen auf Objektleichen)

Ghost Objekte wurden im Inspektor früher weggefiltert. Gab dazu ja nix anzuzeigen. erweis sich als verbesserungswürdig. früher ergab die Direkt Eingabe einer ID eines nicht existierenden Objektes ein leeres Listenfeld im Inspektor. Die neue Version stellt ein solches Objekt nun in einer roten Zeile mit der ID, dem Namen Ghost und dem Typ -- NULL -- dar.

Systemprotokoll				
Spezial	ISE_ID	Name	TypeName	Information
	4245	GHOST	-- NULL --	

klingt erstmal nach wenig nützlich...

Aber.

wir machen mal absichtlich die Aufzählung ID_SYSTEM_VARIABLES kaputt, in dem wir dort einfach mal einen Ghost einhängen. Also den case, den ich letztens mal bei einem hatte, die Middleware stürzte beim Synchronisieren ab.

gefunden wird dieses beim valid Enum Check

```
Überprüfung valide Objekte in Aufzählungen
Erstellt vom SDV V4.06.12G LCL am 28.04.2021 08:55:55

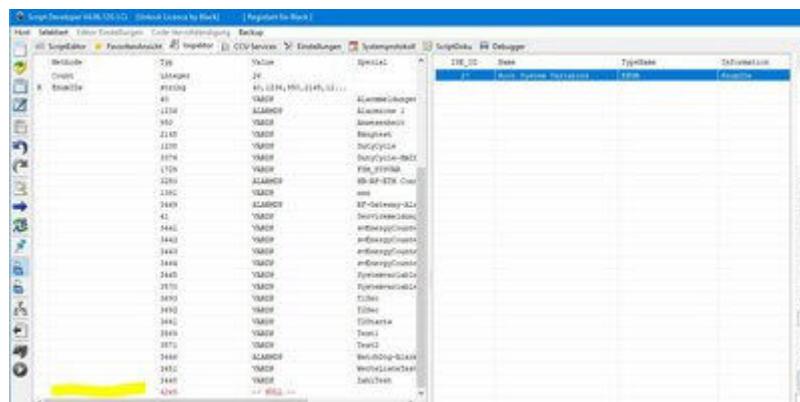
Überprüfen Aufzählung "ID_SYSTEM_VARIABLES" (26 Einträge)
ID {4245} ist eine Leiche in "ID_SYSTEM_VARIABLES", kann entfernt werden.
Überprüfen Aufzählung "ID_CHANNELS" (163 Einträge)
Überprüfen Aufzählung "ID_DEVICES" (9 Einträge)
Überprüfen Aufzählung "ID_INTERFACES" (4 Einträge)
Überprüfen Aufzählung "ID_PROGRAMS" (14 Einträge)
Überprüfen Aufzählung "ID_USERS" (7 Einträge)
Überprüfen Aufzählung "ID_FAVORITES" (6 Einträge)
Überprüfen Aufzählung "ID_ROOMS" (11 Einträge)
Überprüfen Aufzählung "ID_FUNCTIONS" (10 Einträge)
Überprüfen Aufzählung "ID_STRUCTURE" (8 Einträge)
Überprüfen Aufzählung "ID_HISTORYDPS" (4 Einträge)
Überprüfen Aufzählung "ID_DATAPOINTS" (710 Einträge)
Überprüfen Aufzählung "ID_SERVICES" (15 Einträge)
Überprüfen Aufzählung "ID_CALENDARDPS" (4 Einträge)
Überprüfen Aufzählung "ID_RULES" (13 Einträge)
Überprüfen Aufzählung "ID_CONDITIONS" (30 Einträge)
Überprüfen Aufzählung "ID_SCONDITIONS" (41 Einträge)
Überprüfen Aufzählung "ID_DESTINATIONS" (0 Einträge)
Überprüfen Aufzählung "ID_SDESTINATIONS" (49 Einträge)
Überprüfen Aufzählung "ID_SCENES" (16 Einträge)

Gefundene Fehler : 1
Davon korrigiert : 0
```

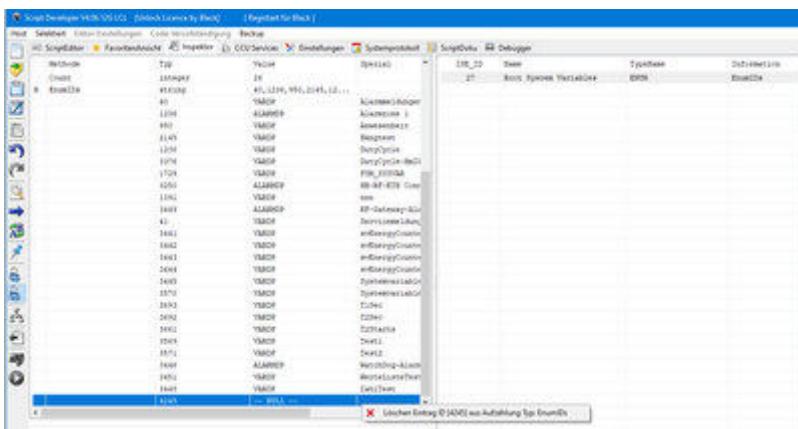
Für erfahrene Nutzer wären nun die weiteren Schritte mit ein paar zusätzlichen tests klar gewesen. Ums nun etwas zu vereinfachen sind die neuen Funktionalitäten nun da. Der Ghosts hängt in ID_SYSTEM_VARIABLES, also erstmal da schauen:

ISE_ID	Name	TypeName	Information
40	Alarmmeldungen	VARDP	0
1236	Alarmzone 1	ALARMDP	
950	Anwesenheit	VARDP	true
2145	Bangtest	VARDP	Test1 Program
1238	DutyCycle	VARDP	0.000000
3376	DutyCycle-RmIB-HAP 0003...	VARDP	0.000000
1725	FSM_SYSVAR	VARDP	false
3250	HB-REF-ETH Connection	ALARMDP	
1391	nmm	VARDP	true
3449	RF-Gateway-Alarm	ALARMDP	
41	Servicemeldungen	VARDP	2
3441	svEnergyCounter_2240_00...	VARDP	0.000000
3442	svEnergyCounter_2626_00...	VARDP	0.000000
3443	svEnergyCounterOldVal_2240	VARDP	0.100000
3444	svEnergyCounterOldVal_2626	VARDP	0.000000
3445	Systemvariable	VARDP	false
3570	Systemvariable 1	VARDP	0.000000
3693	T1Sec	VARDP	47.000000
3692	T2Sec	VARDP	47.000000
3661	T2Starts	VARDP	5.000000
3569	Test1	VARDP	1.000000
3571	Test2	VARDP	2.000000
3446	WatchDog-Alarm	ALARMDP	
3451	WerteListeTest	VARDP	0
3448	ZahlTest	VARDP	1.000000
4245	GHOST	-- NULL --	

die alten Versionen stellten den Ghost nicht dar. hier wie beschrieben: in rot markiert. Nun das Object anklicken, es erfolgt natürlich keine Detailansicht, weil es kein existierendes Objekt ist. rechte Maustaste und suche Referenzbezüge in regadom. Als Ergebnis kommt dann in diesem Fall nur eine Verwendung in ID27, root system variables, heraus, diese nun anklicken und in der Aufzählung erscheint im Detailfeld nun auch der Ghost in rot.



die Kindersicherung rausmachen (Schloss auf) rechte Maustaste auf den Ghost ergibt ein neues Menü



Anhand des Objecttypes des Basisobjectes und der Bezeichnung des IDarrays wählt der SDV die zum entfernen nötige Löschmethode aus und eliminiert das angewählte Object aus der Aufzählung. (nach Bestätigung !)

Mit dieser Vorgehensweise sollten sich nun auch für nicht ganz erfahrene Nutzer Schrott Einträge in Aufzählungen finden, prüfen und auch eliminieren lassen.

4.20 Master und Linksets

Der SDV kann mit Mastersets und auch Linksets umgehen. Mastersets werden automatisch dargestellt, wenn ein Gerät bzw Kanal betrachtet wird, der Mastersets enthält.

Bei Master und bei Linksets in der Anzeige nun komplett Informationen über den Parameter im Hint, auch wenn dieser eine Aufzählung ist

ChDPUsageCount	1434	BS5DP	RmIP-RF.002598E995BCC2::	1410	Hm
ChLinkCount	1435			1411	Hm
ChRoom	1436			1412	Hm
ChFunction	1437			1413	Hm
DPE	1438	1434,1435,143...		1414	Hm
	1439	BS5DP	RmIP-RF.002598E995BCC2::	1415	Hm
	1440	BS5DP	RmIP-RF.002598E995BCC2::	1416	Hm
	1441	BS5DP	RmIP-RF.002598E995BCC2::	1417	Hm
	1442	BS5DP	RmIP-RF.002598E995BCC2::	1418	Hm
	1443	BS5DP	RmIP-RF.002598E995BCC2::	1419	Hm
	1444	BS5DP	RmIP-RF.002598E995BCC2::	1420	Hm
	1445	BS5DP	RmIP-RF.002598E995BCC2::	1421	Hm
	1446	BS5DP	RmIP-RF.002598E995BCC2::	1422	Hm
	1447	BS5DP	RmIP-RF.002598E995BCC2::	1423	Hm
	1448	BS5DP	RmIP-RF.002598E995BCC2::	1424	Hm
	1449	BS5DP	RmIP-RF.002598E995BCC2::	1425	Hm
	1450	BS5DP	RmIP-RF.002598E995BCC2::	1426	Hm
	1451	BS5DP	RmIP-RF.002598E995BCC2::	1427	Hm
	1452	BS5DP	RmIP-RF.002598E995BCC2::	1428	Hm
	1453	BS5DP	RmIP-RF.002598E995BCC2::	1429	Hm
	1454	BS5DP	RmIP-RF.002598E995BCC2::	1430	Hm
	1455	BS5DP	RmIP-RF.002598E995BCC2::	1431	Hm
	1456	BS5DP	RmIP-RF.002598E995BCC2::	1432	Hm
	1457	BS5DP	RmIP-RF.002598E995BCC2::	1433	Hm
	1458	BS5DP	RmIP-RF.002598E995BCC2::	1434	Hm
	1459	BS5DP	RmIP-RF.002598E995BCC2::	1435	Hm
	1460	BS5DP	RmIP-RF.002598E995BCC2::	1436	Hm
	1461	BS5DP	RmIP-RF.002598E995BCC2::	1437	Hm
	1462	BS5DP	RmIP-RF.002598E995BCC2::	1438	Hm
	1463	BS5DP	RmIP-RF.002598E995BCC2::	1439	Hm
	1464	BS5DP	RmIP-RF.002598E995BCC2::	1440	Hm
	1465	BS5DP	RmIP-RF.002598E995BCC2::	1441	Hm
	1466	BS5DP	RmIP-RF.002598E995BCC2::	1442	Hm
	1467	BS5DP	RmIP-RF.002598E995BCC2::	1443	Hm
	1468	BS5DP	RmIP-RF.002598E995BCC2::	1444	Hm
	1469	BS5DP	RmIP-RF.002598E995BCC2::	1445	Hm
	1470	BS5DP	RmIP-RF.002598E995BCC2::	1446	Hm
	1471	BS5DP	RmIP-RF.002598E995BCC2::	1447	Hm
	1472	BS5DP	RmIP-RF.002598E995BCC2::	1448	Hm
	1473	BS5DP	RmIP-RF.002598E995BCC2::	1449	Hm
	1474	BS5DP	RmIP-RF.002598E995BCC2::	1450	Hm
	1475	BS5DP	RmIP-RF.002598E995BCC2::	1451	Hm
	1476	BS5DP	RmIP-RF.002598E995BCC2::	1452	Hm
	1477	BS5DP	RmIP-RF.002598E995BCC2::	1453	Hm
	1478	BS5DP	RmIP-RF.002598E995BCC2::	1454	Hm
	1479	BS5DP	RmIP-RF.002598E995BCC2::	1455	Hm
	1480	BS5DP	RmIP-RF.002598E995BCC2::	1456	Hm
	1481	BS5DP	RmIP-RF.002598E995BCC2::	1457	Hm
	1482	BS5DP	RmIP-RF.002598E995BCC2::	1458	Hm
	1483	BS5DP	RmIP-RF.002598E995BCC2::	1459	Hm
	1484	BS5DP	RmIP-RF.002598E995BCC2::	1460	Hm
	1485	BS5DP	RmIP-RF.002598E995BCC2::	1461	Hm
	1486	BS5DP	RmIP-RF.002598E995BCC2::	1462	Hm
	1487	BS5DP	RmIP-RF.002598E995BCC2::	1463	Hm
	1488	BS5DP	RmIP-RF.002598E995BCC2::	1464	Hm
	1489	BS5DP	RmIP-RF.002598E995BCC2::	1465	Hm
	1490	BS5DP	RmIP-RF.002598E995BCC2::	1466	Hm
	1491	BS5DP	RmIP-RF.002598E995BCC2::	1467	Hm
	1492	BS5DP	RmIP-RF.002598E995BCC2::	1468	Hm
	1493	BS5DP	RmIP-RF.002598E995BCC2::	1469	Hm
	1494	BS5DP	RmIP-RF.002598E995BCC2::	1470	Hm
	1495	BS5DP	RmIP-RF.002598E995BCC2::	1471	Hm
	1496	BS5DP	RmIP-RF.002598E995BCC2::	1472	Hm
	1497	BS5DP	RmIP-RF.002598E995BCC2::	1473	Hm
	1498	BS5DP	RmIP-RF.002598E995BCC2::	1474	Hm
	1499	BS5DP	RmIP-RF.002598E995BCC2::	1475	Hm
	1500	BS5DP	RmIP-RF.002598E995BCC2::	1476	Hm
	1501	BS5DP	RmIP-RF.002598E995BCC2::	1477	Hm
	1502	BS5DP	RmIP-RF.002598E995BCC2::	1478	Hm
	1503	BS5DP	RmIP-RF.002598E995BCC2::	1479	Hm
	1504	BS5DP	RmIP-RF.002598E995BCC2::	1480	Hm
	1505	BS5DP	RmIP-RF.002598E995BCC2::	1481	Hm
	1506	BS5DP	RmIP-RF.002598E995BCC2::	1482	Hm
	1507	BS5DP	RmIP-RF.002598E995BCC2::	1483	Hm
	1508	BS5DP	RmIP-RF.002598E995BCC2::	1484	Hm
	1509	BS5DP	RmIP-RF.002598E995BCC2::	1485	Hm
	1510	BS5DP	RmIP-RF.002598E995BCC2::	1486	Hm
	1511	BS5DP	RmIP-RF.002598E995BCC2::	1487	Hm
	1512	BS5DP	RmIP-RF.002598E995BCC2::	1488	Hm
	1513	BS5DP	RmIP-RF.002598E995BCC2::	1489	Hm
	1514	BS5DP	RmIP-RF.002598E995BCC2::	1490	Hm
	1515	BS5DP	RmIP-RF.002598E995BCC2::	1491	Hm
	1516	BS5DP	RmIP-RF.002598E995BCC2::	1492	Hm
	1517	BS5DP	RmIP-RF.002598E995BCC2::	1493	Hm
	1518	BS5DP	RmIP-RF.002598E995BCC2::	1494	Hm
	1519	BS5DP	RmIP-RF.002598E995BCC2::	1495	Hm
	1520	BS5DP	RmIP-RF.002598E995BCC2::	1496	Hm
	1521	BS5DP	RmIP-RF.002598E995BCC2::	1497	Hm
	1522	BS5DP	RmIP-RF.002598E995BCC2::	1498	Hm
	1523	BS5DP	RmIP-RF.002598E995BCC2::	1499	Hm
	1524	BS5DP	RmIP-RF.002598E995BCC2::	1500	Hm
	1525	BS5DP	RmIP-RF.002598E995BCC2::	1501	Hm
	1526	BS5DP	RmIP-RF.002598E995BCC2::	1502	Hm
	1527	BS5DP	RmIP-RF.002598E995BCC2::	1503	Hm
	1528	BS5DP	RmIP-RF.002598E995BCC2::	1504	Hm
	1529	BS5DP	RmIP-RF.002598E995BCC2::	1505	Hm
	1530	BS5DP	RmIP-RF.002598E995BCC2::	1506	Hm
	1531	BS5DP	RmIP-RF.002598E995BCC2::	1507	Hm
	1532	BS5DP	RmIP-RF.002598E995BCC2::	1508	Hm
	1533	BS5DP	RmIP-RF.002598E995BCC2::	1509	Hm
	1534	BS5DP	RmIP-RF.002598E995BCC2::	1510	Hm
	1535	BS5DP	RmIP-RF.002598E995BCC2::	1511	Hm
	1536	BS5DP	RmIP-RF.002598E995BCC2::	1512	Hm
	1537	BS5DP	RmIP-RF.002598E995BCC2::	1513	Hm
	1538	BS5DP	RmIP-RF.002598E995BCC2::	1514	Hm
	1539	BS5DP	RmIP-RF.002598E995BCC2::	1515	Hm
	1540	BS5DP	RmIP-RF.002598E995BCC2::	1516	Hm
	1541	BS5DP	RmIP-RF.002598E995BCC2::	1517	Hm
	1542	BS5DP	RmIP-RF.002598E995BCC2::	1518	Hm
	1543	BS5DP	RmIP-RF.002598E995BCC2::	1519	Hm
	1544	BS5DP	RmIP-RF.002598E995BCC2::	1520	Hm
	1545	BS5DP	RmIP-RF.002598E995BCC2::	1521	Hm
	1546	BS5DP	RmIP-RF.002598E995BCC2::	1522	Hm
	1547	BS5DP	RmIP-RF.002598E995BCC2::	1523	Hm
	1548	BS5DP	RmIP-RF.002598E995BCC2::	1524	Hm
	1549	BS5DP	RmIP-RF.002598E995BCC2::	1525	Hm
	1550	BS5DP	RmIP-RF.002598E995BCC2::	1526	Hm
	1551	BS5DP	RmIP-RF.002598E995BCC2::	1527	Hm
	1552	BS5DP	RmIP-RF.002598E995BCC2::	1528	Hm
	1553	BS5DP	RmIP-RF.002598E995BCC2::	1529	Hm
	1554	BS5DP	RmIP-RF.002598E995BCC2::	1530	Hm
	1555	BS5DP	RmIP-RF.002598E995BCC2::	1531	Hm
	1556	BS5DP	RmIP-RF.002598E995BCC2::	1532	Hm
	1557	BS5DP	RmIP-RF.002598E995BCC2::	1533	Hm
	1558	BS5DP	RmIP-RF.002598E995BCC2::	1534	Hm
	1559	BS5DP	RmIP-RF.002598E995BCC2::	1535	Hm
	1560	BS5DP	RmIP-RF.002598E995BCC2::	1536	Hm
	1561	BS5DP	RmIP-RF.002598E995BCC2::	1537	Hm
	1562	BS5DP	RmIP-RF.002598E995BCC2::	1538	Hm
	1563	BS5DP	RmIP-RF.002598E995BCC2::	1539	Hm
	1564	BS5DP	RmIP-RF.002598E995BCC2::	1540	Hm
	1565	BS5DP	RmIP-RF.002598E995BCC2::	1541	Hm
	1566	BS5DP	RmIP-RF.002598E995BCC2::	1542	Hm
	1567	BS5DP	RmIP-RF.002598E995BCC2::	1543	Hm
	1568	BS5DP	RmIP-RF.002598E995BCC2::	1544	Hm
	1569	BS5DP	RmIP-RF.002598E995BCC2::	1545	Hm
	1570	BS5DP	RmIP-RF.002598E995BCC2::	1546	Hm
	1571	BS5DP	RmIP-RF.002598E995BCC2::	1547	Hm
	1572	BS5DP	RmIP-RF.002598E995BCC2::	1548	Hm
	1573	BS5DP	RmIP-RF.002598E995BCC2::	1549	Hm
	1574	BS5DP	RmIP-RF.002598E995BCC2::	1550	Hm
	1575	BS5DP	RmIP-RF.002598E995BCC2::	1551	Hm
	1576	BS5DP	RmIP-RF.002598E995BCC2::	1552	Hm
	1577	BS5DP	RmIP-RF.002598E995BCC2::	1553	Hm
	1578	BS5DP	RmIP-RF.002598E995BCC2::	1554	Hm
	1579	BS5DP	RmIP-RF.002598E995BCC2::	1555	Hm
	1580	BS5DP	RmIP-RF.002598E995BCC2::	1556	Hm
	1581	BS5DP	RmIP-RF.002598E995BCC2::	1557	Hm
	1582	BS5DP	RmIP-RF.002598E995BCC2::	1558	Hm
	1583	BS5DP	RmIP-RF.002598E995BCC2::	1559	Hm
	1584	BS5DP	RmIP-RF.002598E995BCC2::	1560	Hm
	1585	BS5DP	RmIP-RF.002598E995BCC2::	1561	Hm
	1586	BS5DP	RmIP-RF.002598E995BCC2::	1562	Hm
	1587	BS5DP	RmIP-RF.002598E995BCC2::	1563	Hm
	1588	BS5DP	RmIP-RF.002598E995BCC2::	1564	Hm
	1589	BS5DP	RmIP-RF.002598E995BCC2::	1565	Hm
	1590	BS5DP	RmIP-RF.002598E995BCC2::	1566	Hm
	1591	BS5DP	RmIP-RF.002598E995BCC2::	1567	Hm
	1592	BS5DP	RmIP-RF.002598E995BCC2::	1568	Hm
	1593	BS5DP	RmIP-RF.002598E995BCC2::	1569	Hm
	1594	BS5DP	RmIP-RF.002598E995BCC2::	1570	Hm
	1595	BS5DP	RmIP-RF.002598E995BCC2::	1571	Hm
	1596	BS5DP	RmIP-RF.002598E995BCC2::	1572	Hm
	1597	BS			

Um gerade für „Nicht-Profis“ derartige Änderungen einfacher zumachen, unterstützt der SDV hierbei, um Bedienfehler bestmöglich auszuschliessen.

The screenshot shows a configuration table with various parameters. One parameter, **P1_ENDTIME_FRIDAY_1**, is highlighted. Below the table, a dropdown menu is open, showing a list of valid values. The value **360** is selected, and a tooltip indicates the valid range is **Gültiger Bereich: 5 ... 1440**.

	MIN_MAX_VALUE_NO...	boolean	0
M	OPTIMUM_START_STOP	boolean	0
M	P1_ENDTIME_FRIDAY_1	i4	360
M	P1_ENDTIME_FRIDAY...	i4	1440
M	P1_ENDTIME_FRIDAY...	i4	1440
M	P1_ENDTIME_FRIDAY...	i4	1440
M	P1_ENDTIME_FRIDAY...	i4	1440
M	P1_ENDTIME_FRIDAY_2	i4	540
M	P1_ENDTIME_FRIDAY_3	i4	1020
M	P1_ENDTIME_FRIDAY_4	i4	1320
M	P1_ENDTIME_FRIDAY_5	i4	1440
M	P1_ENDTIME_FRIDAY_6	i4	1440
M	P1_ENDTIME_FRIDAY_7	i4	1440
M	P1_ENDTIME_FRIDAY_8	i4	1440
M	P1_ENDTIME_FRIDAY_9	i4	1440
M	P1_ENDTIME_MONDAY_1	i4	360

[IselID: 4615 - HmIP-eTRV-2 000A18A9945475:1]

P1_ENDTIME_FRIDAY_1

360

Gültiger Bereich: 5 ... 1440

CCU: 192.168.2.19 [] HTTPS - AUTH FW: 3.59.6.20211009 NewScript4.hsc

Richtiger Typ und von der Firmware gesetzter Wertebereich werden vorgegeben und überwacht. Bzw bei Aufzählungen wird der aufgelöste Text der Werteliste in einer Combobox vorgeschlagen.

The screenshot shows a configuration table with various parameters. One parameter, **MANU_MODE_PRIORITIZATION**, is highlighted. Below the table, a dropdown menu is open, showing a list of valid values. The value **1 = SET_TEMPERATURE_CHANGE_BY_ALL** is selected.

	DECALCIFICATION_...	i4	22
M	DECALCIFICATION_...	i4	6 (SATU
M	DURATION_5MIN	i4	0
M	MANU_MODE_PRIORI...	i4	1 (SET_
M	MIN_MAX_VALUE_NO...	boolean	0
M	OPTIMUM_START_STOP	boolean	0
M	P1_ENDTIME_FRIDAY_1	i4	360

[IselID: 4615 - HmIP-eTRV-2 000A18A9945475:1]

MANU_MODE_PRIORITIZATION

1 = SET_TEMPERATURE_CHANGE_BY_ALL
0 = SET_TEMPERATURE_CHANGE_ONLY_BY_RT_TC_SC_SELF
1 = SET_TEMPERATURE_CHANGE_BY_ALL
2 = SET_TEMPERATURE_CHANGE_ONLY_BY_RT_TC_CCU_SELF
3 = SET_TEMPERATURE_CHANGE_ONLY_BY_CCU
4 = SET_TEMPERATURE_CHANGE_ONLY_BY_SELF
5 = SET_TEMPERATURE_CHANGE_BY_NONE

CCU: 1 script4.

4.20.1 Linksets

Direktverknüpfungen lassen sich auf diese Art und Weise genauso Auflösen und verändern. Zu einigen der Parameter (bin noch nicht fertig) existieren auch schon Beschreibungen in den Hints

The screenshot shows the 'ScriptDeveloper V4.08.03A-HF2 LCL' interface. The 'ScriptEditor' tab is active. A table lists parameters under the 'PARAMSET_LINK' section:

Methode	Typ	Value	Spezial ^	Name	Beschreibung	Empfänger	Sender
DEVICENAME	HmIP-DRS14	00259BE9...		3014F711A0...	NO_DESCRIPTION	0001D3C992ABEE:3 (...)	0001D3C99...
DEVICETYPE	MULTI_MODE_INPUT_TR...			3014F711A0...	NO_DESCRIPTION	00259BE995BCC2:6 (...)	00259BE99...
SERIENNUMMER	00259BE995BCC2:4			3014F711A0...	NO_DESCRIPTION	00259BE995BCC2:10 (...)	00259BE99...
PARAMSET_LINK	string	44 Parameter		3014F711A0...	NO_DESCRIPTION	00259BE995BCC2:14 (...)	00259BE99...
	LONG_COND_VALUE_HI	14	100	3014F711A0...	NO_DESCRIPTION	00259BE995BCC2:18 (...)	00259BE99...

Below the table, a tooltip provides information about the 'LONG_COND_VALUE_HI' parameter:

LONG_COND_V Informationen zum Parameter "LONG_COND_VALUE_HI"
LONG_CT_OFS-----
LONG_CT_OFF Type : INTEGER
LONG_CT_OFF Flags : 1 (Visible)
LONG_CT_ON Value : 100
LONG_CT_ONE
LONG_JT_OFF Min Value: 0
LONG_JT_OFF Max Value: 255
LONG_JT_ON Default : 150
LONG_JT_ONE LONG_COND_VALUE_HI
LONG_MULTIE-----
LONG_OFF_TI Tastendruck-long_Bedingungs (Vergleichs) Wert OBEN
LONG_OFF_TI Der Wert ist der obere/höhe/HI Wert, mit dem ein (vom Sender mitgesender) Wert verglichen wird.
LONG_OFF_TI Mit diesen Werten werden die u.U. vom Sender zusätzlich mit dem Triggerbefehl (Auslöser) mitgesandten Informationen, unter Anwendung der unten aufgeführten Vergleichsoperationen ..._CT,... in jedem Profilabschnitt neu verglichen.
LONG_OFF_DEI Diese Parameter sind nur relevant wenn ein Sender (z.B. Bewegungsmelder,Thermostat) eine Zusatzinformation/Wert zu dem Triggerbefehl (Auslöser) mit sendet. Andernfalls wird dieser ignoriert.

Die Werte jedes Parameters werden auch hier dargestellt, der Parameter kann wie bei den Mastersets beschrieben, auch geändert werden, oder wie im Editor beschrieben automatisiert in ein Skript zum setzen bzw. auslesen verwendet werden.

The screenshot shows the 'ScriptDeveloper V4.08.03A-HF2 LCL' interface. The 'ScriptEditor' tab is active. A table lists parameters under the 'PARAMSET_LINK' section:

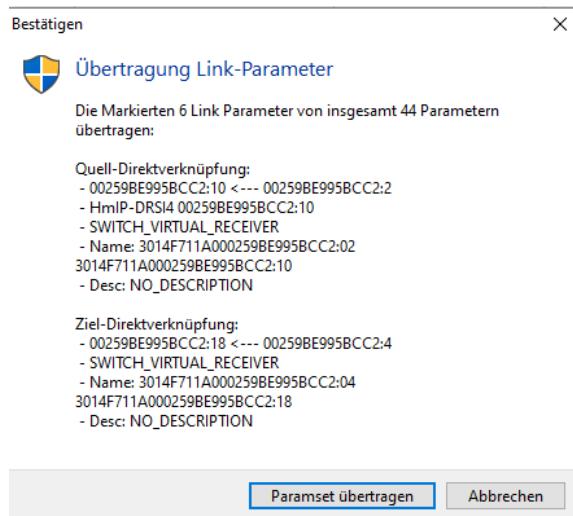
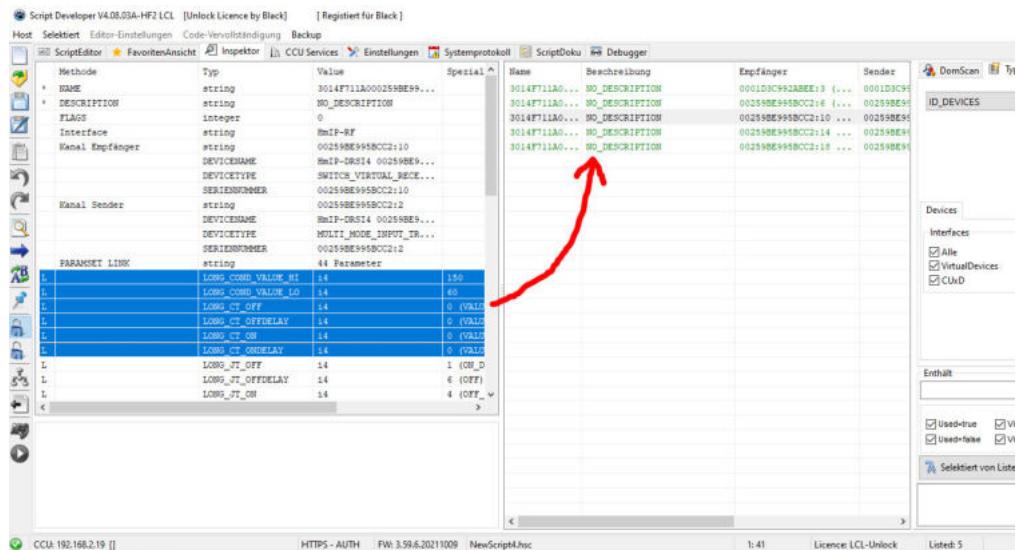
L	LONG_CI_UNDELAY	14	U (VALU
L	LONG_JT_OFF	14	1 (ON_D
L	LONG_JT_OFFDELAY	14	6 (OFF)
L	LONG_JT_ON	14	4 (OFF_
L	LONG_JT_ONDELAY	14	3 (ON)
L	LONG_MULTIEEXECUTE	boolean	0
L	LONG_OFF_TIME_BASE	i4	7 (BASE
L	LONG_OFF_TIME_FA...	14	31
L	LONG_OFF_TIME_MODE	i4	0 (TTMR

A tooltip for the 'LONG_OFF_TIME_FACTOR' parameter is displayed:

LONG_O Informationen zum Parameter "LONG_OFF_TIME_FACTOR"
LONG_O-----
LONG_O Type : INTEGER
LONG_O Flags : 1 (Visible)
LONG_O Value : 31
Eigenschaften Direktverknüpfung
LONG_OFF_TIME_FACTOR

Einer der häufigen Anwendungsfälle bei mir ist Teile eines Linksets von der angewählten Direktverknüpfung in eine andere Direktverknüpfung zu verschieben.

In der Quell Direktverknüpfung werden die Parameter markiert. Mit Druck linke Maustaste und bewegen der Maus färben sich die DVs in der Listenansicht, in die die Parameter verschoben werden, nun grün. Mit der Maus kann nun via Drag and Drop die Ziel DV ausgewählt werden und mit Loslassen der Maustaste erfolgt der Transferprozess. Es erfolgt aber noch eine Sicherheitsabfrage.



Nach Druck auf „Paramset übertragen“ werden die ausgewählten Parameter von der QuellDV in die ZielDV übertragen. Sollten Parameter in der ZielDV nicht übertragen, so erfolgt eine Meldung über den nicht übertragenen Parameter.

Will man nur alle LONG, Alle SHOTZ oder absolut alle Parameter übertragen so gibt es die schnelle Markiermöglichkeit. Rechte Maustaste in die Detailansicht der Direktverknüpfung rechte Maustaste, dann die Markierung auswählen:

The screenshot shows the 'ScriptDeveloper V4.08.03A-HF2 LCL' interface. The 'ScriptEditor' tab is active. A context menu is open over a table row containing the parameter 'LONG_CT_OFFDELAY'. The menu items are:

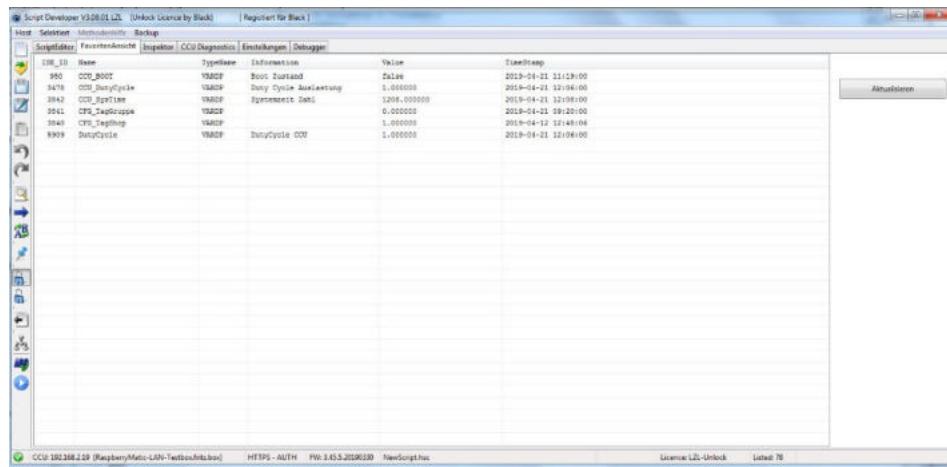
- Markiere Link-Parameter >
- Markiere alle ParamsetLinks
- Markiere alle LONG_ParamsetLinks
- Markiere alle SHORT_ParamsetLinks

Methode	Typ	Value	Spezial
DEVICENAME		HmIP-DRSI4 00259BE9...	
DEVICETYPE		SWITCH_VIRTUAL_RECE...	
SERIENNUMMER		00259BE995BCC2:10	
Kanal Sender	string	00259BE995BCC2:2	
	DEVICENAME	HmIP-DRSI4 00259BE9...	
	DEVICETYPE	MULTI_MODE_INPUT_TR...	
	SERIENNUMMER	00259BE995BCC2:2	
PARAMSET LINK	string	44 Parameter	
L	LONG_COND_VALUE_HI	i4	150
L	LONG_COND_VALUE_LO	i4	60
L	LONG_CT_OFF	i4	0 (VALU
L	LONG_CT_OFFDELAY	i4	0 (VALU
L	LONG_CT_ONDELAY	i4	
L	LONG_JT_OFF	i4	
L	LONG_JT_OFFDELAY	i4	
L	LONG_JT_ON	i4	4 (OFF_
L	LONG_JT_ONDELAY	i4	3 (ON)
L	LONG_MULTIEEXECUTE	boolean	0
L	LONG_OFF_TIME_BASE	i4	7 (BASE

Nach dem Markieren beginnt direkt der Drag and Drop Prozess, mögliche Ziele färben sich grün sie Maus kann bewegt werden. Abgebrochen wird mit Escape.

5 Favoritenansicht

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



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

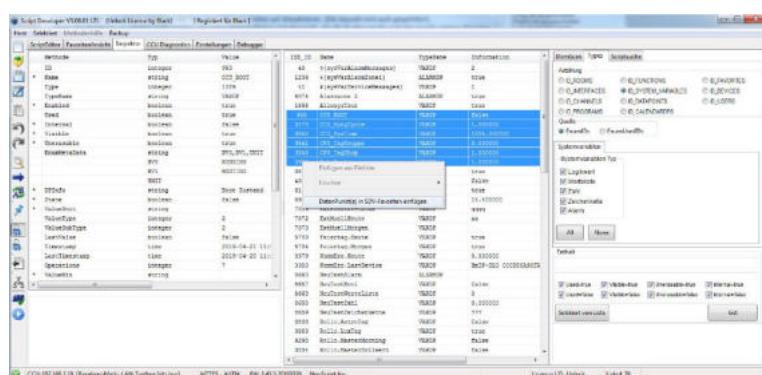
Will man einen Datenpunkt mal eben ändern, einen Doppelklick auf die Zeile der Favoriten, daraufhin wird unter Berücksichtigung des Undo / redo Stacks die Favoriten in die Listenansicht geladen und das angeklickte Objekt zur Bearbeitung in der Detailansicht geöffnet.

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

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

5.1 Hinzufügen von Objekten in die Favoriten

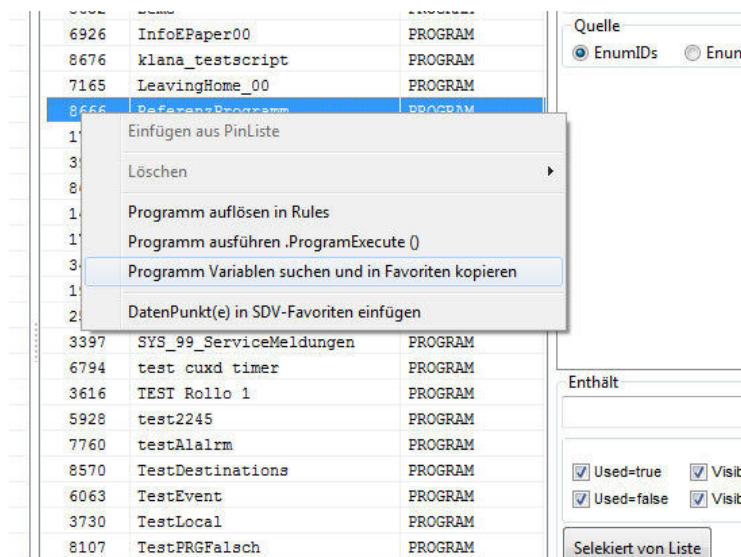
Objekte können aus der Listenansicht des Inspektors in die Favoriten geladen werden. Dazu Objekte Selektieren rechte Maustaste und



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

5.2 Verwendete Objekte eines Programmes in die Favoriten laden

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



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

5.3 Favoritenliste löschen

In die Favoriten reinklicken, rechte Maustaste – Favoritenliste löschen

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

5.4 Einschränkungen

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

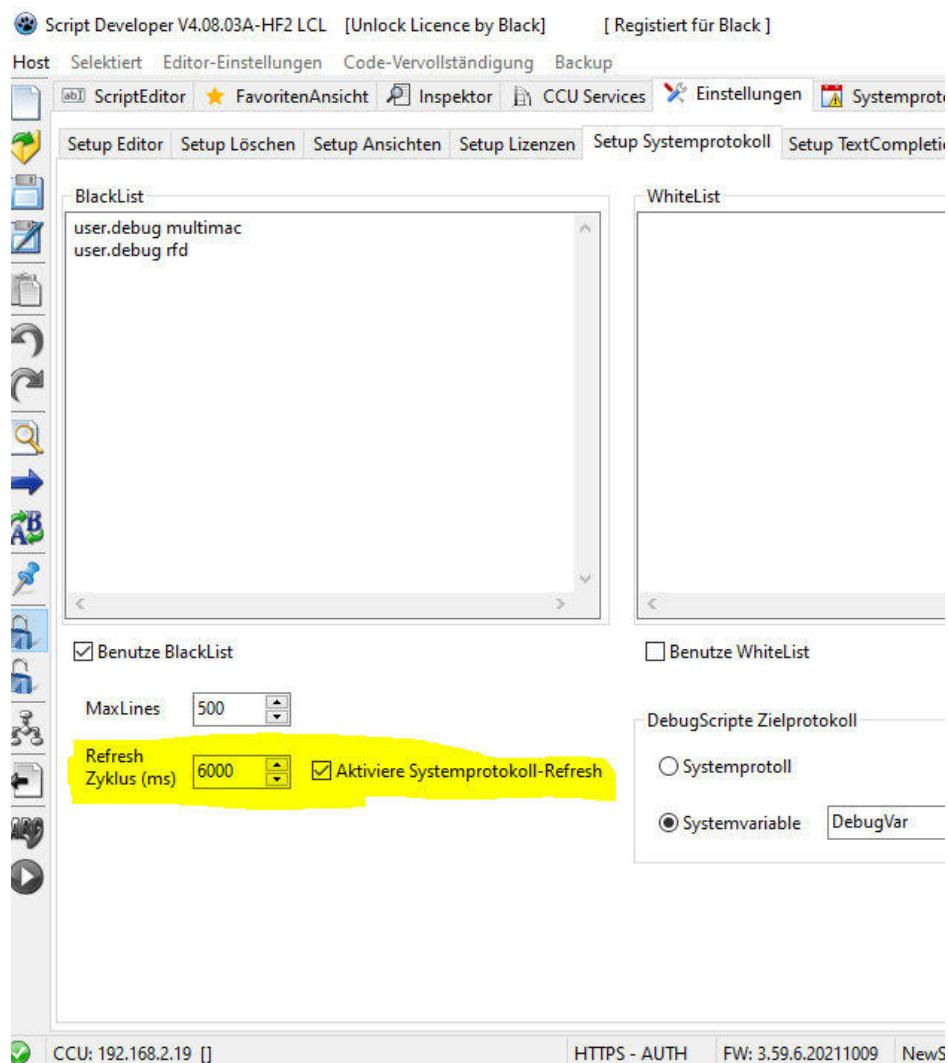
Beim Arbeiten mit 2 CCU's bedeutet das:

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

5.5 Dynamisches Auffrischen von Favoriten

Es kann automatisches aktualisieren des Systemprotokolls angewählt werden. Die Refresh Zeit ist ebenfalls einstellbar. Das Refreh findet nur statt, wenn auch das Systemprotokoll geöffnet ist. Ist automatischer Refesh nicht gewünscht, ist es deaktivierbar.

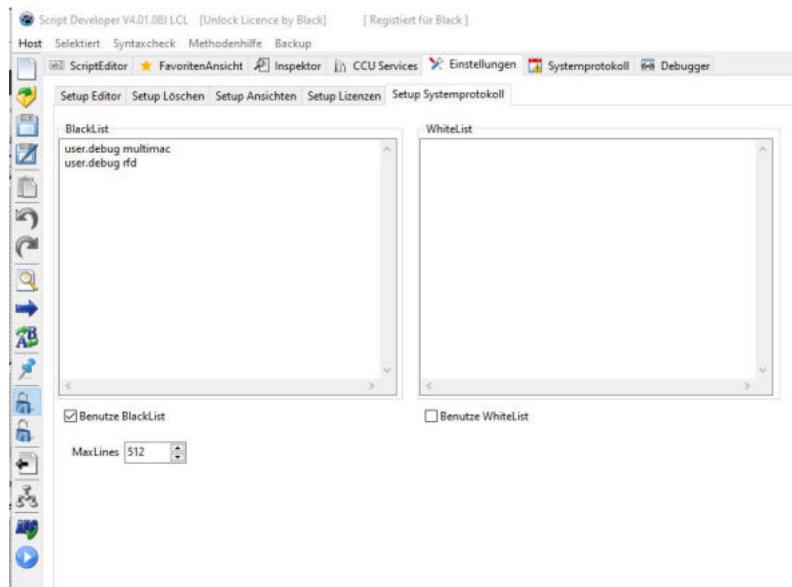
Manueller Refresh ist mit F5 möglich



6 Systemprotokoll

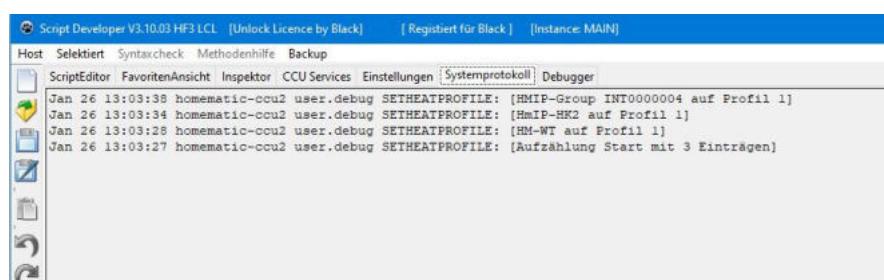
Wird der Reiter angeklickt, so wird das Systemprotokoll der CCU abgerufen in den SDV und dort aufbereitet:

- der aktuellste Eintrag steht immer oben.
- es gibt eine Blacklist, LogEinträge, deren Textausschnitte in der Blacklist vorkommen, werden ignoriert (so benutze Blacklist angehakt ist). Damit blende ich z.B. zum programmieren auf meinem Spielesystem Meldungen des Multimac aus.
- es gibt eine Whitelist. Wenn benutze Whitelist angehakt ist, so wird ein Logeintrag angezeigt, wenn Textausschnitte davon in der Whitelist vorhanden sind (muss vorkommen, sonst wird ignoriert)



Nicht ganz reproduzierbar gibt es manchmal Probleme bei der Übertragung sehr langer Texte, so dass sich die Maximale Zeilenlänge des Protokolls einstellen lässt.

Nützlich in dem Fall, wo Zeitverzögerte Aktionen und Einträge überwacht werden sollen. Die Skriptausgabe des Editors nimmt ja nur den Moment des Skriptausführens. Konkreter Anwendungsfall: das Skript zum zeitlich versetzten Umschalten von Wochenprofilen in verschiedenartigen Geräten:

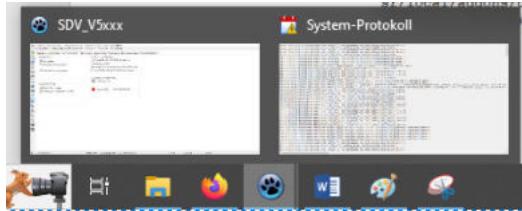


Folgende Schlüssel dazu in der INI nachtragen (bei Bestandsinstallationen)

```
[SYSLOG]
USEBLACKLIST=True
USEWHITELIST=False
BLACKLIST=user.debug multimap\r\nuser.info multimap\r\nuser.debug rfd
WHITELIST=
```

6.1 Undock und Dock des Protokollfensters

Wenn gewünscht (z.B. bei Verwendung von zwei Monitoren) kann der Reiter Systemprotokoll ausgedockt werden und in einem separaten, frei verschiebbaren Fenster dargestellt werden. Im ausgedockten Zustand werden in der Taskleiste auch beide Symbole dargestellt:



6.1.1 Undock

Undock löst das Systemprotokoll aus dem Reiter des SDV und stellt es in einem separaten Fenster dar. Dazu rechte Maustaste ins Systemprotokoll und Undock von SDV Page auswählen.

```
interface pointer are not init
SyntaxError: IDIFaces= interfa
Syn! Aktualisieren F5 8
# p4 ns/
# p4 Undock von SDV Page ns/
Losing fd(10) (127.0.0.1:37882
```

6.1.2 Dock

Dock löscht das Separate Fenster wieder und stellt das Systemprotokoll wieder in dem SDV Reiter dar.

Dazu gibt es 2 Möglichkeiten

1. Das separate Fenster einfach Schließen
2. In dem Systemprotokollfenster rechte Maustaste und Dock an SDV Page auswählen:

```
(127.0.0.1:37882) idle(2307ms
sr/local/addons/cuxd/cuxd.ps)
sr/i Aktualisieren F5 ps)
(127.0.0.1:37882) idle(277ms
sr/i Dock an SDV Page ps)
sr/local/addons/cuxd/cuxd.ps)
(127.0.0.1:41732) idle(28884ms
```

7 Kleine Helfer im Alltag

7.1 Debugging von WebUi Programmen

Mit der Version 3.11.02 wurden die ziemlich weitreichenden Test und Eingriffsmöglichkeiten auch für niedrigere Lizenzlevel freigeben. Da ich schon einige PN Anfragen bezüglich der Möglichkeiten dazu hatte, nun mal eine allgemeine Anleitung für die breite Masse. Die Stammtischmitglieder kannten diese Möglichkeiten ja schon seit langem.

Ansatz.

Man hat ein WebUI Programm geschrieben und möchte dieses nun testen in Realumgebung, also mit den Geräten etc. Wenn man nun Dinge testen möchte wie z.B Helligkeiten, Temperaturen, Luftfeuchtigkeit und vieles mehr war oft der weg, entweder warten bis sich ein entsprechendes Scenario einstellt, den Geräten mit Feuchtigkeit, Heißluft Fön etc zu Leibe rücken, sämtliche Werte über Systemvariablen zwischenkoppeln (so die allgemein üblichen Lösungswege)

Es gibt noch einen anderen, eigentlich ist dieser sogar von EQ3 dokumentiert (offizielle xmlrpc Dokumentation, Kapitel 5.1 Methoden der Logigschicht - event. also keine Geheimnisse, an die man nur mit einem Disassembler kommt oder die man hüten müsste wie die Abschusscodes von Atomraketen 😊) über diese Logigschicht Methode teilen die Schnittstellenprozesse der Rega Änderungen ihrer Zustände mit. Und das ist dann der Punkt an dem sich der SDV auch einklinkt.

Anwender des SDV kannten ja schon die Möglichkeiten Datenpunkte oder Systemvariablen mit der Eigenschaft `operation_write` mit dem SDV zu verändern.

The screenshot shows the Script Developer V5.11.02 interface. The main window has tabs for Host, Selektiert, Syntaxcheck, Methodenhilfe, Backup, Scripteditor, FavoritenAnsicht, Inspector, CCU Services, Einstellungen, Systemprotokoll, and Debugger. The Debugger tab is active, displaying a list of variables:

Methoden	Typ	Value
ID	integer	6925
Name	string	ExtAUtemp
Type	integer	1099
TypeName	string	VARDP
Enabled	boolean	true
Used	boolean	true
Internal	boolean	false
Visible	boolean	true
Unerasable	boolean	false
EnumMetadata	string	MAX, MIN, UNIT
MAX	real	99
MIN	real	-99
UNIT	string	°C
DFInfo	string	Externer Inwert Aussen...
State	real	54.200000
ValueUnit	string	°C
ValueType	integer	4
ValueTypestr	string	Finat
ValueSubtype	integer	0
LastValue	real	0.000000
Timestamp	time	2020-03-04 16:39:13
LastTimestamp	time	1970-01-01 01:00:00
LastTriggererID	integer	6935
LastTriggererTime	time	2020-03-04 16:39:13
LastTriggererTimeSeconds	integer	1583509153
Operations	integer	7
ValueMin	string	-99
ValueMax	string	99
ValueName0	string	0
ValueName1	string	1

Below the variable list, there is a dropdown menu for "DemScan" set to "ID_SYSTEM_VARIABLES". To the right, there is a search interface with fields for "Types" (set to "EnumIDs") and "Scriptsuche". The search results pane shows a list of system variables:

ID	Name	TypeName	Information
40	\$(sysVarAlarmMessages)	VARDP	0
951	\$(sysVarPresence)	VARDP	false
41	\$(sysVarServiceMessages)	VARDP	4
1236	Alarmszone 1	ALARMDP	false
4074	Alarmszone 2	ALARMDP	true
1955	AllwaysTrue	VARDP	true
19725	Anwesenheit	VARDP	true
19726	Anwesenheit.Black	VARDP	true
19727	Anwesenheit.Gast	VARDP	false
19723	Anwesenheit.Ist	VARDP	3
19721	Anwesenheit.Ruhe	VARDP	true
19724	Anwesenheit.string	VARDP	Black, Susa
19727	Anwesenheit.Susa	VARDP	true
9704	AstroTag	VARDP	true
8291	Außentemperatur	VARDP	38.200000
11125	Bedeckheimer Lüftungswemp...	VARDP	JA
950	CCU_BOOT	VARDP	true
3478	CCU_DutyCycle	VARDP	0.000000
3842	CCU_SysTime	VARDP	1933.000000
3841	CPU_Logout	VARDP	0.000000
3840	CPU_Logout	VARDP	1.000000
11007	CMD_SndProfNextProfile	VARDP	10442.3
5609	DutyCycle	VARDP	1.000000
3639	EH_GruppeAbbau	VARDP	true
4079	EH_BeiSchlaf	VARDP	false
6183	EH_ShutDown	VARDP	true
6925	ExtAlert	VARDP	56.200000
7035	ExtFernwartStatus	VARDP	abcd
7072	ExtMailHeute	VARDP	TRALIAIA
7073	ExtMailMorgen	VARDP	
5704	Feiertag.Nacht	VARDP	false
5704	Feiertag.Morgen	VARDP	false
3579	ExtMail.Nacht	VARDP	0.000000
3280	ExtMail.LostDevice	VARDP	HWIP-HW2
6660	ExtTempAlarm	ALARMDP	
6457	ExtTempAbhol	VARDP	false
6463	ExtTempWerteListe	VARDP	1
6458	ExtTempZahl	VARDP	74.980000

Die Systemvariable bzw der Datenpunkt kann so einfach geändert werden zum Testen. Das ist auch nicht so die Große Schwierigkeit. interessant wird nun, ich möchte ein Programm schreiben, welches mit bei Änderung der Luftfeuchtigkeit eines WT2 auf mehr als 80 Prozent ein Programm triggert

Eigentlich ein kleines WebUI Programm nun, um dies zu demonstrieren.

Nun wollen wir allerdings die Funktion dieses Programmes prüfen. Dazu müssen wir den Humidity Datenpunkte ändern. Der Datenpunkt Humidity ist allerdings ein operation_read, operation_event, also kein Datenpunkt, den wir mit state oder value irgendwie geändert bekommen. trotzdem lässt sich im SDV nun dieser Datenpunkt bei Value anwählen und editieren:

wir wollen testen, also stellen wir anstatt der gemessenen 42% nun mal 84% ein und sagen Wert übernehmen. Die Funktion Des SDv erkennt selbstständig, ob dieses ein Datenpunkt ist der sich normal über .State () oder .Value () umschreiben lässt, oder aber ob der etwas brutalere weg über xmlrpc.event zu nehmen ist.

ID	Name	Type	Value
11396	Humi-RF-000A9709A0DA37:1	integer	42

ein Blick in die WebUi verrät uns: die Rega hatte die zwangsläufig untergeschobenen Daten übernommen.

wir konnten sehen, das Programm hatte getriggert:

der Zustand bleibt natürlich so lange bestehen bis... logischerweise der xmlrpc process über den gleichen Weg der Rega ein neues Messergebnis mitteilt. es ist ja der gleiche Weg. Allerdings lassen sich so sämtliche Datenpunkte der Rega setzen bzw verbiegen. Auch zum Testen von Systemmeldungen auf dem Channel:0 funktioniert diese Vorgehensweise. Wenn einer beispielsweise Skripte für Kommunikationsstörungen etc. schreibt, so kann man auf dem Wege zum Testen Störungen setzen und auch wieder wegnehmen.

Logischerweise hier der Einwurf:

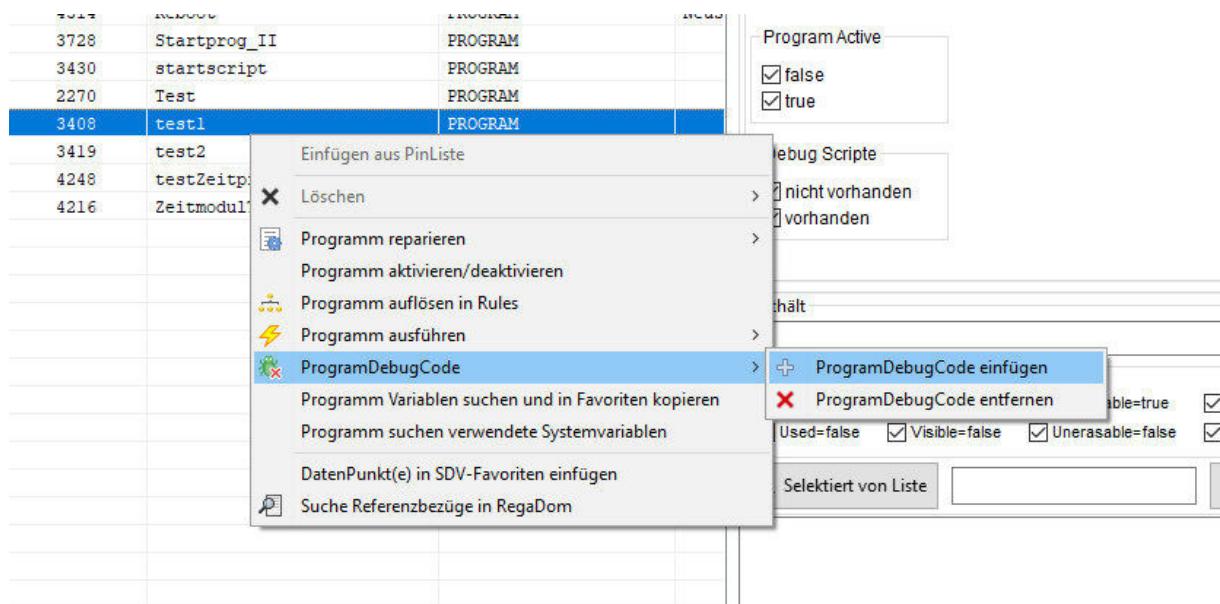
Das ist kein Spielzeug um irgendwelche Störungen wegzuklicken (geht) oder wahllos irgendwelche Daten zu setzen (geht auch) sondern um Sinnvoll mit der CCU zu arbeiten bzw Programme zu testen oder Fehler zu suchen und zu simulieren.

Aus dem Grunde war diese Funktion früher auch höheren Level vorbehalten.

und: ein zwangsumschreiben der Rega auf dem Weg wirkt sich nicht auf den xmlrpc aus, ein IOBroker z.b. bekommt von der IchÄnderEinenDatapointperEvent nix mit, da IOBroker direkt auf den RPC Prozess aufsetzt und von diesem die direkten Daten geliefert bekommt.

7.1.2 Automatisiertes Einfügen von DebugSkripten

In ein oder mehrere, selektierte Programme lassen sich im Inspektor Debugskripte einfügen. Dieses Debugskript schreibt entweder in den „richtigen“ Syslog oder in das CCU Systemprotoll einen Eintrag in der Art: Welches Programm wurde wann mit welchem Typ Datenpunkt mit welchem Wert ausgelöst und in welcher Bedingung des Programmes wurde dies Ausgeführt (Wenn, 1. Sonst Wenn, x. Sonst Wenn, Sonst)

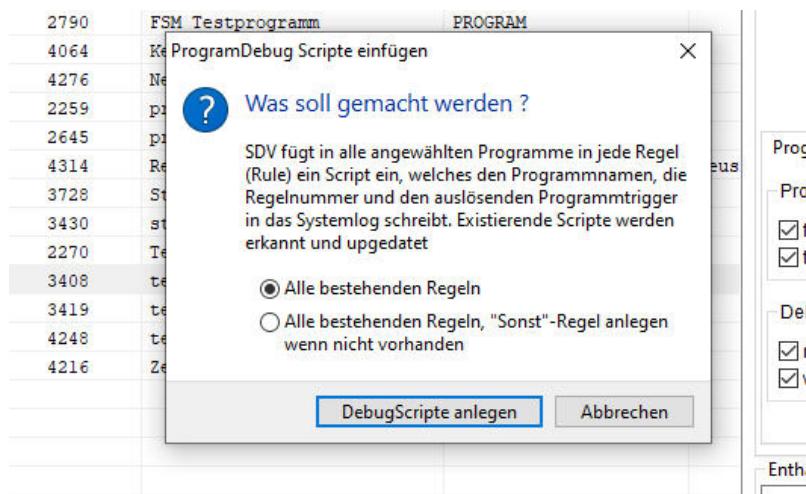


Der Code kann eingefügt und auch wieder gelöscht werden.

rechte Maustaste

ProgrammDebugCode - Program Debug Code einfügen.

es öffnet sich Bestätigungsmenü:



Auswahl:

- alle bestehenden Regeln:

in jede Regel wird im Ausführungsteil das Debug Skript eingefügt. Wenn dieses schon existiert, so wird das Skript nur aktualisiert, es wird kein neues angelegt. Wenn es keinen Sonst teil gibt, so wird im Sonst Teil auch kein Skript angelegt

- alle bestehenden Regeln, Sonst Regel anlegen wenn nicht vorhanden:

in jede Regel wird im Ausführungsteil das Debug Skript eingefügt. Wenn dieses schon existiert, so wird das Skript nur aktualisiert, es wird kein neues angelegt. Wenn es keinen Sonst teil gibt, so wird der Sonst Teil angelegt und dort ebenfalls das Debug Script angelegt.

In dem Debugskript wird vom SDV auch Automatisiert die Regelposition eingetragen, Die Wenn Regel wird mit WENN gekennzeichnet,
Die "SONST WENN" werden durchnummeriert eingetragen, gibt es 3 Sonst-Wenns in einem Programm, so werden diese auch automatisch mit 1. SONST WENN , 2. SONST WENN etc gekennzeichnet, ein SONST wird ebenfalls erkannt und automatisch gekennzeichnet.
Damit lässt ich im Systemprotoll sofort die ausgeführte Regel erkennen.

Name	Beschreibung	Bedingung (Wenn...)
test1		Systemzustand: SysVar bei Änderung auslösen ist wahr Bedingung: Wenn... - ODER: SysVar Ändert sich mit wahr / Bei Änderung auslösen - ODER: Periodisch: Gesetzte beobachtet am 04.07.2023 / Zeitintervall - ODER: HM-RCV-50-BB0Ca5-RF:1 / Zeitintervall lang - ODER: HM-Sec: RHS LEO1248855:1 / Zeitintervall kurz Aktivität: Dann... <input checked="" type="checkbox"/> Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern). - Test1: sofort - dem.GetObject("ID_SYSTEM_VARIABLES").Get("15Sec").State (noch) sofort - SDV_Programm.Debug.Schalt.Y1.0.Irv.Block.In.2021 (erstellt vor...) Aktivität: Dann... <input checked="" type="checkbox"/> Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern). - SDV_Programm.Debug.Schalt.Y1.0.Irv.Block.In.2021 (erstellt vor...)

Nach dem Ausführen von Skripte einfügen wird ein kurzer Report dargestellt, in wie vielen Programmen wurde in wie vielen Regeln das Skript eingeführt bzw in wie vielen Regeln wurde das Skript geupdated.

4064	Keypressesa	PROGRAM
4276	Neues Programm	PROGRAM
2259	prgEnergyCounter_2240_0...	PROGRAM
2645	pi SDV_V4xxxx	
4314	Re	
3728	St Debugscripte in 1 Programm(en) erfolgreich installiert	
3430	st	
2270	Eingefügte Scripte: 0 Te Update vorhandener Scripte: 2	
3408	te	
3419	te	
4248	te	
4216	ZeitmodulTest	PROGRAM

DebugSkripte wieder entfernen.

Programme auswählen rechte Maustaste

ProgrammDebugCode - Program Debug Code entfernen.

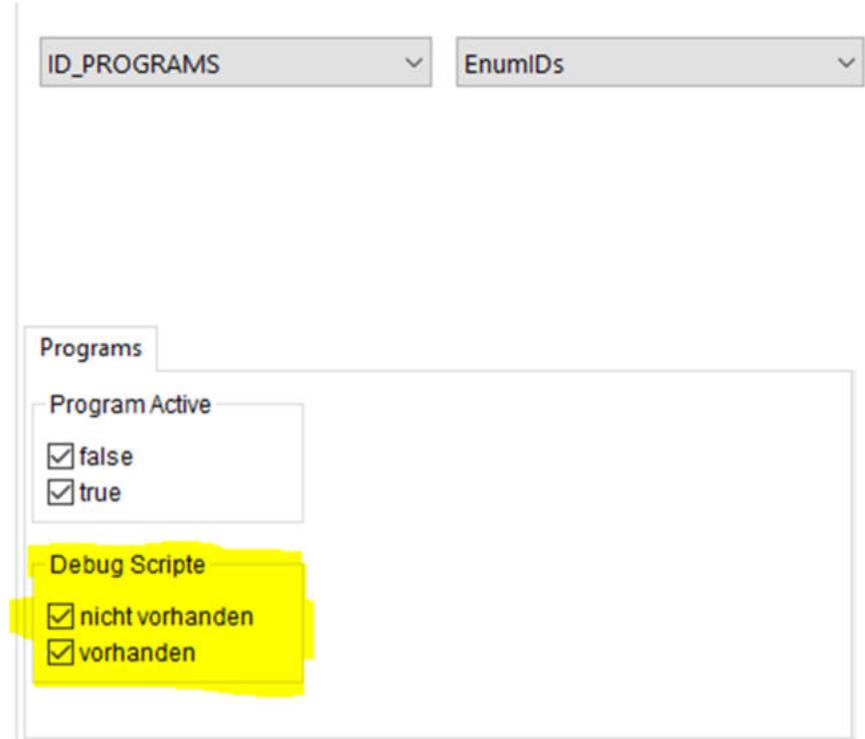
4064	Keypressed	PROGRAM
4276	Neues Programm	PROGRAM
2259	prgEnergyCounter_2240_0...	PROGRAM
2645	pi ProgramDebug Scripte löschen	
4314	Re	
3728	St Was soll gemacht werden ?	
3430	st	
2270	SDV löscht aus allen angewählten Programmen in jeder Regel (Rule) die Debugscripte, welche der SDV selber angelegt hatte	
3408	te	
3419	te	
4248	te	
4216	ZeitmodulTest	

in dem sich öffnenden Bestätigung Menü das Löschen bestätigen.

Nach dem Löschen von Skripte einfügen wird ein kurzer Report dargestellt, in wie vielen Programmen wurde in wievielen Regeln das Skript gelöscht.

Selektieren von Programmen mit Debugskripten / ohne Debugskripten im Inspektor lässt sich unter ID_PROGRAMS anwählen ob nach

- Programmen mit enthaltenem Debugskript
 - Programmen ohne enthaltenem DebugSkript
- oder egal ob enthalten oder nicht selektiert werden soll



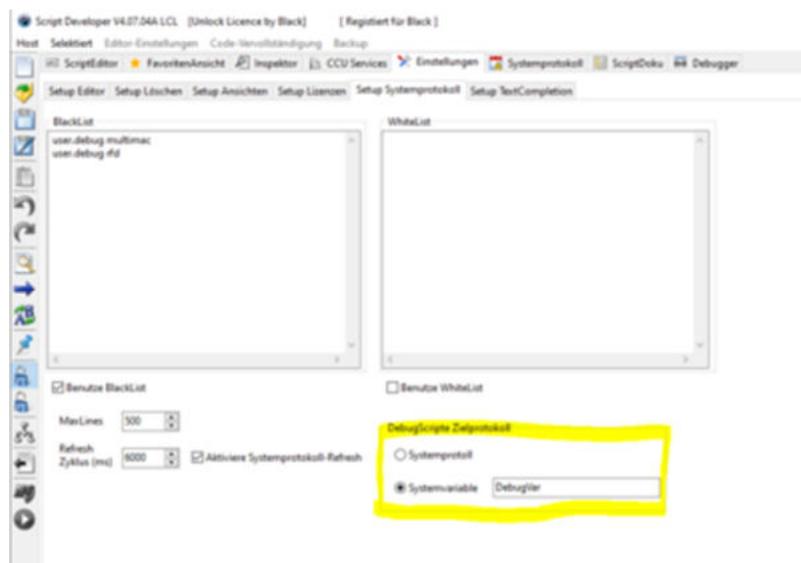
DebuggingScript optional mit Möglichkeit, den Debugtext auch in eine protokollierte Systemvariable zu schreiben.

dazu kann hier definiert werden:

wie bisher: schreiben ins "richtige" Systemprotokoll

oder

schreiben in die hier angegebene Systemvariable.



Hilfetexte sind dazu hinterlegt, die Einstellungen werden in der XML automatisch persistiert

Ist Systemvariable ausgewählt aber keine Systemvariable angegeben: dann gibt es auch keinen Menüeintag: Debugtext schreiben

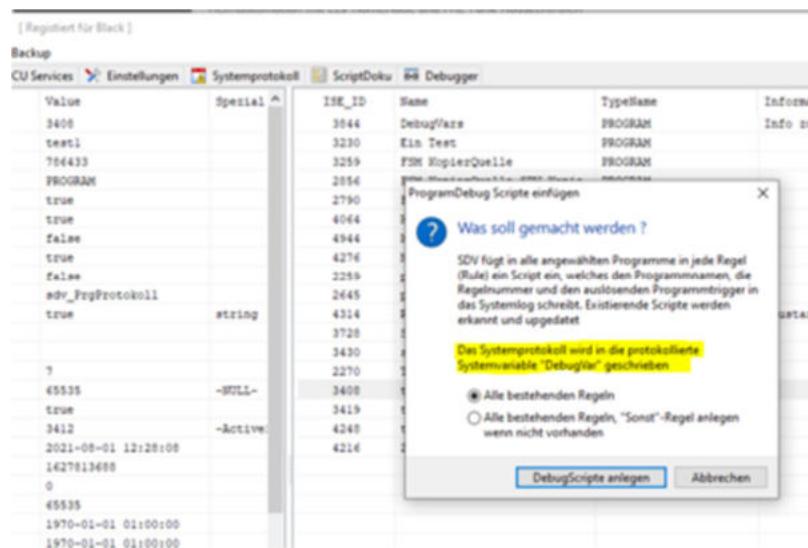
existiert die Systemvariable nicht auf der CCU, so wird der SDV beim Schreiben der Debugskripte diese Systemvariable typ string anlegen.

existiert die Systemvariable schon ist aber kein String, so wird diese gelöscht und als String neu angelegt.

SDV stellt die Systemvariable automatisch als protokolliert ein.

Der eingegebene Name sollte schon möglich sein... \$%&[]§§ lässt sich zwar eintragen, das wäre allerdings dämlich.

Im Menü für das Bestätigen des Skripteintrags kommt noch der Hinweis, wenn Systemvariable angewählt, wo dieses eingetragen wird.



7.2 Backups

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

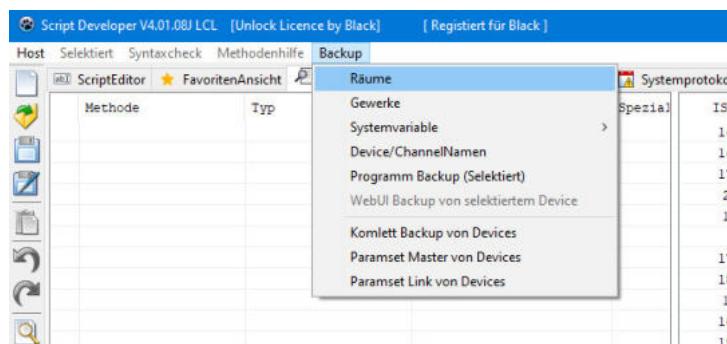
Vielmehr dienen diese im Falle eines Umzuges von einem alt System auf ein Neusystem als Hilfestellung, wenn man das alte Systembackup nicht benutzen will (Loswerden von in den Jahren angesammelten Leichen), oder aber ein inkonsistentes System.

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

z.B.

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

Den passenden Lizenzlevel vorausgesetzt, finden sich die Backups hier:



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

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

Hinweis an die Bestandsnutzer nach Wechsel auf die breaking Change Rega

Aufgrund des geänderten Escaping Verhaltes ist es erforderlich, alle Backups nochmal laufen zu lassen. Grund: Die Wiederherstellungs routinen können sonst aufgrund des Script runtime errors, welcher durch das geänderte Escaping kommt, abbrechen

7.2.1 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 Skripteditor 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.

7.2.2 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 Skripteditor 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.

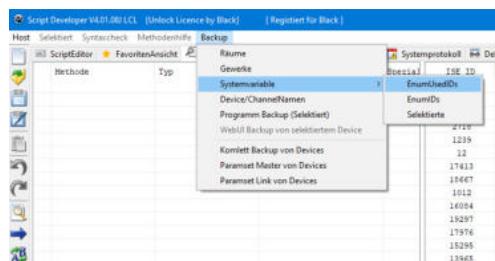
Waren dem alten Gewerk Kanäle zugeordnet, so versucht der SDV nun diese Kanäle des Altsystems über ihren Kanalnamen zu identifizieren. Ist dieses erfolgreich, so wird dieser Kanal dem Gewerk hinzugefügt.

7.2.3 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.

Die zu sichernden Systemvariablen können selektiert werden:



EnumUsedIDs (): Alle Systemvariablen, die in der Aufzählung EnumUsedIDs() gefunden werden, werden gesichert

EnumIDs (): Alle Systemvariablen, die in der Aufzählung EnumIDs() gefunden werden, werden gesichert

Selektierte: Alle Systemvariablen, die in der Listenansicht selektiert sind, werden gesichert.

Auf dem „Neusystem“ lässt sich dieses Programm über den SDV via Skripteditor 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)

7.2.4 Devices und Kanäle

Bei diesem Backup werden von den selektierten Geräten 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.

7.2.5 Backup Programme

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

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

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

Bedeutung der Abkürzungen

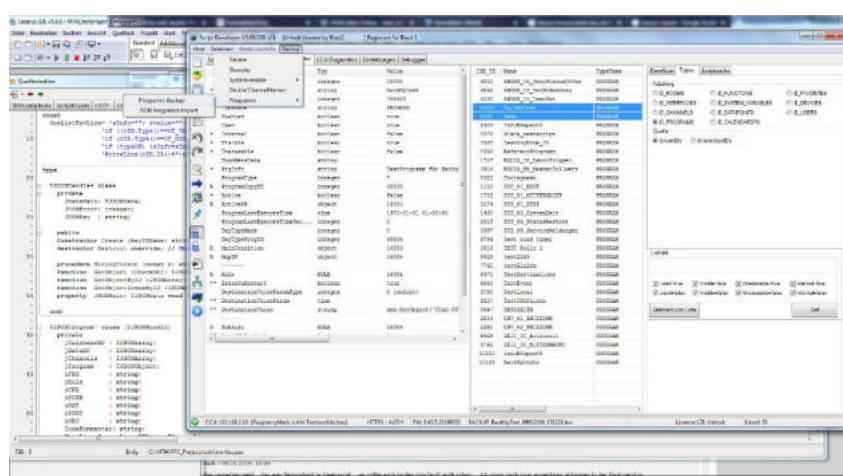
\$NAME: der Name des Programmes wird eingesetzt

\$FW: Firmwareversion

\$DATE: Datum in der Form JJJJMMTT

\$TIME: Zeit in der Form HHMMSS

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



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

Der Programmname heisst 'BACKUP_ + ProgrammName + _ + TTMMYYYY_HHMMSS.hsc bzw .json danach werden die Backups der ausgewählten Programme angelegt.

werden Objekte selektiert, die keine Programme sind, so werden diese natürlich ignoriert.

So sollte es aussehen, wenn alles durchlief:

boolean	false		8666	ReferenzProgramm
string			1747	ROLLO_00_AstroTrigger
string	TestProgramm für Backup		3916	ROLLO_XX_MasterSollwert
integer	7		8622	Stringtest
integer	65535		1433	SYS_01_BOOT
boolean	false		1702	SYS_01_MITTENNACHT
object	10054		3479	SYS_01_STAT
time				stemZeit
integer				statusRestore
integer				rviceMeldungen
integer				timer
object				o 1
object				m
RULE				nations
boolean	true		6063	TestEvent
integer	0 (sofort)		3730	TestLocal
time			8107	TestPRGFalsch
string	dom.GetObject("CUxD.CU		9047	TESTZEITIM
RULE	10063		2003	USV_01_SHUTDOWN
			2280	USV_02_SHUTDOWN
			8528	ZEIT_00_Astrozeit

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

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

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

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

Zum Restore.

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

Bei einem Erfolgreichen Durchlauf sieht die Skriptausgabe so aus:

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

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

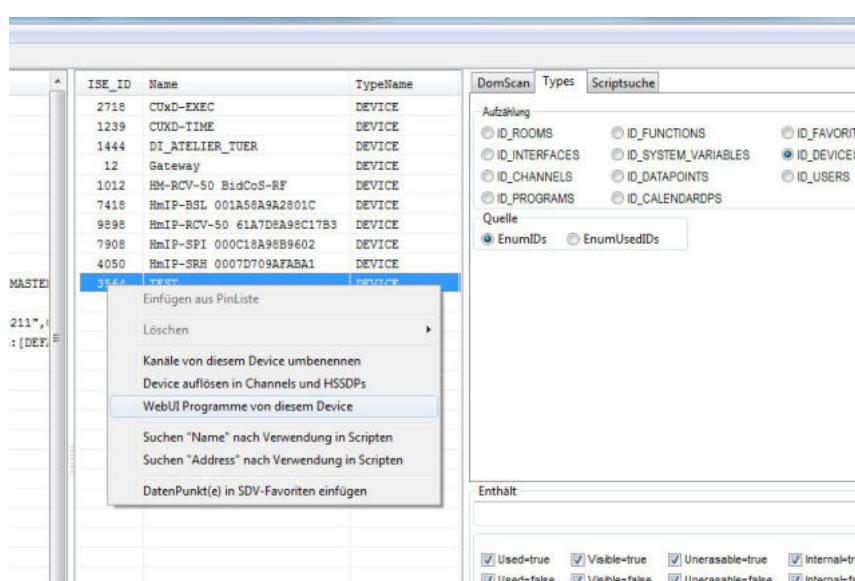
Funktion ist unspektakulär kann die WebUI auch (habt nur programmiert, weil ein Device keine idarray hat, wo die verwendeten Programme drinstehen)

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

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

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

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



7.2.6 WebUi Backup von selektiertem Device

Bekannter weise erfreut uns EQ3 ja ab und an mal mit der Notwendigkeit, ein Gerät mit Werksreset ab und dann wieder anlernen zu müssen. Leider wird beim Ablernen und Neuanlernen:

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

1. der Device und alle Kanalnamen auf Defaulteinstellung zurückgesetzt
 2. Alle Gewerke Zuweisungen der Kanäle sind weg
 3. Alle Raumzuweisungen der Kanäle sind weg
 4. Alles Favoriten Zuweisungen von Kanälen sind weg
 5. In Kanälen zugewiesene Systemvariablen sind weg
 6. Programme, in denen das Device verwendet wurde, sind verstümmelt

Zerrupftes Programm nach Device löschen

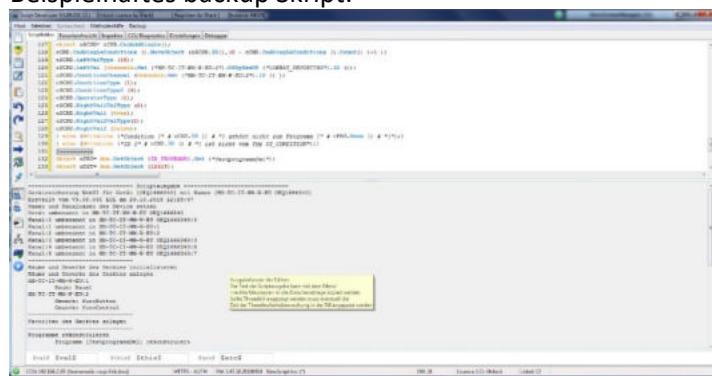


Jedes Mal bedeute dieses eine WebUi Klicki Wiederherstellorgie.

Dies übernimmt nun der Menüpunkt WebUi Backup von Selektiertem Device:

- Vor dem Ablernen wird das Device selektiert und anschließend dieses Backup durchgeführt. Das Backup wird gespeichert, liegt aber auch im clipboard und kann direkt in den Editor eingefügt werden.
 - Das Gerät nun mit Werksreset löschen
 - Gerät wieder anlernen
 - Das vorhin vom SDV automatisch erstellte restore Skript ausführen
 - Fertig.

Beispielhaftes backup Skript:



Das Skript enthält automatisch generierten Code, nach dessen Durchlauf:

1. Stimmt der Gerätename wieder
2. Stimmen alle Kanalnamen wieder
3. Sind alle Kanäle wieder den ursprünglichen Gewerken zugeordnet
4. Sind alle Räume wieder den ursprünglichen Gewerken zugeordnet
5. Sind alle Kanäle wieder in den ursprünglichen Favoriten und an der richtigen Position
6. Systemvariablen, die vorher Kanälen zugewiesen waren, sind es nun auch wieder
7. Alle Programme, in denen Datenpunkte des Device verwendet wurden, sind wiederhergestellt und die Bedingungen auch an der richtigen Position

Alles, was mit der WebUI zu tun hatte, ist nach dem Durchlauf wieder korrigiert. . (Zum Einsatz kommen hier auch Mechanismen aus dem schon lange laufenden Programme-Backup)

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

Ansonsten ist der Identifiziermechanismus

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

Wiederhergestelltes WebUI programm

Fortgeschrittene können mittels des restore Skriptes auch innerhalb der WebUi verschiedene Geräte tauschen, wobei da natürlich die Struktur der Geräte schon stimmig sein muss.

Erfolgreich wurde dieses im Forum schon praktiziert, dabei muss

1. Die Geräteseriennummer innerhalb des Skriptes getauscht werden
2. Die Kanal Struktur überprüft und gegebenenfalls händisch angepasst werden
3. Die Datenpunkte müssen zum Kanal passen.

Ist dieses erfüllt, ließen sich Tauschoperationen auch zwischen identischen, bei entsprechender Anpassung sogar auch zwischen ähnlichen HmIP Geräten durchführen.

7.2.7 Komplett Backup von Device

Mittels Komplett Backup von Device kann von den Mastersets / Linksets ein Komplettes Backup hergestellt werden. Dieser Punkt ist sehr hilfreich, um nach einem Ablernen mit Werksreset / Wiederanlernen alles Geräteeinstellungen sowie alle Direktverknüpfungen wiederherstellen zu können

Von den markierten Devices werden die BackUP Jsons angelegt. Dieses kann dann ganz normal in den SDV Editor geladen werden und dort ausgeführt werden.

Dieses backup ist auch in der Lage, ein Gerät auf einem frischen System wiederherzustellen, nicht nur auf dem Ursprungssystem.

Ein restore macht:

1. Gerätenamen wiederherstellen
2. Alle Channamen wiederherstellen
3. Alle Eigenschaften ChanArchive wiederherstellen
4. Systemvariablen, die in Kanälen zugewiesen wurden, werden angelegt, wenn sie sich noch nicht auf dem System befinden
5. Systemvariablen werden den Kanälen zugewiesen
6. Verwendet Räume und Gewerke werden angelegt, wenn sie sich noch nicht auf dem System befinden
7. Räume und Gewerke werden den Kanälen zugewiesen
8. Alle Mastersets der Geräte werden wiederhergestellt
9. Abweichende Metadaten werden wiederhergestellt
10. Sonderobjekte werden angelegt wenn benötigt, oder verwiesen, wenn schon vorhanden
11. Alle bestehenden Direktverbindungen des Gerätes werden gelöscht
12. Alle Direktverbindungen aus dem Backup werden erzeugt, benannt, und parametert.

Die Ausführung läuft als 3 Pass innerhalb des Editors und kann einige zeit dauern:

Beispiel eines restore eines einfachen 1 Kanal Aktors:

```
JSON Restore Complete Device
ThreadkillTimeout eingestellt auf 20000 ms
Complete Device Backup (PASS 1 von 3) by Black in 2020
Backup erstellt vom SDV V4.01.08J LCL am 24.07.2020 15:42:33
GeräteName : Aktor1
GeräteType : HM-LC-Sw1-DR
Seriennummer: PEQ0480086
-----
Device umbenannt in "Aktor1"
Überprüfung Vorhandensein Räume/Gewerke/Systemvariablen
Kanal PEQ0480086:0 umbenannt in "Aktor1:0". Zuweisungen erfolgt
Kanal PEQ0480086:1 umbenannt in "Aktor1:1". Zuweisungen erfolgt
----- Ausführung -----
15:53:14:788 Sende Script an CCU [TX 3445 Bytes]
15:53:15:141 Empfange Daten von CCU [RX 741 Bytes]
15:53:15:172 CCU Laufzeit 0.343 sec

Complete Device Backup (PASS 2 von 3) by Black in 2020
Datensicherung ParamSet MASTER by Black in 2020
Backup erstellt vom SDV V4.01.08J LCL am 24.07.2020 15:42:33
GeräteName : Aktor1
GeräteType : HM-LC-Sw1-DR
ChannelNames: NO
SerNummer : PEQ0480086
-----
Paramset Device [Aktor1] - Ergebnis: 0: OK
Paramset Channel [Aktor1:1] - Ergebnis: 0: OK
Wiederherstellung Metadaten und Property Objects
----- Ausführung -----
15:53:15:172 Sende Script an CCU [TX 4395 Bytes]
15:53:17:857 Empfange Daten von CCU [RX 1755 Bytes]
15:53:17:857 CCU Laufzeit 2.687 sec
```

```

Complete Device Backup (PASS 3 von 3) by Black in 2020
Datensicherung ParamSet LINKS by Black in 2020
Backup erstellt vom SDV V4.01.08J LCL am 24.07.2020 15:42:33
GeräteName : Aktor1
GeräteType : HM-LC-Sw1-DR
Seriennummer: PEQ0480086
Gelöscht wird : [BidCoS-RF:1]      --> [PEQ0480086:1]
Gelöscht wird : [LEQ1252736:1]     --> [PEQ0480086:1]
Wird nicht gelöscht : [PEQ0480086:1] --> [PEQ0480086:1] Grund: Flag=1
Das Gerät enthielt 3 Direktverknüpfungen
2 dieser Verbindungen wurden gelöscht
1 dieser Verbindungen referenzieren intern (Flags) und wurden nicht gelöscht
-----
1. Direktverknüpfung
-----
Channel Empfänger: Aktor1:1
HssType Empfänger: SWITCH
Adress Empfänger : PEQ0480086:1
Channel Sender   : HM-RCV-50 BidCoS-RF:1
HssType Sender   : VIRTUAL_KEY
Adress Sender    : BidCoS-RF:1
DV (PEQ0480086:1 <<- BidCoS-RF:1) angelegt [HM-RCV-50<> BidCoS-RF:1 mit Aktor1:1 : Standardverknüpfung
Virtuelle Fernbedienung - Schaltaktor]
DV (PEQ0480086:1 <<- BidCoS-RF:1) Parametersatz geladen - 0: OK
-----
2. Direktverknüpfung
-----
Channel Empfänger: Aktor1:1
HssType Empfänger: SWITCH
Adress Empfänger : PEQ0480086:1
Channel Sender   : HM-Sec-RHS LEQ1252736:1
HssType Sender   : ROTARY_HANDLE_SENSOR
Adress Sender    : LEQ1252736:1
DV (PEQ0480086:1 <<- LEQ1252736:1) angelegt [HM-Sec-RHS LEQ1252736:1 mit Aktor1:1{ : Standardverknüpfung
Fenster-Drehgriffkontakt - Schaltaktor]
DV (PEQ0480086:1 <<- LEQ1252736:1) Parametersatz geladen - 0: OK
-----
3. Direktverknüpfung
-----
Channel Empfänger: Aktor1:1
HssType Empfänger: SWITCH
Adress Empfänger : PEQ0480086:1
Channel Sender   : Aktor1:1
HssType Sender   : SWITCH
Adress Sender    : PEQ0480086:1
DV (PEQ0480086:1 <<- PEQ0480086:1) war angelegt, Beschreibung angepasst [name 33 : ÄÖÜ äöüßaa]
DV (PEQ0480086:1 <<- PEQ0480086:1) Parametersatz geladen - 0: OK
Korrektur LINKCOUNT des Gerätes
-----
Folgende LinkCounts wurden angepasst:
----- Ausführung -----
15:53:17:872 Sende Script an CCU [TX 14113 Bytes]
15:53:21:061 Empfange Daten von CCU [RX 9955 Bytes]
15:53:21:076 CCU Laufzeit 3.187 sec
-----
Fertig
Ausführungszeit 6.313 sec

```

7.2.8 Backup Masterset

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

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

```
Script Developer V3.08.148 LCL [Unlock Licence by Back] [Reconnect for Back] [instance: MAIN]
Host Selektori Symtack Methodenhilfe Backup
FavoritenAnsicht Inspector | CCU Diagnostics Erstellungen Debuggen

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
417
418
419
419
420
421
422
423
424
425
426
427
427
428
429
429
430
431
432
433
434
435
436
437
438
438
439
440
441
442
443
444
445
446
447
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
468
469
470
471
472
473
474
475
476
477
478
478
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
517
518
519
519
520
521
522
523
524
525
526
527
527
528
529
529
530
531
532
533
534
535
536
537
538
538
539
540
541
542
543
544
545
546
547
547
548
549
549
550
551
552
553
554
555
556
557
558
558
559
560
561
562
563
564
565
566
567
568
568
569
570
571
572
573
574
575
576
577
577
578
579
579
580
581
582
583
584
585
586
587
587
588
589
589
590
591
592
593
594
595
596
597
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
615
616
617
617
618
619
619
620
621
622
623
624
625
626
627
627
628
629
629
630
631
632
633
634
635
636
637
637
638
639
639
640
641
642
643
644
645
646
647
647
648
649
649
650
651
652
653
654
655
656
657
658
658
659
660
661
662
663
664
665
666
667
667
668
669
669
670
671
672
673
674
675
676
677
677
678
679
679
680
681
682
683
684
685
686
687
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
715
716
717
717
718
719
719
720
721
722
723
724
725
726
727
727
728
729
729
730
731
732
733
734
735
736
737
737
738
739
739
740
741
742
743
744
745
746
747
747
748
749
749
750
751
752
753
754
755
756
757
758
758
759
760
761
762
763
764
765
766
767
767
768
769
769
770
771
772
773
774
775
776
777
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
815
816
817
817
818
819
819
820
821
822
823
824
825
826
827
827
828
829
829
830
831
832
833
834
835
836
837
837
838
839
839
840
841
842
843
844
845
846
847
847
848
849
849
850
851
852
853
854
855
856
857
857
858
859
859
860
861
862
863
864
865
866
867
867
868
869
869
870
871
872
873
874
875
876
877
877
878
879
879
880
881
882
883
884
885
886
887
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
915
916
917
917
918
919
919
920
921
922
923
924
925
926
927
927
928
929
929
930
931
932
933
934
935
936
937
937
938
939
939
940
941
942
943
944
945
946
947
947
948
949
949
950
951
952
953
954
955
956
957
957
958
959
959
960
961
962
963
964
965
966
967
967
968
969
969
970
971
972
973
974
975
976
977
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1015
1016
1017
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1116
1117
1118
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1196
1197
1198
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1987
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2138
2139
2140
2140
2141
2142
2143
2144
2145
2146
2147
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2187
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2197
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2238
2239
2239
2240
2241
2242
2243
2244
2245
2246
2247
2247
2248
2249
2249
2250
2251
2252
2253
2254
2255
2256
2257
2257
2258
2259
2259
2260
2261
2262
2263
2264
2265
2266
2267
2267
2268
2269
2269
2270
2271
2272
2273
2274
2275
2276
2277
2277
2278
2279
2279
2280
2281
2282
2283
2284
2285
2286
2287
2287
2288
2289
2289
2290
2291
2292
2293
2294
2295
2296
2297
2297
2298
2299
2299
2300
2301
2302
2303
2304
2305
2306
2307
2307
2308
2309
2309
2310
2311
2312
2313
2314
2
```

Ein Masterset Backup berücksichtigt auch:

Abweichende Metadaten

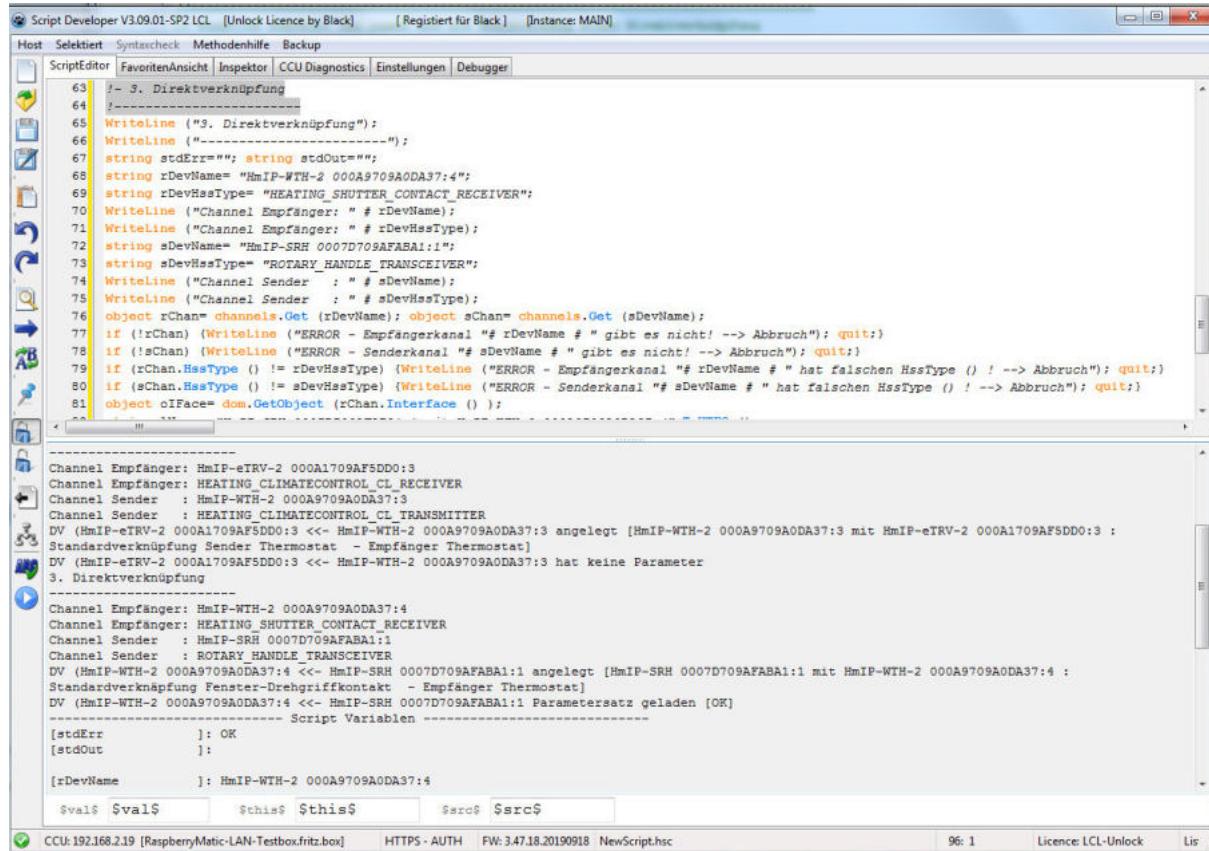
Sonderobjekte

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

7.2.9 Backup Linkset

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

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



The screenshot shows the Script Developer software interface. The top menu bar includes 'ScriptDeveloper V3.09.01-SP2 LCL' and 'Unlock Licence by Block'. The main window has tabs for 'Host', 'Selektiert', 'Syntascheck', 'Methodenhilfe', and 'Backup'. The 'Selektiert' tab is active. Below the tabs is a toolbar with icons for file operations, selection, and search. The main area contains two panes: a code editor on the left and an output pane on the right.

Code Editor Content:

```
63  !- 3. Direktverknüpfung
64  -----
65  WriteLine ("3. Direktverknüpfung");
66  WriteLine ("-----");
67  string stdErr=""; string stdOut="";
68  string rDevName= "HmIP-WTH-2 000A9709A0DA37:4";
69  string rDevHssType= "HEATING_SHUTTER_CONTACT_RECEIVER";
70  WriteLine ("Channel Empfänger: " # rDevName);
71  WriteLine ("Channel Empfänger: " # rDevHssType);
72  string sDevName= "HmIP-SRH 0007D709AFABA1:1";
73  string sDevHssType= "ROTARY_HANDLE_TRANSCIEVER";
74  WriteLine ("Channel Sender : " # sDevName);
75  WriteLine ("Channel Sender : " # sDevHssType);
76  object rChan= channels.Get (rDevName); object sChan= channels.Get (sDevName);
77  if (!rChan) (WriteLine ("ERROR - Empfängerkanal "# rDevName "# gibt es nicht! --> Abbruch"); quit);
78  if (!sChan) (WriteLine ("ERROR - Senderkanal "# sDevName "# gibt es nicht! --> Abbruch"); quit);
79  if (rChan.HssType () != rDevHssType) (WriteLine ("ERROR - Empfängerkanal "# rDevName "# hat falschen HssType () ! --> Abbruch"); quit);
80  if (sChan.HssType () != sDevHssType) (WriteLine ("ERROR - Senderkanal "# sDevName "# hat falschen HssType () ! --> Abbruch"); quit);
81  object oIFace= dom.GetObject (rChan.Interface ());
82
83  -----
84
85  Channel Empfänger: HmIP-eTRV-2 000A1709AF5DD0:3
86  Channel Empfänger: HEATING_CLIMATECONTROL_CL_RECEIVER
87  Channel Sender : HmIP-WTH-2 000A9709A0DA37:3
88  Channel Sender : HEATING_CLIMATECONTROL_CL_TRANSMITTER
89  DV (HmIP-eTRV-2 000A1709AF5DD0:3 <- HmIP-WTH-2 000A9709A0DA37:3 angelegt [HmIP-WTH-2 000A9709A0DA37:3 mit HmIP-eTRV-2 000A1709AF5DD0:3 :
90  Standardverknüpfung Sender Thermostat - Empfänger Thermostat]
91  DV (HmIP-eTRV-2 000A1709AF5DD0:3 <- HmIP-WTH-2 000A9709A0DA37:3 hat keine Parameter
92  3. Direktverknüpfung
93
94  -----
95  Channel Empfänger: HmIP-WTH-2 000A9709A0DA37:4
96  Channel Empfänger: HEATING_SHUTTER_CONTACT_RECEIVER
97  Channel Sender : HmIP-SRH 0007D709AFABA1:1
98  Channel Sender : ROTARY_HANDLE_TRANSCIEVER
99  DV (HmIP-WTH-2 000A9709A0DA37:4 <- HmIP-SRH 0007D709AFABA1:1 angelegt [HmIP-SRH 0007D709AFABA1:1 mit HmIP-WTH-2 000A9709A0DA37:4 :
100 Standardverknüpfung Fenster-Drehgriffkontakt - Empfänger Thermostat]
101 DV (HmIP-WTH-2 000A9709A0DA37:4 <- HmIP-SRH 0007D709AFABA1:1 Parametersatz geladen [OK]
102 -----
103     Script Variablen
104
105 [stdErr           ]: OK
106 [stdOut          ]:
107
108 [rDevName        ]: HmIP-WTH-2 000A9709A0DA37:4
109
110 $val$ $val$      $this$ $this$      $src$ $src$
```

Output Pane Content:

```
CCU: 192.168.2.19 [RaspberryMatic-LAN-Testbox.fritz.box]  HTTPS - AUTH  FW: 3.47.18.20190918  NewScript.hsc  96: 1  Licence: LCL-Unlock  Lis
```

7.3 Umbenennen von Kanälen von Geräten

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

Das Device wird selektiert und der Name der Device geändert.

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

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

Rückfrage mit Ja bestätigen und die Kanäle werden so benannt:

Device: DeviceName

Kanal0 : DeviceName:0

Kanal1 : Devicename:1

Etc...

7.4 Paramset Master

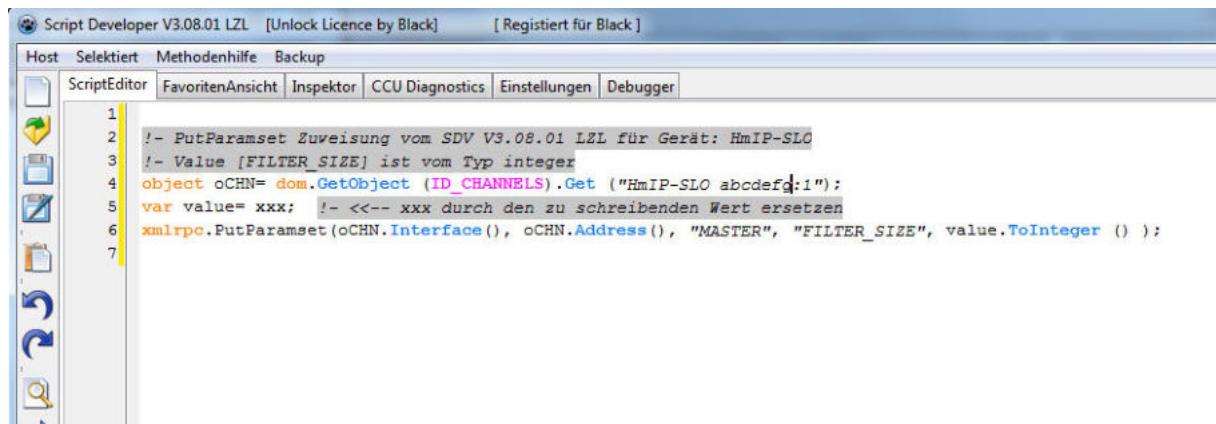
Bei Device, Kanälen, die einen Paramset Master haben, wird dieser mit angezeigt (wenn in der Sicht konfiguriert und mind. Level 5)

Methode	Typ	Value	Spezial	IIS_ID	Name	TypeName
ChNrUsageCount	Integer	0	HmIP-RF..001A55A	2782	HmIP-RC-19 CTXZ801001:9	CHANDEL
ChnLinkCount	Integer	0		2807	HmIP-RC-19 CTXZ801001:4	CHANDEL
B_ChnRoom	string			2832	HmIP-RC-19 CTXZ801001:5	CHANDEL
B_ChnFunction	string			2857	HmIP-RC-19 CTXZ801001:6	CHANDEL
B_DPs	lString	7557,7558,7559,7560,756...		2882	HmIP-RC-19 CTXZ801001:7	CHANDEL
				2907	HmIP-RC-19 CTXZ801001:8	CHANDEL
7557	RS32D		HmIP-RF..001A55A	2932	HmIP-RC-19 CTXZ801001:9	CHANDEL
7558	RS32D		HmIP-RF..001A55A	2957	HmIP-RC-19 CTXZ801001:10	CHANDEL
7559	RS32D		HmIP-RF..001A55A	2982	HmIP-RC-19 CTXZ801001:11	CHANDEL
7560	RS32D		HmIP-RF..001A55A	3007	HmIP-RC-19 CTXZ801001:12	CHANDEL
7561	RS32D		HmIP-RF..001A55A	3032	HmIP-RC-19 CTXZ801001:13	CHANDEL
7562	RS32D		HmIP-RF..001A55A	3057	HmIP-RC-19 CTXZ801001:14	CHANDEL
7563	RS32D		HmIP-RF..001A55A	3082	HmIP-RC-19 CTXZ801001:15	CHANDEL
7564	RS32D		HmIP-RF..001A55A	3107	HmIP-RC-19 CTXZ801001:16	CHANDEL
7565	RS32D		HmIP-RF..001A55A	3345	SB_Schalter:0	CHANDEL
7566	RS32D		HmIP-RF..001A55A	3359	HmIP-LC-0w2-FW LEQ1319211:1	CHANDEL
7567	RS32D		HmIP-RF..001A55A	3395	HmIP-LC-0w2-FW LEQ1319211:2	CHANDEL
7568	RS32D		HmIP-RF..001A55A	4051	HmIP-SR8 0007D709AFA8A1:0	CHANDEL
				4077	HmIP-SR8 0007D709AFA8A1:1	CHANDEL
				7415	HmIP-BSL 001A55A8A2021C:0	CHANDEL
				7444	HmIP-BSL 001A55A8A2021C:1	CHANDEL
				7447	HmIP-BSL 001A55A8A2021C:2	CHANDEL
				7450	HmIP-BSL 001A55A8A2021C:3	CHANDEL
				7455	HmIP-BSL 001A55A8A2021C:4	CHANDEL
				7461	HmIP-BSL 001A55A8A2021C:5	CHANDEL
				7467	HmIP-BSL 001A55A8A2021C:6	CHANDEL
				7473	HmIP-BSL 001A55A8A2021C:7	CHANDEL
				7482	HmIP-BSL 001A55A8A2021C:8	CHANDEL
				7495	HmIP-BSL 001A55A8A2021C:9	CHANDEL
				7508	HmIP-BSL 001A55A8A2021C:10	CHANDEL
				7521	HmIP-BSL 001A55A8A2021C:11	CHANDEL
				7530	HmIP-BSL 001A55A8A2021C:12	CHANDEL
				7543	HmIP-BSL 001A55A8A2021C:13	CHANDEL
				7554	HmIP-BSL 001A55A8A2021C:14	CHANDEL
				7569	HmIP-BSL 001A55A8A2021C:15	CHANDEL
				7587	HmIP-SPT 000C1AA9B9402:0	CHANDEL
				7615	HmIP-SPT 000C1AA9B9402:1	CHANDEL
				7623	HmIP-SLD 00051AA9PFAZ4:2	CHANDEL
				7625	HmIP-SLD 00051AA9PFAZ4:4	CHANDEL
				7645	HmIP-SLD 00051AA9PFAZ4:1	CHANDEL
				7657	HmIP-SLD 00051AA9PFAZ4:2	CHANDEL
				7658	HmIP-SLD 00051AA9PFAZ4:3	CHANDEL

Die Werte lassen sich dann über PutParamset verändern in einem Skript

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

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

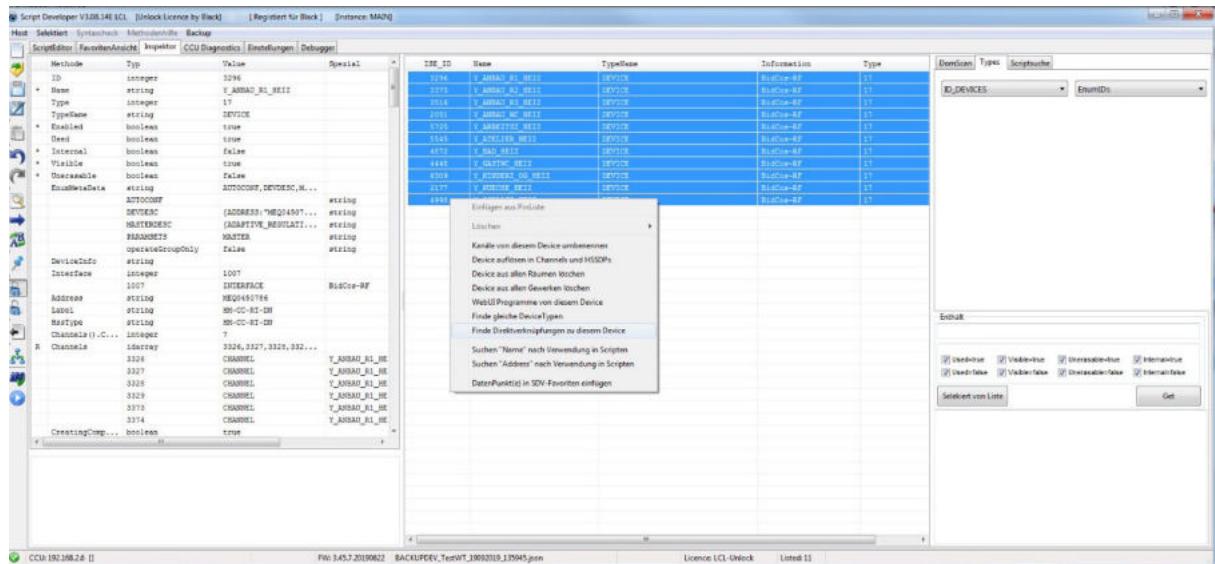


The screenshot shows the Script Developer V3.08.01 LZL interface. The title bar reads "Script Developer V3.08.01 LZL [Unlock Licence by Black] [Registert für Black]". The menu bar includes "Host", "Selektiert", "Methodenhilfe", and "Backup". Below the menu is a toolbar with icons for file operations like Open, Save, and Print. The main area is a code editor with the following content:

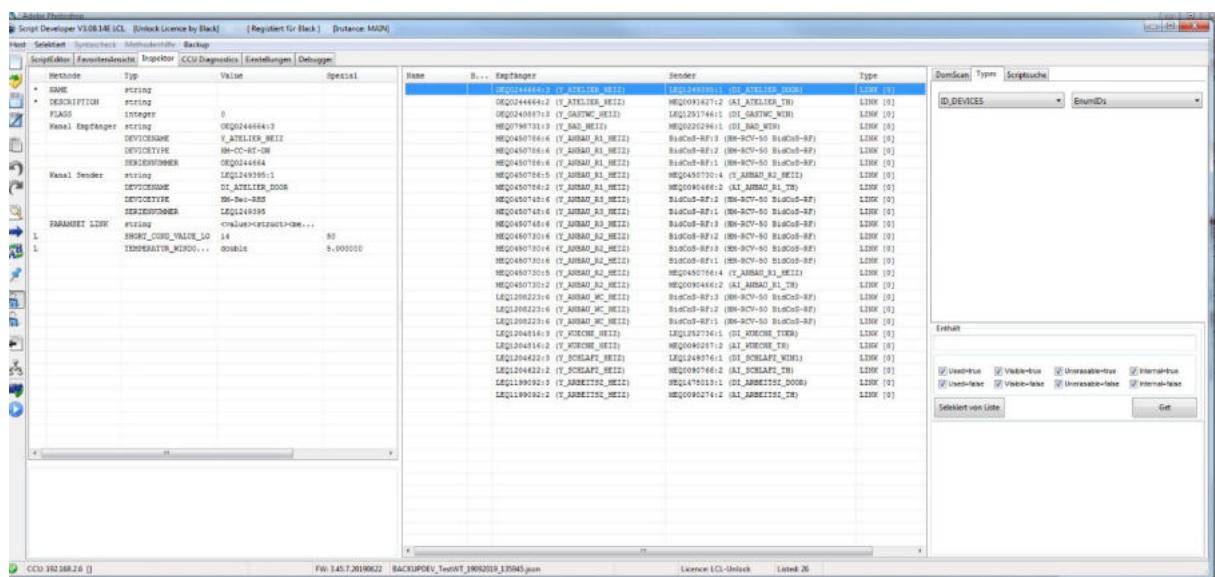
```
1  !- PutParamset Zuweisung vom SDV V3.08.01 LZL für Gerät: HmIP-SLO
2  !- Value [FILTER_SIZE] ist vom Typ integer
3  object oCHN= dom.GetObject (ID_CHANNELS).Get ("HmIP-SLO abcdefg:1");
4  var value= xxx; !- <<- xxx durch den zu schreibenden Wert ersetzen
5  xmipc.PutParamset(oCHN.Interface(), oCHN.Address(),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

7.5 Linkset eines Gerätes

Ist ein Device markiert, so liegt auf der rechten Maustaste in der Listenübersicht der Menüpunkt „Finde Direktverknüpfungen zu diesem Device“



Die markierte Liste wird abgearbeitet, ist das markierte Element ein Device, so wird dazu versucht, die zugehörigen Linksets aufzulösen.



Die Linksets können angewählt werden, Sortiert werden nach Name, Beschreibung, Empfänger und Sender. Gleichzeitig werden die Expertenparameter, so vorhanden unter Paramset Link dargestellt.

Diese können auch markiert werden und im Editor kann für diese Parameter automatisch der Code zum Beschreiben generiert werden.

7.5.1 Kopieren von Teilen eines Linksets einer DV in eine andere DV

Über die Möglichkeiten der WebUI was das Handling von DV angeht sowie dem Datenaustausch zwischen DVs muss ich ja nicht viel erzählen, ist ein Trauerspiel. Aus dem grunde wollte ich mich da auch immer mal drangesetzt haben, wenn man in DV1 etwas geändert hat, dass dieses schnell und ohne clickiclickiOrgien in gleichartigen anderen DVs zu ändern ist.

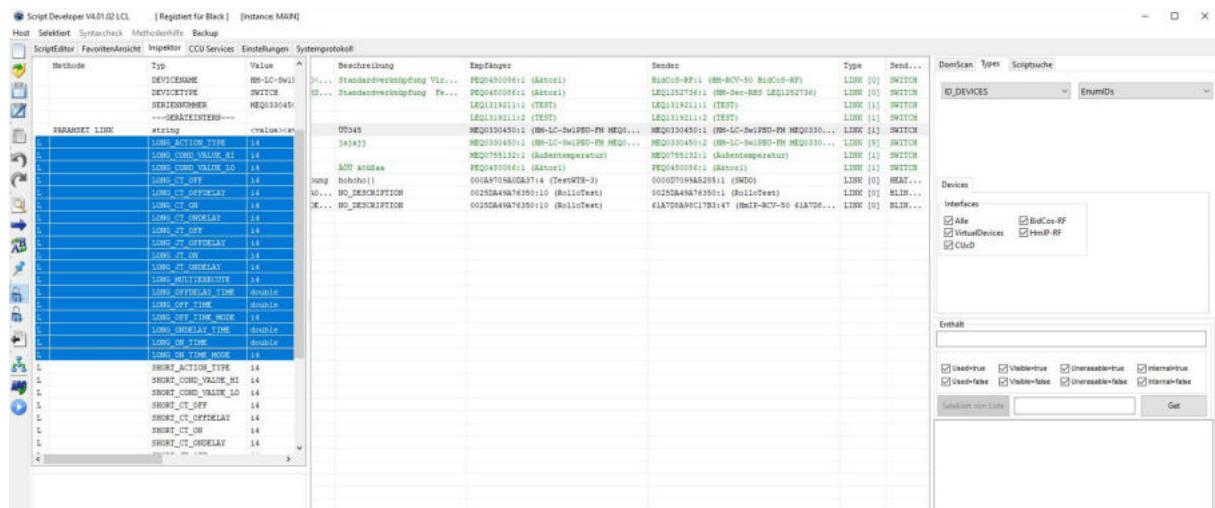
Das nötige Handling von Direktverknüpfungsdaten passte nicht so richtig in die Struktur der Gerätekopien unter CCU Services.

AUs diesem Grunde das Handling bei Direktverknüpfungen im Inspektor um die Drag-Drop Events erweitert.

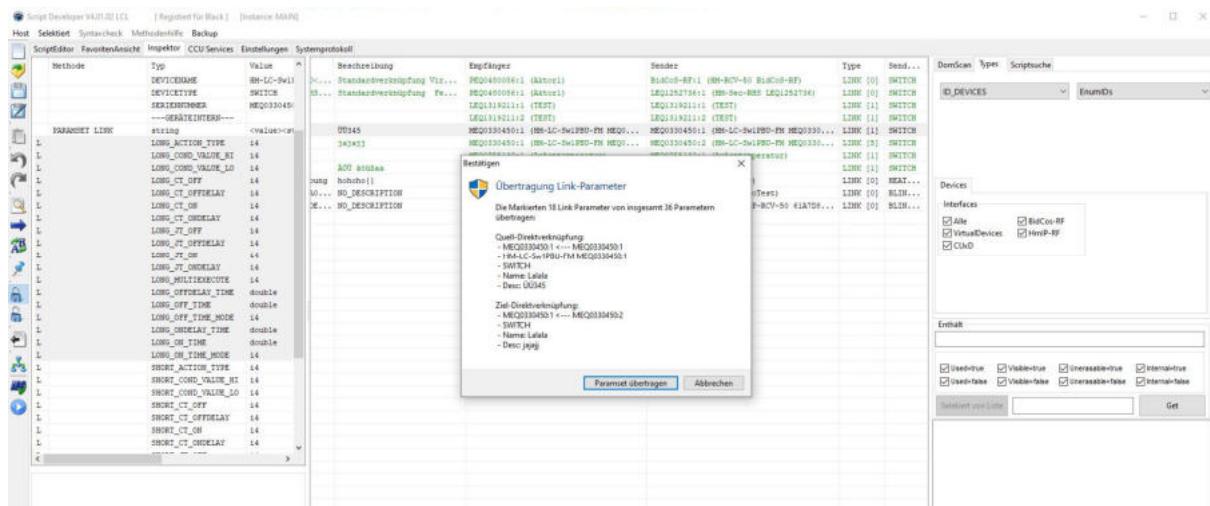
Damit ist es nun möglich, sich eine bestimmte Direktverknüpfung in der Detailansicht darstellen zu lassen, dann markiert man z.B. alle Parameter, die mit SHORT_ beginnen (alles für den kurzen Tastendruck) und zieht, wie in Windows üblich, dieses in die Listenansicht. Mit Beginn des Ziehens werden die anderen Direktverknüpfungen, deren Zielkanaleigenschaften zu dieser an der Maus hängenden Direktverknüpfung passen, grün dargestellt. Auf diesen kann dann die Maus gestellt werden und wie üblich gedropt werden. Es öffnet sich dann ein Abfragemenü, ob wirklich die Parameter von der Quelle Q ins Ziel Z kopiert werden soll, und bei Bestätigung nach Ausführung noch ein Statusdialog.

Beispielhaftes Vorgehen:

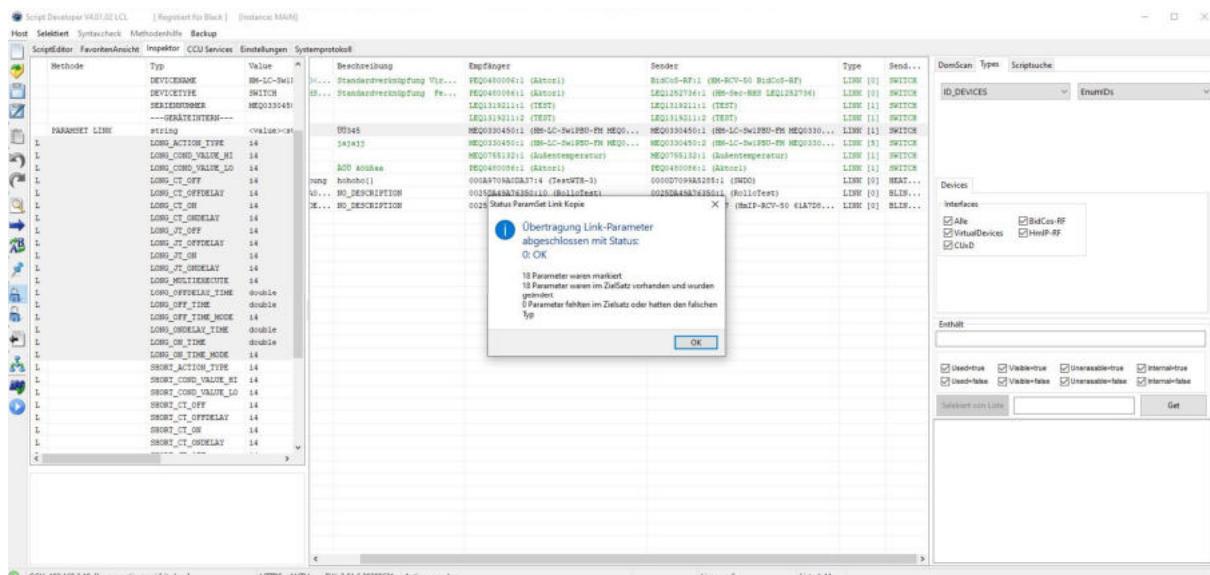
Auswahl einer DV und markieren eines Teiles des Parameter (LONG)



Ziehen von der Detailansicht in die Listenansicht. die möglichen ZieldVs sind nun grün geworden (Die nicht grün Eingefärbte mit der Switch Eigenschaft ist die QuellDV selber, nur ich steh nicht so auf buntes Farbenspiel, drum habe ich diese schwarz gelassen)



Auswahl einer ZielDV und Drop



Nun nochmal kontrolle und bestätigen für Übertragen oder Abbruch.
Statusmessage nach dem Übertragen

Methode	Typ	Value	Beschreibung	Empfänger	Sender	
DEVICENAME	HM-LC-Sw1		<...	Standardverknüpfung Vir...	PEQ0480086:1 (Aktor1)	BidCoS-RF:1 (HM-RCV-50
DEVICETYPE	SWITCH		IS...	Standardverknüpfung Fe...	PEQ0480086:1 (Aktor1)	LEQ1252736:1 (HM-Sec-RJ
SERIENNUMMER	MEQ0330450...			LEQ1319211:1 (TEST)	LEQ1319211:1 (TEST)	LEQ1319211:2 (TEST)
---GERÄTEINTERN---				LEQ1319211:2 (TEST)	LEQ1319211:2 (TEST)	
PARAMSET LINK	string	<value><s...		ÜÜ345	MEQ0330450:1 (HM-LC-Sw1FBU-FM MEQ0...	MEQ0330450:1 (HM-LC-SW
L	LONG_ACTION_TYPE	14	jajajj	MEQ0330450:1 (HM-LC-Sw1FBU-FM MEQ0...	MEQ0330450:2 (HM-LC-Sw	
L	LONG_COND_VALUE_HI	14		MEQ0755132:1 (Außentemperatur)	MEQ0755132:1 (Außentem	
L	LONG_COND_VALUE_LO	14		PEQ0480086:1 (Aktor1)	PEQ0480086:1 (Aktor1)	
L	LONG_CT_OFF	14	hung hohoho[]	000A9709A0DA37:4 (TestWITH-3)	0000D7099A5285:1 (SWDO	
L	LONG_CT_OFFDELAY	14	40...	0025DA49A76350:10 (RolloTest)	0025DA49A76350:1 (Rollo	
L	LONG_CT_ON	14	DE...	0025DA49A76350:10 (RolloTest)	61a7D8A98C17B3:47 (Hml	
L	LONG_CT_ONDELAY	14				
L	LONG_JT_OFF	14				
L	LONG_JT_OFFDELAY	14				
L	LONG_JT_ON	14				
L	LONG_JT_ONDELAY	14				
L	LONG_MULTIEXECUTE	14				
L	LONG_OFF_DELAY_TIME	double				
L	LONG_OFF_TIME_MODE	14				
L	LONG_ON_DELAY_TIME	double				
L	LONG_ON_TIME_MODE	14				
L	SHORT_ACTION_TYPE	14				
L	SHORT_COND_VALUE_HI	14				
L	SHORT_COND_VALUE_LO	14				
L	SHORT_CT_OFF		Markiere alle ParamsetLinks			
L	SHORT_CT_OFFDELAY					
L	SHORT_CT_ON	14				
L	SHORT_CT_ONDELAY	14				

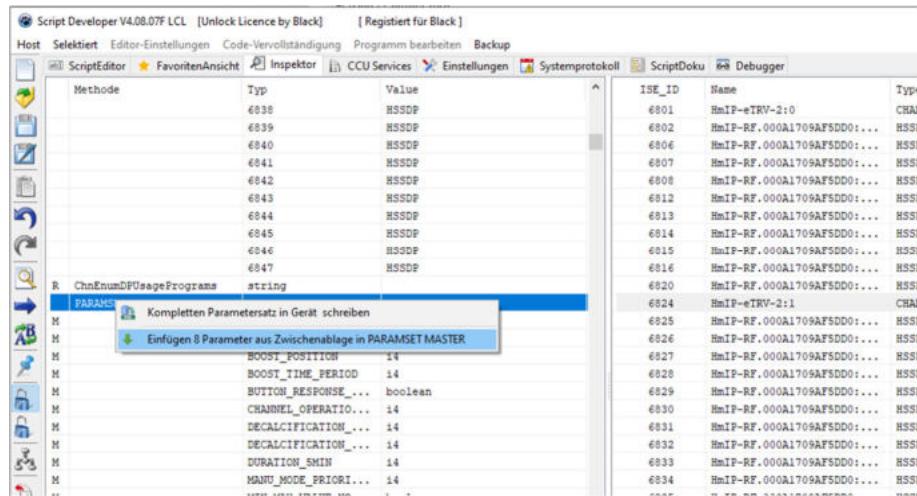
Sollen Alle DV Parameter ausgewählt werden, rechte Maustaste im Detailsview und Markiere alle ParamsetLinks

7.5.2 Einzelne Parameter markieren und in Link oder Masterparameter einfügen

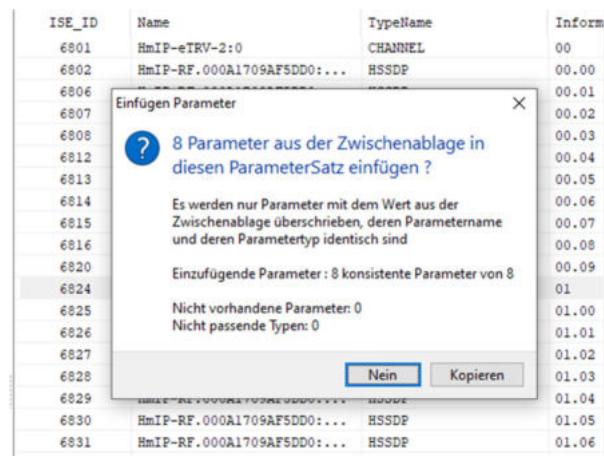
Vorgehensweise: Parameter (link oder Master markieren, rechte Maustaste und

Methode	Typ	Value	ISE_ID	Name
4634	HSSDP		4592	HmIP-eTRV-1:0
4635	HSSDP		4593	HmIP-RF.000A18A
4636	HSSDP		4597	HmIP-RF.000A18A
4637	HSSDP		4598	HmIP-RF.000A18A
4638	HSSDP		4599	HmIP-RF.000A18A
6857	VARDP		4603	HmIP-RF.000A18A
R ChnEnumDPUsagePrograms	string	6501,6558	4604	HmIP-RF.000A18A
6501	PROGRAM		4605	HmIP-RF.000A18A
6558	PROGRAM		4606	HmIP-RF.000A18A
PARAMSET MASTER	string	571 Parameter	4607	HmIP-RF.000A18A
M	ADAPTIVE_REGULATION	14	4611	HmIP-RF.000A18A
M	BOOST_AFTER_WIND...	boolean	4615	HmIP-eTRV-1:1
M	BOOST_POSITION	14	4616	HmIP-RF.000A18A
M	BOOST_TIME_PERIOD	14	4617	HmIP-RF.000A18A
M	BUTTON_RESPONSE...	boolean	4618	HmIP-RF.000A18A
M	CHANNEL_OPERATION...	14	4619	HmIP-RF.000A18A
M	DECALCIFICATION...	14	4620	HmIP-RF.000A18A
M	DECALCIFICATION...	14	4621	HmIP-RF.000A18A
M	Kopieren 8 Master-Parameter in Zwischenablage		4622	HmIP-RF.000A18A
M	MANU_MODE_PRIORI...	14	4623	HmIP-RF.000A18A
M	MIN_MAX_VALUE_NO...	boolean	4624	HmIP-RF.000A18A
M	OPTIMUM_START_STOP	boolean	4625	HmIP-RF.000A18A
M	PI_ENDTIME_FRIDAY_1	14	4626	HmIP-RF.000A18A
M	PI_ENDTIME_FRIDA...	14	4627	HmIP-RF.000A18A
M	PI_ENDTIME_FRIDA...	14	4628	HmIP-RF.000A18A
			4629	HmTP-RF.000A18A

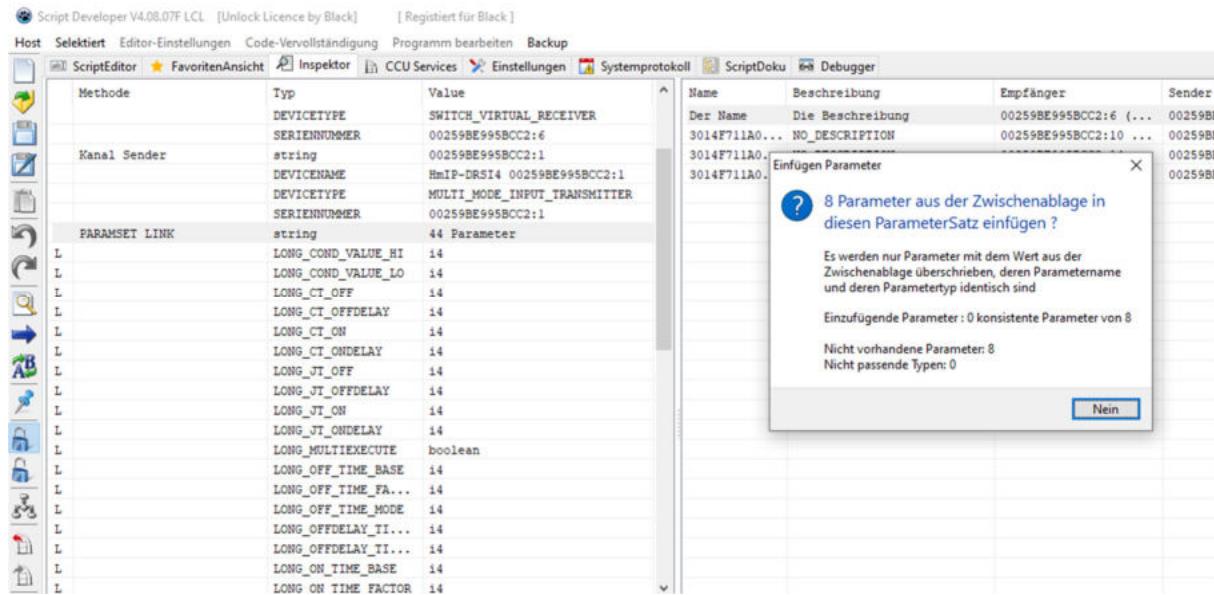
dann das Zieldevice oder Zielchannel auswählen (oder auch Direktverknüpfung) und dort rechte Maustaste auf Paramset Master bzw Paramset Link:



Es erfolgt noch eine Sicherheitsabfrage, wo auch drauf hingewiesen wird, welche Parameter im Ziel nicht vorhanden sind oder welche vorhanden sind aber mit abweichendem Typ: diese werden natürlich nicht mitkopiert.

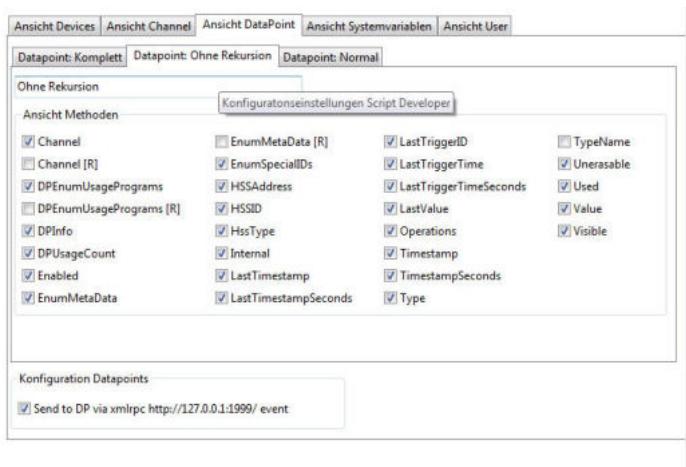


gibt es keine stimmige Übereinstimmung (in dem Beispiel markierte Masterparameter und versuch diese in eine DV einzufügen) und es keine Übereinstimmung gibt, so wird die Kopieren Option auch gar nicht angeboten.



7.6 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 5 braucht es dafür auch)

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

(Wenn die Freigabebedingungen passen)

Dieses Feature ist extrem nützlich aus eigener Erfahrung zum Testen von Programmen, wenn man z.B. Testtrigger und ähnliches von Sensoren prüfen will. Man muss hier nicht z.B mit dem Fön auf einen Thermostaten blasen, damit dieser über 35 Grad geht, einfach den DP anklicken, Value auswählen und die Zahl ändern. Wert übernehmen. (s.a. Kapitel 7.1)

7.7 Querverweise zu Datenpunkten in WebUI-Programmen

Es ist möglich, zu HSSDP, VarDP und AlarmDPs selektiv die Verwendung in WEBUI Programmen zu suchen.

ID	Name	Type	Status	Spesial	ISE_ID	Name	TypeName	Information
67	AlegerMediaAnzeigeger	VarDP			67	AlegerMediaAnzeigeger	VarDP	
68	AlegerTechnikAnzeigeger	VarDP			68	AlegerTechnikAnzeigeger	VarDP	
1433	SYS_01_BOOT	VarDP			1433	SYS_01_BOOT	VarDP	

Als Trigger: der Datenpunkt muss im Bedingungsteil als „bei Änderung“ oder „bei Aktualisierung“ verwendet werden.

Nur prüfen: der Datenpunkt muss im Bedingungsteil als „nur prüfen“ verwendet werden

Als Egal wie in Bedingungen: der Datenpunkt muss im Bedingungsteil vorkommen

Im Zuweisungsteil: Der Datenpunkt wird im Dann oder Sonst Teil verwendet

Egal wo: Der Datenpunkt muss nur irgendwo im WebUI Programm vorkommen

Als Ergebnis erscheinen in der Listenauswahl die Programme, wo der selektierte Datenpunkt unter der angewählten Bedingung vorkommt.

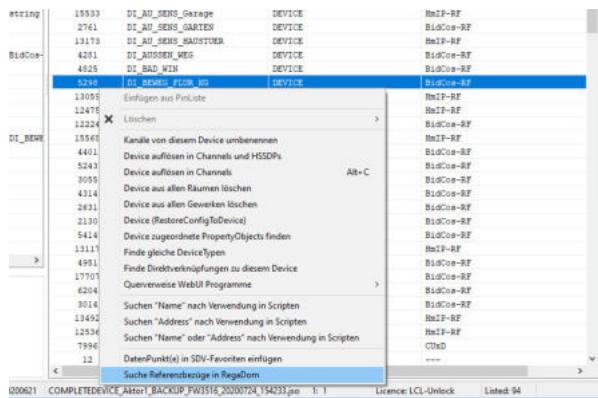
ISE_ID	Name	TypeName	Information
8632	Demo	PROGRAM	DemoTest
1433	SYS_01_BOOT	PROGRAM	CCU im Boot

7.8 Suche Referenzbezüge in Regadom

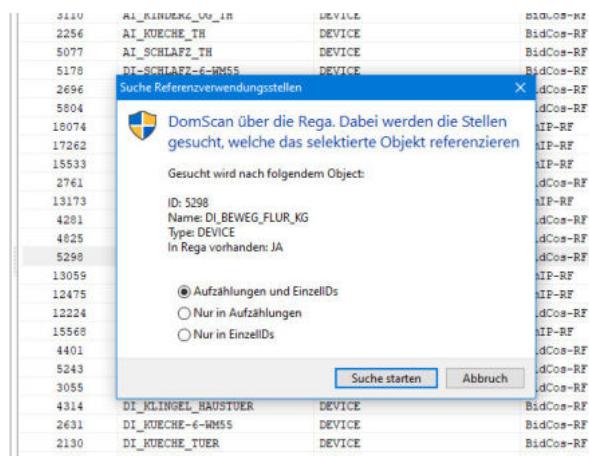
Ein Typischer usecase bei Geisterobjekten oder falschen Bezügen ist Möglichkeit, alle Objekte finden zu können, die irgendwie auf dieses Objekt referenzieren.

Beispielsweise:

In welchen Objekten wird überall auf ein bestimmtes Objekt bezogen, hier mal in dem Beispiel, wo überall wird auf das selektierte Device referenziert.



Dann erfolgt Rückfrage, wo überall gesucht werden soll:



Aufzählungen: Es wird nur in den Typischen Aufzählungen gesucht.

EinzellIDs: Es wird nur in referenzen gesucht, z.B. LeftVal in Singleconditions oder Channel in Datenpunkten

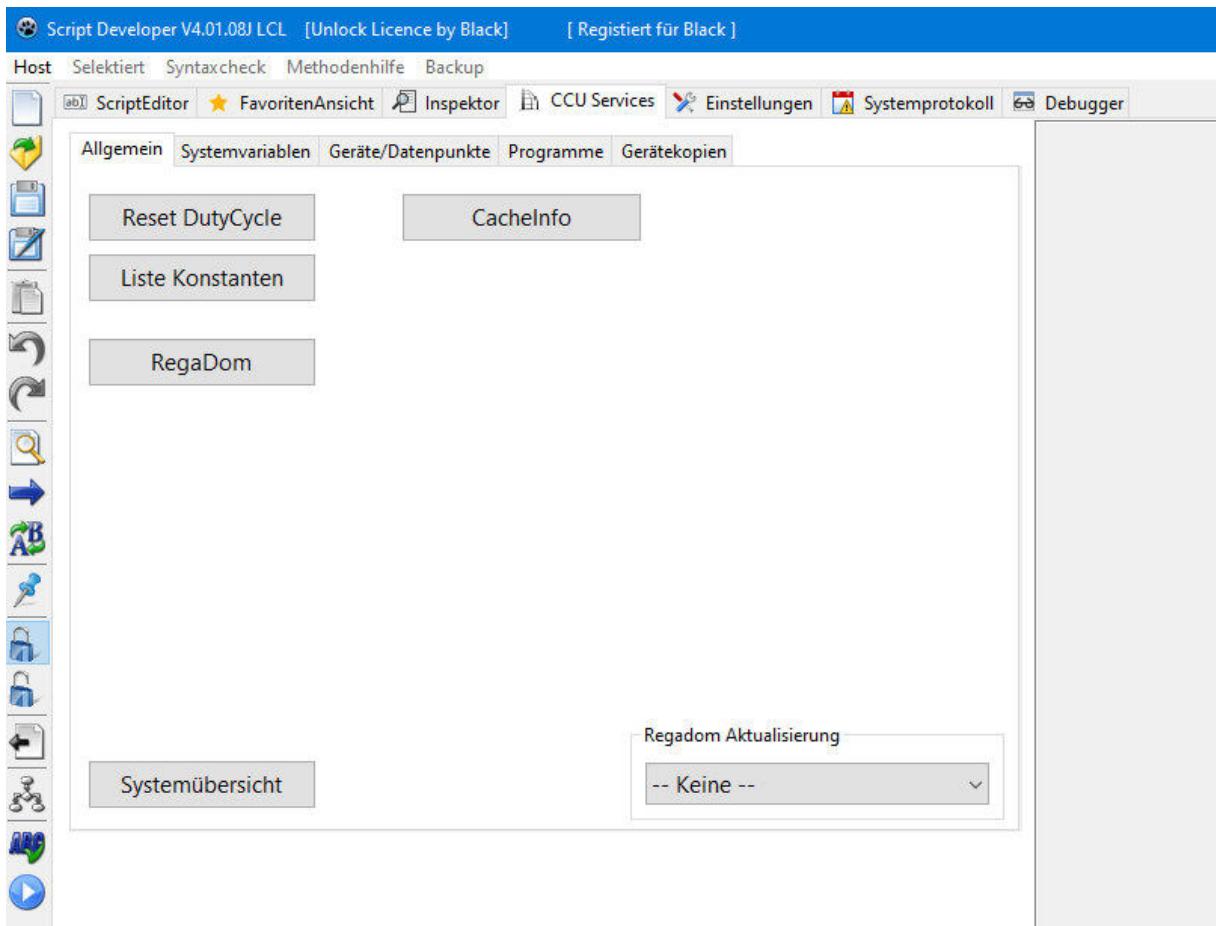
Aufzählungen und EinzellIDs: Es wird versucht die komplette rega durchzuscanen nach irgendwelchen Referenzen auf diese ID (hier 5298, das gewählte Device)

Als Ergebnis bekommt man in der Listenansicht dann eine Liste aller Datenpunkte, die irgendwie auf diese ID referenzieren.

mprotokoll Debugger			
ISE_ID	Name	TypeName	Information
3	Root devices	DEVICES	EnumIDs
5299	DI_BEWEG_FLUR_KG:0	CHANNEL	Device
5328	DI_BEWEG_FLUR_KG:1	CHANNEL	Device

Hier in dem Fall ist alles normal, bei einem Geister Objekt muss nun alles Gefundene durchgegangen und mit menschlicher Intelligenz analysiert werden.

8 Diagnosebild



Bei den meisten Korrekturläufen werden folgende Aktionen vorgeschlagen:

Nur prüfen: Der SDV macht nichts ausser einer Prüfung und Ausgabe der Ergebnisse im Ausgabefeld.

Prüfen und Ergebnis in den Inspektor: Auch hier nur eine Prüfung, aber gefundene Objekte werden zur weiteren Untersuchung in die Detailansicht des Inspektors geladen.

Prüfen und korrigieren: Bei Fehlern, die der SDV auch korrigieren kann, werden diese wie unter nur prüfen im Ausgabefeld dargestellt, gleichzeitig versucht der SDV eine Reparatur

8.1 Allgemein

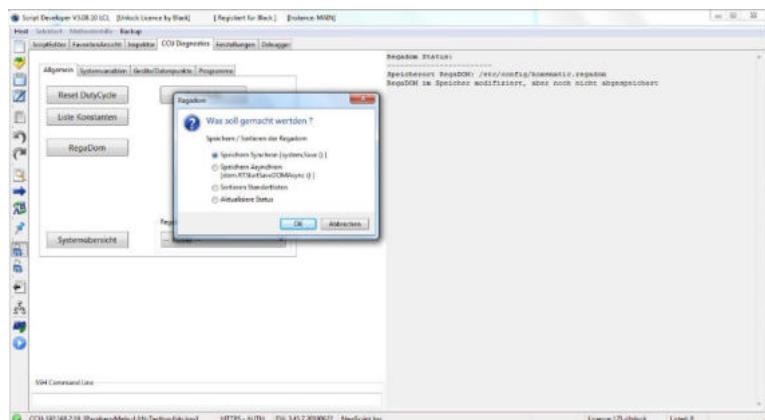
Reiter für Allgemeine Einstellungen

8.1.1 Liste Konstanten

Alle Systemkonstanten der Rega werden dargestellt

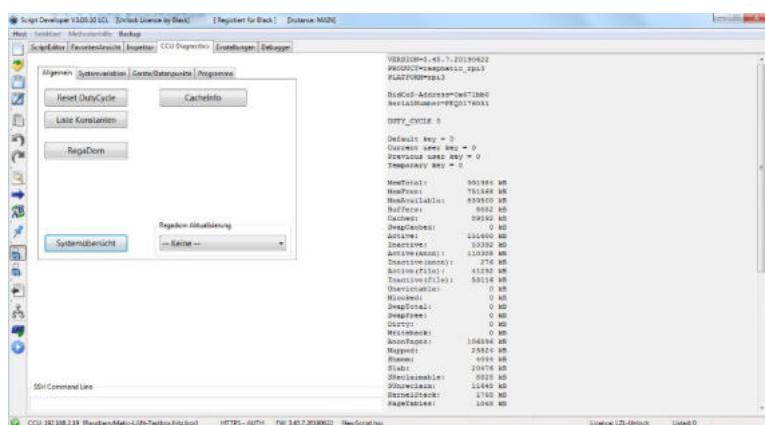
8.1.2 Regadom

Status der Rega abrufen und Speichern anstoßen



8.1.3 Systemübersicht

Allgemeine Information über die CCU

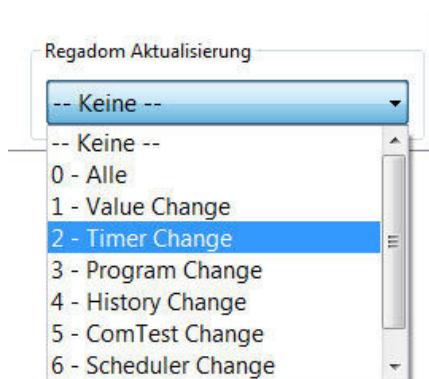


8.1.4 CacheInfo

Darstellung des CacheInhaltes der CCU

8.1.5 Regadom Aktualisierung

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



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

8.1 SSH Funktionalität

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

Was ist dazu nötig ?

In der SDV Ini müssen die nötigen Schlüssel eingegeben sein.

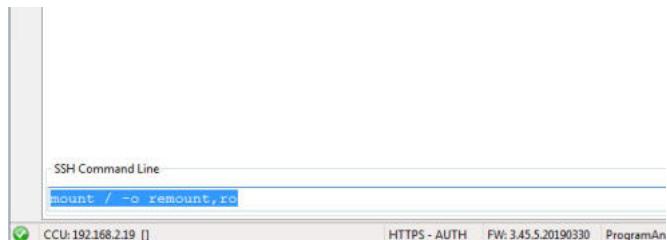
```
[CCU1]
IP=192.168.2.xx
USERNAME=xxxx
PASSWORD=yyyy
USEHTTPS=true
SSHUSERNAME=root      Bestandsnutzer: diese schlüssel hinzufügen
SSHPW=sshpw
```

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

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

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

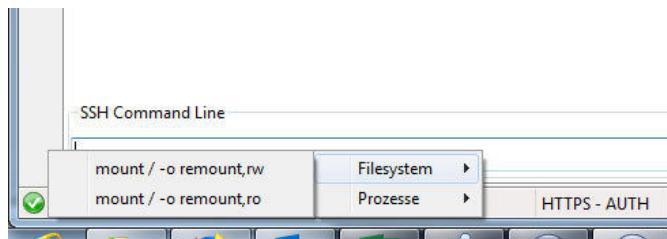
In dieser Zeile lassen sich nun einfache SSH Kommandos ausführen.



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

The screenshot shows the Script Developer V1.01.11.21 interface. The top menu bar includes 'File', 'Select...', 'Methodology', 'Backup', 'Unlock License by Block', and 'Register New Block'. Below the menu is a toolbar with icons for 'ScriptEditor', 'Inspector', 'CCU Diagnostic', 'Einstellungen', and 'Debugger'. A central window titled 'Restore PräsentzVar 599' displays a file tree. To the left of the tree are several buttons: 'Check Internal Syntax', 'Check Program Structure', 'Logic Konstanten', 'Check Condition Konstanten', 'Check MetaTags SysVar', and 'Check Channel SysVar'. At the bottom left is an 'Aktualisieren' button. On the far left, there's a vertical toolbar with icons for file operations like Open, Save, Print, and Help. The bottom left corner shows 'SSH Command Line'. The bottom right corner displays system information: CCU 102.108.2.19 [1], HTTPS - AUTH Fw 2.41.1.23080209 Progranalysator2/Hci, 3.19, License L21-Unlock, Listed 5.

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



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

```
>>>top  
ThreadKilled
```

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

INI Schlüssel SSHLINK=false

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

USERACCESS=1+

USERLOGIN=

Wichtig ist das + !!!!!!!

rennt er dabei gegen Access denied.

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

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

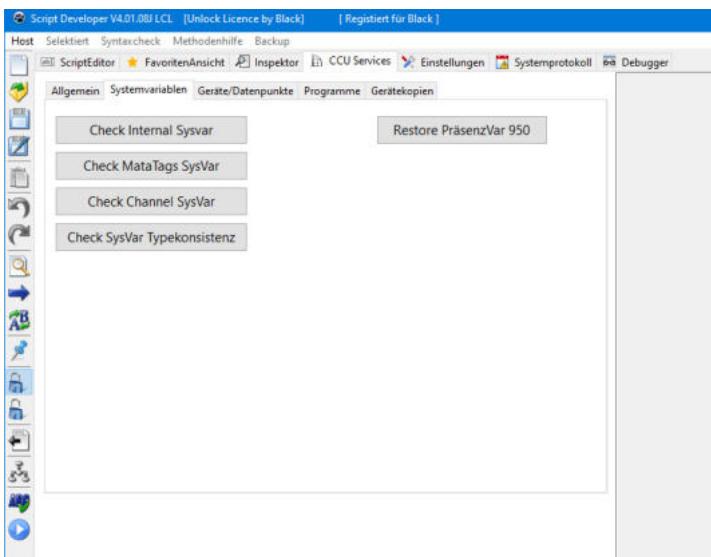
```
plink -ssh root@IPderCCU -pw SSHPASSWORT top - n1
```

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

damit ist aber wieder Windows im Spiel.

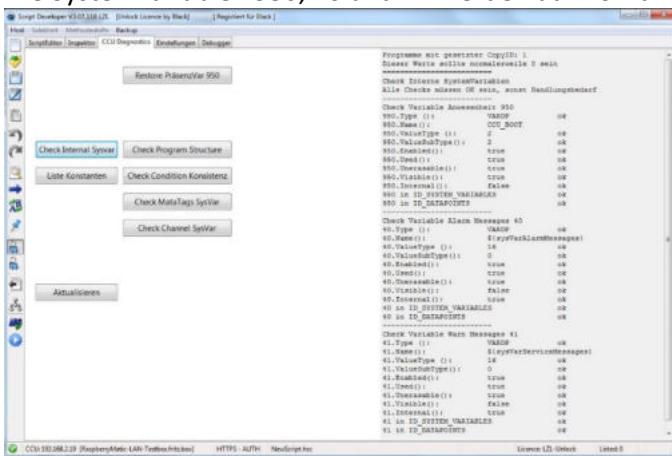
Wer weder CuxD den Access setzen möchte noch plink benutzen will, der kann den SDV nutzen bis auf die SSH Funktionalität.

8.2 Diagnosen Systemvariablen



8.2.2 Check Internal Sysvar

Die Systemvariablen 950, 40 und 41 werden auf Vorhandensein und richtige Parameter überprüft.

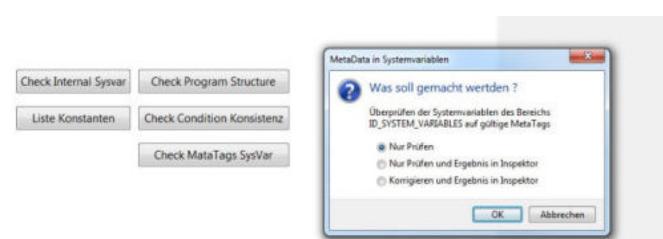


8.2.3 Check metaTags Sysvar

Systemvariablen werden auf korrekte Metadaten hin überprüft. Der SDV nimmt von der Prüfung aber selbstdefinierte Metatags, welche mit _sdv_ beginnen, aus.

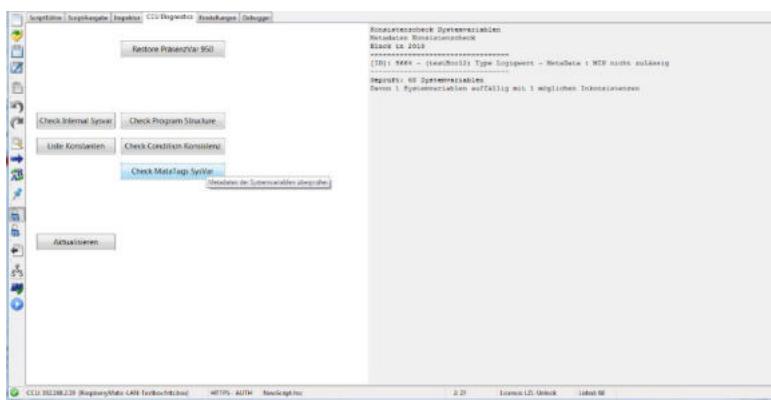
Dieser Menüpunkt überprüft die Systemdaten auf Konsistenz von dem Typ, welcher durch ValueType und ValueSubtype spezifiziert ist und den Einträgen unter EnumMetadata.

Druck erzeugt folgendes Menü:



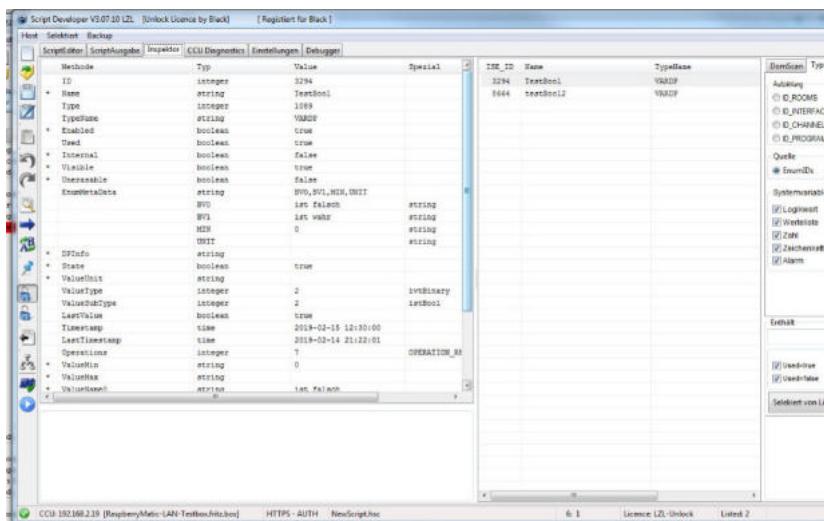
Punkt 1: nur Prüfen

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



Punkt 2: Nur Prüfen und Ergebnis in Explorer

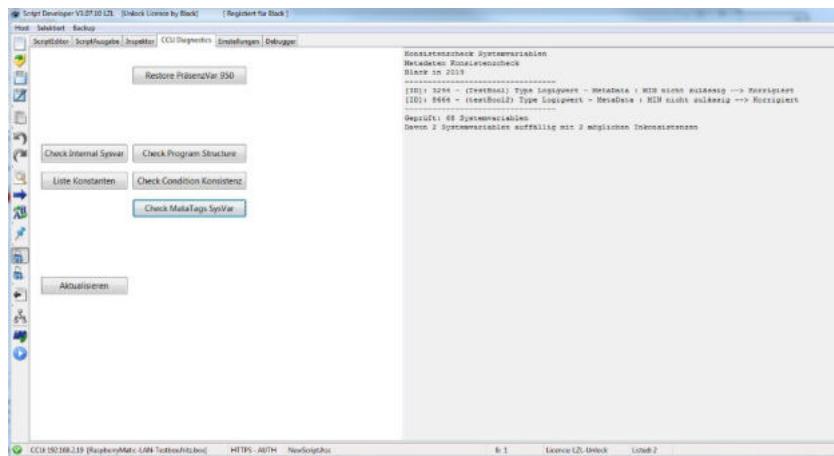
Genau wie Punkt 1, es wird eine Überprüfung gemacht, zusätzlich wird das Ergebnis noch in die Auswahlliste des Inspektors geladen. Dort kann überprüft und gegenbenenfalls auch manuell korrigiert werden.



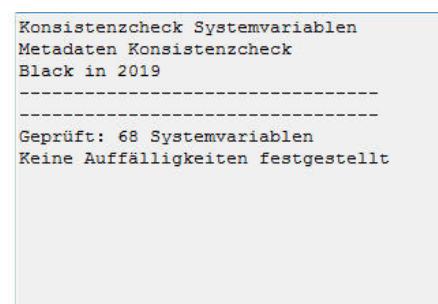
Punkt 3: Korrigieren und Ergebnis in den Inspektor

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

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

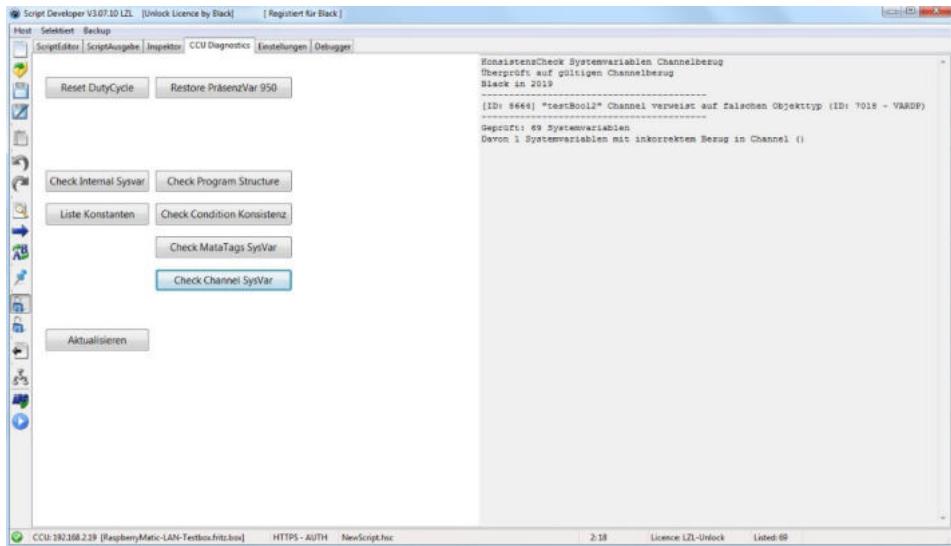


Ein nochmaliger Durchlauf bringt dann die Fehlerfrei-Nachricht.



8.2.3 Check Channel Sysvar

Prüf und opriionaler Korrekturlauf ob eine Systemvariable mit einem Eintrag Channel auf wirklich unter DPs() des jeweiligen Channels eingehangen ist.



8.2.4 Check Sysvar Typkonsistenz

Es wird überprüft, ob Systemvariablen einen der erlaubten Typen haben.

Überprüft die Typekonsistenz von Systemvariablen:

OT_VARDP: Logikwert: Type() und ValueSubType

OT_VARDP: Zahl: Type() und ValueSubType

OT_VARDP: String: Type() und ValueSubType

OT_VARDP: WerteListe: Type() und ValueSubType

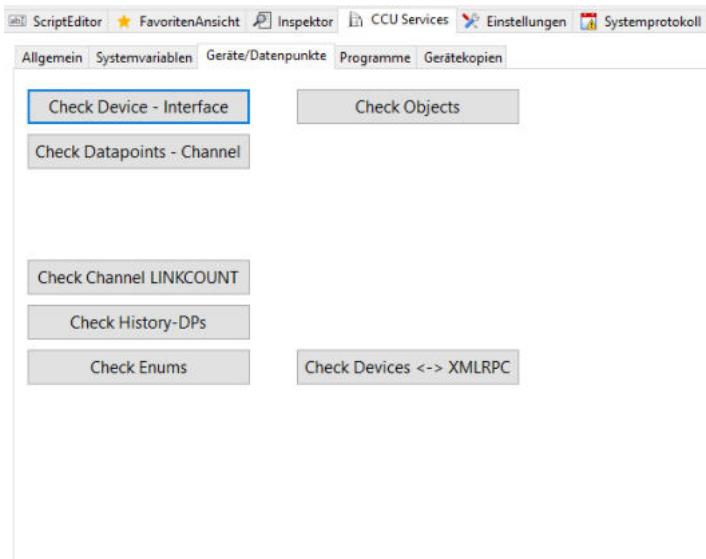
OT_VARDP: WerteListe: Vorhandensein von ValueList

OT_ALARMDP: Type() und ValueSubType

8.2.5 Restore Präsenzvar 950

Kann eine Abhanden gekommene Präsenz Systemvariable wieder rekonstruieren

8.3 Geräte Datenpunkte



8.3.1 Check Device-Interface

Analyse Interface Eintrag in Devices

Überprüfung aller Devices auf konsistenten Eintrag von der Interface ID in der Methode .Interface().
Da Interface() readonly ist, kann der SDV einen eventuellen Fehler nicht reparieren.

8.3.2 Check Datapoints Channel

Testroutine für:

Überprüft HSSDPs, ob diese korrekt in datapoints
eingehangen sind, ob die Channelbezüge korrekt sind
und die HssAdress passt

8.3.3 Check Channel Linkcounts

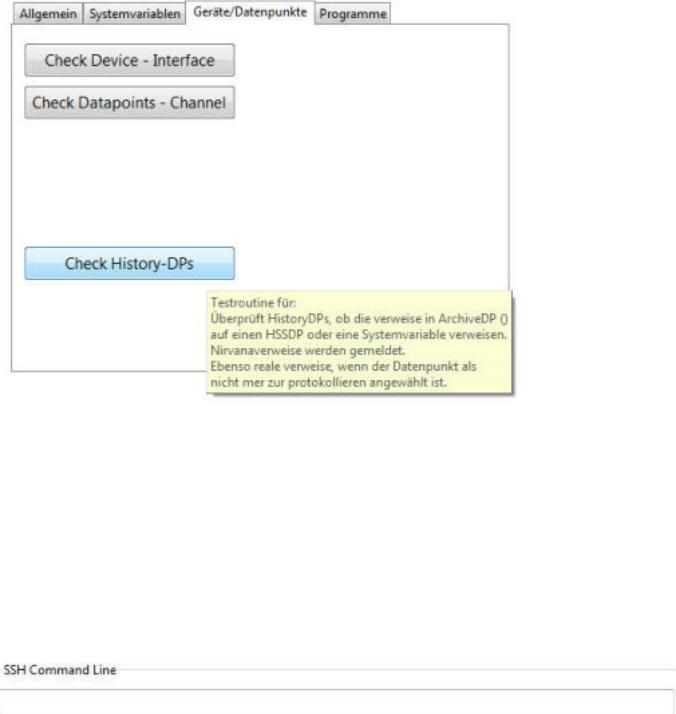
LINKCOUNT enthält die Anzahl an Direktverknüpfungen eines Kanals. Mit der Zeit kann dieser Wert inkonsistent werden. Diese Routine prüft und korrigiert auf Wunsch den LinkCount Eintrag in den Metadaten der Channels

8.3.4 Check History DPs

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

Kann man so machen, muss man aber nicht.

Mit dem Punkt



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



Nur Anzeige:

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

Anzeige und List im Inspektor

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

Anzeige und Löschen

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

Lauf alles in Ordnung:

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

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

```
Analyse History DPs
Black im August 2019
-----
HistoryDP 11208 verweist in ArchiveDP [6461] auf Systemvariable mit DPArchive ()=false. Empfehlung löschen.

Es wurden 12 HistoryDPs überprüft
Dabei wurden 1 Inkonsistenz(en) festgestellt
Davon wurden 0 Inkonsistenz(en) korrigiert
```

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

8.3.5 Check Enums

Testroutine für:

Überprüft Enums auf folgende Konsistenz:

a:) EnumTypes ()

1. ID_ROOMS müssen Type etRoom
 2. ID_FUNCTIONS müssen Type etFunction
 3. ID_FAVORITE müssen Type etFavorite
- haben. Korrektur ist möglich

b:) DomScan nach Enums gefundene Enums

1. etRoom müssen in ID_ROOMS
 2. etFunction müssen in ID_FUNCTIONS
 3. etFavorite müssen in ID_FAVORITE
- gelistet sein

gleichzeitig wird noch nach etUnknown gesucht

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

8.3.6 Check Objects

Komplexe Testroutine für:

Überprüfung richtige Objekttypen in Aufzählungen

Gegenprüfung von beispielsweise Channel - Datenpunkt auf korrekten Verweis

Überprüfung richtiger Objekttyp von referenzieren Objekten

Überprüfung korrekter Address und HSSAddress bei Geräten

Es wird immer mit Test und Gegentest gearbeitet

Beispielhafter Testlauf:

```
Analyse Aufzählungstypen
Erstellt vom V4.01.08J LCL
Black in 2020
-----
Überprüfung 23 Objekte in Aufzählung : ID_DEVICES
Überprüfung 4 Objekte in Aufzählung : ID_INTERFACES
Überprüfung 16 Objekte in Aufzählung : ID_ROOMS
Überprüfung 10 Objekte in Aufzählung : ID_FUNCTIONS
Überprüfung 15 Objekte in Aufzählung : ID_FAVORITES
Überprüfung 247 Objekte in Aufzählung : ID_CHANNELS
Überprüfung 98 Objekte in Aufzählung : ID_SYSTEM_VARIABLES
Überprüfung 1615 Objekte in Aufzählung : ID_DATAPOINTS
Überprüfung 19 Objekte in Aufzählung : ID_PROGRAMS
Überprüfung 12 Objekte in Aufzählung : ID_CALENDARDFS
Überprüfung 98 Objekte in Aufzählung : ID_SERVICES
Überprüfung 10 Objekte in Aufzählung : ID_RULES
Überprüfung 38 Objekte in Aufzählung : ID_CONDITIONS
Überprüfung 125 Objekte in Aufzählung : ID_SCONDITIONS
Überprüfung 0 Objekte in Aufzählung : ID_DESTINATIONS
Überprüfung 27 Objekte in Aufzählung : ID_SDESTINATIONS
Überprüfung 8 Objekte in Aufzählung : ID_USERS
-----
Ausgeführte Prüfungen: 12491
-----
Step 2 - DomScan über die Rega mit Abgleich: Objekte in Listen
Scanbereich RegaDom IseID (1) bis IseID (65535)
    Object [19580,SDV_FA] vom Typ ENUM ist vom Typ etUnknown
    Object [19581,SDV_CHN] vom Typ ENUM ist vom Typ etUnknown
    Object [19582,SDV_DP] vom Typ ENUM ist vom Typ etUnknown
    Object [19658,SDV_FA] vom Typ ENUM ist vom Typ etUnknown
    Object [19659,SDV_CHN] vom Typ ENUM ist vom Typ etUnknown
    Object [19660,SDV_DP] vom Typ ENUM ist vom Typ etUnknown
    Object [19661,SDV_FA] vom Typ ENUM ist vom Typ etUnknown
    Object [19662,SDV_CHN] vom Typ ENUM ist vom Typ etUnknown
    Object [19663,SDV_DP] vom Typ ENUM ist vom Typ etUnknown
    Object [19728,SDV_FA] vom Typ ENUM ist vom Typ etUnknown
    Object [19729,SDV_CHN] vom Typ ENUM ist vom Typ etUnknown
    Object [19730,SDV_DP] vom Typ ENUM ist vom Typ etUnknown
    Object [19731,SDV_FA] vom Typ ENUM ist vom Typ etUnknown
    Object [19732,SDV_CHN] vom Typ ENUM ist vom Typ etUnknown
    Object [19733,SDV_DP] vom Typ ENUM ist vom Typ etUnknown
    Object [19734,SDV_FA] vom Typ ENUM ist vom Typ etUnknown
    Object [19735,SDV_CHN] vom Typ ENUM ist vom Typ etUnknown
    Object [19736,SDV_DP] vom Typ ENUM ist vom Typ etUnknown
-----
In der RegaDom überprüfte Objekte : 2532
Höchste verwendete ID in RegaDom : 20270
GeisterObjekte ohne Listenzuordnung: 18
```

Hier finden sich Geister Testobjekte des Types OT_ENUM (ist das testsystem)

8.3.7 Check Devices <--> XMLRPC

Folgendes wird überprüft:

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

Aufgelistet werden:

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

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

Fehlerfrei sollte so ein Ergebnis kommen:

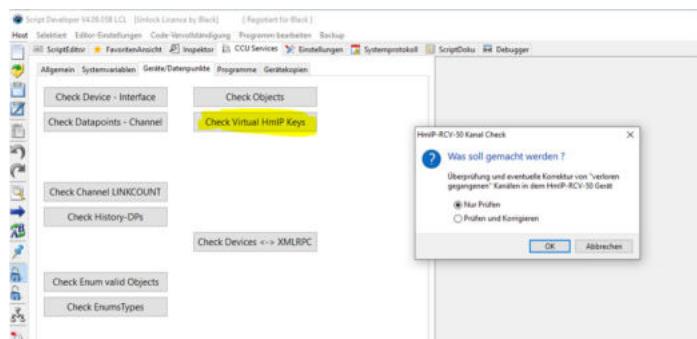
```
Folgende Geräte wurden an den Schnittstellenprozessen gefunden:
    CUXD      : 2
    BidCos-RF : 9
    BidCos-Wired : 0
    HmIP-RF   : 9
    VirtualDevices: 2
Abfrage der Geräte aus der RegaHSS
RegaScan über Ise-Bereich von ID (1) bis ID (65535)
Folgende Geräte wurden in der Rega gefunden:
    CUXD      : 2
    BidCos-RF : 9
    BidCos-Wired : 0
    HmIP-RF   : 9
    VirtualDevices: 2
Alle OT_DEVICE sind in ID_DEVICES gelistet
-----
Es gibt keine Inkonsistenz zwischen RegaDom und den Schnittstellenprozessen
```

8.3.8 Check Virtual HmIP Keys.

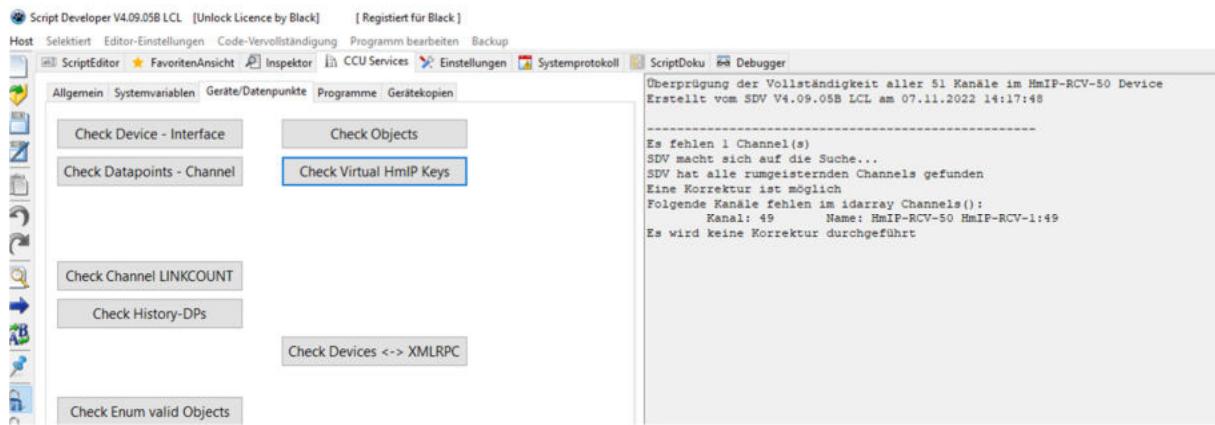
Da ja "magisch verschwundene" Kanäle aus der virtuellen HmIP Fernbedienung kein Einzelfall sind, habe ich im SDV eine Möglichkeit programmiert, das Gerät auf diesen Effekt zu prüfen und wenn das HmIP-RCV-50 wirklich einen Teil seiner Kanäle vergessen hat, dieses automatisiert wieder zu korrigieren.

Trotz allem natürlich: VORHER BACKUP MACHEN

Es besteht die Möglichkeit nur prüfen oder prüfen und korrigieren

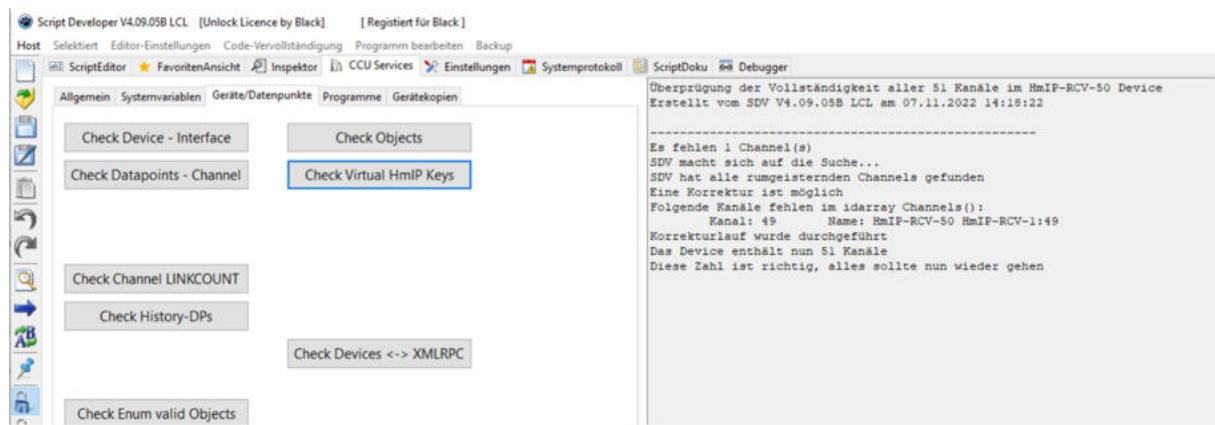


Nur Prüfen checkt nur, listet eventuell Fehler auf, korrigiert aber nichts
 Gibt es einen Fehler, erscheint dieses:

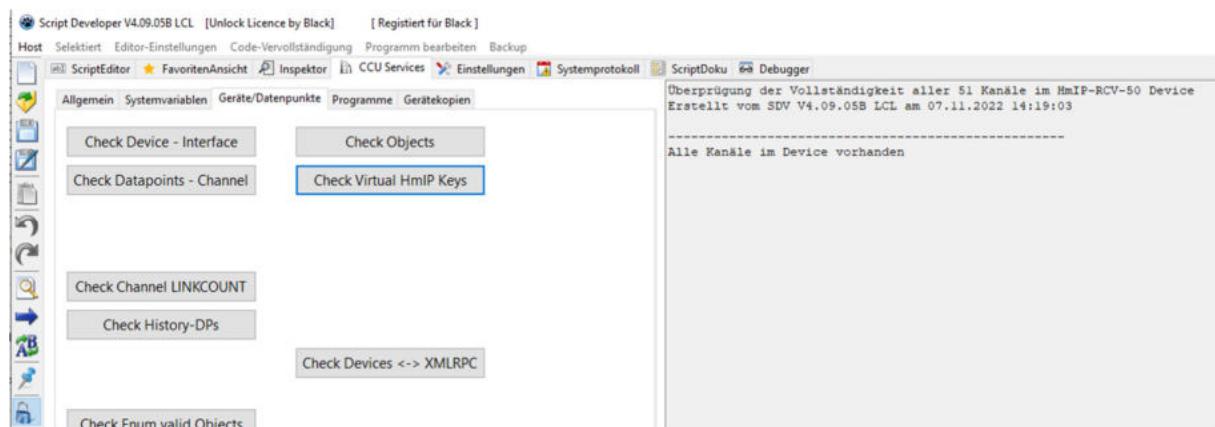


Der SDV prüft, ob alle benötigten Kanäle (er muss 51 finden) noch in der Rega rumgeistern,
 Reparaturen werden nur gemacht, wenn die Geister und Realobjekte vorhanden sind)

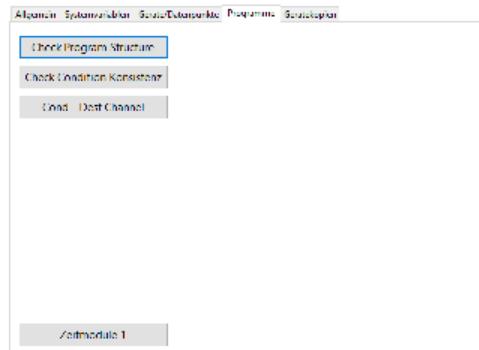
Eine Reparatur bastelt den verschwundenen Kanal wieder an die richtige Stelle zurück



Ein nochmaliger Lauf bringt dann die Meldung, keine Fehler mehr da

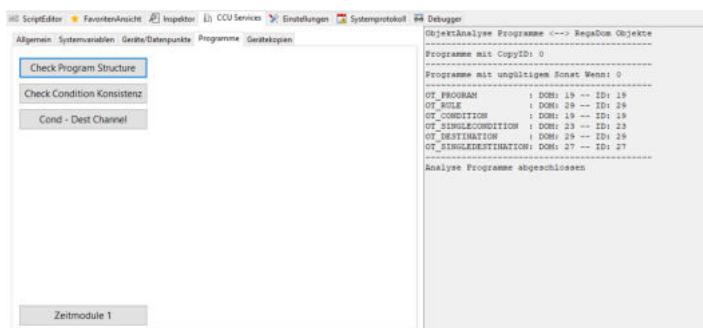


8.4 Programme



8.4.1 Check Program Structure

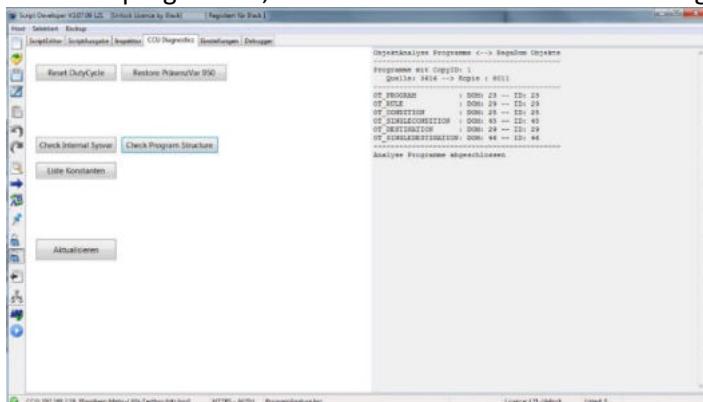
Mittels der Programmanalyse kann die Konsistenz der Programmobjekte überprüft werden. So sieht der Aufruf auf, wenn die Prüfroutine erfolgreich über die Programme gelaufen ist.



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.



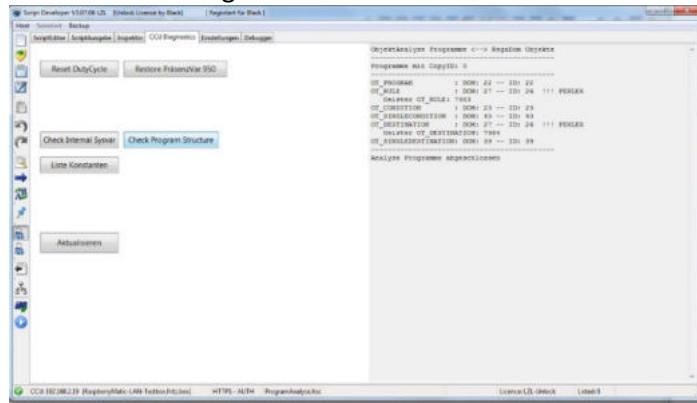
Hier wird das ursprüngliche Programm und die (Geisterkopie) daraus angezeigt. Der erste Versuch wäre: einen Reboot zu versuchen und zu überprüfen, ist der Eintrag des Geisterprogrammes weg oder nicht. Wenn nicht, kann manuell versucht werden, die Objekte der Kopie (nicht des Quellprogrammes) zu löschen

2. Programmfragmentreste

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

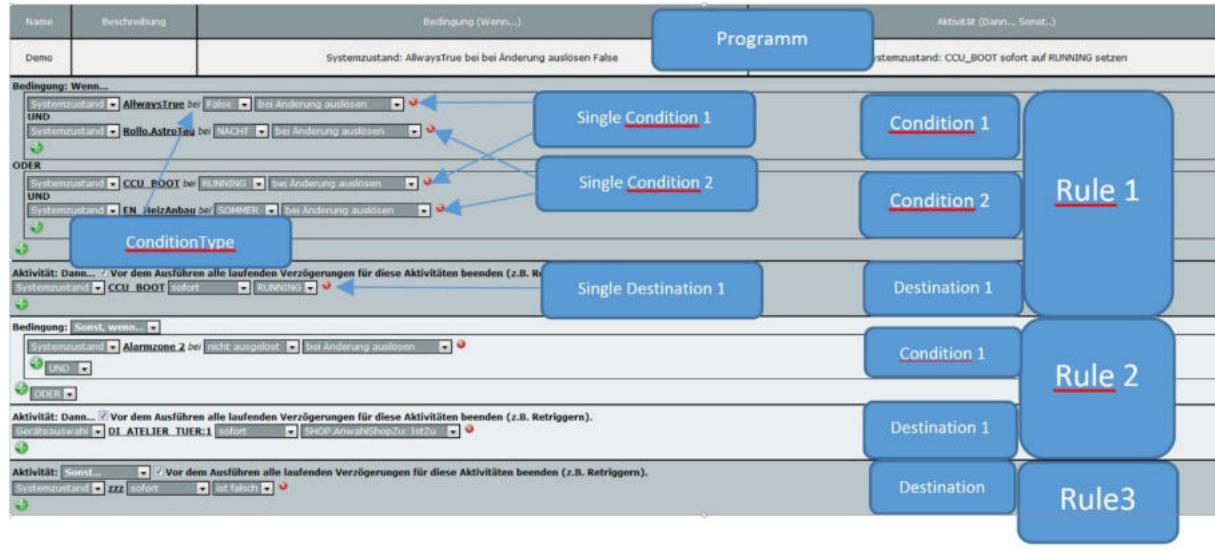
Provoziert habe ich es hier in dem Testfall, in dem ich ein RULE Object mal händisch angelegt habe.

Der Testlauf schlägt hier an:

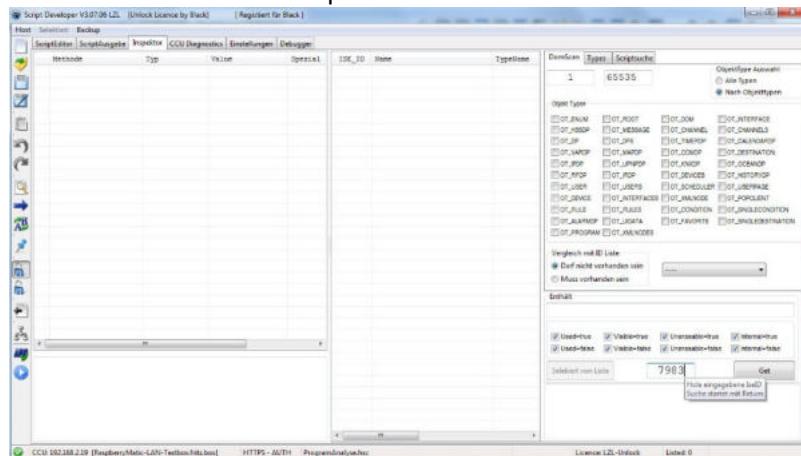


Es wird unter den Objecten OT_RULE und OT_DESTINATION angezeigt, dass sich in der Rega 2 Objekte tummeln, welche keinen Bezug zu dem Inhalt eines Programmobjektes haben. (2 deshalb, weil ein dom.CreateObject (OT_RULE) auch gleichzeitig das zu der Rule gehörende OT_DESTINATION Object erzeugt)

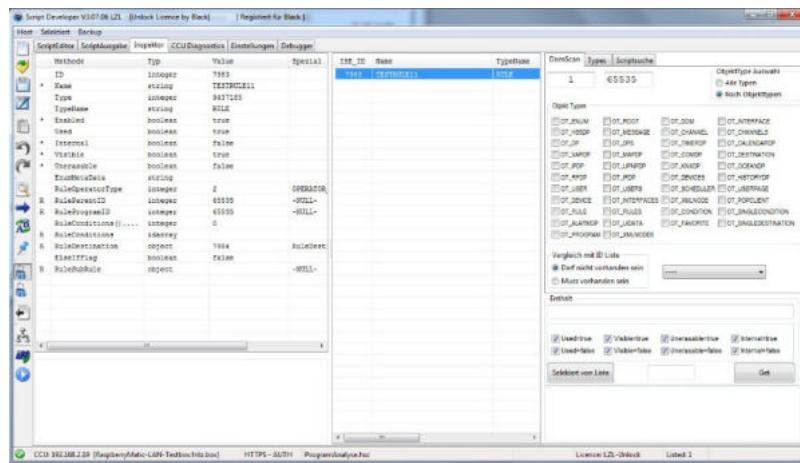
Ein Programm ist auf der CCU so Aufgebaut:



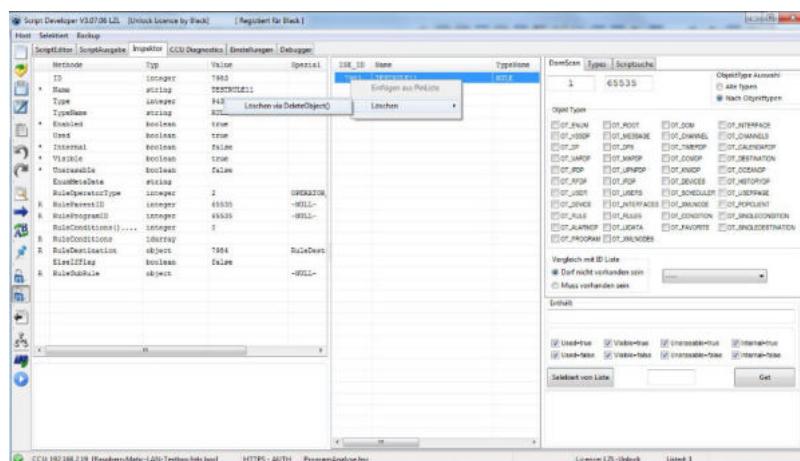
Die isID des gefundenen Objektes wird ausgegeben
diese kann man sich im Inspektor ansehen



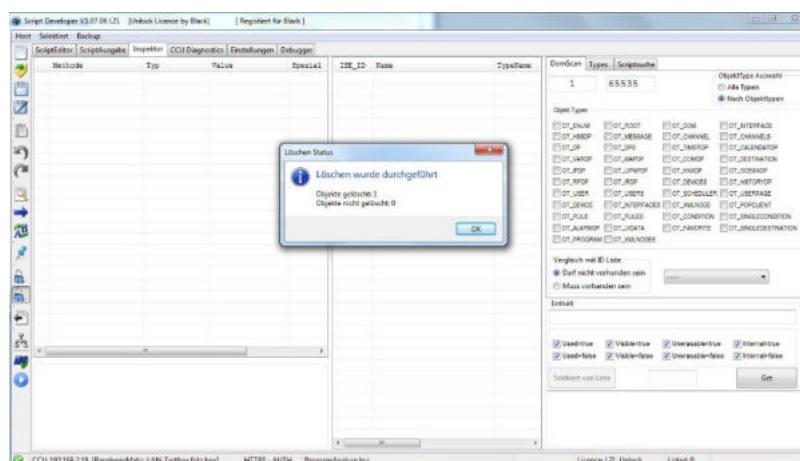
Dazu die gefundene ID in die Suchmaske eintragen und Enter drücken



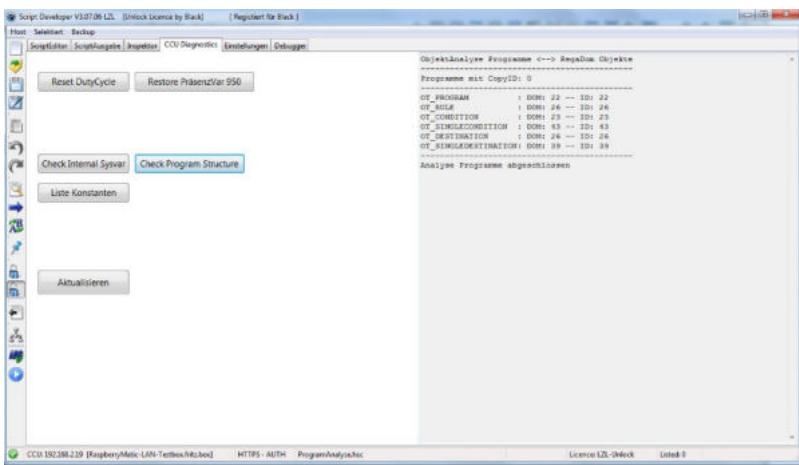
Man sieht, ein Object ohne Bezüge. es kann also entfernt werden. Löschen Freigeben, Programmunterobjekte müssen auch freigeben sein zum löschen, dann



es wurde gekillt



Die abschliessende Kontrolle zeigt:



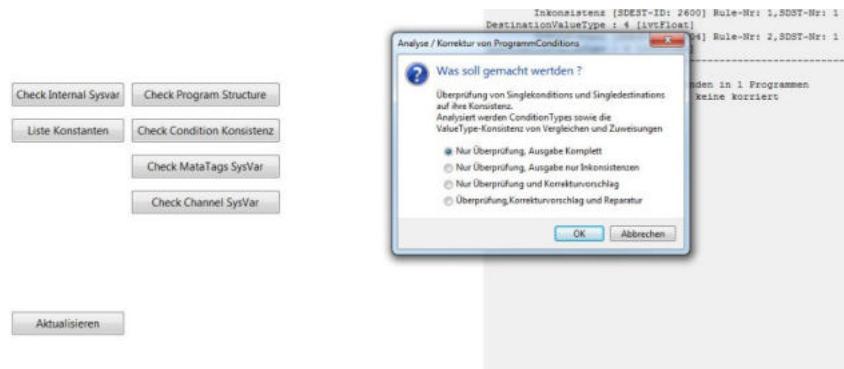
Es ist weg.

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

8.4.1 Check Condition Konsistenz.

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

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



Auswahl ist selbsterklärend.

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

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

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

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

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

Gepruft: 37 Programme
Dabei wurden 4 Inkonsistenzen gefunden in 3 Programmen
Von diesen 4 Inkonsistenzen wurden keine korrigiert
```

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

Wenn ja, wird im einem Korrekturlauf diese Inkonsistenzen beseitigt.

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

Es gibt kein UNDO, das geht nur über Restore

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

Analyse Program: Alarmtest [ID: 8528]
Inkonsistenz [SCND-ID: 8567] Rule-Nr: 1,CND-Nr: 1, SCND-Nr: 1 -- LeftVal: 2 [ivtBinary], RightVal1: 4 [ivtFloat]
Ursache Boolvergleich ivtBinary Links mit nicht ivtFloat Wert rechts
Umrechnen von rechts Wert 15.000000 [ivtFloat] in true [ivtBinary]
Kann vom SDV korrigiert werden
Korrigiert durch RightVal1ValType (ivtBinary)

Analyse Program: Stringtest [ID: 8622]
Inkonsistenz [SCND-ID: 9258] Rule-Nr: 1,CND-Nr: 1, SCND-Nr: 1 -- LeftVal: 20 [ivtString], RightVal1: 16 [ivtInteger]
Ursache Stringvergleich ivtString Links mit nicht ivtInteger Wert rechts
Umrechnen von rechts Wert 25 [ivtInteger] in 25 [ivtString]
Kann vom SDV korrigiert werden
Korrigiert durch RightVal1ValType (ivtString)

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

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

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

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

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

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

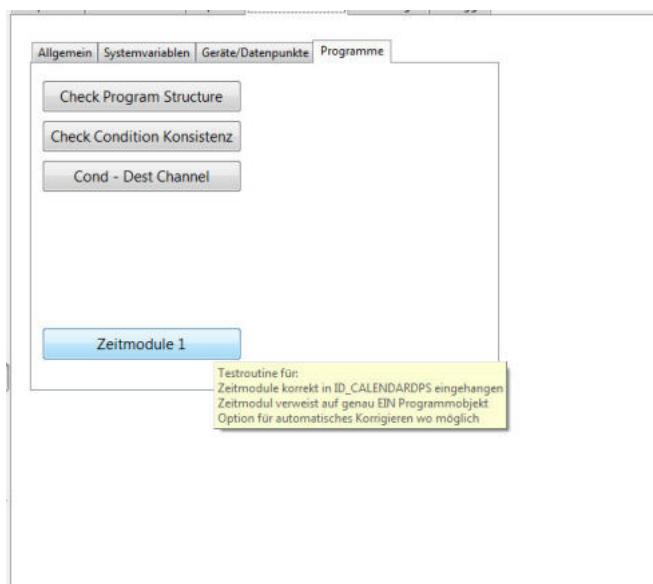
8.4.3 Cond- Destination Channel

Testroutine für:

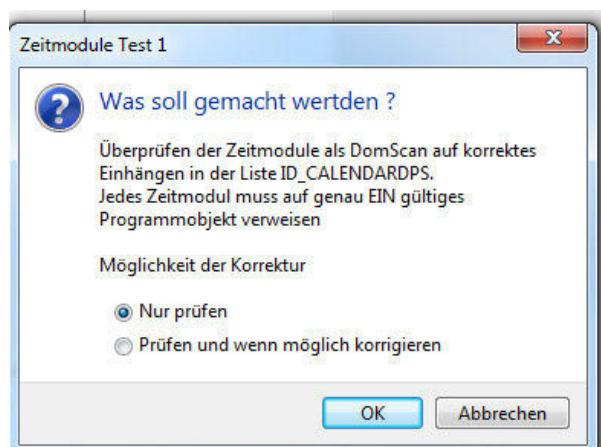
ConditionChannel und Destinationchannel korrekter Verweis auf das referenzierte Objekt bzw ID_ERROR sowie korrekte Verwendung von ivtSystemID bzw ivtObjectId.

Hintergrund: Eine Systemvariable steht in SCNDs und SDSTs normalerweise als ivtSystemID vermerkt, HSSDPs als ivtObjectId. Systemvariablen, die Kanälen zugewiesen werden, werden auch als ivtObjectId geführt. Der Testlauf überprüft auf die korrekte Anwendung des Objectbezeichners und kann dieses auch korrigieren.

8.4.4 Test und Korrektur Zeitmodule

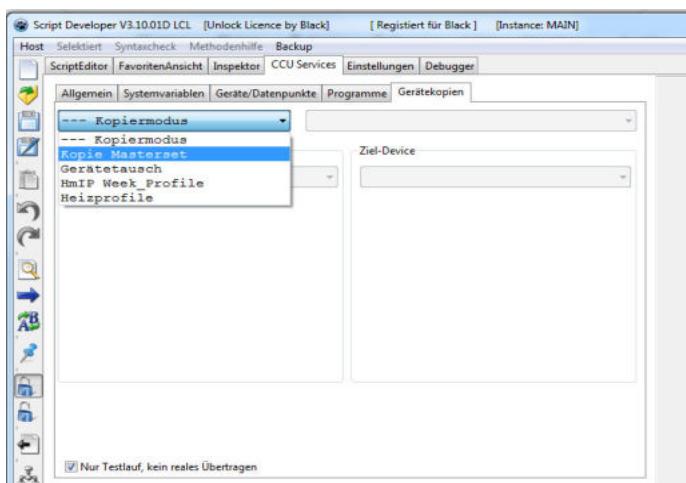


Seit Version 3.08.10 ist das Menü CCU Diagnostics in Reitern strukturiert. Unter Programme findet sich nun die Testoption Zeitmodule 1 . Hierbei wird geprüft auf korrektes Einhängen des Zeitmodules in ID_CALENDAR DPS, ob jedem Zeitmodul genau ein Programmobjekt zugeordnet ist. Korrektur kann da erfolgen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



9 Gerätekopieren

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



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

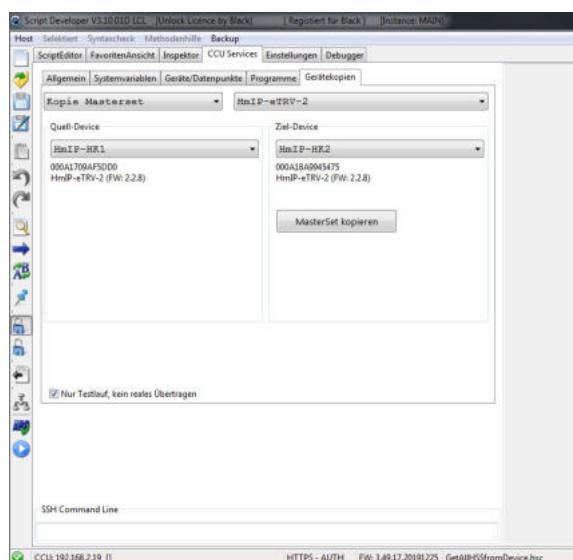
Ähnliche Geräte sind im Source Code Hardcoded, momentan sind folgende Gruppen angelegt:

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

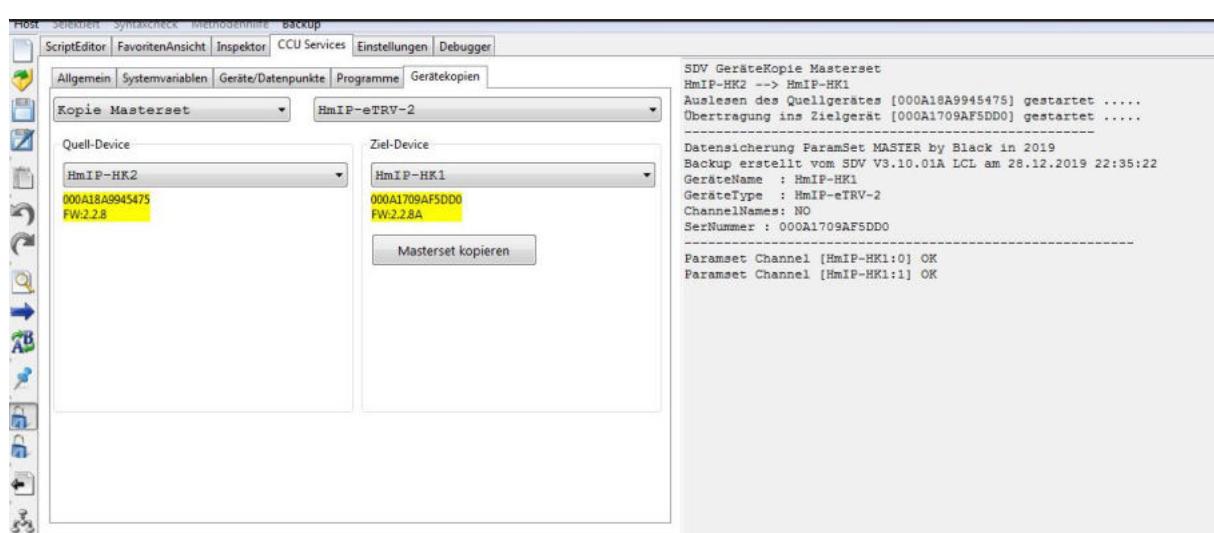
9.1 Kopieren Masterset

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

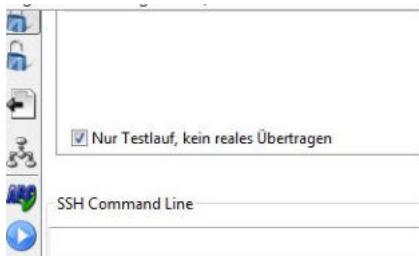
Unter Gerätetype werden die Gerätetypen vorgeschlagen, die mindestens 2 mal vorhanden sind, dann daraufhin das Ziern und das Quelldevice auswählen.



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



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



Mögliche Fehlerbilder:

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

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

Ein Channel lässt sich nicht zuordnen:

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

9.2 Gerätetausch

noch nicht implementiert

9.3 Kopieren Week Profiles

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

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

Wird trotzdem übertragen gedrückt läuft folgende, erweiterte Logik:

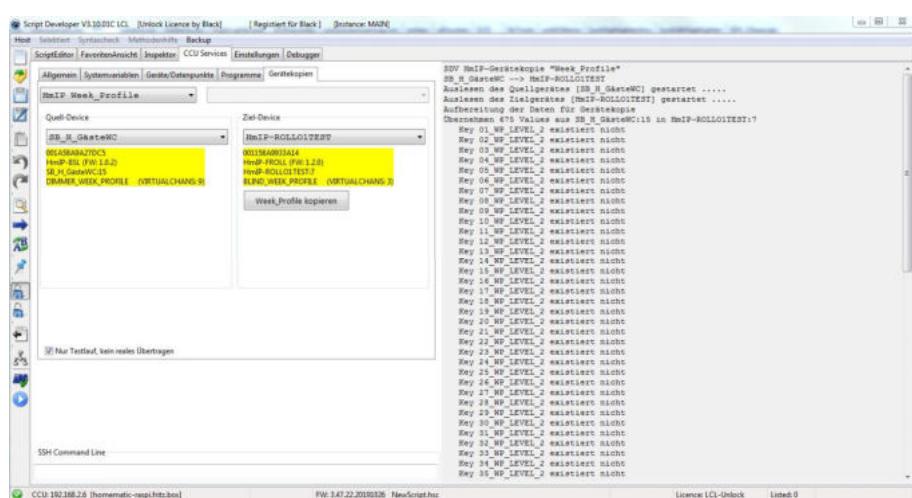
prüfen ob es den Key in dem Quelldevice gibt: wenn nein, Meldung und der alte Wert im Zieldevice bleibt erhalten. Hier in Bild zu sehen, ein BSL hat im Gegensatz zu einem FROLL keinen LEVEL_2 drum die Menge an Meldungen.

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

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

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

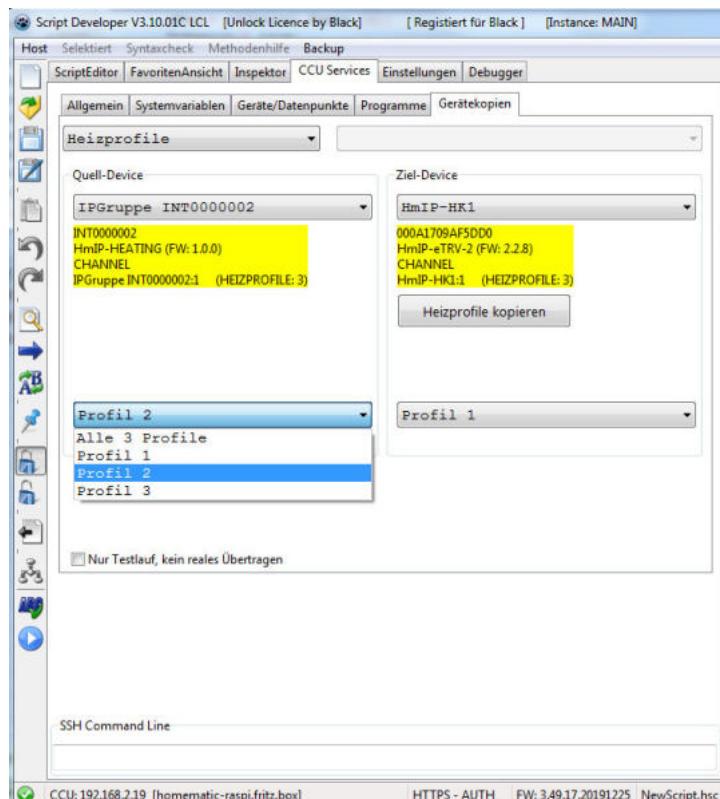
Es können dabei die Wochenprofile (mit den o.g. Einschränkungen) auch unter unterschiedlichen IP Geräten kopiert werden.



9.4 Kopieren Heizprofile

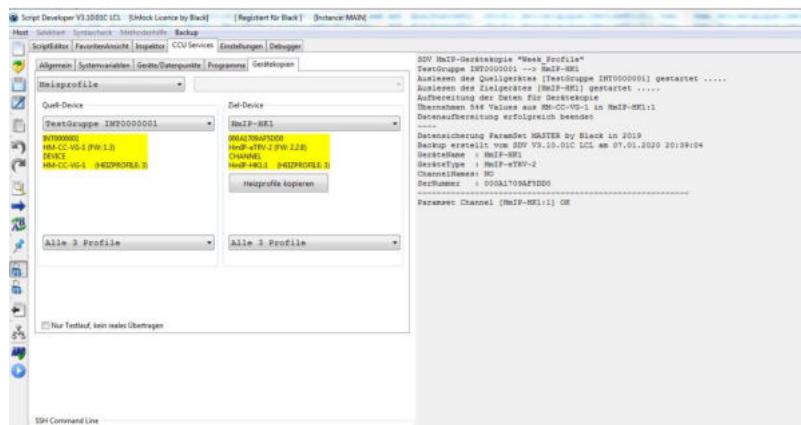
Es können Profile selektiv auch zwischen verschiedenen Gerätetfamilien kopiert werden, also z.b. von IPgruppen in IP Thermostate, oder von IP Thermoste in Classic Thermostate, oder von einer Classic Gruppe in einen IP Thermostaten. Bei mehreren Profilen in einem Gerät können alle Profile oder auch selektiv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

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

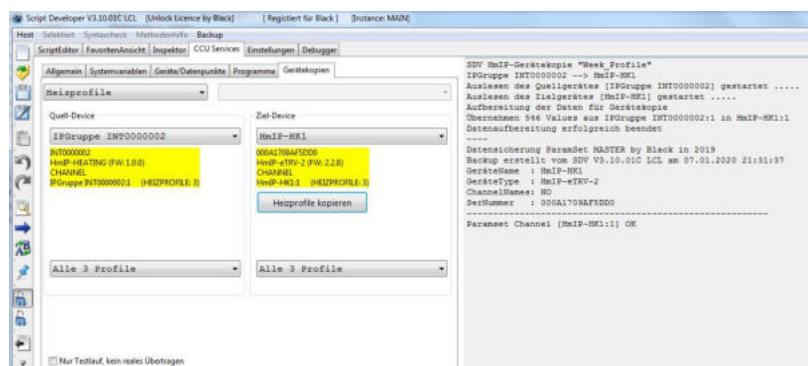


Zwischen allen Geräten classic HK Thermostat, classic WT Thermostat, classic Heizgruppe sowie zwischen IP HK Thermostaten verschiedener Ausführungen, dem IP Wandthermostaten und der IP Heizgruppe können nun wild die Heizprofile gerätegruppenübergreifend hin und her kopiert werden.

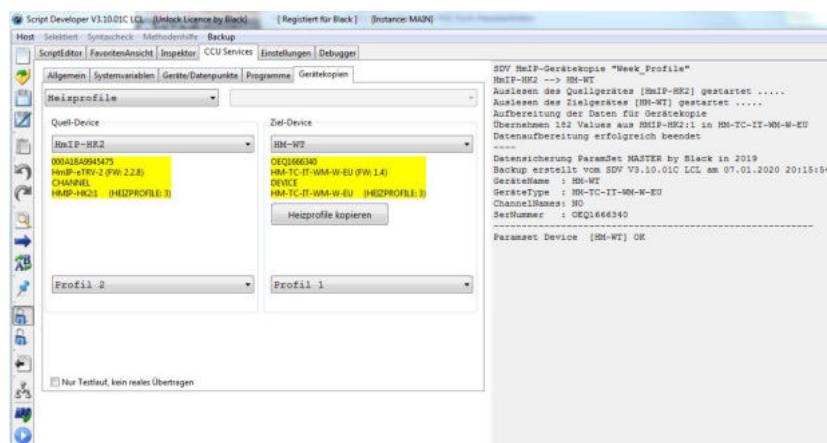
Beispiel: Kopieren aller 3 Profile einer classic Heizgruppe in einen IP HK-Thermostat



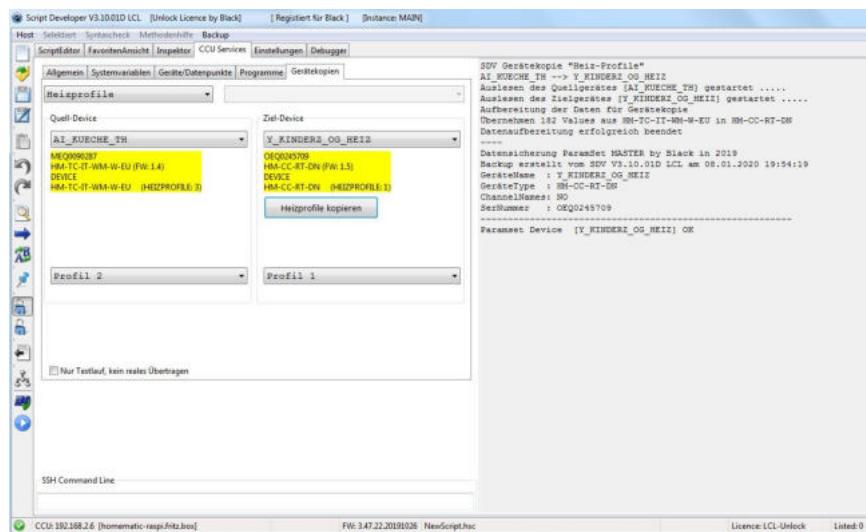
Beispiel: Kopieren aller Profile einer IP Heizgruppe in einen IP Wandthermostaten



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



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



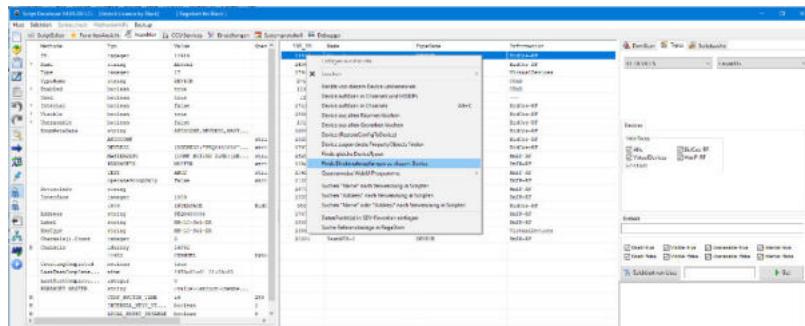
10 Direkte Verknüpfungen DVs

Der SDV unterstützt bei der Untersuchung und Bearbeitung von Direktverknüpfungen. Dies soll nicht die WEB-UI ersetzen, sondern mit Funktionen und Filtern unterstützen, welche die WebUI nicht hat.

Es gibt keinen Direkten Menüpunkt für Direkte Verknüpfungen, Die Arbeitsoberfläche ist der Inspektor.

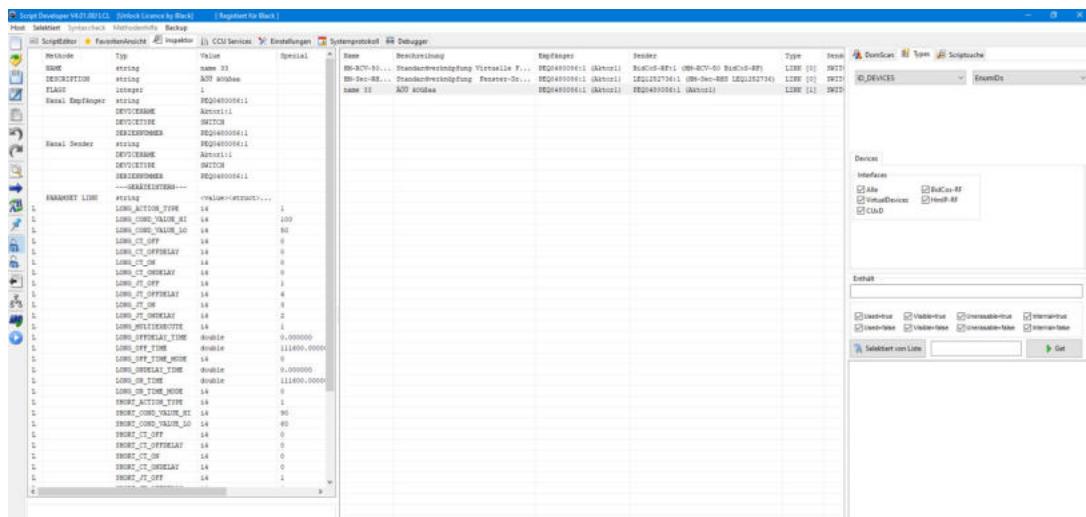
Beispiel:

Darstellen aller Direktverknüpfungen eines Gerätes



Gerät im Inspektor auswählen, rechte Maustaste und in dem Menü dann den Unterpunkt: Finde Direktverknüpfungen zu diesem Device:

Hat das Device keine Direktverknüpfungen, so erscheint ein leeres Listenfeld, dieses hier hat aber DVs, diese werden nun gelistet:

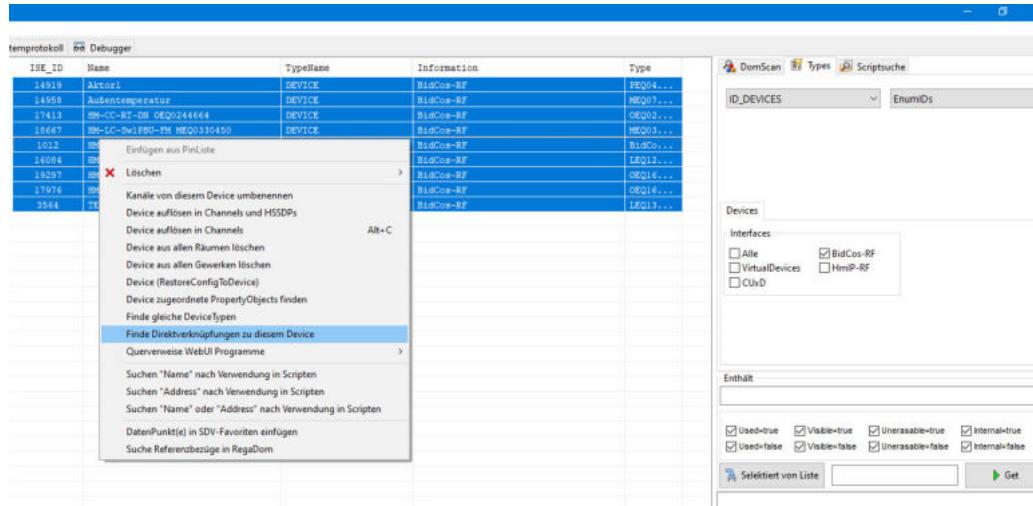


Der SDV listet hierbei wirklich alles auf, auch Geräteinterne Direktverknüpfungen, welche in der WEbUI nicht unter DVs erscheinen.

Diese können nach allen Spaltenkriterien sortiert werden (Auf Empfänger, Sender etc klicken). Klick auf die DV in der Listenansicht öffnet in der Detailansicht alle Daten, die mit einem * gekennzeichneten Parameter können verändert werden. Aus Sicherheitsgründen ist die Namen und Beschreibung Bearbeitung bei DVs mit einem Flag >0 gesperrt. Wie schon von den Mastersets bekannt und in der Editorbeschreibung nachzulesen ist es hier auch möglich, z.B. einen oder mehrere L Parameter zu markieren und dann im Editor automatisch den korrekten Code für PutParamset bzw getparamset erzeugen zu lassen. Der SDV parametriert, da er da Sender und Empfänger kennt, direkt den kompletten Aufruf richtig.

Natürlich kann auch direkt von mehreren oder von allen Devices die Direktverknüpfungen aufgelöst werden. Dazu in der Listenansicht mehrere oder alle Geräte anwählen, rechte Maustaste und auflösen.

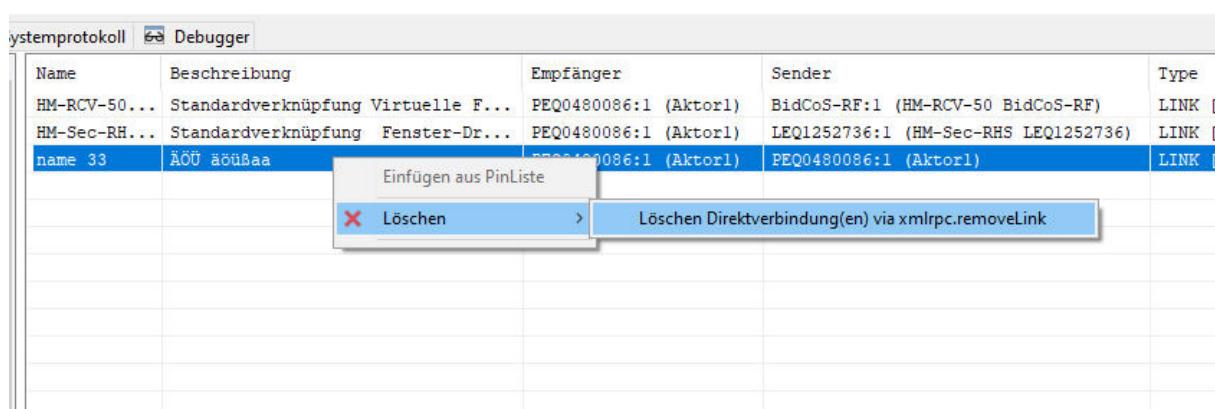
Sinnig bei der Vorauswahl ist noch: alle Geräte mit einem Interface, z.B. alle Bidcos Geräte. Dazu in Device vorselektion BidCosRF anwählen, Deviceliste generieren, Ctrl-A in der Listenansicht, rechte Maustaste, DVs auflösen und schon hat man alle DVs von BidCos Geräten.



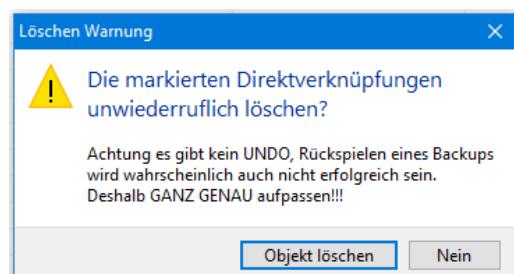
Ergibt dann aufgelöst:

10.1 Löschen von Direkten Verknüpfungen

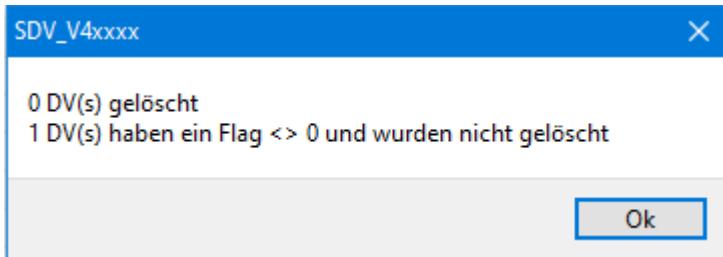
Eine oder auch mehrere Direkte Verknüpfungen lassen sich markieren und löschen. Dazu muss das Schloss offen sein und gemäß Kapitel 4.6 das Löschen von DVs erlaubt sein.



Vorsichtshalber muss das nochmals bestätigt werden



Hier in diesem Fall griff dann noch der besondere Schutzmechanismus, diese DV hat ein Flag <>0 und wird deshalb, obwohl angewählt und bestätigt, nicht gelöscht.



10.2 Sichern von Direktverknüpfungen

Wie im Kapitel 7.2.7 (Sichern komplettes Device) bzw 7.2.9 beschrieben können die Direkten Verknüpfungen zu einem oder auch mehreren selektierten Geräten im JSON Format gesichert werden. Dazu wie oben beschrieben Geräte wählen und dann Hauptmenü Backup.

Edit: Ab der letzten 4.x Version können auch einzelne Direktverknüpfungen exportiert werden

Dazu wie üblich DVs auflösen, DV anwählen und mit diesem Menüpunkt wird die einzelne DV exportiert

The screenshot shows the 'Backup' menu open, with the 'Paramset Link (selektiert)' option highlighted. The main window displays a table of device parameters, and the right side shows a 'Debugger' pane with some log entries.

11. SDV Programmeditor

Der Programmeditor des SDV wurde von mir entwickelt, um die krampfhaften Begrenzungen und Einschränkungen der WebUI zu umgehen. Hinzu kommt der verschwenderische Umgang der WebUI mit IsIDs bei der Programmierung. Ziel sollte ein Editor sein, der nicht stylisch bunt sein musste, sondern der Möglichkeiten bietet, die in der WebUI fehlen und sich aufgrund der internen Arbeitsweise, wie in der WebUI Programme editiert werden, wohl auch nicht implementieren lassen. Unter diesem Aspekt entstand 2022 im Grundzug und 2023 dann letztlich ausprogrammiert dieser Editor.

In dem Zusammenhang einen ausdrücklichen Dank an @Nimmnenkeks und @RolandT, die in der Alpha Test und Entwicklungsphase mit Ideen und Ausprobieren immer an der bloody edge eines Alpha Systems zu dem jetzigen Editor beigetragen haben.

Ich denke auch, das ist jetzt erst der erste Wurf, es wird anhand der Möglichkeiten nun weitere Ideen und Erweiterungen geben mit der Zeit.

Der Editor hier ist nicht unbedingt gedacht für einen absolut unbedarften Anfänger, - für jemand mit bisschen Verständnis über seine CCU ist er aber ein exzellentes Hilfsmittel

Solange noch raw-beta Phase ist der Editor an Level6 gebunden, wird später auch noch Standart-5 werden. Die Anzahl an Programmzeilen ist mittlerweile auch gut im fünfstelligen Bereich (ohne Bibliothekskomponenten oder Classes aus dem SDV selber, nur für den Editor programmierte Zeilen)

11.1 Speichermanagement

Das Speichermanagement beschreibt, wie die Rega und der SDV-Programmeditor sowohl ihre Daten halten als auch mit dem Verbrauch von IsIDs umgehen.

11.1.1 Allgemeine Definitionen

Um gewisse Eigenschaften zu Umschreiben wird folgendes Wording benutzt:

11.1.1.1 Strukturverändernde Operationen

Strukturveränderung bedeutet, es wurden Änderungen im Programmeditor durchgeführt, die sich nicht in die bestehende Struktur des Ursprungsprogrammes schreiben lassen. (Beispiel: Hinzufügen / Löschen von Objekten, Verschieben von Objekten außerhalb ihres ursprünglichen Geltungsbereiches.

11.1.1.2 Werteverändernde Operationen

Eine Werteverändernde Operation verändert wie der Name schon sagt nur Werte innerhalb eines Objektes, verändert aber die Struktur eines Programmes nicht. Beispiel wäre das Ändern eines Vergleichswertes oder das Ändern einer Verzögerung etc.

11.1.1.3 Virtuelles Programm

Ein virtuelles Programm ist ein Programm, welches als neues Programm im SDV-Programmeditor erzeugt wurde. Solange dieses noch nicht durch Hochladen als reales Programm in der Rega angelegt wurde, existiert dieses nur im Speicherbaum des SDV-Programmeditors.

11.1.1.4 Childs und Parents

Ein Childobjekt ist ein Objekttyp, der unterhalb seines Parent im Baum angelegt ist. In einem Destination Objekt z.B. ist das Child die SingleDestination.

11.1.1.5 Abkürzungen und Objekthierarchien

-GRP: Programmgruppe (Kommt noch, siehe Issue im SDV Thread)

- PRG: Program - Programmobjekt, die Objekte, die unter Programme in der WebUI gelistet werden

-MAIN- Condition -Maincondition – Bedingungsblock, spezielle Eigenschaften unter Main Condition

-RULE – Regelobjekt, die Wenn, SonstWenn, Sonst Blöcke der WebUI, eine Rule kann mehrere Conditions, muss aber genau eine Destination enthalten

-CND – Condition - Bedingungsblock , beinhaltet verschiedene Bedingungen

-SCND – Singlecondition - ist eine Einzelne logische Bedingung

-DST – Destination – Anweisungsblock – Eine Regel enthält immer genau einen Anweisungsblock

-SDST- SingleDestination – einzelne Anweisung, z.b. ein Skript oder eine Zuweisung

PRG

- MAIN (gibt es genau einmal bei einem PRG)
 - + SCND (optional)
 - + SCND (optional)
- RULE (1. Rule = WENN)
 - + CND (optional, sollte aber keine leeren Bedingungen enthalten)
 - . SCND
 - . SCND (optional)
 - + CND (n)
 - . SCND (n)
 - . SCND (n)
 - + DST
 - . SDST (sollte keine leeren Anweisungen enthalten)
 - . SDST (n) optional
- RULE (2. Rule = 1. SONST WENN , oder auch Sonst)
 - ... (CND + DST)
- RULE (x)

Parent	Objekt	Child
	PRG	→ RULE
PRG	← RULE	→ CND
PRG	← RULE	→ DST
RULE	← CND	→ SCND
RULE	← DST	→ SDST
	PRG	→ MAIN
PRG	← MAIN	→ SCND

11.1.2 Speichermanagement der WebUI/ der Rega

Die WebUI wirft bei jeder kleinsten Programmänderung mit dem Verbrauch von IsIDs wild um sich
Dazu ein kleines Beispiel

Gegeben sei folgendes WebUI Programm

Legende:

P: Programmkopfobjekt

T: Der Programmrumph (dies können mehrere hundert verschiedene Objekte sein)

_ : Lücken in der Rega

PT

Nun wird dieses Programm editiert

PT_ET_E

Und das editierte Programm wird übernommen

P__T

Nun stellt man fest, eine Zahl war falsch, editieren:

P__TP_ET_E

Und wieder übernommen

P____T

Durch die interne Arbeitsweise (Programmkopie in der Rega selber) und eine höchst ineffektive Zusammenführung des geänderten Programmes wachsen die Lücken in der Rega unaufhörlich.

Vergleichbar ist es mit der Expansion des Universums. Die Materie (die Objekte) bleiben mengenmäßig gleich, der Raum (die Regalücken) wachsen unaufhörlich. Mir sind Systeme bekannt, deren IDs im 1000000 Zahlenraum angekommen sind und hauptsächlich aus Lücken bestehen.

11.1.3 Speichermanagement des SDV Programmeditors

Im Gegensatz zur Rega/WebUI arbeitet der SDV-Programmeditor komplett RAM orientiert. Von dem/den Programmen werden Tree-structure Abbilder mit allen benötigten Elementen im Arbeitsspeicher des Computers angelegt.

Damit sind alle Änderungen Rega unabhängig, es können alle erdenklichen Operation, die mit dem WebUI Programmeditor nicht gingen, durchgeführt werden. Dazu zählt:

- Reihenfolge von Objekten frei verschieben (auch Regel Objekte)
- Kopieren von Objekten und Einfügen an beliebiger Stelle (Auch aus anderen Programmen)
- Löschen bzw Einfügen von Objekten an beliebigen (aber erlaubten) Stellen
- Direktes Bearbeiten von Programmstrukturelementen
- Möglichkeiten erweiterter Eigenschaften die die WebUI nicht kennt (Main-Conditions, Logikerweiterung)
-

Dank RolandT (der mich mit seinem Undo Redo Wunsch bis kurz vor den Wahnsinn getrieben hat) verfügt der SDV Programmeditor über eine (momentan 100 stufige) Undo/Redo Funktionalität, jede Form von Änderungen können zurückgenommen werden.

Das geänderte Programm muss natürlich dann am Ende entweder verworfen (Die Änderungen finden sich nicht in der Rega wieder oder gespeichert werden).

11.1.4 Speicherstrategie des SDV-Programmeditors

Ein Programm kann vom SDV-Programmeditor mit mehreren möglichen Strategien gespeichert werden. Der SDV-Programmeditor räumt dabei jeweils seinen Bereich der Rega auf, um möglichst wenige, im Idealfall keine Lücken zu hinterlassen.

11.1.4.1 Speichern (Komplett Anlage)

Dieser Punkt kann bei jeder Änderung durchgeführt werden, Strukturverändernde Operationen lassen sich allerdings nur so in der Rega speichern:

Strategie: Das Komplette Programm wird aus der Rega gelöscht, anschließend wird das Programm komplett neu geschrieben.

xxPT

xxPT (Änderung im Ram)

xx (Löschen des Programms)

xxPT (nach Neuschreiben)

Wenn auch mehrere strukturelle Änderungen am Programm hintereinander gemacht werden, entstehen keine Lücken, da der SDV-Programmeditor vor dem Speichern alles wegräumt und dann neu sauber schreibt.

Für den Hinterkopf: Hier wird alles neu geschrieben, auch die ID des Programmobjektes kann sich hierbei ändern. Dies mögen manche Middlewares nicht, die dann ein Neueinlesen der Rega erwarten. Dafür gibt es den nächsten Punkt ^^

11.1.4.2 Speichern Komplettanlage (ID von Programm bleibt)

Dies entspricht dem Punkt 11.1.4.1, nur das das Programmobjekt beim Neuschreiben seine ursprüngliche ID behält, der Rest wird aber weggeräumt und wie unter 11.1.4.1 neu geschrieben

xxPT

xxPT (Änderung im Ram)

xx (Löschen des Programms)

xxP (Neuanlage des Programmobjektes unter alter alten ID)

XXPT (nach Neuschreiben)

Diese Form ist Middleware freundlich

11.1.4.3 Speichern Override

Wurden nur Werteverändernde Operationen gemacht, die Struktur ist aber identisch, so kann via Override das im SDV Programmeditor geänderte Programm direkt in die Struktur des Ursprungspfrogrammes geschrieben werden. Dies braucht überhaupt keine neuen IDs.

Keine Sorge, man muss dieses nicht alles im Kopf behalten, der SDV-Programmeditor prüft anhand eines Hashes, ob Strukturelle oder nur Werteändernde Operationen vorlagen und graut entsprechende Menüpunkte aus. Ebenso gibt's eine voreinstellbare Speicherstrategie, welche dies automatisiert nach den Prioritäten abhandelt, so dass nur der der Speicher Button von Relevanz ist. Trotzdem ist es hilfreich, die interne Arbeitsweise der Rega als auch des SDV-Programmeditors zu kennen und bewerten zu können

xxPT

xxPT (Änderung im Ram)

xxPT (nach überschreiben)

11.1.4.5 Virtuelles Programm real in der CCU speichern (Neuanlage)

Legt ein virtuelles Programm in der Rega der CCU an, dadurch wird ein virtuelles Programm zu einem normalen, realen Programm.

11.1.4.6 Speichern unter neuem Namen (Neuanlage)

Erstellt eine Kopie des bearbeiteten Programmes, in der Rega, wobei dem Namen bei Speichern der Suffix „_SDVKopie“ hinzugefügt wird. Entspricht funktional 11.1.4.1

11.1.4.7 Reload von CCU Programm (Programm nach Programmnamen)

Sinnig für den Fall, dass eine Änderung im SDV-Programmeditor sich während der Arbeit als falscher Weg herausstellt, der Punkt verwirft alle Änderungen und liest das Rega Programm neu in den Speicher.

11.1.4.4 Speichern Override (Extend)

Noch nicht freigegeben, soll eine Mischung aus 11.1.4.3 und 11.1.4.2 werden

11.2 Der Programmeditor

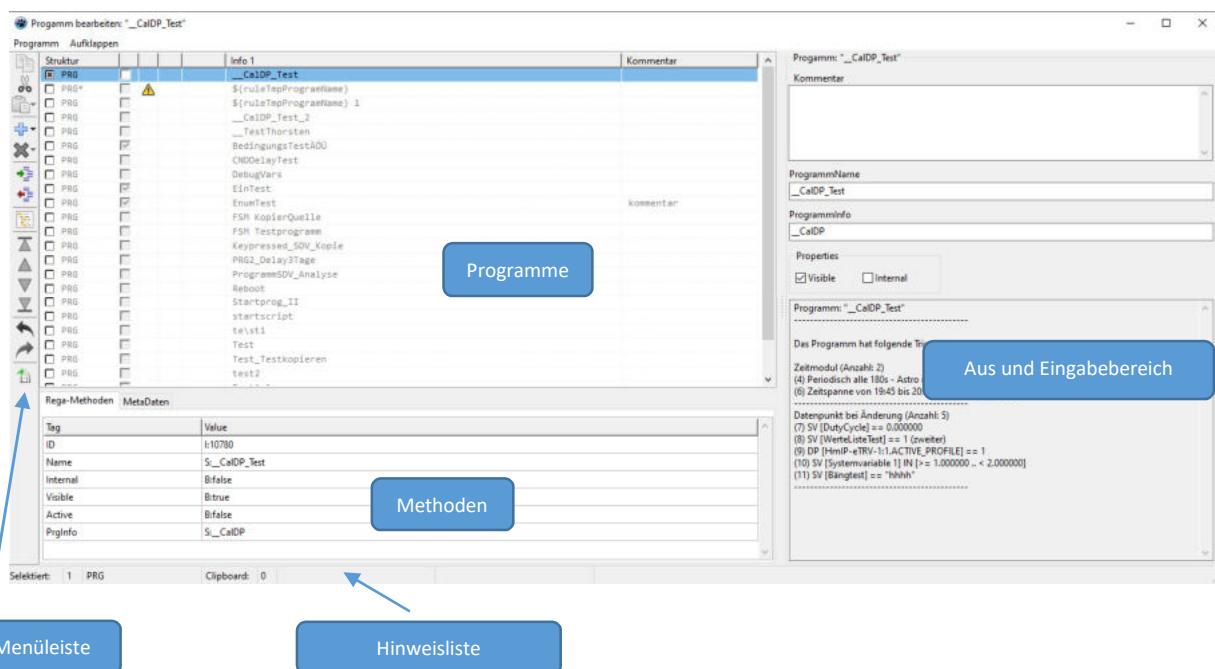
Der Programmeditor wird gestartet, indem der Menüpunkt des Hauptmenüs (Programme bearbeiten) angeklickt wird. Ist im Inspektor in der Listenansicht kein Programm selektiert, so startet der SDV-Programmeditor ohne aktives Arbeitsprogramm. Ist im Inspektor ein Programm selektiert so startet der Editor mit diesem selektierten Programm als aktives Arbeitsprogramm.

Erstmal ein kleines Vorwort zu dem Programmeditor. Im Programmeditor kann immer ein Programm gleichzeitig bearbeitet werden. Dieses ist das Arbeitsprogramm. Das Arbeitsprogramm kann auch gewechselt werden. Alle anderen Programme können dargestellt und auch in Einzelheiten betrachtet werden, aber aktive Änderungen gehen nur im Arbeitsprogramm.

Im SDV-Programmeditor wird immer das Arbeitsprogramm und wahlweise alle oder die ebenfalls zur Anzeige selektierten Programme dargestellt. Dies ermöglicht eine Anpassung der Anzeige und Erhöhung der Übersichtlichkeit. Soll nur in einem Programm schnell ein Wert geändert werden so ist es sinnvoll, nur das Arbeitsprogramm darzustellen. Soll aus einem bestimmten Programm Teile in das Arbeitsprogramm kopiert werden, so wählt man nur dieses Programm zusätzlich an und blendet alle anderen aus.

Das Arbeitsprogramm

- steht im SDV-Programmeditor immer ganz oben.
- wird in schwarzer Schrift dargestellt



Welches Objekt selektiert ist (blaue Zeile) wird hier angezeigt: **Selektiert: 1 PRG**

Welches Objekt im Clipboard liegt, wird hier angezeigt: **Clipboard: 1 SCND**

Das Arbeitsprogramm steht immer an 1. Stelle:

Das gelb markierte, bei einem Arbeitsprogramm ausgegraute Kreuz gibt an, das Programm ist immer eingebendet

Programm bearbeiten: '_CalDP_Test'

Aufklappen

Struktur			Info 1
PRG			_CalDP_Test
PRG*			\$ruleTmpProgramName
PRG			\$ruleTmpProgramName
PRG			_CalDP_Test_2
PRG			_TestThorsten
PRG			BedingungsTestAÖU
PRG			CNDDelayTest
PRG			DebugVars

Soll nur in dem Arbeitsprogramm etwas gemacht werden, lassen sich alle anderen Programme schnell ausblenden, bei Bedarf mit einem Button natürlich auch einblenden.

Programm bearbeiten: '_CalDP_Test'

Aufklappen

Struktur			Info 1
PRG			_CalDP_Test
MAIN			PROGRAMMHAUPTBEDINGUNG
RULE			MENN
CND	O		BEDINGUNGSBLOCK
SCND	U	⌚ ZP	ZM [Periodisch alle 180s - Astro nachts]
SCND	U	?	SV [AlwaysTrue] == true (ist wahr)
SCND	U	⌚ ZP	ZM [Zeitspanne von 19:45 bis 20:15 - täglich, jeden Tag]
SCND	U	✓ CHG	SV [DutyCycle] == 0.000000
SCND	U	✓ CHG	SV [WertelisteTest] == 1 (zweiter)
SCND	U	✓ CHG	DP [HmIP-eTRV-1:1.ACTIVE_PROFILE] == 1
SCND	U	✓ CHG	SV [Systemvariable 1] IN [>= 1.000000 .. < 2.000000]
SCND	U	✓ CHG	SV [Bängtest] == "hhhh"
DST			RETR ANWEISUNGSBLOCK

Soll noch ein oder auch mehrere weitere Programme sichtbar sein, der Rest aus Übersichtlichkeitsgründen aber ausgeblendet werden, so lassen sich in den nicht Arbeitsprogrammen die Kreuz Markierungen setzen und dann lassen sich alle Programme ohne Markierungen ausblenden oder aber wieder einblenden

Programm bearbeiten: '_CalDP_Test'

Aufklappen

Struktur			Info 1
PRG			_CalDP_Test
MAIN			PROGRAMMHAUPTBEDINGUNG
RULE			MENN
CND	O		BEDINGUNGSBLOCK
SCND	U	⌚ ZP	ZM [Periodisch alle 180s - Astro nachts]
SCND	U	?	SV [AlwaysTrue] == true (ist wahr)
SCND	U	⌚ ZP	ZM [Zeitspanne von 19:45 bis 20:15 - täglich, jeden Tag]
SCND	U	✓ CHG	SV [DutyCycle] == 0.000000
SCND	U	✓ CHG	SV [WertelisteTest] == 1 (zweiter)
SCND	U	✓ CHG	DP [HmIP-eTRV-1:1.ACTIVE_PROFILE] == 1
SCND	U	✓ CHG	SV [Systemvariable 1] IN [>= 1.000000 .. < 2.000000]
SCND	U	✓ CHG	SV [Bängtest] == "hhhh"
DST			RETR ANWEISUNGSBLOCK
PRG*			\$ruleTmpProgramName
PRG			\$ruleTmpProgramName
PRG			_CalDP_Test_2
PRG			_TestThorsten
PRG			BedingungsTestAÖU
PRG			CNDDelayTest
PRG			DebugVars
PRG			EinstTest
PRG			Zm...Zsch

Teile dieses/dieser Programme können dann z.b. als Kopierquelle genutzt werden

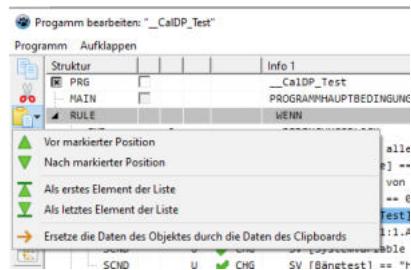
Programm bearbeiten: '_CalDP_Test'

Aufklappen

Struktur			Info 1
PRG			_CalDP_Test
MAIN			PROGRAMMHAUPTBEDINGUNG
RULE			MENN
CND	O		BEDINGUNGSBLOCK
SCND	U	⌚ ZP	ZM [Periodisch alle 180s - Astro nachts]
SCND	U	?	SV [AlwaysTrue] == true (ist wahr)
SCND	U	⌚ ZP	ZM [Zeitspanne von 19:45 bis 20:15 - täglich, jeden Tag]
SCND	U	✓ CHG	SV [DutyCycle] == 0.000000
SCND	U	✓ CHG	SV [WertelisteTest] == 1 (zweiter)
SCND	U	✓ CHG	DP [HmIP-eTRV-1:1.ACTIVE_PROFILE] == 1
SCND	U	✓ CHG	SV [Systemvariable 1] IN [>= 1.000000 .. < 2.000000]
SCND	U	✓ CHG	SV [Bängtest] == "hhhh"
DST			RETR ANWEISUNGSBLOCK
PRG*			BedingungsTestAÖU

Doppelklick auf eine Zeile öffnet vorhandene Unterobjekte (Vorhandensein von Unterobjekten zeigt das schwarze Dreieck)

11.2.1 Positionsangaben beim Einfügen / Generieren



11.2.1.1 Vor markierter Position

Das in der Zwischenablage liegende Objekte / das neu erzeugte Objekt wird vor der markierten Position im Editor eingefügt

11.2.1.2 Nach markierter Position

Das in der Zwischenablage liegende Objekte / das neu erzeugte Objekt wird nach der markierten Position im Editor eingefügt

11.2.1.3 Am Anfang der Liste

Das in der Zwischenablage liegende Objekte / das neu erzeugte Objekt wird als erstes Element in der Liste dieser Objekte dargestellt

11.2.1.4 Am Ende der Liste

Das in der Zwischenablage liegende Objekte / das neu erzeugte Objekt wird als letztes Element in der Liste dieser Objekte dargestellt

11.2.1.5 Ersetze Daten des Objektes durch Daten des Clipboards

Dies kopiert nicht das Objekt selber , sondern nur den Inhalt des Objektes in ein Objekt des gleichen Types

11.2.1.6 Als erstes Element der Ziel-ChildListe

Beispielhaft: im Clipboard liegt ein SCND Objekt, Angeklickt ist ein CND Objekt, so kann das SCND Objekt des Clipboards am Anfang der SCND Liste des CND Objekte eingefügt werden

11.2.1.7 Als letztes Element der Ziel-ChildListe

Beispielhaft: im Clipboard liegt ein SCND Objekt, Angeklickt ist ein CND Objekt, so kann das SCND Objekt des Clipboards am Ende der SCND Liste des CND Objekte eingefügt werden

11.2.2 Menüleiste

Die Buttons der Menüleiste haben alle die Hintfunktionalitäten hinterlegt. Tasten, die im Augenblick nicht freigegeben sind, werden ausgegraut dargestellt

11.2.2.1 Kopieren



Das selektierte Objekt (samt seinen untergeordneten Child-Objekten) wird auf das Editoreigene Clipboard kopiert. (Tastenkürzel CTRL+C)

11.2.2.2 Ausschneiden



Nur im Arbeitsprogramm möglich. Das selektierte Objekt (samt seinen untergeordneten Child-Objekten) wird an der aktuellen Stelle ausgeschnitten und auf das Editoreigene Clipboard kopiert. (Tastenkürzel CTRL+X)

11.2.2.3 Einfügen



Nur im Arbeitsprogramm möglich. Das Objekt auf dem Clipboard wird an der markierten Stelle im Programm eingefügt. Der SDV Editor prüft dabei intern, ob ein Einfügen des Objektes des Clipboards an der Stelle möglich ist. Welches Objekt im Clipboard liegt, lässt sich in der Statuszeile sehen.

Clipboard: | 1 | SCND

Tastenkürzel CTRL+V

11.2.2.4 Erzeugen



Erzeugt den angegebenen Objekttyp an der im aufklappenden Untermenü ausgewählten Stelle. Es ist hier auch möglich, mitten in einem Programm ein zusätzliches Regelobjekt zu erzeugen. Wenn ein Programmobjekt angeklickt ist, so enthält das Auswahlmenü zusätzlich auch die Möglichkeit ein zusätzliches, neues Programm zu erzeugen. Ein neues Programm hat erst einmal immer den Namen: „Neues Programm (UTC des Erzeugens)“ Tastenkürzel Ctrl + Plustaste

11.2.2.5 Löschen



Löscht das angewählte Objekt. Ein Programm kann auch gelöscht werden, dazu muss es aber das Arbeitsprogramm sein. Es erfolgen auch entsprechende Sicherheitsabfragen.

Bis auf das Löschen von Programmen kann jede Aktion durch Undo/Redo wieder rückgängig gemacht werden

[11.2.2.6 Alle Programme darstellen](#)



Alle Programme, auch die nicht angekreuzt sind, werden dargestellt

[11.2.2.7 Nur angekreuzte Programme darstellen](#)



Stellt nur noch die angekreuzten Programme dar, alle anderen werden ausgeblendet

[11.2.2.8 Alle untergelagerten Objekte \(Childs\) aufklappen](#)



Alle von dem angewählten Objekt vorhandenen untergeordneten Objekte (Childs) werden aufgeklappt. Wird benutzt zum kompletten Aufklappen eines Programmes oder eines anderen, noch eingeklappten Unterpunktes.

[11.2.2.9 Verschieben an Anfang](#)



Das angewählte Objekt wird an die erste mögliche Position in seinem Parent verschoben

[11.2.2.10 Verschieben um eine Position nach oben](#)



Das angewählte Objekt wird um eine Position nach oben in seinem Parent verschoben

[11.2.2.11 Verschieben um eine Position nach unten](#)



Das angewählte Objekt wird um eine Position nach unten in seinem Parent verschoben

[11.2.2.12 Verschieben ans Ende](#)



Das angewählte Objekt wird an die letzte mögliche Position in seinem Parent verschoben

[11.2.2.13 UNDO](#)



Einen Bearbeitungsschritt rückgängig. Jede Aktion, sei es Einfügen, Verschieben, Werte ändern kann mit Undo Rückgängig gemacht werden. Es ist nicht so wie in der WebUI, eine Anweisung löschen und weg ist sie. Nur das Löschen von ganzen Programmen kann so nicht mehr rückgängig gemacht werden. Tastenkürzel Ctrl-Z. Der Hint zeigt die aktuelle Tiefe des Undo Stacks:

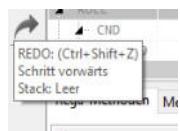


11.2.2.14 REDO



Ein oder mehrere REDO Schritt wieder rückgängig machen. Jede Aktion, sei es Einfügen, Verschieben, Werte ändern kann mit Undo Rückgängig gemacht werden. Es ist nicht so wie in der WebUI, eine Anweisung löschen und weg ist sie. Nur das Löschen von ganzen Programmen kann so nicht mehr rückgängig gemacht werden.

Tastenkürzel Ctrl-Shift-Z. Der Hint zeigt die aktuelle Tiefe des Redo Stacks:



11.2.2.15 Upload in CCU



Das Arbeitsprogramm wird gemäß eingestellter Strategie in die CCU geladen. Ein Virtuelles Programm wird in der CCU neu erzeugt. Ein Upload löscht den Undo/Redo Stack.

11.2.3 Spalten des Programmeditors

The diagram illustrates the columns of the Program Editor. Column 1 shows the program name '_CalDP_Test'. Column 2 shows the 'Programm Aufklappen' button. Column 3 shows the 'Struktur' tree view with nodes like PRG, MAIN, and RULE. Column 4 shows the 'Info 1' panel with details like 'PROGRAMMHAUPTBEDINGUNG' and 'WENN'. Column 5 shows the 'Kommentar' (Comment) column. Column 6 shows the 'Befehlszeile' (Command Line) column, which displays the command 'ZM [Periodisch alle 180s - Astro nachts]'. Column 7 shows the 'Darstellung der Befehlszeile' (Display of Command Line) column, which also displays the same command. The bottom section shows the 'Rega-Methoden' and 'MetaDaten' tabs, and the status bar at the bottom indicates 'Selektiert: 1 PRG' and 'Clipboard: 1 SCND'.

1. Information über die Objektart, das Markieren durch Ankreuzen und die unterlagerte Struktur
2. Aktiv Spalte: Ein Programm oder auch eine Main Condition ist aktiviert
3. Verknüpfung Bedingungsblock in SPS Schreibweise
4. Verknüpfung Einzelne Bedingung in SPS Schreibweise
5. Aktionsspalte:
 - Einzelbedingung: Art der Triggerung bzw nur Prüfen
 - Anweisungsblock: Retriggern oder nicht (Checkbox)
 - Einzelanweisung: Verzögerung der Ausführung in Sekunden
6. Infospalte:
 - Beschreibung des Objektes
 - PRG= Programmnamen
 - RULE= Art und Anzahl der Regel
 - CND= Bedingungssblock
 - SCND= Art der Anweisung (Kurzschriftform)

ZM: Zeitmodul
 DP: Datenpunkt eines Gerätes
 SV: Systemvariable
 AV: Alarmvariable

 - DST: Anweisungsblock
 - SDST: Art der Anweisung in Kurzschriftweise
 - Skript: Homematic Skript
 - DP: Datenpunkt eines Gerätes
 - SV: Systemvariable
 - AV: Alarmvariable
7. Kommentar Jedes Objekt kann mit einem Kommentar versehen werden. Dieser ist allerdings von mir begrenzt auch maximal 500 Zeichen. (Krieg und Frieden oder die Bibel sollte also nicht als Kommentar hinterlegt werden). Damit ist endlich eine Beschreibung der Aktion, des Vergleiches möglich. Die WebUI beherrscht das nicht, der Inspektor des SDV schon.

Das Methodenfeld ist in der jetzigen Version noch sichtbar. (Debugginghilfe). Hier ist hinterlegt, wie der SDV-Programmeditor das Programm der WebUI eingelesen hat bzw. wie er dieses wieder zusammenbauen würde. Im Gegensatz zum Inspektor, wo immer nur ein Schritt geändert werden kann und dies auch direkt in der Rega gemacht wird, arbeitet der SDV-Programmeditor ja komplett im Speicher des PCs.

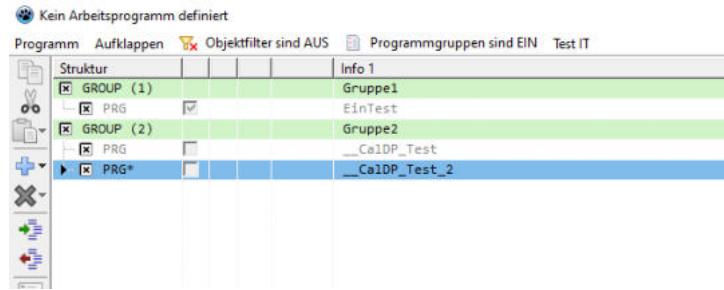
Im Aus/Eingabebereich des SDV-Programmeditors können die Informationen zu einem Objekt in menschenlesbar freundlicher Form dargestellt werden bzw. wenn dieses ein Arbeitsprogramm ist, dort auch geändert werden.

11.2.4 Programmgruppen

Programmgruppen sind eine Option im SDV-Programmeditor, um Programme funktionalen Gruppen zuzuordnen. Dies hat bei vielen Programmen nur einen „ordnerischen“ Aspekt, keinen funktionalen. Die WebUI betrifft diese Option nicht, die WebZU kann wie mit so vielem mit den Programmgruppen nichts anfangen und stellt Programme mit aktivierter Gruppenoption trotzdem so wie immer dar.

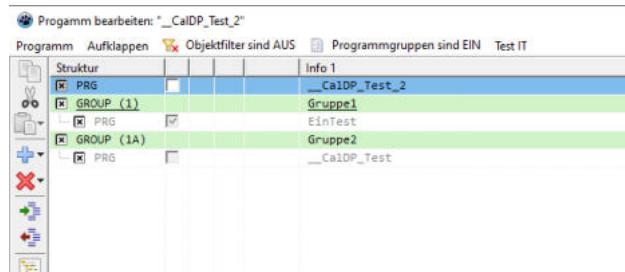
11.2.4.1 Gruppenbearbeitung ein/ausschalten

Im SDV Programmeditor kann die WebUI übliche Darstellung eingeschaltet werden oder auch die Programmgruppendarstellung



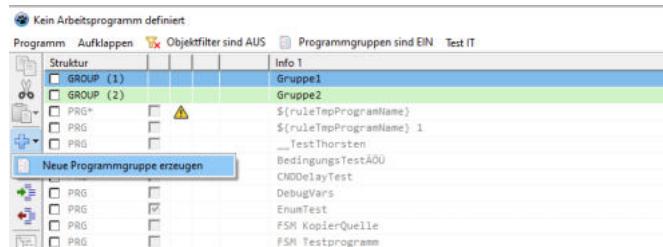
Gruppen haben die einstellbare Hintergrundfarbe und die Kennung GROUP. Der Name der Gruppe wird unter Info1 dargestellt

In der Klammer (1) in dem Beispiel wird die Anzahl der Programme angezeigt, die dieser Gruppe zugeordnet sind. Ein (A) in der Klammer bedeutet, dass das Arbeitsprogramm, welches aber immer noch ganz oben dargestellt wird, sich in dieser Gruppe befindet.

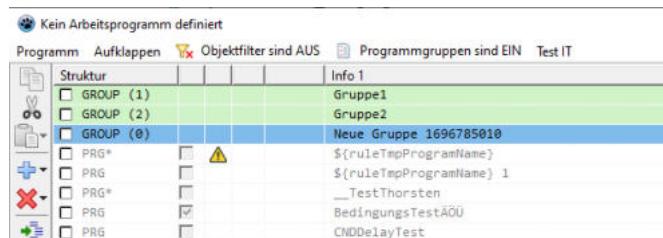


11.2.4.2 Neue Gruppe anlegen

Um eine neue Gruppe anzulegen müssen die Programmgruppen eingeschaltet sein und der Cursor sich auf einer Gruppe oder einem Programm befinden.

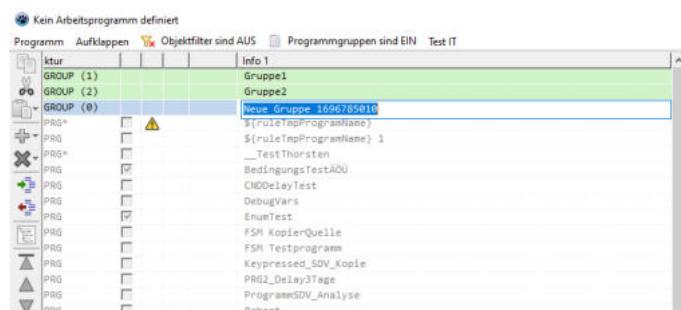


Legt eine neue Gruppe mit zufälligem (in dem Fall ist die Nummer die UTC zeit) Namen an



11.2.4.3 Gruppennamen ändern

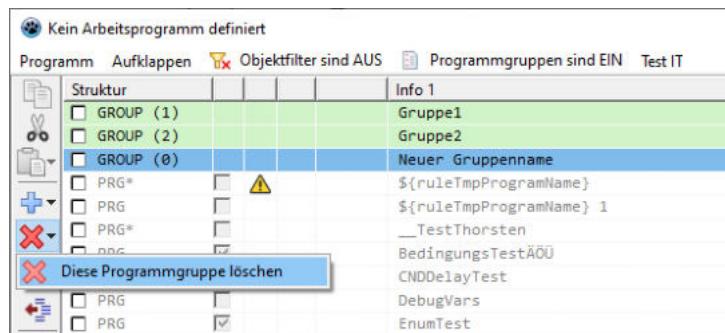
Der Gruppennamen lässt sich durch Klicken auf den Namen in der Info1 Spalte ändern. Der Name muss allerdings eindeutig sein.



Enthält eine Gruppe beim Ändern des Gruppennamens schon Programme, so werden die Programme auch dem neuen Gruppennamen zugewiesen. Dies führt auch ein Rega Write aus (ohne ID Strukturen zu ändern !!) Diese Funktion fällt allerdings nicht unter das UNDO/REDO ! Also Rückgängigmachen nur durch eintippen alter Gruppenname. Ausnahme: Das Arbeitsprogramm. Wenn das Arbeitsprogramm durch die Änderung des Gruppennamens betroffen ist, so wird die Änderung erst wirksam nach Schreiben des Arbeitsprogrammes in die Rega. Wir behalten dabei im Hinterkopf, das Arbeitsprogramm liegt immer komplett im Speicher des Programmeditors und wird erst durch Schreibenpersistiert.

11.2.4.4 Löschen einer Programmgruppe

Eine Programmgruppe lässt sich einfach durch anklicken und dann durch das rote X löschen.

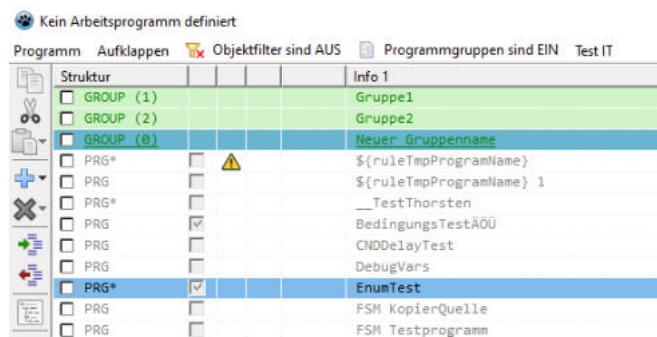


Wichtig: Ein Löschen einer Programmgruppe löscht NICHT eventuell in der Gruppe zugewiesene Programme. Programme, die noch in einer zu löschen Gruppe liegen, werden als „Ohne Gruppenzuordnung“ im Programmbaum eingesortiert. (Quasi so wie ohne Programmgruppendarstellung)

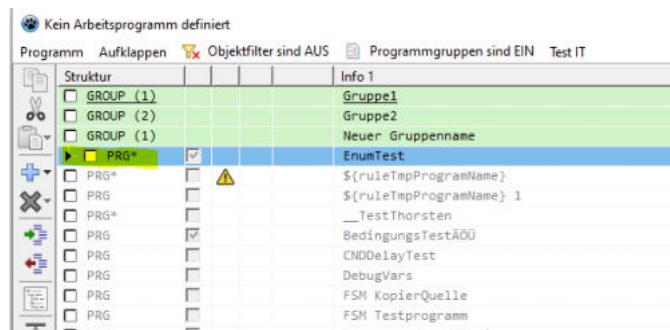
Dies führt auch ein Rega Write aus (ohne ID Strukturen zu ändern !!) Diese Funktion fällt allerdings nicht unter das UNDO/REDO ! Ausnahme: Das Arbeitsprogramm. Wenn das Arbeitsprogramm durch die Änderung des Gruppennamens betroffen ist, so wird die Änderung erst wirksam nach Schreiben des Arbeitsprogrammes in die Rega.

11.2.4.5 Zuweisen eines Programmes in eine Gruppe

Dies geschieht durch Drag/Drop. Programm packen und in die gewünschte Gruppe verschieben.



Und in die Gruppe Neuer GruppenName abwerfen und das Programm wandert in die Gruppe:

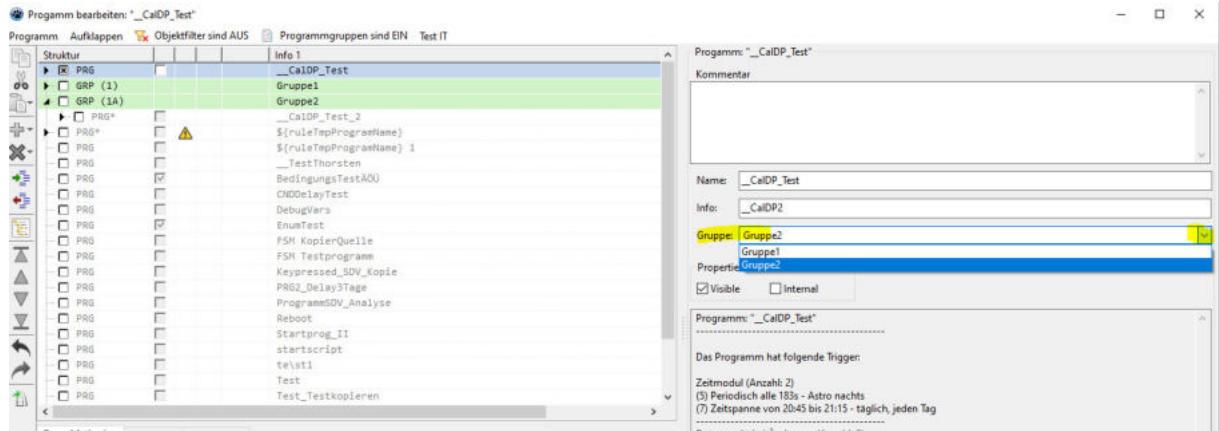


Auch hier gilt: Schreiben in Rega wird durchgeführt. Und: Das Arbeitsprogramm kann ich so nicht ändern

Zuweisen einer Gruppe eines Arbeitsprogrammes.

Ein Arbeitsprogramm kann nicht via Drag-Drop in eine Gruppe verschoben werden.

Die Änderung einer Gruppe geschieht über das Combofeld Gruppe:



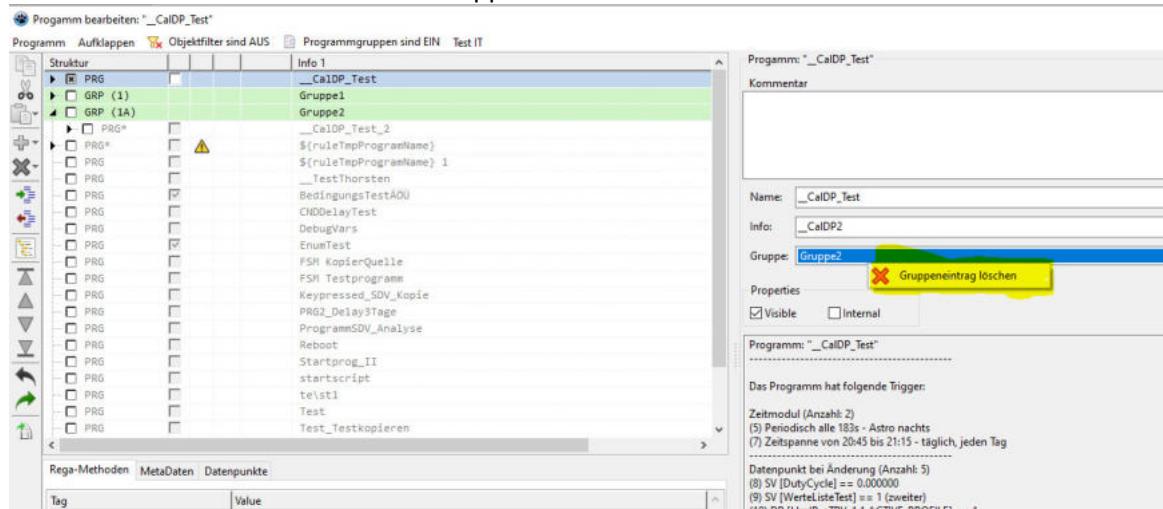
In einem Arbeitsprogramm wirkt dies auch über Undo/Redo, wichtig ist, die finale Zuweisung erfolgt erst durch persistieren des SDV-Programmeditors durch Hochladen in die CCU.

Ebenso wird das Programm aus einer Gruppe gelöscht.

11.2.4.6 Löschen eines Programmes aus einer Gruppe

Als Arbeitsprogramm:

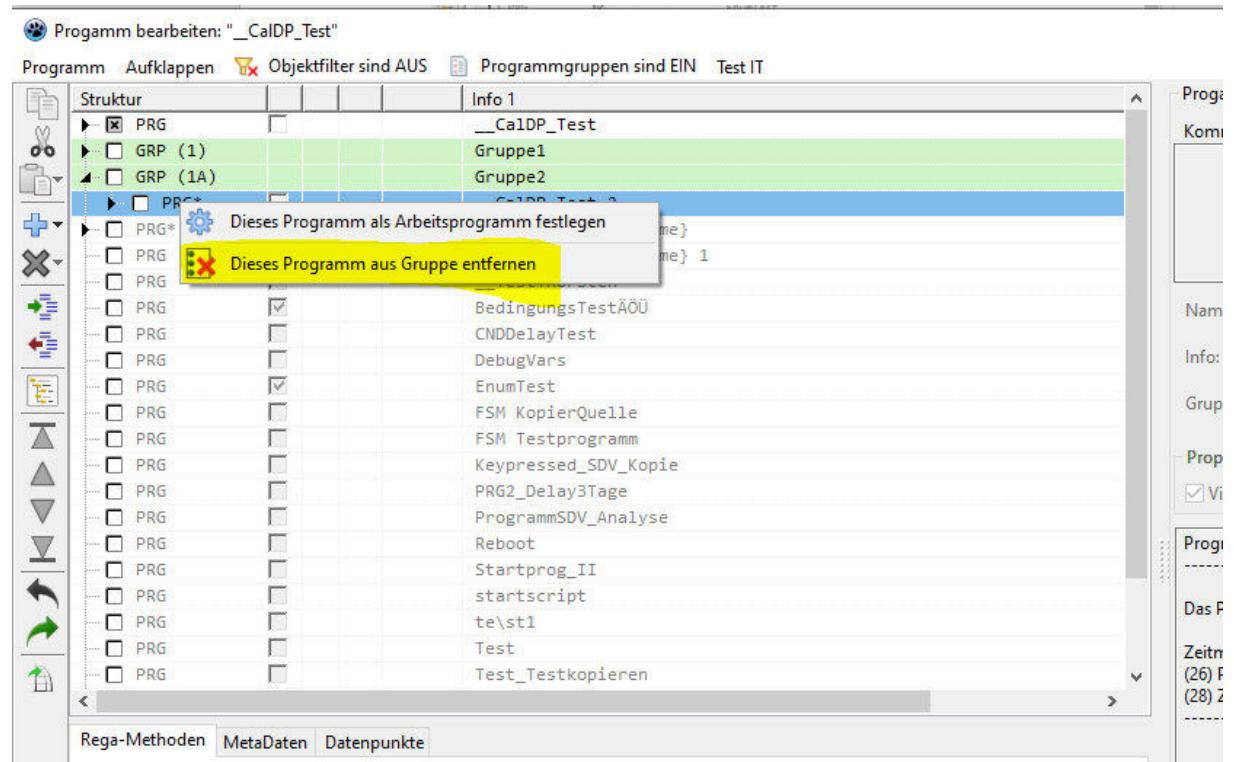
Rechte Maustaste auf den Combo: Gruppe



Hier gilt auch: Undo/Redo arbeitet hier wie immer im Arbeitsprogramm, die finale Zuweisung erfolgt erst durch Persistieren via Hochladen in die CCU

Als nicht Arbeitsprogramm:

Rechte Maustaste auf das Programm in der Gruppe:



Dann ist das Programm aus der Gruppe entfernt.

11.3 Drag – Drop

Über Drag Drop lassen sich Teile des Arbeitsprogrammes innerhalb des Arbeitsprogrammes verschieben bzw. Teile von Nicht Arbeitsprogrammen in das Arbeitsprogramm hineinkopieren.

Hier Beispiel: Dragmode Kopieren von einem Objekt aus einem Nicht-Arbeitsprogramm

Selektiert:	1	SCND	Clipboard:	1	SCND	DragMode: Kopieren
-------------	---	------	------------	---	------	--------------------

Die Zeilen, wo das markierte Objekt abgelegt werden darf, werden grün eingefärbt. Bei Zeilen wo keine Ablage möglich ist, ist der Dragcursor auch ein stilisiertes Verbottsschild.

Hier im Beispiel: eine SCND ist markiert, sie darf Abgelegt werden im MAIN, in einer CND und in SCNDs.

The screenshot shows the Script-Developer V5.03.xx interface with the following details:

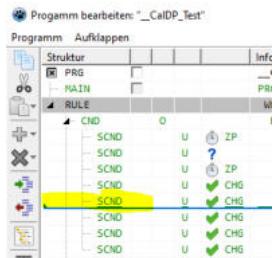
- Toolbar:** Includes icons for file operations (New, Open, Save, Print), selection tools (Select, Cut, Copy, Paste), and search.
- Menu Bar:** Shows "Programm bearbeiten: '_CalDP_Test'" and "Aufklappen".
- Left Sidebar:** Contains icons for Struktur (Structure), Praktische Werkzeuge (Practical Tools), and others.
- Main Area:** Displays a hierarchical tree structure of program blocks:
 - PRG
 - MAIN
 - RULE
 - CND 0 (highlighted in blue)
 - SCND (highlighted in blue)
 - DST
 - SDST 0
 - SDST 0
 - SDST 3610
 - SDST 0
 - PRGS*
 - MAIN
 - RULE
 - CND 0 (highlighted in blue)
 - SCND (highlighted in blue)
- Status Bar:** Shows "Rega-Methoden" and "MetaDaten".

Genaues Bestimmen der Ablageposition. Am besten an diese Vorgehensweise halten, diese ist immer zielführend.

Objekt markieren mit rechter Maustaste, Maustaste aber ca 1 Sekunde gedrückt halten, bis in der Statuszeile der Dragmode erscheint. Nun ist das Control im Verschiebemodus. Abgebrochen wird dieses mit Escape oder Ablegen des Objektes in einem verbotenen Bereich.

Dann in der ersten Spalte die Maus an die Position bewegen, wo abgelegt werden soll. Man sieht dort auch schon einen blauen Teilrahmen. Im Dragmode klappen unterlagerte Objekte beim Draufziehen der Maus automatisch auf.

Das Objekt wird hinter dem mit der Mausposition definierten (hier gelb markierten) Objekt eingefügt. Wenn dies das Parent Objekt des markierten Quellobjektes ist, erfolgt das Einfügen in der Childlist an letzter Stelle.



Das Objekt wird vor dem mit der Mausposition definierten (hier gelb markierten) Objekt eingefügt. Wenn dies das Parent Objekt des markierten Quellobjektes ist, erfolgt das Einfügen in der Childlist an erster Stelle.



Loslassen der Maus führt dann zum Ablegen des Objektes samt aller seiner Untergeordneten Objekte an der definierten Stelle.

Sollte dabei ein Fehler passiert sein.. Richtig, der SDV-Programmmeditor hat eine UNDO- REDO Funktion, mit der sich alle Schritte (bis auf Löschen von Programmen) zurücknehmen und auch wieder vorspulen lassen.

11.4 Undo Redo

Der SDV-Programmmeditor hat einen recht tiefen Undo-Redo Stack (zum jetzigen Zeitpunkt 100 Stufen). Damit lassen sich alle Aktionen (Einfügen, Verschieben, Drag Drop, Werteänderungen, einfach alles, bis auf das komplette Löschen von Programmen) auf Knopfdruck wieder zurücknehmen. Die Idee dazu stammte von RolandT, die Möglichkeiten sind natürlich weitreichend, die Programmierung dessen hatte es aber in sich.

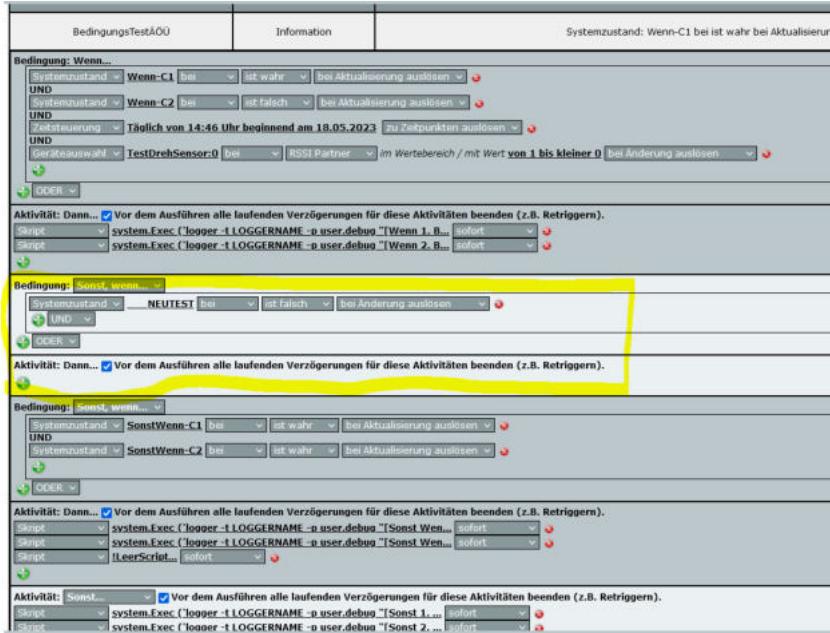
Aber. Es Geht.

11.5 Exemplarische Aufgabenstellungen

Die nun hier beschriebenen Aufgabenstellungen sind mit dem SDV-Programmeditor in wenigen Sekunden erledigt, während dies in der WebUI nur durch aufwendige und langwierige Maus-Clicks Orgien zu lösen ist.

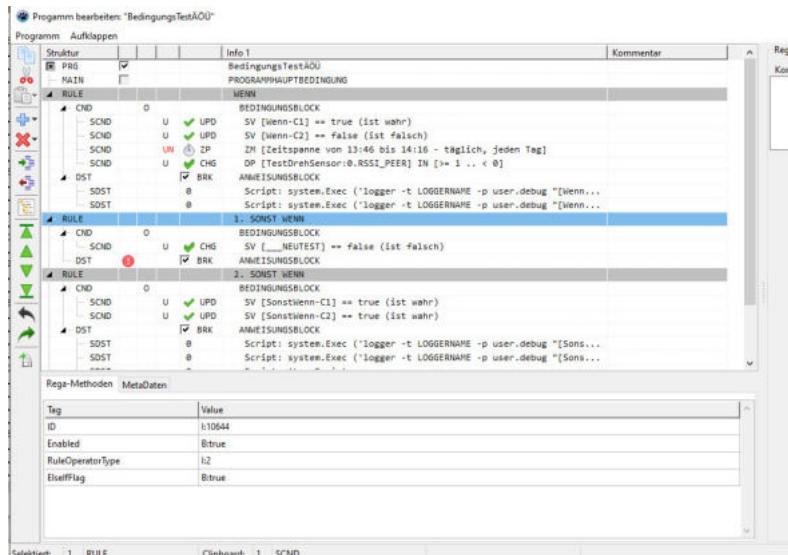
11.5.1 Löschen einer Regel inmitten eines Programmes

Gegeben sei folgendes WebUI Programm. Die Gelb umrandete Regel mittendrin soll weg

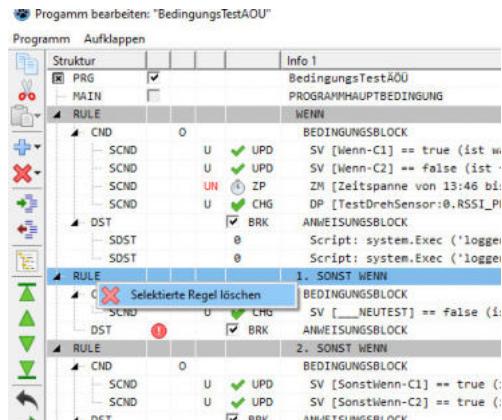


Vorgehensweise:

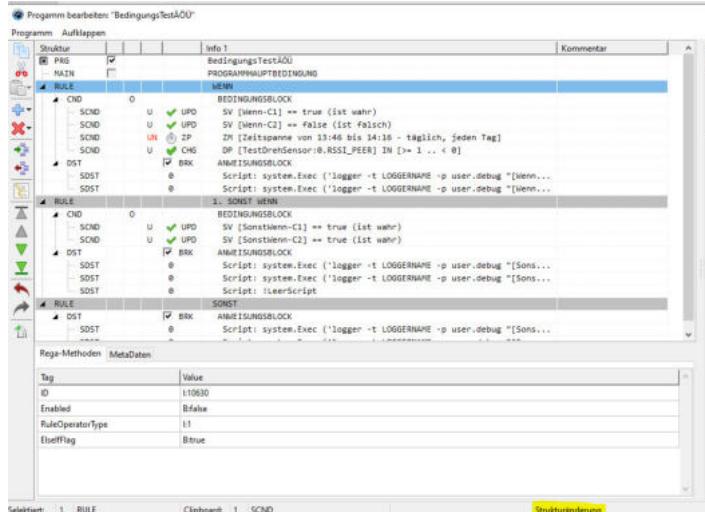
Das Programm im SDV als Arbeitsprogramm öffnen und die 1 -SonstWenn Bedingung selektieren (erkennt man schnell wieder an der leeren Aktivität, heisst keine Single Destinations wie in der WebUI)



Das rote Kreuz oder Ctrl-Del öffnet das Bestätigungs Menü:

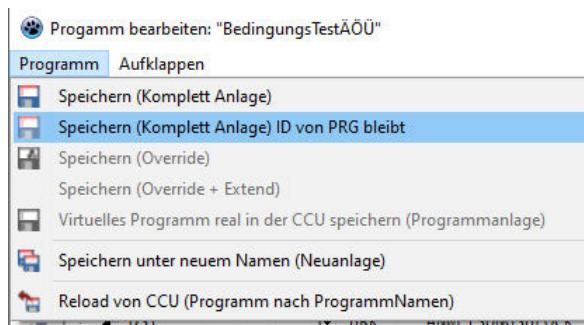


Bestätigen und weg ist die Regel:

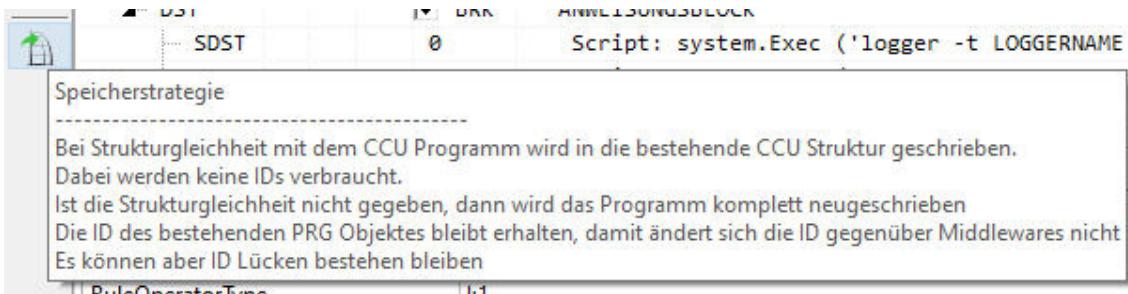


Wir sehen unten auch: es ist eine Strukturänderung. Diese Änderung könnte ich jetzt mit Undo auch wieder zurücknehmen. Wollen wir aber nicht. Wichtig: Der SDV Programmeditor arbeitet ja immer im Speicher des PC. Wir haben nun in der Baumstruktur diese Regel eliminiert, diese muss aber noch auf die CCU. Solange wir nicht auf Speichern gedrückt haben, haben wir in der CCU noch nichts geändert oder kaputt gemacht.

Also Speichern: oben auf Programm zeigt uns schon die eingeschränkten Möglichkeiten bei einer Strukturänderung, hier ist nur Neuanlage, kein Override möglich (siehe 11.1.5 Speicherstrategien)

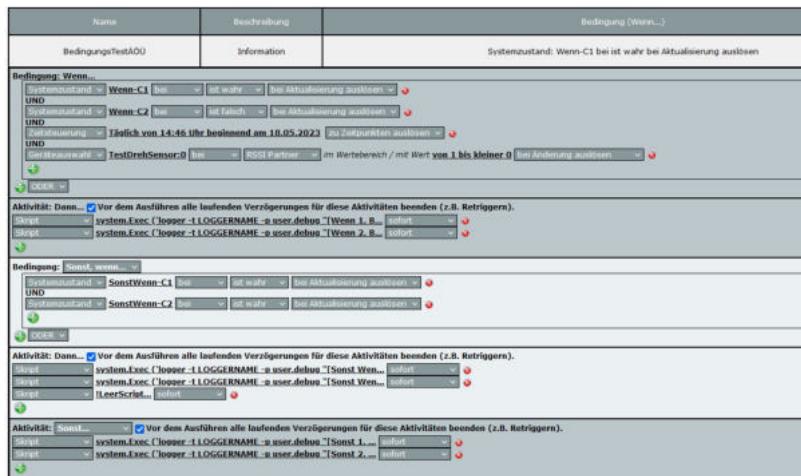


Da die Upload Taste sich im Setup des SDV mit der entsprechenden Strategie einstellen lässt (und vorab beim ersten Mal die von mir empfohlene Strategie schon voreingestellt hat, lässt ich mit der Maus via HINT kontrollieren)



Schreibt ein Druck auf die Upload Taste das Programm dann neu in die CCU. Im Gegensatz zur WebUI wird dieses aber jedes Mal sauber neu angelegt und alle alten Reste entfernt, so dass das berühmte Kaputteditieren von Programmen so nicht vorkommen kann.

Und dann nach dem Schreiben des Programmes sieht die WebUI wie gewollt so aus:



Das Ganze dauerte ein paar Sekunden, da braucht die WebUI schon länger beim Handeditieren um den Auswahldialog einer Systemvariablen aufzuklappen. Dafür würde jetzt jede weitere Bearbeitung keine IsID Lücken entstehen lassen.

11.5.2 Einfügen einer neuen Regel vor dem Wenn

Es soll eine neue Regel vor dem Wenn eingefügt werden, aus dem bestehenden Wenn soll dann das erste Sonst Wenn werden.

The screenshot shows a table of rules. A blue arrow points from a callout box labeled "Hier neue Regel einfügen" to the first row of the table, which contains a "WENN" (If) condition. The table has columns for Name, Beschreibung (Description), and Bedingung (Wenn...).

Name	Beschreibung	Bedingung (Wenn...)
BedingungsTestÄÖU	Information	Systemstandard: Wenn C1 bei Wahr bei Aktualisierung aktualisieren
WENN:		Bedingung: Wenn... C1 ist wahr C2 ist falsch UND Täglich von 13:46 Uhr aktualisiert am 10.05.2022 UND TestDrehSensor:0 ist Wahrdeutsch / mit Wert 1000
Aktivität: Dazu... <input checked="" type="checkbox"/> Vor dem Ausführen alle laufenden Verzögereungen für diese Aktivitäten beenden (z.B. Refreshen).		system.Exec ('logger :LLONGERNAME: a.setsValue ("Wenn 1. R.")')
		system.Exec ('logger :LLONGERNAME: a.setsValue ("Wenn 2. R.")')
Bedingung:		Bedingung: Wenn... Sensoren: C1 ist wahr Sensoren: C2 ist falsch UND Täglich von 13:46 Uhr aktualisieren
Aktivität: Dazu... <input checked="" type="checkbox"/> Vor dem Ausführen alle laufenden Verzögereungen für diese Aktivitäten beenden (z.B. Refreshen).		problemExec ('logger :LLONGERNAME: a.setsValue ("Sonst Wenn")')
		problemExec ('logger :LLONGERNAME: a.setsValue ("Sonst Wenn")')
Aktivität: Leer... <input checked="" type="checkbox"/> Vor dem Ausführen alle laufenden Verzögereungen für diese Aktivitäten beenden (z.B. Refreshen).		Leer

Ich wünsche mit der WebUI dann schon mal viel Spass ^^

Im SDV Programmeditor Arbeitsprogramm definieren, erste Regel anklicken, dann entweder den + Button oder CTRL+PlusTaste für Erzeugen

The screenshot shows the SDV Programmeditor with the program "BedingungsTestÄÖU". A context menu is open over the first "WENN" (If) block, with the option "Neue Regel vor aktueller Position erzeugen" (Create new rule before current position) highlighted. The menu also includes options like "Neue Regel nach aktueller Position erzeugen", "Neue Regel an erster Position erzeugen", and "Neue Regel an letzter Position erzeugen". The main workspace shows the program structure with a "BEDINGUNGSBLOCK" (Condition Block) containing code related to RSSI_PEER and a "SCND" (SCND) command.

Neue Regel vor aktueller Position bzw. Neue Regel an erster Position werden eine neue erste Regel erzeugen.

The screenshot shows the SDV Programmeditor after creating a new rule before the first condition. The program structure now includes a new "WENN" (If) block at the top level, labeled "1. SONST WENN". Below it is the original "WENN" block labeled "2. SONST WENN". The original "WENN" block now has a "BRK" (Break) command attached to its "DST" (Destination) branch. The workspace shows the updated program structure with the new rule added.

Es existiert nun ein neues leeres Wenn an erster Position, das alte Wenn ist zum 1. SonstWenn geworden. Dies kann nun im SDV-Programmeditor oder in der WebUI bearbeitet werden. Wieder Upload in die CCU und wir haben ein neues, leeres Wenn. Das Ganze dauerte auch wieder nur wenige Sekunden.

Und das Ergebnis wie erwartet in der WebUI

The screenshot shows the configuration of a rule named 'BedingungTestAÖU' in the WebUI. The rule consists of three parts:

- Wenn-C1:** A condition where 'Wenn-C1' is true (ist wahr). It includes a timer condition 'ZP [Zeitspanne von 13:46 bis 14:16 - täglich, jeden Tag]' and a script 'DP [TextDrehSensor:0.RSSI_PERR] IN [>= 1 .. < 0]'. An action 'BRK ANHEISUNGSBLOCK' is triggered.
- Wenn-C2:** A condition where 'Wenn-C2' is false (ist falsch). It includes a timer condition 'ZP [Zeitspanne von 13:46 bis 14:16 - täglich, jeden Tag]' and a script 'DP [TextDrehSensor:0.RSSI_PERR] IN [>= 1 .. < 0]'. An action 'BRK ANHEISUNGSBLOCK' is triggered.
- Wenn-C3:** A condition where 'Wenn-C3' is true (ist wahr). It includes a timer condition 'ZP [Zeitspanne von 13:46 bis 14:16 - täglich, jeden Tag]' and a script 'DP [TextDrehSensor:0.RSSI_PERR] IN [>= 1 .. < 0]'. An action 'BRK ANHEISUNGSBLOCK' is triggered.

11.5.3 Vertauschen des Wenn und des 1. Sonst-Wenn Regelblocks

Aus dem 1. Beispiel soll die Wenn Regel mit der 1. Sonstwenn Regel vertauscht werden. Es bieten sich 2 Lösungsansätze an.

1. Ausschneiden der Kompletten 1. Sonst Wenn Regel
Einfügen der Sonst-Wenn Regel aus der Zwischenablage vor der Wenn Regel
2. Drag/Drop der Sonstwenn Regel und Ablegen vor der Wenn Regel
3. Verschieben der Regel via Button

Los geht's. Lösung mit Ausschneiden:

The screenshot shows the rule structure in the CCU software. A 'SONST WENN' block is currently selected in the clipboard. The rule structure includes:

- Wenn:** A condition where 'Wenn-C1' is true (ist wahr). It includes a timer condition 'ZP [Zeitspanne von 13:46 bis 14:16 - täglich, jeden Tag]' and a script 'DP [TextDrehSensor:0.RSSI_PERR] IN [>= 1 .. < 0]'. An action 'BRK ANHEISUNGSBLOCK' is triggered.
- Sonst:** A condition where 'Sonst Wenn-C1' is true (ist wahr). It includes a timer condition 'ZP [Zeitspanne von 13:46 bis 14:16 - täglich, jeden Tag]' and a script 'DP [TextDrehSensor:0.RSSI_PERR] IN [>= 1 .. < 0]'. An action 'BRK ANHEISUNGSBLOCK' is triggered.

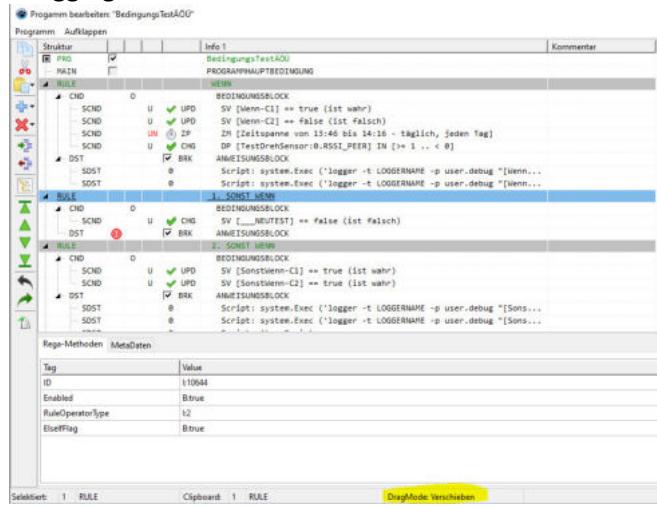
Dann die Wenn Bedingung anklicken, Einfügen anwählen vor der angewählten Position. Unten sehen wir, im Clipboard liegt die Rule, die wir eben ausgeschnitten haben. Also nun Einfügen.
Und wir sehen das Ergebnis, die Regel liegt nun vor dem alten Wenn

The screenshot shows the rule structure in the CCU software after reordering. The 'SONST WENN' block is now positioned before the original 'Wenn' block. The rule structure is identical to the one in the previous screenshot, but the order of the blocks has been changed.

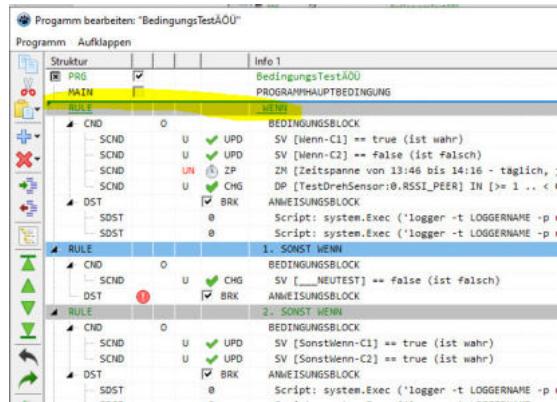
Nach dem Übertragen ist nun das geänderte Programm in der CCU.

Ansatz 2: Drag Drop

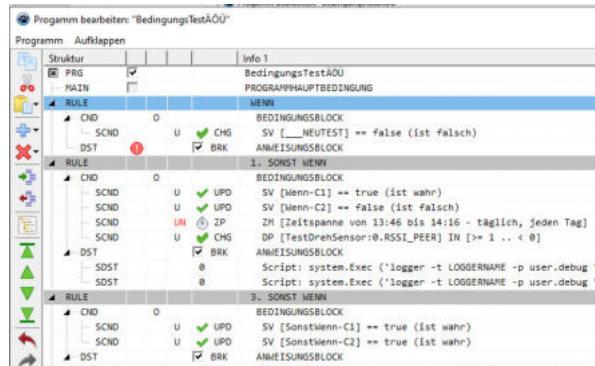
Die zu verschiebende Rule (das 1. Sonst wenn auswählen), rechte Maustaste, die berühmte Sekunde gedrückt halten, und wir sehen: DragMode: verschieben. Innerhalb des Arbeitsprogrammes ist Dragging immer Move.



Nun noch die Rule auf das erste, auch grün dargestellte Wenn schieben, dabei auf den oberen Teil des Rahmens für vor der Position achten



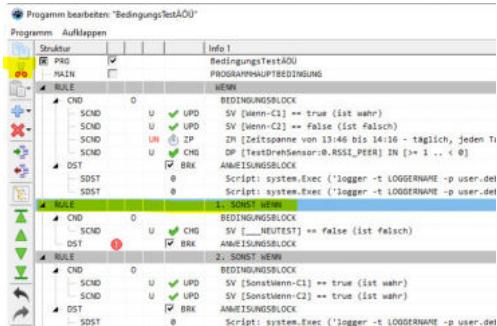
Und fertig:



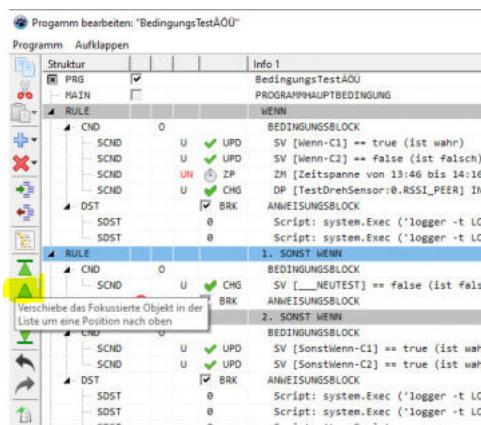
Nach dem Übertragen ist nun das geänderte Programm in der CCU.

Zum Schluss noch Verschieben über Buttons

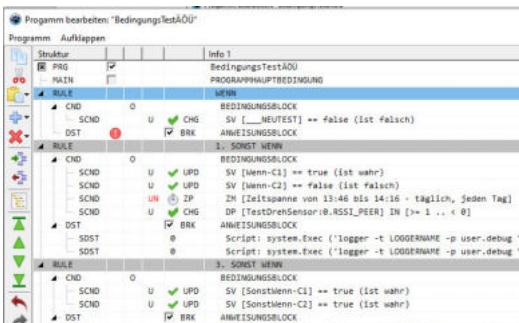
Wir wählen wieder die 2. Regel



Nun aber den Button „ein Element nach oben verschieben“



Ergibt dann durch einen Tastendruck das gewünschte Ergebnis



Nach dem Übertragen ist nun das geänderte Programm in der CCU.

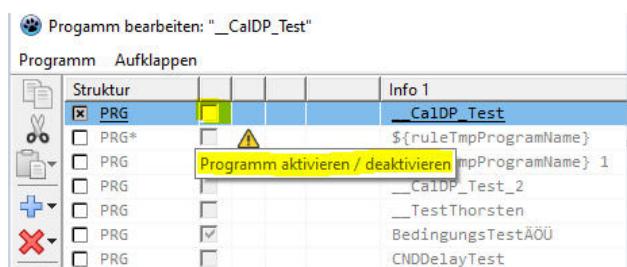
11.6 Anzeigen bzw. Ändern von Programmen oder Programmunterobjekten

In einem definierten Arbeitsprogramm können Werte geändert werden. Ist ein Programm nicht als Arbeitsprogramm definiert, so werden zwar alle Werte angezeigt, diese werden aber ausgegraut dargestellt.

Nicht Arbeitsprogramm: (ausgegraut)	Arbeitsprogramm (editierbar)
 <p>ProgramName: CDRP Test</p> <p>Requirements:</p> <ul style="list-style-type: none"> CDRP <p>Properties:</p> <ul style="list-style-type: none"> Variable Internal <p>Programm "CDRP Test"</p> <p>Das Programm hat folgende Eingänge:</p> <ul style="list-style-type: none"> Zulieferer (Maschine 2) Produktionsf. 20 - Automatische Werkzeugmaschine Werkstatt 1 - Regelung, je nach Läng <p>Eingangsvariablen für die Produktion (Maschine 1)</p> <ul style="list-style-type: none"> (0) D1 [Durchfluss] --> 1000000 (0) D2 [Durchfluss] --> 1000000 (0) D3 [Durchfluss] --> 1000000 (0) D4 [Durchfluss] --> 1000000 (0) D5 [Durchfluss] --> 1000000 (0) SV [Bauzeit] --> 1000 	 <p>ProgramName: CDRP Test</p> <p>Requirements:</p> <ul style="list-style-type: none"> CDRP <p>Properties:</p> <ul style="list-style-type: none"> Variable Internal <p>Programm "CDRP Test"</p> <p>Das Programm hat folgende Eingänge:</p> <ul style="list-style-type: none"> Zulieferer (Maschine 2) Produktionsf. 20 - Automatische Werkzeugmaschine Werkstatt 1 - Regelung, je nach Läng <p>Eingangsvariablen für die Produktion (Maschine 1)</p> <ul style="list-style-type: none"> (0) D1 [Durchfluss] --> 1000000 (0) D2 [Durchfluss] --> 1000000 (0) D3 [Durchfluss] --> 1000000 (0) D4 [Durchfluss] --> 1000000 (0) D5 [Durchfluss] --> 1000000 (0) SV [Bauzeit] --> 1000

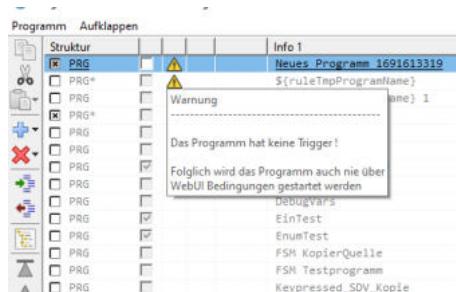
11.6.1 Programmobjekt (PRG)

Veränderbar beim Programmobjekt ist der Kommentar, der Name und die ProgrammInfo sowie die Eigenschaften Visible und Internal im linken Feld. Aktive wird geändert direkt im Kasten im Programm Bereich des SDV-Programmeditors



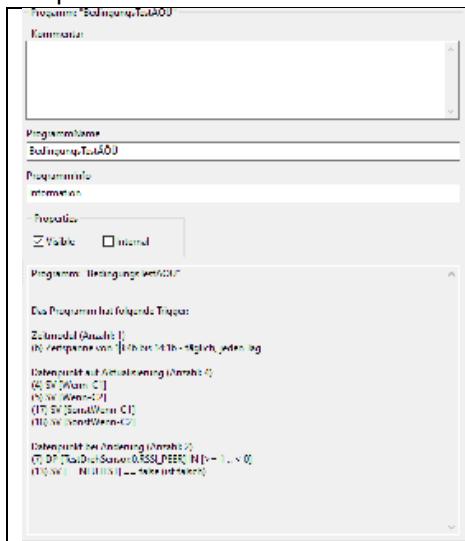
Der Editor macht auch gleichzeitig eine Analyse. Durch Darstellen des gelben oder des roten Ausrufezeichens macht der Editor auf unpassende Angaben aufmerksam.

Bei einem neuen Programm fehlen natürlich Trigger, folglich erscheint da im Hint:



Bei einem Programm wird im Ein/Ausgabefenster auch gleichzeitig dargestellt, welche Einzelnen Bedingungen das Triggern des Programmes und damit das logische Überprüfen der einzelnen Regeln auslösen.

Beispielhaft:



Hier in dem Programm die Trigger

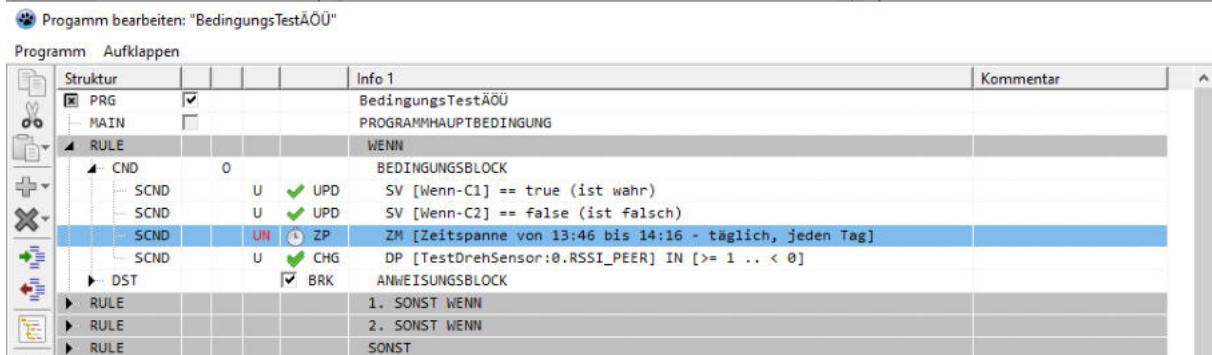
-1 Zeitmodul

-4 Datenpunkte, die auf aktualisierung Triggern

-2 Datenpunkte, die bei Änderung triggern (mit der Angabe des Bereichs)

Wenn ich nun die genaue Bedingung und deren Einstellungen wissen will, kann ich im Programmteil suchen, es geht aber auch elegant schneller, Doppelklick auf die Zeile im Ein/Ausgabe Bereich und der passende Teil wird im Programm Vereich aufgeklappt und der Cursor springt an die Bedingung:

Hier exemplarisch: ich will ins Zeitmodul, also Doppelklick auf die Zeitspanne



Im Ein/Ausgabebereich stehen dann die genauen Informationen zu der angezeigten Singlecondition, in diesem Fall das Zeitmodul. Dazu aber später mehr.

11.6.2 Programmhauptbedingung (Main-Condition)

Die WebUI kann mit einer Main Condition nichts anfangen. Nichts desto trotz ist dieses ein extrem nützliches Instrument, um Programme einfacher und auch übersichtlicher zu gestalten. Der Main-Condition ist ein eigenes Kapitel gewidmet. In einer Main Condition kann nur der Kommentar geändert werden, erzeugen lassen sich nur einzelne Bedingungen (SCND). Aus Kompatibilitätsgründen lässt sich im Setup die Main-Condition komplett ausblenden.

11.6.3 Regelblock

In einer Regel kann nur der Kommentar vergeben werden. Die DST in einer Regel kann nicht gelöscht oder verschoben werden. In einem Regelblock lassen sich Bedingungsblöcke erzeugen.

11.6.4 Bedingungsblock

Der Bedingungsblock ist Teil seiner Regel und enthält eine oder mehrere Bedingungen (Ausnahme: die Sonst Regel enthält keine Bedingungsblöcke und auch keine Bedingungen. Von einem Bedingungsblock kann die Logische Verknüpfung eingestellt und ein Kommentar eingegeben werden.

Dazu in der Spalte mit der Logischen Verknüpfung klicken, die Spalte verbreitert sich automatisch und das Auswahlmenü für die Verknüpfung erscheint.

Struktur		Info 1	
PRG	<input checked="" type="checkbox"/>	BedingungsTe:	
MAIN	<input type="checkbox"/>	PROGRAMMHAUPT	
RULE		WENN	BEDINGUNGSBLOCK
CND	O	U	UPD SV [Wenn-]
SCND	U	U	UPD SV [Wenn-]
SCND	O	UN	ZP ZM [Zeits;
SCND	U	U	CHG DP [TestDi
DST		BRK	ANWEISUNGSE

11.6.5 Die Einzel-Bedingung

Die Einzelbedingung ist Teil einer Bedingung. Von einer Einzelbedingung kann wie gewohnt der Kommentar geändert werden sowie die Art der Logischen Verknüpfung bzw. Die Trigger / Prüfbedingung.

11.6.5.1 Logische Verknüpfung

CND		0	BEDINGUNGSBLOCK
SCND	U	?	ZP ZM [Periodisch alle 180s - Astro nachts]
SCND	U	?	SV [AlwaysTrue] == true (ist wahr)
SCND	UN	ZP	ZM [Zeitspanne von 19:45 bis 20:15 - täglich, jeden Tag]
SCND	O	CHG	SV [DutyCycle] == 0.000000
SCND	ON	CHG	SV [WerteListeTest] == 1 (zweiter)
SCND	U	CHG	DP [HMI-eTRV-1:1.ACTIVE_PROFILE] == 1
SCND	U	CHG	SV [Systemvariable 1] IN [>= 1.000000 .. < 2.000000]
SCND	U	CHG	SV [Bängtest] == "hhhh"
DST		RETR	ANWEISUNGSBLOCK
SDST	0		Script: system.Exec ('logger -t LOGGERNAME -p user.debug *[Zeit
SDST	0		DP [HMI-eTRV-1:1.ACTIVE_PROFILE] := 1
enver	DATA		EV [Lautstahl Intervall] := 9 Minuten

Die Schreibweise stammt aus der SPS-AWL Programmierung. Seit einigen Firmware-Versionen hat die WebUI leider eine „Auffälligkeit“. Ob Absichtlich oder nicht entstanden weiß ich nicht. Einer der Vorteile dieser Notation ist, dass ich die Logik frei mischen kann. Das funktioniert auch und die WebUI kann es auch richtig anzeigen. ABER. Öffne ich ein Programm mit der erweiterten Logik des SDV-Programmmeditors in der WebUI, so wird diese Logik zwar richtig dargestellt, aber – beim Schreiben wird völlig unnötig eigentlich bei allen SCNDs die Verknüpfung der ersten SCND eingetragen. Halt nur das, was die „dumme“ WebUI kann. Es ist jetzt die Entscheidung des Anwenders: Ich will die WebUI jederzeit nutzen können, dann Finger weg von der freien Notation. Wenn ich allerdings den SDV-Programmeditor als Mastereditor nutze, so merkt sich dieser „seine“ Logik. Wenn die WebUI die dann beim Speichern zerstört hat, reicht es das Programm wieder im SDV-Programmeditor zu öffnen und einfach via Override zu speichern, dann wird wieder die „gemerkte“ SDV-Editorlogik in die Rega geschrieben. Solange das nicht gepatcht ist in der WebUI geht's leider nicht anders.

11.6.5.2 Triggerungsarten bzw. nur Prüfen

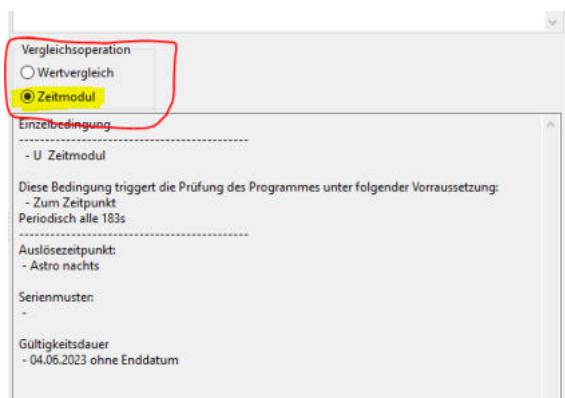
Nur prüfen:

	NP	Nur Prüfen triggert kein Programm, sondern diese Bedingung wird einzig und alleine beim Logiktest geprüft.
	ZM	Zeitmodul zu Zeitpunkt (Programmauswertung wird getriggert)
	UPD	Bei Aktualisierung (Update). Ein neues Eintreffen eines Wertes führt zur Triggerung der Programmauswertung
	CHG	Bei Änderung (Change): Ein neu eingetroffener Wert muss die Logische Bedingung erfüllen, in der vorherigen Auswertung aber diese noch nicht erfüllt haben.
	IMP	Bei Tastendruck (Nur bei Longpressed oder Shortpressed)

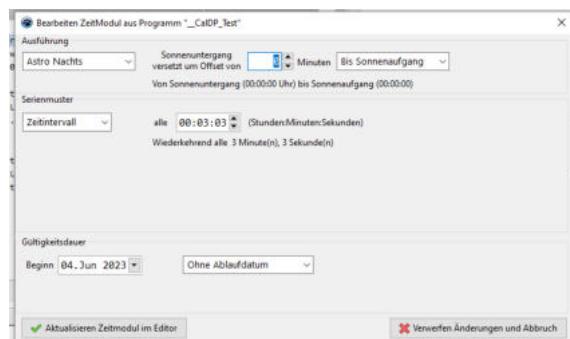
11.6.5.3 Zeitmodul

Eine Einzelbedingung kann entweder ein Zeitmodul sein oder eine Vergleichsoperation.

Die Umschaltung wird in dem Auswahlmenü getätigkt



In dem Informationsfenster werden die Daten zu dem Zeitmodul dargestellt. Geändert wird durch Doppelklick in das Zeitmodul-Informationsfenster oder aber durch Doppelklick in die Info Zelle des Zeitmoduls im Programmeditor.



Das Menü entspricht dem Änderungsmenü aus dem Inspektor unter 4.10.1

11.6.5.4 Die normale Einzelbedingung (Variablenvergleich)

Ein Vergleich findet mit einem Gerätedatenpunkt (DP), einer Systemvariablen oder einer Alarmvariablen statt. Die Linke Seite bei dem Vergleich ist also immer eine Variable, hier abgekürzt Source genannt. Je nach Typ des Source ergeben sich folgende Vergleichsmöglichkeiten.

Source ändern:

Doppelklick auf die Sourcevariable öffnet das Datenpunkt-Auswahlmenüfenster. Es werden die Datenpunkte vorgeschlagen, die an dieser Stelle auch erlaubt sind (read bzw. event). Im Gegensatz zur WebUI ändert der SDV-Skripteditor, wenn nachträglich der Typ geändert wird, den Typ der Vergleichswerte automatisch richtig mit.

Tastendruck (kurz oder lang): Keine weitere Angabe nötig

String: vergleich nur auf gleich

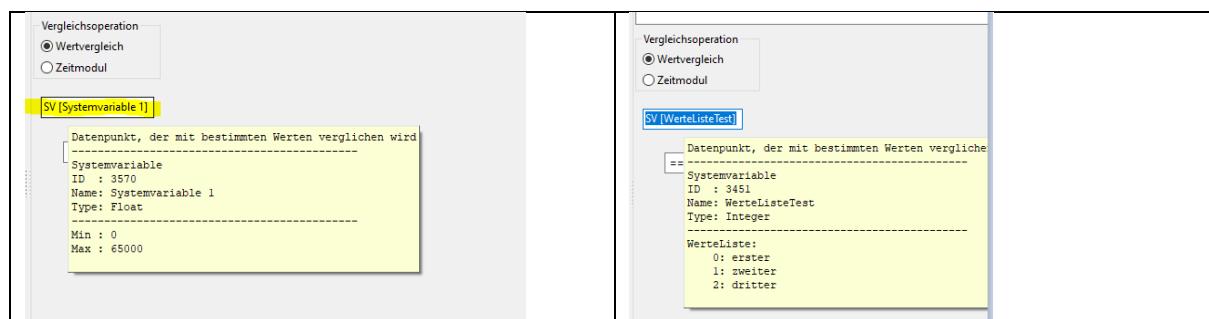
Enum: (Ein Integer mit zugeordneter Werteliste)

Hier ist es ein bisschen komplizierter, es kann ein Kompatibilitätsmodus zur WebUI eingestellt werden, hierbei ist wie in der WebUI nur == (bei) möglich. Die Rega beherrscht allerdings den vollen integer Vergleich. Nachteil, die WebUI kann es nicht darstellen.

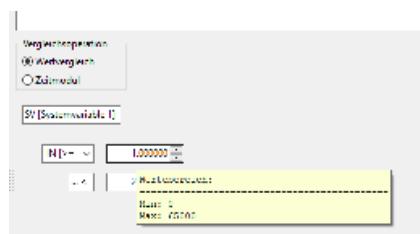


Integer/real: voller Vergleich

Der Hint auf die SourceVariable zeigt Einzelheiten zu der Variable an:



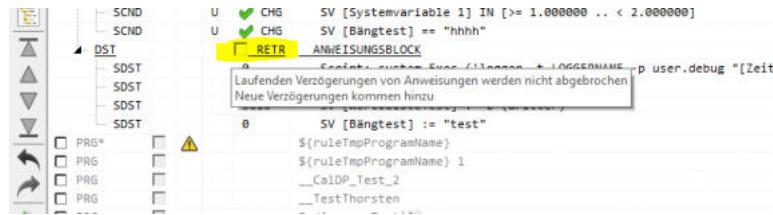
Der Hint auf den Vergleichswert zeigt den gültigen Wertebereich an



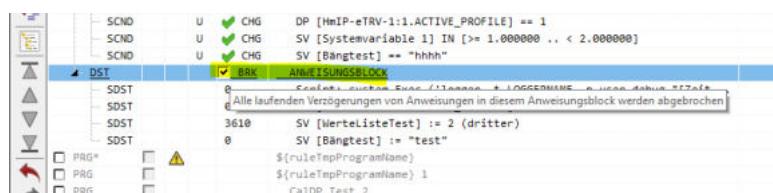
11.6.6 Anweisungsblock

Im Anweisungsblock wird festgelegt, ob die unter dem Anweisungsblock angelegten Anweisungen retriggert werden oder nicht (siehe auch Handbuch zur WebUI)

Dies wird im Editorfeld eingestellt



Bzw.



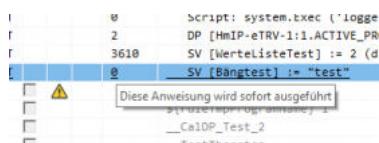
Als Gedankenstütze wird dieses im Hint von einer Einzelanweisung und auch im Eingabebereich einer Einzelanweisung dargestellt.

11.6.6 Die einzelne Anweisung

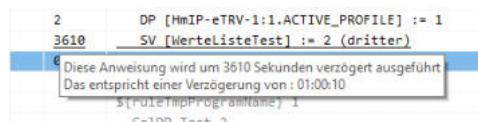
Die einzelne Anweisung (SDST für Single Destination). Eine Einzelanweisung kann ein Script sein oder eine Zuweisung. Beiden gemeinsam ist die Möglichkeit, die Ausführung zu verzögern

Einstellbarkeit im Editorfeld	Einstellbarkeit im Datenfeld

Im Hint werden über dem Editorfeld auch die Hinweise zu der Verzögerung angezeigt:

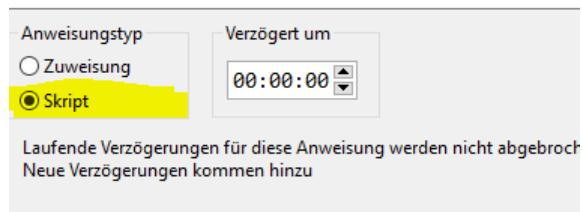


Bzw:



11.6.6.1 Script

Ein Script wird in einer Einzelanweisung hier angewählt:



Anweisungstyp: Zuweisung Skript
Verzögert um: 00:00:00

```
system.Exec ('logger -t LOGGERNAME -p user.debug "[Zei
Test]"');
if ("S98765") {
}
!test
#
```

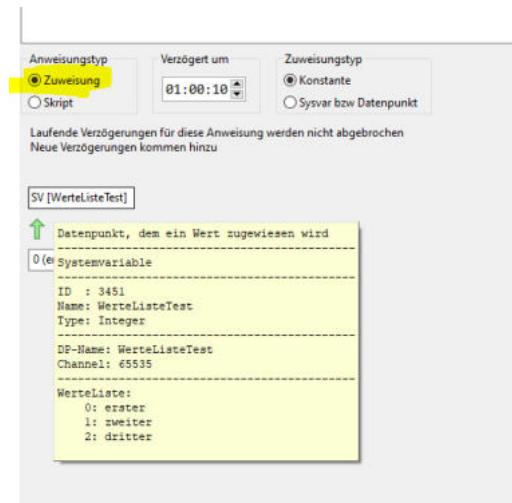
Im Anzeigefeld wird das Skript als Text dargestellt. Eine schnelle Änderung kann in dem Anzeigefeld vorgenommen werden. Hier natürlich ohne die weitreichenden Möglichkeiten des SDV-Skripteditors. Soll der Editor genutzt werden, so reicht ein Doppelklick in das Anzeigefeld, und das Skript wird im Skripteditor des SDV zur Bearbeitung bereitgestellt.

Wird der Skripteditor wieder verlassen, indem der Programmeditor geöffnet wird, so wird das Script des SDV-Programmeditors übernommen.

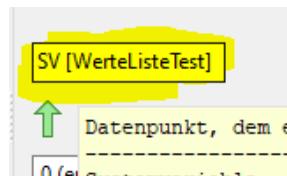
Soll dies nicht gewünscht sein → es gibt die Undo Taste.

11.6.6.2 Variablenzuweisung

Eine Variablenzuweisung weist, wie der Name schon sagt, einer Variablen eine Konstante oder den Wert einer anderen Variablen zu

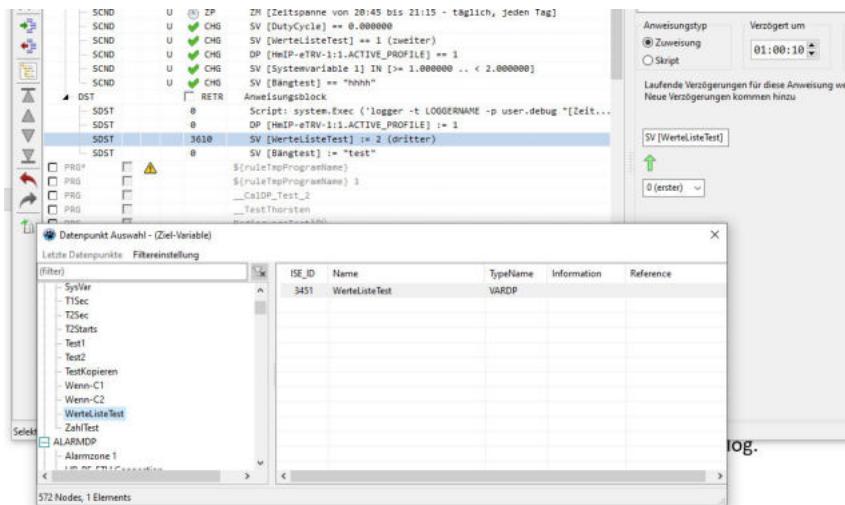


Der TargetVariable , in dem Fall SystemVariable SV mit dem Namen WertelistenTest wird ein Wert , hier in dem Fall eine Konstante zugewiesen.

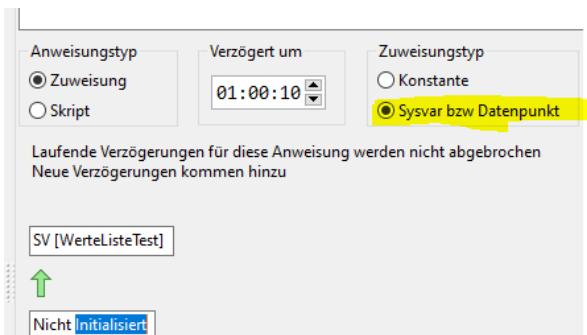


Im Hint werden Informationen über die Target Variablen angezeigt.

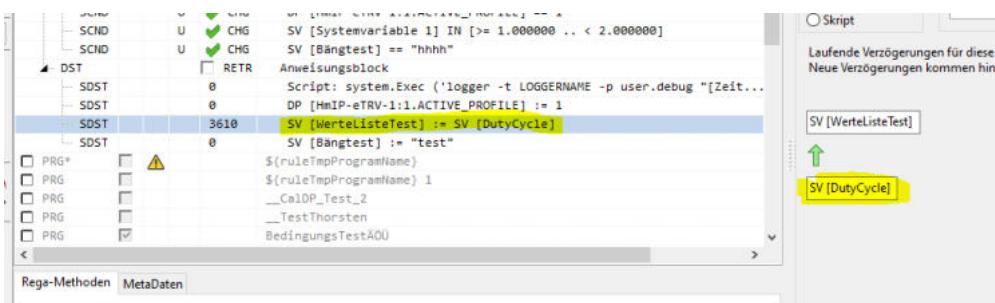
Ändern der Target Variablen (oder erste Zuweisung, wenn diese noch nicht definiert ist)
Doppelklick auf das Eingabefeld der Target Variablen öffnet den Auswahldialog.



Ebenso kann der Inhalt einer Variablen der Target Variablen zugewiesen werden



Dargestellt wird dieses so:



11.7 Die Maincondition

Die Maincondition ist eine Option in der Rega, die von der WebUI allerdings nicht genutzt wird. Der SDV konnte im Inspektor schon nach Programmen mit Mainconditions filtern, ebenso unterstützen die Backup Routinen des SDV die MainConditions (schon seit Jahren)

Was ist eine Maincondition ?

Eine Maincondition ist eine Art Hauptschalter, deren Bedingung erfüllt sein muss, damit das Programm überhaupt ausgeführt wird. Das bedeutet, dass mit einer Maincondition auch die Ausführung des Sonst Teiles unterbunden wird. Damit ist zum Beispiel die komplette Unterdrückung der Ausführung bei Systemstart möglich.

Ebenso werden Programme kürzer und Übersichtlicher, wenn mehrere Regeln zum Einsatz kommen und immer Bedingungen geprüft werden müssen, die sich so einfach in die Maincondition setzen lassen.

Warum unterstützt die WebUI das nicht ? Gute Frage... Nächste Frage

Kann die WebUI Mainconditions anzeigen ? Nein

Es gibt so eine goldene Regel: in einer Maincondition Bedingungen immer nur auf Nur Prüfen stellen.

Kleines Beispiel für ein kleines Programm, welches bei Systemboot nicht ausgeführt werden soll und nur ausgeführt werden soll, wenn AlwaysTrue auch true ist.

Das Original Programm:

Struktur		Info 1	
PRG	<input checked="" type="checkbox"/>	BedingungsTestÄÖÜ	
MAIN	<input type="checkbox"/>	Programmhauptbedingung	
RULE		Wenn	
RULE		1. Sonst Wenn	
RULE		2. Sonst Wenn	
RULE		Sonst	
DST		Anweisungsblock	
SDST		BRK	
SDST		0	Script: system.Exec ('logger -t LOGGERNAME -p user.debug "[Sons...")
SDST		0	Script: system.Exec ('logger -t LOGGERNAME -p user.debug "[Sons...")
PRG*			\${ruleTmpProgramName}

In den Wenn und Sonst wenn ließe sich die oben genannte Bedingung noch einbauen, im Sonst aber nicht

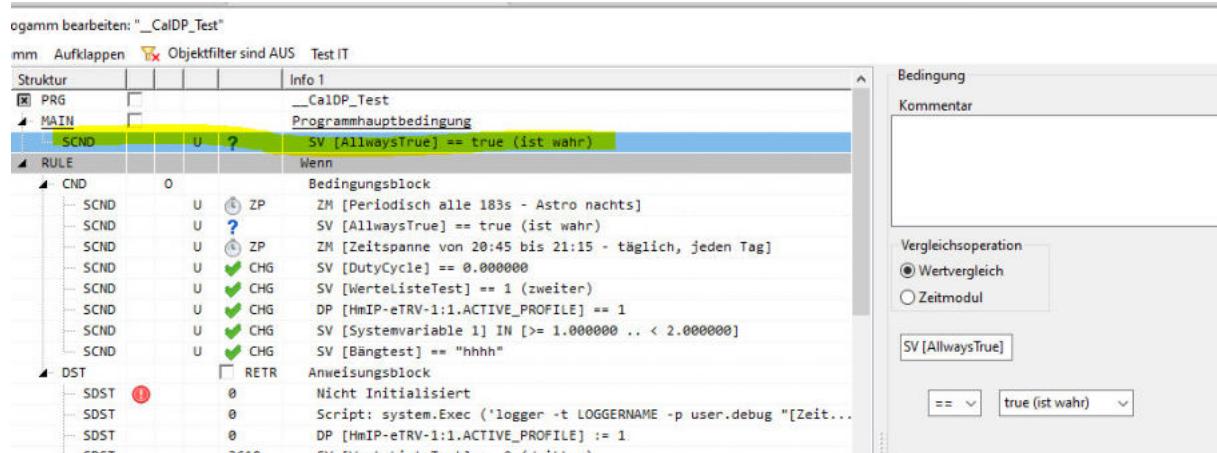
Also die Maincondition markieren, + Button und neue Bedingung anlegen

Programm Aufklappen Objektfilter sind AUS			
Struktur		Info 1	
PRG	<input checked="" type="checkbox"/>	BedingungsTestÄÖÜ	
MAIN	<input type="checkbox"/>	Programmhauptbedingung	
RULE		Wenn	
RULE		1. Sonst Wenn	
RULE		2. Sonst Wenn	
Neue Bedingung an erster Position erzeugen		st	Anweisungsblock
Neue Bedingung an letzter Position erzeugen			Script: system.Exec ('logger -t LOGGE
SDST		0	Script: system.Exec ('logger -t LOGGE

Ergibt neue Bedingung

Struktur				Info 1	
PRG				BedingungsTestÄÖÜ	
MAIN				Programmhauptbedingung	
SCND		U	?	Nicht Initialisiert	
RULE				Wenn	
RULE				1. Sonst Wenn	
RULE				2. Sonst Wenn	
RULE				Sonst	
DST		BRK		Anweisungsblock	
SDST		0		Script: system.Exec ('logger -t LOGGE	
SDST		0		Script: system.Exec ('logger -t LOGGE	

Die neuen Singleconditions müssen wir noch logisch mit der Reboot Bedingung verknüpfen. Also in das nicht definiert doppelklicken und Always True auswählen



12 License Disclaimer

Folgende Software wurde verwendet bei der Konzeptionierung und Programmierung des SDV

12.1 OpenSSL

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.]

*/

12.2 Synapse Ararat

Synapse TCP/IP and serial library

This project is freeware and open source under modified BSD style licence.

License

Each source file, what is protected by this license, have license text on the begin of file. Used license is very free for you. You can use it in any your application, include commercial. However follow next license statement, please!

⚠ International copyright law allows me to license you to use the Synapse code on these terms and conditions - if you do not comply with these terms and conditions then you cannot use the Synapse code!

If you use my libraries in commercial project, I will be very happy if you send me a some [donation](#). Thank you very much!

BDS style license

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

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. Neither the name of Lukas Gebauer nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "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 REGENTS 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.

12.3 Compiler und RAD

Programmiert wurde der SDV unter Lazarus unter Win64

