


27. August 2025

© Symcon GmbH 2025
www.symcon.de

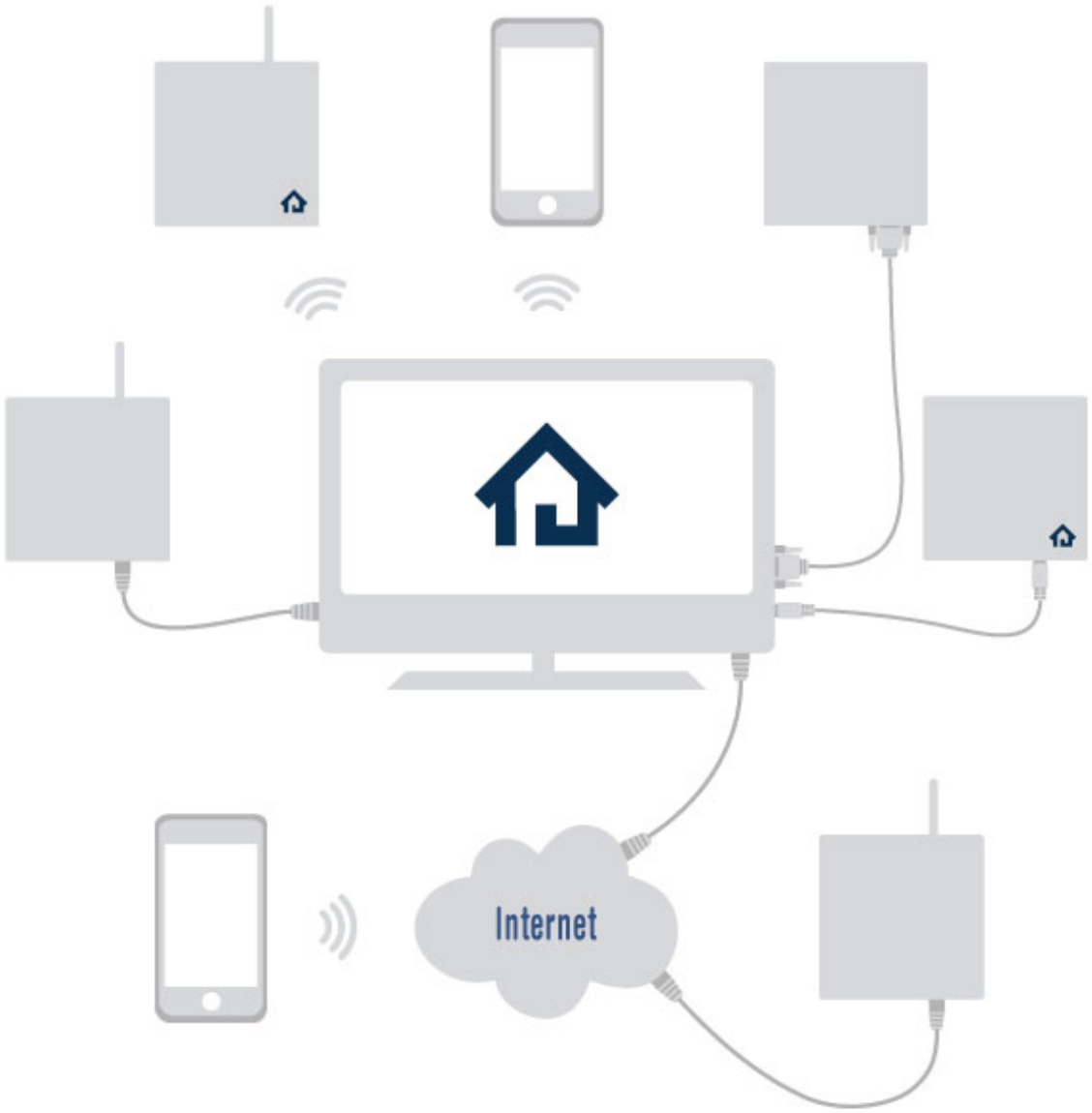
# Einführung

Dies ist die Dokumentation zu IP-Symcon.
Ziel ist es, sowohl technische Hintergründe, Basiswissen, Funktionsweisen, als auch die Entwicklung eigener Skripte und Automatisierung zu unterstützen. Beispiele werden hierbei verwendet um die Anschaulichkeit zu erhöhen.
Die gesamte Dokumentation steht jederzeit online, als auch offline zur Verfügung.
Was IP-Symcon ist, wird auf der Produktseite beschrieben.

## WIE FUNKTIONIERT IP-SYMCON

### HERSTELLERUNABHÄNGIGKEIT

IP-Symcon verknüpft verschiedenste Geräte von verschiedenen Herstellern unter einem Dach. Ein einheitliches Schema macht diese innerhalb der Software zugänglich.



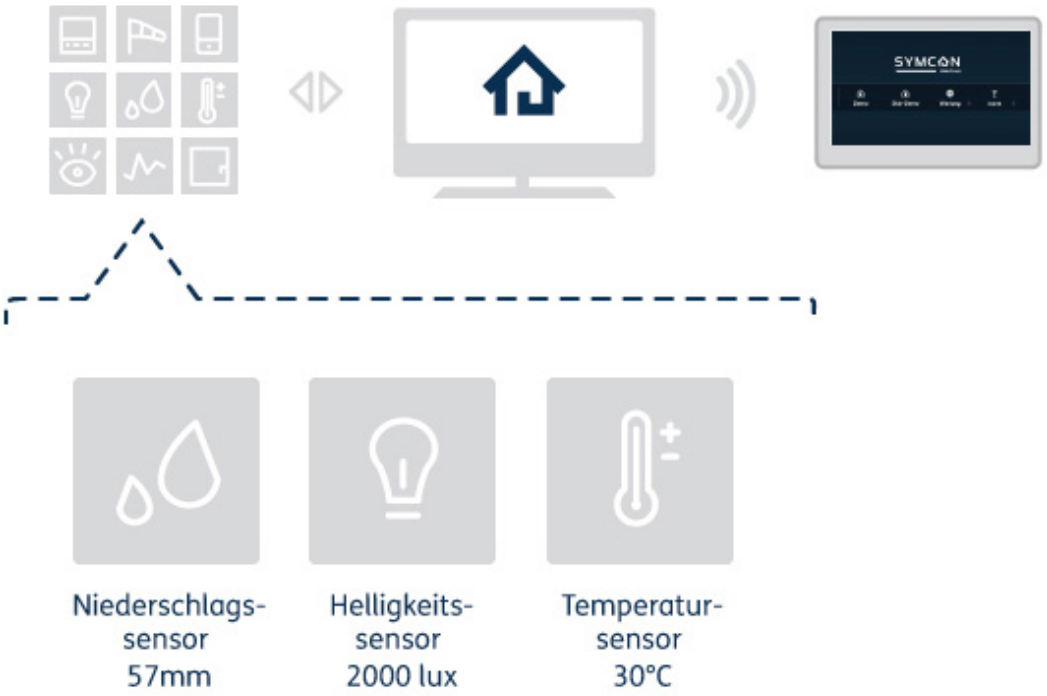
## ANBINDUNG

IP-Symcon nutzt dazu eigene Konfiguratoren, die Hersteller eigenen Protokolle und stellt über ein einfachen modularen Aufbau eine Verbindung zu den Geräten her. Dabei ist es egal, ob das Herstellersystem Funk- oder Kabelgebunden ist. Die Software interpretiert die Verbindung immer nach dem gleichen Schema: Geräteinstanz -> Gateway -> I/O



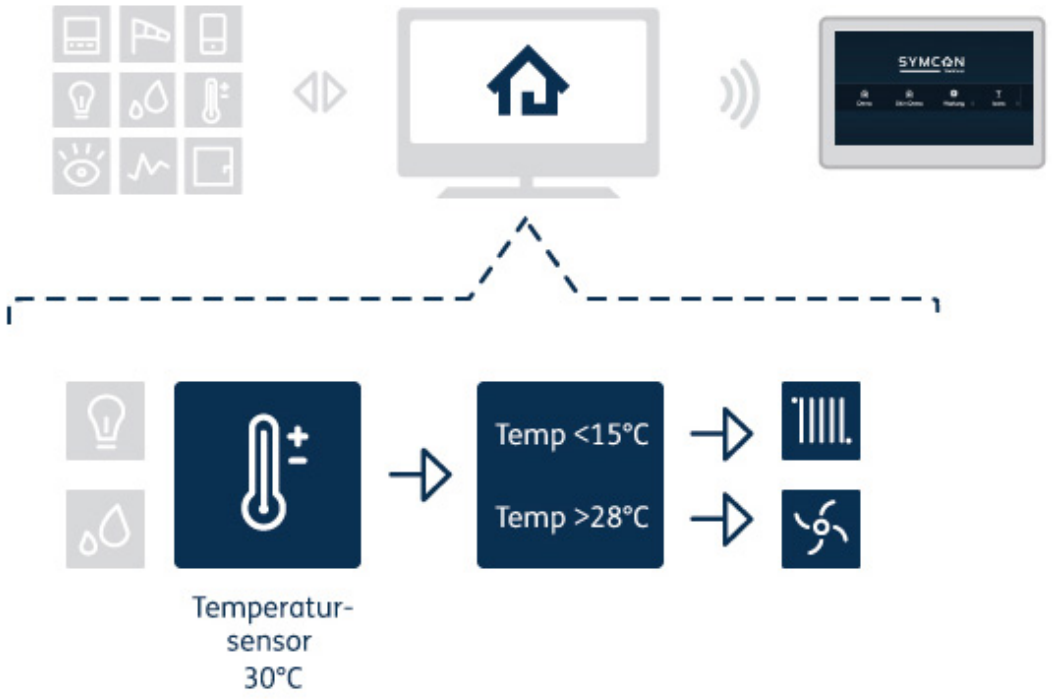

## GERÄTESTATUS

Wenn die Verbindung steht, können Variablen und Zustände der Geräte von IP-Symcon ausgelesen und gesetzt werden. Das Gerät steht zur Steuerung bereit.




## VERWALTUNG

Die Verwaltungskonsole ist das technische Nervenzentrum. Über diese können die an den Server angeschlossenen und eingerichteten Geräte angesteuert werden. Hier wird auch die Intelligenz und tatsächliche Automatisierung via Skripten eingerichtet und verwirklicht.




## VISUALISIERUNG

IP-Symcon stellt anschließend die Visualisierung der eingerichteten Geräte zur Verfügung und über das WebFront werden Funktionen und Daten anschaulich zur Verfügung gestellt. Auch ist ein Steuern über die Visualisierung kein Problem.




## ERSTE SCHRITTE

Über den Schnelleinstieg werden viele Grundlagen erklärt.

# Systemvoraussetzungen

## HARDWARE

Für den Betrieb von IP-Symcon ist ein System mit den aufgeführten Betriebssystemen nötig. Ein LAN-Anschluss ist für die meisten gängigen Gateways ebenfalls unabdingbar. Es empfehlen sich stromsparende Komponenten, da das System durchgängig laufen muss. Unsere eigene SymBox kann für viele Anwendung ebenfalls hervorragend verwendet werden.

## SOFTWARE

Auf vielen Betriebssystemen kann IP-Symcon als Server eingerichtet werden. Für die mobilen Betriebssysteme (iOS, Android) steht eine speziell angepasste App zur Verfügung, welche die, vom IP-Symcon Server zur Verfügung gestellte, Visualisierung darstellt.
Eine komplette Liste, welche IP-Symcon Version mit welchem Betriebssystem funktioniert, kann in der Versionenübersicht nachgelesen werden.

## LIMITATIONEN VERSCHIEDENER BETRIEBSSYSTEME

### LIMITATIONEN AUF SYMBOX, RASPBERRY PI, LINUX, MACOS UND NAS-SYSTEMEN

Standardmäßig sind die Kompatibilitätsfunktionen deaktiviert. Siehe Spezialschalter zum Reaktivieren. Für Projekte die Kompatibilität zu IP-Symcon 3.4 oder älter benötigen.
Kein MediaPlayer (zur Zeit nur in Windows vorhanden)
Text to Speech: Das TTS-Modul kann momentan nur in der Windows Installation gestartet werden. Alternativ kann das "Text to Speech (AWS Polly)" - Modul aus dem Module Store genutzt werden. Dies funktioniert auf allen Betriebsystemen.
System Informationen: Die Funktion Sys_GetSpooler ist nur unter Windows verfügbar.

# Versionenübersicht

Versionspolitik
Version 8.1
Version 8.0
Version 7.2
Version 7.1
Version 7.0
Version 6.4
Version 6.3
Version 6.2
Version 6.1
Version 6.0
Version 5.5
Version 5.4
Version 5.3
Version 5.2
Version 5.1
Version 5.0
Version 4.4
Version 4.3
Version 4.2
Version 4.1
Version 4.0
Version 3.4

## VERSIONSPOLITIK

| BETRIEBSSYSTEM | GEPLANTE UNTERSTÜTZUNG |
|---|---|
| SymOS | immer |
| Windows Desktop | Alle mit der jeweiligen benötigten PHP Unterstützung |
| Windows Server | Alle mit der jeweiligen benötigten PHP Unterstützung |
| MacOS | Die letzten 2 Versionen zum Zeitpunkt des Entwicklungsbeginns |
| Linux (Ubuntu) | Letzte LTS (Long Term Support) Version zum Zeitpunkt des Entwicklungsbeginns |
| Raspberry Pi | Aktuelles Release zum Zeitpunkt des Entwicklungsbeginns |
| Docker | Docker kompatibler Container |
| QNAP | Docker kompatibler Container |
| Synology | Docker kompatibler Container |

## VERSION 8.1

| BETRIEBSSYSTEM | UNTERSTÜTZT | VERSION | ANMERKUNG |
|---|---|---|---|
| SymOS | ja | 8.1 | Betriebssystem für die SymBox |
| Windows Desktop | ja | ab Windows 8 | nur 64bit Version |
| Windows Server | ja | ab Windows Server 2012 | nur 64bit Version |
| MacOS | ja | ab Version 13 | Intel/Apple M1 |
| Linux (Ubuntu) | ja | Ubuntu 24.04 | nur 64bit Version |
| Raspberry Pi | ja | Raspberry Pi OS (Bookworm oder neuer) | 32bit/64bit (kein Pi 1, Zero 1) |
| Docker | ja | Intel/AMD 64-Bit, ARM7, ARM64 CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Docker |
| QNAP | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter QNAP |
| Synology | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Synology |
| iOS | nur Steuerung | ab iOS 13 | Visualisierung via App möglich |
| Android | nur Steuerung | ab Android 5.0 | Visualisierung via App möglich |

## VERSION 8.0

| BETRIEBSSYSTEM | UNTERSTÜTZT | VERSION | ANMERKUNG |
|---|---|---|---|
| SymOS | ja | 8.0 | Betriebssystem für die SymBox |
| Windows Desktop | ja | ab Windows 8 | nur 64bit Version |
| Windows Server | ja | ab Windows Server 2012 | nur 64bit Version |
| MacOS | ja | ab Version 13 | Intel/Apple M1 |
| Linux (Ubuntu) | ja | Ubuntu 24.04 | nur 64bit Version |
| Raspberry Pi | ja | Raspberry Pi OS (Bullseye oder neuer) | 32bit/64bit (kein Pi 1, Zero 1) |
| Docker | ja | Intel/AMD 64-Bit, ARM7, ARM64 CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Docker |
| QNAP | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter QNAP |
| Synology | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Synology |
| iOS | nur Steuerung | ab iOS 12 | Visualisierung via App möglich |
| Android | nur Steuerung | ab Android 5.0 | Visualisierung via App möglich |

## VERSION 7.2

| BETRIEBSSYSTEM | UNTERSTÜTZT | VERSION | ANMERKUNG |
|---|---|---|---|
| SymOS | ja | 7.2 | Betriebssystem für die SymBox |
| Windows Desktop | ja | ab Windows 7 | nur 64bit Version |
| Windows Server | ja | ab Windows Server 2008 R2 | nur 64bit Version |
| MacOS | ja | ab Version 13 | Intel/Apple M1 |
| Linux (Ubuntu) | ja | Ubuntu 24.04 | nur 64bit Version |
| Raspberry Pi | ja | Raspberry Pi OS (Bullseye oder neuer) | 32bit/64bit (kein Pi 1, Zero 1) |
| Docker | ja | Intel/AMD 64-Bit, ARM7, ARM64 CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Docker |
| QNAP | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter QNAP |
| Synology | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Synology |
| iOS | nur Steuerung | ab iOS 11 | Visualisierung via App möglich |
| Android | nur Steuerung | ab Android 5.0 | Visualisierung via App möglich |

## VERSION 7.1

| BETRIEBSSYSTEM | UNTERSTÜTZT | VERSION | ANMERKUNG |
|---|---|---|---|
| SymOS | ja | 7.1 | Betriebssystem für die SymBox |
| Windows Desktop | ja | ab Windows 7 | nur 64bit Version |
| Windows Server | ja | ab Windows Server 2008 R2 | nur 64bit Version |
| MacOS | ja | ab Version 12 | Intel/Apple M1 |
| Linux (Ubuntu) | ja | Ubuntu 22.04 | nur 64bit Version |
| Raspberry Pi | ja | Raspberry Pi OS (Bullseye oder neuer) | 32bit/64bit |
| Docker | ja | Intel/AMD 64-Bit, ARM6, ARM7, ARM64 CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Docker |
| QNAP | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter QNAP |
| Synology | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Synology |
| iOS | nur Steuerung | ab iOS 11 | Visualisierung via App möglich |
| Android | nur Steuerung | ab Android 5.0 | Visualisierung via App möglich |

## VERSION 7.0

| BETRIEBSSYSTEM | UNTERSTÜTZT | VERSION | ANMERKUNG |
|---|---|---|---|
| SymOS | ja | 7.0 | Betriebssystem für die SymBox |
| Windows Desktop | ja | ab Windows 7 | nur 64bit Version |
| Windows Server | ja | ab Windows Server 2008 R2 | nur 64bit Version |
| MacOS | ja | ab Version 12 | Intel/Apple M1 |
| Linux (Ubuntu) | ja | Ubuntu 22.04 | nur 64bit Version |
| Raspberry Pi | ja | Raspberry Pi OS (Bullseye oder neuer) | 32bit/64bit |
| Docker | ja | Intel/AMD 64-Bit, ARM6, ARM7, ARM64 CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Docker |
| QNAP | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter QNAP |
| Synology | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Synology |
| iOS | nur Steuerung | ab iOS 11 | Visualisierung via App möglich |
| Android | nur Steuerung | ab Android 5.0 | Visualisierung via App möglich |

## VERSION 6.4

| BETRIEBSSYSTEM | UNTERSTÜTZT | VERSION | ANMERKUNG |
|---|---|---|---|
| SymOS | ja | 6.4 | Betriebssystem für die SymBox |
| Windows Desktop | ja | ab Windows 7 | nur 64bit Version |
| Windows Server | ja | ab Windows Server 2008 R2 | nur 64bit Version |
| MacOS | ja | ab Version 10.15 | Intel/Apple M1 |
| Linux (Ubuntu) | ja | Ubuntu 20.04 | nur 64bit Version |
| Raspberry Pi | ja | Raspberry Pi OS (Bullseye oder neuer) | 32bit/64bit |
| Docker | ja | Intel/AMD 64-Bit, ARM6, ARM7, ARM64 CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Docker |
| QNAP | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter QNAP |
| Synology | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Synology |
| iOS | nur Steuerung | ab iOS 9.3 | Visualisierung via App möglich |
| Android | nur Steuerung | ab Android 5.0 | Visualisierung via App möglich |

## VERSION 6.3

| BETRIEBSSYSTEM | UNTERSTÜTZT | VERSION | ANMERKUNG |
|---|---|---|---|
| SymOS | ja | 6.3 | Betriebssystem für die SymBox |
| Windows Desktop | ja | ab Windows 7 | nur 64bit Version |
| Windows Server | ja | ab Windows Server 2008 R2 | nur 64bit Version |
| MacOS | ja | ab Version 10.15 | Intel/Apple M1 |
| Linux (Ubuntu) | ja | Ubuntu 20.04 | nur 64bit Version |
| Raspberry Pi | ja | Raspberry Pi OS (Bullseye oder neuer) | 32bit/64bit |
| Docker | ja | Intel/AMD 64-Bit, ARM6, ARM7, ARM64 CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Docker |
| QNAP | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter QNAP |
| Synology | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Synology |
| iOS | nur Steuerung | ab iOS 9.3 | Visualisierung via App möglich |
| Android | nur Steuerung | ab Android 5.0 | Visualisierung via App möglich |

## VERSION 6.2

| BETRIEBSSYSTEM | UNTERSTÜTZT | VERSION | ANMERKUNG |
|---|---|---|---|
| SymOS | ja | 6.2 | Betriebssystem für die SymBox |
| Windows Desktop | ja | ab Windows 7 | nur 64bit Version |
| Windows Server | ja | ab Windows Server 2008 R2 | nur 64bit Version |
| MacOS | ja | ab Version 10.15 | Intel/Apple M1 |
| Linux (Ubuntu) | ja | Ubuntu 20.04 | nur 64bit Version |
| Raspberry Pi | ja | Raspberry Pi OS (Bullseye oder neuer) | 32bit/64bit |
| Docker | ja | Intel/AMD 64-Bit, ARM6, ARM7, ARM64 CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Docker |
| QNAP | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter QNAP |
| Synology | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Synology |
| iOS | nur Steuerung | ab iOS 9.3 | Visualisierung via App möglich |
| Android | nur Steuerung | ab Android 5.0 | Visualisierung via App möglich |

## VERSION 6.1

| BETRIEBSSYSTEM | UNTERSTÜTZT | VERSION | ANMERKUNG |
|---|---|---|---|
| SymOS | ja | 6.1 | Betriebssystem für die SymBox |
| Windows Desktop | ja | ab Windows 7 | nur 64bit Version |
| Windows Server | ja | ab Windows Server 2008 R2 | nur 64bit Version |
| MacOS | ja | ab Version 10.15 | Intel/Apple M1 |
| Linux (Ubuntu) | ja | Ubuntu 20.04 | nur 64bit Version |
| Raspberry Pi | ja | Raspberry Pi OS (Buster oder neuer) | 32bit/64bit |
| Docker | ja | Intel/AMD 64-Bit, ARM6, ARM7, ARM64 CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Docker |
| QNAP | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter QNAP |
| Synology | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Synology |
| iOS | nur Steuerung | ab iOS 9.3 | Visualisierung via App möglich |
| Android | nur Steuerung | ab Android 5.0 | Visualisierung via App möglich |

## VERSION 6.0

| BETRIEBSSYSTEM | UNTERSTÜTZT | VERSION | ANMERKUNG |
|---|---|---|---|
| SymOS | ja | 6.0 | Betriebssystem für die SymBox |
| Windows Desktop | ja | ab Windows 7 | nur 64bit Version |
| Windows Server | ja | ab Windows Server 2008 R2 | nur 64bit Version |
| MacOS | ja | ab Version 10.15 | Intel/Apple M1 |
| Linux (Ubuntu) | ja | Ubuntu 20.04 | nur 64bit Version |
| Raspberry Pi | ja | Raspberry Pi OS (Buster oder neuer) | 32bit/64bit |
| Docker | ja | Intel/AMD 64-Bit, ARM6, ARM7, ARM64 CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Docker |
| QNAP | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter QNAP |
| Synology | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Synology |
| iOS | nur Steuerung | ab iOS 9.3 | Visualisierung via App möglich |
| Android | nur Steuerung | ab Android 5.0 | Visualisierung via App möglich |

## VERSION 5.5

| BETRIEBSSYSTEM | UNTERSTÜTZT | VERSION | ANMERKUNG |
|---|---|---|---|
| SymOS | ja | 5.5 | Betriebssystem für die SymBox |
| Windows Desktop | ja | ab Windows 7 | nur 64bit Version |
| Windows Server | ja | ab Windows Server 2008 R2 | nur 64bit Version |
| MacOS | ja | ab Version 10.14 | |
| Linux (Ubuntu) | ja | Ubuntu 20.04 | nur 64bit Version |
| Raspberry Pi | ja | Raspbian Buster | |
| Docker | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Docker |
| QNAP | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter QNAP |
| Synology | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Synology |
| iOS | nur Steuerung | ab iOS 9.3 | Visualisierung via App möglich |
| Android | nur Steuerung | ab Android 5.0 | Visualisierung via App möglich |

## VERSION 5.4

| BETRIEBSSYSTEM | UNTERSTÜTZT | VERSION | ANMERKUNG |
|---|---|---|---|
| SymOS | ja | 1.8 | Betriebssystem für die SymBox |
| Windows Desktop | ja | ab Windows 7 | nur 64bit Version |
| Windows Server | ja | ab Windows Server 2008 R2 | nur 64bit Version |
| MacOS | ja | ab Version 10.13 | |
| Linux (Ubuntu) | ja | Ubuntu 18.04 | nur 64bit Version |
| Raspberry Pi | ja | Raspbian Buster | |
| Docker | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Docker |
| QNAP | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter QNAP |
| Synology | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Synology |
| iOS | nur Steuerung | ab iOS 9.3 | Visualisierung via App möglich |
| Android | nur Steuerung | ab Android 5.0 | Visualisierung via App möglich |

## VERSION 5.3

| BETRIEBSSYSTEM | UNTERSTÜTZT | VERSION | ANMERKUNG |
|---|---|---|---|
| SymOS | ja | 1.7 | Betriebssystem für die SymBox |
| Windows Desktop | ja | ab Windows 7 | nur 64bit Version |
| Windows Server | ja | ab Windows Server 2008 R2 | nur 64bit Version |
| MacOS | ja | ab Version 10.13 | |
| Linux (Ubuntu) | ja | Ubuntu 18.04 | nur 64bit Version |
| Raspberry Pi | ja | Raspbian Buster | |
| Docker | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Docker |
| QNAP | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter QNAP |
| Synology | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Synology |
| iOS | nur Steuerung | ab iOS 9.3 | Visualisierung via App möglich |
| Android | nur Steuerung | ab Android 5.0 | Visualisierung via App möglich |

## VERSION 5.2

| BETRIEBSSYSTEM | UNTERSTÜTZT | VERSION | ANMERKUNG |
|---|---|---|---|
| SymOS | ja | 1.6 | Betriebssystem für die SymBox |
| Windows Desktop | ja | ab Windows 7 | nur 64bit Version |
| Windows Server | ja | ab Windows Server 2008 R2 | nur 64bit Version |
| MacOS | ja | ab Version 10.12 | |
| Linux (Ubuntu) | ja | Ubuntu 18.04 | nur 64bit Version |
| Raspberry Pi | ja | Raspbian Buster | |
| Docker | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Docker |
| QNAP | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter QNAP |
| Synology | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Synology |
| iOS | nur Steuerung | ab iOS 9.3 | Visualisierung via App möglich |
| Android | nur Steuerung | ab Android 5.0 | Visualisierung via App möglich |

## VERSION 5.1

| BETRIEBSSYSTEM | UNTERSTÜTZT | VERSION | ANMERKUNG |
|---|---|---|---|
| SymOS | ja | 1.5 | Betriebssystem für die SymBox |
| Windows Desktop | ja | ab Windows 7 | nur 64bit Version |
| Windows Server | ja | ab Windows Server 2008 R2 | nur 64bit Version |
| MacOS | ja | ab Version 10.11 | |
| Linux (Ubuntu) | ja | Ubuntu 18.04 | nur 64bit Version |
| Raspberry Pi | ja | Raspbian Stretch | |
| Docker | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Docker |
| QNAP | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter QNAP |
| Synology | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Synology |
| iOS | nur Steuerung | ab iOS 9.3 | Visualisierung via App möglich |
| Android | nur Steuerung | ab Android 5.0 | Visualisierung via App möglich |

## VERSION 5.0

| BETRIEBSSYSTEM | UNTERSTÜTZT | VERSION | ANMERKUNG |
|---|---|---|---|
| SymOS | ja | 1.4 | Betriebssystem für die SymBox |
| Windows Desktop | ja | ab Windows 7 | nur 64bit Version |
| Windows Server | ja | ab Windows Server 2008 R2 | nur 64bit Version |
| MacOS | ja | ab Version 10.11 | |
| Linux (Ubuntu) | ja | Ubuntu 18.04 | nur 64bit Version |
| Raspberry Pi | ja | Raspbian Stretch | kein Raspbian Jessie |
| Docker | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Docker |
| QNAP | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter QNAP |
| Synology | ja | Intel/AMD 64-Bit CPU | bestimmte Prozessorarchitektur nötig; weiteres unter Synology |
| iOS | nur Steuerung | ab iOS 9.3 | Visualisierung via App möglich |
| Android | nur Steuerung | ab Android 5.0 | Visualisierung via App möglich |

## VERSION 4.4

| BETRIEBSSYSTEM | UNTERSTÜTZT | VERSION | ANMERKUNG |
|---|---|---|---|
| SymOS | ja | 1.3 | Betriebssystem für die SymBox |
| Windows Desktop | ja | ab Windows 7 | |
| Windows Server | ja | ab Windows Server 2008 R2 | |
| MacOS | ja | ab Version 10.11 | |
| Linux (Ubuntu) | ja | Ubuntu 16.04 | nur 64bit Version |
| Raspberry Pi | ja | Raspbian Stretch | kein Raspbian Jessie |
| iOS | nur Steuerung | | Visualisierung via App möglich |
| Android | nur Steuerung | | Visualisierung via App möglich |

## VERSION 4.3

| BETRIEBSSYSTEM | UNTERSTÜTZT | VERSION | ANMERKUNG |
|---|---|---|---|
| SymOS | ja | 1.2 | Betriebssystem für die SymBox |
| Windows Desktop | ja | ab Windows 7 | |
| Windows Server | ja | ab Windows Server 2008 R2 | |
| MacOS | ja | ab Version 10.10 | |
| Linux (Ubuntu) | ja | Ubuntu 16.04 | nur 64bit Version |
| Raspberry Pi | ja | Raspbian Jessie | kein Raspbian Stretch |
| iOS | nur Steuerung | | Visualisierung via App möglich |
| Android | nur Steuerung | | Visualisierung via App möglich |

## VERSION 4.2

| BETRIEBSSYSTEM | UNTERSTÜTZT | VERSION | ANMERKUNG |
|---|---|---|---|
| SymOS | ja | 1.1 | Betriebssystem für die SymBox |
| Windows Desktop | ja | ab Windows Vista | |
| Windows Server | ja | ab Windows Server 2008 | |
| MacOS | ja | ab Version 10.9 | |
| Linux (Ubuntu) | ja | Ubuntu 16.04 | nur 64bit Version |
| Raspberry Pi | ja | Raspbian Jessie | kein Raspbian Stretch |
| iOS | nur Steuerung | | Visualisierung via App möglich |
| Android | nur Steuerung | | Visualisierung via App möglich |

## VERSION 4.1

| BETRIEBSSYSTEM | UNTERSTÜTZT | VERSION | ANMERKUNG |
|---|---|---|---|
| SymOS | ja | 1.1 | Betriebssystem für die SymBox |
| Windows Desktop | ja | ab Windows Vista | |
| Windows Server | ja | ab Windows Server 2008 | |
| MacOS | ja | ab Version 10.9 | |
| Linux (Ubuntu) | ja | Ubuntu 16.04 | nur 64bit Version |
| Raspberry Pi | ja | Raspbian Jessie | kein Raspbian Wheezy |
| iOS | nur Steuerung | | Visualisierung via App möglich |
| Android | nur Steuerung | | Visualisierung via App möglich |

## VERSION 4.0

| BETRIEBSSYSTEM | UNTERSTÜTZT | VERSION | ANMERKUNG |
|---|---|---|---|
| SymOS | ja | 1.0 | Betriebssystem für die SymBox |
| Windows Desktop | ja | ab Windows Vista | |
| Windows Server | ja | ab Windows Server 2008 | |
| MacOS | ja | ab Version 10.8 | |
| Linux (Ubuntu) | ja | Ubuntu 14.04 | nur 64bit Version |
| Raspberry Pi | ja | Raspbian Wheezy | kein Raspbian Jessie |
| iOS | nur Steuerung | | Visualisierung via App möglich |
| Android | nur Steuerung | | Visualisierung via App möglich |

## VERSION 3.4

| BETRIEBSSYSTEM | UNTERSTÜTZT | VERSION | ANMERKUNG |
|---|---|---|---|
| Windows Desktop | ja | ab Windows 2000 | |
| Windows Server | ja | ab Windows Server 2003 | |
| iOS | nur Steuerung | | Visualisierung via App möglich |
| Android | nur Steuerung | | Visualisierung via App möglich |

# Vertragsbestimmungen



Unsere "Allgemeinen Vertragsbestimmungen des Software-Kaufvertrags" kann jederzeit herunterladen werden: Download



Eine ausführliche Erklärung der Nutzungsrechte befindet sich hier: Nutzungsrechte.

# Nutzungsrechte

Die rechtsverbindlichen, allgemeinen Vertragsbestimmungen des Software-Kaufvertrags befinden sich zum Download unter Vertragsbestimmungen

Eine IP-Symcon Lizenz berechtigt, IP-Symcon beliebig oft in einem Privathaushalt zu installieren. Für gewerbliche Installationen wird für jeden im Betrieb befindlichen Server/Rechner eine separate IP-Symcon Lizenz benötigt. Es kann zu jeder Zeit nur eine produktive Installation pro Lizenz in Betrieb sein, die mit dem Server für Push-Benachrichtigungen oder dem Connect-Dienst kommuniziert. Jegliche weitere Installationen ist somit nur für Programmier- und Testzwecke verwendbar. Eine IP-Symcon Lizenz gilt für alle Endgeräte (Betriebssysteme), auf denen IP-Symcon verfügbar ist. Für eine Online-Installation wird eine gültige Subskription benötigt. Falls eine Installation von IP-Symcon nach Ablauf der Subskription gewollt ist, wird ein selbsterstelltes Backup benötigt.

Die IP-Symcon Pro Konsole kann ohne IP-Symcon lizenzfrei vom IP-Symcon Server heruntergeladen und automatisch durch die integrierte Updatefunktion aktualisiert werden.

# Subskription

Eine Subskription stellt in seiner Laufzeit die aufgeführten Dienste zur Verfügung. Diese werden von manchen Modulen zum Betrieb benötigt (z.B. Connect Dienst für Alexa Smart Home Skill)

## DIENSTE

Folgende Dienste brauchen eine aktive Subskription.

### INSTALLATION/UPDATE

Bei Neuinstallation und Updates von IP-Symcon wird eine aktive Subskription benötigt.

### PUSH NOTIFICATION

Über unseren Server werden Push Notifications auf die mobilen Apps gesendet.

### CONNECT DIENST

Es wird ein verschlüsselter Zugriff via unseren Connect-Dienst Server zur Verfügung gestellt, um von überall sicher auf das System zugreifen zu können.

## LAUFZEIT

Die Laufzeit (Gültigkeit) der Subskription kann über eines der drei folgenden Punkte festgestellt werden:

in der letzten Mail einer Lizenz bzw. Subskription Bestellung: "... ist gültig bis: TT.MM.JJ"
in unserem Forum unter "Lizenzverwaltung". Hierzu muss man sich dort registrieren.
per Mail oder über das Kontaktformular anfragen
ein Monat vor Ablauf der Subskription wird eine Mail mit dem Hinweis des baldigen Ablaufens verschickt



Bei einer Subskription handelt es sich **nicht** um ein Abonnement.
Bei Ablauf der Subskription kann diese im IP-Symcon Shop neu erworben/verlängert werden.
Einen Monat vor Ablauf wird automatisch eine Hinweis-Mail von unserem System versendet.
Bei einer Bestellung wird das Zeitkontingent einer eventuellen Restlaufzeit hinzuaddiert (ähnlich einer Prepaid-Karte).

Weitere Informationen über die verwendete Lizenz gibt es in der Verwaltungskonsole über das Informationsicon oben rechts.

# Schnelleinstieg

## IP-SYMCON

IP-Symcon ist eine Serversoftware, welche es als Ziel hat verschiedene Automationssysteme in einem Server zu vereinen, verknüpfen und automatisieren. Dies bedeutet, dass Systeme, welche nicht miteinander kommunizieren können, in eine einheitliche übersichtliche Konfiguration zusammgeführt werden. Auch bietet IP-Symcon eine einheitliche Visualisierung und Steuerung, sowohl für den Browser als auch via Mobile-Apps für iOS und Android. Eine umfassende Beschreibung kann auf der Produktseite nachgelesen werden.

## UNTERSTÜTZTE SYSTEME

IP-Symcon unterstützt eine große Vielfalt an namhaften Herstellern und Protokollen. Eine Übersicht der nativen Systeme kann in der Modulreferenz nachgeschlagen werden. Zusätzlich wird der Module Store angeboten. Dieser ist eine Platform bei dem Module von der Community eingereicht, qualitativ kontrolliert und kostenfrei dem Nutzer zur Verfügung gestellt werden.

## IP-SYMCON EDITIONEN

Es gibt drei verschiedene Lizenzen: Basic, Professional und Unlimited. Diese sind zeitlich unbegrenzt gültig und bieten die gleichen Systemunterstützungen. Die Editionen unterscheiden sich einzig bei Anzahl der Variablen, WebFronts, RTSP-Streams und ab der Professional-Lizenz in der Möglichkeit die Visualisierung WebFront mit einem erweiterten Editor individuell anzupassen. Eine genaue Auflistung aller Funktionen befindet sich in der Funktionsübersicht . Es kann jederzeit eine bestehende Lizenz zu einer höheren Version upgegradet werden. Eine Übersicht der Upgrades kann im Shop eingesehen werden.

### SUBSKRIPTION

Mit der Subskription werden Features, welche unser Server bietet, aktiviert. Darunter fallen z.B. Push Notification, Updates, Connect Control (insbesondere wichtig für Amazon Alexa und Google Assistant ). Eine ausgelaufene Subskription beinträchtigt nicht den laufenden Server, es werden einzig die mit der Subskription verbundene Features deaktiviert. Eine Subskription kann jederzeit über den Shop verlängert werden.

## TESTVERSION

Es ist möglich eine Demo anzufordern . Diese Demo ist ein Monat gültig und bietet den gleichen Funktionsumfang wie eine Professional Lizenz.

## HARDWAREVORAUSSETZUNGEN

Der IP-Symcon Server muss dauerhaft laufen. Dafür ist ein SymBox , Windows, MacOS, Linux Ubuntu, Raspberry Pi oder Docker (auch QNAP, Synology) System nötig. Welche

Version des jeweiligen Betriebssystems nötig ist, kann unter der Versionenübersicht im Bereich Systemvorraussetzungen nachgeschlagen werden. Des Weiteren werden für die verbauten Systeme sogenannte Gateways als Schnittstelle benötigt. Welche Gateways unterstützt werden, kann in der jeweiligen Modulreferenz und deren Geräteliste nachgeschlagen werden. Einige Gateways werden im Shop oder sogar als SymBox-Erweiterung angeboten.

### SYMBOX FUNKTIONALITÄTEN

Die SymBox ist der hauseigene Server. Dieser kann in einem Hutschienengehäuse für den Schaltschrank oder in einem Aluminiumgehäuse gekauft werden. Neben dem sehr geringen Stromverbrauch (max. 3W) bietet die SymBox mit dem SymOS ein speziell angepasstes Betriebssystem und zugeschnittene Weboberfläche für IP-Symcon. Neben Updates und Einstellungen kann auf diesem bequem die Datensicherung verwaltet werden. Sollte eine Symbox im Einsatz sein, kann das SymOS im lokalen Netzwerk über "symbox.local " erreicht werden.

## INSTALLATION VON IP-SYMCON

Die Installation ist für das jeweilige System unter Installation beschrieben. Nach einer erfolgreichen Installation startet der IP-Symcon Dienst automatisch und ist bereit zur Konfiguration.

## TRAY FUNKTIONALITÄTEN

Nach der Installation steht unter Windows und MacOS nun die Tray-Anwendung zur Verfügung. Die Tray-Anwendung befindet sich abhängig vom Betriebssystem als IP-Symcon Symbol im jeweiligen System-Tray. Über die Anwendung kann der Dienst gestartet, gestoppt und aktualisiert werden. Unter "Information" befindet sich die Möglichkeit einen Fernzugriff einzurichten.

## KONFIGURATION

Zur Konfiguration wird Verwaltungskonsole genutzt. Diese ist von überall aus dem Netzwerk via Browser aufrufbar. Um diese zu öffnen muss lediglich "[IPServer]:3777/console/" in die Adresszeile eingetragen werden. Sollte eine SymBox im Einsatz sein ist dies "symbox.local:3777/console/ ".

## GRUNDLAGEN

Im Bereich Grundlagen werden die wesentlichen Grundbegriffe und deren Handhabung beschrieben. Der nächste Schritt sind die Komponenten, welche Funktionalitäten zur Steuerung, Automatisierung und Visualisierung erklären.

## VISUALISIERUNG UND STEUERUNG

Die Visualisierung WebFront der eingerichteten Instanzen wird automatisch erstellt und kann den persönlichen Bedürfnissen weiter angepasst werden. Eine Steuerung findet

über die Visualisierung statt. Eine weitere Möglichkeit sind die kostenlosen mobilen Apps für iOS und Android.
Unter Visualisierungen sind alle nötigen Informationen zu finden.

## SICHERHEIT

Um den wichtigen Aspekt Sicherheit in der Hausautomation abzudecken ist in den Werkseinstellungen ein Zugriff von außen nicht möglich. Weitere Informationen zu allen Sicherheitsaspekten können im Abschnitt Sicherheit nachgeschlagen werden. Zusätzlich prüft ein Sicherheits-Widget die Konfiguration und weist mit einer Übersicht auf gegebenfalls zu setzende Kennwörter hin.

## DATENSICHERUNG

Zur Datensicherung kann ein Backup erstellt werden. Dieses kann dann auf einem anderen oder neuen System installiert werden. Genaue Schritte dafür können unter Backup einspielen gefunden werden. Ein solches Backup ist auch bei einem Plattformwechsel vollständig kompatibel.

## CONNECT CONTROL

Das Connect Control dient zur verschlüsselten Kommunikation nach Außen. Es ermöglicht die gesicherte Steuerung und Visualisierung außerhalb ihres lokalen Netzwerks. Bestimmte Funktionen und Module (z.B. Amazon Alexa, Push Notifications) benötigen eine dauerhafte Internetverbindung. Diese wird ebenfalls durch das verschlüsselte Connect Control gewährleistet.

## FERNZUGRIFF

Der Fernzugriff bietet die Möglichkeit IP-Symcon von außen zu konfigurieren. Erst bei Vergabe eines Passworts wird der Zugang nach außen über das lokales Netzwerk hinweg aktiviert. So ist es möglich eine vollumfängliche Fernwartung durchzuführen. Dazu wird das Connect Control benötigt.

## WEITERE INFORMATIONEN UND VIDEOS

Fortführende Informationen und Beispiele befinden sich auf dem Symcon YouTube Kanal und unter Vorgehensweisen. Neben der Dokumentation bietet auch das Forum eine große Menge an Informationen, Skriptsammlungen und Ideen, sowie eine breite Basis von Profis und Neueinsteigern.

# Video-Tutorials

Für den schnellen Einstieg in IP-Symcon sind hier einige nützliche Video-Anleitungen:



Alle Videos befinden sich auf unserem YouTube-Kanal.

- Unsere SymBox
  - Teil 1 - Zahlen, Daten, Fakten
  - Teil 2: Inbetriebnahme + Recovery Tool
  - Teil 3: KNX Demo & Pro Konsole
  - Teil 4: Tailscale VPN (Modul)
- Grundlagen
  - Teil 1: Grundlagen
  - Teil 2: Installationen auf Windows & Raspberry Pi
  - Teil 3: Symcon mit Docker
  - Teil 4: macOS & Synology Installation
  - Teil 5: Update mit Portainer
  - Teil 6: Geräte einbinden & konfigurieren
  - Teil 7: Benutzeroberfläche & Navigation
  - Teil 8: Variablendarstellung
  - Teil 9: Location Control Modul
- Logikpläne
  - Teil 1: Einführung und Grundlagen
  - Teil 2: Sublogikpläne
  - Teil 3: Ausschaltverzögerung - Sublogikplan
  - Teil 4: Watchdog - Sublogikplan
  - Teil 5: Türklingel-Trigger - Sublogikplan
  - Teil 6: min/max Werte Loggen - Sublogikplan
- Visualisierung
  - WebFront Funktionen und der WYSIWYG-Editor
  - Symcon Connect- Die sichere Verbindung zu IP-Symcon und Webfront
  - IP-Symcon Webinar: Preview zur Visualisierung
  - IP-Symcon Webinar: Öffentliche Beta der neuen Visualisierung
- Entwickler-Webinare
  - Einstieg für Entwickler
  - Dynamische Module
  - Automatisch Testen
  - Konfigurationsformulare und deren Elemente
  - Konfigurator Element und Datenfluss
  - Anktionen erstellen
  - Git Basics
  - Änderungen am PHP-SDK zur 7.0
  - IP-Symcon Entwickler Webinar: HTML-SDK
  - Darstellung in Modulen
- Grundlagen - Webinare
  - IP-Symcon Grundlagen Webinar
  - IP-Smcon Grundlagen Webinar - Teil 2
  - IP-Symcon Entwickler Webinar: Module automatisiert testen

IP-Symcon Webinar Grundlagen Visualisierung
IP-Symcon Webinar: Tabs & Widgets
Releases
Version 8.0 Preview
Version 7.2
Version 7.1
Version 7.0
Version 6.4
IPSView & IPSWorkflows
Energie Management mit IP-Symcon
Version 6.3
Version 6.2
Version 6.1
Version 6.0
Version 5.5
Version 5.4



Event verpasst? Schau sie gerne nach!

System einrichten
KNX Geräte einrichten inkl. Rückmeldeadressen
SymBox Pro mit KNX Data Secure einrichten
KNX: IP-Secure einrichten
xComfort
dS-Einrichten
Simens OZW einrichten
XML-Export mithilfe der ETS6
In IP-Symcon ein BACnet Gerät einrichten
In IP-Symcon einen MQTT-Client einrichten
In IP-Symcon einen MQTT-Server einrichten
In IP-Symcon ein KNX System mit Konfigurator einrichten
XML-Export mithilfe der ETS
In IP-Symcon ein Z-Wave System einrichten
In IP-Symcon ein LCN System einrichten
Symbox - Erste Einrichtung der IP-Symcon Zentrale
OPC- EXport mithilfe der ETS(ab v5.6.5)
In IP-Symcon eine Alexa Steuerung in unter 10 Minuten einrichten
In IP-Symcon ein HomeMatic System einrichten
HomeMatic Firewalleinstellungen für einen funktionalen Rückkanal
In IP-Symcon ein 1-wire System einrichten
IP-Adresse in der ETS herausfinden
In IP-Symcon ein KNX System einrichten
OPC-Export mithilfe der ETS(bis v 5.6.4)

# Installation

Installation für Windows
Installation für MacOS
Installation für Linux
Installation für Raspberry Pi
Installation für Docker
Installation für QNAP
Installation für Synology
Installation für SymBox

## INSTALLATION

Das erstmalige Installieren und Inbetriebnehmen auf den verschiedenen Plattformen.
Dazu nutzt IP-Symcon einen auf das jeweilige System bzw. Plattform angepassten Installationsweg.
Ausführliche Informationen kann unter dem jeweiligen System nachgelesen werden.



Um IP-Symcon den Zugriff auf bestimmte Funktionen (z.B. Updates) außerhalb zu gewähren ist es gegebenenfalls nötig diese in der Firewall einzurichten. Weitere Informationen können unter Firewall gefunden werden.

## UPDATE

IP-Symcon kann bei gültiger Subskription aktualisiert werden.
Wie ein Update initiert wird kann unter dem jeweiligen System nachgelesen werden.
Für ein Update muss der IP-Symcon Server kontaktiert werden, um die neusten Pakete anzufordern. Sollte dies fehlschlagen, so gibt es mehrere mögliche Ursachen:

Die Subskription ist abgelaufen. Es sollte eine neue im Shop erworben werden.
Die Proxyeinstellungen sind falsch.
Die Internet-Verbindung ist nicht aufgebaut.

## DATENSICHERUNG/BACKUP

Um Datenverlust zu vermeiden gibt es verschiedene Wege IP-Symcon-Daten zu sichern.
Diese sind unter Datensicherung beschrieben.

## PLATTFORMWECHSEL

IP-Symcon kann von einer Plattform auf eine Andere umziehen ohne eine Neueinrichtung vorzunehmen.
Das Verfahren ist unter Plattformwechsel erklärt.

## DEINSTALLATION

Wie IP-Symcon deinstalliert wird kann unter dem jeweiligen System nachgelesen werden.

## LIZENZ ÄNDERN

Sollte sich die Lizenz ändern, kann diese innerhalb von IP-Symcon über die Verwaltungskonsole ausgetauscht werden.
Die nötigen Schritte können unter Lizenz ändern gefunden werden.

## MIGRATION

Eine Migration von einer alten Version auf eine neue Version ist ohne Datenverlust möglich.
Im linksseitigen Menü können alle bisherigen Migrationen gefunden werden.
Es sollten von Version zu Version die einzelnen Migrationen durchgeführt werden.

# Windows

Die Installation ist nur mit einer gültigen Subskription möglich. Dennoch kann ohne gültige Subskription eine vorhandene Datensicherung der Software wieder eingespielt werden. Siehe: Plattformwechsel

Es wird eine regelmäßige Datensicherungen empfohlen, damit bei einem technischen Defekt das System auf dem möglichst aktuellen Stand bleibt. Zumal eine aktive Subskription gebraucht wird, um bei vollständigen Verlust eine IP-Symcon Version erneut downloaden zu können.

## INHALT


Installation
Update
Deinstallation


## INSTALLATION

### VERSIONSVORAUSSETZUNG

Welche Version von IP-Symcon mit welcher Windowsversion funktioniert, kann der Versionenübersicht entnommen werden.

### INSTALLATION

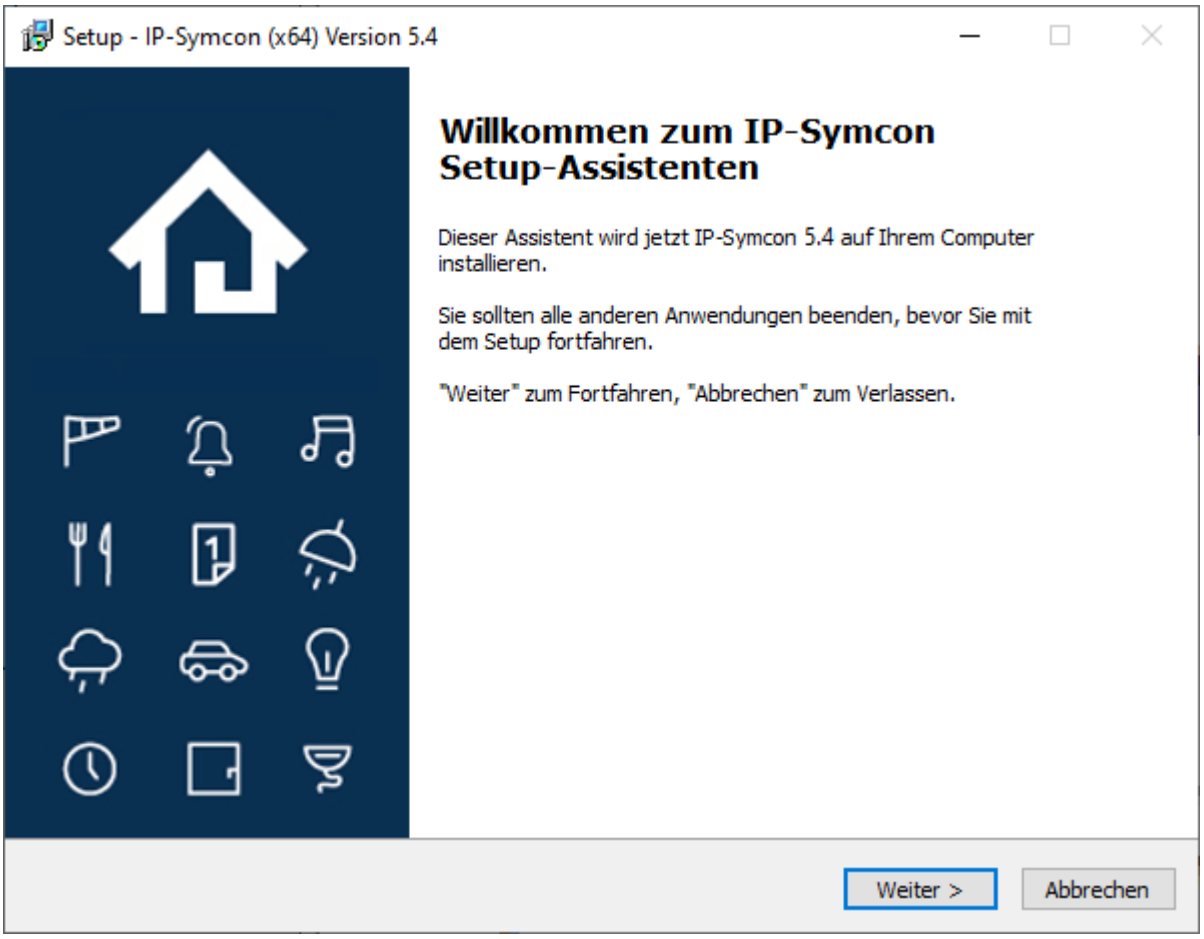
**Download:** Setupdatei (64-Bit)
IP-Symcon ist eines der wenigen Programme, das sich sehr sauber installieren und auch wieder rückstandsfrei deinstallieren lässt, da es keine Unmenge von Einträgen in der Windows-Registry vornimmt. Letztendlich wird nur der Lizenzkey gespeichert.

### HINWEISE

Um IP-Symcon installieren zu können, wird eine Internetverbindung benötigt. Falls auf dem Ziel-Rechner keine Verbindung vorhanden sein sollte, muss IP-Symcon auf einem anderen Rechner installiert und hinterher auf den Ziel-Rechner kopiert werden.

### SETUP

Nach dem Start des Setups und Auswahl der Sprache sollte folgendes Fenster erscheinen:



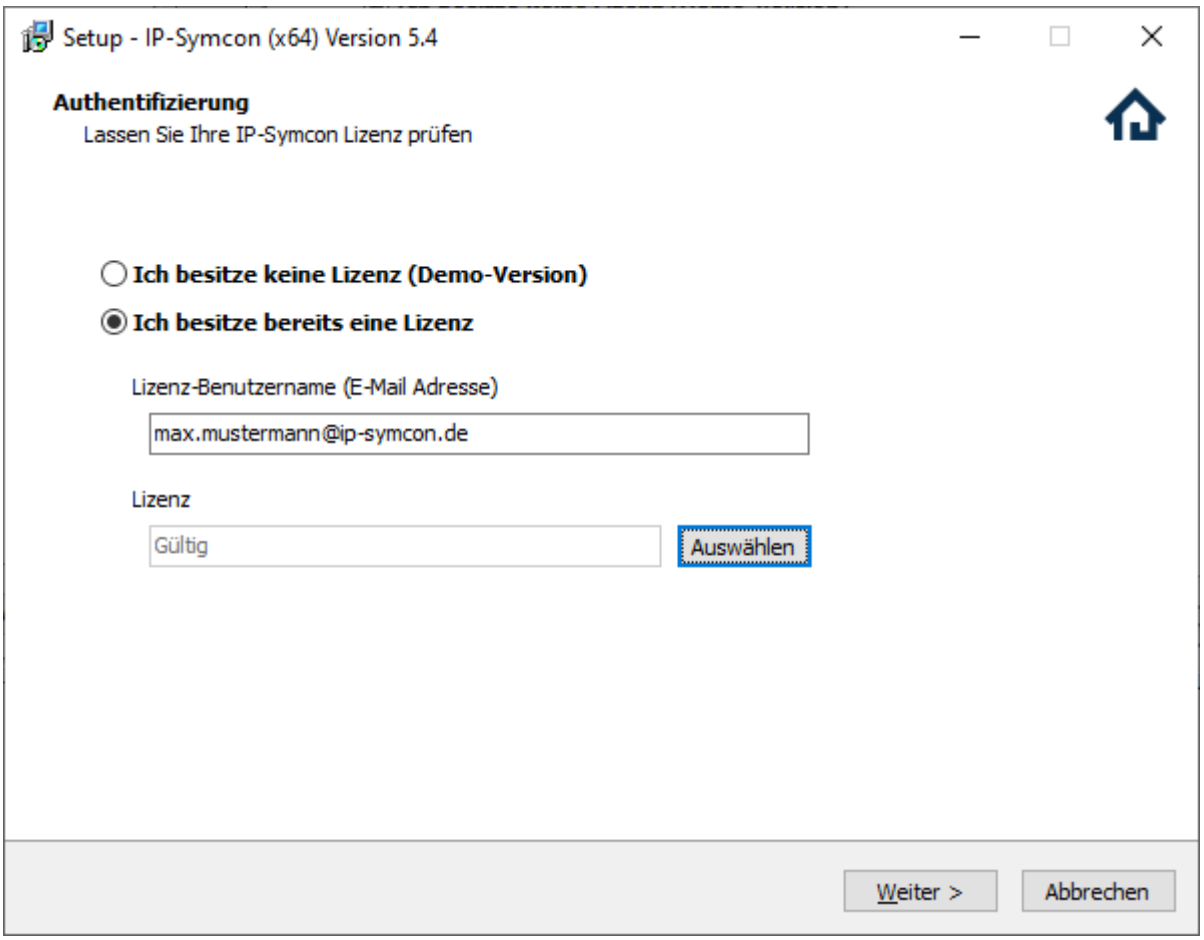

Zum nächsten Schritt kann der Dialog mit "Weiter" bestätigt werden.

## LIZENZDATEN ANGEBEN

Es kann eine Demo Version oder, durch Angabe des Benutzernamens und der Lizenzdatei, eine vollwertige Lizenz genutzt werden. Anschließend kann der Dialog mit "Weiter" beendet werden.



Der Benutzername entspricht normalerweise der E-Mail-Adresse, die beim Bestellvorgang angegeben wurde.




Die Subskription wird nun überprüft.
Nur mit einer gültigen Subskription oder als Demo-Edition kann IP-Symcon installiert werden.
Falls die Subskription abgelaufen ist, kann eine Verlängerung im Shop bestellt werden: Shop
Falls keine Verbindung zum Server hergestellt werden kann, ist eine Kontrolle der Internetverbindung oder der Proxyeinstellungen nötig.
Nach dem Akzeptieren der Lizenzbedingungen muss ein geeigneter Pfad ausgewählt (z.B. C:\IP-Symcon) und mit "Weiter" bestätigt werden.

Bei der Installation wird automatisch der Ordner C:\ProgramData\Symcon\ angelegt. In diesem befinden sich alle Serverdaten, also die individuellen Daten der lokalen Installation. Wenn der Ordner für die Serverdaten verlegt werden soll ist dies ab Version 6.0 möglich. Serverdaten verlegen

Mit dem Button "Installieren" beginnt das Kopieren der Daten.
Nach einem Klick auf OK ist die Installation auf dem System fertiggestellt. Abhängig davon, welche der Optionen aktiviert sind, wird der IP-Symcon Dienst und die IP-Symcon Tray-Anwendung automatisch gestartet.
Das IP-Symcon WebFront sollte nun auf http://ipadresse:3777/ erreichbar sein.
Die IP-Symcon webbasierte Verwaltungskonsole kann über http://ipadresse:3777/console/ aufgerufen werden. Alternativ kann die Pro Konsole verwendet werden.

## WO FINDE ICH WAS?

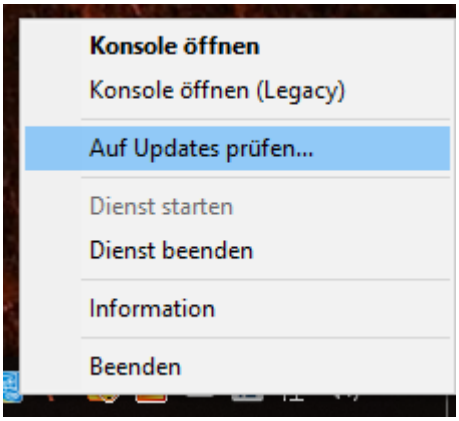
C:\ProgramData\Symcon\ (Settings, Skripte, Medien...)
C:\ProgramData\Symcon\logs\ (Logfiles...)

### FERNZUGRIFF KENNWORT SETZEN

Über Fernzugriff kann nachgeschlagen werden, wie der Fernzugriff eingerichtet wird.

## UPDATE

Über das Tray Icon kann via "Rechtslick -> Auf Updates prüfen -> Herunterladen" die neuste Versionen heruntergeladen und installiert werden. Der Dienst muss während der Aktualisierung einmal beendet werden und wird im Anschluss automatisch wieder neugestartet.

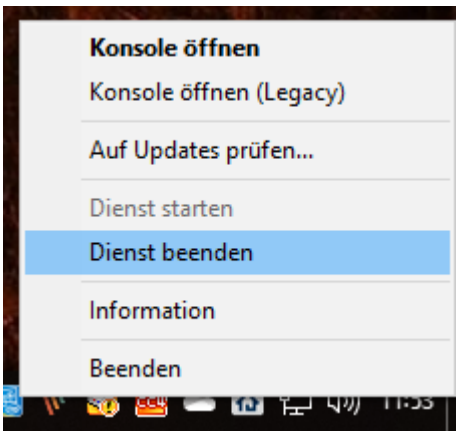



## DEINSTALLATION

Vor der Deinstallation von IP-Symcon, muss der Dienst angehalten werden.

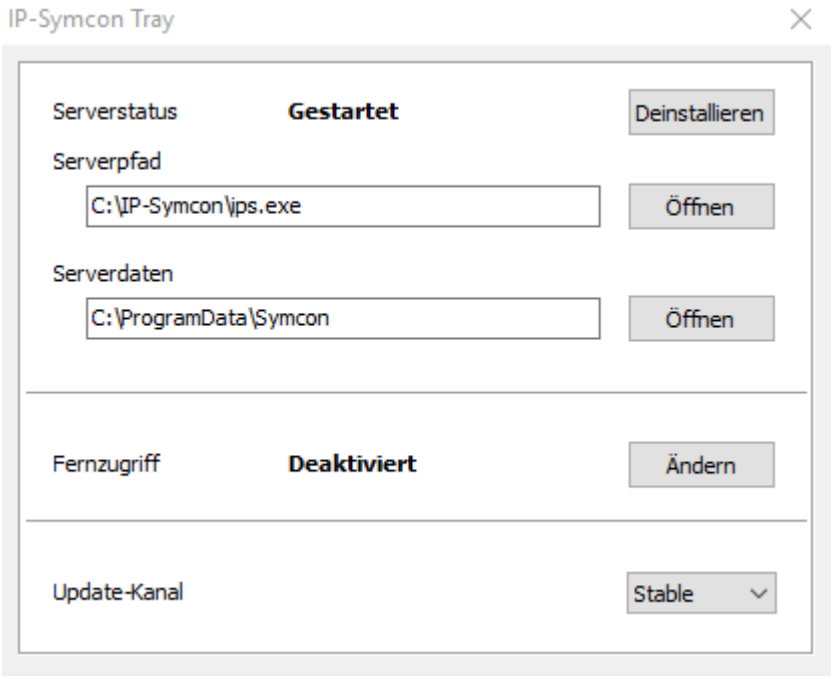
### DIENST STOPPEN

Über das Kontextmenü der Tray-Anwendung kann über den Menüpunkt “Dienst anhalten” der Dienst gestoppt werden.



### DIENST/SERVER DEINSTALLIEREN

Über das Kontextmenü des Trayicons kann der Dialog "Information" geöffnet werden. Folgendes sollte zu sehen sein:




“Deinstallieren“ entfernt den Dienst von dem System.

### IP-SYMCON LÖSCHEN

Um die Deinstallation zu beenden, muss der komplette IP-Symcon Ordner gelöscht oder in den Papierkorb verschoben werden.

## SERVERDATEN VERLEGEN

> Es ist davon abzusehen den Standardordner zu verlegen. In gewissen Konstellationen kann dies aber nützlich sein, z.B. für automatisierte Backups.

Folgende Schritte sind erst ab Version 6.0 möglich.
Den IP-Symcon Dienst beenden
Im Startmenü "regedit" eingeben und den Registrierungs-Editor starten
Zu dem Pfad HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\IP-Symcon gehen
In diesem eine neue Zeichenfolge erstellen und "DataPath" nennen
In die Zeichenfolge "DataPath" den neuen vollständigen Pfad angeben.
Die Daten von C:\ProgramData\Symcon an den neuen Order verschieben
Überprüfen, dass der C:\ProgramData\Symcon Order leer ist oder nicht mehr existiert
Den IP-Symcon Dienst starten
Ab diesem Zeitpunkt startet IP-Symcon aus diesem Pfad.



Extrem wichtig: Der C:\ProgramData\Symcon Pfad muss entweder nicht existieren oder komplett leer sein.

# MacOS

Die Installation ist nur mit einer gültigen Subskription möglich. Dennoch kann ohne gültige Subskription eine vorhandene Datensicherung der Software wieder einspielt werden. Siehe: Plattformwechsel

Es wird eine regelmäßige Datensicherungen empfohlen, damit bei einem technischen Defekt das System auf dem möglichst aktuellen Stand bleibt. Zumal eine aktive Subskription gebraucht wird, um bei vollständigen Verlust eine IP-Symcon Version erneut downloaden zu können.

## INHALT


Installation
Update
Deinstallation


## INSTALLATION

### VERSIONSVORAUSSETZUNG

Welche Version von IP-Symcon mit welchem MacOS funktioniert, kann der Versionenübersicht entnommen werden.

### INSTALLATION

**Download:** Setupdatei
ZIP entpacken und in den Programme Ordner kopieren.
Programm starten. Dienst entsprechend installieren und starten.
Das IP-Symcon WebFront sollte nun auf http://ipadresse:3777/ erreichbar sein.
Die IP-Symcon webbasierte Verwaltungskonsole kann über http://ipadresse:3777/console/ aufgerufen werden. Alternativ kann die Pro Konsole verwendet werden.

### WO FINDE ICH WAS?

/Library/Application Support/Symcon/ (Settings, Skripte, Medien...)
/Library/Logs/Symcon/ (Logfiles...)

### FERNZUGRIFF KENNWORT SETZEN

Über Fernzugriff kann nachgeschlagen werden, wie der Fernzugriff eingerichtet wird.

## UPDATE

Über das Tray Icon kann via "Klick -> Auf Updates prüfen -> Installieren" die neuste Version heruntergeladen und isntalliert werden. Nach erfolgreicher Installation muss der der Dienst einmal über das Tray Icon beendet und wieder gestartet werden.

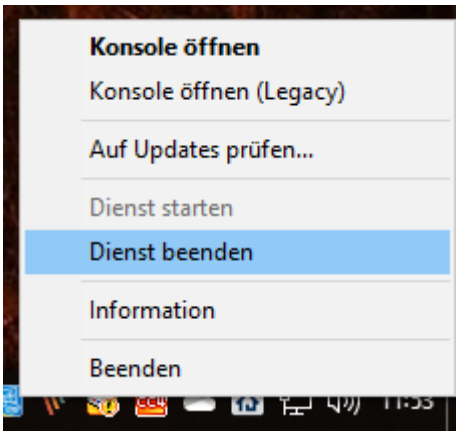



## DEINSTALLATION

Vor der Deinstallation von IP-Symcon, muss der Dienst angehalten werden.

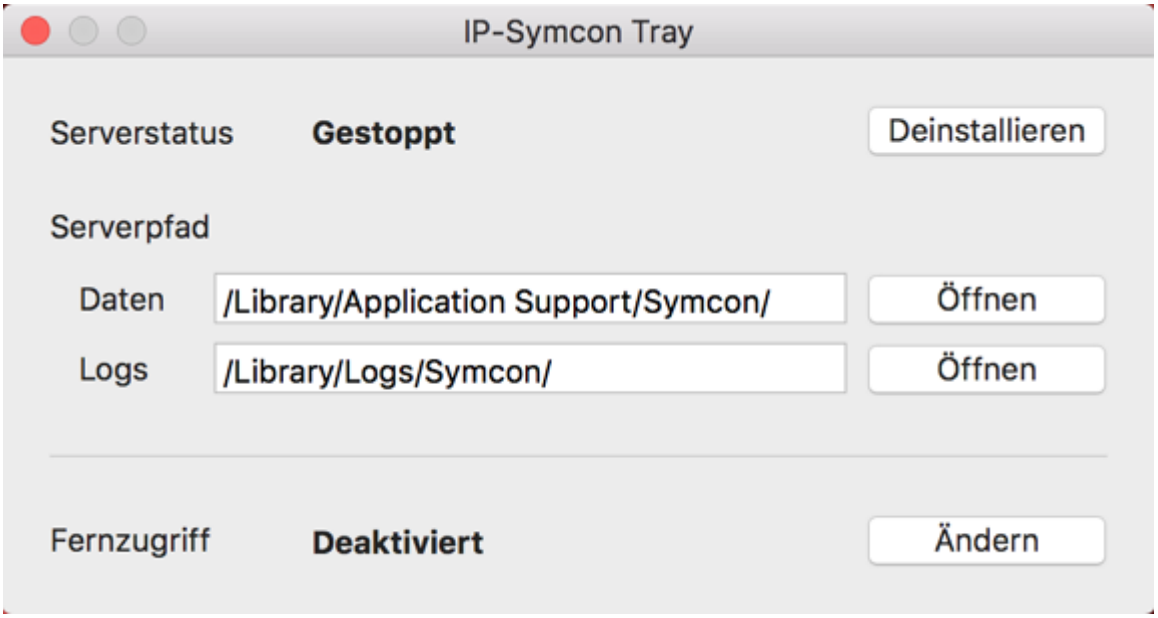
### DIENST STOPPEN

Über das Kontextmenü der Tray-Anwendung kann über den Menüpunkt “Dienst anhalten” der Dienst gestoppt werden.




### DIENST/SERVER DEINSTALLIEREN

Über den Dialog "Information" kann das Kontextmenü geöffnet werden. Folgendes sollte zu sehen sein:
MacOS




“Deinstallieren“ entfernt den Dienst von dem System.

## IP-SYMCON LÖSCHEN

Um die Deinstallation zu beenden, muss der komplette IP-Symcon Ordner gelöscht oder in den Papierkorb verschoben werden.

# Linux

Die Installation ist nur mit einer gültigen Subskription möglich. Dennoch kann ohne gültige Subskription eine vorhandene Datensicherung der Software wieder einspielt werden. Siehe: Plattformwechsel

Es wird eine regelmäßige Datensicherungen empfohlen, damit bei einem technischen Defekt das System auf dem möglichst aktuellen Stand bleibt. Zumal eine aktive Subskription gebraucht wird, um bei vollständigen Verlust eine IP-Symcon Version erneut downloaden zu können.

## INHALT


Installation
Update
Deinstallation


## INSTALLATION

### VERSIONSVORAUSSETZUNG

Welche Version von IP-Symcon mit welchem Ubuntu funktioniert, kann der Versionenübersicht entnommen werden.

### VORBEREITUNG

Zuerst folgenden Befehl ausführen und überprüfen, ob die vollständig Zeit korrekt ist, oder evtl. die Zeitzone noch eingestellt werden muss

```
date
```

ggf. zum Einstellen der Zeitzone

```
sudo dpkg-reconfigure tzdata
```

### INSTALLATION

```
wget -qO- https://apt.symcon.de/install.sh | bash /dev/stdin
```

Folgende Befehle ausführen:

```
sudo apt-get update
sudo apt-get install symcon
```

Das IP-Symcon WebFront sollte nun auf http://ipadresse:3777/ erreichbar sein.
Die IP-Symcon webbasierte Verwaltungskonsole kann über http://ipadresse:3777/console/ aufgerufen werden. Alternativ kann die Pro Konsole verwendet werden.

### WIE STARTE UND STOPPE ICH DEN IP-SYMCON-DIENST?

```
sudo /etc/init.d/symcon start
sudo /etc/init.d/symcon stop
sudo /etc/init.d/symcon restart
```

### WO FINDE ICH WAS?

/usr/bin/symcon - Executable
/usr/share/symcon/ - Static Data (IP-Symcon Installation)
/var/lib/symcon/ - Variable Data (Settings, Skripte, Medien...)
/var/log/symcon/ - Log Files (Logfiles...)

### WIE KANN ICH SCHAUEN, OB DER DIENST KORREKT LÄUFT?

```
sudo ps x | grep symcon
```

### WIE KANN ICH DAS LOGFILE ANSEHEN/VERFOLGEN?

```
tail -f /var/log/symcon/logfile.log
```

### FERNZUGRIFF KENNWORT SETZEN

Über Fernzugriff kann nachgeschlagen werden, wie der Fernzugriff eingerichtet wird.

## UPDATE

Über das Terminal müssen folgende Befehle genutzt werden, um IP-Symcon auf den aktuellsten Stand zu bringen.

```
sudo apt-get update
sudo apt-get upgrade
```



Sofern beim Upgrade das "symcon" Paket zurückgehalten wird, ist folgender Befehl erforderlich:

sudo apt-get dist-upgrade

Der Dienst wird automatisch beendet und im Anschluss neugestartet.

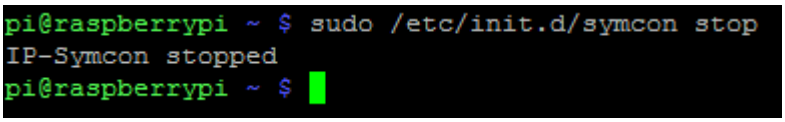
## DEINSTALLATION

Vor der Deinstallation von IP-Symcon, muss der Dienst angehalten werden.

### DIENST STOPPEN

Über die Konsole muss der Befehl eingegeben werden, welcher den IP-Symcon Server stoppt. Dieser ist für Linux und Raspberry Pi identisch.

```
sudo /etc/init.d/symcon stop
```




### DIENST/SERVER DEINSTALLIEREN

Über die Konsole muss der Befehl eingegeben werden, welcher IP-Symcon deinstalliert. Dieser ist für Linux und Raspberry Pi identisch.

```
sudo apt-get purge symcon
```

```
pi@raspberrypi ~ $ sudo apt-get purge symcon
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut.
Statusinformationen werden eingelesen.... Fertig
Die folgenden Pakete wurden automatisch installiert und werden nicht mehr benöti
gt:
  libfontconfig1-dev libpthread-stubs0 libpthread-stubs0-dev libx11-dev
  libxau-dev libxcb1-dev libxdmcp-dev libxpm-dev x11proto-core-dev
  x11proto-input-dev x11proto-kb-dev xorg-sgml-doctools xtrans-dev
Verwenden Sie »apt-get autoremove«, um sie zu entfernen.
Die folgenden Pakete werden ENTFERNT:
  symcon*
0 aktualisiert, 0 neu installiert, 1 zu entfernen und 0 nicht aktualisiert.
Nach dieser Operation werden 0 B Plattenplatz zusätzlich benutzt.
Möchten Sie fortfahren [J/n]? j
(Lese Datenbank ... 104352 Dateien und Verzeichnisse sind derzeit installiert.)
Entfernen von symcon ...
IP-Symcon is not running.
dpkg: Warnung: Während Entfernens von symcon ist Verzeichnis »/etc/symcon/forms«
 nicht leer, wird daher nicht gelöscht
dpkg: Warnung: Während Entfernens von symcon ist Verzeichnis »/etc/symcon/webfro
nt« nicht leer, wird daher nicht gelöscht
Trigger für man-db werden verarbeitet ...
pi@raspberrypi ~ $
```

## IP-SYMCON LÖSCHEN

Um die Deinstallation zu beenden, muss der komplette IP-Symcon-Daten (/var/lib/symcon) Ordner gelöscht oder in den Papierkorb verschoben werden.

# Raspberry Pi

Die Installation ist nur mit einer gültigen Subskription möglich. Dennoch kann ohne gültige Subskription eine vorhandene Datensicherung der Software wieder einspielt werden. Siehe: Plattformwechsel

Es wird eine regelmäßige Datensicherungen empfohlen, damit bei einem technischen Defekt das System auf dem möglichst aktuellen Stand bleibt. Zumal eine aktive Subskription gebraucht wird, um bei vollständigen Verlust eine IP-Symcon Version erneut downloaden zu können.

## INHALT

Installation
Update
Deinstallation

## INSTALLATION

### VERSIONSVORAUSSETZUNG

Welche Version von IP-Symcon mit welchem Betriebssystem funktioniert, kann der Versionenübersicht entnommen werden.

### VORBEREITUNG

Zuerst folgenden Befehl ausführen und überprüfen, ob die vollständig Zeit korrekt ist, oder evtl. die Zeitzone noch eingestellt werden muss

```
date
```

ggf. zum Einstellen der Zeitzone

```
sudo raspi-config
```

### INSTALLATION

```
wget -qO- https://apt.symcon.de/install.sh | bash /dev/stdin
```

Auf der Shell jetzt folgende Befehle ausführen

```
sudo apt-get update
sudo apt-get install symcon
```

Das IP-Symcon WebFront sollte nun auf http://ipadresse:3777/ erreichbar sein.
Die IP-Symcon webbasierte Verwaltungskonsole kann über http://ipadresse:3777/console/ aufgerufen werden. Alternativ kann die Pro Konsole verwendet werden.

### WIE STARTE UND STOPPE ICH DEN IP-SYMCON-DIENST?

```
sudo /etc/init.d/symcon start
sudo /etc/init.d/symcon stop
sudo /etc/init.d/symcon restart
```

### WO FINDE ICH WAS?

/usr/bin/symcon - Executable
/usr/share/symcon/ - Static Data (IP-Symcon Installation)
/var/lib/symcon/ - Variable Data (Settings, Skripte, Medien...)
/var/log/symcon/ - Log Files (Logfiles...)

### WIE KANN ICH SCHAUEN, OB DER DIENST KORREKT LÄUFT?

```
sudo ps x | grep symcon
```

### WIE KANN ICH DAS LOGFILE ANSEHEN/VERFOLGEN?

```
tail -f /var/log/symcon/logfile.log
```

### FERNZUGRIFF KENNWORT SETZEN

Über Fernzugriff kann nachgeschlagen werden, wie der Fernzugriff eingerichtet wird.

## UPDATE

Über das Terminal müssen folgende Befehle genutzt werden, um IP-Symcon auf den aktuellsten Stand zu bringen.

```
sudo apt-get update
sudo apt-get upgrade
```



Sofern beim Upgrade das "symcon" Paket zurückgehalten wird, ist folgender Befehl erforderlich:

sudo apt-get dist-upgrade

Der Dienst wird automatisch beendet und im Anschluss neugestartet.

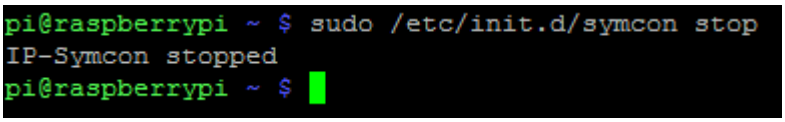
## DEINSTALLATION

Vor der Deinstallation von IP-Symcon, muss der Dienst angehalten werden.

### DIENST STOPPEN

Über die Konsole muss der Befehl eingegeben werden, welcher den IP-Symcon Server stoppt. Dieser ist für Linux und Raspberry Pi identisch.

```
sudo /etc/init.d/symcon stop
```




### DIENST/SERVER DEINSTALLIEREN

Über die Konsole muss der Befehl eingegeben werden, welcher IP-Symcon deinstalliert. Dieser ist für Linux und Raspberry Pi identisch.

```
sudo apt-get purge symcon
```

```
pi@raspberrypi ~ $ sudo apt-get purge symcon
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut.
Statusinformationen werden eingelesen.... Fertig
Die folgenden Pakete wurden automatisch installiert und werden nicht mehr benöti
gt:
  libfontconfig1-dev libpthread-stubs0 libpthread-stubs0-dev libx11-dev
  libxau-dev libxcb1-dev libxdmcp-dev libxpm-dev x11proto-core-dev
  x11proto-input-dev x11proto-kb-dev xorg-sgml-doctools xtrans-dev
Verwenden Sie »apt-get autoremove«, um sie zu entfernen.
Die folgenden Pakete werden ENTFERNT:
  symcon*
0 aktualisiert, 0 neu installiert, 1 zu entfernen und 0 nicht aktualisiert.
Nach dieser Operation werden 0 B Plattenplatz zusätzlich benutzt.
Möchten Sie fortfahren [J/n]? j
(Lese Datenbank ... 104352 Dateien und Verzeichnisse sind derzeit installiert.)
Entfernen von symcon ...
IP-Symcon is not running.
dpkg: Warnung: Während Entfernens von symcon ist Verzeichnis »/etc/symcon/forms«
 nicht leer, wird daher nicht gelöscht
dpkg: Warnung: Während Entfernens von symcon ist Verzeichnis »/etc/symcon/webfro
nt« nicht leer, wird daher nicht gelöscht
Trigger für man-db werden verarbeitet ...
pi@raspberrypi ~ $
```

## IP-SYMCON LÖSCHEN

Um die Deinstallation zu beenden, muss der komplette IP-Symcon Daten-Ordner (/var/lib/symcon) gelöscht oder in den Papierkorb verschoben werden.

# Docker

Die Installation ist nur mit einer gültigen Subskription möglich.

Es wird eine regelmäßige Datensicherungen empfohlen, damit bei einem technischen Defekt das System auf dem möglichst aktuellen Stand bleibt. Zumal eine aktive Subskription gebraucht wird, um bei vollständigen Verlust eine IP-Symcon Version erneut downloaden zu können.

Es gibt seperate Installationsanleitungen für QNAP und Synology

## INHALT


Installation
Update
Hinweise


## INSTALLATION

### VORAUSSETZUNG

Intel/AMD 64-Bit, ARM7, ARM64 CPU
Docker

Da Container im Docker Kontext nicht "aktualisiert" werden können, müssen diese gelöscht und in der neuen Version neu erstellt werden. Dies ist auch problemlos möglich, da sich alle Daten außerhalb des Containers befinden.

### LIMITATIONEN

Bisher ist die Sprache auf de_DE gesetzt
Bisher ist die Zeitzone auf Europe/Berlin gesetzt
Wenn der Container nicht interaktiv, sondern losgelöst laufen soll, muss das Flag -d angegeben werden
Bisher ist nur Port 3777 gemappt. Werden weitere Ports benötigt, müssen diese ebenfalls angegeben werden!

### INSTALLIEREN

```
docker pull symcon/symcon:stable
```

### STARTEN

```
docker run --rm \
--name symcon \
--hostname symcon \
-p 3777:3777 \
-v /opt/symcon/data:/var/lib/symcon \
-v /opt/symcon/log:/var/log/symcon \
-v /opt/symcon:/root \
symcon/symcon:stable
```

Das IP-Symcon WebFront sollte nun auf http://ipadresse:3777/ erreichbar sein.
Die IP-Symcon webbasierte Verwaltungskonsole kann über http://ipadresse:3777/console/ aufgerufen werden. Alternativ kann die Pro Konsole verwendet werden.

## UPDATE

Installieren erneut ausführen
Starten erneut ausführen
Dadurch wird der Container entfernt und neu erstellt.
Die Daten befinden sich außerhalb des Containers und bleiben erhalten.

## KOMFORTABLERE UPDATES MIT PORTAINER

Das Tool "Portainer" hilft bei der Aktualisierung von Docker Containern, sodass dies über einen Knopfdruck passieren kann. Das webbasierte Tool steht kostenlos zur Verfügung und kann ebenfalls als Docker Container installiert werden.

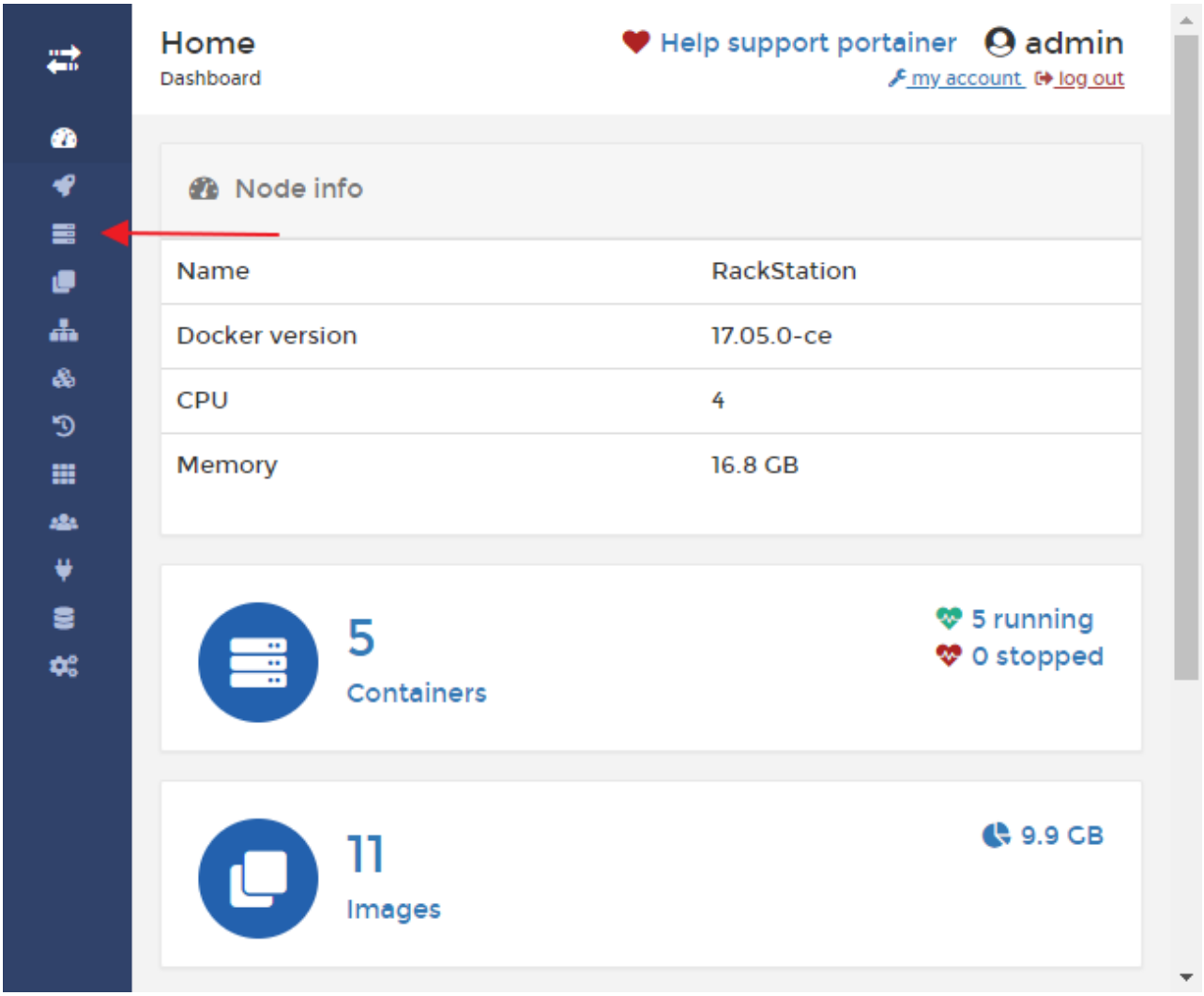
### INSTALLIEREN

```
docker volume create portainer_data
docker run -d -p 9000:9000 -v /var/run/docker.sock:/var/run/
docker.sock -v portainer_data:/data portainer/portainer
```

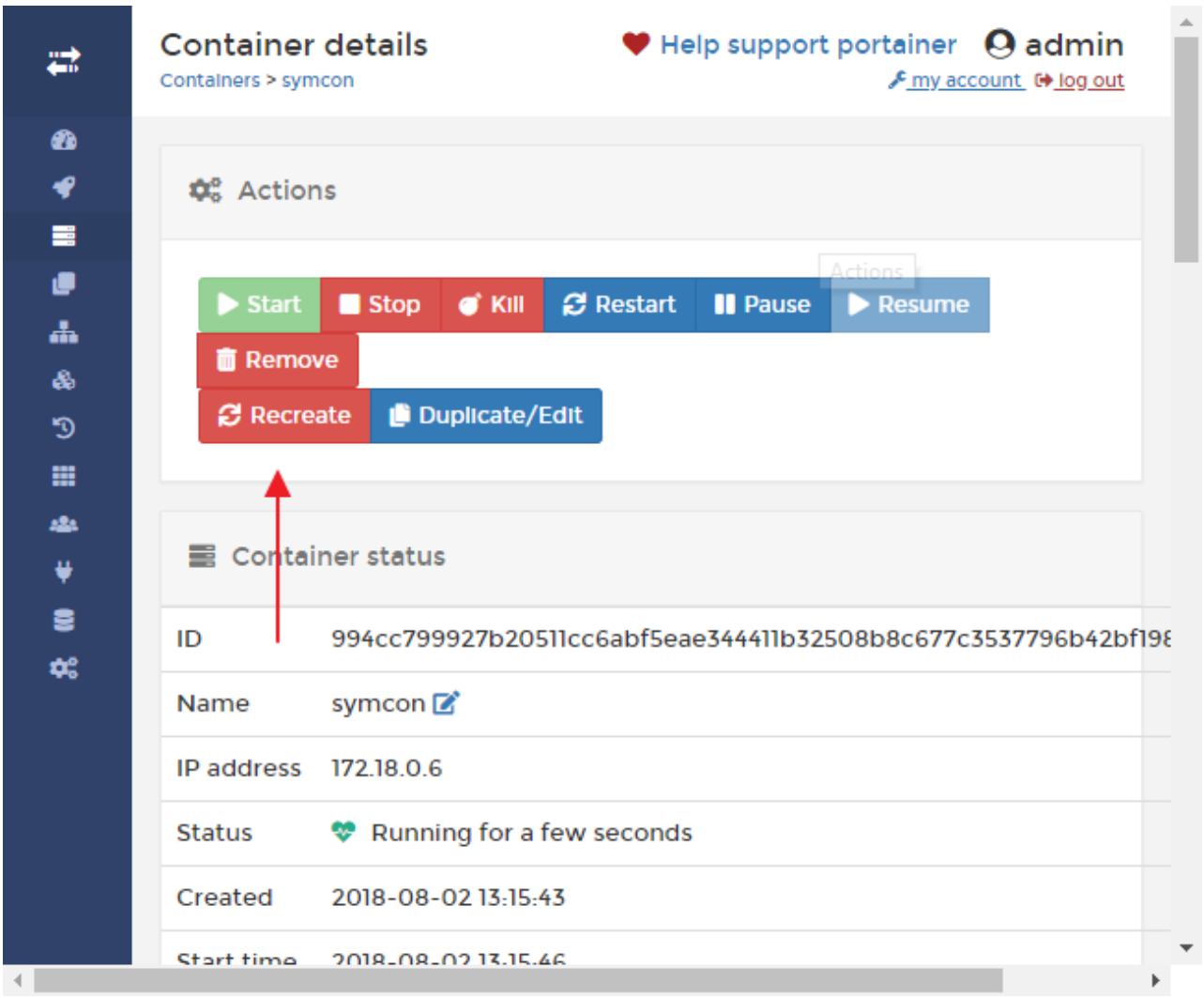


Es kann erforderlich sein den "docker run" Befehl als "sudo docker run" zu starten

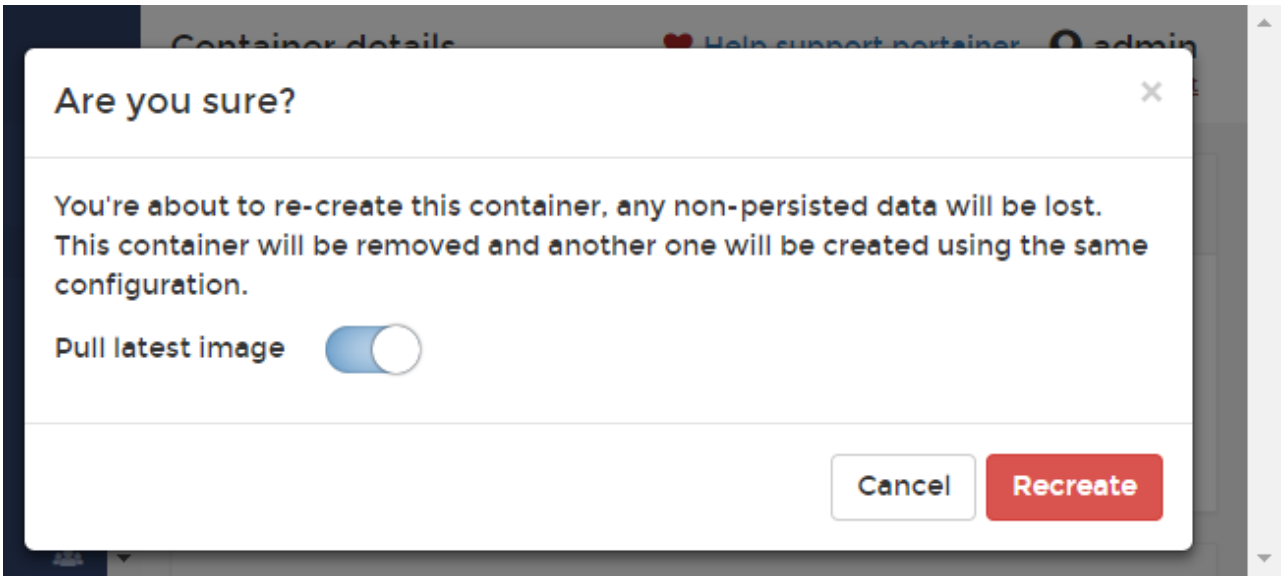
Nach der Installation steht die Oberfläche unter http://127.0.0.1:9000/ zur Verfügung. Gegebenfalls muss die IP-Adresse noch angepasst werden. Bei der ersten Einrichtung muss ein Kennwort vergeben werden und außerdem im nächsten Schritt noch eine Verbindung zum lokalen Server hergestellt werden. Nach erforgreicher Installation sollte das Interface wie folgt aussehen.



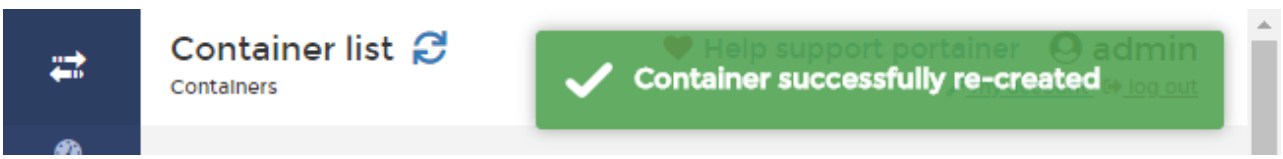

Im Reiter "Containers" kann dann der "symcon" Container ausgewählt werden. Über die Option "Recreate" ist eine komfortable Aktualisierung möglich. Der Container wird dabei gelöscht, das neue Image heruntergeladen und anschließend mit den selben Einstellungen neu erstellt. Dies kann je nach Internet-Verbindung einige Minuten dauern.




Wichtig ist, dass im Abfrage-Dialog die Option "Pull latest image" aktiviert ist.




Nach dem Aktualisieren wird kurzzeitig folgende Meldung angezeigt.



# HINWEISE

## NATSUPPORT FÜR KNX UND HOMEMATIC

Aufgrund der Containerstruktur wird bei KNX- und HomeMatic-Nachrichten standardmäßig eine nur intern verwendete IP-Adresse als Antwortadresse mitgeteilt, so dass die Antworten des Systems nicht korrekt empfangen werden können. Um diese Nachrichten korrekt zu empfangen muss der NATSupport eingerichtet werden.

Der Spezialschalter "NATSupport" muss aktiviert werden und danach IP-Symcon neu gestartet werden.

Für den Container muss der Port des entsprechenden Systems gemappt werden. Bei KNX ist dies standardmäßig 52000 und bei HomeMatic 5544. So müsste beispielsweise bei KNX der Port 52000 auf 52000 gemappt werden.

In der betroffenen I/O- oder Splitter-Instanz erscheint nun ein neues Feld "PublicIP". Hier muss die IP eingetragen werden unter welcher Docker von aussen erreichbar ist. Für KNX muss diese im Splitter "EIB Gateway" eingetragen werden und für HomeMatic im I/O "HomeMatic Socket".

# QNAP

Die Installation ist nur mit einer gültigen Subskription möglich.

Es wird eine regelmäßige Datensicherungen empfohlen, damit bei einem technischen Defekt das System auf dem möglichst aktuellen Stand bleibt. Zumal eine aktive Subskription gebraucht wird, um bei vollständigen Verlust eine IP-Symcon Version erneut downloaden zu können.

## INHALT


Installation
Update
Deinstallation
Hinweise

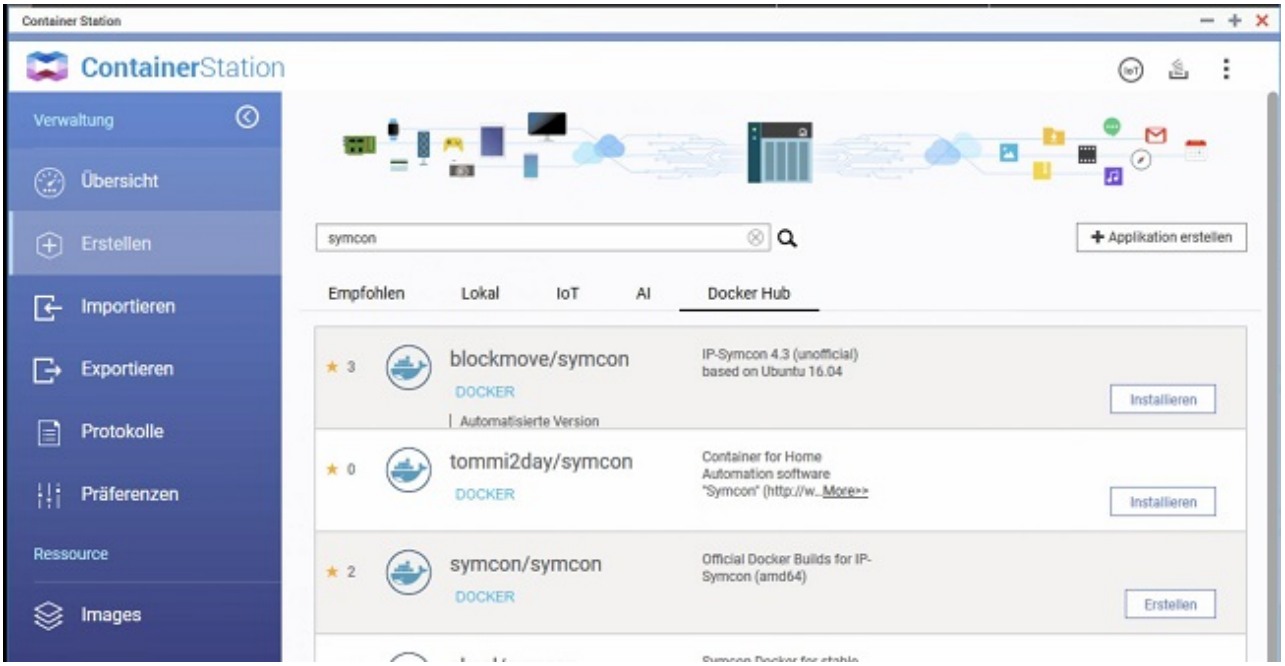

## INSTALLATION

### VORAUSSETZUNG

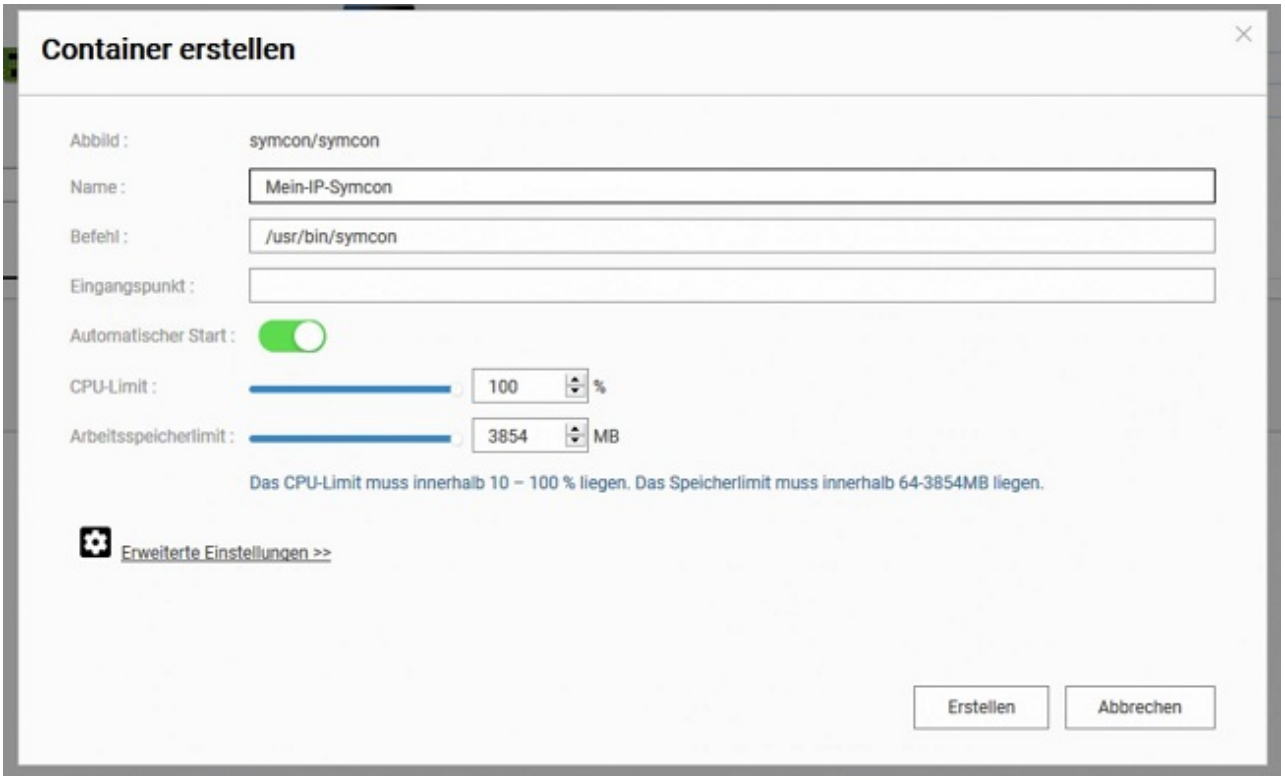
NAS mit Container Station Unterstützung

### INSTALLIEREN

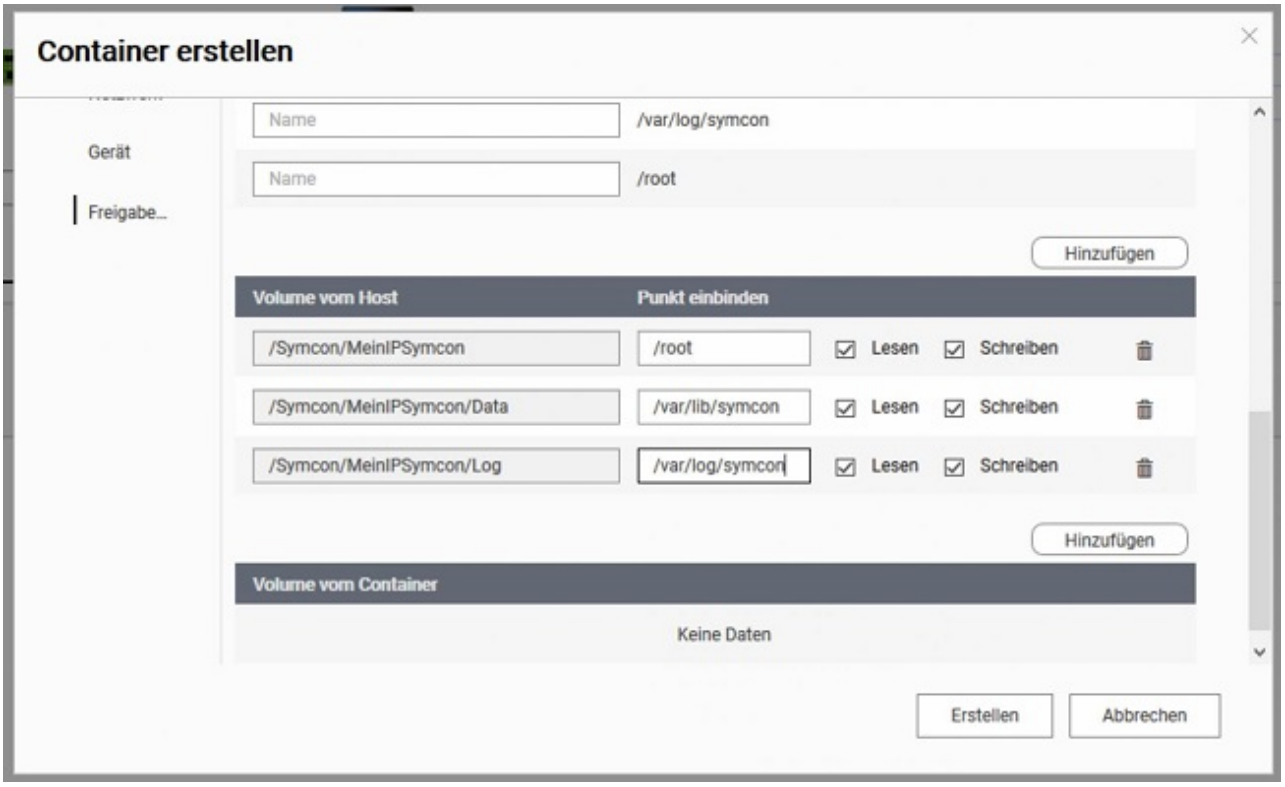
Container Station öffnen
Nach symcon/symcon im Docker Hub suchen und das Abbild installieren/erstellen (Beim Tag z.B. stable auswählen)



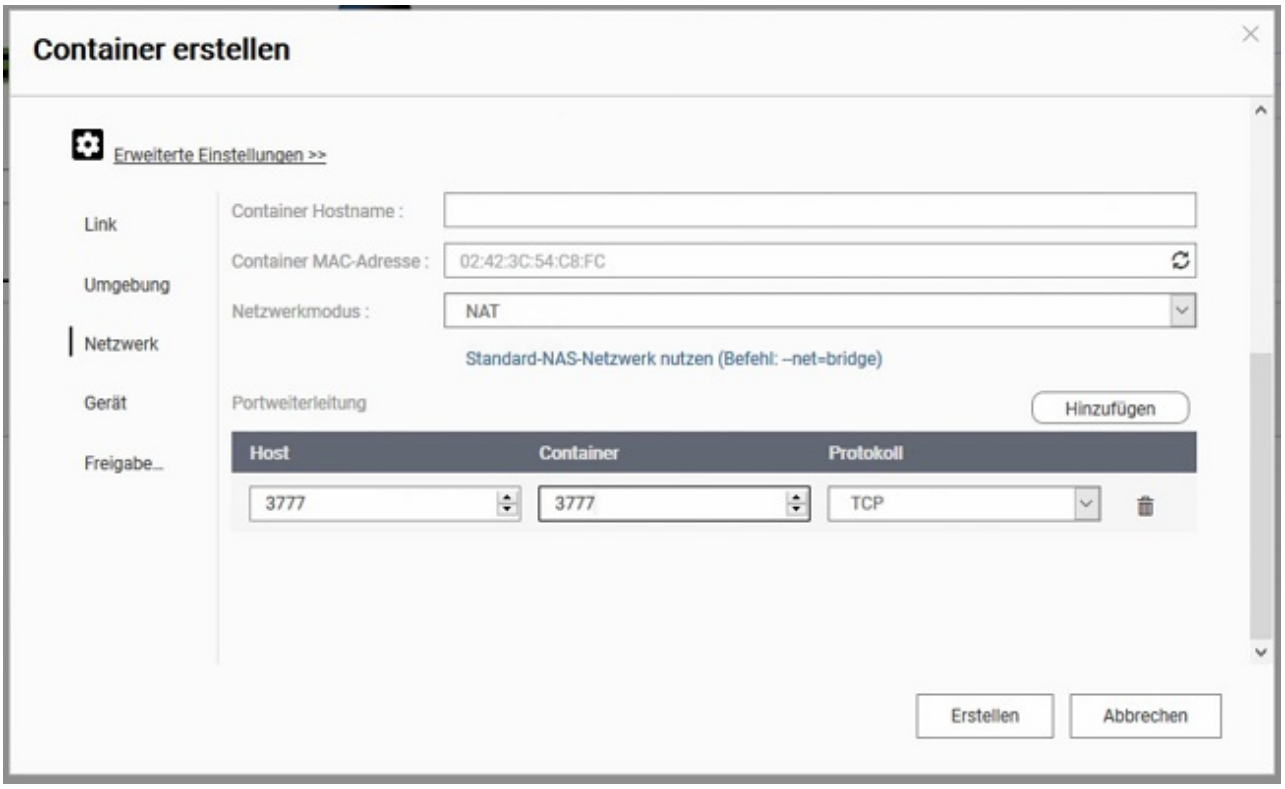

Namen vergeben und die Erweiterten Einstellungen öffnen




Freigabe Einstellungen anpassen (/root, /var/lib/symcon, /var/log/symcon müssen angegeben werden!)




Netzwerk Einstellungen anpassen (3777 auf 3777 TCP)




Mit Erstellen bestätigen um den Container zu erstellen und zu starten
Das IP-Symcon WebFront sollte nun auf http://ipadresse:3777/ erreichbar sein.
Die IP-Symcon webbasierte Verwaltungskonsole kann über http://ipadresse:3777/console/ aufgerufen werden. Alternativ kann die Pro Konsole verwendet werden.

## FERNZUGRIFF KENNWORT SETZEN

Über Fernzugriff kann nachgeschlagen werden, wie der Fernzugriff eingerichtet wird.

## UPDATE

Zum Update ist es erforderlich den Container zu entfernen und dann neu zu erstellen, wie unter "Installieren" beschrieben ist. Über das Zusatztool "Portainer" kann das Aktualisieren wesentlich vereinfacht werden. Eine Anleitung dazu befindet sich im Docker Bereich .

## DEINSTALLATION

Für die Deinstallation kann der Container gestoppt und danach entfernt werden. Die verknüpften Verzeichnisse für die Daten müssen manuell entfernt werden.

## HINWEISE

### NATSUPPORT FÜR KNX UND HOMEMATIC

Aufgrund der Containerstruktur wird bei KNX- und HomeMatic-Nachrichten standardmäßig eine nur intern verwendete IP-Adresse als Antwortadresse mitgeteilt, so dass die Antworten des Systems nicht korrekt empfangen werden können. Um diese Nachrichten korrekt zu empfangen muss der NATSupport eingerichtet werden.

Der Spezialschalter "NATSupport" muss aktiviert werden und danach IP-Symcon neu gestartet werden.

Für den Container muss der Port des entsprechenden Systems gemappt werden. Bei KNX ist dies standardmäßig 52000 und bei HomeMatic 5544. So müsste beispielsweise bei KNX der Port 52000 auf 52000 gemappt werden.

In der betroffenen I/O- oder Splitter-Instanz erscheint nun ein neues Feld "PublicIP". Hier muss die IP eingetragen werden unter welcher Docker von aussen erreichbar ist. Für KNX muss diese im Splitter "EIB Gateway" eingetragen werden und für HomeMatic im I/O "HomeMatic Socket".

# Synology

Die Installation ist nur mit einer gültigen Subskription möglich.

Es wird eine regelmäßige Datensicherungen empfohlen, damit bei einem technischen Defekt das System auf dem möglichst aktuellen Stand bleibt. Zumal eine aktive Subskription gebraucht wird, um bei vollständigen Verlust eine IP-Symcon Version erneut downloaden zu können.

## INHALT


Installation
Update
Deinstallation
Hinweise

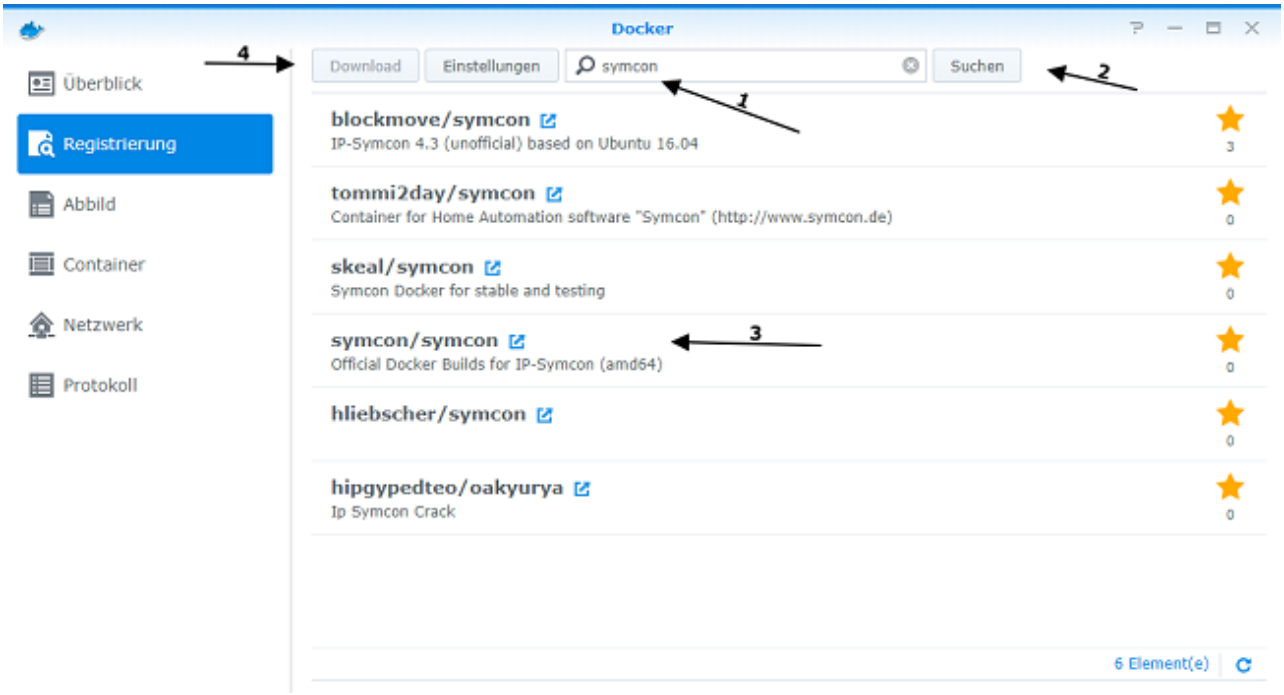

## INSTALLATION

### VORAUSSETZUNG

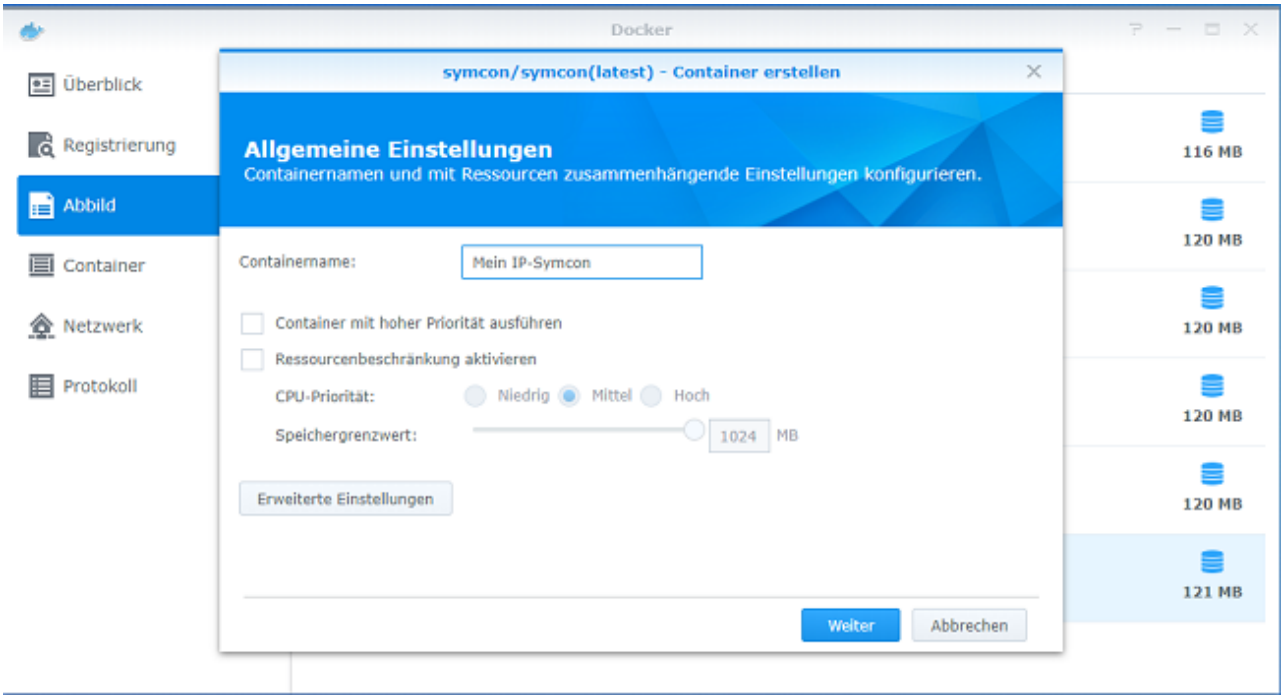
NAS mit Container Manager Unterstützung

### INSTALLIEREN

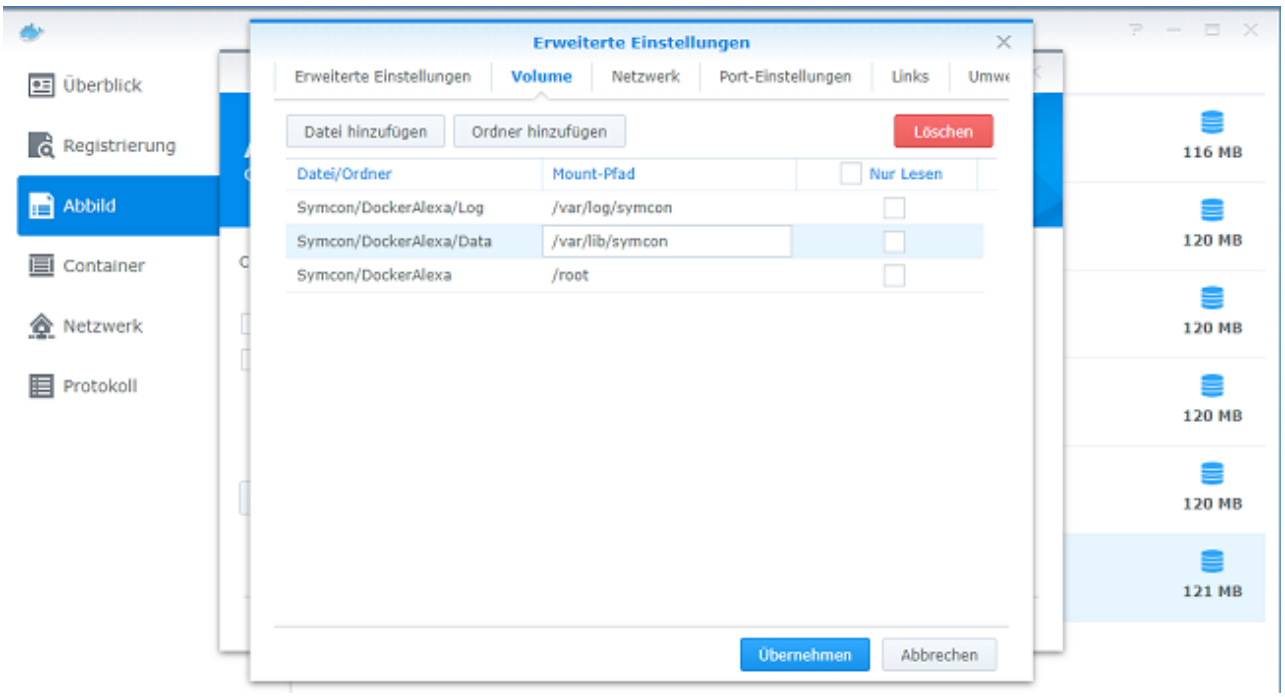
Nach Symcon in der Registrierung suchen und das Abbild herunterladen (Beim Tag z.B. latest auswählen)



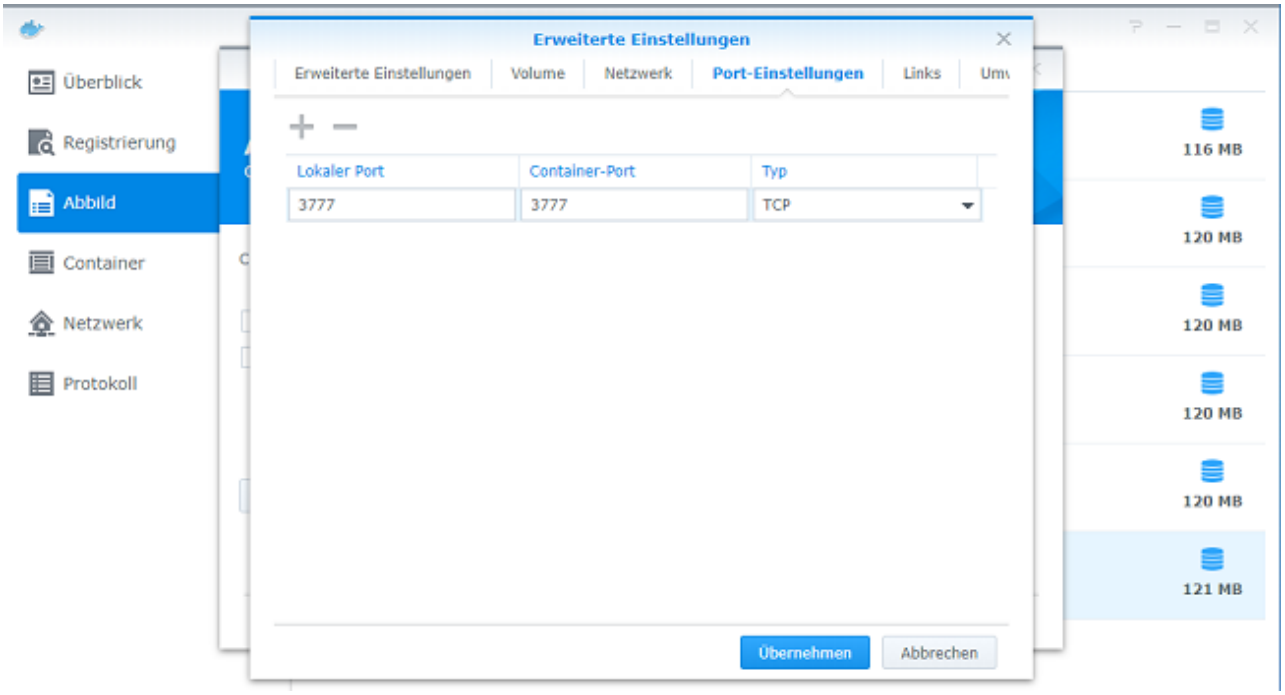

Unter Abbild das neue Abbild auswählen und auf Start klicken
Namen vergeben




Volume Einstellungen anpassen (/root, /var/lib/symcon, /var/log/symcon müssen angegeben werden!)




Port Einstellungen anpassen (3777 auf 3777)




Mit OK bestätigen, auf Weiter klicken und mit Übernehmen den Container starten
Das IP-Symcon WebFront sollte nun auf http://ipadresse:3777/ erreichbar sein.
Die IP-Symcon webbasierte Verwaltungskonsole kann über http://ipadresse:3777/console/ aufgerufen werden. Alternativ kann die Pro Konsole verwendet werden.

## UPDATE

Es wird empfohlen über das Zusatztool "Portainer" zu Aktualisieren. Eine Anleitung dazu befindet sich im Docker Bereich .
Alternativ kann auch wiefolgt der Container aktualisiert werden:
Docker-Abbild neu herunterladen (Registrierung - Suche "symcon" - Download)
Warten bis das neue Image heruntergeladen ist (bei Abbild verschwindet die Zahl)

Symcon Container stoppen
Symcon Container leeren (Aktion - Inhalt löschen)
Symcon Container wieder starten

## DEINSTALLATION

Für die Deinstallation kann der Container gestoppt und danach entfernt werden. Die verknüpften Verzeichnisse für die Daten müssen manuell entfernt werden.

## HINWEISE

### NATSUPPORT FÜR KNX UND HOMEMATIC

Aufgrund der Containerstruktur wird bei KNX- und HomeMatic-Nachrichten standardmäßig eine nur intern verwendete IP-Adresse als Antwortadresse mitgeteilt, so dass die Antworten des Systems nicht korrekt empfangen werden können. Um diese Nachrichten korrekt zu empfangen muss der NATSupport eingerichtet werden.

Der Spezialschalter "NATSupport" muss aktiviert werden und danach IP-Symcon neu gestartet werden.
Für den Container muss der Port des entsprechenden Systems gemappt werden. Bei KNX ist dies standardmäßig 52000 und bei HomeMatic 5544. So müsste beispielsweise bei KNX der Port 52000 auf 52000 gemappt werden.
In der betroffenen I/O- oder Splitter-Instanz erscheint nun ein neues Feld "PublicIP". Hier muss die IP eingetragen werden unter welcher Docker von aussen erreichbar ist. Für KNX muss diese im Splitter "EIB Gateway" eingetragen werden und für HomeMatic im I/O "HomeMatic Socket".

# SymBox

Die Installation ist nur mit einer gültigen Subskription möglich. Dennoch kann ohne gültige Subskription eine vorhandene Datensicherung der Software wieder einspielt werden. Siehe: Plattformwechsel

Es wird eine regelmäßige Datensicherungen empfohlen, damit bei einem technischen Defekt das System auf dem möglichst aktuellen Stand bleibt. Zumal eine aktive Subskription gebraucht wird, um bei vollständigen Verlust eine IP-Symcon Version erneut downloaden zu können.

## INSTALLATION/UPDATE/WERKSEINSTELLUNGEN

Für weitere Informationen siehe Installationsanleitung .

# Firewall

Damit IP-Symcon mit den verschiedenen Diensten aus dem Internet kommunizieren kann, sind folgende Einstellung bei einer Firewall vorzunehmen.

## EMPFOHLENE EINSTELLUNGEN

| ADRESSE | PORT | PROTOKOLL | FUNKTION |
|---|---|---|---|
| apt.symcon.de | 443 | TCP | Installation, Updates |
| api.symcon.de | 443 | TCP | Module Store, Persönlicher Bereich |
| symcon-store.s3.eu-west-1.amazonaws.com | 443 | TCP | Module Store (Daten) |
| live.symcon.de | 443 | TCP | Installation, Updates, Connect, Push-Nachrichten, Anzeige der Subskriptionslaufzeit |
| zwdb.symcon.de | 443 | TCP | Z-Wave Gerätedatenbank |
| dwd.symcon.de | 443 | TCP | WebFront DWD-Wetter |
| ipmagic.de | 50000 | TCP | IP-Symcon Connect |
| github.com | 443 | TCP | Module Control (sofern genutzt) |

## EINSTELLUNGEN NUR RELEVANT FÜR SYMBOX

| ADRESSE | PORT | PROTOKOLL | FUNKTION |
|---|---|---|---|
| ipmagic.de | 60000 | TCP | SymOS Connect |
| ipmagic.de | 53671 | TCP | SymOS Connect (KNX) |
| ipmagic.de | 53672 | TCP | SymOS Connect (KNX) |
| ipmagic.de | 54114 | TCP | SymOS Connect (LCN) |

## CLOUDFRONT-EDGE-SERVER

Die *.symcon.de Adressen werden über AWS in der EU-West-1 Zone (Irland) abgebildet. Die Endpunkte terminieren auf AWS CloudFront und haben eine TTL von 60 Sekunden. Bei einer dynamischen Auflösung ist somit sicherzustellen, dass die Firewall die aktuell genutzten IP-Adressen oft genug ermittelt.
Aktuelle Listen der genutzten IP-Adressen befinden sich hier: Standorte und IP-Adressbereiche von CloudFront-Edge-Servern

# Plattformwechsel

Ein Plattformwechsel von IP-Symcon ist problemlos möglich. Dies funktioniert ohne jeglichen Verlust von Einstellungen oder Daten.

## SCHRITTE ZUM PLATTFORMWECHSEL

Folgende Schritte müssen beachtet werden.
IP-Symcon auf der Zielplattform installieren.
IPS-Dienst auf der Start- und Zielplattform beenden.
Backup erstellen auf der Startplattform.
Dann via Backup einspielen dieses in die neue Plattform integrieren.
Dienst auf der Zielplattform starten.

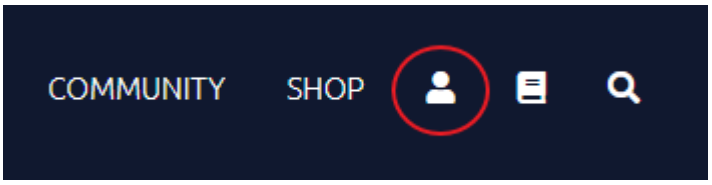


Es sind immer die Start- und Zielversionen von IP-Symcon zu beachten. Wenn die Version unterschiedlich ist, müssen gegebenfalls die Migrationen beachtet werden. z.B. bei einem Wechsel von 3.4 zu einer höheren Version ist dies unabdingbar.

# Lizenzverwaltung

## LIZENZVERWALTUNG

Zur einfachen Handhabung und Übersicht gibt es die Möglichkeit die Lizenzen auf der Homepage in der Lizenzverwaltung zu verwalten.
In der Lizenzverwaltung kann unter anderem die Dauer der Subskription nachgesehen oder die Lizenzdatei erneut verschickt werden.
Die Lizenzverwaltung ist im Persönlichen Bereich zu finden. Dieser ist über das Personenicon oben rechts auf der Homepage erreichbar. Nach der Anmeldung kann die Lizenzverwaltung geöffnet werden.







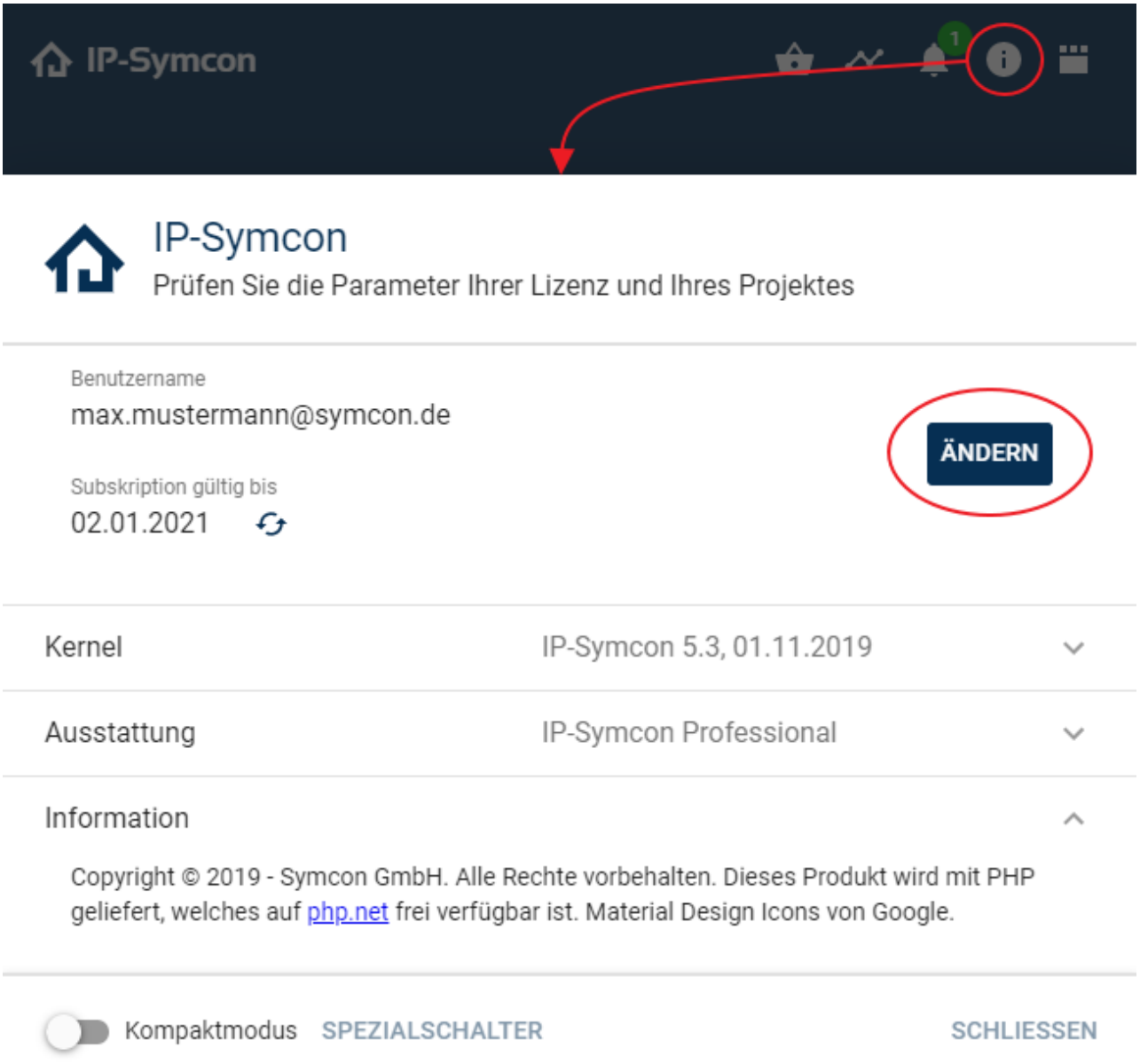
Die Konten für Forum und Homepage sind momentan noch getrennt. Zukünftig ist allerdings vorgesehen diese zusammenzulegen.

## LIZENZ ÄNDERN

Nach jeglicher Lizenzänderung muss der Dienst von IP-Symcon neugestartet werden

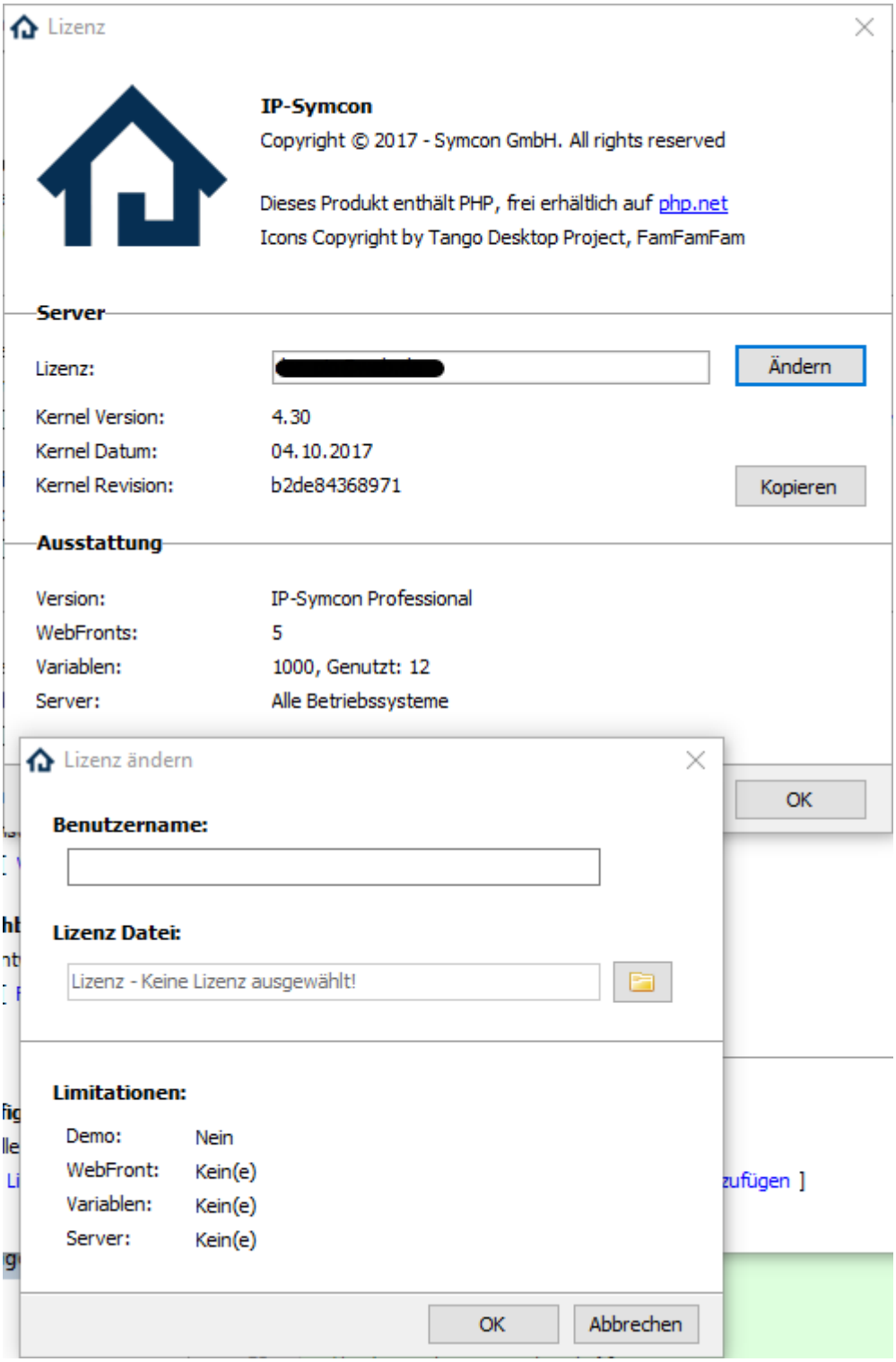
## AB VERSION 5.0

Die Lizenz kann innerhalb der Verwaltungskonsole im Informationsmenü geändert werden.



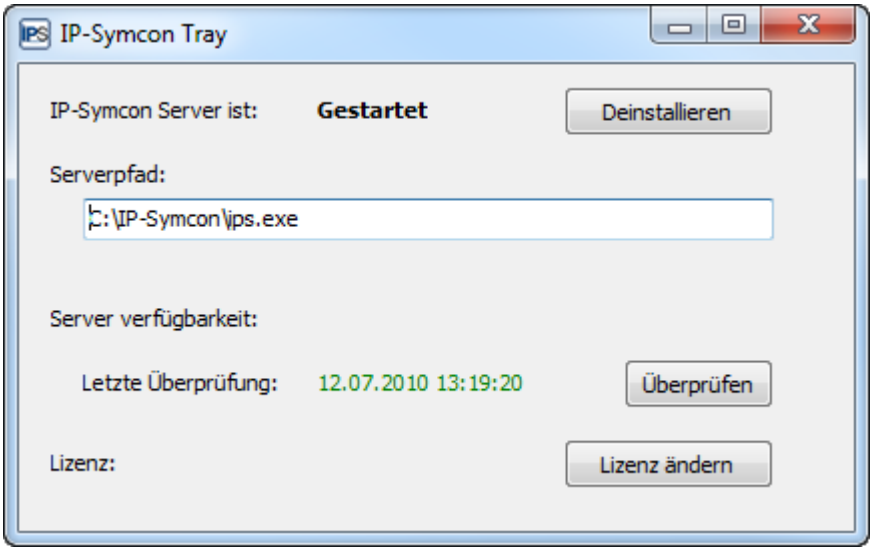

## AB VERSION 4.0 BIS 4.4

Die Lizenz kann innerhalb der Verwaltungskonsole unter "Mein IP-Symcon -> Lizenz anzeigen" geändert werden.




## BIS VERSION 3.4

Eine neue Lizenz die Aufgrund eines Lizenz-Upgrades erhalten wurde, kann im Kontextmenü der Tray-Anwendung unter den Menüpunkt “Information” eingetragen werden. Das Kontextmenü kann mit einem "Rechtsklick" auf die Tray-Anwendung geöffnet werden.




Über den Button “Lizenz ändern” kann der Dialog geöffnet werden. In diesem kann der Benutzername angegeben und die neue Lizenzdatei auswählt werden. Nach dem Bestätigen wird die neue Lizenz gespeichert. Zur Aktivierung der neuen Lizenz muss der IP-Symcon Dienst einmal neugestartet werden.

# Migrationen

Die Migrationen erläutern die Features und Fixes der jeweiligen Version.
Ebenso ist eine Beschreibung vorhanden, was bei Versionswechsel bzw. -update zu beachten ist.

| MIGRATION | VERÖFFENTLICHUNG |
| --- | --- |
| V8.0->V8.1 | Q3/2025 |
| V7.2->V8.0 | Q1/2025 |
| V7.1->V7.2 | Q3/2024 |
| V7.0->V7.1 | Q1/2024 |
| V6.4->V7.0 | Q4/2023 |
| V6.3->V6.4 | Q2/2023 |
| V6.2->V6.3 | Q4/2022 |
| V6.1->V6.2 | Q2/2022 |
| V6.0->V6.1 | Q1/2022 |
| V5.5->V6.0 | Q3/2021 |
| V5.4->V5.5 | Q4/2020 |
| V5.3->V5.4 | Q2/2020 |
| V5.2->V5.3 | Q4/2019 |
| V5.1->V5.2 | Q3/2019 |
| V5.0->V5.1 | Q2/2019 |
| V4.4->V5.0 | Q3/2018 |
| V4.3->V4.4 | Q4/2017 |
| V4.2->V4.3 | Q3/2017 |
| V4.1->V4.2 | Q2/2017 |
| V4.0->V4.1 | Q4/2016 |
| V3.4->V4.0 | Q1/2016 |
| V3.3->V3.4 | Q2/2015 |
| V3.2->V3.3 | Q1/2015 |
| V3.1->V3.2 | Q3/2014 |
| V3.0->V3.1 | Q2/2014 |
| V2.7->V3.0 | Q3/2013 |
| V2.6->V2.7 | Q1/2013 |
| V2.5->V2.6 | Q3/2012 |
| V2.4->V2.5 | Q4/2011 |
| V2.3->V2.4 | Q2/2011 |
| V2.2->V2.3 | Q3/2010 |
| V2.1->V2.2 | Q1/2010 |
| V2.0->V2.1 | Q3/2009 |
| V1.0->V2.0 | Q1/2009 |
| V0.0->V1.0 | Q1/2005 |

# V8.0->V8.1 (Q3/2025)

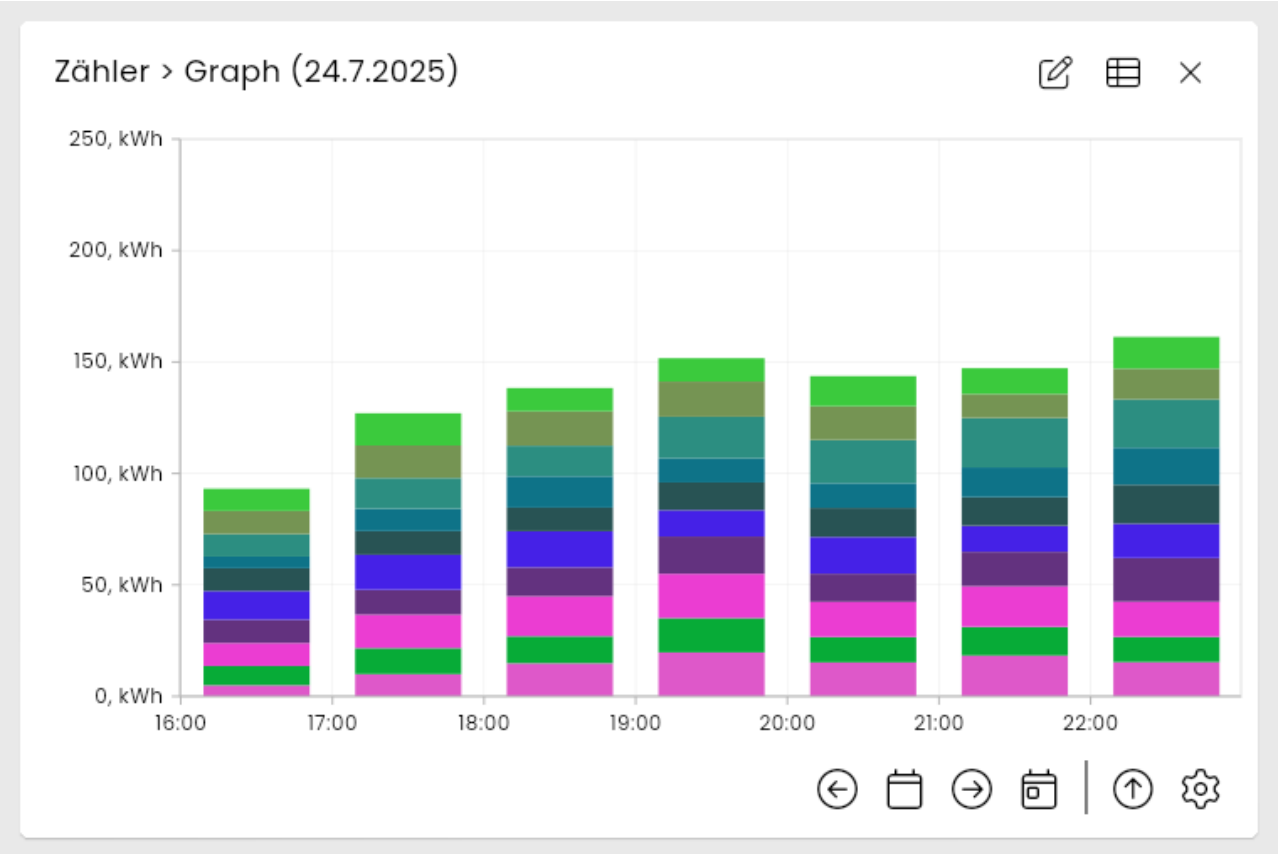
Wesentliche Änderungen
Migration
Liste der Änderungen
Neue und aktualisierte Module

## WESENTLICHE ÄNDERUNGEN

Moderne Diagramme
Energie Manager
HomeMatic HCU (Beta)
Enterprise Features für alle

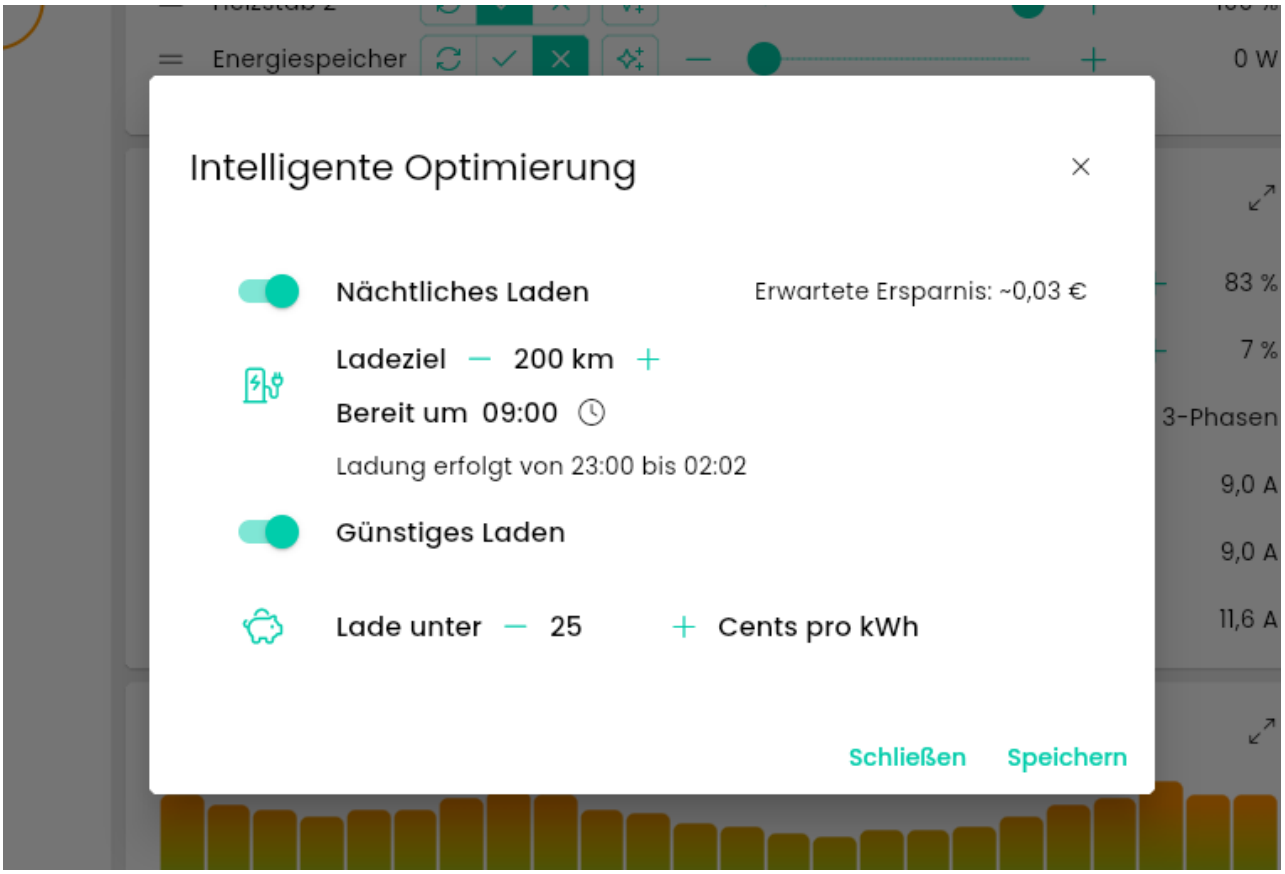
### MODERNE DIAGRAMME

Diagramme informieren über den Verlauf von Werten und sind somit kaum aus Symcon wegzudenken. Diese Ansicht wurde komplett überarbeitet. Damit sehen die Diagramme nicht nur modern aus, sie bieten auch noch diverse neue Funktionen. Der Bereich der angezeigten Werte kann nun frei ausgewählt und stufenlos angepasst werden. Auch die Auflösung der dargestellten Werte kann nun frei ausgewählt werden. Soll der Verbrauch im Laufe des Tages lieber pro Stunde oder pro 15 Minuten angezeigt werden? All das und noch mehr kann unkompliziert eingestellt und betrachtet werden.




### ENERGIE MANAGER

Auch der Energie Manager erstrahlt zur Version 8.1 in neuem Glanze. So werden mit dem Update viel mehr Geräte unterstützt, insbesondere auch Energiespeicher, welche nicht direkt schaltbar sind. Zur Vereinfachung der Konfiguration all dieser Geräte wurden Vorlagen eingeführt. So muss nur noch die passende Vorlage ausgewählt werden und die Konfiguration ist automatisch eingetragen. Und sollte die richtige Vorlage nicht dabei sein, kann die Konfiguration selbstverständlich manuell angepasst werden.
Darüber hinaus wurde die Unterstützung für die rechtlichen Vorgaben aus §9 EEG und §14a EnWG umgesetzt. Schließlich wurde für dynamische Strompreise günstiges Laden implementiert. Hiermit kann ein Energiespeicher oder ein E-Auto automatisch geladen werden, wenn der Strompreis gerade niedrig genug ist. So können die guten Konditionen optimal ausgenutzt werden.




## HOMEMATIC HCU (BETA)

Mit der HCU kann nun auch das neueste HomeMatic Gateway in Symcon verwendet werden. Das Gateway bindet sich wie gewohnt nativ in Symcon ein und bietet die bekannten Komfortfeatures. Es wird per Discovery gefunden und die eingebundenen Geräte und Gruppen können direkt per Konfigurator erstellt und sofort genutzt werden.




### ENTERPRISE FEATURES FÜR ALLE

Im kleinen Rahmen können ausgewählte Enterprise Features nun ganz ohne kostenpflichtige Erweiterung genutzt werden. Das ermöglicht es zum einen die Features vorab zu testen, natürlich auch im Rahmen einer Demo-Version. Oder vielleicht gibt es einfach einen kleinen Anwendungsfall wie ein einzelnes BACnet-Gerät. Oder der Wohnwagen mit eigenem Symcon soll ins Hauptsystem synchronisiert werden? Das ist jetzt für jeden nutzbar.

| ENTERPRISE FEATURE | KOSTENFREIER NUTZUNGSUMFANG |
|---|---|
| Sync Remote | Ein Remote kann integriert werden |
| BACnet | Bis zu 3 BACnet Datenpunkte können integriert oder als BACnet Server bereitgestellt werden |
| OPC UA | Bis zu 3 OPC UA Knoten können integriert werden |
| Translation | Bis zu 10 Strings können je Sprache übersetzt werden |
| RBAC | Die Verbindung zum eigenen LDAP Server kann vorab getestet werden |

## MIGRATION

Bitte erstellen Sie sicherheitshalber ein Backup, sollten wider erwarten Probleme bei der Aktualisierung auftreten.

Falls Sie von einer früheren Version aktualisieren, lesen Sie bitte vorher die passenden Migrationsanleitungen.

Die aktuellste Version kann wie unter Installation beschrieben installiert werden.

## LISTE DER ÄNDERUNGEN

### ALLGEMEIN

Standardmäßig werden Objekte mit leerem Namen statt „Unbenanntes Objekt“ erstellt
Bessere Erkennung von Abbrüchen bei Streams (Upload wurde unnötig zu lange beansprucht wenn die Verbindung über das Relay lief)
Neu: DoorIP: Neuer Befehl DOORIP_Reject zum Ablehnen von Anrufen
Neu: Shutter Control: Berechnung von Laufzeiten mit Millisekunden Präzision
Neu: Spezialschalter ProxySTUN zum Deaktivieren der STUN Anfragen
Neu: Spezialschalter ProxyTURN zum Deaktivieren des TURN Relays
Neu: Eltako FHK14 Instanz kann mit einem Thermokon SR0x kombiniert werden zur komfortableren Einzelraumregelung
Neu: Gitter-Konfiguration einer Visualisierung kann heruntergeladen/hochgeladen werden
Neu: EnOcean: Timeout zwischen zwei Befehlen kann eingestellt werden
Neu: Einlern-Wizard für das Shutter Control
Neu: Wireless M-Bus: Zentrales Schlüssel Management mit Import von diversen CSV Formaten (siehe Wireless M-Bus )
Neu: 15 Minuten Aggregation
Neu: Client Socket unterstützt SSL Verbindungen über Zertifikate
Neu: WebServer/ServerSocket akzeptiert jetzt Private Keys mit Kennwort
Neu: Neue XY-Darstellung wird von KNX DPT 242.600 unterstützt
Neu: Debug Funktion: IPS_DisconnectWebRTCConnections() zum Trennen von Kamera Verbindungen
Neu: Verbessertes Handling bei überschreiten des Connect Traffic Limits (Liste der Top-Verbraucher wird angezeigt, Automatisches Zurücksetzen nach 24 Stunden)
Neu: WebServer Instanz wird standardmäßig mit Port 443 und SSL erstellt
Neu: Links werden initial mit der ID 1 (ungültiges Ziel) erstellt
Fix: Tailscale Modul (Beta) funktioniert wieder korrekt und vergisst die Einstellungen nicht mehr
Fix: Archiv: Bei Daten >2GB in der Variable konnten die Statistiken als negative Werte angezeigt werden
Fix: Wireless M-Bus: WebHook Pfad für WebTech Geräte gekürzt, um Limitation einzuhalten
Fix: Shutter Control: Problem beim Nachführen der Position gelöst
Fix: ModBus: Schreiben von Hex-kodierten
Fix: Energieverteilung: Restriktion auf Int/Float Variablen
Fix: Berechnung vom Connect Verbrauch bei großen Dateien, die fragmentiert übertragen wurden, konnte ggf. zu hoch sein
Fix: Client Socket: Prüfung von Zertifikaten gegen offizielle Domains funktioniert korrekt
Fix: HomeMatic kann nun 0.0.0.0 (Alle) als Ereignisserver auswählen
Fix: Mirroring: Replikation als Standby legt die I/Os korrekt im Standby Status an
Fix: Mirroring: Replikation als Standby blockiert die zyklischen Ereignisse korrekt
Fix: Der Wochenplan konnte in sehr speziellen Fällen einmalig die falsche Aktion ausführen (Danke @bumaas)
Fix: Streams konnten beim Disconnect den Dienst nach und nach blockieren bis der interne WebServer nicht mehr reagiert hat
Fix: Streams zeigen eine Fehlermeldung an, wenn diese blockiert sind anstatt den ganzen

Server lahm zu legen
Fix: GetValueFormatted funktioniert für Float-Variablen mit Intervallen

## GERÄTE

Ab sofort sind für BACnet und OPC UA jeweils 3 Instanzen erstellbar (ohne kostenpflichtige Erweiterung) erstellbar. Bei Sync Remote ist 1 Instanz erstellbar. Beim Translation Control können bis zu 10 Strings pro Sprache zum Testen übersetzt werden.
Neu: KNX: Setzen der physikalischen Adresse der SymBox mit KNX kann direkt im Gateway durchgeführt werden. Optimal in Kombination der ETS Applikation mit Version 1.1 der SymBox
Neu: LCN UMR wird korrekt im Konfigurator erkannt
Neu: Wireless M-Bus kann nun mit dem Webtech SWAN Gateway verwendet werden
Neu: HCU-Integration
- Neu: Discovery Instanz
- Neu: Geräte werden mit Kategorien erstellt
- Neu: Grundlegende Unterstützung für Gruppen
- Neu: Unterstützung für mehr Geräte. Besonders externe
- Neu: Mehr Informationen im Konfigurator
- Neu: Die Gruppen SWITCHING und HEATING werden unterstützt
- Neu: Es werden nur Variablen für Kanäle erstellt, die in Benutzung sind
- Neu: Farbtemperatur für Geräte und Gruppen
- Fix: Korrekte min/max Werte für Temperatur und Farbtemperatur

Neu: MQTT Client kann den verbesserten Client Socket nutzen, um gesicherte Verbindungen über Zertifikate zu verwenden
Neu: ModBus: Eigene Faktoren können in der Konfiguration verwendet werden
Neu: M-Bus: Keep-Alive (alle 3 Minuten) für Gateways
Neu: wM-Bus: Besseres Caching von Compact Frame Formaten
Neu: CMI: Nur die aktivierten Abfragen werden auch wirklich angefragt
Neu: Eltako FUTH
Neu: DMX (Out) over RS485 (Experimentell)
Neu: Alle „alten“ HomeMatic Instanzen wurden intern zu HomeMatic CCU umbenannt
Neu: Eltako FIUS65 wird unterstützt
Neu: Alle Symcon LAN Gateway’s können aus den jeweiligen Discovery Instanzen neu gestartet und die BaudRate (sofern relevant) konfiguriert werden
Fix: BACnet: PresentValue wurde fälschlicherweise als PresetValue geschrieben
Fix: Eltako RGBW: Status vom Weiß Kanal wird korrekt gesetzt
Fix: EnOcean: Fix für EnOcean Shutter Ex Funktionen
Fix: KNX: Bei der Konfiguration der physikalischen Adresse wird die letzte bekannte angezeigt statt der Standardadresse

## ARCHIV

Neu: Spezialschalter ArchiveRecordLimit ist standardmäßig auf 50000 (ist auch der neue Minimalwert)
Fix: Werte Rohansicht von Zähleraggregation korrekt aus

## LOGIKPLÄNE

Neu: Logikpläne können in einem nicht ausführbaren Zustand gespeichert werden
Neu: Das Kopieren/Einfügen zwischen mehreren Plänen auf einem System wird unterstützt
Fix: Copy & Paste für alle Browser

## AKTIONEN

Neu: Aktionen für Darstellungen angepasst
Beinhaltet insbesondere eine komplette Überarbeitung der Gruppenschaltung

## KACHEL VISUALISIERUNG

Diagramme
- Neu: Zoomen durch klicken. Vgl. alte Charts
- Neu: Freie zeitliche Auflösung
- Neu: Erweitere den Namen von Variablen in Diagrammen um übergeordnete Namen falls sie sonst nicht eindeutig sind
- Neu: Auflösung kann vorausgewählt werden
- Fix: Zoomen kann über Esc unterbrochen werden
- Fix: Label von Boolean Charts werden immer angezeigt
- Fix: Verbesserter Fehler, wenn ein Variable nicht existiert
- Fix: Graphen listen bei Anzeige des Intervalls im Titel das Zieldatum nicht mehr mit an, wenn dies nur als Ziel ist (z.B. bei Anzeige von 1.5. 00:00 bis 1.6. 00:00 wird nur 1.5 im Titel angezeigt)
- Fix: Diverse Performance-Verbesserungen
- Fix: Legende kann nicht mehr aus dem Chart heraus geschoben werden
- Fix: Zeige Rohwerte korrekt an, wenn diese vor dem angezeigten Bereich liegen
- Fix: Verhindere Fehler bei Graphen mit Minimum = Maximum und nicht-dynamischer Anzeige
- Fix: Verbesserte automatisierte Bestimmung von Minimum/Maximum/Schrittweite
- Fix: Verhindere Fehler beim Rauszoomen, falls der Startzeitpunkt sich nicht ändert (z.B. bei Stundenansicht von 0:00 bis 1:00 auf täglich)
- Fix: Reicht der Platz nicht für alle Beschriftungen der X-Achse werden gleichmäßig Beschriftungen ausgeblendet
- Fix: Bei Monats- oder Jahresgraphen bleiben die Beschriftungen nun konstant beim ersten Tag
- Fix: Lade mehr Werte vor, damit die Linie zum nächsten Wert immer sichtbar ist
- Fix: Ermögliche wie vorgesehen die Auswahl der Auflösung bei reinen Balkengraphen
- Fix: Balkengraphen mit einem einzelnen Wert zeigen den Balken nun in korrekter Breite an
- Fix: Ermögliche wie vorgesehen die Auswahl der Auflösung bei reinen Balkengraphen
- Fix: Zeige Gradienten bei Variable mit Hintergrundfarbe und Graphen korrekt an
- Fix: Formatiere unglücklich gerundete Zahlen mit Nachkommastellen korrekt
- Fix: Zeige ersten Wert bei CSV-Ansicht wie vorgesehen mit an
- Fix: Fehler beim Laden von Daten werden korrekt dargestellt

Neu: Die Farben der Designs können angepasst, hoch- und heruntergeladen werden
Neu: Energieoptimierer unterstützt Darstellung
Neu: Darstellungen von Energiespeichern ohne Aktion im Energieverbrauch Optimierer

Neu: Neue Icons von @da8ter
Neu: Visualisierung wird standardmäßig über /tile statt /preview geöffnet (/preview leitet weiter auf /tile)
Neu: Option nur das Icon in der Kategorieleiste anzuzeigen
Neu: Titel in der Legacy Ansicht und Objekt-Listen werden hervorgehoben und nicht eingerückt
Neu: xy Farbe kann als kombiniertes Element genutzt werden
Neu: Text innerhalb von Schiebereglern wird dynamisch skaliert
Neu: Text von Wertanzeige wird basieren auf festen Stufen skaliert, um Variationen der Größe zu reduzieren
Neu: Neues Wärmepumpen Icon wird in der Energieverteilung genutzt
Neu: Batterie-Icon der Energieverteilung unterstützt Darstellungen und mehr Profile
Neu: Info-Variable in der Energieverteilung wird zusätzlich im Chart dargestellt
Fix: Unter gewissen Umständen konnte zwischen Hoch- und Querformat das Layout nicht korrekt geladen werden
Fix: Formatierter Wert von Variablen mit leerer/ungültiger Darstellung wie vorgesehen
Fix: Rotation von Lautstärkeregler von Mediaplayer angepasst
Fix: Führende Nullen werden bei der Datumsformatierung berücksichtigt
Fix: Werte von Textfeldern werden nicht mehr überschrieben, wenn sich ein übergeordnetes Element aktualisiert
Fix: Die breite des Shutter-Sliders in der Listenansicht wird korrekt berechnet
Fix: Heruntergeladene Designs werden in den Apps korrekt geöffnet
Fix: Hintergrundfarbe von Dialogen
Fix: Letzter Tag bei monatlichen zyklischen Ereignissen wird korrekt angezeigt
Fix: Längeres Timeout bei Verbindung
Fix: Schriftarten werden wieder korrekt geladen für HTMLBoxen und HTML Kacheln
Fix: Neu erstellte Variablen mit Farbdarstellung funktionieren wie vorgesehen
Fix: Favoriten sind Teil des Browser-Verlaufs
Fix: Verbesserte Fehlerbehandlung für Anfragen an den Server
Fix: Doppelte Darstellung von Kategorien beim ändern der Geräte Orientierung
Fix: Hintergrundbilder von Kategorien, werden nicht mehr angezeigt, nachdem sie verschoben wurden
Fix: Höhe des Titels wird bei kleineren Kacheln angepasst
Fix: Aufzählungen mit einem Element in Gitter-Ansicht funktionieren wieder

## DARSTELLUNGEN

Neu: xy Kodierung für die Farbauswahl mit CIE Diagramm zur Auswahl
    Unterstützt eigene Farbräume
    Unterstützt eigene Farbkurven
Neu: Darstellung Dauer unterstützt Stunden:Minuten
Neu: Min/Max Feld für absolute Wertanzeigen

## WEBFRONT

Fix: Letzter Tag bei monatlichen zyklischen Ereignissen wird korrekt angezeigt

## KONSOLE

Neu: Unterstütze mehr Bildformate für Medienobjekte. Insbesondere webm funktioniert nun, es sollten aber alle üblichen Formate funktionieren
Neu: Import/Export von Ablaufplänen
Neu: Unterstützung für webp bei der Vorschau von Medien (Bild)
Neu: wM-Bus: Link Mode kann eingestellt/eingeschränkt werden
Neu: Das Kopieren/Einfügen zwischen mehreren Objektbäumen auf einem System wird unterstützt
Fix: Variablen mit der Darstellung Dauer werden korrekt formatiert und zählen bei Bedarf dauerhaft hoch bzw. runter
Fix: Kleine Performance-Verbesserung bei Aktualisierung von Variablen
Fix: Korrekte Formatierung von Variablen ohne Darstellung oder mit ungültiger Darstellung
Fix: Tree's merken sich beim Aktualisieren wieder die bereits ausgeklappten Knoten
Fix: M-Bus/wM-Bus: Profile (insbes. Icons) für Wasser Geräte korrigiert
Fix: Darstellungsmanager funktioniert wieder
Fix: Beim Versuch die Dauer-Darstellung zu aktualisieren verursachen Links ohne Ziel keine Fehler mehr
Fix: Zeige die Vorschau von PNG-Bildern wie vorgesehen an
Fix: Unterstütze Darstellungen im TestCenter

### SYMOS

Neu: Reconnect Timer läuft nicht mehr nur alle 30 Minuten, sondern wie das Connect Control in IP-Symcon alle 2^Fehlversuche Minuten und 30% Varianz beim Reconnect. Damit sollten anfängliche Problem mit der Internetverbindung schneller gelöst sein.

### PHP-SDK

Neu: Feld "validPresentations" für SelectVariable zur Einschränkung auf bestimmte Darstellungen
Neu: Dokumentation für IPSModuleStrict
Neu: IPSModuleStrict nutzt jetzt die neue GetCompatibleParent Funktion.
Neu: IPSModuleStrict wird zu IP-Symcon 8.1 auch im Module Store offiziell unterstützt
Fix: Sofern keine weitere Property/Attribut hinzugefügt/gelöscht wurde, wurde die Migration in PHP-Modulen nicht korrekt ausgeführt
Fix: Type Hints für IPSModuleStrict verbessert

## AKTUALISIERTE MODULE

### AKTIVLISTE

Neu: Skript zum Deaktivieren aller aktiven Geräte
Neu: Buttons für einfacheres Umbenennen

### ALARMIERUNG

Neu: Nacht-Alarm

## AMAZON ALEXA

Neu: Beschreibendere Namen in den Tabellen
Neu: Unterstützung von neuen Darstellungen
Neu: Zeige mehr Zeilen pro Gerätetyp an
Fix: Korrektes Handling mit PHP 8.2

## BACKUP

Neu: Bessere Unterstützung für große Backups

## BENACHRICHTIGUNG

Neu: Unterstützung für neue Visualisierung

## CSV ZIP EXPORT

Neu: Option zur Auswahl des Dezimaltrennzeichens

## EKEY BIONYX

Neu: Verbesserungen zu den WebHooks
Neu: Unterstützung für die Notification API

## ENERGIE MANAGER

Neu: Vorlagen für eine einfache Konfiguration
Neu: Unterstütze Einschränkung der Energieproduktion entsprechend §9 EEG
Neu: Deaktiviere Geräte, die auch nach mindestens fünf Minuten und zehn Updates nicht reagieren
Neu: Günstiges Laden/Günstige Laufzeit
New: Neue Variable über den Betriebsmodus von Energiespeichern (wird von der Visualisierung verwendet)
Neu: Unterstützung von Statusvariablen für Energiespeicher
Neu: Unterstützung von zwei Variablen im relativen Modus: Eine für Überschuss und eine für Defizit
Neu: Unterstütze eine Steuerung der Entladung von Energiespeichern
Neu: Unterstütze prozentuale Ladeziele für Elektrofahrzeuge
Neu: Möglichkeit den tatsächlichen Verbrauch für jeden Gerätetyp anzugeben
Neu: Unterstütze das indirekte Schalten von Energiespeichern ohne Aktion
Neu: Unterstütze Geräte mit nicht-Legacy Darstellungen
Neu: Unterstütze Energiespeicher, die eine einzelne Variable für Laden und Entladen verwenden
Neu: Verbessertes Formular für Energiespeicher
Neu: Leistung für E-Fahrzeuge kann pro Phase statt insgesamt geschaltet werden
Neu: Betrachte entladende Batterien als negativen Überschuss
Neu: Unterstütze Restriktionen nach §14a

Neu: Betrachte den Verbrauch von manuell gesteuerten Geräten im absoluten Modus
Neu: Energiepreise sind optional für Nachtladen
Neu: Optionale Variablen für An/Aus und Phasenwechsel von E-Fahrzeugen
Neu: Energieeinheit für E-Fahrzeuge auswählbar: Watt oder Ampere
Neu: Aktiviere Geräte mit niedrigerer Priorität wenn die Energie nicht für höhere Priorität ausreicht
Neu: Nächtliches Laden/Laufzeit
Neu: Gerätetyp Energiespeicher
Neu: Unterstützung von Mindestlaufzeit und Nachlaufzeit
Fix: Verhindere ein "Zittern" in bestimmten Konstellationen
Fix: Aktualisiere Modus, falls sich der Gerätestatus ändert und der Modus nicht automatisch ist
Fix: Dimme gesperrte Geräte korrekt wenn keine Energie vorhanden ist
Fix: Nachtladen konnte in manchen Konstellationen die falsche Zeit wählen
Fix: Funktioniere weiterhin wenn das Format der Variable für Energiepreise ungültig ist
Fix: "Batterie" zu "Energiespeicher" umbenannt

## LINGG&JANKE KNX QUICK

Neu: Verbesserte Darstellung als Liste

## SZENEN-STEUERUNG

Neu: Erlaube das Editieren von Werten in der Konsole

## TREPPENHAUSLICHTSTEUERUNG

Neu: Aktion beim Deaktivieren kann ausgewählt werden

## WATCHDOG

Neu: Spalte zum Aktiv schalten der Aktoren eingefügt

# V7.2->V8.0 (Q1/2025)

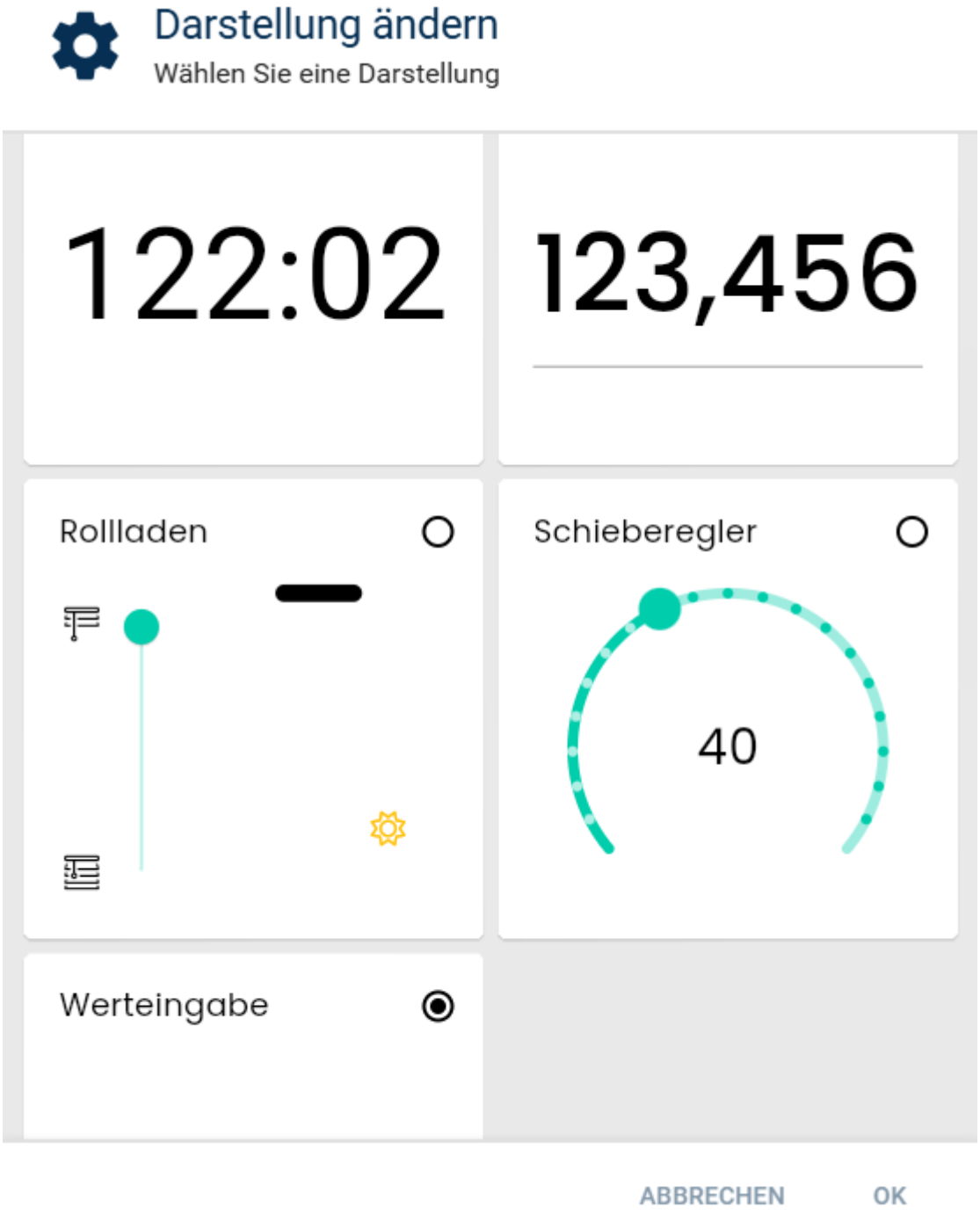
Wesentliche Änderungen
Migration
Liste der Änderungen
Neue und aktualisierte Module

## WESENTLICHE ÄNDERUNGEN

Darstellungen
Türsprechanlagen
Wireless M-Bus
Benutzerverwaltung

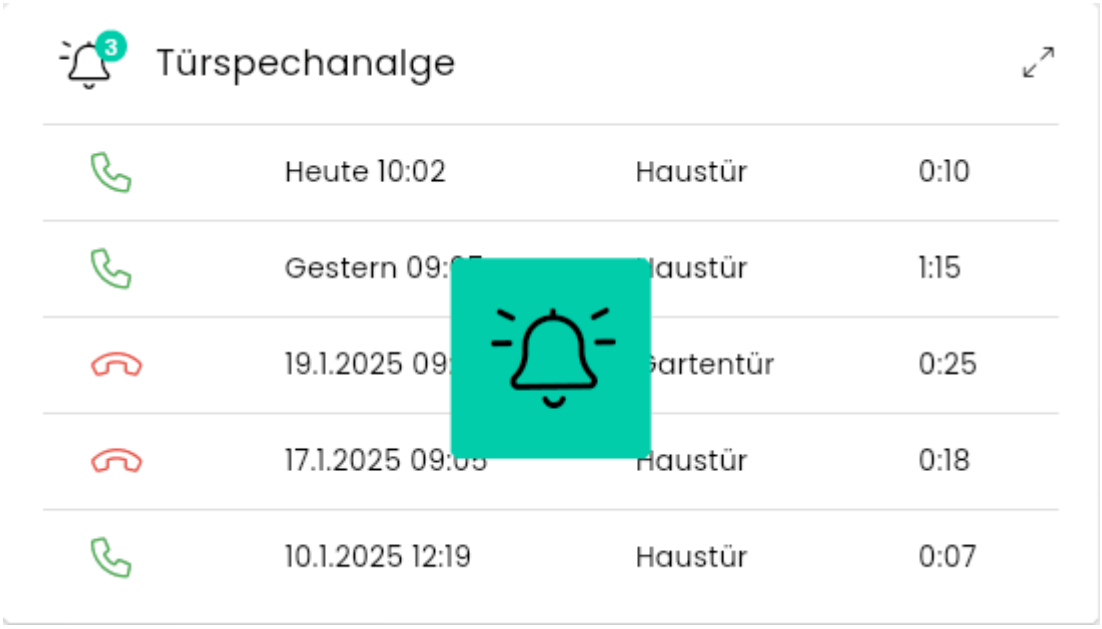
### DARSTELLUNGEN

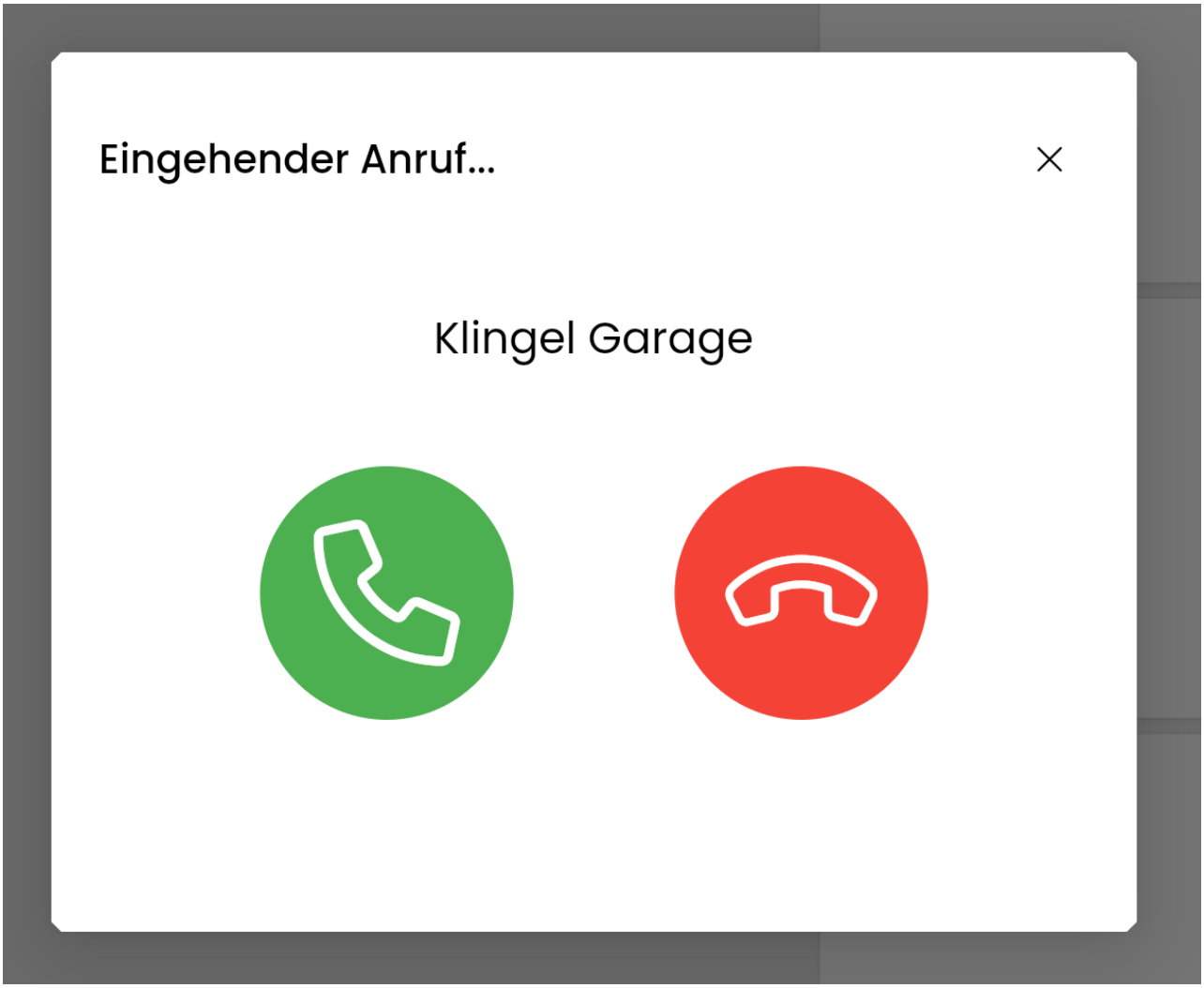
Darstellungen bieten eine neue Dimension bei der Konfiguration von Variablen. Nach der Wahl eines Darstellungstyps wie "Schieberegler", "Farbe" oder "Dauer", kann dieser individuell mit Hilfe von verschiedenen Parametern konfiguriert werden. Diese neue Struktur ermöglicht eine Vielzahl neuer Möglichkeiten für Variablen um diese in der Visualisierung zu verwenden.




## TÜRSPRECHANLAGEN

Durch die Einbindung von Türsprechanlagen können diese nahtlos in der Visualisierung verwendet werden. Bei einem Klingeln kann direkt in einem Dialog das Bild der Kamera betrachtet und mit dem Besucher geredet werden um schließlich die Tür zu öffnen.



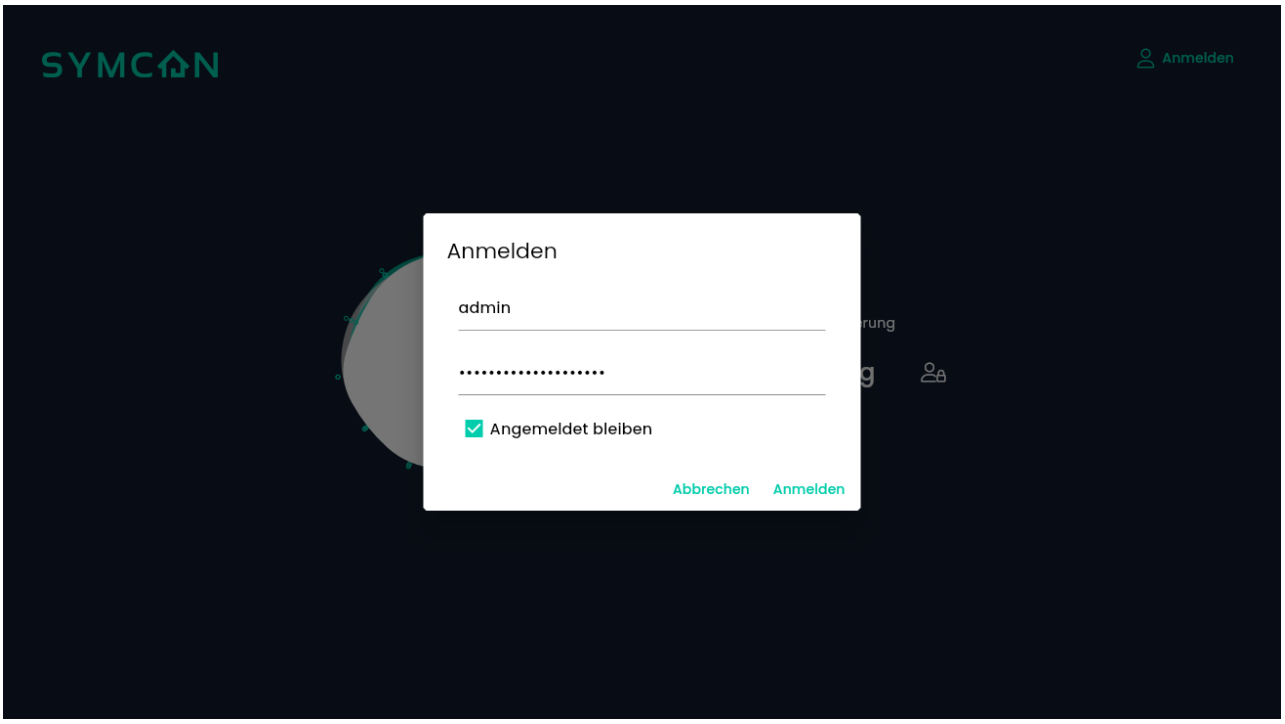

**WIRELESS M-BUS**

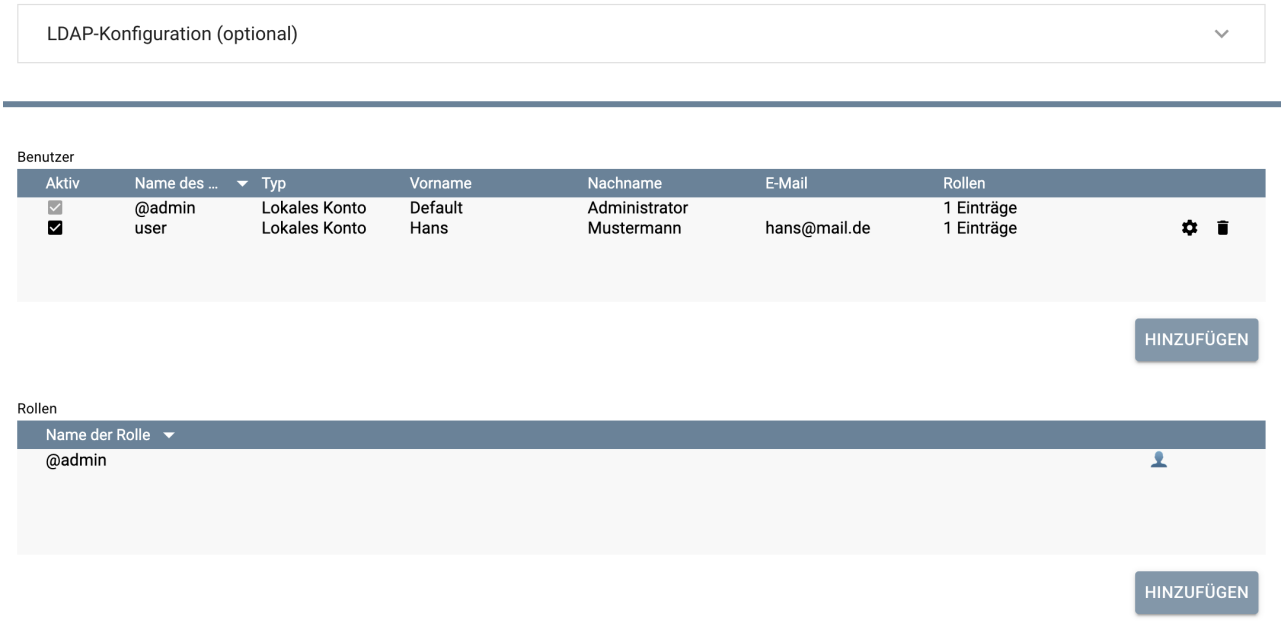
Zusätzlich zu den drahtgebundenen M-Bus-Geräten sind mit der Version 8.0 die kabellosen Geräte von Wireless M-Bus nutzbar. Diese können über die gängigen Gateways von WE und IMST angesteuert werden.

| Name | ID | Hersteller | Version | Medium | Status | Verschlüsselu... | RSSI | Daten | InstanzID | |
|---|---|---|---|---|---|---|---|---|---|---|
| Wireless M-Bus Device (02355964, EFE, Heat) | 02355964 | EFE | 0 | Heat | 0 | | N/A | | #35671 | |
| Wireless M-Bus Device (42069120, DME, Gas) | 42069120 | DME | 0 | Gas | 179 | | N/A | | #21970 | |
| Wireless M-Bus Device (20119827, DWZ, Water) | 20119827 | DWZ | 2 | Water | 0 | OMS (Mode 5) | -61 dbm | | Kein(e) | |
| Wireless M-Bus Device (20844457, KAM, Cold Water) | 20844457 | KAM | 29 | Cold Water | 16 | BSI (Mode 7) | -56 dbm | | Kein(e) | |
| Wireless M-Bus Device (23160294, ITW, Heat) | 23160294 | ITW | 0 | Heat | 19 | OMS (Mode 5) | -52 dbm | | Kein(e) | |
| Wireless M-Bus Device (43152149, EFE, Heat) | 43152149 | EFE | 0 | Heat | 0 | OMS (Mode 5) | -86 dbm | | Kein(e) | |
| Wireless M-Bus Device (43152150, EFE, Heat) | 43152150 | EFE | 0 | Heat | 0 | OMS (Mode 5) | -86 dbm | | Kein(e) | |
| Wireless M-Bus Device (43152155, EFE, Heat) | 43152155 | EFE | 0 | Heat | 0 | OMS (Mode 5) | -80 dbm | | Kein(e) | |
| Wireless M-Bus Device (43152157, EFE, Heat) | 43152157 | EFE | 0 | Heat | 0 | OMS (Mode 5) | -87 dbm | | Kein(e) | |
| Wireless M-Bus Device (43152159, EFE, Heat) | 43152159 | EFE | 0 | Heat | 0 | OMS (Mode 5) | -85 dbm | | Kein(e) | |
| Wireless M-Bus Device (43152163, EFE, Heat) | 43152163 | EFE | 0 | Heat | 0 | OMS (Mode 5) | -81 dbm | | Kein(e) | |
| Wireless M-Bus Device (43152169, EFE, Heat) | 43152169 | EFE | 0 | Heat | 0 | OMS (Mode 5) | -76 dbm | | Kein(e) | |
| Wireless M-Bus Device (43152170, EFE, Heat) | 43152170 | EFE | 0 | Heat | 0 | OMS (Mode 5) | -90 dbm | | Kein(e) | |
| Wireless M-Bus Device (43152172, EFE, Heat) | 43152172 | EFE | 0 | Heat | 0 | OMS (Mode 5) | -88 dbm | | Kein(e) | |
| Wireless M-Bus Device (43152173, EFE, Heat) | 43152173 | EFE | 0 | Heat | 0 | OMS (Mode 5) | -82 dbm | | Kein(e) | |
| Wireless M-Bus Device (43152175, EFE, Heat) | 43152175 | EFE | 0 | Heat | 0 | OMS (Mode 5) | -83 dbm | | Kein(e) | |
| Wireless M-Bus Device (43152177, EFE, Heat) | 43152177 | EFE | 0 | Heat | 0 | OMS (Mode 5) | -77 dbm | | Kein(e) | |
| Wireless M-Bus Device (43152178, EFE, Heat) | 43152178 | EFE | 0 | Heat | 0 | OMS (Mode 5) | -73 dbm | | Kein(e) | |
| Wireless M-Bus Device (43152185, EFE, Heat) | 43152185 | EFE | 0 | Heat | 0 | OMS (Mode 5) | -85 dbm | | Kein(e) | |
| Wireless M-Bus Device (43152186, EFE, Heat) | 43152186 | EFE | 0 | Heat | 0 | OMS (Mode 5) | -79 dbm | | Kein(e) | |
| Wireless M-Bus Device (71927603, DME, Water) | 71927603 | DME | 118 | Water | 112 | BSI (Mode 7) | -46 dbm | | Kein(e) | |
| Wireless M-Bus Device (76102435, DME, Heat) | 76102435 | DME | 64 | Heat | 112 | BSI (Mode 7) | -45 dbm | | Kein(e) | |

## BENUTZERVERWALTUNG

Für Enterprise-Kunden bietet die neue Version eine Benutzerverwaltung (RBAC). Hier können verschiedene Benutzer angelegt und in beliebige Benutzergruppen kategorisiert werden. Visualisierungen können für Benutzergruppen oder einzelne Benutzer freigegeben werden. Benutzerkonten können entweder direkt in IP-Symcon angelegt werden oder per LDAP importiert werden.






## MIGRATION

Bitte erstellen Sie sicherheitshalber ein Backup, sollten wider erwarten Probleme bei der Aktualisierung auftreten.

Falls Sie von einer früheren Version aktualisieren, lesen Sie bitte vorher die passenden Migrationsanleitungen.

Die aktuellste Version kann wie unter Installation beschrieben installiert werden.

## LISTE DER ÄNDERUNGEN

### ALLGEMEIN

Neu: PHP 8.3
Neu: VoIP: Lokale Nummer, die Angerufen wird in der Connection „Local“
Neu: Enterprise Funktion für Benutzerverwaltung und LDAP Synchronisation
Neu: Connect Control: Neue Buttons, um direkt die Visu, Konsole oder SymOS zu öffnen
Neu: Limit für Anzahl der Aktionen im Wochenplant entfernt
Neu: Limit für Anzahl von Assoziationen entfernt
Neu: Schalter in der Instanz der Kachelvisualisierung um die neuen Diagramme zu aktivieren
Neu: Für Logins über das RBAC/LDAP Feature, kann über den Spezialschalter „ServerTokenExpiration“ die Zeit bis zum Ablaufen des Token und einem erzwingen zum erneuten Login festgelegt werden.
Neu: GetValueFormatted für Darstellungen
Neu: ProxyTURN Spezialschalter zum Deaktivieren vom Streams Relay (bei Problemen mit

zu hohem Upload)
Neu: ProxySTUN Spezialschalter, sofern die Streams nur lokal genutzt werden sollen und keine Kommunikation mit unserem STUN Server gewünscht ist.
Fix: DoorIP - Fehlende Push-Nachrichten
Fix: Notification Control: Sortierung nach Updated funktioniert nun wie erwartet
Fix: UDP Socket konnte stecken bleiben wenn der Bind beim Senden einen Fehler geworfen hat
Fix: Remote Sync: Problem mit Fehlermeldung „Endlosschleife“ korrigiert
Fix: Wird ein Profil auf einen leeren String gesetzt, wird die dazugehörige Darstellung gelöscht
Fix: DMX Output zu DMX Kanal umbenannt, da dieser Mittlerweile auch als DMX Input genutzt werden kann
Fix: Darstellungen mit Intervallen werden im WebFront korrekt angezeigt
Fix: Sync Remote: Probleme beim Löschen von Variablen gelöst
Fix: Fehler in Verbindung mit älteren Versionien der Energieverteilung behoben
Fix: WebSocket: WC_* Funktionen funktionieren korrekt, wenn Leerzeichen/Umlaute im Pfad enthalten sind
Fix: Neue Icons können für Push Nachrichten korrekt verwendet werden mit längeren Namen
Fix: Bei neuen SymBox’en wird korrekt die neue Visualisierung als Standard gewählt
Fix: Sicherheitslücke unter Windows im WebServer (Die Lücke betrifft nur die Windows Version und kann nicht über Symcon Connect ausgenutzt werden. Vielen Dank an @Dustin723 für’s finden und melden!)
Fix: Sync Remote: AC_* Funktionen konnten vom Master aus zum Remote nicht korrekt in PHP genutzt werden
Fix: Float-Werte werden in Regeln von Ereignissen korrekt verglichen
Fix: Rollladensteuerung: Bei manuellem Richtigungswechsel (und Empfang vom STOP) konnte wurde ein interner Stop Timer nicht beendet, wodurch der Rollladen bei der manuellen Fahrt ggf. falsch gestoppt wurde
Fix: Kalender setzt die Variablen wieder korrekt, wenn der Endzeitpunkt erst am nächsten Tag ist
Fix: $_IPS[‚TARGET‘] wurde bei TimerEvents nicht mehr korrekt gesetzt (vmtl. bereits seit der 7.0 defekt)

## GERÄTE

Neu: Wireless M-Bus unterstützt IMST iM871A-USB und IMST iM891A-XL/iU891A-XL
Neu: EnOcean: Neuer RCM100 Modul für Rollladen
Neu: Wireless M-Bus Unterstützung
Neu: KNX: Konfigurator erlaubt das Hinzufügen von Keywords, die Entfernt werden sollen
Neu: KNX: Für die neuen KNX Geräte-Instanzen kann je GA ein optionaler Name vorgegeben werden
Neu: KNX: Konfigurator erlaubt den Export der genutzten GAs (Kann im „Secure Interface“ der SymBox in der ETS ab der 6.3 eingefügt werden)
Neu: EnOcean: Für den Modus FGW14 kann die BaseID angegeben werden
Neu: EnOcean: Beim Eltako Switch kann als Schaltmodus auch A5-38-08 verwendet werden
Neu: Charts werden jetzt immer im Vollbild angezeigt
Neu: EnOcean: Wartezeit bis Timeout kann eingestellt werden
Neu: EnOcean: Neue Instanz für den Eltako FWRB
Neu: M-Bus: Timeouts können im Gateway eingestellt werden (Sinnvoll z.B. bei 300 Baud)

Neu: wMBus: Wireless M-Bus Konfigurator zeigt Informationen besser an
Neu: EnOcean: Option, um beim Eltako Dimmer bei An auf die letzte Dimstufe anzuschalten
Neu: KNX: Physikalische Adresse der SymBox KNX Schnittstelle kann direkt aus der Splitter Instanz gesetzt werden. (Notwendig für die ETS Applikation 1.1 der SymBox)
Neu: Eltako FHK14 Instanz kann mit einem Thermokon SR0x kombiniert werden zur komfortableren Einzelraumregelung
Neu: EnOcean: Timeout zwischen zwei Befehlen kann eingestellt werden
Fix: Assoziationen mit %f /%d funktionieren wieder wie vorgesehen
Fix: Skripteditor unter Firefox 131.0.2 geht wieder
Fix: KNX: Einige TTL Dateien konnte nicht gelesen werden mit mit einem „stoi“ Fehler
Fix: KNX: Beim TTL Export wurden bereits erkannte Status GA nicht korrekt aus der Liste gefiltert
Fix: KNX: Beim TTL Export wurde der „unbekannt“ Tag nicht korrekt dargestellt
Fix: KNX: DPT 1.024 und 1.100 haben jetzt korrekte Profile (waren seit der 7.2 vertauscht)
Fix: OPCUA: Variablen werden wie vorgesehen nicht entfernt, falls der Datentyp vom Datenpunkt sich ändert
Fix: HomeMatic: Beim Trennen der Verbindung wurde ein falsches Paket gesendet
Fix: EnOcean: Korrigiere Profil für D5-00-01 (.Reversed fehlte)
Fix: OPC UA: Sofern der Server den Wert „Null“ gesendet hat, wurde die Variable gelöscht und dann, beim Empfang des korrekten Werts, mit anderer ID neu erstellt. IP-Symcon verhindert dies nun und loggt das Problem ins Meldungsfenster.
Fix: KNX: Der SubTag Lamelle wurde nicht korrekt ausgewertet bei neuen KNX Geräten
Fix: wMBus: Bei Mode 7 mit Informationen in der TPL wurde ggf. nicht korrekt entschlüsselt
Fix: MQTT Client: Retain Flag wurde nicht korrekt weitergeleitet
Fix: ModBus: Schreiben von Hex-kodierten Strings korrigiert
Fix: EnOcean: FHK14 Thermostat Schwellwert verbessert

## KACHEL VISUALISIERUNG

Neu: Neue Graphen können verwendet werden
Neu: Darstellung von Nachtladen im Energieverbrauchoptimierer
Neu: Prognose für Energieoptimierer wird unterstützt
Neu: In den Servereinstellung werden verbundene Server/Visu hervorgehoben
Neu: Im Server Info Dialog wird der Verbindungstyp angezeigt
Neu: Annehmen von Anrufen direkt über die DoorIP-Instanz möglich
Neu: Nächtliches Laden für Batterien im Energieoptimierer
Neu: Beim Schalten von Variablen wird der neue Wert sofort angewandt und ggfs. nach kurzer Zeit zurückgesetzt, falls er vom Server nicht bestätigt wird. Dadurch fühlen sich diverse Bedienvorgänge flüssiger an.
Neu: Neue Typen Kälte und Klimaanlage für Energieverteilung
Neu: Hinweis und korrektes Handling, falls sich der Server in einer anderen Zeitzone befindet
Neu: Nächtliches Laden für Verbraucher im Energieverbrauch-Optimierer
Neu: eigenes Format für Uhrzeitdarstellung in der Appbar
Neu: Energieoptimierer
Neu: Nicht-stören-Modus für Door IP
Neu: LDAP: „angemeldet bleiben“ Option im Web
Neu: Ansicht von Aufzählung als Dropdown, Aufzählung nebeneinander und Wertedarstellung vereinheitlicht

Neu: Halloween Theme
Neu: Animation in der Energieverteilung sind wieder da.
Neu: Dem HTML-SDK und den HTML-Boxen stehen jetzt verschiedene Farben in Form von CSS-Variablen zur Verfügung (–accent-color, --content-color, --card-color). Die Variablen sind einfach über
var(--accent-color) abrufbar.
Neu: Warnung beim Löschen eines verbundenen Servers
Neu: Bei neuen Graphen werden wie ursprünglich die Namen der Boolean-Graphen eingeblendet
Neu: Die neuen Charts aktualisieren sich nach 2 Minuten, wenn der aktuelle Zeitpunkt angezeigt wird
Neu: Generator Typ für Energieverteilung
Neu: Einstellbares Format für den angezeigten Namen des Inhaltswechslers
Neu: Instanzkonfiguration
    Die Kategorieleiste kann in einer Zeile dargestellt werden.
    Option Kategorien nur in der Seitenleiste anzuzeigen
Neu: In den Apps werden Kacheln im Editiermodus über ein langes drücken verschoben
Neu: Nutze beim Kopieren der Versionsinformation ein formatiertes Datum anstatt eines Zeitstempels
Neu: Ladeanzeige beim Laden von neuen Werten für Graphen
Neu: Option für statische Begrüßung
Fix: Edit-Button wird auch in der Kachel ausgeblendet, wenn der entsprechende Schalter in der Visualisierungsinstanz deaktiviert wird
Fix: Löschen eins verbundenen Servers funktioniert wie vorgesehen
Fix: Der Schieberegler kann bei bestimmten Schrittweiten nicht mehr über seine Grenzwerte hinaus geschoben werden.
Fix: Sekundäre Variable in der Energieverteilung ist jetzt immer verfügbar
Fix: Bedienelemente für Aufzählungen überarbeitet
Fix: Breite von Slidern in Listenansicht vereinheitlicht
Fix: Komma als Dezimaltrennzeichen für Profile
Fix: LDAP: Fehler, wenn es nur Visualisierungen gibt, die nur für autorisierte Benutzer angezeigt werden.
Fix: LDAP: Drawer: Visualisierungen die nur für autorisierte Benutzer sind werden auch nur für diese angezeigt.
Fix: LDAP: Kein automatisches Öffnen, wenn eine versteckte Visualisierung mit Autorisierung vorhanden ist
Fix: Stringvariablen werden aktualisiert, wenn das Eingabefeld abgewählt wird
Fix: Größe von Textfelder in der Listendarstellung verringert, damit es besser in die Zeile passt
Fix: Fehler in Verbindung mit dem Passwortdialog behoben
Fix: Endlosschleife beim Verbinden mit einzelnem Server behoben
Fix: Liste gespeicherter Server wird richtig aktualisiert
Fix: Nicht-Geräte Instanzen können nur noch über Links angezeigt werden
Fix: Zuordnung von Fehlermeldungen verbessert
Fix: Korrigierte Ausrichtung von TextBoxen im Vollbild-Modus
Fix: Ist eine Visualisierung nur im lokalen Netz verfügbar, wird sie bei einem Zugriff über Connect als gesperrt dargestellt
Fix: Korrekte Prozentberechnung in der Energieverteilung, bei Prozentprofilen, die nicht von 0 bis 100 laufen
Fix: Events werden aus Dialogen heraus richtig aktualisiert
Fix: Erlaube Visu-Wechsel bei gelöschter Hauptkategorie

Fix: Bei Kategorie Kacheln wird nur ein Icon angezeigt
Fix: Icons werden beim Konfigurieren von Instanzen korrekt in der Titelzeile angezeigt
Fix: Größe und Position von Icons der Automationen angepasst
Fix: Alle Kombinierte Elemente werden als deaktiviert angezeigt, wenn sie deaktiviert sind
Fix: Objekte werden als deaktiviert angezeigt, wenn übergeordnete Objekte deaktiviert sind
Fix: Beim Konfigurieren von Instanzen wird die Sichtbarkeit des tatsächlichen Objektes genutzt, nicht die des Ursprungsobjektes
Fix: Icons dürfen jetzt 64 Zeichen lang sein für Push Nachrichten
Fix: Wird das Bild einer Kategorie aktualisiert, so wird die Kachel direkt aktualisiert
Fix: Variablen mit ~UnixTimestamp Profil können wieder verändert werden
Fix: Lesbarkeit des Zeitauswahldialogs im Darkmode verbessert
Fix: Graphen können nun aus der Energieverteilung geöffnet werden, falls der Button über HTML-Inhalt schwebt
Fix: Variablen mit HTML-Inhalt werden auch in der App neu geladen, falls der Variableninhalt aktualisiert, aber nicht geändert wird
Fix: Gehe in einen Fehlerzustand, wenn die Übergabe zu einer neuen Verbindung nicht funktioniert, anstatt bei „Verbinde…“ festzuhängen
Fix: Wird ein Reconnect durchgeführt warten Anfragen an den Server bis dieser abgeschlossen ist
Fix: Versuche bei einer Testnachricht ohne dazugehörige Visualisierung nicht eine nicht-existente Visualisierung zu öffnen
Fix: Skalierung in HTML-Elementen, die bei einigen mobilen Geräten falsch war, sollte nun korrekt funktionieren
Fix: Fange seltenen Absturz bei SVG-Graphen ab
Fix: Ehemalige Heuristik für HTML-Boxen wird weiterhin bei Zugriff auf ältere Symcon-Systeme angewandt
Fix: Datumsdarstellungen in Listenansicht nutzen wie vorgesehen nur den vorhandenen Platz
Fix: Lade HTML-Inhalte bei maximierten Kacheln nicht beim Ein- oder Ausblenden einer Tastatur neu
Fix: Fehler bei maximierter Thermostat Instanz behoben
Fix: Größe des Popups der Energieverteilung skaliert mit dem Inhalt
Fix: Die Zeit kann bei einmaligen Ereignissen wie vorgesehen ausgewählt werden
Fix: Datum/Uhrzeit Dialoge von Ereignissen nutzen den gleichen Stil, wie alle anderen
Fix: Fehler im HTML-SDK werden an den Benutzer weitergeleitet
Fix: Zurücksetzten des Kachel-Layouts bei fehlerhafter Gitterkonfiguration
Fix: Doorip - Dialog wird angezeigt, wenn bei geöffneter App auf eine Push-Benachrichtigung getippt wird
Fix: Energieoptimierer: Bedingungstext wird bei wenig Platz in eine zweite Zeile geschoben
Fix: Energieverteilung: geöffnete Dialoge werden nicht mehr am Bildschirmrand abgeschnitten
Fix: Buttons bei maximiertem Chart werden jetzt immer korrekt dargestellt
Fix: Chart Daten werden initial besser angezeigt
Fix: Sehr ähnliche Werte werden im Chart besser dargestellt
Fix: Slider wird nie ohne Label angezeigt
Fix: Hintergrund-Gradient wird bei maximierten Charts nicht mehr angezeigt
Fix: Im Wochenplan können Zeiträume < 30 Minuten ausgewählt werden
Fix: Rundungsfehler bei negativen Werden bei Schiebereglern
Fix: Uhrzeit wird bei dem Wert 0 korrekt dargestellt
Fix: Nicht existierendes Profil bei Variablen ohne Aktion führt nicht mehr zum Absturz

Fix: Versteckte Visualisierungen konnten IP-Autostart nicht verwenden
Fix: Geloggte String-Variablen nutzen immer die bestehende Tabellen-Darstellung
Fix: Die Einstellungen von verlinkten Inhaltswechsler werden wie vorgesehen übernommen
Fix: HTMLBox'en werden bei Variablenänderung korrekt aktualisiert
Fix: Overlay der Energieverteilung funktioniert korrekt über HTMLBoxen
Fix: Nicht existierende Objekte in den Favoriten führen nicht mehr zu Fehlern
Fix: Das Löschen von leeren Bereichen im Editiermodus funktioniert wie vorgesehen
Fix: Schieberegler werden hervorgehoben, wenn ihre Farbe dem Hintergrund entspricht
Fix: Streams verbinden sich nach einem Verbindungsabbruch neu
Fix: Die Rollladen Darstellung erkennt jetzt die neuen Variablendarstellungen
Fix: Scrollen in der Ausgabe von Skripten möglich
Fix: Verbesserte Logik für die Wiederverbindung von Streams
Fix: Punktbewegung der Energieverteilung aktualisiert sich wieder korrekt
Fix: URLs von hinzugefügten Servern werden aktualisiert
Fix: Bei mehreren Werten zum gleichen Zeitpunkt werden diese im neuen Graphen korrekt sortiert
Fix: Aktualisiere bei Verbindung via URL die Serverinformationen und verhindere so doppelte Einträge in der Serverliste oder unpassende Namen und Versionen
Fix: Hinweis, wenn die Stream Verbindung über das Relay stattfindet, wird wieder angezeigt
Fix: Kennwörter, welche UTF-8 Zeichen enthalten funktionieren wieder korrekt

## DARSTELLUNGEN

Neu: Darstellungen sind nun nutzbar.
Neu: Legacy für die bisherigen Profile
Neu: Schieberegler
    Tausendertrennzeichen
    Intervalle
Neu: Shutter
    Zusätzliche Parameter für die Lamellenposition und Sonnenposition
Neu: Datum/Uhrzeit
    Datum und Uhrzeit können separat angezeigt werden
    Das Darstellungsformat kann angepasst werden
Neu: Dauer
    Dauer bis/seit einem Zeitpunkt
    Aktuell auf Stunden begrenzt
Neu: Darstellung Aufzählung
    Verschiedene Anordnungen: Spalte, Reihe, Liste
Neu: Darstellung Schalter
    Eigene Icons für Werte
    Anpassungen des Leuchtens
Neu: Werteingabe
    Präfix/Suffix
Neu: Wertanzeige
    Tausendertrennzeichen
    Intervalle
Neu: Webinhalt ( ehemals ~HTMLBox)
    Direktes einbinden einer Webseite
Neu: Farbe

Verschiedene Kodierungen: RGB, CMYK, HSV, HSL
Neu: Icons sind bei nicht Legacy-Darstellungen Teil der Darstellung
Neu: Vorlagen für Wertanzeige uns Schieberegler

## WEBFRONT

Neu: Darstellungen werden im WebFront und den mobilen Apps angezeigt
Fix: Verhindere doppelte Präfix/Suffix in konvertierten Wertanzeigen

## KONSOLE

Neu: Hinweis beim versuch eine Vorlage zu wählen, wenn keine existiert
Neu: Funktion zum Umwandeln von Profilen in neue Darstellungen
Neu: Tooltips für zu lange Icon-Namen
Neu: Vereinheitliche Dialoge für Referenzprüfung, verbessert insbesondere den Referenzdialog im Module Store
Neu: Bei neu erstellten Variablen wird standardmäßig eine passende Nicht-Legacy-Darstellung ausgewählt
Neu: Speichern von Vorlagen direkt aus dem Variablen-
Neu: Passe Anzahl der Elemente pro Zeile bei der Auswahl von Darstellung oder Vorlage an den verfügbaren Platz an
Fix: Ungültige Vorlagen können gelöscht werden
Fix: Beim Erstellen von Vorlagen kann, falls benötigt, die passende Gruppe ausgewählt werden
Fix: Benutzerdefinierte Darstellung von Statusvariablen kann gelöscht werden
Fix: Benutzerdefinierte Darstellungen wird wie erwartet bevorzugt, wenn vorhanden
Fix: Archivdialoge können nun auch bei sehr langen Variablennamen problemlos geschlossen werden
Fix: Größe der Kachel kann im Vollbild der Vorschau der Darstellungen wieder verändert werden
Fix: kosmetische Fehlermeldung beim öffnen des Variablen bearbeiten Dialogs verhindert
Fix: Aus dem Meldungsfenster heraus geöffneter Wert bearbeiten Dialog funktioniert wie vorgesehen
Fix: Mehrfachauswahl in Konfiguratoren funktioniert wieder wie vorgesehen
Fix: Übersetzung von Icon Namen verbessert
Fix: Fehler beim öffnen des Profil Dialogs
Fix: Objektbaum wird nach dem ändern einer Darstellung aktualisiert
Fix: Verbessertes Bearbeiten der Darstellung von mehreren Variablen
Fix: Deaktivierte Standardaktionen werden von Darstellungen berücksichtigt
Fix: Fehlerhafte Ketten Verbindungen
Fix: Fehlerhafte Erkennung ob eine Variable eine Aktion hat
Fix: Reaggregations-Dialog wird nur noch angezeigt, wenn benötigt
Fix: Gruppen von Vorlagen werden korrekt ausgewertet
Fix: Profile werden beim ändern von Darstellungen aktualisiert
Fix: Möglichen Fehler beim Speichern/Laden von Vorlagen behoben
Fix: Wende Konvertierung von Darstellungen bei der Wertdarstellung korrekt an, sodass keine Nachkommastellen bei Integer-Werten angezeigt werden
Fix: Vorlagenauswahl kann bei unglücklichem Timing nicht mehr die Auswahlkringel verhindern

Fix: Verhindere Fehler beim Versuch in einem Archiv mit sehr vielen Variablen aufzuräumen
Fix: Verdichtung kann wieder korrekt konfiguriert werden
Fix: Variablen mit der Darstellung Dauer werden korrekt formatiert und zählen bei Bedarf dauerhaft hoch bzw. runter

### SYMOS

Fix: Aktivierung von Lizenzen (über Key) funktioniert über einen neuen Endpunkt ***

### PHP-SDK

Neu: Neue Bedingung „hasNoIdent“ für Aktionen um diese nicht anzuzeigen, falls eine Instanz ein untergeordnetes Objekt mit einem bestimmten Ident hat
Neu: MaintainVariable* Funktion erlaubt jetzt auch die Angabe einer Presentation als Array
Neu: RegisterVariable* Funktionen erlauben die Angabe einer Presentation als Array
Fix: Listen sind jetzt in Trees nutzbar
Fix: IPSList als Type Hint für IPSModule und IPSModuleStrict erlauben

## AKTUALISIERTE MODULE

### TELEFONKETTE

Neu: Hinzufügen einer Ansage via Text To Speech(AWSPOLLY)

# V7.1->V7.2 (Q3/2024)

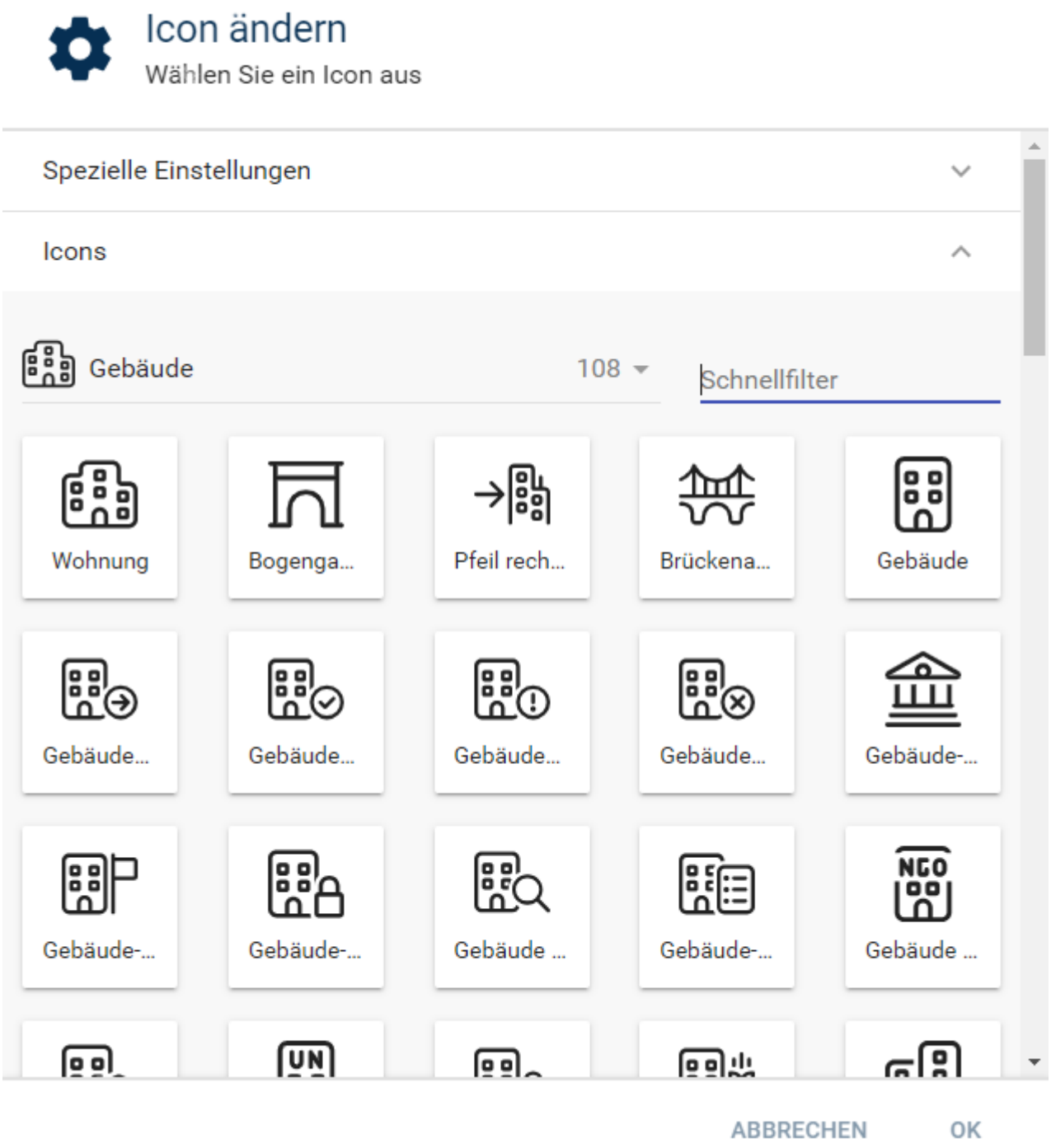
Wesentliche Änderungen
Migration
Liste der Änderungen
Neue und aktualisierte Module

## WESENTLICHE ÄNDERUNGEN

Neue Icons
KNX: Semantischer Export
Verbesserte Synchronisation entfernter Systeme
Stabile Streams in Visualisierung

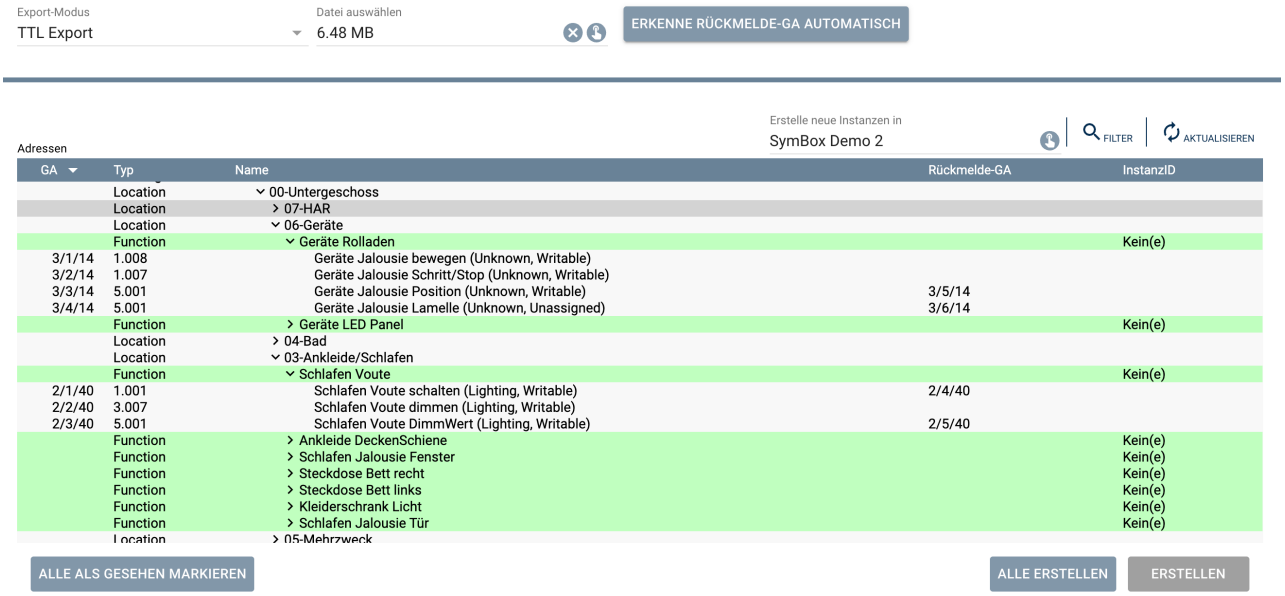
### NEUE ICONS

Der Umfang der Icons wurde um ein vielfaches erhöht. Anstatt vorher 165 Icons kann nun aus den 3787 verschiedenen Icons von Font Awesome Pro für alle möglichen Anlässe ausgewählt werden. Damit bei dieser umfangreichen Iconmenge dennoch das optimale Icon gewählt werden kann, unterstützt die Verwaltungskonsole eine Aufteilung nach Kategorie sowie einen intelligenten Schnellfilter.



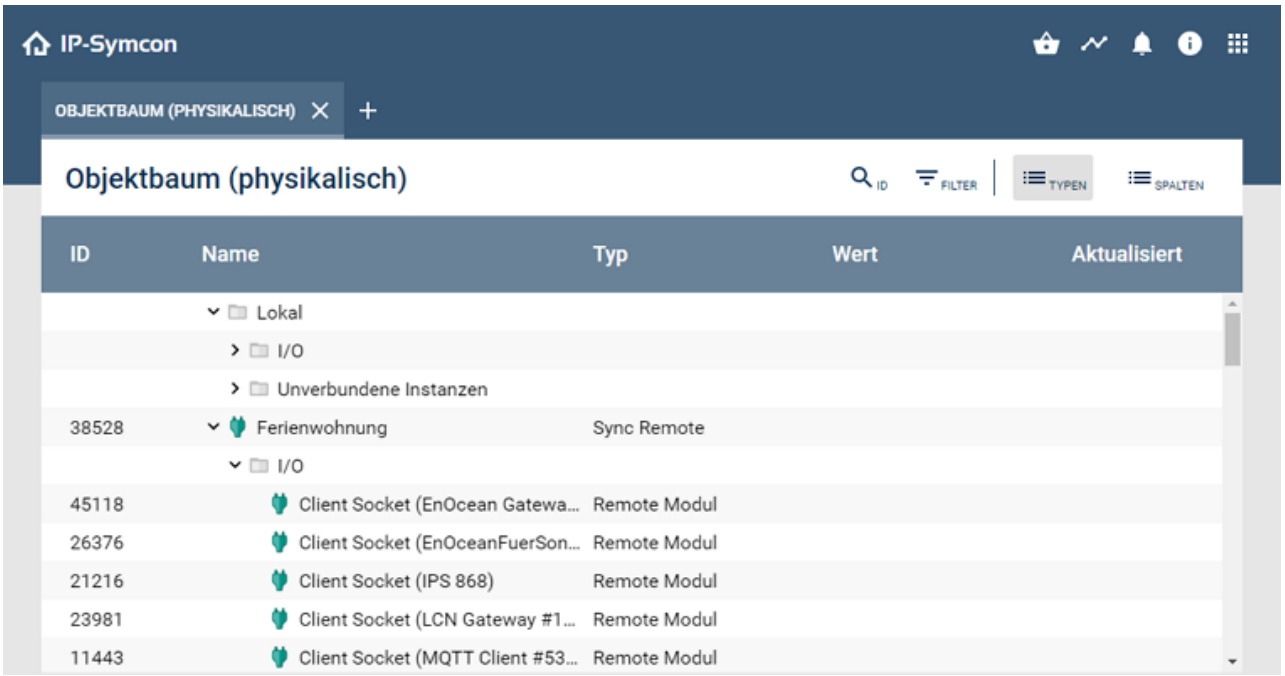

## KNX: SEMANTISCHER EXPORT

Im KNX Konfigurator ist nun der semantischen Export (TTL) integriert. So kann die Zeit zur Einrichtung der Visualisierung auf ein Minimum reduziert werden. Seit der ETS 6.1.1 ist der semantische Export verfügbar, der viele weitere Informationen aus dem KNX Projekt exportieren kann. Sofern das KNX Projekt mit der Gebäudestruktur und den Gerätefunktionen projektiert wurde, kann diese Information genutzt werden, um die Visualisierung fast vollständig damit aufzubauen - ein manuell Nacharbeiten oder Zuweisen von Status-Adressen, Namen oder der Gebäudestruktur ist nicht mehr erforderlich.




## VERBESSERTE SYNCHRONISATION ENTFERNTER SYSTEME

Die Einbindung des Sync Remote für unsere Enterprise-Kunden wurde weiter verbessert. Die Konsole zeigt nun auch Spezialkategorien wie Kern, Discovery, etc. des entfernten Systems an. Auch wird die Darstellung im physikalischen Objektbaum nun nach lokalen und entfernten Systemen aufgeteilt, sodass schnell der physikalische Aufbau eines bestimmten entfernten Systems ausgewertet werden kann.




## STABILE STREAMS IN VISUALISIERUNG

Die Stabilität von Videostreams in der Visualisierung wurde verbessert. Vorher konnte es in manchen Konstellationen dazu kommen, dass Streams abbrechen oder IP-Symcon abstürzt. Die entsprechenden Fehler wurden behoben, sodass Streams nun in allen Fällen wie vorgesehen in der Visualisierung verwendet werden.

## MIGRATION

Bitte erstellen Sie sicherheitshalber ein Backup, sollten wider erwarten Probleme bei der Aktualisierung auftreten.

Falls Sie von einer früheren Version aktualisieren, lesen Sie bitte vorher die passenden Migrationsanleitungen.

Die aktuellste Version kann wie unter Installation beschrieben installiert werden.

## LISTE DER ÄNDERUNGEN

### ALLGEMEIN

Neu: Logikpläne (ehemals Workflows) sind jetzt nativ integriert. (Das Module Store Modul kann gelöscht werden)
Neu: Button zum bequemen Erstellen von passenden Zertifikaten für alle SSL Instanzen wie Server Socket, Web Server oder OPC UA Client
Neu: Connect: Link zum Versenden per E-Mail wieder hinzugefügt
Neu: Sync Remote: Burst Abfragen nach einem RequestAction (um Traffic zu sparen)
Neu: Sync Remote: Unterstützung für Diff-Snapshots, um Traffic zu sparen
Neu: HTTP Client: VerifyHost/VerifyPeer Option
Neu: HTTP Client: Eigene Header können definiert werden
Neu: Translate: Anzeige der erkannten Systemsprache
Fix: Sync Remote: Fehler (Invalid Data Type) mit komplexeren Rückgabetypen gelöst
Fix: Fehlermeldung bei Aktion „Lese eine Objekteigenschaft aus“ gelöst

### GERÄTE

Neu: KNX: TTL (Semantischer Export) kann im Konfigurator importiert werden
Neu: KNX: KNX Gerät als Instanz, welches mehrere DPTs unter einer Instanz abbilden kann
Neu: KNX: Gruppenadresse der Instanz wird in der Info Spalte angezeigt
Neu: DMX: DMX Input Instanz, um z.B. DMX Fader über einen ArtNET Input empfangen zu können
Neu: BACnet: Unterstützung für den Typ „Accumulator“
Neu: ModBus: Fix für Abfrage von Datenblöcken in Kombination mit „Little Endian (Bytes getauscht)“

### KACHEL VISUALISIERUNG

Neu: Font Awesome Pro (Light + Brands) integriert (Insgesamt >3000 Icons)
Neu: Neue Fehlermeldung, wenn Endgerät keinen h264 Dekoder besitzt
Neu: Einstellungen für Energieverteilung wurden in die Kachelkonfiguration verschoben

(Danke an @yansoph!)
Neu: Sprachen-Auswahl auf Login-Seite bei Web, sofern das Translation Modul aktiv ist
Neu: Kategorien die als Kachel dargestellt werden können ihr Icon als Inhalt anzeigen
Neu: Zeilenhöhe der Listendarstellung angepasst
Fix: Zustand des Wochenplans wird korrekt synchronisiert
Fix: Demo funktioniert wieder korrekt
Fix: Verbesserungen bei parallelen Streams in der Kachel Visualisierung (weniger Abstürze!)

### WEBFRONT

Neu: Font Awesome Pro (Light + Brands) integriert (Insgesamt >3000 Icons)

### KONSOLE

Neu: Sync Remote: Darstellung vom spezial Kategorien (z.B. I/O, Splitter…) direkt unterhalb der Sync Remote Instanz
Neu: Sync Remote: Sync Remote Instanzen werden im physikalischen Objektbaum beachtet
Neu: Sync Remote: Remote IDs werden im Skript Editor korrekt aufgelöst
Neu: Auswahl aller neuen Font Awesome Icons inkl. Filterfunktion
Fix: Z-Wave Assozationen konnten nicht gelöscht werden

### SYMOS

Neu: Aktuell kann eine /mnt/system/.wireguard erstellt werden, die dann beim Start korrekt geladen wird und den wg0 aktiviert. Wir werden dies noch erweitern, sodass die Config Datei direkt über UI im SymOS geändert werden kann und auch zur Laufzeit übernommen wird.

### PHP-SDK

Neu: Font-Awesome Pro Icons (Light/Brands) können direkt verwendet werden
Fix: SelectValue in Konsole korrigiert
Fix: CheckBox wird bei disabled = false korrekt dargestellt

## AKTUALISIERTE MODULE

### RECHENMODULE

Neu: Modul zum Skalieren zwischen zwei Wertebereichen

### REPORT MODUL

Fix: Zeige Verbrauch korrekt an, Lücken beim Multi Report sind nun möglich

## SNOM

Neu: Unterstützung für Expansion Modul

## SPOTIFY

Fix: Korrektes Handling der Wiedergabe von gespeicherten Tracks

## TAILSCALE VPN

Fix: Fehlende Übersetzung und Doku-Link ergänzt

# V7.0->V7.1 (Q1/2024)

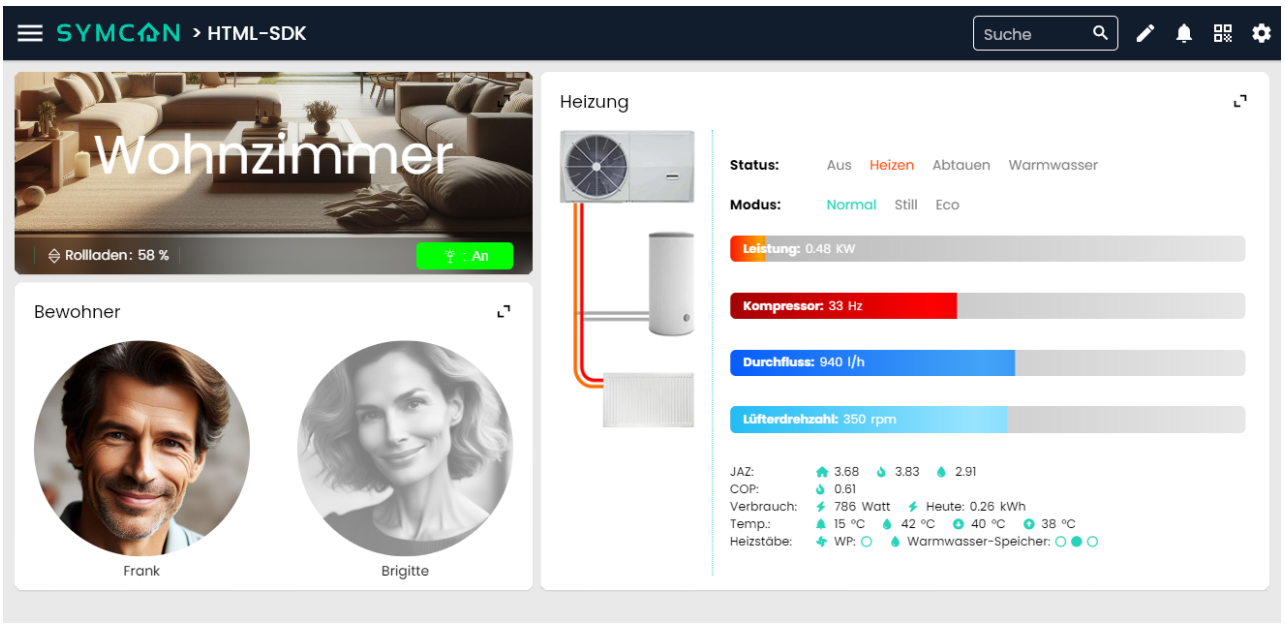
Wesentliche Änderungen
Migration
Liste der Änderungen
Neue und aktualisierte Module

## WESENTLICHE ÄNDERUNGEN

HTML-SDK
Logikpläne
Optimierte Darstellungen
VPN per SymBox

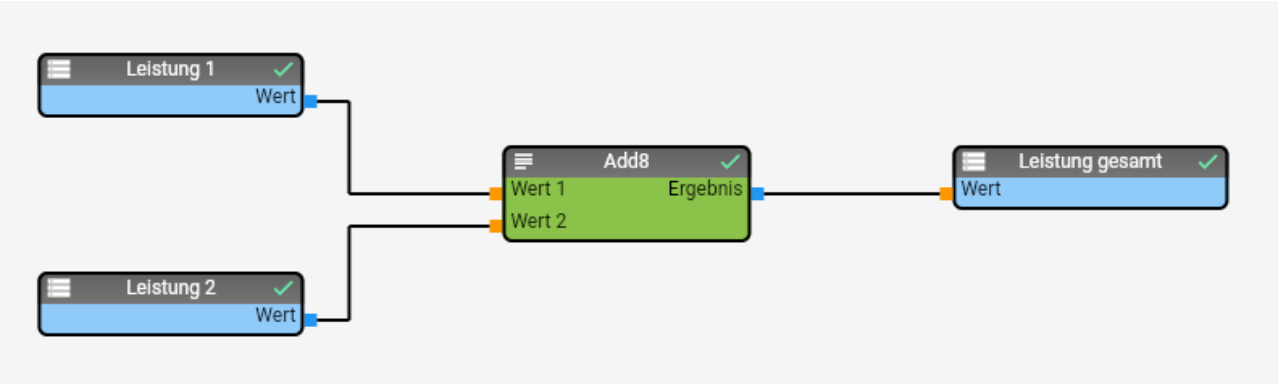
### HTML-SDK

Das neue HTML-SDK ermöglicht es Entwicklern individuelle Darstellungen für ihre Module anzubieten. So werden ganz neue Darstellungen wie die Wärmepumpe oder Anwesenheitsanzeige ermöglicht, welche die Kachel Visualisierung noch einmal aufwerten.



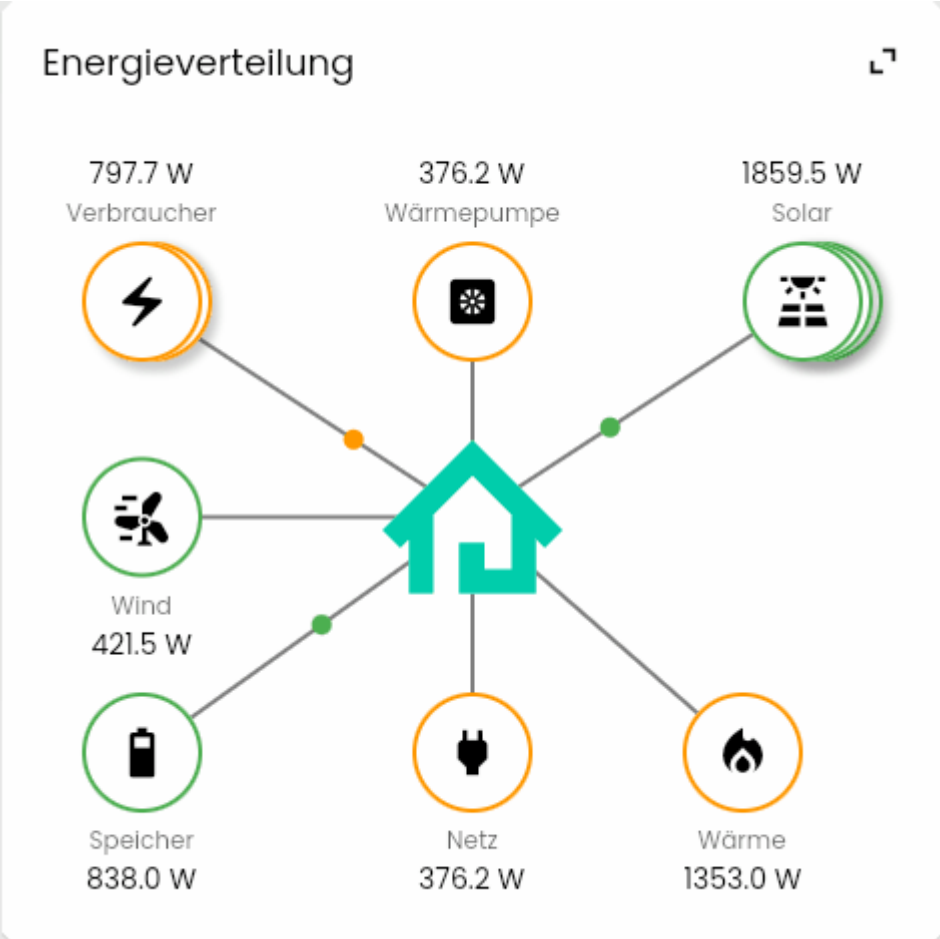

### LOGIKPLÄNE

Jeder entwickelt seine Automationen anders. Der eine erfreut sich am unkomplizierten Konfigurieren und Analysieren von Ablaufplänen, der andere genießt die Flexibilität von PHP-Skripten. Insbesondere Elektriker denken wiederum in Schaltplänen. Und auch diese Art der Automation wird nun für alle als Logikplan nutzbar.
Die bisherigen IPSWorkflows von IPSView kommen als Logikpläne fest nach IP-Symcon und können uneingeschränkt verwendet werden und bieten so noch mehr Möglichkeiten die optimale Automation zu generieren. Die Dokumentation ist auch bereits migriert:
Logikpläne




## OPTIMIERTE DARSTELLUNGEN

Mit der Version 7.1 kommen auch einige neue Möglichkeiten für die Kachel Visualisierung. Nun kann die Darstellung von Kategorien individuell angepasst werden. Damit können beispielweise die Kategorien auf einer Startseite als Kachel mit Bild dargestellt werden, während sie auf Unterseiten als Kategorieleiste genutzt werden. In diesem Rahmen kommen auch die aus dem WebFront bekannten Inhaltswechsler zurück, welche eine neue mögliche Darstellung für Kategorien sind.
Auch die Energieverteilung bietet diverse neue Möglichkeiten. Insbesondere ist es nun möglich mehrere Einträge gruppiert werden, sodass fünf separate Verbraucher platzsparend gestapelt dargestellt werden können. Die individuellen Verbräuche sind weiterhin via Auswahl platzsparend auslesbar.




## VPN (WIREGUARD) IN DER SYMBOX

Mit dem aktualisierten SymOS kann VPN über WireGuard/TailScale auf der SymBox eingerichtet werden. Damit kann auch aus der Ferne auf das Netzwerk hinter der SymBox zugegriffen werden. Dies ermöglicht eine unkomplizierte Konfiguration der entfernten Geräte, wodurch die Wartung deutlich vereinfacht wird.




AKTUALISIEREN

Version: 1.58.2

Dienst: Läuft!

Tunnel: Verbunden!

## MIGRATION

Bitte erstellen Sie sicherheitshalber ein Backup, sollten wider erwarten Probleme bei der Aktualisierung auftreten.

Falls Sie von einer früheren Version aktualisieren, lesen Sie bitte vorher die passenden Migrationsanleitungen.

Die aktuellste Version kann wie unter Installation beschrieben installiert werden.

## LISTE DER ÄNDERUNGEN

### ALLGEMEIN

Neu: Liste in der Connect Control Instanz, um für alle Visualisierungen den QR-Code anzeigen zu lassen oder diese direkt zu öffnen
Neu: Variablen unterhalb der Energieverteilung können deaktiviert werden
Neu: In der Kachel-Visualisierung Instanz gibt es einen Button, um diese die Kachel Visualisierung statt dem WebFront als Standard zu setzen. Das WebFront ist dann weiter über /legacy/ verfügbar.
Neu: Spezialschalter „ArchiveRecordLimitBool“ entfernt. Es gilt immer „ArchiveRecordLimit“.
Fix: Bei AC_FetchChartData* werden beim Erreichen der Limits die Fehler korrekt geworfen
Fix: Laden der php.ini robuster gemacht bei fehlerhaften Einträgen
Fix: $_IPS[‚CONFIGURATOR‘] wieder verfügbar, wenn auf dem WebFront eine Variablenaktion ausgeführt wurde
Fix: Doku Link für Kachel-Visualisierung korrigiert
Fix: Einige undokumentierte WFC_* Funktionen wie Visibility funktionierten nicht mehr korrekt seit einigen Versionen
Fix: Rate Limit für Google Assistant Status Reports erhöht
Fix: WebHook/OAuth Registrierung robuster gestaltet

## GERÄTE

Neu: MQTT Konfiguratoren ermitteln den Datentyp so gut es geht automatisch für die Erstellung der Geräte-Instanzen
Neu: M-Bus: Neue MBUS_UpdateValuesEx Funktion zum Abfragen spezieller Daten
Neu: CMI: Für UVR610 wird der DL-Bus ebenfalls abgefragt
Neu: OPCUA: Konfigurator stark verbessert
Neu: OPCUA: RequestedMaxReferencesPerNode auf 100 gesetzt, da einige Geräte mit 0 (=Ohne Limit) nicht korrekt umgehen
Neu: LCN Shutter: Bei Relais Ansteuerung wird nur noch 1 Kommando statt 2 gesendet
Neu: Unterstützung für neues eHZ LAN Gateway (für Lastgangzähler nach IEC 62056), welches wir auf der Light+Building zeigen werden
Neu: ModBus: TimeoutTime Property auf Gateway Ebene, um die Wartezeit in Sekunden anzugeben (Aktuell nur über IPS_SetProperty verfügbar)
Fix: KNX: Sporadische Probleme bei KNX/IP (TCP/Secure) Verbindungen gelöst
Fix: EnOcean: Power/Energy Measurement für D2-01-0E korrigiert
Fix: ModBus: VALUES bei virtuellen Adressen enthalten jetzt die Ergebnisse der vorherigen virtuellen Lesevorgänge
Fix: KNX: Disconnect Request wurde nicht korrekt beantwortet, sondern eine Fehlermeldung im Meldungsfenster geloggt
Fix: Sync Remote: Fehlermeldungen beim Schalten über die Visualisierung gelöst
Fix: Sync Remote: Fehlermeldungen beim CSV Export gelöst
Fix: Sync Remote: Medien (Bilder) werden jetzt korrekt übertragen

## KACHEL VISUALISIERUNG

Neu: Gruppieren in der Energieverteilung muss explizit aktiviert werden
Neu: Bessere Performance beim Rendern von Kachelansichten
Neu: Im Navigationsmenü kann zwischen den Visualisierungen des aktuellen Servers gewechselt werden
Neu: Darstellung des Wochenplans in einzelner Kachel optimiert

Neu: Visualisierungen eines Servers müssen nicht mehr explizit eingerichtet werden (App)
Neu: Name der Visualisierung wird im Tab-Titel angezeigt (Web)
Neu: Checkboxen in Chart Legende zusammengefasst
Neu: Minimale Laufzeit für Skript-Buttons
Neu: ‚Z2M.occupied_heating_setpoint' Profil funktioniert für Heizungskacheln Temperatur
Neu: IPSView nimmt die ganze Kachel ein (Benötigt noch Anpassungen von @Brownson)
Neu: ‚Lamella/Shutter.Reversed' wird unterstützt
Neu: Werte der Energieverteilung werden jetzt nach Typ gruppiert
Neu: Bunte Emojies
Neu: Energieverteilung bietet mehrere Variablen pro Gerät an und ermöglicht die Berechnung für Gruppen anzupassen
Neu: Neue Logik zur Verbindung zum Server
    Wechseln zwischen Verbindungen verbessert
    Neue Ansicht mit Log bei Neuverbindung
    Verständlichere Fehlermeldungen
Neu: Farbe des Speichers in der Energieverteilung angepasst (seit 7.1)
Neu: Höhe des Diagramm-Dialogs angepasst
Neu: CSV Export
Neu: Aktualisierungsrate der Begrüßung verbessert
Neu: Deep-Linking: QR-Codes können von anderen Apps gescannt werden
Neu: Fehlermeldung beim Schalten werden unten als rote Meldung dargestellt und nicht mehr gestapelt
Neu: Eine eventuelle Rückgabe von Automationen wird als Dialog angezeigt
Neu: Zeige mehr Informationen beim Versuch die Verbindung via App herzustellen an
Neu: Die Darstellung einzelner Kacheln kann aus dem Editier-Modus heraus verändert werden
Neu: Darstellungen für Kategorien:
    Kategorien können unabhängig voneinander in der Kategorieleiste oder als einzelne Kachel dargestellt werden
Neu: Inhaltswechsler
    Es kann ausgewählt werden, welche Objekte im Inhaltswechsler angezeigt werden sollen
    Elemente können nach einer festgelegten Zeit automatisch gewechselt werden
Neu: Für Listendarstellung von Instanzen: Nach Veränderung der aktiven Elemente werden neue Kinder automatisch angezeigt
Neu: Sollte versucht werden mehr HTML-Inhalte anzuzeigen als technisch Möglich, wird ein passender Fehler angezeigt.
Neu: Leistungsmodus der Energieverteilung wird gespeichert und wurde um statische Pfeile ergänzt
Neu: Aus dem Popup der Energieverteilung können Charts geöffnet werden
Neu: Wenn keine automatische Verbindung möglich ist, wird ein Button zum Neuverbinden angeboten
Neu: Energieverteilung
    Netz: Überschuss wird mit Verbrauch verrechnet
    Hat ein Gerät mehrere Variablen werden beide im Popup angezeigt
Fix: Chart-Buttons werden nicht mehr hinter Instanzen angezeigt
Fix: Anpassung der Skalierung des Logos in der Energieverteilung
Fix: Änderungen an Kategorien werden in der Kategorieleiste richtig angezeigt
Fix: Schieberegler stellen sich nun korrekt zurück, falls die Schaltoperation fehlschlägt
Fix: Namen einzelner Geräte werden in der Energieverteilung korrekt angezeigt
Fix: Schieberegler dürfen eine Schrittweite von 0 haben

Fix: Nicht-Geräte-Instanzen können verlinkt werden
Fix: Schnellwechsel nur für mehrere Visualisierungen verfügbar. Deaktivierte Visualisierungen sind deaktiviert
Fix: Autofill für Passwortdialog (Web-Only)
Fix: Fehler bei der Darstellung einer bei der Variablen gelöscht wurden
Fix: Einstellungen verlinkter Charts werden korrekt gespeichert
Fix: initial Anzeigen bei Charts wird berücksichtigt
Fix: Fehler wird angezeigt, wenn die aktuelle Kategorie gelöscht wurde
Fix: Minimaler Größenunterschiede der Kacheln behoben
Fix: Link zur Doku führt zur Doku und nicht in Forum
Fix: Fehlermeldung bei fehlenden Variablen im Chart
Fix: Uhrzeit in der Appbar aktualisiert sich sekündlich
Fix: Fehler bei automatischer Kachelauswahl
Fix: Präziser Farbwahl-Dialog funktioniert jetzt korrekt
Fix: Fehler in Verbindung mit dem ‚~Temperature.Difference' und ‚~Temperature.Fahrenheit' Profilen behoben
Fix: Slider ohne Schrittweite
Fix: Konfiguration Synchronisation auf mehreren Geräten
Fix: Charts mit widersprüchlichen Einstellungen in der Instanzkonfiguration
Fix: Fehler beim verschieben von Kacheln
Fix: Seitenübergangsanimationen reduziert (Web)
Fix: String-Eingabe zeigt die Korrekte Tastatur auf Samsung Geräten an (App)
Fix: Breitenabschätzung von Aufzählungen in Listen korrigiert
Fix: Auf älteren Geräten, die kein WebAssembly unterstützen, wird eine passende Fehlermeldung angezeigt, dass die
    Visualisierung auf diesen Geräten nicht unterstützt wird
Fix: Bei Diagrammen in den Apps mit Legenden in der Seitenleiste heben beim Hovern den korrekten Bereich hervor
Fix: Bei Visualisierungen mit langem Namen wird dieser im Navigationsmenü passend gekürzt
Fix: Höhe des Headers des Navigationsmenüs wird in den Apps korrekt dargestellt
Fix: Sehr lange Strings werden in der Darstellung gekürzt, können in TextBoxen aber komplett geladen werden
Fix: Stream Kachel zeigt Fehlermeldung, wenn Stream kein RTSP/RTSPS ist
Fix: Schieberegler in Dialogen aktualisiert springe nicht mehr zurück
Fix: Fehler behoben, der beim Löschen von Benachrichtigungen auftreten konnte
Fix: Schließe Verbindungen zum Server, wenn Visualisierung nicht geladen werden kann
Fix: Instanzen deren Darstellungstyp geändert wurde aktualisieren sich korrekt
Fix: Bei Verbindung zum lokalen Netzwerk und späterem Wechsel auf Connect wird das gespeicherte Kennwort verwendet
Fix: Korrektes Handling bei der initialen Erstellung eines Links in der Hauptkategorie
Fix: HTML-Boxen aktualisieren sich auch bei Variablenaktualisierung
Fix: Verhindere einige Verbindungsabbrüche
Fix: Inhaltswechsler
    Hintergrundfarbe wird von Variablen übernommen
    Titelgröße wird für jedes Objekt berechnet
Fix: Legenden werden nach Instanz-Einstellungen geladen
Fix: Fehler behoben, wenn Variablen in der Diagramm-Legende gefiltert und ausgewählt werden
Fix: Multigraphen werden korrekt in Inhaltswechslern angezeigt
Fix: Authentifizierung für HTML-Boxen in Apps hinzugefügt

Fix: Button zum neu verbinden funktioniert wie vorgesehen
Fix: Links auf die Hauptkategorie werden angezeigt
Fix: Neue Visualisierungen mit Passwörtern können wieder geöffnet werden
Fix: Neu hinzugefügte Server werden korrekt gespeichert
Fix: Füge Timeouts bei Verbindungsversuchen hinzu, damit diese nicht hängen bleiben
Fix: Verbesserte Performance bei Graphen mit sehr vielen Variablen
Fix: Stelle Zeiten im CSV-Dialog wie vorgesehen in der Zeitzone des Servers dar
Fix: Direkte Links auf passwortgeschützte Visualisierungen funktionieren wieder
Fix: String Variablen werden in den Zusatzinformationen der Energieverteilung korrekt angezeigt
Fix: Ungültige Variable in Begrüßungszeile wird korrekt abgefangen
Fix: Performance von Charts mit vielen Variablen verbessert
Fix: Darstellung der Zeit im CSV-Dialog korrigiert
Fix: Darstellung von Enumerationen in Listen verbessert

## WEBFRONT

Fix: Assoziationen mit Platzhaltern und Aktion zeigen immer den korrekten Variablenwert an (konsistent zur neuen Visualisierung)
Fix: IPSView Views konnten nicht geöffnet werden, wenn /legacy/ verwendet wurde

## KONSOLE

Neu: Zeige beim Speichern der Debug-Ausgaben in eine Datei den Ort dieser Datei an
Neu: Kopieren/Ausschneiden/Einfügen/Verlinken via Kontextmenü funktioniert nun analog zu Strg + C/X/V/L
Neu: Dokumentation „Button“ in der Konsole öffnet die Dokumentation automatisch in der korrekten Sprache
Neu: Zeige Logikpläne als Automation immer an und installiere bei Bedarf die benötigten Module aus dem Module Store
Neu: Konfiguration der Spalten bei PHP-Information wird gespeichert
Fix: Zeige korrekten Titel im Dialog beim erstellen von Kategorien via Alt + 0
Fix: Seltenen Fehler korrigiert, der beim Einklappen von unsichtbaren Knoten auftreten konnte
Fix: Beim Abdocken von IPSView kommt immer der Hinweis, wenn es ungespeicherte Änderungen gibt
Fix: Neu hinzugefügte Visualisierungen werden im Visualisierung-Widget korrekt hinzugefügt
Fix: String-Variablen können nun aus der Konsole reaggregiert werden (auch wenn dies lediglich die Rahmendaten wie Start- und Endzeit neu berechnet)
Fix: Im Module Store konnte bei Modulen aus der Vorschau fälschlicherweise ein Update angezeigt werden wenn auf dem eigenen System eine Beta- oder Testversion installiert ist, dies wurde behoben
Fix: Bei Sortierung nach Namen in der Timer Information wird nun primär nach Namen und nicht mehr nach Pfad sortiert
Fix: IPSView Updates können nur installiert werden, wenn eine gültige IPSView Subskription vorhanden ist
Fix: Verdichtung für String Variablen korrekterweise deaktiviert (Nur für Bool/Int/Float verfügbar)

Fix: Behebe Typo beim sicheren Löschen
Fix: Korrigiere Meldung, wenn ein Logikplan erstellt wird, das benötigte Modul aber noch nicht installiert ist
Fix: Wird die Prüfung beim sicheren Löschen abgebrochen wird wie vorgesehen der gesamte Löschvorgang abgebrochen
Fix: Korrigiere Validierung, ob das Modul zu einer Instanz existiert
Fix: Verhindere Fehler beim Bearbeiten von mehreren Ereignissen mit unterschiedlichen Aktionen
Fix: Springe wie vorgesehen zur letzten Position bei Objektauswahlen, wenn das aktuell ausgewählte Objekt nicht (mehr) existiert
Fix: Cache LocationControl ID im Location Widget (Verbessert die Performance massiv auf Systemen mit sehr viel Variablenänderungen)

## PRO KONSOLE

Neu: Bibliotheken auf die neuste Version aktualisiert (Es wird mind. Windows 8 und Windows Server 2012 benötigt)
Neu: Beim Hinzufügen von Servern kann das System mit angegeben werden
Neu: SymOS Connect für KNX für SymBox'en direkt integriert
Neu: SymOS Connect für LCN für SymBox'en direkt integriert

## SYMOS

Neu: WireGuard im Kernel aktiviert (wg utility ist auch verfügbar)
Neu: Kernel auf 6.1.61 aktualisiert
Neu: SymOS Connect für KNX ist offiziell verfügbar
Neu: SymOS Connect für LCN ist offiziell verfügbar
Neu: VPN über Tailscale Modul verfügbar
Fix: In manchen Situationen wurde der verfügbare Speicherplatz nicht korrekt ermittelt

## PHP-SDK

Neu: HTML-SDK für die Kachel Visualisierung
Neu: Copy & Paste innerhalb von List und Tree via Strg + C/Strg + V möglich
Neu: Configurator unterstützt nun align und sortColumn analog zu List/Tree
Neu: link/download darf auch in Listen auf Column Ebene gesetzt werden
Neu: SelectCondition unterstützt die Eigenschaft „caption"
Neu: SelectAction verbessert mit Auswahl zwischen generellen Aktionen und Aktionen mit Ziel
Neu: Analog zu SelectCondition unterstützt nun auch SelectAction das Feld „caption"
Fix: description bei QrCode funktioniert jetzt wie vorgesehen
Fix: onClick wird nun wirklich nur bei angezeigtem leeren Text und nicht bei einem „falsey" deaktiviert
Fix: Bei on-Skripten und einer Liste ohne ausgewählte Zeile und add-Werte wird die aktuelle Zeile nun wie vorgesehen mit null initialisiert
Fix: Verhindere Fehler, wenn in einem Konfigurator die Sortierung deaktiviert wird
Fix: SelectAction verwendet wieder wie vorgesehen die gesamte verfügbare Breite

## NEUE MODULE

### EKEY BIONYX

Integration von ekey bionyx Geräten direkt in IP-Symcon
IP-Symcon empfängt die Aktivitäten des Fingerscanners über Webhooks

### SNOM

Integration von Snom Geräten direkt in IP-Symcon

## AKTUALISIERTE MODULE

### AMAZON ALEXA

Fix: Korrektes Handling mit PHP 8.2

### ANWESENHEITS-SIMULATION

Fix: Anpassungen für IP-Symcon 7.0

### BENACHRICHTIGUNG

Fix: Einige kleinere Fehler behoben

### ENERGIEVERBRAUCH OPTIMIERER

Neu: Unterstützung für E-Autos

### GARDENA SMART SYSTEM

Fix: Intervall für die Gerätesuche erhöht

### GOOGLE ASSISTANT

Fix: Korrektes Handling von PHP 8.2

### SPOTIFY

Neu: Profile angepasst, damit sie mit dem Mediaplayer der neuen Visualisierung kompatibel sind

## TEXT TO SPEECH (AWS POLLY)

Fix: Unterstützung für IP-Symcon 7.x

# V6.4->V7.0 (Q4/2023)

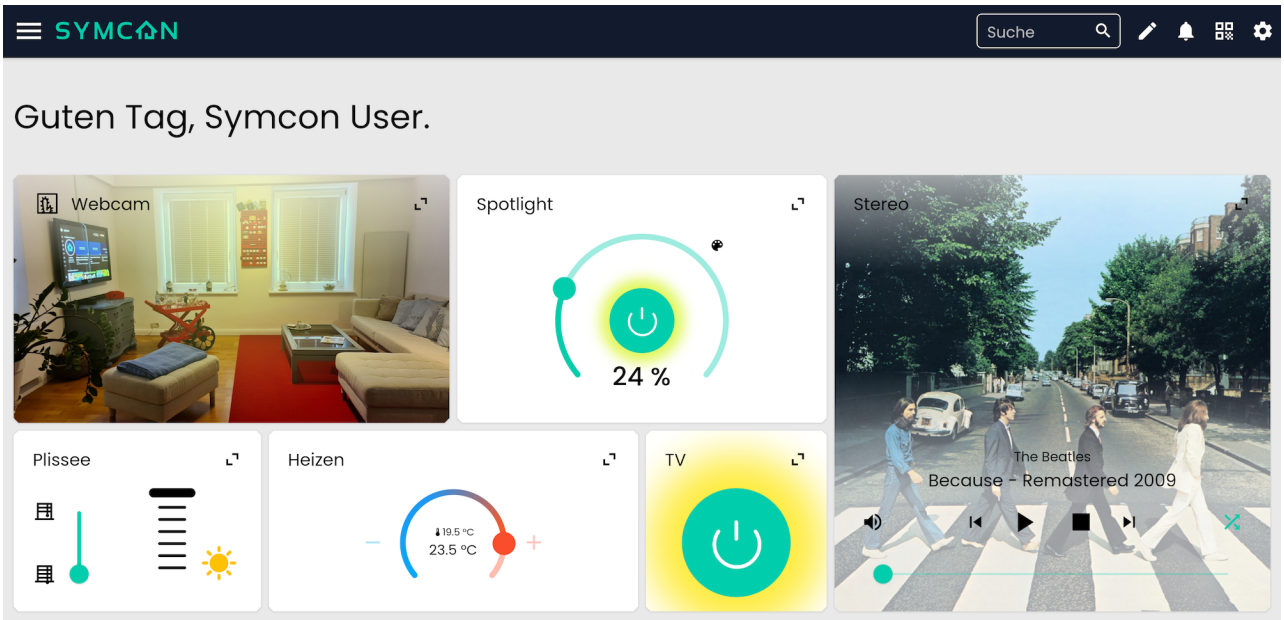
Wesentliche Änderungen
Migration
Liste der Änderungen
Neue und aktualisierte Module

## WESENTLICHE ÄNDERUNGEN

Neue Visualisierung
Integriertes Energie-Management
SymBox Pro mit KNX Data Secure
Stark verbesserte ModBus Anbindung

### NEUE VISUALISIERUNG

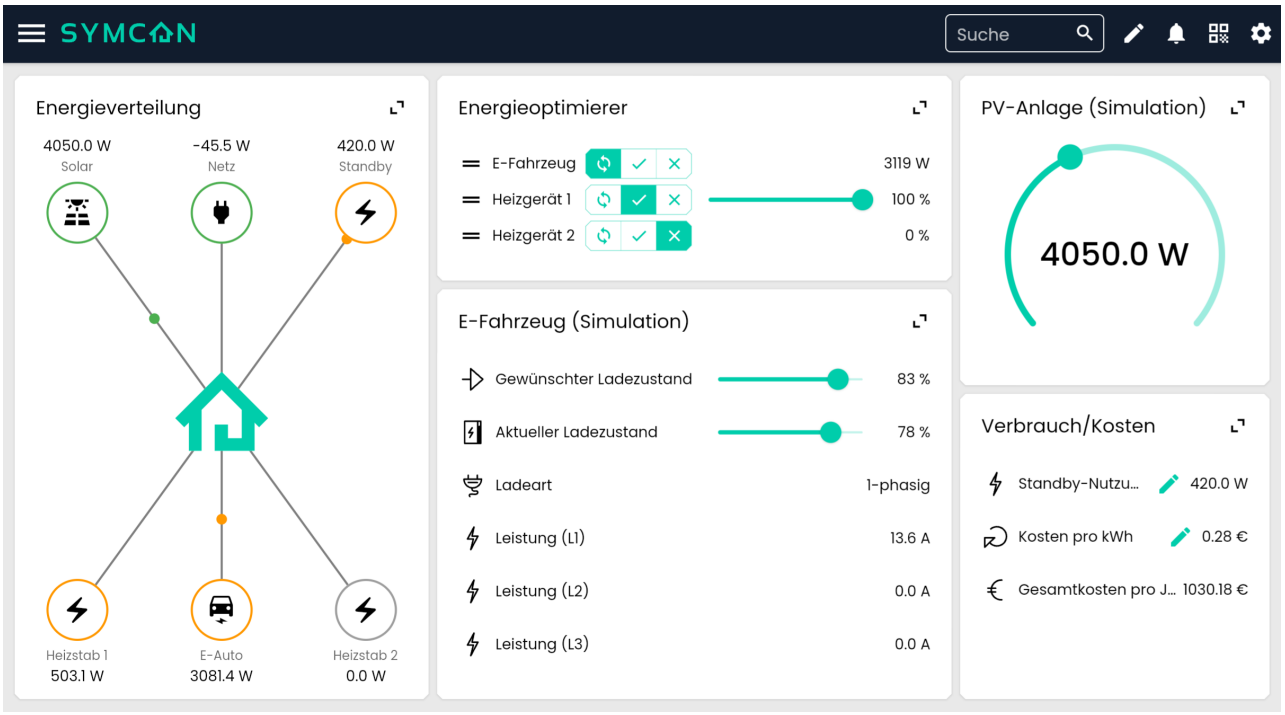
Die neue, moderne Visualisierung im Kachel-Design ist ab sofort verfügbar. Mit hellem und dunklem Design, Drag&Drop Editor zum Positionieren der Kacheln und vielen spezialisierten Kacheln für die relevantesten Anwendungszwecke. Verfügbar für Web, iOS und Android bietet es eine plattformübergreifende, konsistente Darstellung. Eine Demo ist unter https://demo.symcon.de verfügbar!
Um die neue Visualisierung nutzen zu können, muss einfach eine neue Instanz vom Typ "Kachel Visualisierung" erstellt werden.
Download für iOS: https://apps.apple.com/de/app/symcon-visualization/id1566124129
Download für Android: https://play.google.com/store/apps/details?id=de.symcon.visualization




### INTEGRIERTES ENERGIE-MANAGEMENT

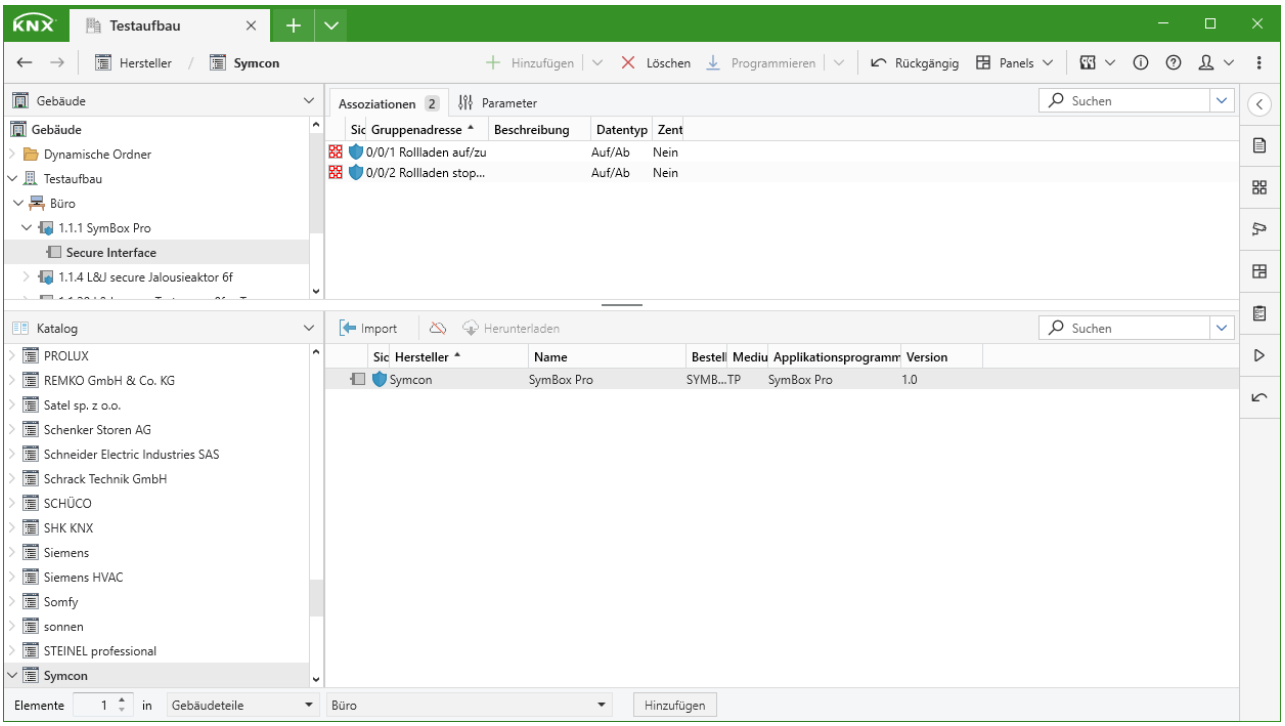
PV-Überschuss Optimierung mit wenig Aufwand! Der neue Energieverbrauch-Optimierer und die Energieverteilung in der neuen Visualisierung helfen die vorhandene Energie optimal zu verbrauchen. Die neue Darstellung gibt dabei immer den Überblick über alle

Werte. Vorlagen für ModBus machen die mehrfache Einrichtung zum Kinderspiel.



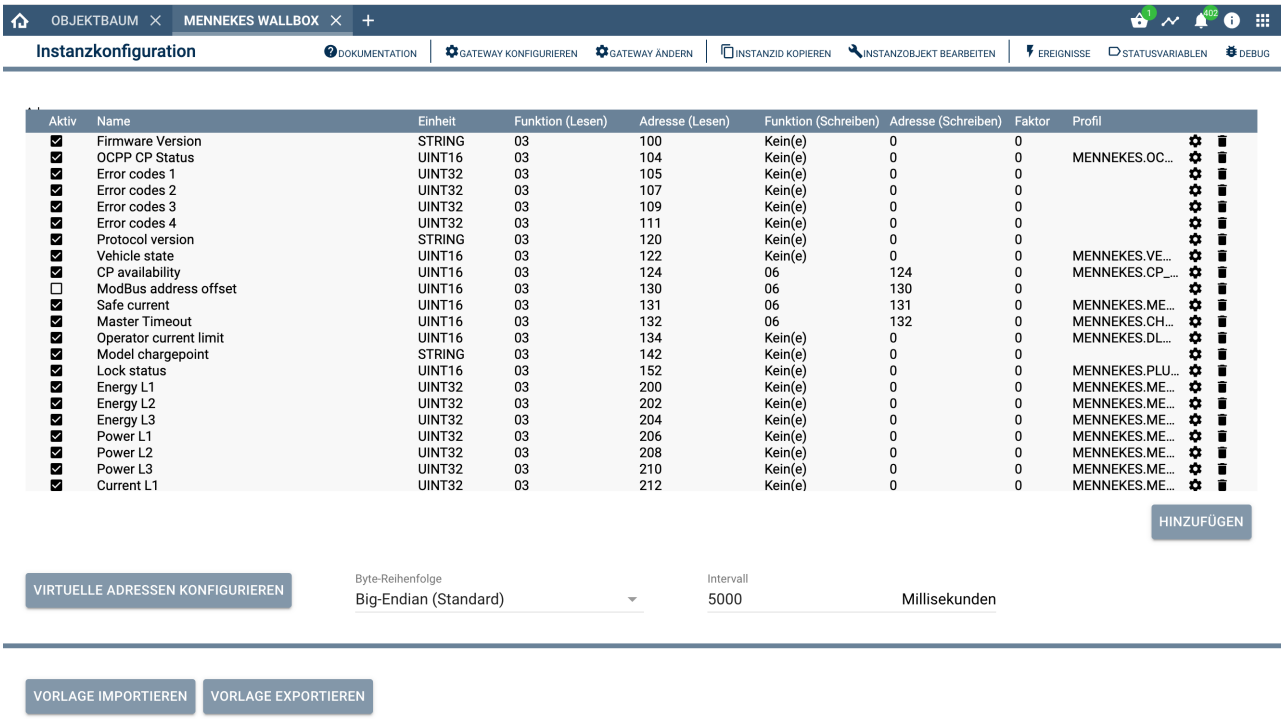

## SYMBOX PRO MIT KNX DATA SECURE

Die SymBox Pro ist ab sofort offiziell von der KNX Association zertifiziert und ist im Produktkatalog der ETS auffindbar. Dadurch können Gruppenadressen direkt auf den Tunnel der SymBox Pro gezogen werden, womit sich Filtertabellen und die KNX Data Secure Verknüpfungen vollständig automatisch konfigurieren. Ein Video zur Einrichtung ist auf unserem YouTube Kanal zu finden: https://www.youtube.com/watch?v=aA8mcKEyHOY&t=1s



## STARK VERBESSERTE MODBUS ANBINDUNG

Die ModBus Anbindung wurde überarbeitet, sodass direkt alle Adressen eines Geräts in einer Liste eingetragen werden können inklusive aller Namen, Parameter und Übersetzungen der Namen. Zusätzlich kann die Gerätekonfiguration jederzeit Exportiert und Importiert werden, wodurch die Einrichtung mehrerer Geräte ein Kinderspiel wird und die Vorlagen auch in unserer Community geteilt werden können. Wir haben im ersten Schritt Vorlagen für die Wallboxen/Ladesäulen von Mennekes, ABL, KEBA und ALFEN erstellt, welche in der Dokumentation heruntergeladen werden können. Die bereits geteilten Vorlagen aus der Community sind hier zu finden: https://community.symcon.de/c/86




## MIGRATION

Bitte erstellen Sie sicherheitshalber ein Backup, sollten wider erwarten Probleme bei der Aktualisierung auftreten.

Falls Sie von einer früheren Version aktualisieren, lesen Sie bitte vorher die passenden Migrationsanleitungen.

PHP wurde von 7.4 auf 8.2 aktualisiert. Insbesondere Nutzer der IPSLibrary/IPSLogger sollten VOR dem Update auf IP-Symcon 7.0 diese aktualisieren. Notfalls können die notwendigen Änderungen manuell nachgepflegt werden (Details). Die meisten PHP-Module wurden bereits aktualisiert, sodass diese bereits mit IP-Symcon 7.0 kompatibel sind.
Unter Windows wurden einige selten verwendete PHP-Extensions entfernt, sodass aktuell

nur noch IMAP, OCI, PGSQL und SNMP optional installiert werden können. Die meisten anderen PHP-Extensions sind bereits standardmäßig auf allen Plattformen inkludiert (Details).
Die Pfade zum /user/ und /skins/ Ordner werden beim Upgrade verändert, sodass diese auf der Hauptebene und nicht mehr im /webfront/ liegen. Ggf. müssen PHP-Skripte angepasst werden, die auf diese Ordner hart kodiert zugreifen.

Die aktuellste Version kann wie unter Installation beschrieben installiert werden.

## LISTE DER ÄNDERUNGEN

### ALLGEMEIN

Neu: Neue Visualisierung verfügbar (Siehe: https://community.symcon.de/t/133822)
Neu: Neue Streams-Integration in der neuen Visualisierung inkl. 4K Support (Siehe: https://community.symcon.de/t/134356)
Neu: PHP 8.2
Neu: Einige selten genutzte PHP-Erweiterungen unter Windows entfernt
Neu: Einige Profile für Rollladen/Lamellen verbessert, sodass direkt die neue Kachel in der Visu verwendet wird
Neu: Spezialschalter "DefaultVisualization", mit der man einstellen kann, welche Visualisierung standardmäßig geöffnet wird (0 = WebFront, 1 = Tile)
Neu: Spezialschalter "CompatibilityDeprecations", um auch die Deprecations wie utf8_* von PHP 8.2 zu aktivieren
Neu: Spezialschalter "DisableLogfile", um das Logfile fast komplett zu deaktivieren
Neu: Alle Bibliotheken aktualisiert. (z.B. auch OpenSSL auf 3.0)
Neu: Fehlende Icons für UnixTimestamp*, HexColor, TWColor ergänzt
Neu: Konstante SCRIPTTYPE_IPSWORKFLOW hinzugefügt
Neu: Konstante MEDIATYPE_IPSVIEW hinzugefügt (ersetzt MEDIATYPE_FORM)
Neu: Bei (z.B. UTF-8) Fehlern im Datenfluss wird die Meldung nur einmal für alle verbundenen Instanzen geloggt und zusätzlich mit mehr Informationen ausgegeben
Neu: Das Neuverbinden von fehlerhaften Instanzen verbessert und konfigurierbar gemacht
Neu: Weitere Modulnamen werden jetzt übersetzt (z.B. Energieverteilung)
Neu: ModBus ASCII wird unterstützt (für z.B. ABL eMH1)
Neu: KNX: DPT 14.080
Neu: KNX: DPT 21.1002, 21.1200, 21.1201
Neu: KNX: DPT 205.103
Neu: KNX: DPT 275.100
Neu: KNX: Passwort für Secure Schlüsselbund wird überprüft (und ggf. fehlerhaft gemeldet)
Neu: DMX: Option zum Dauerhaften senden von Paketen hinzugefügt
Neu: Bibliothek für VideoStreams+VoIP aktualisiert (Löst hoffentlich die Abstürze bei den Streams)
Neu: Farbtemperatur und Temperatur Regler werden auch in Listen eingefärbt
Neu: Bei Auswahl der Demo kann die Visualisierung gewählt werden
Neu: Unterstütze Übersetzung der Visualisierungsnamen (Enterprise)
Neu: Verbesserter Precheck für veraltete IPSLibrary/IPSLogger in der autoload.php
Neu: ModBus: Neuer STRING (HEX) Datentyp, um Binärdaten auszulesen
Neu: ModBus: Virtuelle Adressen werden nach den normalen Adressen positioniert
Neu: ModBus: „null“ Werte bei virtuellen Adressen werden als fehlender Wert interpretiert

Neu: ModBus: Beim Schreiben in virtuellen Adressen ist nun $VALUES verfügbar mit den aktuellen Werten
Fix: ModBus: Manuell erstellte Variablen unterhalb der Geräte-Instanz werfen keine Fehler mehr
Fix: ModBus: Leere „Leseaktion“ bei virtuellen Adressen funktioniert wie erwartet
Fix: Workaround für defekte php.ini nach Upgrade unter Windows
Fix: Sofern NATSupport in den Spezialschaltern aktiv ist, wird der Spezialschalter NATPublicIP den mobilen Apps ebenfalls mitgeteilt
Fix: Der Demoserver wird nicht als Server abgespeichert
Fix: Baumauswahl im Drawer wird erst bei Bedarf geladen und unterstützt so deutlich umfangreichere Bäume
Fix: Aktualisiere Änderungen an Kategorienamen innerhalb der Kategorieleiste
Fix: Zeige Legende in Graphen an der Seite korrekt an, falls Bedienungselemente ebenfalls an der Seite sind
Fix: Datumsauswahldialoge über HTML Boxen funktionieren wie vorgesehen
Fix: Problem mit HTMLBox über HTMLBox gelöst
Fix: ModBus: Überspringe Inf/NaN Werte, damit Vorlagen bei diesem nicht kritischen Fehler weiter abfragen. (Der Wert wird nicht in die Variable geschrieben und ein Hinweis ins Debug Log notiert)
Fix: BACnet: Dekodieren von UCS-2 Zeichenketten korrigiert (Dadurch konnte der Einlesevorgang der Datenpunkte abbrechen/stecken bleiben)
Fix: Geräte für Benachrichtigungen können in den Visualisierung Instanzen bearbeitet werden
Fix: KNX: SymBox mit KNX wird in der Discovery Instanz korrekt angezeigt
Fix: Precheck für autoload.php nach hinten verlegt, sodass alle Kernel Module verfügbar sind
Fix: UVR610 unterstützt ebenfalls M-Bus abfragen
Fix: Absturz korrigiert, wenn eine Ereignis Bedingung gelöscht wurde, die nicht existierte
Fix: "cert" Ordner wird nicht mehr erstellt, der seit langem nicht mehr genutzt wurde
Fix: Löschen von Skripten/Medien bewegt diese wieder in den ‚deleted‘ Order (seit 6.3)
Fix: __generated.inc.php wird jetzt vorab generiert und dann getauscht und löst dadurch Fehlermeldungen beim ModuleReload
Fix: Intensity.1 Profile benötigte keine Nachkommastellen
Fix: Security-Context für Visu’s wurde zu restriktiv berechnet, sodass Objekte unterhalb von Variablen, die mit Ereignissen verknüpft waren, manchmal nicht angezeigt wurden
Fix: Manuelles Auslesen von Lizenz Informationen im Util Control wieder möglich
Fix: ModBus Geräte konnte nicht mehr konfiguriert werden, da dynamische Formulare nicht geladen wurden (seit letztem Update)
Fix: CORS Fehler korrigiert beim Laden von Schriftarten in HTMLBoxen

## GERÄTE

Neu: ModBus: Stark verbesserte Anbindung mit Im/-Export Funktion für Vorlagen
Neu: ModBus: Byte Order wurde im Splitter entfernt und ist jetzt nur noch in den Geräte/Coil/Adress Instanzen verhanden. Zusätzlich kann in der Geräte Instanz die Byte-Order pro Adresse überschrieben werden
Neu: ModBus: Byte Order Optionen „Little Endian“ und „Big Endian (Byte Swap)“ hinzugefügt
Neu: EnOcean: Zusätzliche Werte für STM550 (D2-14-41) verfügbar
Neu: JSON Dekoder/MQTT Device: Option zum Sperren der Variablen, sodass keine

Variablen erstellt/gelöscht werden bei falschen Datenpaketen
Neu: KNX: Setzen der physikalischen Adresse/programmieren der Applikation verbessert (Nur SymBox)
Neu: LCN: Sperren für Regler können in der Visu ausgewertet/gesetzt werden
Neu: SMTP Unterstützt senden von HTML, wenn und im Inhalt gefunden wird
Neu: WebSocket Client und WebHooks unterstützen den binären Datentransfer
Fix: SSE Client (z.B. für Philips Hue V2) erkennt Verbindungsabbrüche besser
Fix: Hinweis im HTTP Client bzgl. Kompatibilität korrigiert
Fix: M-Bus: Bei Dekodierfehlern einzelner Felder, wird nicht die gesamte Verarbeitung abgebrochen
Fix: RegVar_SendText in Kombination mit ServerSocket korrigiert
Fix: Z-Wave: Wenn ein Gateway überlastet ist und Pakete „gedroppt“ hat, breche nach „Retry“ Fehlversuchen ab, anstatt es weiter zu versuchen
Fix: MQTT: Topics werden korrekt auf UTF-8 Konformität geprüft und mit besserer Fehlermeldung bemängelt
Fix: DigitalStrom: Einige Binärtypen ergänzt
Fix: ServerSocket: Timeout beim TLS Handshake implementiert, sodass dieser nicht mehr feststecken kann bei instabilen Verbindungen
Fix: VoIP: Bei vielen ausgehen Anrufen konnte die ConnectionIDs durcheinander geraten
Fix: EnOcean: Verbessertes Profil für Backup Battery (D2-14-41)
Fix: KNX: Absturz korrigiert, sofern fehlerhafte Pakete ohne Daten empfangen wurden
Fix: KNX: Erkannte Gruppenadressen für Statusrückmeldungen werden nicht doppelt hinzugefügt

## WEBFRONT

Neu: DWD kann nun auch Vertikal dargestellt werden (Option verfügbar im Editor)
Neu: Wenn standardmäßig die neue Visualisierung aktiv ist, kann das WebFront über /legacy/ erreicht werden
Neu: /webfront/user Ordner wird nach user migriert
Neu: /webfront/skins Ordner wird nach skins migriert
Fix: Rote Fehlermeldung, welche beim Start kurz aufflackern konnte, wurde korrigiert

## KONSOLE

Neu: Visualisierungs-Widget unterstützt nun auch die neue Visualisierung
Neu: IPSWorkflow wird nach der Konvertierung direkt geöffnet
Neu: Dialoge können nun in der webbasierten Konsole vergrößert werden (bisher nur in der Pro Konsole möglich)
Neu: Übersetzungen für Modulnamen/-aliasse werden geladen (noch nicht ganz vollständig für interne Module)
Neu: Moduldaten werden lokal gehändelt, wodurch beispielsweise Konfiguratoren deutlich schneller arbeiten
Fix: Erkennung von korrekten E-Mail-Adressen für Benutzernamen im Infodialog korrigiert
Fix: Verhindere eine falsche Ziel-ID beim mehrfachen Öffnen eines Ereignisses
Fix: Zeige bei Ereignissen beim Öffnen eine Meldung an, falls die aktuelle Aktion nicht zum Zielobjekt passt
Fix: Update Widget funktioniert nun auch korrekt für Docker armhf/arm64
Fix: Verhindere Fehler im Auswahlbaum, wenn ein Schnellfilter gesetzt ist und neue

Objekte hinzugefügt werden
Fix: Vorschau von Bildern funktioniert nun auch bei Bildern mit Großbuchstaben in der Dateiendung
Fix: SelectCondition wird nun insbesondere in Listen korrekt dargestellt, wenn die betroffene Variable nicht mehr existiert
Fix: Wird man vom Testing-Kanal wieder ausgeladen, so verhält sich der Module Store so als wenn der Kanal geschlossen wäre
Fix: Duplizieren oder Verschieben mit aktivem Schnellfilter funktioniert nun wie vorgesehen
Fix: Keine Fehlermeldungen mehr beim Verschieben des Browserfensters
Fix: Sortierung auf Spalten wie Editieren/Löschen ist nicht mehr möglich
Fix: Besserer Fehler, falls Skripte durch Popup-Blocker o.ä. geblockt werden
Fix: Vorschau von GIFs funktioniert nun
Fix: Bei fehlerhaft geladenen PHP-Modulen wird jetzt eine bessere Fehlermeldung angezeigt
Fix: Bei fehlhaften dynamischen Formularen in Listen von PHP-Module wird eine bessere Fehlermeldung angezeigt
Fix: Beim Öffnen der Konsole werden gemerkte Tabs, die fehlerhaft sind, automatisch geschlossen anstatt irreführende Fehler anzuzeigen
Fix: Tooltip im Skript Editor funktioniert wieder korrekt
Fix: Copy&Paste im Skript Editor funktioniert korrekt, wenn in der Pro Konsole mehrere offen sind

## SYMOS

Neu: IP-Adresse kann ohne vollständiges Recovery geändert werden (Dokumentation, Kapitel 5.7)
Fix: fnmatch Funktion fehlte auf der SymBox
Fix: Beim Neustart von IP-Symcon konnte fälschlicherweise eine Fehlermeldung auftreten, dass IP-Symcon bereits gestartet sei, obwohl dies nicht der Fall war
Fix: WebFronts zu Visualisierungen umgenannt
Fix: Resize der Partition/Dateisystem konnte bei SymBoxen, die in den letzten Wochen verkauft wurden, unvollständig sein. Das SymOS erkennt dieses Problem und fordert zur Korrektur auf. (Die SymBox'en, welche standardmäßig mit einer neueren SymOS Version ausgeliefert werden, kann man gut an dem neuen Sticker mit der Seriennummer/MAC-Adresse auf der Rückseite erkennen)
Fix: Sofern die Integrator Weiterleitung aktiv war, konnte man sich nicht korrekt über SymOS Connect einloggen
Fix: SymOS konnte Abstürzen, wenn Subskription über das Jahr 2038 hinaus ging

## PHP-SDK

Neu: Preview für IPSModuleStrict verfügbar
Neu: Parameter "form" für List/Tree ermöglicht individuelle Bearbeiten-Dialoge (Dokumentation)
Neu: Neues Element QrCode um QR-Codes darzustellen (Dokumentation)
Neu: Ist bei einer List/Tree keine Zeile ausgewählt, werden die add-Werte übergeben
Neu: IPSModuleStrict bietet RegisterHook und RegisterOAuth (damit entfallen die alten Workarounds mit WebHookModule und WebOAuthModule und Einträge von Modulen

werden damit automatisch aufgeräumt)
Neu: Migrate Funktion zum Migrieren der Persistenz (=Attributen/Properties) (Dokumentation)
Neu: Bei Configurator ist eine Kategorie wählbar unter welcher die Objekte erstellt werden sollen
Neu: Parameter „download“ für Elemente mit on-Callbacks um Downloads zu ermöglichen
Neu: Modulname und Aliasse werden anhand der locale.json übersetzt
Fix: visible und enabled funktioniert für SelectCondition nun wie vorgesehen
Fix: Bei dynamischen Formularen in Listen werden „add“ Werte beim Hinzufügen übernommen auch wenn kein dazugehöriges „edit“ Feld vorhanden ist.

### ENTERPRISE

Fix: Sync: Version vom Remote darf älter sein. D.h. der Master in 7.0 kann Remotes in Version 6.4 synchronisieren
Fix: VHost Enterprise Feature funktionierte ggf. nicht korrekt für die iOS App

## NEUE MODULE

### OCPP

Integration von OCPP Geräten direkt in IP-Symcon
Anzeige des Verbrauchs einzelner Ladepunkte
Anzeige, ob gerade eine Transaktion läuft bei Ladepunkten

## AKTUALISIERTE MODULE

### ANWESENHEITS-SIMULATION

Neu: Verwenden von SelectVariable und Hinzufügen einer Status-Spalte

### BETRIEBSSTUNDENZÄHLER

Neu: Preis für Kosten kann aus einer Variable gelesen werden

### GARDENA SMART SYSTEM

Fix: Initiale Verbindung korrigiert
Fix: Überschreiben der Mähdauer korrigiert

### HOME CONNECT

Fix: Endlosschleife bei Wiederverbindung

## JSON EXPORT

Neu: CORS Header eingefügt um Integration in Web Anwendungen zu erlauben

## SZENEN-STEUERUNG

Fix: Szene wird korrekt aufgerufen, auch wenn ein Gerät innerhalb der Szene fehlen sollte

## TELEFONANSAGE

Fix: Fehler beim Starten korrigiert

## VERBRAUCHSVERHALTEN

Fix: Bessere Vorhersage (Betrachtet nur positive Werte)

## WEBGRAPH

Neu: Benutzeroberfläche optimiert
Fix: Referenzsuche, Kontinuierliche monatliche Darstellung im März korrigiert

# V6.3->V6.4 (Q2/2023)

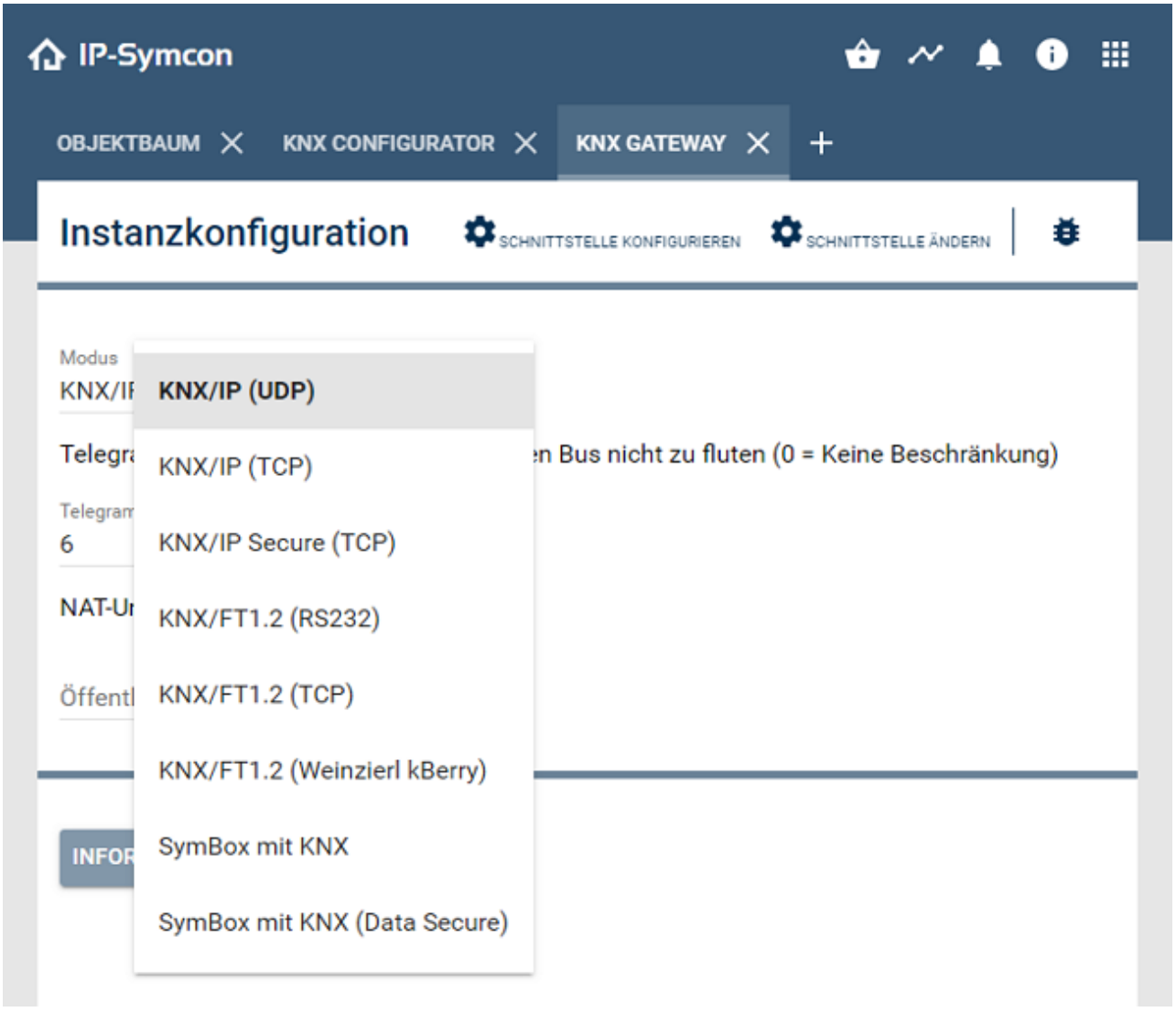
Wesentliche Änderungen
Migration
Liste der Änderungen
Neue und aktualisierte Module

## WESENTLICHE ÄNDERUNGEN

KNX Secure
Übersichtlichere Diagramme mit mehreren Variablen
Automatische Backups via FTP
OPC UA und mehr für Enterprise

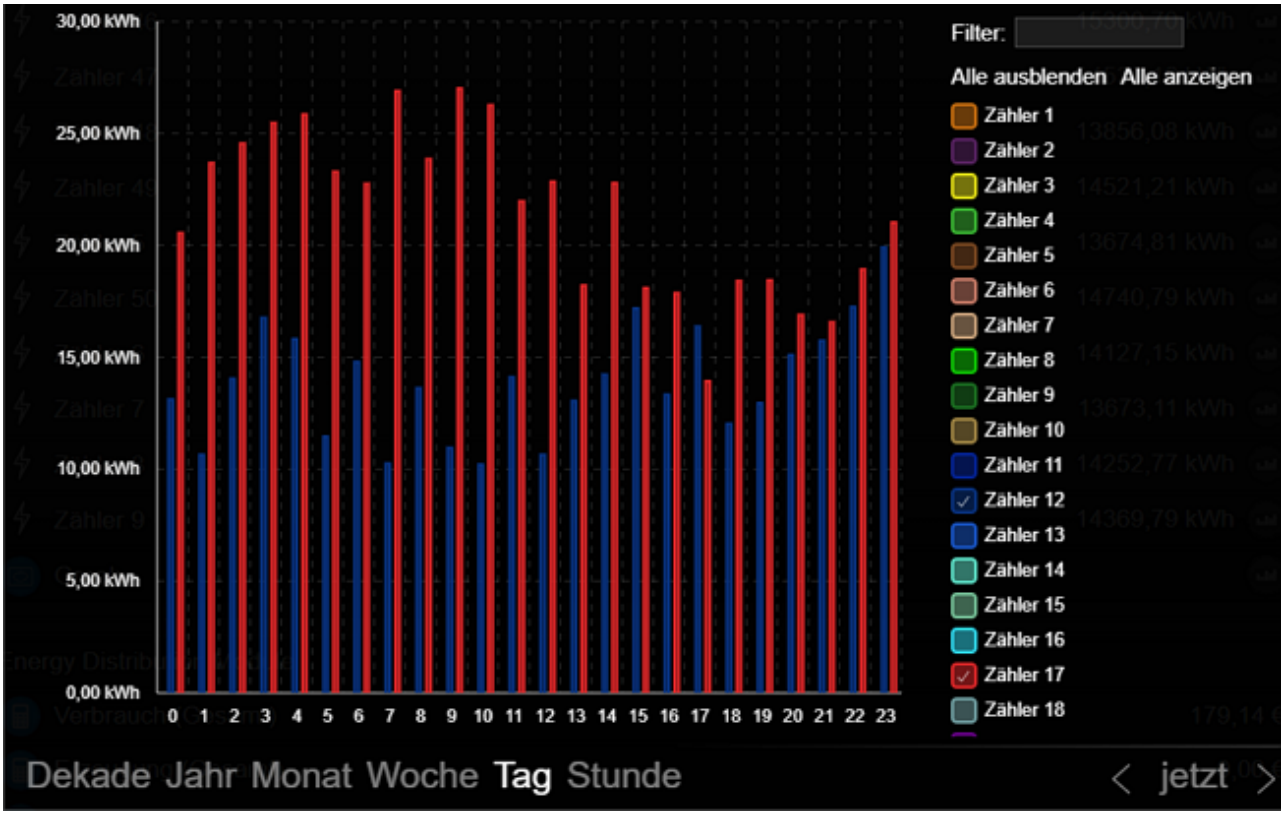
### KNX SECURE

Die Anbindung zu KNX wurde um KNX IP Secure und KNX Data Secure Unterstützung erweitert. So wird die Kommunikation von Visualisierung bis zu den Endgeräten auf IP und TP Ebene verschlüsselt und vor unberechtigtem Zugriff geschützt.



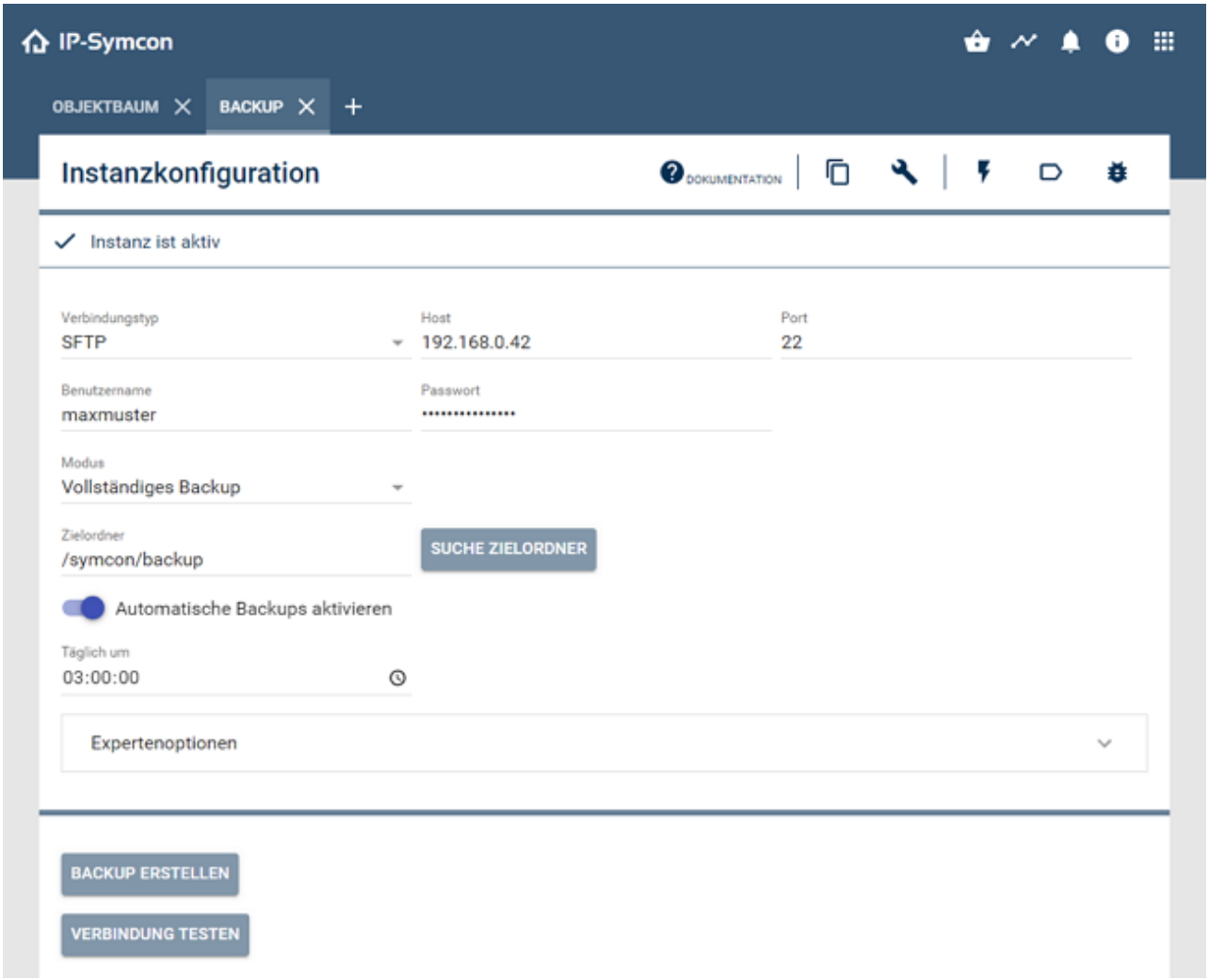
## ÜBERSICHTLICHERE DIAGRAMME MIT MEHREREN VARIABLEN

Die Graphen wurden insbesondere für Graphen mit sehr vielen (z.B. hunderten) Variablen hin optimiert. Werden Variablen ausgeblendet, passen sich die dargestellten Achsen automatisch an die ausgewählten Variablen an und bieten so deutlich mehr Platz für die Verläufe. Darüber hinaus wird die Legende bei vielen Variablen nun als Extraspalte angezeigt und bietet dort diverse hilfreiche Features wie einen Filter oder Buttons zum Ein- oder Ausblenden aller Variablen, wodurch die optimale Auswahl an Variablen unkompliziert und effizient ausgewählt werden kann.



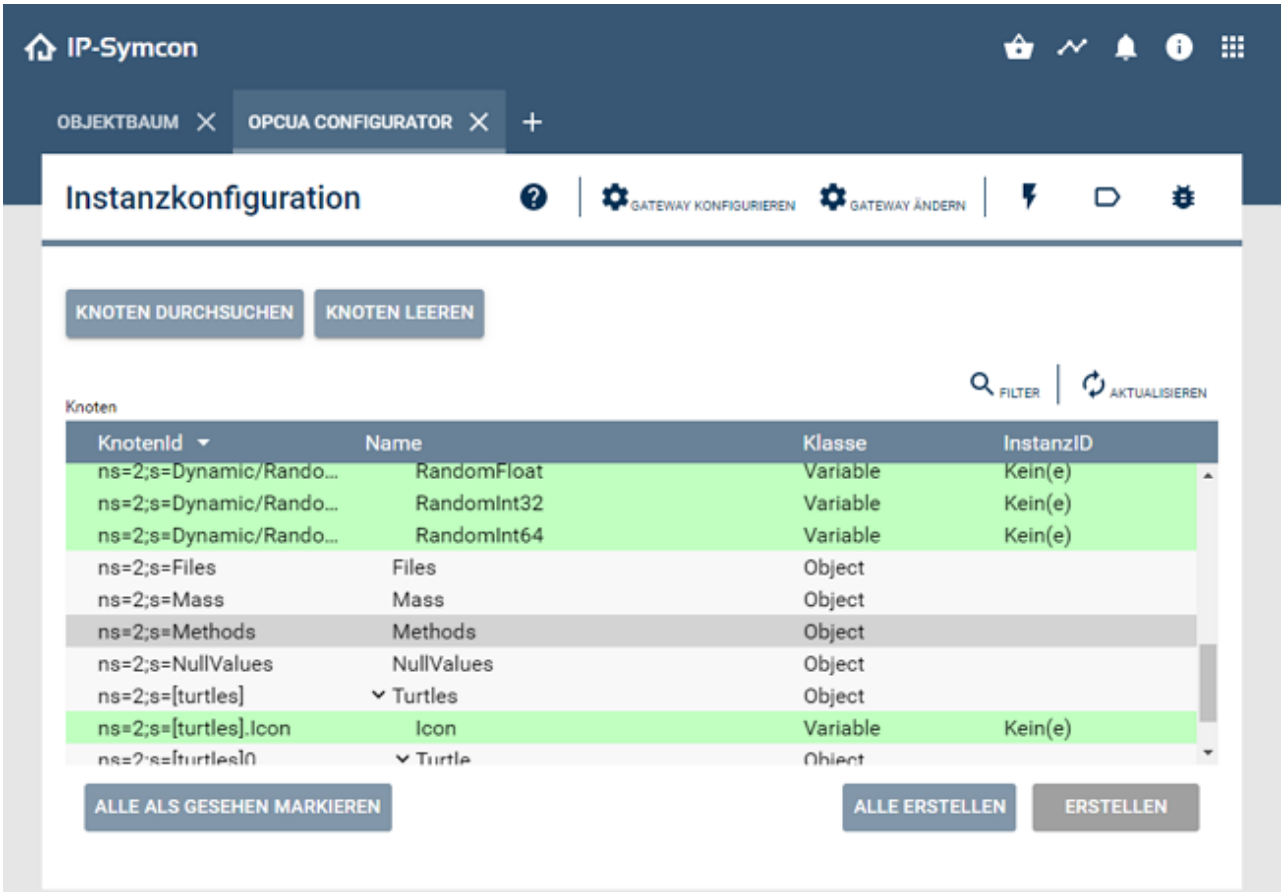

## AUTOMATISCHE BACKUPS VIA FTP

Keine Lust mehr auf manuelle Backups? Dank dem neuen Backup-Modul für FTP/FTPS/SFTP muss gar nicht mehr daran gedacht werden. Das Modul erstellt nach Wunsch vollständige oder inkrementelle Backups und speichert diese typischerweise auf einem NAS-, Cloud- oder externen Speicher, damit es im Falle eines Falles wieder eingespielt werden kann.




## OPC UA UND MEHR FÜR ENTERPRISE

Auch für unsere Enterprise-Kunden gibt es einige spannende neue Features. Nun können Sie OPC UA Geräte einbinden. Oder verwenden Sie die Sync-Module, welche das Zusammenführen von vielen IP-Symcon Systemen in ein Master System unterstützen.




Die Enterprise-Module sind kostenpflichtige Erweiterungen, welche zu jeder bestehenden IP-Symcon Lizenz erworben werden kann. Für Details steht unser Vertrieb zur Verfügung.

## MIGRATION

Bitte erstellen Sie sicherheitshalber ein Backup, sollten wider erwarten Probleme bei der Aktualisierung auftreten.

Falls Sie von einer früheren Version aktualisieren, lesen Sie bitte vorher die passenden Migrationsanleitungen.

Die aktuellste Version kann wie unter Installation beschrieben installiert werden.

## LISTE DER ÄNDERUNGEN

### ALLGEMEIN

Neu: Referenzsuche berücksichtigt IPSView's

Neu: Spezialschalter zum Überschreiben der genutzten Sprache (Locale)
Neu: Funktion zum Migrieren von alten IPSWorkflows
Neu: VoIP_PlayWave validiert die WAV Datei, ob diese Unterstützt wird
Neu: Neues Profil ~Lamella
Fix: VoIP_PlayWave funktioniert wieder korrekt (inkl. PlayFinished Event)
Fix: StartTime wurde auf der SymBox falsch gesetzt, wenn der NTP Server zu langsam ist
Fix: Beim Beenden vom IP-Symcon Dienst warten wir maximal 5 Minuten auf das Beenden von PHP Threads
Fix: Löst das DNS-SD Fehler 87 Problem
Fix: SSDP: Besser Erkennung von Headern mit falscher Groß-/Kleinschreibung
Fix: Bessere Sanity Check für Datumsfuntionen
Fix: Installer überprüft, ob QWave verfügbar ist (Betrifft nur Windows Server 2008/2012)

## AKTIONEN

Neu: Aktionen für Addition/Substraktion bei getriggerten Variablen

## GERÄTE

Neu: KNX IP Secure / KNX Data Secure Unterstützung
Neu: MQTT: Möglichkeit ein abweichendes Topic für Aktionen zu definieren (z.B. mqtt/device/1234/set)
Neu: xComfort Jalousieaktor (Neue Generation mit mehr Feedback auf Kanal 0)
Neu: EnOcean D2-14-41
Neu: OPC UA Client (nur Enterprise)
Fix: KNX: In Profilen wurde „pulses“ nicht korrekt nach „Impulse“ übersetzt
Fix: ModBus: DeviceID ist im Formular auf 255 limitiert
Fix: SetPoint Berechnung beim Eltako FTR65DS, FTR65HS, FUTH55D, FUTH65D korrigiert

## WEBFRONT

Neu: Diagramme laden nur noch die wirklich ausgewählten Charts (Damit können z.B. auch 500 Charts in einem Diagramm drin sein und es werden nur die ausgewählten geladen)
Neu: Diagramme merken sich im WebFront, welche Auswahl in der Legende getroffen wurde
Neu: Legenden von Diagrammen enthalten auch Boolean Graphen und können somit ausgeblendet werden
Neu: Listenbasierte Legende enthält Buttons für Alle ausblenden/Alle einblenden
Neu: Umschalten zwischen Schwebender/Listenbasierter Legende kann über die WebFront Visualisierung Instanz vorgenommen werden im Bereich Darstellung
Fix: WebFront startet wieder automatisch korrekt bei Neustarts
Fix: Bessere Erkennung von einem Verbindungsabbruch (insbesondere auf Safari Mobile)

## KONSOLE

Neu: Bedingungen in Listen werden passend formatiert
Neu: Ändern sich die Werte in listenartigen Elementen einer Instanzkonfiguration bleibt die aktuelle Scrollposition erhalten
Neu: Speichere im Statuswidget welche Informationen dargestellt werden
Neu: Variablen mit nicht mehr existierenden Variablenaktionen werden als fehlerhaft

markiert
Neu: Beim Versuch einen veralteten IPSWorkflow zu öffnen wird eine Migration ins aktuelle Format angeboten
Fix: Stelle Tooltips von langen Strings im Objektbaum korrekt dar
Fix: Verwende bei der Konfiguration von Diagrammen für Balkengraphen den korrekten Typ für Profile
Fix: Seltener Fehler, dass der Snapshot nicht geladen werden konnte (wenn mehrere Konsolen offen waren)
Fix: Mausklicks im Skripteditor landen wieder an der richtigen Position

### SYMOS

Neu: NTP Einstellungen verbessert, sodass NTP Server auch per DHCP empfangen werden können
Neu: Connect: Neuer Spezialschalter CheckInterval zum Ändern des Prüfintervalls
Neu: Bei Logfiles kann mit „Neuste“ direkt an Ende gesprungen werden
Neu: Unter „Support → System anzeigen“ wird dargestellt, wenn eine RTC in der SymBox verbaut ist
Fix: Der Zurück Buttons bei Logfiles funktioniert wieder korrekt
Fix: KNX/ETS: Programmieren der physikalischen Adresse inkl. Applikation korrigiert
Fix: SymOS Oberfläche ich nun korrekt über IPv6 erreichbar
Fix: Interner Server Error beim Download von Logfiles korrigiert

### PHP-SDK

Neu: Werden Tree und Configurator als Variable im Konfigurationsformular verwendet, beinhaltet nun jeder Knoten auch die Information „expanded“
Neu: Der Callback „onExpand“ für Tree und Configurator wird ausgelöst, wenn ein Knoten ausgeklappt wird
Fix: Ist ein Zahlenfeld im TestCenter leer wird nicht mehr auf einen ungültigen Wert geschaltet
Fix: Timer Intervall beim PHP SDK wird geprüft, sodass dieser nicht negativ wird
Fix: Bessere Fehlermeldung, wenn ein Button einen ungültigen Link zurückgibt
Fix: Bei der Erstellung von Instanzen werden OpenObjectButtons immer deaktiviert, damit die Konfiguration der Kette nicht unterbrochen wird
Fix: loadValuesFromConfiguration funktioniert wieder wie vorgesehen innerhalb von ExpansionPanels
Fix: SelectValue verursacht beim Öffnen der Konfiguration keine Fehler mehr und reagiert wie vorgesehen auf Änderungen

### ENTERPRISE

Neu: Remote Sync (siehe: https://www.youtube.com/watch?v=njqcztynZjA)
Neu: Replication Sync (siehe: https://www.youtube.com/watch?v=njqcztynZjA)
Neu: Translation Modul kann Übersetzung direkt über DeepL vornehmen
Neu: Je WebFront können ein/mehrere Domains (VHosts) definiert werden, für die das WebFront angezeigt werden soll

## NEUE MODULE

### BACKUP (FTP/FTPS/SFTP)

Vollständige Backups jeden Tag zu einer gewünschten Uhrzeit
Optional auch Inkrementelle Backups (sodass das Backup immer nur aktualisiert wird)
Beliebige Ordner können gefiltert werden
Variable zur Uhrzeit des letzten erfolgreichen Backups
Variable zur letzten Backupgröße

## AKTUALISIERTE MODULE

### SPOTIFY

Neu: Funktion SPO_MakeAPIRequest um eigene Anfragen an die Spotify-API zu tätigen
Fix: Verhindere eine fälschliche Annahme, dass die Wiedergabe aktiv ist, sollten die Rückgabewerte der API problematisch sein

### CSV ZIP EXPORT

Neu: Besserer Download und Download von Rohdaten

# V6.2->V6.3 (Q4/2022)

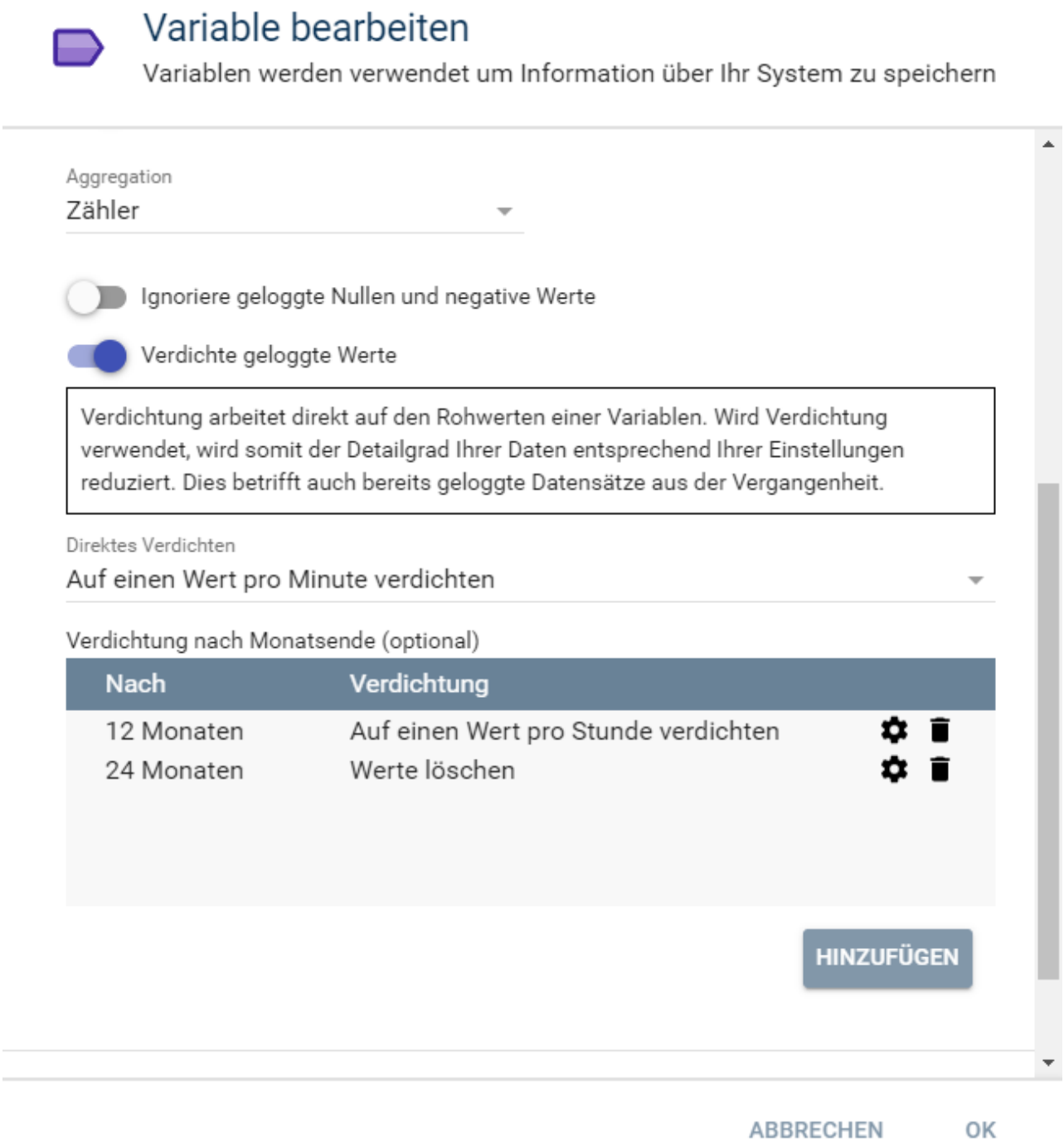
Wesentliche Änderungen
Migration
Liste der Änderungen
Neue und aktualisierte Module

## WESENTLICHE ÄNDERUNGEN

Automatisches Verdichten von Archivdaten
Alle Graphentypen miteinander kombinierbar
Integration von IPSViewDesigner und IPSWorkflows
BACnet Server

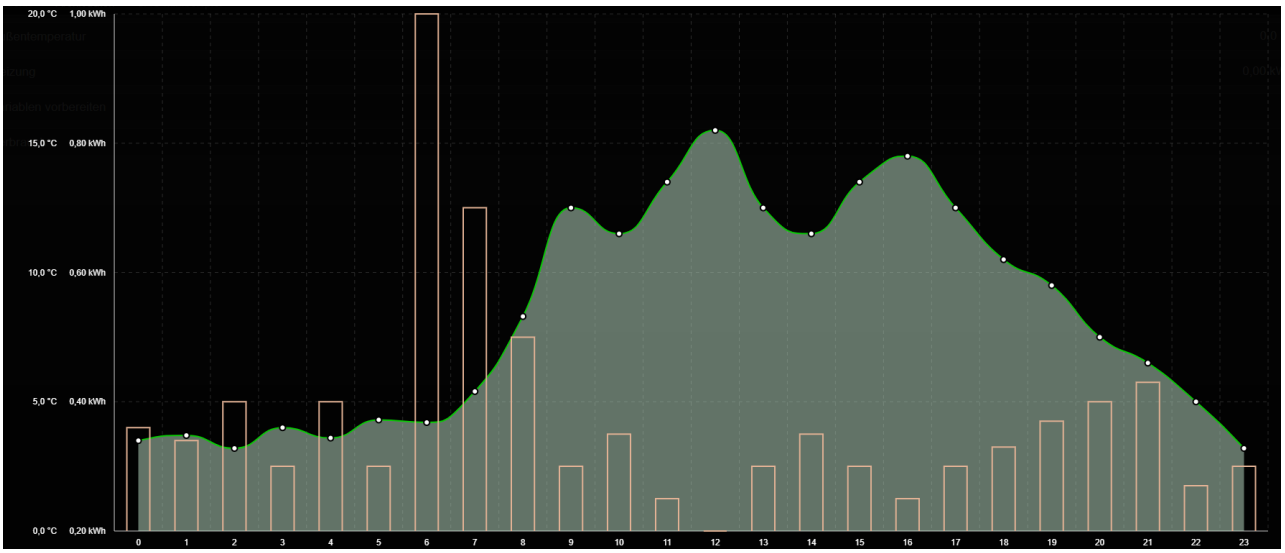
### AUTOMATISCHES VERDICHTEN VON ARCHIVDATEN

Das Logging von Variablendaten ist wichtig um diverse Rückschlüsse zu ziehen und beispielsweise den Energieverbrauch zu verstehen und dadurch zu optimieren. Diverse Sensoren schicken hierbei hochauflösende Daten, wobei teilweise mehr als ein Datensatz pro Sekunde empfangen und im Archiv geschrieben wird. Dieser hohe Informationsgehalt schlägt sich natürlich im Speicherplatz nieder, obwohl die hohe Auflösung meisten Daten nach einer gewissen Zeit uninteressant werden. So reicht nach einem Jahr vielleicht ein Temperaturwert pro Stunde vollkommen aus und Werte, die älter als zwei Jahre sind, sind komplett uninteressant. Hierbei unterstützt die Verdichtgung, welche nach individuell einstellbaren Zeitabständen die Datensätze auf den gewünschten Detailgrad runterbricht oder auch komplett löscht. Dies kann pro Variable eingestellt werden, wodurch für jeden Fall die optimale Verdichtung gewählt werden kann.



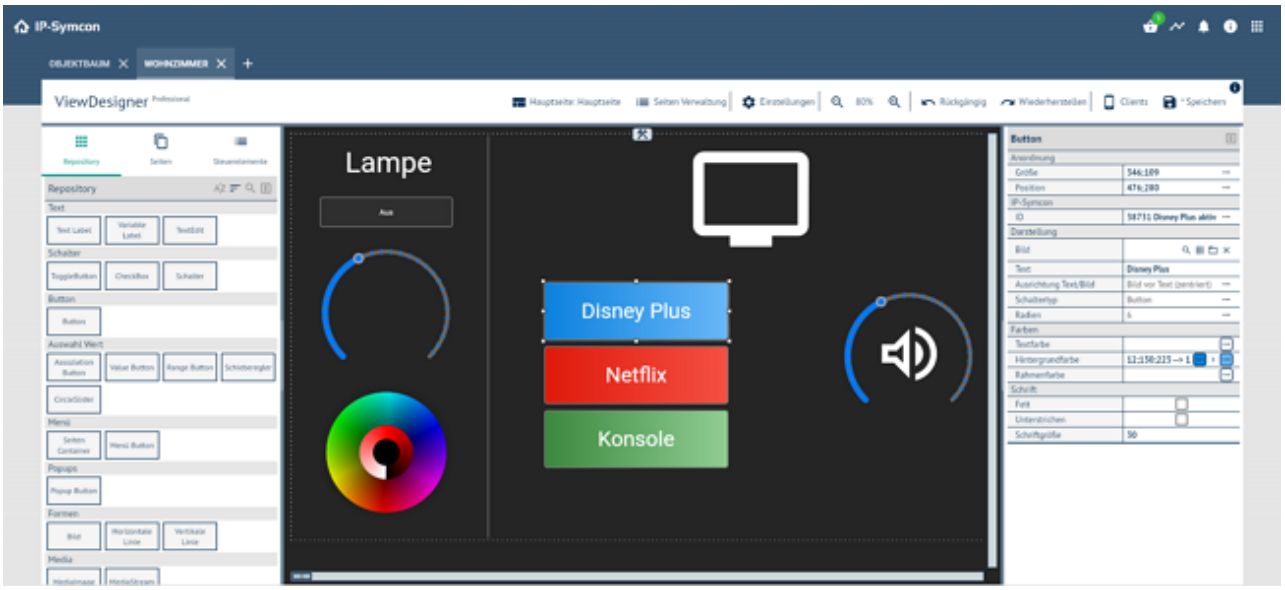

## ALLE GRAPHENTYPEN MITEINANDER KOMBINIERBAR

Wie viel Energie verbraucht die Heizung in Abhängigkeit der Außentemperatur? Dank der neuen Kombination aller Graphentypen miteinander in einem einzelnen Diagramm, kann dies nun optimal zusammengeführt werden. Dabei wird der Zähler der Heizung wie gewohnt als Balkengraph und die Temperatur als Liniengraph dargestellt. In welchen Zimmern sind die Heizkörper aktiv? Erweitern Sie das Diagramm gerne auch noch um Booleangraphen um alle Informationen an einer Stelle zu präsentieren.




## INTEGRATION VON IPSVIEWDESIGNER UND IPSWORKFLOWS

Kein Bedarf mehr für spezielle Anwendungen zu IPSStudio. Seit einiger Zeit können IPSViews bereits im WebFront dargestellt werden. Mit der neuen Version zieht nun auch die Konfiguration in die Konsole ein. So können IPSViews und Workflows plattformunabhängig direkt in der Konsole bearbeitet werden.




Zur Verwendung von IPSStudio in der Konsole ist eine Professional Lizenz für IPSStudio erforderlich.

## BACNET SERVER

IP-Symcon kann nun nicht nur als BACnet Client, sondern auch als BACnet Server agieren. Dadurch kann IP-Symcon als Bridge agieren und Werte anderer Systeme an BACnet weiterleiten oder auch eigene Parameter zur Verfügung stellen. So können beliebige Werte in IP-Symcon anderen BACnet Clients zur Verfügung gestellt werden.

Das BACnet Modul ist eine kostenpflichtige Erweiterung, welche zu jeder bestehenden IP-Symcon Lizenz erworben werden kann. Für Details wenden Sie sich bitte an unseren Vertrieb.

## MIGRATION

Bitte erstellen Sie sicherheitshalber ein Backup, sollten wider erwarten Probleme bei der Aktualisierung auftreten.

Falls Sie von einer früheren Version aktualisieren, lesen Sie bitte vorher die passenden Migrationsanleitungen.

Die aktuellste Version kann wie unter Installation beschrieben installiert werden.

## LISTE DER ÄNDERUNGEN

### ALLGEMEIN

Neu: Verbesserte Sicherheit: Auch bei Anmeldungen über Localhost wird ein Kennwort erfordert, sofern der Fernzugriff aktiviert ist. Dies betrifft insbesondere Windows User, die nun ggf. ein Kennwort eingeben müssten.
Neu: Neue Version der IPSWorkflows wird unterstützt
Neu: Neue Version vom IPSViewDesigner wird unterstützt
Neu: Besserer Hinweis, wenn IP-Symcon initial von extern eingerichtet werden soll
Neu: Intern auf std::filesystem und std::variant (von C++20) umgestellt
Neu: Windows Installer setzt die IP-Symcon Version in den Datei-Eigenschaften
Neu: Windows Version kann auch unter Windows 10 (ARM64) installiert werden (ohne Gewähr und Support!)
Neu: Bibliothek zum Berechnen von Datum/Uhrzeit ausgetauscht
Neu: Bibliothek zum Lesen/Schreiben der php.ini ausgetauscht
Neu: Intern wurde alles auf Int64 umgestellt. Dadurch können Int64 Werte (insbesondere KNX DPT 29.x) vollständig dargestellt und archiviert werden. (Benötigt ein 64-Bit System. d.h. nicht für Raspberry Pi 32 Bit verfügbar. Dort ist zumindest PHP weiter auf 32 Bit limitiert)
Neu: Spezialschalter CompatibilitySloppyUTF8 der standardmäßig an ist, um die Konvertierung von nicht UTF-8 Konformen String wie bisher durchzuführen
Neu: CC_GetTrafficCounter um den exakten Verbrauch vom Connect Dienst der 24 Stunde Periode zu ermitteln
Neu: IPS_GetKernelDirSpace, um den freien Speicherplatz vom Datenlaufwerk anzuzeigen
Neu: Der Docker Tag testing ist ab sofort als Multi-Arch Image für amd64, arm64 und armhf verfügbar
Neu: GetValueFormatted/GetValueFormattedEx geben den unformatierten Wert auch ohne Profil zurück
Neu: Das Tray-Icon unter Windows öffnet bei Doppelklick die Pro Konsole. Der

Verbindungsassistent ist separat im Kontextmenü verfügbar
Neu: SSE: Unterstützung für fragmentierte Events
Neu: IP-Symcon kann mit Einschränkungen auch ohne Root Rechte gestartet werden
Neu: IPS_ExecuteEx liefert Rückgabewert der Ausführung, wie auch IPS_Execute dies tut
Fix: Bessere Fehlermeldung im Logfile/Lizenz Ändern Dialog, wenn Subskription aufgrund einer veralteten Lizenz nicht ermittelt werden kann.
Fix: Bessere SSDP Suche unter Windows mit mehrere Netzwerkadaptern
Fix: Bessere Fehlermeldung, wenn Icons von defekten Skins nicht geladen werden konnten
Fix: Das Tray-Icon unter Windows sollte nicht mehr verschwinden
Fix: UTF-8 Problem im MQTT Konfigurator gelöst
Fix: UTF-8 Problem im Util Control gelöst
Fix: UTF-8 Problem mit Skriptausgaben von Ereignissen gelöst
Fix: Register Variable war ggf. falsche Fehlermeldung aufgrund fehlender Initialisierung einer internen Variable
Fix: Tray-Icon unter Windows prüft, ob der Dienst korrekt läuft, bevor er die Pro Konsole öffnet
Fix: Include-Pfad wurde unter Windows nicht korrekt gesetzt für Ereignisse mit ScriptText
Fix: Veraltete Spezialschalter werden nicht mehr angezeigt
Fix: Weiteres UTF-8 Problem im Util Control gelöst
Fix: IPS_Execute konnte (seit 4.0) keine Argumente mit Zeilenumbrüchen weiterreichen
Fix: Util Control zeigt die gefundenen Fehlerstellen für Umlaute/ShortTags wieder korrekt an
Fix: Referenzsuche/Skriptsuche kommt mit ungültigem UTF-8 besser klar
Fix: Weiteres UTF-8 Problem im MediaPlayer (Windows) korrigiert
Fix: Weitere UTF-8 Problem bei HID (Windows) korrigiert
Fix: Boolean (Variable) auf Boolean (Variable) Vergleich in bedingten Ereignissen korrigiert
Fix: Sicherheitslücke im WebServer korrigiert, welche nur mit direktem Netzwerkzugriff oder einer Port Weiterleitung ins Internet ausgenutzt werden kann

## ARCHIV

Neu: Verdichten der Werte ist verfügbar!
Fix: Beim Entfernen aller Variablen aus dem Archiv wurde dies nicht korrekt abgespeichert
Fix: Bei einem Neustart konnten aktuelle Aggregationsdaten von Variablen mit Standardaggregation leicht abweichen. Dies wurde behoben.
Fix: Alle Archiv-bezogenen Dateioperationen werden mehrmals versucht, um Timing Probleme mit Backup-Software unter Windows zu umgehen.

## AKTIONEN

Neu: Neue Aktion für den Ablaufplan zum Nachführen den Wochenplan-Ereignisses
Neu: Neue Aktion zum Senden von Push-Nachrichten mit Inhalt aus einer Variable
Fix: Aktionen mit potentiell nicht gesetzten Objekten, z.B. Push-Nachrichten, werden wieder korrekt formatiert
Fix: Archiv-Aktion konnte Werte für die Jahresaggregation nicht korrekt abrufen

## GERÄTE

Neu: BACnet Server (Beliebige Variablen können im BACnet zur Verfügung gestellt werden)
Neu: KNX: DPT 1.024, 8.012, 12.1200, 12.1201, 13.1200, 13.1201, 13.016, 14.1200, 14.1201 hinzugefügt

Neu: Eltako FWWKW71L
Neu: Update Button in Konfiguration für OneWire, ModBus/Siemens Instanzen
Neu: I/O SSE: SSL-Zertifikat Prüfung kann deaktiviert werden
Neu: LCN Konfigurator zeigt Seriennummer, Firmware und Hardware an
Neu: LCN Quitterungen werden strikter erwartet/ausgewertet
Neu: Z-Wave: PHP-Befehle ZW_RequestPriorityGroup, ZW_UpdatePriorityGroup und ZW_UpdatePriorityReturnGroup
Neu: KNX DPT 9.030
Neu: MQTT: Neues Format Binär, welches binären Payload als HEX in String-Variablen speichert (Umgeht korrekt Probleme mit nicht validem UTF-8 in Strings)
Neu: BACnet: Relinquish Present Value Button direkt in der Instanz
Fix: BACnet: String im UCS-2 Format werden korrekt konvertiert
Fix: KNX: Einige Profile waren nicht korrekt UTF-8 kodiert
Fix: HomeMatic: Bessere Fehlermeldung beim Verbindungsaufbau
Fix: KNX Konfigurator funktioniert korrekt, auch wenn nicht unterstützte DPTs in der XML existieren
Fix: Kleine Anpassung den WebHooks, sodass diese mit OCPP kompatibel sind
Fix: SSE: Accept Header wird korrekt auf text/event-stream gesetzt, sodass z.B. die Hue Bridge kein application/json sendet
Fix: RegisterVariable: Interne Warteschlange wird auf 512kB limitiert
Fix: OZW: Bei mehreren Geräten konnten die DataPoint Instanzen dem falschen Splitter zugeordnet werden
Fix: OZW: Setze Wert auf 0, falls ---- empfangen wird (anstatt einer Fehlermeldung im Logfile)
Fix: ModBus Konfigurator konnte nicht geöffnet werden, wenn „fremde“ Instanzen am Splitter hingen
Fix: KNX DPT 12.1200, 12.1201, 13.1200, 13.1201 korrigiert
Fix: EnOcean: Eltako WWLW → Emuliere Status korrigiert
Fix: BACnet: Fehlermeldung beim Öffnen der Discovery Instanz gelöst (seit letztem Release)
Fix: BACnet: Nicht unterstützte NSDU Nachrichten werden nur im Debug Log gemeldet
Fix: MQTT: Ungültiges UTF-8 im Payload wird korrekt an PHP-Module weitergegeben
Fix: KNX Konfigurator mit mehreren IP-Gateway hat die Instanzen ggf. an das falsche Gateway verbunden

## WEBFRONT

Neu: Bar/Line/Bool Graphen können alle in einem Diagramm kombiniert werden
Neu: Exakte Werte können via Hover auch für HD/RAW Werte abgelesen werden
Neu: IE11 wird nicht mehr korrekt unterstützt und wird als inkompatibler Browser identifiziert
Neu: Unterstützung einer listenbasierten Legende inkl. Filter sobald 10 oder mehr Graphen in einem Diagramm sind
Neu: Unterstützung für den neuen „Initial sichtbar“ Parameter bei Diagrammen
Fix: Legende vom Graphen mit gleicher MedienID aber anderem Offset werden korrekt getrennt
Fix: SVG: Werte-Overlays waren für einige Diagramme mit Achse auf der rechten Seite vertauscht
Fix: ~Door Profil hatte ein vertauschtes Icon
Fix: Sofern alle WebFronts unsichtbar waren, gab es eine kryptische Fehlermeldung

## APP

Fix: ~Door Profil hatte ein vertauschtes Icon

## KONSOLE

Neu: Module Store überarbeitet
    Module, die nur in anderer Sprache verfügbar sind, können dargestellt werden
    Zeige Text an, wenn keine Module gefunden wurden
Neu: IPS-Views und Workflows können direkt in der Konsole erstellt und bearbeitet werden
Neu: Optimiere Dialog zur Aktionsauswahl
Neu: Button um Mitteilungen im Status-Tab zurückzusetzen
Neu: In Objektauswahlfeldern kann bei sicherer Verbindung (HTTPS oder Pro Konsole) via Strg + C eine ObjektID kopiert und via Strg + V eingefügt werden
Neu: Beim Springen zu einem Objekt wird die Auswahl im Objektbaum nur aktualisiert, falls das Objekt noch nicht ausgewählt wurde. So bleibt eine mögliche Mehrfachauswahl erhalten
Neu: Scrolle beim „Gateway ändern“-Dialog zum aktuell ausgewählten Gateway
Neu: „Gateway ändern“-Dialog leicht visuell überarbeitet
Neu: Erhöhe Performance durch Verwendung von Lizenz- und Serverdaten aus dem Snapshot anstatt immer wieder anzufragen
Neu: Erhöhe Performance des Status Widget
Neu: Bessere Fehlermeldung, wenn nach der Lizenzänderung ein erneutes Einloggen notwendig ist
Neu: Dialog für Spezialschalter verbessert (Zeigt nicht übernommene Änderungen und Standardwerte an)
Neu: Meldung, falls der Speicherplatz knapp wird
Neu: Wird mit dem Schnellfilter im „Instanz erstellen“-Dialog nichts gefunden werden, kann direkt der Module Store geöffnet werden
Neu: Zeige Hinweis, falls einer der DebugDisable* Spezialschalter aktiv ist
Neu: Bei Diagrammen kann angegeben werden, ob ein Chart standardmäßig aktiv sein soll
Neu: „Passwort vergessen“-Button beim Login
Fix: Zeile Zeilenumbrüche in Beschreibungen im Module Store wie vorgesehen an
Fix: Visualisierungen können im Objektbaum nicht mehr verschoben werden
Fix: Verhindere, dass Zeilen bei inaktiven Tabs beim Wechsel des Kompaktmodus verschwinden
Fix: Verbesserte Fehlermeldung bei ungewöhnlichen Fehlern beim Verschieben von Objekten
Fix: Werden alle Dialoge geschlossen, beispielsweise von einem Konfigurator in einem Popup, so können diese wie vorgesehen danach wieder geöffnet werden
Fix: Bei Listen in Dialogen wird nun die Scrollposition korrekt initialisiert und es kommt nicht zu „leeren“ Listen
Fix: Da Firefox die Clipboard-API im Browser nicht vollständig unterstützt, wird dies nun überprüft und die Fehlermeldung entsprechend angepasst
Fix: Bei langen Strings wird der Variableninhalt im Dialog zum Variable bearbeiten nicht mehr doppelt angezeigt
Fix: Beim Bewegen eines Elements per Drag & Drop stellt der Zeiger nun korrekt dar, dass es sich um eine Bewegen-Aktion und nicht um eine Kopieren-Aktion handelt
Fix: Bei einer Anpassung der Graphenreihenfolge in einem Diagramm wird dies korrekt abgespeichert
Fix: Die Standardaktion wird nicht mehr modifiziert, wenn dies nicht explizit im Dialog getan wird
Fix: Zeige keine Warnung im Visualisierungs-Widget, falls keine Lizenz gesetzt ist
Fix: Wird ein Ereignis erstellt, wird wie erwartet im Objektbaum dahin gesprungen
Fix: Wird eine Liste oder ein Baum erneut geöffnet, wird die aktuelle Scrollposition

aktualisiert. Dies verhindert den Fehler, dass die Profilauswahl unter bestimmten Umständen leer war
Fix: Wird bei zyklischen Ereignissen das Tagesmuster angepasst, so werden auch dazugehörige Änderungen korrekt übernommen, auch wenn die Standardeinstellungen beibehalten werden
Fix: Fordere nur eine Reaggregation aufgrund von veränderter Ausdünnung, wenn entsprechende Einstellung tatsächlich angepasst wurde
Fix: Verhindere kosmetische Fehlermeldung im Profildialog
Fix: Verhindere Fehler, beim duplizieren von Profilen und direktem Löschen von Assoziationen
Fix: Verhindere Fehler beim initialen Rendern eines Objektbaums unter Firefox
Fix: Liefert ein Konfigurator oder eine Discovery Instanz bei der Gerätesuche ungültige Werte, löst dies keinen Fehler aus
Fix: Unter Firefox wurden Listen in seltenen Fällen leer dargestellt und erst beim Scrollen aktualisiert
Fix: Mehr Informationen, wenn die „Timestamp is missing“ Fehlermeldung kommt
Fix: Zeige Listen in Dialogen wieder korrekt an, z.B. Ereignisliste von Skripten oder Auswahl der Systeme bei der Gerätesuche
Fix: Dialog zum prüfen im Konfigurator behoben

## PRO-KONSOLE

Neu: Kleine Anpassungen für den neuen IPSViewDesigner
Fix: Werden die Skripteinstellungen in einem abgedockten Skripteditor geändert, wird dies für alle Fenster übernommen
Fix: Kopieren von ObjektIDs aus Auswahlfeldern in der Pro Konsole führte zu einem Fehler

## SYMOS

Neu: Kernel auf 5.10.110 aktualisiert
Neu: SymOS: Einrichten der Lizenz speichert das Lizenzdatum und zeit es bei „Lizenz anzeigen“ an
Neu: Aktuelle IP-Adresse kann über Support → System anzeigen angefragt werden

## PHP-SDK

Neu: SelectValue Element (Dokumentation)
Neu: Feld „discoveryInterval“ für Configurator ermöglicht eine Einschränkung der automatischen Gerätesuche
Neu: Standardwert für SelectObject und dessen Derivate ist nun 1, also „Nichts ausgewählt“
Fix: Existiert das Ziel in einem SelectAction nicht mehr, so wird dies passend dargestellt
Fix: Zeige SelectColor in List oder Tree über dem Grau der ausgewählten Zeile an
Fix: Wird ein Icon im SelectIcon ausgewählt, wird wie vorgesehen die „Änderungen übernehmen“-Zeile eingeblendet
Fix: Bessere Fehlermeldung, wenn eine verschachtelte Liste kein gültiges JSON ist
Fix: Verschachtelte Listen haben nun in PHP-Funktionen wie vorgesehen Zugriff auf $id
Fix: Soll ein PopupAlert geöffnet werden während der Tab mit der Instanz nicht aktiv ist, wird das Popup erst geöffnet sobald der Tab wieder aktiv wird
Fix: Spalten mit SelectDate/SelectTime/SelectDateTime in List und Tree werden korrekt sortiert

## NEUE MODULE

### SNMP

Offizielles SNMP Modul der Symcon GmbH. Unterstützt SNMP V1/V2/V3 und kann alle OIDs eines Geräts per Walk anzeigen. Per Checkbox können die gewünschten Werte als Variablen in IP-Symcon erstellen werden, welche dann zyklische aktualisiert werden.

## AKTUALISIERTE MODULE

### AMAZON ALEXA

Fix: Verbesserte Fehlermeldungen

### ANWESENHEITSSIMULATION

Fix: In seltenen Fällen wurde die Simulation fälschlicherweise gestoppt

### BENACHRICHTIGUNG

Neu: Unterstützung für neues Telegram Bot Modul

### GOOGLE ASSISTANT

Neu: Link um den Skill bei Google zu bewerten
Fix: Geräte, die 30 Minuten keine Updates schicken, werden nicht mehr als offline angenommen

### RECHENMODULE

Fix: Fehlende Konfigurationselement wieder hinzugefügt

### REPORT MODUL

Neu: Temperature Variable ist optional

### SNMP

Neues Modul im Module Store

### SPOTIFY

Fix: Gehe korrekt mit abgelaufenen oder falschen Token um

Fix: Verhindere Fehler bei deaktivierter Wiedergabe

## TELEFONANSAGE

Neu: Benutze das neue SelectModule Element

## VIRTUELLE MESSSTELLE

Neu: Verbesserte Berechnung

# V6.1->V6.2 (Q2/2022)

## WESENTLICHE ÄNDERUNGEN


Gleichzeitiges Editieren
Verbesserter Module Store
Verbesserte Performance
Neue Module von der Symcon GmbH

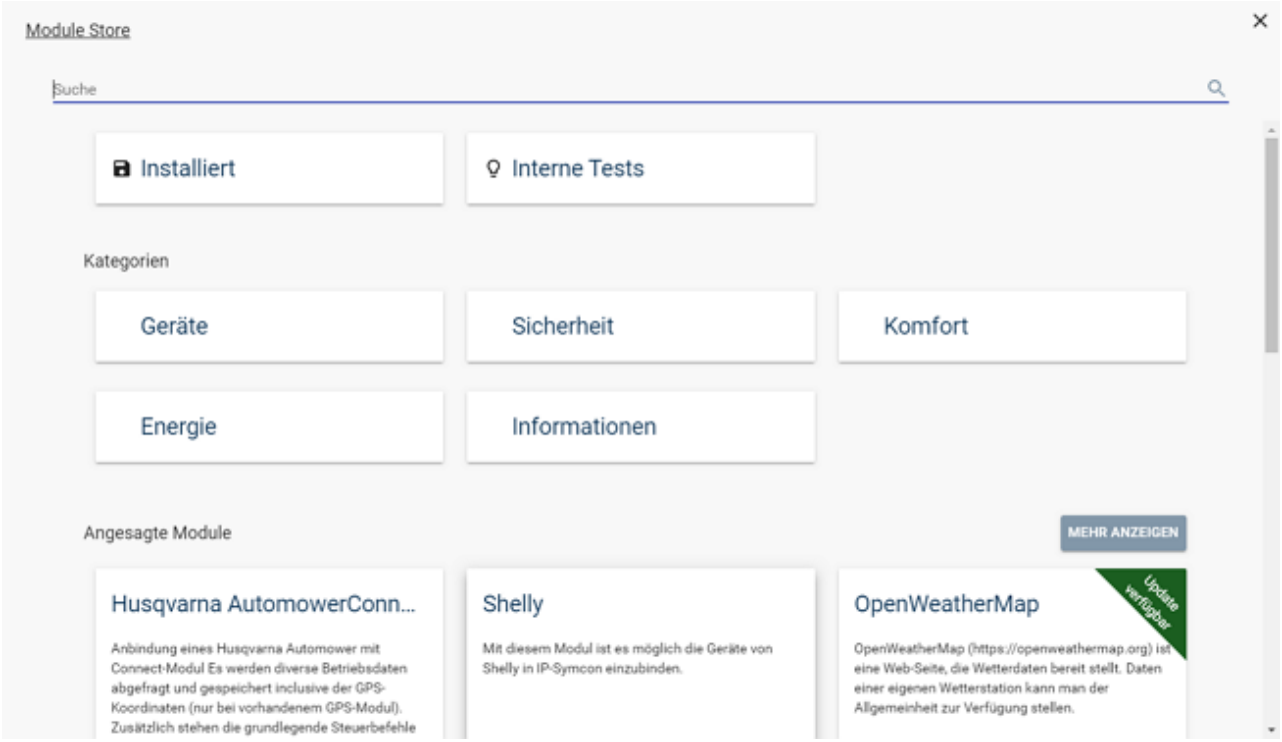

### GLEICHZEITIGES EDITIEREN

Mit der neuen Version können in der Konsole beliebig viele Objekte gleichzeitig bearbeitet werden. So kann beispielsweise schnell und unkompliziert das Logging für viele Variablen aktiviert oder die Sichtbarkeit im WebFront für viele Objekte gleichzeitig konfiguriert werden. So gelangt man bei einer großen Menge von Operationen mit viel weniger Klicks ans Ziel.




### VERBESSERTER MODULE STORE

Der Module Store wurde an diversen Stellen verbessert und erweitert. So weist der Module Store nun auf den ersten Blick auf das Icon auf verfügbare Updates hin und ermöglicht auch eine direkte Installation aller Updates. Dazu kommt eine Anzeige verschiedener interessanter Modullisten: Welche Module sind angesagt? Welche Module sind ganz neu oder haben gerade erst ein Update erhalten? Schließlich ist nun zusätzlich zum Stable und Beta Kanal auch der Testing Kanal verfügbar. Dieser ermöglicht es Entwicklern gezielt Tester für interne Modultests einzuladen. Dann können nur eingeladene Benutzer über den Module Store die Testing-Version des Moduls installieren.




### VERBESSERTE PERFORMANCE

Zur 6.2 haben wir an diversen Stellen noch weitere Verbesserungen der Performance durchgeführt. Werden Objektbäume mit mehreren Tausend Objekten geöffnet, so geschieht dies nun nahezu sofort. Auch das Ausklappen von Kategorien mit extrem vielen Kindern geht nun noch schneller als jemals zuvor. Auch der Start von IP-Symcon wurde beschleunigt. Werden die Variablen ins Archiv geladen, so werden alle Prozessorkerne parallel ausgenutzt, wodurch IP-Symcon nach einem Start schneller zur Verfügung steht. Schließlich wurde der MQTT Server optimiert und kann nun performant mit viel mehr Geräten gleichzeitig umgehen.

### NEUE MODULE VON DER SYMCON GMBH

Auch bei der Entwicklung von Modulen hat sich einiges getan. So sind einige neue Module im Module Store zu finden bzw. Erweiterungen zu existierenden Modulen:

- Betriebsstundenzähler (Dokumentation)
- Virtuelle Messstelle (Dokumentation)
- CSV Zip Export (Dokumentation)
- Variablen Vergleich (Dokumentation)
- Verbrauchsverhalten (Dokumentation)
- Virtueller Zähler (Dokumentation)
- Symcon Report (Erweitert um den Energie Report, Dokumentation )
- JSON Exporter (Dokumentation)

## MIGRATION

Bitte erstellen Sie sicherheitshalber ein Backup, sollten wider erwarten Probleme bei der Aktualisierung auftreten.

Falls Sie von einer früheren Version aktualisieren, lesen Sie bitte vorher die passenden Migrationsanleitungen.

Es wird unter Linux mindestens Ubuntu 20.04 erfordert
Es wird beim Raspberry Pi mindestens Raspbian OS (Bullseye) erfordert

## LISTE DER ÄNDERUNGEN

### ALLGEMEIN

Neu: MessageQueue wird parallel abgearbeitet und verbessert die Latenz, wenn z.B. PHP Module diese vorher zu lange blockiert hatten
Neu: Datenfluss für PHP-Module optimiert (sollte CPU-Last leicht senken)
Neu: Datenfluss-Filter (RegEx) von PCRE zu RE2 gewechselt (wesentlich bessere Performance)
Neu: Archiv aktualisiert Aggregationen beim Start parallel
Neu: Beim Ausführen von Skripten werden PHP-Module erkannt und als solche mit SenderID angegeben
Neu: IP-Symcon prüft, ob die Subskription für die IP-Symcon Version noch gültig ist
Neu: ServerSocket unterstützt IPv6 sofern verfügbar
Fix: Wir prüfen jetzt vor dem Installieren, ob die Linux Version aktuell genug ist und brechen ansonsten mit einer passenden Fehlermeldung ab
Fix: Fehlermeldung beim Verlinken von access_id.log korrigiert
Fix: Prüfung, ob Raspbian Bullseye / Ubuntu 20.20 oder neuer installiert ist funktioniert wieder korrekt
Fix: Prüfung der Demo-Lizenz/Subskription war nicht vollständig korrekt
Fix: Einige veraltete Aktionen wurden unter Windows nicht aufgeräumt
Fix: Abstürze mit HomeKit auf älteren Macs
Fix: URL von ip-symcon.de auf symcon.de 1 in den UPNP Anfragen angepasst
Fix: Speicherleck unter Windows bei SSL-Verbindungen gelöst
Fix: Seltenes Speicherleck bei RTSP Streams und Verbindungsabbrüchen gelöst
Fix: Hoher Speicherverbrauch (durch zu viel Caching) bei vielen PHP-Threads und hohen Speicherverbrauch in Skripte (z.B. durch große IPSView Dashboards) gelöst
Fix: Durch eine Umstellungen von GitHub müssen wir alle git:// Skins/Module zu https:// konvertieren (Betrifft nur Skins und alte Module)
Fix: Bei zu schnellem Shutdown von IP-Symcon konnte dies am Ende abstürzen
Fix: Sys_GetCPUInfo lieferte unter Windows 0 beim ersten Aufruf

### AKTIONEN

Neu: Mathematische Funktionen können nun sowohl mit einem konstanten als auch dem Wert einer anderen Variable verwendet werden

## GERÄTE

Neu: W&T unterstützt CO2 Sensoren
Neu: BACnet: Datenpunkt to Geräten werden einer gerätespezifischen Kategorie erstellt
Neu: MQTT: Button in den Konfiguratoren, der erlaubt die Liste zu leeren
Fix: KNX: Verwerfe Pakete von alten Verbindungen
Fix: MQTT: Client gibt Hinweis, wenn ein Neu-Verbinden nach einer Änderung erforderlich ist
Fix: ModBus: String-Werte filtern korrekt Null-Characters am Ende
Fix: BACnet: Absturz bei einigen Geräten, wenn beim Anfragen der Objektliste nicht wie erwartet die Länge gemeldet wurde
Fix: BACnet: Lade Objektliste in einem Stück, sofern das Gerät dies unterstützt
Fix: KNX: Button zum Aktivieren des Programmiermodus auf der SymBox funktioniert wie erwartet
Fix: Gerätesuche für unsere LAN-Gateway ist schneller
Fix: IRTrans: Konfigurationsseite konnte bei Fehlerhafter Verbindung nicht geöffnet werden
Fix: KNX: Problem beim Trennen der Verbindung beim Beenden von IP-Symcon gelöst
Fix: KNX: DPT 19: Wochentag wurde falsch gesendet, wenn RequestAction genutzt wurde
Fix: MQTT: Fehlerhafte Verbindungen werden besser/schneller getrennt, damit sich (z.B. die Shelly's) korrekt neu verbinden können

## KONSOLE

Neu: Store wurde überarbeitet
    Module werden nun alphabetisch sortiert
    Eine grüne Zahl am Module Store Icon zeigt an, wenn Updates vorhanden sind
    Button um alle installierten Module gleichzeitig zu aktualisieren
    Spezialkategorien für Top Downloads, Angesagt Module, Neueste Updates und Neueste Module
    Unterstützung der Spezialkategorie für Testing (allerdings kann aktuell noch nicht im persönlichen Bereich eingeladen werden)
Neu: Mehrfachauswahl wird nun bei allen Operationen im Objektbaum unterstützt, insbesondere auch bei „Objekt bearbeiten"
Neu: Zeige Dateipfad in PHP Informationen an
Neu: Verbesserte Performance für alle Arten von Bäumen und Listen in der Konsole
Neu: Mehrere bewegbare Dialoge werden leicht versetzt geöffnet
Neu: Doku-Widget bietet nun einen Button, um die Dokumentation direkt zu öffnen
Neu: Filter im Archiv
Neu: Aggregationstyp wird im Icon im Archiv angezeigt
Neu: Zeige Profiltyp im Profil-Manager an
Fix: Beim Löschen von Variablen aus Diagrammen wird dies korrekt gespeichert
Fix: In offenen Skripten suchen schlägt nicht mehr fehl, wenn manche der Skripte während der aktuellen Session noch nicht ausgewählt wurden
Fix: Verhindere fälschliche Anzeige, dass der Browser veraltet ist, falls die Verbindung langsamer ist
Fix: Lokalisierte Darstellung von Kommazahlen im Archiv und „Variable bearbeiten"-Dialog
Fix: Das Achievement-Widget (Aprilscherz 2020) verursacht keine Fehler mehr bei neu erstellten Objekten, die direkt wieder gelöscht werden

Fix: Wird der Name eines Ereignisses geändert, wird dies direkt im Objektbaum dargestellt
Fix: Verhindere fälschliche geforderte Reaggregation bei neu erstellten Variablen
Fix: Variablen werden in Dialogen wie dem Auswahlbaum einer auslösenden Variable eines Ereignisses korrekt zentriert dargestellt
Fix: Kann eine Subskription aufgrund einer fehlenden Internetverbindung nicht geprüft werden, wird dies dem Benutzer mitgeteilt
Fix: Listen wurden manchmal leer dargestellt und erst beim Scrollen korrekt dargestellt

## PRO-KONSOLE

*Keine Änderungen*

## SYMOS

Neu: Kernel auf 5.10.92 aktualisiert
Neu: IPv6 ist verfügbar (noch nicht aus SymOS heraus konfigurierbar)
Neu: Anzeige der SymBox Version auf der Support Seite mit Speicher/RAM Größe
Neu: Zeige Compute Module Version auf der Support/System-Seite
Neu: Backups werden mit dem Format Jahr/Monat/Tag anstatt Tag/Monat/Jahr erstellt für eine bessere Sortierbarkeit
Neu: „Backup wiederherstellen“ Button direkt auf der Backup-Seite
Fix: Einspielen von Backups ist auch ohne Internet-Verbindung wieder möglich
Fix: Beim Zurücksetzen auf Werkseinstellung wird auch der Update-Kanal korrekt zurückgesetzt
Fix: Logfile-Ansicht funktioniert wieder korrekt, sobald man auf „Mehr“ klickt
Fix: VoIP auf der SymBox wieder verfügbar
Fix: Backups über 2GB können korrekt wiederhergestellt werden
Fix: Backups im Safari werden als zipbackup heruntergeladen, um nicht automatisch entpackt zu werden

## PHP-SDK

Neu: „Nichts ausgewählt“ entspricht bei SelectObject und dessen Derivaten nun dem Wert 1, wodurch die Hauptkategorie mit der ID 0 bewusst ausgewählt werden kann
Neu: Flag „deprecated“ für veraltete Aktionen um Sie korrekt in ihre Nachfolger zu konvertieren
Fix: PHP-Module konnten Abarbeitung von KERNELSTARTED im MessageSink behindern

# V6.0->V6.1 (Q1/2022)

## WESENTLICHE ÄNDERUNGEN

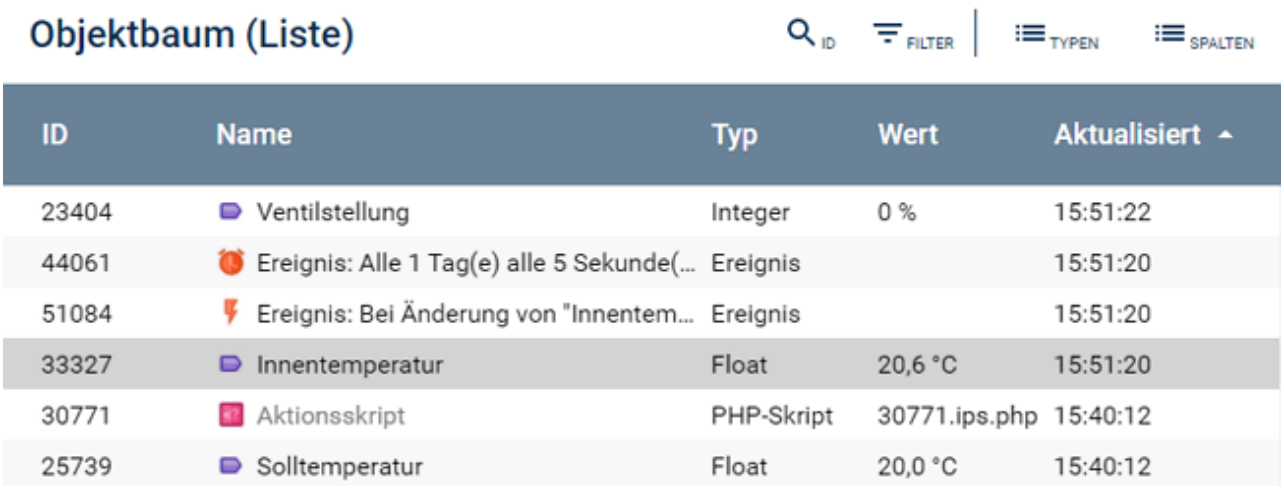

Mehr Performance
Objektbaum in Listenansicht
Gerätesuche in Sprachassistenten
SymOS mit 64 Bit


### MEHR PERFORMANCE

Die Performance von allen Baum- und Listendarstellungen in der Konsole wurde noch einmal deutlich verbessert. Werden tausende Objekte gleichzeitig dargestellt oder Kategorien mit mehr als 1000 Unterobjekten geöffnet, läuft dennoch alles flüssig weiter. Und auch wenn sich an allen Stellen im Objektbaum gleichzeitig etwas ändert, stört dies die Nutzung nicht durch Verzögerungen oder Ruckler, es kann einfach flüssig weiter gearbeitet werden.

### OBJEKTBAUM IN LISTENANSICHT

In der Listenansicht des Objektbaums können alle Objekte in IP-Symcon gleichzeitig dargestellt werden. Damit kehrt die klassische Ansicht aus der Legacy-Konsole in die aktuelle Konsole zurück und ermöglicht eine Sortierung der Objekte nach beliebigen Spalten. Damit kann schnell gesehen werden, welche Objekte im Baum häufig aktualisiert werden oder welche Instanzen fehlerhaft oder inaktiv sind.




### GERÄTESUCHE IN SPRACHASSISTENTEN

Alle Sprachassistenten bieten nun eine Gerätesuche. So müssen nicht mehr alle Geräte einzeln eingepflegt werden. Stattdessen wird die Gerätesuche einmal gestartet und entdeckt passende Geräte. In der Auswahl kann dann schnell ausgewählt werden, welche dieser Geräte tatsächlich im Sprachassistenten integriert werden sollen und schon ist der Assistent blitzschnell bereit zur Verwendung.




### SYMOS MIT 64 BIT

IP-Symcon und das SymOS nutzen auf der SymBox jetzt das volle Potential der 64-Bit-Architektur. Dadurch läuft IP-Symcon noch performanter und ermöglicht noch größere Projekte auf der SymBox.

### WEBINAR ZUM THEMA IP-SYMCON 6.1

## MIGRATION

Bitte erstellen Sie sicherheitshalber ein Backup, sollten wider erwarten Probleme bei der Aktualisierung auftreten.

Falls Sie von einer früheren Version aktualisieren, lesen Sie bitte vorher die passenden Migrationsanleitungen.

# LISTE DER ÄNDERUNGEN

## ALLGEMEIN

Neu: access_*.log Dateien vom WebServer werden automatisch rotiert
Neu: Receive-/Forward-DataFilter in PHP-Modulen können größer als 16kb sein und sollten wesentlich performanter ausgewertet werden
Neu: Profile Watt und Electricity.Wh
Neu: IPS_GetSystemLanguage, um die erkannte Systemsprache auszugeben
Neu: Spezialschalter zum Debuggen (Deaktiviert jeweils GetSnapshot, GetSnapshotChanges, WebSocket Rückkanal, Streams)
Neu: Spezialschalter, um RTSP Streams über TCP statt UDP zu empfangen
Neu: Sofern "MessageQueueWatch" aktiv ist, werden Statistiken pro Instanz geführt (IPS_GetInstanceMessageStatistics())
Neu: Statistiken für Anzahl der Nachrichten im Datenfluss
Neu: Einige Bibliotheken aktualisiert
Fix: max_execution_time ebenfalls für ini_set deaktiviert
Fix: Google Assistant Request Limit wird bei vielen Geräten korrekt erweitert, wenn die GA Instanz erst nach der Connect Control Instanz geladen wurde
Fix: Notification Control unterstützt Views von IPSView wieder korrekt
Fix: IP-Symcon wird korrekt gefunden, wenn ein & im Namen der Hauptkategorie stand
Fix: Ablaufpläne mit Wartezeit blockieren nicht mehr beim Herunterfahren
Fix: Zeitzone Etc/UTC (Default auf Ubuntu) startet korrekt
Fix: Beim Start werden unvollständige Objekte aus dem Snapshot entfernt
Fix: Limitierung bei RTSP/MJPEG Streams greift jetzt korrekt und verhindert massiven Speicherverbrauch, sofern Clients keine Daten mehr abgenommen haben
Fix: Beim Ändern von „Erweiterten Bedingungen“ mit Variablenvergleichen wurden die neue Werte als 0 statt dem eigentlich Wert in der Settings gespeichert, sodass nach einem Neustart die „Erweiterte Bedingung“ falsch konfiguriert war
Fix: Dekodieren von Absendern bei POP3 Nachrichten korrigiert (Bei IMAP bereits seit 2016 korrekt)
Fix: DNS-SD: Unter Windows konnte in seltenen Fällen sehr viele gleiche Fehlermeldungen auftreten
Fix: Löschen von Skripten/Medien legt diese wieder im „deleted“ Ordner ab
Fix: WebFront: Sofern Diagramme im WebFront sind und diese Variablen enthalten, welche Unterobjekte besaßen, wurden diese in seltenen Fällen nicht im WebFront angezeigt
Fix: FlowWatchTxLimit/FlowWatchRxLimit/MessageQueueWatchLimit kann mit 0 deaktiviert werden

## ARCHIV

Fix: Falls es Daten aus der Zukunft gibt, üblicherweise durch eine Korrektur der Uhrzeit, so werden diese bei der minütlichen Aggregation einfach nicht angezeigt anstatt Fehler zu verursachen

## AKTIONEN

Fix: Ereignisse mit fehlerhafter Aktion werden korrekt im Log protokolliert anstatt die Abarbeitung mit einer Exception zu beenden

## GERÄTE

Neu: HomeMatic: Konfigurator zeigt Geräte als Baum an, sodass die Kanäle unterhalb angezeigt werden
Neu: HomeMatic: Datenfluss unterstützt Komplexe Werte (als JSON-kodiert)
Neu: HomeMatic: Datenfluss unterstützt Anfrage für HMScript
Neu: HomeMatic: Konfigurator/Socket kann Gruppen (Port 9292) auslesen
Neu: HomeMatic: Konfigurator zeigt gute Gerätenamen an
Neu: Technische Alternative CMI: Vollständige Unterstützung von Protokoll-Version 6.0
Neu: Eltako: Schalter für Serie 62 werden besser Unterstützt (Eigene Instanz)
Neu: BACnet: Unterstützung für MultiState
Neu: BACnet: COV kann ein-/ausgeschaltet werden
Neu: BACnet: Priorität beim Senden kann eingestellt werden
Neu: BACnet: BACNET_RelinquishPresentValue zum „Ausnullen“ es vorher gesendeten Wertes
Neu: BACnet: Strings mit ISO 8859-1 Kodierung werden unterstützt
Neu: KNX: Status-GA Erkennung kann nun auf verschiedenen Ebnen suchen
Neu: KNX: Für Weinzierl kBerry und SymBox mit KNX kann der Programmiermodus für die physikalische Adresse aus der Gateway Instanz aktiviert/deaktiviert werden
Neu: LCN: Warten auf Quittierungen ist möglich und standardmäßig aktiv
Neu: Z-Wave: Schönere Positionierung von Buttons in Popups der Konfiguration
Neu: ModBus: Option zum Byteweise verdrehen von Strings
Neu: ModBus: Optionen werden je nach gewähltem Datentyp angezeigt/versteckt
Neu: W&T unterstützt CO2 Sensoren
Fix: KNX: Fehlermeldung vom expliziten Lesen von Werten auf dem Bus, sofern die GA unter „Mehr?“ definiert war
Fix: BACnet: Priorität wird standardmäßig als 8 (Handbetrieb) gesendet
Fix: BACnet: Daten vom Typ Enum werden korrekt gesendet
Fix: BACnet: Fehler im Geräte-Konfigurator sofern keine Objekte verfügbar sind
Fix: BACnet: Forwarded-NPDUs werden im BVLC Header unterstützt
Fix: BACnet: Ignoriere WhoHas Service Requests, auf welche wir nicht antworten müssen
Fix: Elatako FAHB130 sollte Daten korrekt empfangen
Fix: Z-Wave: NodeID wird bei Nonce Requests angezeigt
Fix: MQTT: Korrigiert „Packet was not fully consumed“ Fehlermeldungen, welche bei abgebrochenen Verbindungen auftreten konnten
Fix: digitalStrom: Workaround für Bug in dSS Firmware 1.19.2
Fix: Z-Wave Konfigurator fragt Node Informationen seltener ab
Fix: ModBus: Auslesen von Strings wurde Byteweise verdreht
Fix: BACnet: Unterstützung von Geräte mit UCS-2 Kodierung in Strings
Fix: JSON Decoder kommt mit Sonderzeichen im Identifikator zurecht
Fix: KNX: DPT 3.x kann korrekt geschaltet werden

## I/OS

Neu: WebSocket Client kann benutzerdefinierte Header mitsenden
Fix: HTTP/WebSocket Bibliothek aktualisiert

## KONSOLE

Neu: Performance von Listen/Bäumen mit vielen Elementen drastisch erhöht
Neu: Objektbaum kann als Listenansicht dargestellt werden
Neu: Die Gerätesuche informiert auch über deaktivierte/fehlerhafte Konfiguratoren/

Discoveries
Neu: In Bedingungen können Variablen mit Variablen verglichen werden
Neu: Objekte, die neu erstellt wurden und deren Elternobjekt direkt geändert wird, flackern nicht mehr in der Hauptkategorie
Neu: Diagramme vom Typ Bool, Standard, Zähler können alle miteinander kombiniert werden
Neu: Prüfe im Profilmanager, ob die Assoziationswerte von Float-Profilen zu den eingestellten Nachkommastellen passen
Neu: Logik beim Duplizieren von Aktionen im Ablaufplan optimiert
Neu: Wird im Eingabefeld der Iconauswahl etwas eingegeben, so öffnet sich eine Autovervollständigung und bietet passende Icons an
Neu: Das Icon von Variablen mit Aktion wird dunkler dargestellt als von Variablen ohne Aktion
Fix: Assoziationen von Stringprofilen werden in Erstellungsreihenfolge angezeigt
Fix: Schöne Fehlermeldung für den „out-of-memory“ Bug von Firefox (Version 92 bis 93)
Fix: Werden Zeiten im Events- oder Timertab sortiert, kommt „Nie“ unabhängig von der Sortierrichtung immer nach unten
Fix: Wird ein Objekt unter sich selbst dupliziert, wird eine Endlosschleife verhindert
Fix: Spezialkategorien werden wie vorgesehen gefiltert
Fix: Für alle Tastenkombinationen wird das Cmd von Macs als Strg gewertet
Fix: Aktualisiere verwendete Bibliotheken
Fix: Kontextmenü kann unter Safari wie vorgesehen mit Strg + Klick geöffnet werden und schließt sich nicht direkt wieder
Fix: Fehler behoben, wenn die Position mehrerer Objekte gleichzeitig aktualisiert wird (typischerweise ein Objekt und ein Link darauf)
Fix: Behebe Fehler, wenn bei einer erweiterten Bedingung initial ein String gewählt wird
Fix: Eingaben für Link-Parameter nutzen die volle verfügbare Breite
Fix: Bessere Fehlermeldung, wenn die Verbindung abbricht
Fix: Verhindere Fehler beim Verschieben von Aktionen im Ablaufplan auf die oberste Ebene
Fix: Typo in „Autor“ im Instanz-Dialog
Fix: Wird bei der Betrachtung von Archivwerten das Datum manuell eingegeben, wird das Datum erst ausgewertet sobald der Wert gültig ist
Fix: Beim Hinzufügen von Bedingungen mit prozentualen Variablen wird der neue Wert echt gerundet und so meistens der eingegebene Wert auch angezeigt wird
Fix: Bei sehr vielen Instanzen kommt es während der Gerätesuche nicht mehr zum Überlauf

## PRO-KONSOLE

Neu: Die Breite von Dialogen kann angepasst werden und wird auch gespeichert
Neu: Upgrade auf neuste Bibliotheken
Neu: MacOS wird für als Universal App für Intel/Apple Silicon erstellt
Neu: „Versionsinformationen kopieren“ Button
Fix: Beim Öffnen der Konsole, öffnet sich diese ohne ein flackerndes Zwischenfenster

## SYMOS

Neu: Wechsel auf 64-Bit für SymOS und IP-Symcon

## PHP-SDK

Neu: Parameter „color“, „bold“, „underline“ und „italic“ für Label

Neu: Erweiterte Datenfluss-Funktionen SendDataToParentEx und SendDataToChildrenEx mit der alternative Pakete versendet werden können (wovon die Ziel-Instanz nur das erst-beste Verarbeitet)
Neu: Parameter poup.closeCaption für PopupButton
Neu: Parameter popup.buttons für PopupButton und PopupAlert
Neu: Parameter des Popups von PopupButton und PopupAlert können abgesehen von items dynamisch geändert werden
Neu: Wird innerhalb von einer Sekunde nach einer Schaltung im TestCenter der neue Wert nicht von der Variable bestätigt, setzt sich der dargestellte Wert zurück
Neu: Element SelectIcon zur Icon-Auswahl
Neu: Ist bei einem Image onClick gesetzt, so wird das Mauszeiger über dem Image zum Zeiger
Fix: Lazy-Container (ExpansionPanel) stellen Ihren Inhalt auch vor dem ersten Ausklappen bereit
Fix: Hat eine Liste nicht die volle Breite, wird der „Hinzufügen“-Button dennoch rechts unter der Liste angezeigt
Fix: „Falsey“ Werte für SelectAction werden als „Keine Aktion“ interpretiert
Fix: SetBuffer bei PHP-Module prüft und limitiert den Speicherverbraucht auf max. 1 MB pro Instanz. Das Limit pro Buffer liegt bei 512kB.

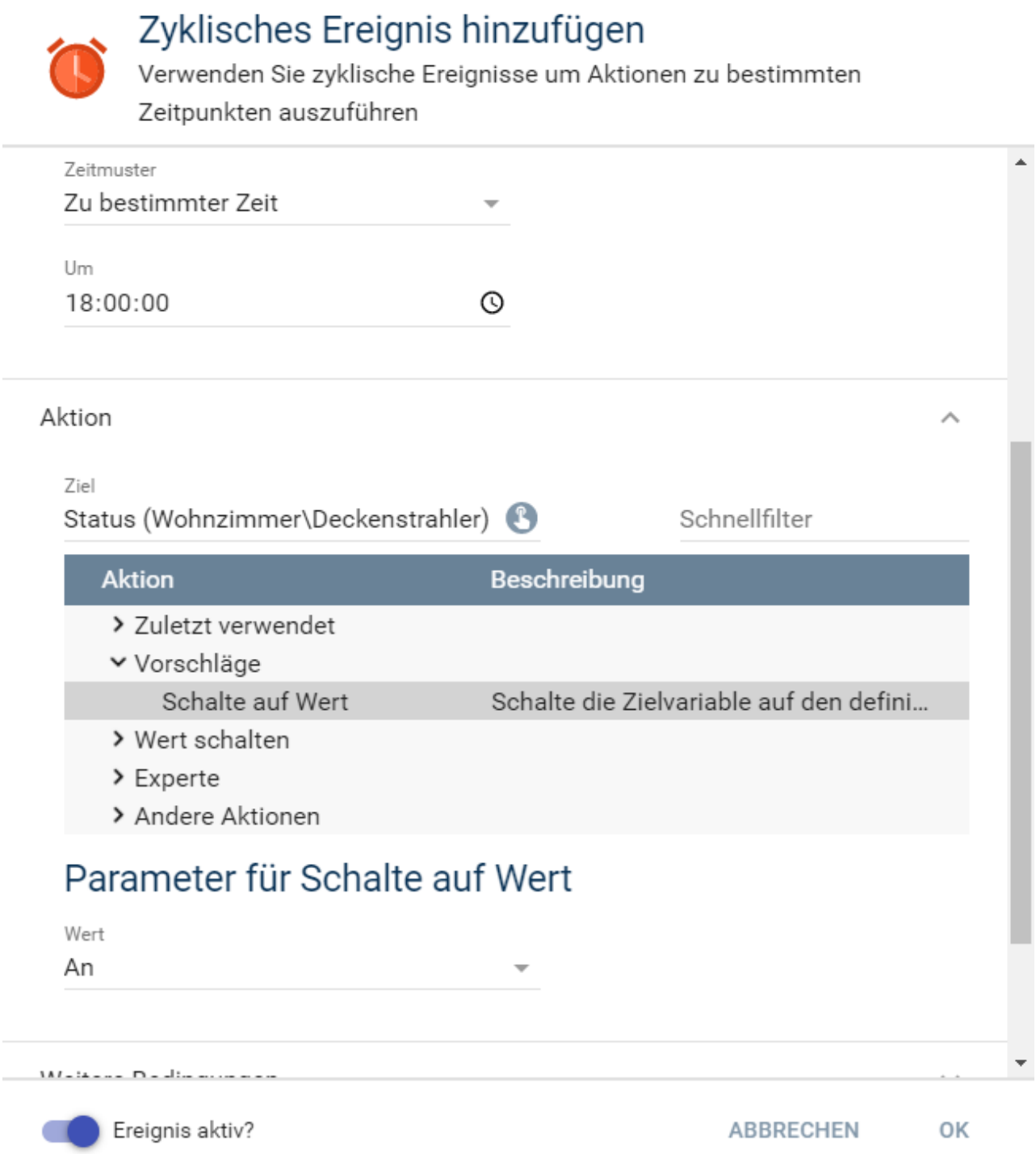
# V5.5->V6.0 (Q3/2021)

## WESENTLICHE ÄNDERUNGEN

Aktionen
Ablaufpläne
Gardena Smart System und Home Connect
Erweiterte Unterstützung von Betriebssystemen

### AKTIONEN

Als neuer Baustein zur Automation stehen jetzt Aktionen zur Verfügung. Diese ermöglichen die Bedienung und Konfiguration aller Objekttypen. Aktionen werden primär von Ereignissen und den neuen Ablaufplänen verwendet. So kann unkompliziert auf die individuellen Funktionen einer Instanz zugegriffen werden, wie das Verschicken von Push Nachrichten über das WebFront oder E-Mails via SMTP. Es können Ereignisse aktiviert oder alle Lampen innerhalb einer Kategorie eingeschaltet werden. Die gewohnte Individualisierbarkeit von IP-Symcon ermöglicht es auch Entwicklern eigene Aktionen zu gestalten, welche zu ihren Modulen passen.



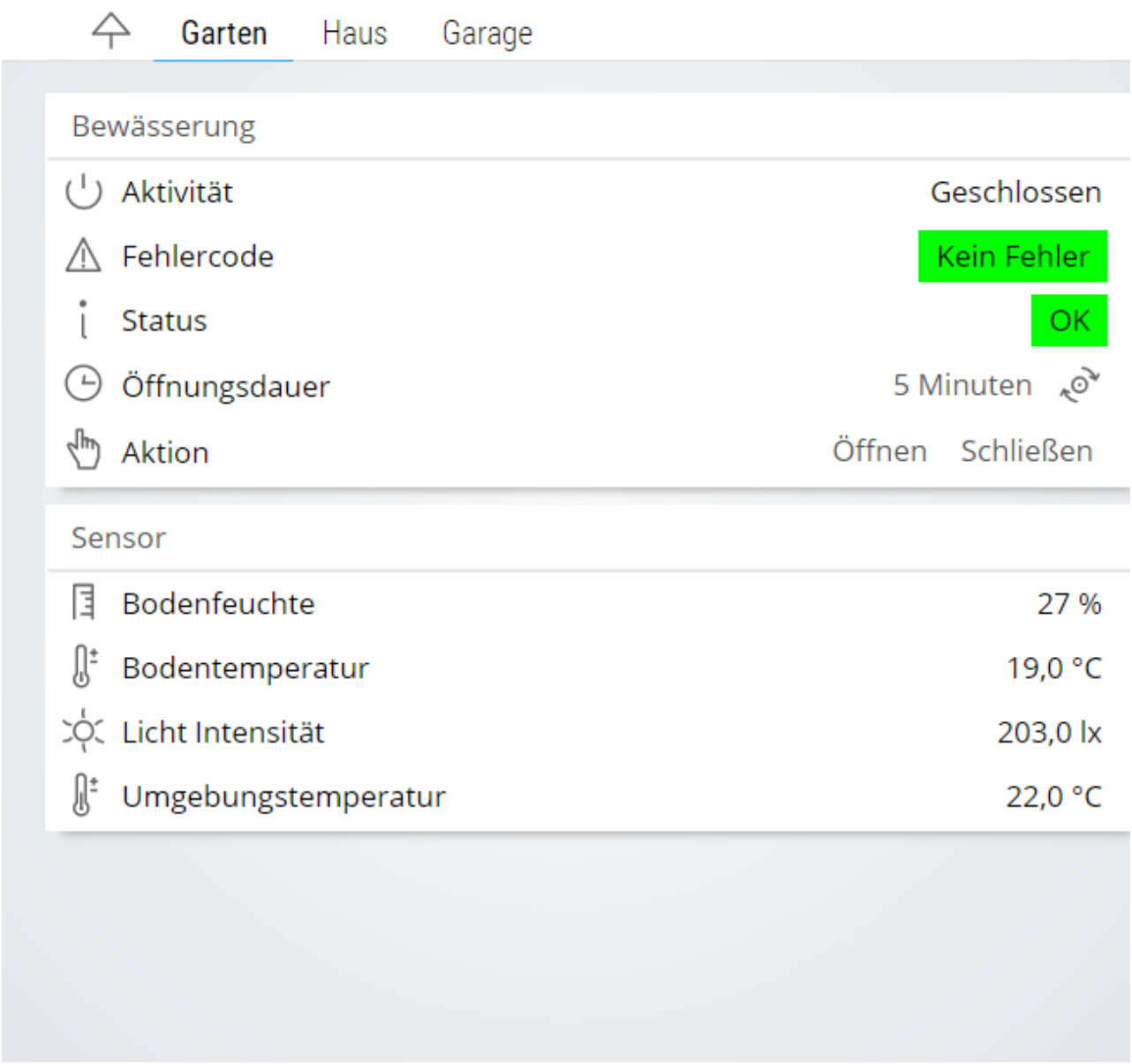

## ABLAUFPLÄNE

Die Version 6.0 führt Ablaufpläne ein, eine neue Art der Automation. Hier können mehrere Aktionen hintereinander ausgeführt werden. Durch "Wenn ..., dann"-Aktionen können Aktionen sogar ineinander verschachtelt werden. Auf diese Weise kann beispielsweise ein Licht unkompliziert angeschaltet und nach 30 Sekunden wieder ausgeschaltet werden. Hilfreiche Analys-Tools finden schnell die Ursache, falls mal etwas nicht so passiert, wie erwartet.




## GARDENA SMART SYSTEM UND HOME CONNECT

Mit der neuen Version kann nun auch der Garten und die Küche automatisiert werden. IP-Symcon ist nun zu GARDENA smart system bzw. Home Connect kompatibel. Hierfür wurden Module entwickelt, welche über den Module Store installiert werden können.




### ERWEITERTE UNTERSTÜTZUNG VON BETRIEBSSYSTEMEN

IP-Symcon kann bereits auf diversen Betriebssystemen verwendet werden. Mit der Version 6.0 steht IP-Symcon nun zusätzlich als natives Paket für Raspberry Pi OS (64-Bit Beta) und als Docker-Version für Arm64 und Apple Silicon (M1) Geräte zur Verfügung.

### WEBINAR ZUM THEMA IP-SYMCON 6.0 INKL. AKTIONEN UND ABLAUFPLAN

## MIGRATION

Bitte erstellen Sie sicherheitshalber ein Backup, sollten wider erwarten Probleme bei der Aktualisierung auftreten.

Falls Sie von einer früheren Version aktualisieren, lesen Sie bitte vorher die passenden Migrationsanleitungen.

Die Legacy Konsole ist nicht zu IP-Symcon 6.0 mehr kompatibel und wurde entfernt. Wir empfehlen den Wechsel auf unsere Pro Konsole.
Das WebFront Retro/Retro Mobile ist zu IP-Symcon 6.0 mehr kompatibel und wurde entfernt.
Zur Verwendung über die mobilen Apps werden diese in der Version 5.5 oder neuer benötigt.
Zur Verwendung mit IPSStudio wird dies in der Version 5.0 oder neuer benötigt.
Zur Verwendung mit dem AIO CREATOR NEO von mediola wird dies in der Version vom 2021-07-06 oder neuer benötigt.
Zur Verwendung mit IPSRCommand (Download) wird die Version 1.05 oder neuer benötigt.
Werden Ereignisse per Skript erstellt, so muss diesen per IPS_SetEventAction eine Aktion zugewiesen werden, damit sie fehlerfrei auslösen können
WebFronts ohne Kennwort sind standardmäßig nicht mehr von extern nutzbar. Ein WebFront ohne Kennwort muss ab IP-Symcon 6.0 explizit im „Sicherheit“ Bereich der WebFront Instanz dafür freigeschaltet werden.
Sys_GetURLContent hat bisher das SSL Zertifikat nicht korrekt überprüft. Dies wurde korrigiert.

## LISTE DER ÄNDERUNGEN

## ALLGEMEIN

Neu: PHP auf 7.4.20 aktualisiert
Neu: Der Ablaufplan wurde als neuer Typ der Automation eingeführt
Neu: ARM64 (aarch64) Version für Raspberry Pi und Docker (inkl. Apple M1)
Neu: RTSPS und HTTPS wird bei Streams unterstützt (gültige Zertifikate werden benötigt!)
Neu: max_execution_time in PHP auf 0 gesetzt, weil es Ursache für viele Abstürze war.
Neu: set_time_limit darf seit der 6.0 nicht mehr aufgerufen werden. Einige PHP-Module und Skript müssen angepasst werden, sodass diese die Funktion nur auf Versionen kleiner 6.0 aufgerufen wird.
Neu: Windows Daten Pfad kann nach der Installation verändert werden
Neu: Notification Control verbessert, sodass es die neuen Möglichkeiten der Liste nutzt
Neu: Hinweise im Instanz-Debug von I/O, wenn Demo-Version abgelaufen ist
Neu: WebFront wird ebenfalls deaktiviert, sobald die Demo-Version abläuft
Neu: Einige der Kern Instanzen (z.B. Archiv) sollten nur einmal erstellt werden dürfen. Wir prüfen dies nun beim Erstellen.
Neu: Intern werden alle Netzwerkzugriffe über cURL abgebildet (SSL-Zertifikate werden jetzt überall konsistent und korrekt geprüft und es gibt viel bessere Fehlermeldungen bei z.B. Timeouts oder Zertifikatsfehlern)
Neu: Image Grabber unterstützt jetzt auch Digest Authentication
Neu: VerifyPeer/VerifyHost kann im HTTP Client, POP3, IMAP, SMTP Modul und HomeMatic Socket aktiviert/deaktiviert werden
Neu: Docker Builds werden jetzt mit einem (Docker) Suffix markiert, da diese in Zukunft für mehrere Plattformen verfügbar sein können
Neu: User-Agent wird in allen internen Netzwerkzugriffen und allen PHP Netzwerkzugriffen auf IP-Symcon/Version gesetzt

Neu: Legacy Konsole/Dashboard wird (da inkompatibel) nicht mehr ausgeliefert (Reduziert das Setup um ~5MB)
Neu: Verbindungen mit Legacy Konsole/Dashboard werfen eine aussagekräftige Fehlermeldung
Neu: Neue Profil Millivolt
Neu: Bessere Absturz-Berichte werden gesendet
Neu: Google Assistant Request Limit erhöht sich automatisch, wenn viele Variablen eingerichtet sind
Fix: Absturz bei Fehlerhaften Daten in IPS_SetConfiguration korrigiert
Fix: DNS-SD: IPv4 Adressen landen nicht mehr im IPv6 Array
Fix: IP-Symcon bricht Startvorgang ab, sofern Port 3777 nicht verfügbar ist
Fix: In seltenen Fällen konnte die Konsole den Snapshot nicht laden wenn der interne Timestamp auf 0 stand
Fix: Kleine Optimierung beim Ausführen von Text-Skripten (+ im Log wird die Größe nach der erfolgreichen Ausführung korrekt angezeigt)
Fix: Sys_GetMemeryInfo funktioniert unter Linux mit >4 GB RAM
Fix: WebHooks liefern statt HTTP/1.0 200 X ein HTTP/1.1 200 OK, sofern keine andere Status-Zeile per PHP definiert wurde (HTTP/1.0 200 X ist laut HTTP Specs vollkommen valide, aber z.B. Loxone scheint damit aktuell Probleme zu haben)
Fix: Notification Control verursacht keine Fehler mehr, wenn kein Gerät ausgewählt ist
Fix: PHP-Fehler werden als Rohtext dargestellt und nicht mehr als HTML
Fix: Wochenplanaktionen mit schwarzer Farbe können gelöscht werden
Fix: Graphen: Nachkommastellen vom Profil werden beachtet sofern möglich
Fix: Graphen: Am 01. eines kurzen Monats werden Graphen wieder korrekt gerendert
Fix: Icon-Namen aller Objekte dürfen keine Leerzeichen enthalten. Dies wird korrekt validiert.
Fix: Das Zurücksetzen von lokalen Änderungen im Module Control konnte in sehr seltenen Fällen fehlschlagen
Fix: Das Aufräumen der Logdateien konnte bei sehr weniger Speicherplatz fehlschlagen, sodass das Logging deaktiviert wurde (SymBox)
Fix: Der Update Kanal unter Ubuntu/Rasperry Pi wird korrekt erkannt, wenn https verwendet wird
Fix: TextParser: Genutzte Variablen werden bei Referenzen einbezogen
Fix: Bei niedrigem Speicherplatz wurden ggf. zu viele Logfiles gelöscht, wodurch das Aufräumen
Fix: XDebug Extension wird korrekt als Zend-Extension geladen (Windows)
Fix: VoIP funktioniert auf der SymBox wieder korrekt (Nur SymBox)
Fix: Neue Installationen mit dem Windows-Installer verhinderten das korrekte Einrichten von Lizenzen und fielen in den Demo-Modus zurück (Nur Windows)
Fix: $_IPS['TRIGGER'] Systemvariable fehlte seit dem 6.0 Release
Fix: Suche/Ersetzt in Dateien funktioniert weiterhin korrekt, wenn eine Skriptdatei fehlt
Fix: Abarbeitung von EventLimit war um eins versetzt (seit 4.0)
Fix: Absturz korrigiert, sofern beim Beenden das Schreiben der Settings fehlgeschlagen ist
Fix: Absturz im WS Client korrigiert, welcher bei ständigem Verbinden/Trennen auftreten konnte
Fix: Absturz/Speicherleck bei RTSP Streams wenn diese oft und schnell im WebFront hin und her geschalten wurden
Fix: Speicherleck in Sys_GetNetworkInfo unter Windows korrigiert
Fix: Link zum alten Community-Dashboard angepasst

## ARCHIV

Neu: Spezialschalter „ArchiveWatch“, der einige Debug-Ausgaben vom Archiv aktiviert

Fix: Bei AC_GetAggregatedValues konnte manchmal die Duration falsch zurückgegeben werden
Fix: Werte können mit . und , als Dezimaltrennzeichen hinzugefügt werden
Fix: Beim Reaggregieren konnten ungültige UTF8 Zeichen an die Konsole gesendet werden, die dadurch neu geladen hat

## AKTIONEN

Neu: Aktionen wurden eingeführt und ersetzen die Muster aus Ereignissen
Neu: Aktion um ein Ereignis zu aktivieren/deaktivieren
Neu: Aktionen um die Rechenarten (Addieren, Subtrahieren, Multiplizieren, Dividieren) mit dem Wert einer anderen Variablen zu verwenden
Neu: Aktion für Gruppenschaltung aller Variablem mit einem definierten Profil innerhalb einer Kategorie
Neu: Aktion um eine Push-Nachricht via WebFront zu verschicken
Neu: Aktionen um Variable auf den Wert einer anderen Variablen zu setzen oder schalten
Neu: Aktion um eine E-Mail via SMTP-Instanz zu verschicken
Neu: Neue Aktion zum Warten mit einer Wartezeit, welche per Variable festgelegt wird
Neu: Neue Aktionen zum Schalten mit einem Invertierten/Multiplizierten/Dividierten Wert
Neu: Neue Aktion zum Konkatenieren von String-Werten
Neu: Aktion um aggregierte Werte via Archiv abzufragen, insbesondere auch fließend die letzten 24 Stunden und dergleichen
Neu: Die Aktion um Instanzfunktionen auszuführen ermöglicht es nun, den Rückgabewert in eine Variable schreiben zu lassen
Neu: Wenn-Dann Aktion, welche ein Ereignis als Bedingung hat
Neu: Aktion für „Wenn …, dann“ mit mehrfachen Bedingungen
Neu: Neue Aktion „Bedingung“, um Ablaufplan bei nicht-erfüllung abzubrechen

## GERÄTE

Neu: MQTT Client: Die ClientID kann angepasst werden
Neu: MQTT: Retained Messages können eingesehen und gelöscht werden
Neu: MQTT: Retained Messages werden nach 7 Tagen (Einstellbar) aufgeräumt ohne Update
Neu: MQTT: Retained Messages werden verzögert in die Persistenz übertragen, um die Performance zu verbessern
Neu: MQTT Geräte Instanzen unterstützen JSON, welches automatisch in Variablen/ Instanzen zerlegt wird
Neu: ModBus: Im ModBus Gateway können ganze Blocke definiert werden, welche am Stück ausgelesen werden sollen. Dadurch kann die Performance wesentlich verbessert werden. Die Timer in den jeweiligen Instanzen sollten dann deaktiviert werden!
Neu: ModBus: Unterstützung for den Datentyp: String
Neu: ModBus: Konfigurator, welcher eine Übersicht aller eingerichteten ModBus Instanzen anzeigt (Vorbereitung für einen Export/Import zur einfachen Konfiguration von ModBus Geräten)
Neu: KNX Discovery erkennt IP-Gateways anhand der gemerkten MAC Adresse, sofern die IP-Adresse sich geändert hat
Neu: KNX: „Status emulieren“ Eigenschaft
Neu: KNX: Module von EIB konsistent nach KNX (Legacy) umbenannt
Neu: KNX: Schlagwörter zum Erkennen von Status-GAs im Konfigurator sind jetzt konfigurierbar

Neu: KNX: XML-Export ist im Konfigurator standardmäßig ausgewählt
Neu: KNX: Sofern der OPC-Export ausgewählt wird, gibt es eine Warnmeldung, dass der XML-Export bevorzugt genutzt werden sollte
Neu: EnOcean: Unterstützung für Rauchmelder D2-14-30
Neu: EnOcean: Unterstützung für hochauflösende Ansteuerung von Eltako FRGBW71L
Neu: EnOcean: Unterstützung der Konfiguration von D2-01-* Geräten für Energie-/ Leistungsmessung
Neu: JSON Decoder Modul zum leichten Einlesen von JSON Dateien über den HTTP Client
Neu: Eltako FABH130
Neu: LCN Gateway steht standardmäßig auf 200 Dimschritte
Neu: Z-Wave: Expertenoption zum „Sperren der Variablen“ um das Erstellen von falschen Variablen zu verhindern, sofern Sensoren fehlerhafte Daten senden.
Neu: Image Grabber/HTTP Client: Auswahl zwischen Basic/Digest Authentifizierung
Neu: Eltako: FD62NP-230V, FD62NPN-230V, FL62-230V, FL62NP-230V, FR62-230V, FR62NP-230V werden unterstützt
Neu: KNX: Wenn Fehler beim Verbindungsaufbau auftreten zeigen wir eine bessere Fehlermeldung
Neu: Eltako: FJ62/12-36V DC, FJ62NP-230V werden unterstützt
Fix: Eltako FAFT60 Batterie Anzeige wurde nicht korrekt in Volt umgerechnet
Fix: KNX: „Status emulieren“ ist jetzt auch unter Windows verfügbar
Fix: EnOcean: Konfigurator zeigt das generische „Geräte hinzufügen“ nicht mehr als neues Gerät an
Fix: ModBus: Konfigurator zeigt das generische „Geräte hinzufügen“ nicht mehr als neues Gerät an
Fix: Shutter Control: Bessere Beschreibung der Zeitangaben
Fix: MQTT: Server Konfiguration konnte manchmal wegen UTF-8 Fehler nicht geöffnet werden
Fix: HomeMatic Socket blockiert beim Verbinden nicht mehr die MessageSink
Fix: ModBus Gateway intern umgebaut, sodass der Datenfluss für PHP-Module korrekt funktioniert
Fix: OneWire Test-Center ist konsistent verfügbar
Fix: Text Parser / JSON Decoder erstellen standardmäßig einen HTTP Client bei Neu-Erstellung
Fix: MQTT Client Configurator erstellt Client Devices (und nicht mehr Server Devices)
Fix: ALL4000 konnte nicht eingerichtet werden
Fix: KNX: Icon für DPT 14.068 korrigiert
Fix: HomeMatic: LOW_BAT wird korrekt das Battery Profil zugeordnet
Fix: EnOcean: General Profile Telegramme werfen keine Fehlermeldung mehr
Fix: JSON Decoder: Seltene Fehlermeldung beim Schreiben Werten korrigiert
Fix: EnOcean: D2-01-* Typen laut aktueller EEP Beschreibung angepasst
Fix: KNX: Erkenne Status-GA durchsucht nun alle Keywords einzeln

## I/OS

Neu: Neuer Datenfluss für UDP Socket, um Unicast/Broadcast gleichzeitig über einen I/O abbilden zu können
Neu: Server Socket: Zertifizierungsstelle kann optional angegeben werden
Neu: SSE Client kann eigene, in der Konfiguration definierte, Header mitsenden
Neu: SSE Client versendet ein eigenes Datenpaket, welches das SSE Protokoll direkt geparst ausgibt
Neu: Serial Port erkennt Symlinks welche auf ttyUSB* gehen (Insbesondere nützlich für udev Regeln)

## KONSOLE

Neu: Initiales Kopieren/Verlinken/Auslöserauswahl springt zur Position des Zielobjektes
Neu: Spalte „Geändert“ im Objektbaum
Neu: Bearbeiten-Dialog im Archiv zeigt nun die gewählte Variable an
Neu: Verwendete Bibliotheken aktualisiert
Neu: Schönere Fehlermeldungen, wenn der Login aus unbekannten Gründen fehlschlagt
Neu: Beim Umbenennen wird der aktuelle Name ausgewählt, sodass er beim Eingeben eines neuen Namens direkt überschrieben wird
Neu: Der Variable-bearbeiten-Dialog unterstützt nun auch ein Schalten von Variablen
Neu: Werden Variablen geloggt, kann dies nun am dazugehörigen Icon im Objektbaum erkannt werden
Neu: Anzeige von einer roten Warnmeldung wenn es starke Verzögerungen auf dem Rückkanal gibt
Neu: Schnellfilter im Objektbaum filtert auch die Beschreibungsspalte, falls diese sichtbar ist
Neu: Die Konsole überprüft nun Browserfeatures und nicht die Browserversion um ggfs. eine Warnung anzuzeigen. Dies verhindert insbesondere die Warnung bei aktuellen Edge-Browsern.
Neu: Im Skripteditor kann nun via Kontextmenü auf die Zwischenablage zugreifen (Kopieren/Ausschneiden/Einfügen)
Neu: Der Module Store ist nicht mehr Beta
Neu: Hat eine via Konfigurator gelöschte Instanz physikalisch untergeordnete Instanzen, so wird angeboten auch diese zu löschen
Neu: Mehrere Zeilen können im Ablaufplan ausgewählt und beispielsweise gleichzeitig dupliziert werden
Neu: Enter und Entf können als Tastaturkürzel im Ablaufplan verwendet werden
Neu: Im Info-Dialog wird beim Kopieren die Architektur einbezogen
Fix: Bei Bedingungen geht beim Wechsel von Variablen mit Assoziationen der Wert nicht verloren
Fix: Die Konsole generiert beim Speichern eine neue Skriptdatei, sofern die alte gelöscht wurde
Fix: Bearbeite Wochenpläne wie vorgesehen, wenn nur eine Wochenplanaktion gesetzt ist und diese angeklickt wird
Fix: Nicht-Gerät-Instanzen können bei der Erstellung nicht (unsichtbar) in eine Kategorie geschoben werden
Fix: Gebe eine sprechende Fehlermeldung zurück, wenn im Archiv versucht wird ein Wert mit einem neunstelligen Zeitstempel hinzuzufügen
Fix: Kaputte Module im Module Store konnten unter gewissen Umständen nicht geöffnet werden. Dies ist wieder möglich
Fix: Verhindere kosmetische Fehlermeldung, welche gelegentlich auftrat, wenn eine Instanz beim Schließen das Tabs gespeichert wurde
Fix: Bei erweiterten Bedingungen mit Wochentagsregeln, erkennt die Konsole nun korrekt, wenn die Regel per Skript gelöscht wird
Fix: Verhindere Fehler, wenn der Skripteditor eine Autovervollständigung versucht, während er innerhalb eines String ist
Fix: Endlosschleife/Defekter Objektbaum bei IPS_GetLocation korrigiert, wenn ein Ereignis mit Auslöser auf eine Variable unterhalb vom selben Ereignis erstellt wurde
Fix: Defekte Module im Module Store können editiert werden
Fix: Anzeige einer Warnmeldung, wenn die Demo abgelaufen ist
Fix: Beim Aktivieren einer Lizenz über das Demo Widget wird korrekt der „Neustart“-Balken angezeigt
Fix: Module Store: Beim Erstellen von Instanzen wurden benötigte Spliiter + I/O Instanzen erstellen, eine Konfiguration war bisher jedoch nur im Nachhinein möglich. Mit diesem

Update können diese Instanz direkt Konfiguriert werden, genau wie im „Instanz erstellen“ Dialog.
Fix: Behebe Fehler, welcher beim Neu laden zu „Uncaught Cannot find object with Id“ führen konnte
Fix: Update-Widget funktioniert korrekt für arm64 Versionen
Fix: Würde der Wert einer Profilassoziation überschrieben werden, erscheint eine Frage, ob dies wirklich passieren soll
Fix: Seltener Fehler behoben, welcher Emojis als Symbol in einer Liste abgeschnitten hat
Fix: In seltenen Konstellationen konnte ein Ereignis ein Icon für Erfüllung der erweiterten Bedingungen anzeigen, obwohl keine Bedingungen gesetzt wurden. Dies wurde behoben
Fix: Icons in Listen und Bäumen sind nicht mehr dragbar
Fix: Anzeige von Boolean Assoziation „true“ im Profilmanager gefixt
Fix: Erstellen von Medien (Dokumenten) über das „+“ korrigiert

## PRO-KONSOLE

Neu: Links werden im Standardbrowser und nicht mehr innerhalb der Konsole geöffnet
Fix: Verhindere kosmetische Fehlermeldung, wenn beim Schließen eines Tabs mehrere Tablisten existieren
Fix: Medien können wie vorgesehen heruntergeladen werden

## WEBFRONT

Neu: WebFront Retro/Retro Mobile entfernt
Neu: WebFronts ohne Kennwort sind standardmäßig nicht mehr von extern nutzbar
Neu: Wird ein Server manuell hinzugefügt, so wird geprüft, ob es sich dabei um eine IP-Symcon handelt
Fix: Nach einer versuchten (und fehlgeschlagenen) Verbindung zu einem IP-Symcon vor 5.0, ist der Verbindungsassistent weiter verwendbar
Fix: Charts in Panels werden beim Tab-Wechsel mit korrekter Zeit angezeigt
Fix: WebFront Visualization Instant hat den SecurityContext zu stark reduziert, wenn in den Standardeinstellungen die Startkategorie der mobilen App verändert wurde

## SYMOS

Neu: KNX/LCN kann über SymOS Connect und ETS/LCN Pro aus der Ferne programmiert werden. Diese Funktion befindet sich in einer internen Beta-Version und eine Freischaltung dafür kann über support@symcon.de angefragt werden.
Fix: Backups mit der Größe > 2,1GB sind jetzt möglich
Fix: Beim Booten der SymBox wurde der „IP-Symcon startet“ Dialog manchmal übersprungen

## PHP-SDK

Neu: SDK für Aktionen
Neu: Neues Element SelectAction
Neu: Neues Element ScriptEditor
Neu: Neues Element ColumnLayout
Neu: Inhalt von RowLayout und ColumnLayout kann dynamisch angepasst werden
Neu: Listen können durch den Benutzer bei gesetztem Parameter „changeOrder“

umsortiert werden
Neu: Parameter für SelectVariable um erlaubte Variablen einzuschränken
Neu: on-Skripte können nun auch als Array eingetragen werden
Neu: Image unterstützt nun width und onClick
Neu: Parameter ‚loadValuesFromConfiguration' für List und Tree
Neu: Configurator versucht die Spaltenbreite bei Knappheit besser zu verteilen und zeigt ggf. einen Scrollbalken
Neu: List und Tree unterstützen nun „enabled"
Neu: Parameter „moreDigits" für NumberSpinner
Neu: Werden bei einem NumberSpinner ohne „moreDigits" mehr Nachkommastellen als erlaubt eingegeben, wird dies als Fehler angezeigt
Fix: Dynamisch gesetzte Spaltennamen von Listen werden übersetzt
Fix: Werden dynamisch neue Werte für ein Element Tree gesetzt, wird die aktuelle Sortierungsspalte beibehalten
Fix: Ist beim Configurator visible auf false gesetzt, so wird das Element nun auch korrekt versteckt
Fix: Wird der Suffix eines NumberSpinner dynamisch angepasst, so wird der neue Wert wie vorgesehen übersetzt

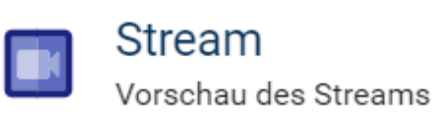
# V5.4->V5.5 (Q4/2020)

## WESENTLICHE ÄNDERUNGEN


Verbesserte Streams
Mobile Apps überarbeitet
Optimiertes Archiv
Neue Geräteanbindungen


### VERBESSERTE STREAMS

Die Behandlung von Streams wurde optimiert. Egal ob ein Stream als RTSP oder MJPEG vorliegt, dieser ist nun nahezu live in der Visualisierung verfügbar, ganz gleich ob lokal oder per Fernzugriff. Auch können Streams nun unter Android, iOS, WebFront und der Verwaltungskonsole betrachtet werden.



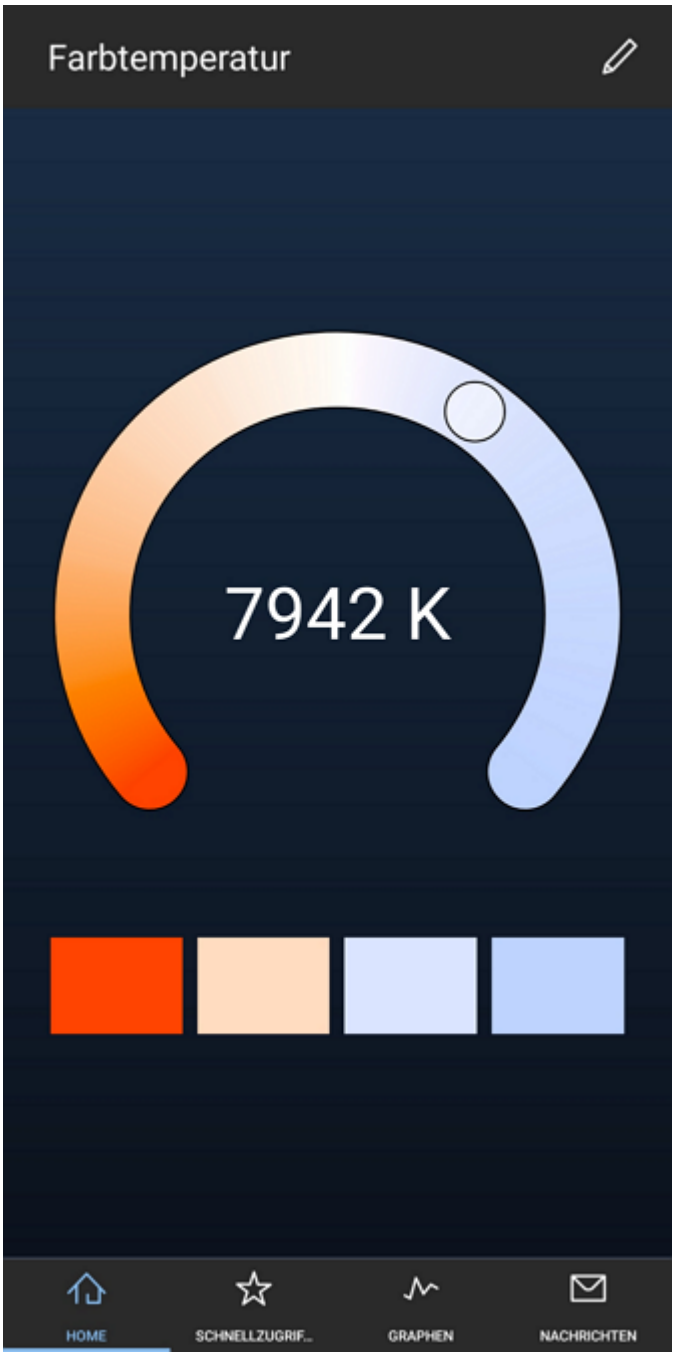



SCHLIESSEN

### MOBILE APPS ÜBERARBEITET

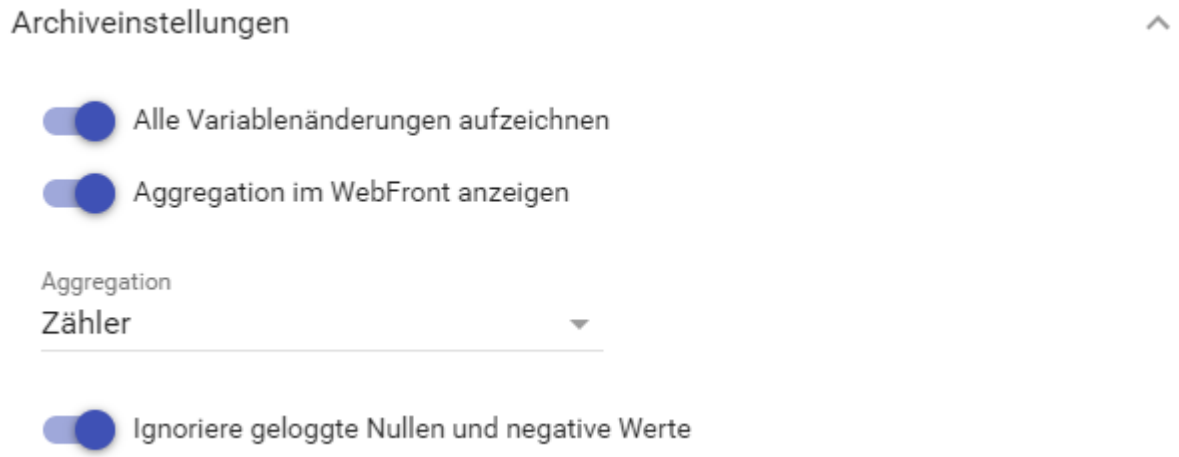
Die Darstellung des Objektbaums in den Apps und im WebFront wurde gleichgezogen. Dadurch ist nun in den Apps eine tiefere Verschachtelung darstellbar. Um direkt in die richtige Visualisierung zu kommen, kann dies nun in der App voreingestellt werden. Noch

mehr Aufwand spart man über die Schnellzugriffsleiste. Hier kann komfortabel auf Graphen, Favoriten und Nachrichten zugegriffen werden. Für die Einstellung der Farbtemperatur von Lampen gibt es nun eine optimierte Darstellung. Die oben beschriebenen Optimierungen der Streams gelten selbstverständlich auch für die Apps.




## OPTIMIERTES ARCHIV

Das Archiv wurde grundlegend überarbeitet. Dabei wurde es unter der Haube komplett optimiert, so dass nun beispielsweise Aggregationen deutlich schneller laufen. Gleichzeitig wurde der Speicherverbrauch drastisch gesenkt, da die Aggregationsdaten nun nicht mehr im Speicher gehalten werden. Zusätzlich ist es als neues Feature nun möglich bei einem Zähler Nullen und negative Werte zu ignorieren, welche in manchen Fällen bei Fehlfunktionen gesendet werden.




### NEUE GERÄTEANBINDUNGEN

Mit der neuen Version wurden diverse neue Geräte und Schnittstellen implementiert. So kann IP-Symcon nun auch nativ als MQTT Client verwendet werden und für EnOcean wurden zwölf zusätzliche EEPs eingebunden. Darüber hinaus kann CMI nun auch mit UVR65, UVR67, UVR 610 und CAN-EZ3 verwendet werden. Dazu kommen noch einige Optimierungen für KNX, S7 und eKey.

## MIGRATION

Bitte erstellen Sie sicherheitshalber ein Backup, sollten wider erwarten Probleme bei der Aktualisierung auftreten.

Falls Sie von einer früheren Version aktualisieren, lesen Sie bitte vorher die passenden Migrationsanleitungen.

## LISTE DER ÄNDERUNGEN

## ALLGEMEIN

Neu: PHP 7.4.11 (Bitte beachten: https://www.php.net/manual/de/migration74.deprecated.php)
Neu: RTSP Streams werden nach MJPEG umgewandelt
- Bild wird sehr zeitnah und fast live angezeigt
- Nun auch Unterstützung für Safari und iOS verfügbar
- Benötigt die Apps in Version 5.5
- Auflösung, Qualität und Frame-Rate wird beim Connect-Dienst reduziert, um Traffic zu sparen

Neu: MJPEG Streams werden über IP-Symcon geroutet
- Direkt über den Connect Dienst verfügbar

Basis Authentifizierung wird über IP-Symcon erledigt und ist somit uneingeschränkt nutzbar
Benötigt die Apps in Version 5.5
Neu: Tray Icon für Windows komplett erneuert mit High DPI Support
Neu: Schalter um "Verschachtelung" zu aktivieren ist standardmäßig an und nur noch auf Legacy-Systemem verfügbar
Neu: Schalter, um Enums anstatt CheckBox'en in den Apps zu nutzen (benötigt Apps in der Version 5.5)
Neu: Connect ist für Neuinstallationen automatisch aktiviert
Neu: Wochenplan kann maximal 32 Aktionen (vorher 8) haben
Neu: Autostart im WebFront unterstützt die CIDR Notation
Neu: IP-Autostart mit CIDR 0.0.0.0/0 wird auch über Connect-Dienst ausgewertet
Neu: Mac Version wird korrekt von Apple Notarisiert, sodass Gatekeeper glücklich ist
Neu: WebFront ist jetzt vom Typ Visualisierung und nicht mehr Konfigurator
Neu: WebServer ist sofort beim Start verfügbar (und nicht erst beim Kernel Betriebsbereit)
Neu: Durch besseres Caching sollte die Konsole ca. 1 - 3 Sekunden schneller Starten
Neu: GetValueFormattedEx, welches einen weiteren Parameter akzeptiert. (Wert, der zum Formatieren genutzt werden soll)
Neu: Web Server SSL Konfiguration geschieht jetzt über das Konfigurationsformular (Zertifikate im cert Ordner werden migriert und gelöscht)
Neu: Web Server nutzt jetzt SSL 1.2 (SSL Labs attestiert IP-Symcon damit wieder die Note A)
Neu: Settings haben eine Schema-Version um zukünftig inkompatible Downgrades zu verhindern
Neu: Instanzen werden beim Starten in der Reihenfolge Core, Visualization, Discovery, Configurator, Device, Splitter, I/O erstellt, wodurch Probleme wie z.B. Verbindungsprobleme bei MQTT gelöst werden.
Neu: WebFront Visualisierung: Hinweis, dass die Standard Graphen Optionen nur für das WebFront gelten
Neu: OpCache ist jetzt standardmäßig aktiv - kann jedoch in den Spezialschaltern wieder deaktiviert werden
Neu: max_memory_limit darf nun auf maximal 128MB erhöht werden
Neu: CC_GetRequestLimitCount und NC_GetRequestLimitCount mit dem die Anzahl der Anfragen, welche gegen das tägliche Limit zählen, abgefragt werden kann
Neu: NotificationWatch Spezialschalter welcher alle Anfragen an den Push-Nachrichten Server ins Log ausgibt
Fix: Proxy Stream URLs filtern Benutzername/Passwort aus der Adresse im Logfile
Fix: Lock Profil nutzt das adaptive Icon
Fix: Duplizieren von Variable ohne Logging korrigiert
Fix: Demo-Überprüfung unternimmt nun 5 Versuche á 10 Sekunden (manchmal war nach einem Neustart kein Internet verfügbar und die Demo startete ohne korrekte Endzeit und lief nur 6 Stunden)
Fix: Duplizieren von Instanzen konnte Fehlschlagen, wenn die Instanzen unterschiedlich verbunden waren und dabei eine Instanz keine Verbindung hatte
Fix: Attribute werden in der Persistenz direkt beim Register initialisiert
Fix: In seltenen Fällen konnten Fehler in der Konsole erscheinen, wenn Objekte während des Öffnens gelöscht wurden
Fix: Sys_GetNetworkInfo liefert nur aktive Netzwerk-Adapter
Fix: Bei nicht mehr existierenden Variablen wird das Logging auf Inaktiv gestellt
Fix: PRCE JIT unter MacOS korrigiert
Fix: Nachrichten werden in korrekter Reihenfolge vor KR_READY verarbeitet, sodass z.B. Siemens Gateway die IS_ACTIVE Nachricht korrekt erhalten und verarbeiten
Fix: User-Agent wurde bei Anfragen nicht korrekt gesetzt
Fix: Löst hoffentlich Abstürze in Verbindung mit dem ServerSocket
Fix: Anzahl der CPU’s konnte auf z.B. Intel NUCs nicht korrekt ermittelt werden

Fix: MJPEG Streams, welche keine HTTP Ranges unterstützen, konnten nicht genutzt werden
Fix: Seltene Abstürze beim Beenden, wenn während des Herunterfahrens „die Glocke“ gearbeitet hat
Fix: Proxy: Im Log werden username, password, user, pw, usr, pwd Werte maskiert
Fix: Netzwerk Bibliothek auf Stand der 5.4 zurückgesetzt um ggf. Probleme mit der neuen Version auszuschließen
Fix: MessageQueueWatch Meldungen zu KL_NOTIFY (lila) befördert
Fix: Notification Control: Beim Versenden von Test-Nachrichten war „Sent!“ nicht übersetzt
Fix: Datenfluss ignoriert Pakete > 16kb sofern der ReceiveDataFilter/TransmitDataFilter gesetzt ist um Abstürze in der RegEx Engine zu umgehen. Dies wird im Debug Log der betroffenen Instanz geloggt.
Fix: Fehlerhafte JSON-RPC Requests konnten IP-Symcon zum Absturz bringen
Fix: WebServer konnte bei starker Nutzung vom IPSView Web-Client abstürzen
Fix: Abstürze unter Raspberry Pi & SymBox aufgrund von fehlerhafter PCRE/JIT Bibliothek gelöst
Fix: Endlosschleife/Absturz bei IPS_GetLocation korrigiert, wenn ein Ereignis mit Auslöser auf eine Variable unterhalb vom selben Ereignis erstellt wurde

## ARCHIV

Neu: Optimierung des Archivs
Neu: Archiv nutzt jetzt intern Attribute anstatt Properties. (Die AC_* Funktionen benötigen kein IPS_ApplyChanges mehr!)
Neu: Bei Zähleraggregationen können Nullen ignoriert werden
Neu: Aggregationsdaten einer Zählervariable beginnen nun direkt beim ersten Wert, nicht erst beim ersten Delta
Neu: Wird das Logging einer Variablen erstmalig aktiviert, wird direkt der aktuelle Wert geloggt
Neu: Erlaube Migration der Daten von Integer Variablen zu Float Variablen
Neu: Erlaube parallele Aggregation mehrere Variablen
Neu: Ignoriere Nullen wurde erweitert, sodass es auch negative Werte filtert
Fix: Die Referenzen des Archivs werden bei Bedarf aufgeräumt. So sollte es nicht mehr möglich sein, dass Variablen als Referenz das Archiv angeben, dort allerdings nicht zu finden sind.
Fix: Alternative Implementation für die kontinuierliche Aktualisierung der Datengröße, behebt möglicherweise die negativen Größenangaben
Fix: Bei einer Reaggregation mit vielen CPUs konnte diese evtl. in einem Deadlock enden
Fix: Problem behoben, wenn es im DB-Ordner Daten gab, aber keinerlei Referenz auf die Variable mehr existierte
Fix: Problem beim Starten mit der letzten Version korrigiert
Fix: Statistiken von String-Variablen wurden nicht korrekt berechnet
Fix: Archiv: In seltenen Fällen konnte ein Endlosschleife entstehen und alles an RAM verbrauchen, sodass IP-Symcon abstürzt
Fix: Archiv: Seit der Optimierung aus der vorletzten Beta konnten Race-Conditions auftreten und den Dienst zum Absturz bringen
Fix: Archiv: Viele Graphen bei Systemen ohne SSD konnten das Gesamtsystem verlangsamen
Fix: Archiv: Beim Duplizieren von Variablen konnte der initiale Wert doppelt geloggt werden
Fix: AC_AddLoggedValues versucht mehrfach auf die Datei zuzugreifen, sofern diese aktuell „blockiert“ wird

## GERÄTE

Neu: CMI: Unterstützung für UVR65, CAN-EZ3, UVR610, UVR67
Neu: EnOcean: EnOcean Gateway zeigt die EURID an
Neu: EnOcean: D2-15-00, D2-07-00, D2-03-00, D2-03-0A, D2-03-10, D2-A0-01
Neu: EnOcean: A5-20-06, A5-10-1E, A5-10-22, A5-10-23
Neu: EnOcean: F6-01-01 und F6-10-01
Neu: MQTT Client (inkl. Device, Splitter und Konfigurator)
Neu: MQTT: Nutzt die Verbesserung vom Server Socket besser, indem er weniger blockiert
Neu: MQTT: Nachrichten mit Retain Flag werden auch bei IP-Symcon Neustart korrekt retained
Neu: KNX Shutter unterstützt das prozentgenaue Positionieren von Lamellen
Neu: eKey: Unterstützung für Rare Net Protokoll
Neu: ALL3653 zu den unterstützen Geräte hinzugefügt
Neu: S7 unterstützt nun den Typ "String"
Neu: HomeMatic IP ist standardmäßig beim Erstellen aktiv
Neu: Bessere Fehlermeldung, wenn HomeMatic Kanal (z.B. RF) nicht verfügbar ist
Fix: Min/Max bei DPT9 realistischer gestaltet
Fix: LCN Module Konfigurator werden korrekt übersetzt
Fix: VoIP: Manchmal konnte VoIP stecken bleiben, wenn ein Anruf abgelehnt wurde
Fix: LCN sollte sich beim Start direkt verbinden, ohne das bei KR_READY eine Erneutes verbinden angestoßen wird
Fix: Die Initialisierung von KNX, FHZ, IRTrans, MQTT Client, Siemens S5/S7/Logo, WMRS200, Z-Wave wird asynchron gemacht und verzögert nicht mehr den Start
Fix: EnOcean: D2-01-xx Einzelkanal Rollladenaktoren, wie z.B. der OPUS Shutter, können jetzt korrekt angesteuert werden
Fix: EnOcean: F6-05-00 und F6-05-02 fehlten die Konfigurationsformulare
Fix: MQTT: Beim Empfang von unvollständigen Datenpaketen, konnten bereits vollständige Datenpakete verworfen werden
Fix: MQTT: Keep-Alives werden beachtet und Clients nach Timeout Disconnected (+LWT versendet)
Fix: Web Server: Beim Importieren von Zertifikaten konnten unter Windows diese nicht korrekt aufgeräumt werden
Fix: MQTT: Wenn die Session Cleanup Zeit versehentlich zu klein Eingestellt wird, werden noch aktive Sessions nicht fälschlicherweise mit aufgeräumt
Fix: MQTT: Löst hoffentlich ein Problem, wobei Geräte fälschlicherweise das LWT empfangen und dann nicht mehr geschaltet werden können
Fix: WebServer hat in einigen Fällen nach dem Import die SSL-Zertifikate nicht in die Persistenz geschrieben
Fix: Z-Wave Geräte nutzen jetzt korrekt das Battery.100 Profil
Fix: MediaPlayer warf eine Fehlermeldung, wenn ein WAC_Stop direkt nach IP-Symcon start ausgeführt wurde
Fix: SSDP Suche fand einige Geräte nicht, welche sich zu 100% an die Spezifikation gehalten haben
Fix: KNX Konfigurator: Wenn ein XML-Export viele GAs hatte, welche keine DPT Zuordnung haben, konnte der Konfigurator ewig laden und IP-Symcon zum Absturz bringen
Fix: Ausgehende Anrufe per VoIP konnte nicht korrekt abgebrochen werden
Fix: DNS-SD: Abstürze unter Linux, Ubuntu, RPi wenn gleichzeitig eine Anfrage lief und die Instanz übernommen wurde
Fix: MQTT: Abstürze bei defekten Paketen korrigiert. (Jetzt kommt eine Fehlermeldung „Consumed Packet is larger than buffer“)
Fix: MQTT: Abstürze/Unendlich viele Log-Nachrichten mit „Skipping Cleanup“ korrigiert
Fix: Siemens S7/Logo: Verbinden konnten fehlschlagen, wenn Anfragen bereits vor der erfolgreichen Verbindung per Skript angefordert wurden

Fix: confirm-Dialoge bei Z-Wave waren nicht übersetzt
Fix: confirm-Dialog beim Zurücksetzen vom Z-Wave Gateway hinzugefügt
Fix: Neue TCM515 Revision sendet neue Bestätigungsmeldungen, welche nicht korrekt ausgewertet wurden
Fix: LCN Shutter hat den Status nicht korrekt nachgeführt, wenn die Invertierung aktiv war
Fix: Alte EIB RGB/RGBW Instanzen können wieder korrekt Rückmeldungen empfangen
Fix: EEP A5-14-08 wird jetzt korrekt geladen und ist verfügbar

## I/OS

Neu: I/O für Server Side Event (SSE Socket)
Neu: WebSocket Client (inkl. SSL/TLS) [Datenfluss wie bei ClientSocket]
Neu: WWW Reader zu HTTP Client umbenannt
Neu: Server Socket unterstützt SSL/TLS
Neu: ServerSocket bietet die Option zum ignorieren von Verbindungsabbrüchen
Neu: ServerSocket kann 5 Verbindungen gleichzeitig verarbeiten
Neu: Client Socket hat SSL/TLS Unterstützung
Fix: Probleme beim Senden von sehr vielen Paketen über die I/Os (insbesondere Server Socket bei MQTT) gelöst
Fix: ServerSocket hat Feld für Zertifikate angezeigt, obwohl SSL deaktiviert war
Fix: Beim gleichzeitigem Öffnen/Schließen von I/Os konnte IP-Symcon abstürzen

## KONSOLE

Neu: Medienkategorie entfernt - Medien ohne übergeordnetes Objekt werden nun direkt auf der Hauptebene des Objektbaumes angezeigt
Neu: Speichere die letzte Spaltenauswahl in Objektbäumen für zukünftige Objektbäume
Neu: Duplikation eines Objektes in einer Spezialkategorie geschieht direkt vor Ort ohne Positionsauswahl
Neu: Spezialkategorie "Visualisierungen" beinhaltet WebFronts
Neu: Konsole wartet bis der Kernel Betriebsbereit ist und zeigt eine Meldung, dass IP-Symcon noch startet
Neu: Konsole zeigt an, wenn der Server heruntergefahren wird und startet die Konsole neu, sobald IP-Symcon wieder verfügbar ist
Neu: Per Doppelklick auf einen Stream wird eine passende Vorschau gegeben
Neu: Permanente Benachrichtigung, sofern die Lizenz/Spezialschalter geändert wurde; der Neustart jedoch nicht durchgeführt wurde
Neu: Unbehandelte Fehler werden sofort angezeigt, damit diese gemeldet werden können
Neu: Im Archive Control werden beim Auswählen einer nicht mehr existierenden Variablen nicht verwendbare Aktionen ausgegraut
Neu: Im Archive Control kann nun auch nach der Statusspalte sortiert werden
Neu: Werden Variablen mit geloggten Werten gelöscht, so wird gefragt ob die geloggten Werte ebenfalls gelöscht werden sollen
Neu: Inhalt des Variableneditors (Doppelklick auf Variablenwert) kann markiert und somit auch kopiert werden
Neu: Grundlegende Eigenschaften von IPS-View-Formularen können nun auch in der Konsole angepasst werden
Fix: Verwendete Bibliotheken aktualisiert
Fix: Nach dem Schließen von Dialogen wird der Fokus wie vorgesehen wieder auf den Objektbaum geschoben, damit Tastenkürzel direkt weiter verwendet werden können
Fix: Spezialkategorien unterstützen kein "ObjektID kopieren" mehr
Fix: Instanzen, die über einen Konfigurator erstellt wurden, können nun wie vorgesehen

andere valide Verbindungen finden obwohl sie sich initial anders verbunden haben
Fix: Mögliches Problem gefixt, wodurch der gespeicherte Zustand der Konsole beim Start verloren gehen konnte
Fix: Shortcuts im Skripteditor funktionieren nun auch unter Mac und verwenden auch Cmd statt Strg
Fix: Verifkation in einer ValidationTextBox wird nicht mehr ein Zeichen zu spät angewandt
Fix: PopupAlerts konnten zu groß werden
Fix: Strings bei "Bei bestimmter Wert" werden nicht mehr mit 'null' initialisiert
Fix: I/Os können in der physikalischen Ansicht korrekt gelöscht werden
Fix: Wird für eine neue Variable der Aggregationstyp Zähler beim Aktivieren des Loggings ausgewählt, wird dies wie vorgesehen übernommen.
Fix: Das Schnellausführungswidget wird nicht mehr leer angezeigt, wenn während dessen Ladevorgang auf ein Tab gewechselt wird
Fix: Bei fehlerhaften Debug-Meldungen werden die problematischen Rohdaten angezeigt
Fix: Im ArchiveControl wird nicht mehr versucht nicht-Variablen-Objekte mit gelöschten Variablen gleicher ID gleichzusetzen
Fix: Bei einem Fehler beim Speichern im Profilmanager, wird dieser auch ausgegeben
Fix: Verhindere Fehler beim Öffnen eines ausgelösten Ereignisses dessen Auslöser nicht mehr existiert
Fix: Sonderzeichen in Variablenaktionen mit Strings werden korrekt beim Öffnen des Ereignisses wieder dargestellt
Fix: Instanzaktionen mit Stringparametern können nun auch wie vorgesehen mit Sonderzeichen verwendet werden
Fix: Autovervollständigung im Skripteditor um mehr Konstanten und Systemvariablen erweitert
Fix: Werte Auslösungwert in ausgelösten Ereignissen bei "leerem" Wert korrekt aus
Fix: Drag & Drop im Objektbaum leicht optimiert
Fix: Sonderzeichen im DebugLog werden nun wie vorgesehen ausgewertet
Fix: Der Lizenz-Benutzername wird nun einheitlich als solcher bezeichnet
Fix: Das Schnellausführungswidget zeigt die Objektinformation hinter der ID wie vorgesehen nur mit dem Namen ohne Pfad an
Fix: Beim Öffnen eines Moduls nach einem Update konnte ein Fehler kommen. Dieser wurde behoben
Fix: Erkenne bei Tastenkombinationen im Objektbaum wie vorgesehen, wenn kein Objekt ausgewählt ist und tue entsprechend nichts
Fix: NumberSpinner setzt jetzt die minimum/digits Parameter korrekt um
Fix: Eingabe von Datum/Uhrzeit im Safari korrigiert bei der manuellen Archiv-Daten Eingabe
Fix: Sporadische Fehlermeldungen beim Editieren vom Wochenplan korrigiert
Fix: Wird im Such- oder Filterfeld des Objektbaums etwas per Strg + V eingefügt, wird nicht zusätzlich ein möglicherweise kopiertes Objekt im Objektbaum eingefügt
Fix: Float-Variablen mit Assoziationsprofilen werden nun wie vorgesehen bei erweiterten Bedingungen eines Ereignisses gehändelt
Fix: Die Werte von Variablen mit Prozentprofilen werden wie vorgesehen bei erweiterten Bedingungen dargestellt
Fix: Drücken von STRG+V im Suchfeld fügt keine neuen Instanzen im Objektbaum hinzu
Fix: In seltenen Konstellationen, könnten Textfelder mit $-Zeichen in der Instanzkonfiguration zu Fehlern führen. Dies wurde behoben
Fix: Profile konnten nach einem Neustart in der Pro Konsole fehlen
Fix: Im Snapshot konnte das compatibility Feld doppelt vorhanden sein
Fix: Bei erweiterten Bedingungen mit Wochentagsregeln, erkennt die Konsole nun korrekt, wenn die Regel per Skript gelöscht wird
Fix: Wird beim Hinzufügen einer ObjektID im Skripteditor die Auswahl per Enter bestätigt resultiert dies nicht mehr in einem Zeilenumbruch
Fix: Die ausgewählte Instanz bei einem SelectModule-Feld wird nun wie erwartet angezeigt
Fix: Verhindere kosmetischen Fehler mit _isPrimaryDialog

Fix: Defekte Module im Module Store können editiert werden

## PRO-KONSOLE

Neu: "Letzte Verbindung" wird kontinuierlich aktualisiert solange ein Fenster offen bleibt
Neu: Filter-Möglichkeit für Server
Neu: Bessere Fehlermeldung, wenn selbst-signierte Zertifikate verwendet werden
Fix: Beim Umdocken einer Instanzkonfiguration konnten Werte in verschachtelten Elementen (z.B. RowLayout) verloren gehen. Dies wurde behoben
Fix: Bei "Erneut suchen" von Servern wird die Verbindung zum Netzwerk komplett neu aufgebaut. So können Server gefunden werden, obwohl man sich erst nach dem Öffnen des Verbindungsassistenten mit dem Netzwerk verbunden hat
Fix: Schlägt das Laden von Lizenzen aus der Lizenzverwaltung fehl, z.B. weil kein Internet vorhanden ist, so wird der Benutzer abgemeldet und bleibt nicht bei "Wird eingeloggt..." stehen
Fix: Unter Linux wird der in der sources.list gewählte Update Kanal korrekt erkannt
Fix: Unter Linux wird geprüft, ob der update-manager verfügbar ist
Fix: Sind Tabs mit gleichem Namen in verschiedenen Tablisten offen, so werden sie namentlich differenziert als wären sie in der selben Liste
Fix: Verwendete Bibliotheken aktualisiert
Fix: Login zur Lizenzverwaltung funktioniert nach Überarbeitung des persönlichen Bereichs wieder
Fix: Neue Benutzer oder Benutzer mit geändertem Benutzernamen können sich wieder via E-Mail anmelden

## WEBFRONT

Neu: Unterstützung für IPSView WebViews (Als Element und direkt als Medien-Objekt)
Neu: Unterstützung für neue RTSP Streams (werden direkt im Bild und nicht als iFrame dargestellt)
Neu: Unterstützung für MJPEG Streams, welche direkt über IP-Symcon / Symcon Connect abgerufen werden
Neu: Streams unterstützen im Inhaltswechsler die Skalieren und Aspect Ratio Einstellung
Neu: Streams werden, wenn über Connect dargestellt, nach 1 Minute pausiert
Neu: Aktionen im Wochenplan sind scrollbar und können maximal 32 Aktionen darstellen
Neu: Dialog für Verbindungsabbrüche überarbeitet
Fix: Legende mit „großer“ Legende haben das Aktivierungshäkchen falsch skaliert
Fix: Beim Erstellen von einer Menüleiste direkt gefolgt von einer Kategorie konnte die Startkategorie nicht gewählt werden
Fix: Ein WFC_SwitchPage wird als Interaktion vom Idle Widget ausgewertet
Fix: Die Meldung, dass die Zeitzone nicht stimmt, konnte nicht „weggeklickt“ werden
Fix: Einstellungen aus 3.x und älter mit versteckten Root-Objekte wurde im Editor nicht korrekt übersprungen
Fix: WebFront läuft bei Verbindungsverlust sofort neu, sofern möglich
Fix: Wir probieren das Neu-Verbinden jetzt am Anfang schneller und zum Ende hin mindestens alle 10 Sekunden
Fix: Popup Module laden Streams nicht mehr im Hintergrund, sondern nur, wenn diese wirklich sichtbar offen sind
Fix: Charts in Panels werden beim Tab-Wechsel mit korrekter Zeit angezeigt

## SYMOS

Neu: Alle SymBox'en die nach August 2019 gekauft wurden, erhalten ~4 GB zusätzlichen Speicher!
Neu: Benachrichtigung für alle, welche noch ein CM1 in der SymBox nutzen
Neu: Wechsel von uclibc auf glibc (Wir nutzen glibc auf den anderen Plattformen auch schon)
Neu: SymOS aktualisiert, sodass es ebenfalls die neusten Bibliothek-Versionen enthält
Fix: Backups können jetzt mehrere GB groß sein, sofern es genug Speicherplatz auf der SymBox gibt
Fix: Größe reduziert, sodass die Updates auf der SymBox 1 mit alter/kleiner Systempartition besser funktionieren

## LEGACY-KONSOLE

Neu: Neuer Typ Visualisierung wird unterstützt
Neu: Alle Instanzen werden über die webbasierte Konsole konfiguriert
Neu: Hinweis, dass die Legacy Konsole ab 2021 entfernt wird

## PHP-SDK

Neu: Parameter "name" im Configurator kann Namen der erstellten Instanzen überschreiben, insbesondere auch für Splitter und I/Os in einer Kette
Neu: RegisterOnceTimer, um asynchrone Sofort-Aktionen auszuführen
Neu: Image Element kann jetzt auch Streams per mediaID anzeigen
Neu: Neues Element: SelectProfile
Neu: Formularelement "SelectInstance" kann mit "validModules" auf bestimmte Module restriktiert werden
Neu: Neues Formularelement "SelectModule" um per Dropdown Instanzen eines bestimmten Modultyps auszuwählen
Fix: Das TestCenter passt sich nun dynamisch an Änderungen der Aktionsvariablen an
Fix: Die Parameter "info" und "position" werden nun auch wie vorgesehen bei erstellten Verbindungen angewandt
Fix: Wenn eine Zeile in einer Liste nicht editierbar ist, wird auch die CheckBox ausgegraut

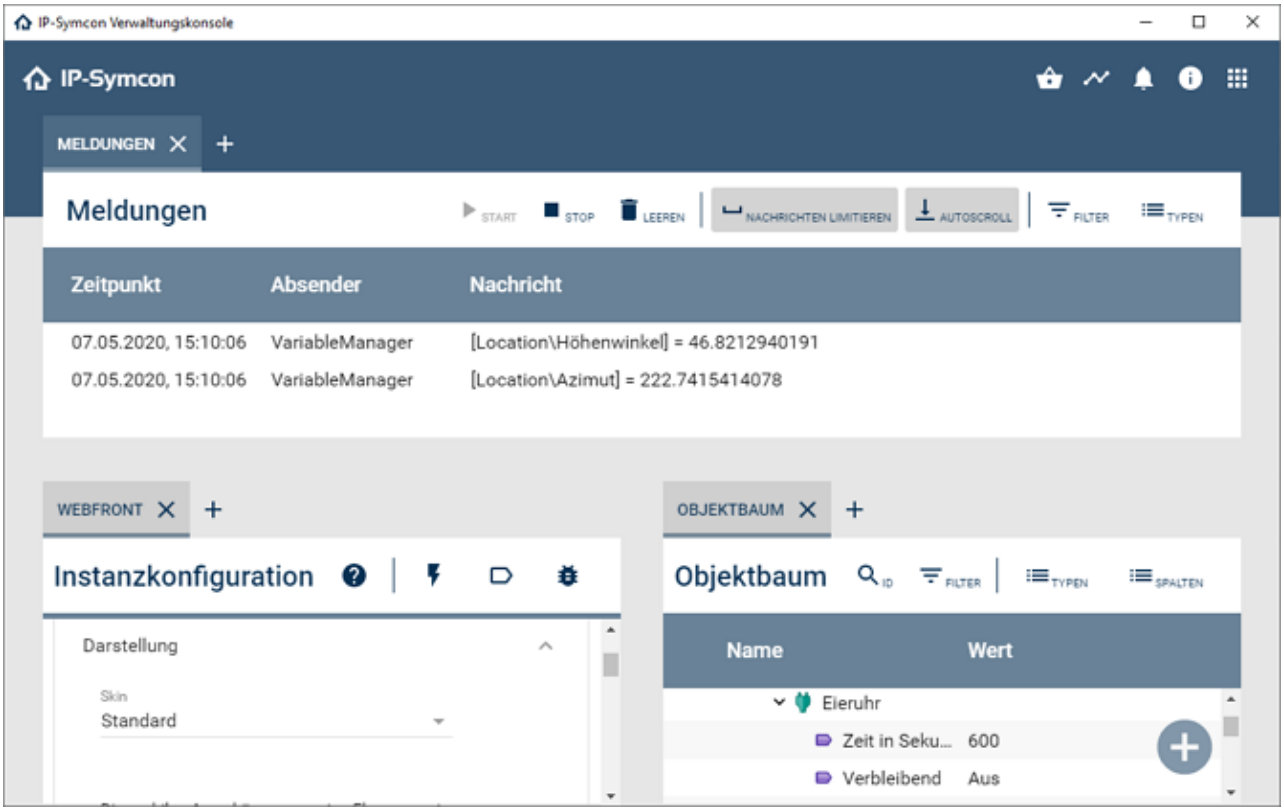
# V5.3->V5.4 (Q2/2020)

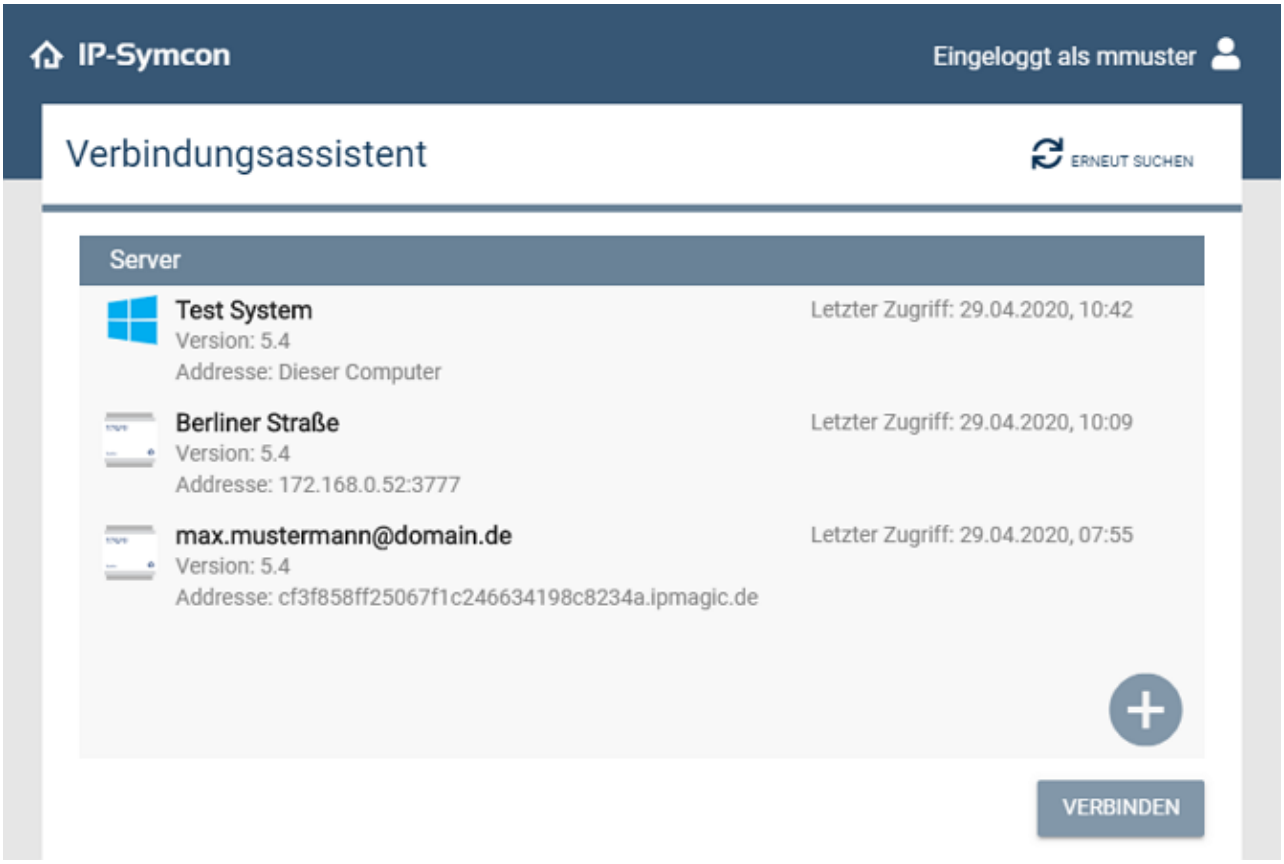
## WESENTLICHE ÄNDERUNGEN

Pro Konsole
SymOS Connect
Hovereffekte bei Graphen
Verbesserte Geräteunterstützung

### PRO KONSOLE

Eine neue Konsole für die Konfiguration von IP-Symcon wurde eingeführt. Als native Anwendung für Windows, Ubuntu und Mac bietet die Pro Konsole eine deutlich erhöhte Performance. Die neue Konsole bietet nicht nur alle Funktionen der webbasierten Konsole, sondern auch noch einige mehr. So können beispielsweise Tabs ab- und umgedockt werden um optimal mit mehreren Tabs zu arbeiten. Im Verbindungsassistenten können mehrere IP-Symcon Server verwaltet werden. Server im lokalen Netzwerk und entfernte Server aus der Lizenzverwaltung werden automatisch gefunden. Weitere Server können manuell hinzugefügt werden. Zusätzliche Informationen gibt es in der Dokumentation zur Pro Konsole.



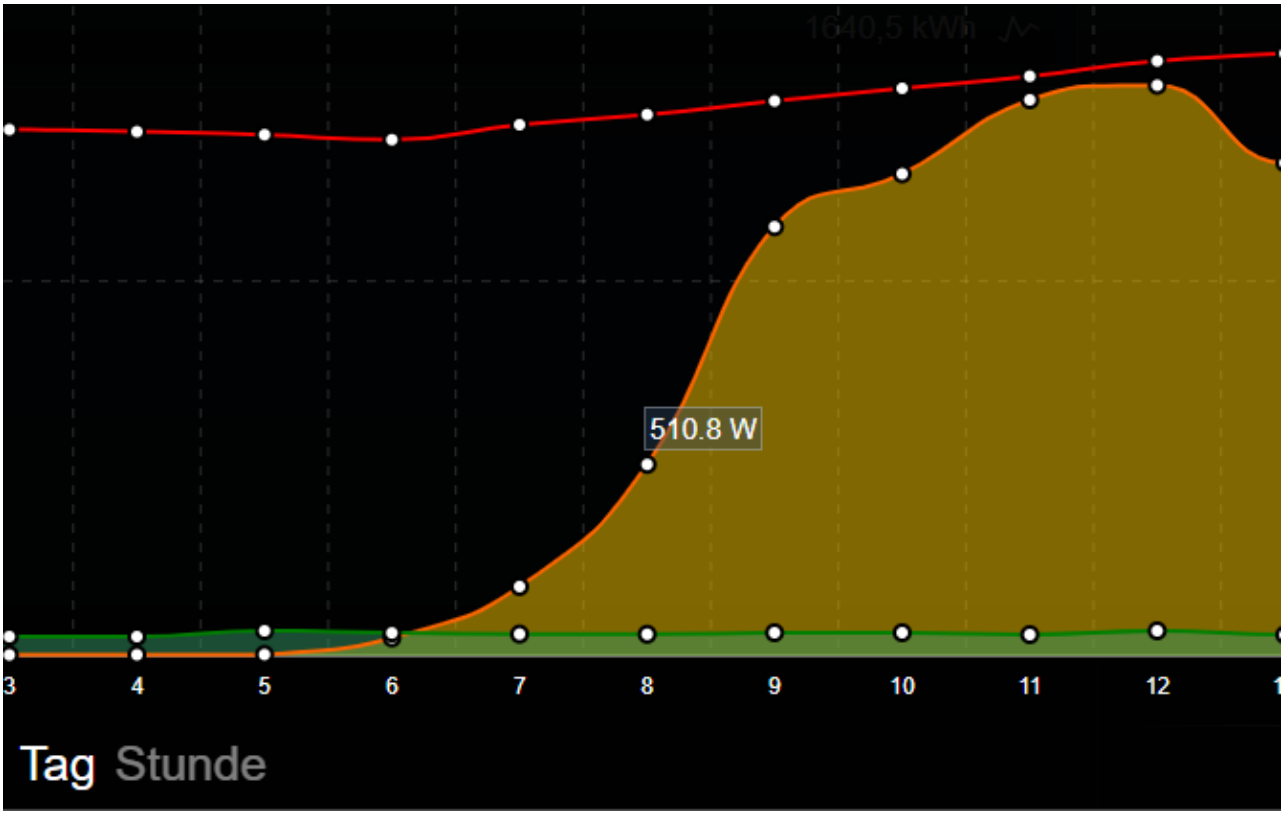



## SYMOS CONNECT

Mit SymOS Connect wird ein komfortabler Fernzugriff auf das SymOS der SymBox eingeführt. So kann von überall auf das SymOS zugegriffen, Updates durchgeführt oder IP-Symcon neu gestartet werden. Das SymOS Connect ist unabhängig von der Subskription, d.h. ein Zugriff ist immer möglich.

## HOVEREFFEKTE BEI GRAPHEN

Die Graphen im WebFront wurden weiter verbessert. Durch ein Hovern mit der Maus über einem Datenpunkt oder einem Balken wird der dazugehörige exakte Wert angezeigt. So bieten die Graphen einen deutlich erhöhten Komfort und können viel präziser ausgewertet werden.




### VERBESSERTE GERÄTEUNTERSTÜTZUNG

Die native Unterstützung von Geräten wurden an diversen Stellen verbessert. So wurde der KNX Import optimiert, so dass nun beispielsweise auch Teile von Exports eingebunden werden können. Für Z-Wave Geräte werden nun einige weitere Informationen der Geräte ausgelesen und können für vielfältige Anwendungsfälle verwendet werden. Bei xComfort werden nun auch die neuen Fensterkontakte CDWA/* unterstützt.

## MIGRATION

Bitte erstellen Sie auf jeden Fall ein Backup Ihres gesamten IP-Symcon Ordners, um auf die Version 5.3 zurück zu wechseln, sollten wider erwarten Probleme mit der Version 5.4 auftreten.

Falls Sie von der Version 3.4 updaten, lesen Sie bitte vorher die Migrationsanleitung zur Version 4.0: Migration auf Version 4.0 , Version 4.1: Migration auf Version 4.1 , Version 4.2: Migration auf Version 4.2 , Version 4.3: Migration auf Version 4.3 , Version 4.4: Migration auf Version 4.4 , Version 5.0: Migration auf Version 5.0 , Version 5.1: Migration auf Version 5.1 , Version 5.2: Migration auf Version 5.2 und Version 5.3: Migration auf Version 5.3 .

Es wird Raspbian Buster für Raspberry Pi erfordert! (Generelle Versionenübersicht)
Es wird Ubuntu 18.04 für Linux 64 Bit erfordert! (Generelle Versionenübersicht)
Für Microsoft Windows steht ausschließlich eine 64-Bit Version zur Verfügung.

# LISTE DER ÄNDERUNGEN

## ALLGEMEIN

Neu: PHP auf 7.3.16 aktualisiert
Neu: Bibliotheken aktualisiert
Neu: PHP Threads standardmäßig auf 25 erhöht
Neu: VoIP: Anrufe können gezielt per Skript angenommen/abgelehnt werden
Neu: DNS-SD: Ex* Varianten der Abfrage Funktionen hinzugefügt um Timeouts selber definieren zu können
Neu: Profile ~Playback und ~PlaybackPreviousNext zur Steuerung von Medienwiedergabe
Neu: Snapshot für mobile Apps filtert weniger Objekte (ggf. werden dadurch mehr Objekte in der App angezeigt)
Neu: Spezialschalter: NATPublicIP (Globale Public IP Option für den NAT-Support)
Neu: Spezialschalter: LogfileFilter (RegEx zum Filtern von Nachrichten)
Neu: Reagiere unter Windows auch auf das PRESHUTDOWN Event, wodurch der Dienst mehr Zeit zum Herunterfahren haben sollte
Neu: Maximale Ausgabegröße kann über ini_set('ips.output_buffer', Größe in Bytes); vergrößert werden.
Neu: Benachrichtigungen können global im Notification Control deaktiviert werden
Neu: Limitation für täglich versende Push Nachrichten (250). Per Spezialschalter (NotificationLimit) auf 1000 erhöhbar
Neu: Limitation für tägliche Reconnect's zum Connect Dienst (25). Per Spezialschalter (ConnectLimit) auf 100 erhöhbar
Neu: Google State Report Anfragen auf 1000 pro Tag limitiert
Fix: Abstürze beim Auflegen von VoIP Verbindung unter Windows gelöst
Fix: Z-Wave: Fehlende Übersetzungen nachgepflegt
Fix: Z-Wave: Assoziationen können wieder in Gruppen hinzugefügt werden
Fix: HomeMatic: AIR_PRESSURE darf auch vom Typ Float sein
Fix: KNX: XML-Import verbessert, wenn mehrere DPTs/DPSTs genannt werden
Fix: Seltener Deadlock in Verbindung mit Ereignissen korrigiert
Fix: ekey: InputID wird nun korrekt auf "nicht definiert" gesetzt
Fix: Modbus: Problem bei Modbus TCP/IP in Verbindung mit viele Geräte-IDs am selben Strang gelöst
Fix: Übersetzungen für z.B. Archiv vervollständigt
Fix: Legacy Heating Control hat das Resend Interval bei Experten-Skripten nicht berücksichtigt
Fix: IP-Symcon ist abgestürzt wenn eine php.ini von MacOS Systemen unter Windows/Linux verwendet wurde
Fix: DNS-SD: Modul limitiert die Anzahl der Einträge auf 60.
Fix: Connect: Unter Linux wurden fälschlicherweise die ssh config Dateien gelesen
Fix: Connect: "Per E-Mail senden" funktioniert wieder korrekt

## GERÄTE

Neu: Xcomfort: Unterstützung für neue xComfort Fensterkontakte (CDWA/*)
Neu: xComfort Konfigurator: Unterstützung für Temperatur-Sensoren am RC Touch
Neu: Z-Wave: RSSI/SecurityKey Information zu ZW_GetInformation hinzugefügt
Neu: Z-Wave: ZW_GetInformation liefert Zeitstempel, wann Pakete gesendet/empfangen wurden
Neu: KNX: XML-Import kann nun auch mit Exports von Teilen (z.B. nur ein Zimmer) umgehen

Neu: KNX: XML-Import ohne zugewiesen DPT bietet beim Erstellen einen Dialog zum auswählen der gewünschten DPT
Fix: Z-Wave: Erkennung von "fehlerhaften" Knoten verbessert
Fix: Z-Wave: Abfrage der Manufacturer Specific Klasse reduziert (fast alle Geräte antworten nicht korrekt und wir brauchen die Informationen eigentlich nicht)
Fix: Z-Wave: Fibaro RGBW (Neue Generation) funktioniert nun korrekt
Fix: Z-Wave: Fehlermeldungen im Debug bzgl. falscher Länge (sofern Security genutzt wurde) korrigiert
Fix: Z-Wave: Ignoriere ein weiteres Fibaro-Gerät bei Multilevel-Sensorabfrage
Fix: KNX: DPT 11 hat beim Senden vom Datum den Tag um 1 versetzt versendet
Fix: M-Bus: Kleine Wartezeit eingefügt, sofern mehrere Pakete hintereinander abgefragt werden
Fix: ModBus: Bessere Kompatibilität beim Auslesen von Coils bei Geräte die sich nicht sauber an die Spezifikation halten
Fix: KNX Discovery konnte Konfigurator nicht erstellen, wenn der IP Interfacename genau 30 Zeichen lang war
Fix: HomeMatic: Wenn der Socket inaktiv ist, fragt der Konfigurator die Raum-Informationen auch nicht ab
Fix: UVR1611 setzt den Port korrekt auf 40000

## PHP-SDK

Neu: Der Inhalt eines Konfigurators kann wie eine Liste für Aktionen verwendet werden, wird allerdings nicht als Eigenschaft gespeichert
Neu: Flag noDiscovery im Configurator ermöglicht es Geräte aus der Gerätesuche herauszuhalten, z.B. für ein generisches "Gerät hinzufügen"
Neu: Flags editable und deletable für List und Tree geben an ob ein einzelnes Element editiert/gelöscht werden darf
Neu: Datenfluss gibt Rückgabewerte von Children an den Parent weiter (z.B. Device an Splitter)
Neu: Parameter multiline für ValidationTextBox
Fix: minimum und maximum vom Numberspinner können wie vorgesehen per Dynamik angepasst werden
Fix: validate vom ValidationTextBox und PasswordTextBox können wie vorgesehen per Dynamik angepasst werden

## WEBFRONT

Neu: Hover-Werte bei Balkengraphen und Linengraphen (Nur normal, nicht bei HD/Raw)
Neu: Hinweis, wenn die Externe Seite HTTP Seiten laden will, obwohl das WebFront über HTTPS geladen wurde (Moderne Browser blockieren sogenannten Mixed Content)

## KONSOLE

Neu: Performance bei Drag & Drop und Anpassung vieler Baumelemente gleichzeitig, z.B. beim Filtern, optimiert
Neu: Suchen/Ersetzen-Dialog sortiert die angezeigten Skripte
Neu: Zeige Pfad zum Objekt als Tooltip des Tabs an
Neu: Die Variablenaktionen "Addieren" und "Subtrahieren" verwenden für den Summanden/Subtrahenden nun die Nachkommastellen des Profils
Neu: Wird der Filter im Objektbaum angepasst wird zum aktuell ausgewählten Objekt

gescrollt
Neu: Tabs können via Mittelklick geschlossen werden
Neu: "Wer benutzt dieses Profil?"-Dialog betrachtet nun auch Charts
Neu: Button im Debug Log hinzugefügt, um zur dazugehörigen Instanzkonfiguration zu springen
Neu: Skripteditor intern komplett umgebaut, dadurch sollte der Editor deutlich schneller laden
Neu: Nachrichtenlänge in Debuglogs auf 255 gekürzt und "Erweitern"-Funktion im Kontextmenü hinzugefügt
Neu: Frage nach dem Löschen von Instanzen nur einmal nach, ob unbenutzte Elterninstanzen gelöscht werden sollen, nicht nach jeder einzelnen Instanz
Neu: Kompaktmodus überarbeitet: Keine "Ränder" mehr und ein paar Größen angepasst
Neu: Dialog um in allen Skripten zu ersetzen fragt vorm Ersetzen, ob ungespeicherte Skripte gespeichert werden sollen
Neu: Dialog zum Auswählen eines Profils ist nun im klassischen Look von OK/Abbrechen-Dialogen
Neu: Der Inhalt von Mitteilungs-Popups kann markiert und somit auch kopiert werden
Neu: Initialer Wert von neuen Assoziationen wird auf Basis der aktuell höchsten Assoziation und Schrittweite vorgeschlagen
Neu: Bei Klick auf "Neu Laden" nach einem Verbindungsabbruch wird geprüft, ob die Konsole wieder verfügbar ist bevor neu geladen wird
Neu: Versuche bei mehreren geöffneten Konsolen den Snapshot von einer anderen offenen Konsole abzufragen. So sollte insbesondere das Öffnen von abgedockten Fenstern der Pro Konsole beschleunigt werden
Neu: Spalten im Objektbaum werden automatisch ausgeblendet, wenn der Platz nicht ausreicht
Neu: Referenzsuche im Module Store (unter Weitere Optionen für installierte Module)
Neu: Beim Löschen von Modulen im Module Store wird geprüft, ob dazugehörige Instanzen noch existieren
Neu: Alle X <Zeiteinheit> im zyklischen Ereignis wird mit 1 initalisiert
Neu: Filter im Statuswidget werden gespeichert
Neu: Ist beim Drücken von Strg + O im Skripteditor eine InstanzID ausgewählt, springe direkt zu dieser
Neu: "Nachfolgende Ereignisse ausführen" wird nun als Dropdown dargestellt
Neu: Einige Eingaben in Ereignisdialogen sind nun auf voller Breite
Neu: Die Eingabe von auslösenden Werten in auslösenden Ereignissen ist nun ein Nummernfeld bei numerischen Variablen
Neu: Im Objektbaum können Objekte nun via Rechtsklick -> Objekt verschieben verschoben werden
Neu: Formatiere Variablenwerte bei Erweiterten Bedingungen entsprechend des Variablenprofils
Neu: Markiere Filter im Objektbaum, wenn der Filter aktiv ist
Neu: Statusdialog ist nun modal und kann auch als Tab geöffnet werden
Neu: PHP Code der gewählten Aktion kann im "Befehle testen"-Dialog per Strg + C in die Zwischenablage kopiert werden
Neu: Variablenaktion "Auf auslösenden Wert schalten" kann nun bei prozentualen Variablen den Wert prozentual oder absolut übertragen. (50% von Intensity.255 werden 50% bei Intensity.100)
Neu: "Zu Objekt springen" kann im Skripteditor bei Auswahl einer ObjektID per Rechtsklick oder das Menü Textbearbeitung ausgeführt werden
Neu: Im Instanz erstellen/bearbeiten-Dialog passt sich die Ortsanzeige für Modultypen an, welche in eine eigene Sonderkategorie kommen
Neu: Im Archive Control wurde die Namensspalte auf eine Namens- und eine Ort-Spalte aufgeteilt
Neu: Wurde ein Log gestoppt, wird dies nicht mehr gespeichert, bei einem Neustart der

Konsole läuft das Log also weiter
Fix: Werden Spalten hinzugefügt oder entfernt, bleiben Sonderaktionen für spaltenspezifische Doppelklicks bei den korrekten Spalten
Fix: Sortierung der Transferprüfung im Module Control ist nun unabhängig von Groß- und Kleinschreibung
Fix: Info zur Variablenanzahl im Lizenz-Widget für die Unlimited Edition passend formatiert
Fix: Schöne Fehlermeldung beim Versuch die Subskriptionsüberprüfung ohne Lizenz zu aktivieren
Fix: Variableneditor ist nun auf 100 Einträge beschränkt
Fix: Größe des Wertefensters des Variableneditors korrigiert
Fix: Icons für den Icon-Auswahl-Dialog laden deutlich schneller
Fix: Anpassung der Fenstergröße in Safari werden in der Formatierung korrekt umgesetzt
Fix: Biete im Skripteditor keine Autovervollständigung innerhalb von Strings oder Kommentaren
Fix: Formulierung bei Nachfolgender Ausführung von ausgelösten Ereignissen optimiert
Fix: Dialog zur Discovery-Auswahl scrollt nun wie vorgesehen nur den Inhalt des Baumes
Fix: Aktionsbeschreibung in Ereignisdialogen passt sich auch bei Variablen ohne Aktion korrekt an
Fix: Die Gerätesuche wertet Konfigurator-Einträge ohne InstanzID wie vorgesehen als nicht erstelltes Gerät aus
Fix: Bewegbare Dialoge werden korrekt aufgeräumt nachdem sie geschlossen wurden
Fix: Dropdowns werden wie vorgesehen geschlossen, wenn auf einen bewegbaren Dialog geklickt wird
Fix: Prüfung ob Von und Bis bei zyklischen Ereignissen korrekt gesetzt sind
Fix: Einige verwendeten Bibliotheken aktualisiert
Fix: Ausklappende Kontextmenüs auf kleinen Bildschirmen werden korrekt formatiert
Fix: Nummern in Listen, z.B. für NumberSpinner, werden korrekt numerisch sortiert
Fix: Ladenachrichten beim Start der Konsole werden nun wie vorgesehen auf englisch übersetzt
Fix: Warte mit dem neu laden der Instanzkonfiguration bis die Antwort auf ApplyChanges zurückkommt
Fix: Debuglog versucht nicht mehr nach Verbindungsabbruch das Debug aktiv zu halten
Fix: Werden bei einem Löschvorgang ein Objekt und seine Kinder markiert, kommt es nicht mehr zu einer Fehlermeldung
Fix: Ändert sich ein übergeordnetes Objekt einer auslösenden Variable eines Ereignisses, wird die Bezeichnung des Ereignisses wie vorgesehen aktualisiert
Fix: Verhindere Verbindungsabbruch bei Konfiguratoren mit sehr vielen Geräten
Fix: Generierte Aktionsskripte werden übersetzt
Fix: Formulierung im Lizenz-Widget angepasst, so dass deutlich wird, dass nicht die Lizenz, sondern die Subskription abläuft
Fix: "Befehl einfügen" formatiert den eingefügten Code auch mehrzeilig wie vorgesehen
Fix: Prüfe ob der zurückgegebene Link eines Buttons Zeilenumbrüche enthält und gebe ggf. passende Fehlermeldungen
Fix: Korrekte HEX-Darstellung im Debug-Log - Es wird nicht versucht gelegentlich Zeichen zusammenzuziehen, da diese als UTF-8 interpretiert werden
Fix: Anzeige vom Teilpfad zur Tabunterscheidung wird auch bei mehreren gleichnamigen Tabs gleichzeitig korrekt ausgewertet
Fix: Ident wird wie vorgesehen aktualisiert
Fix: Eigenschaften, die via GetConfigurationForParent gesetzt werden, werden auch übernommen, falls die dazugehörigen Elemente noch inaktiv sind, z.B. in einem eingeklapptem ExpansionPanel
Fix: Wurde in der HTML-Darstellung der Skriptausgabe auf eine externe Seite gelinkt, kann das Skript wie vorgesehen erneut ausgeführt werden
Fix: Ausgelöste Ereignisse mit einer Boolean-Variable ohne Profil als Auslöser funktionieren wieder wie vorgesehen

Fix: Gruppen in Wochenplänen werden nun in korrekter Reihenfolge gespeichert
Fix: Initialer auslösender fester Wert bei ausgelösten Ereignissen funktioniert bei Boolean-Variablen ohne Profil wie vorgesehen
Fix: Bei Wechsel der auslösenden Variablen sollte der bisher ausgewählte Wert wie vorgesehen stets passend zur neuen Variable konvertiert werden
Fix: Ein seltener Fehler beim Ändern eines ausgelösten Ereignisses auf einen anderen Variablentyp gefixt
Fix: Wird im Skripteditor als Teil der Ereigniskonfiguration Enter gedrückt, wird dies nicht mehr als Bestätigung des Dialogs gewertet
Fix: Tabs hatten manchmal einen falschen Namen (Name not yet loaded)

## SYMOS

Neu: SymOS Connect. Direkter Zugriff auf SymOS über unsere Cloud*
Neu: Ohne SymOS Kennwort sind keine Anfragen über SymOS Connect erlaubt
Neu: SymOS Web-Server komprimiert Daten per GZIP und unterstützt Caching
Neu: IP-Symcon wird bei einem Crash automatisch neu gestartet
Neu: Weiterleiten ins WebFront (SymOS wird dabei übersprungen)
Neu: Integrator Nachricht kann über Einstellungen gesetzt werden
Neu: Checkbox um #integrator Nachricht zu deaktivieren
Neu: Unterbau von Polymer 1 auf Polymer 3 migriert (Im Prinzip fast neu geschrieben )
Neu: Kernel auf 4.14.114 aktualisiert
Neu: #delete heißt nun #files
Neu: Splashscreens entfernt
Fix: Dateien werden nach Datum sortiert
Fix: Sofern der Speicherplatz gering ist, werden alte IP-Symcon/SymOS Versionen gelöscht
Fix: Sonderzeichen beim SymOS Kennwort konnten (selten) Probleme beim Einloggen machen
Fix: Löst hoffentlich das Problem, dass DHCP nach Auslaufen der Lease keine IP-Adresse mehr bezogen hat
Fix: E-Mail Adresse mit längeren Domain Namen (z.B. swiss) werden akzeptiert

## PRO KONSOLE

Neu: Die Pro Konsole wurde veröffentlicht

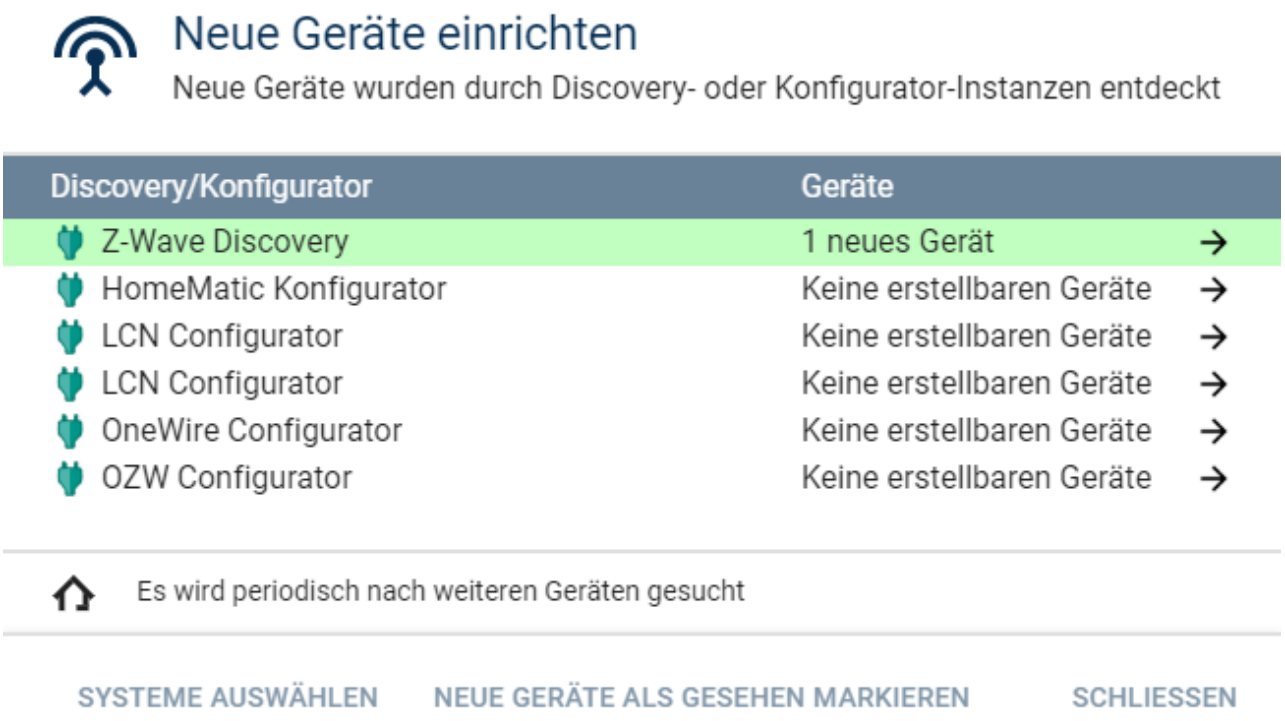
# V5.2->V5.3 (Q4/2019)

## WESENTLICHE ÄNDERUNGEN

Gerätesuche
Verbesserter KNX Import
Konfigurator für M-Bus
Komfortverbesserungen in der Verwaltungskonsole

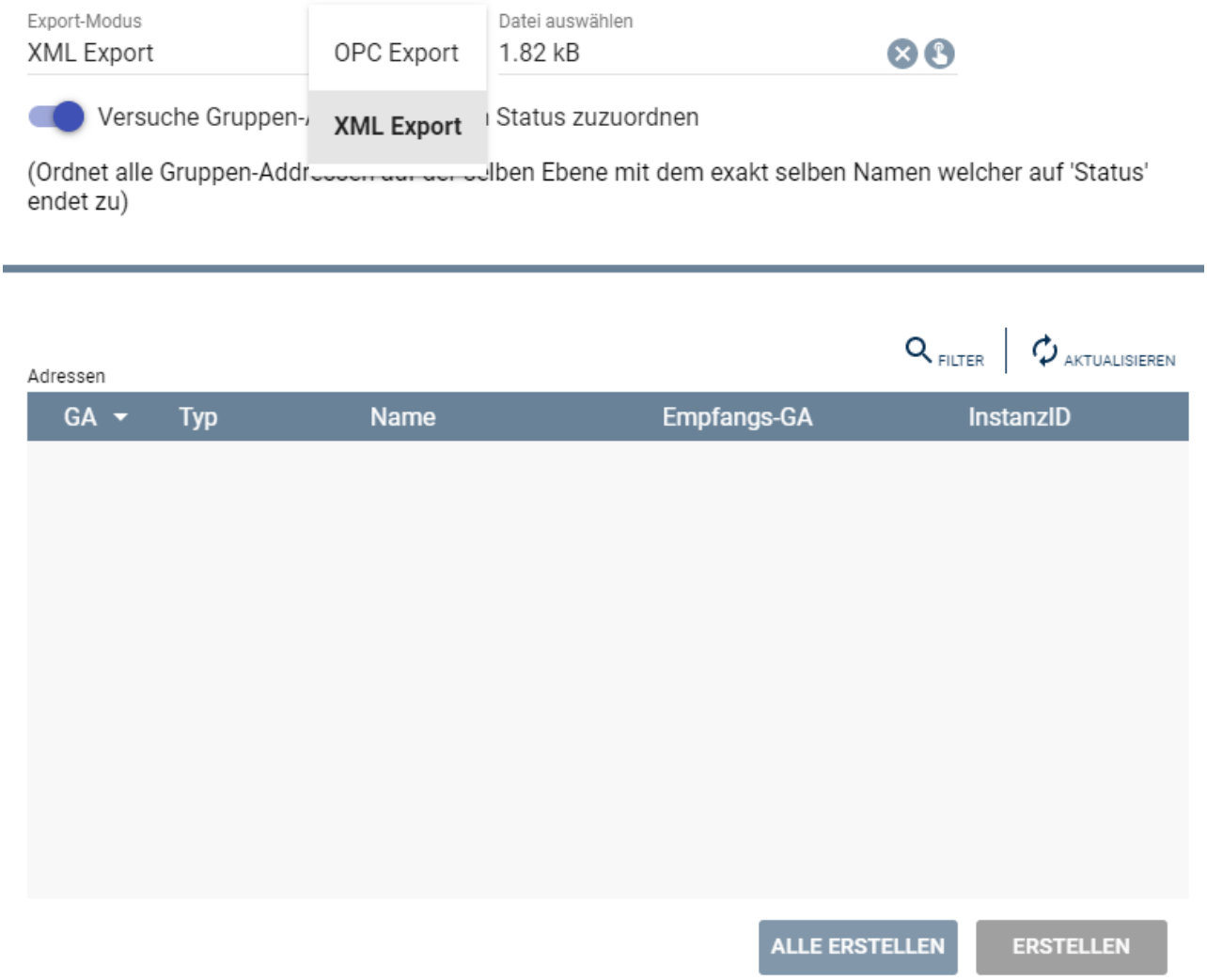
### GERÄTESUCHE

Mit der neuen Version sucht IP-Symcon automatisch in regelmäßigen Abständen nach neuen Geräten und zählt diese in der Gerätesuche auf. Nach der Systemauswahl erfolgt die Detektion von Gateway und Geräten durch die Gerätesuche und diverse neue Discovery- und Konfigurator-Instanzen bei den meisten Systemen automatisch und es müssen nur noch die zu erstellenden Geräte ausgewählt werden. Weitere Informationen gibt es in der Dokumenentation zur Gerätesuche und in unserem Grundlagen-Webinar .



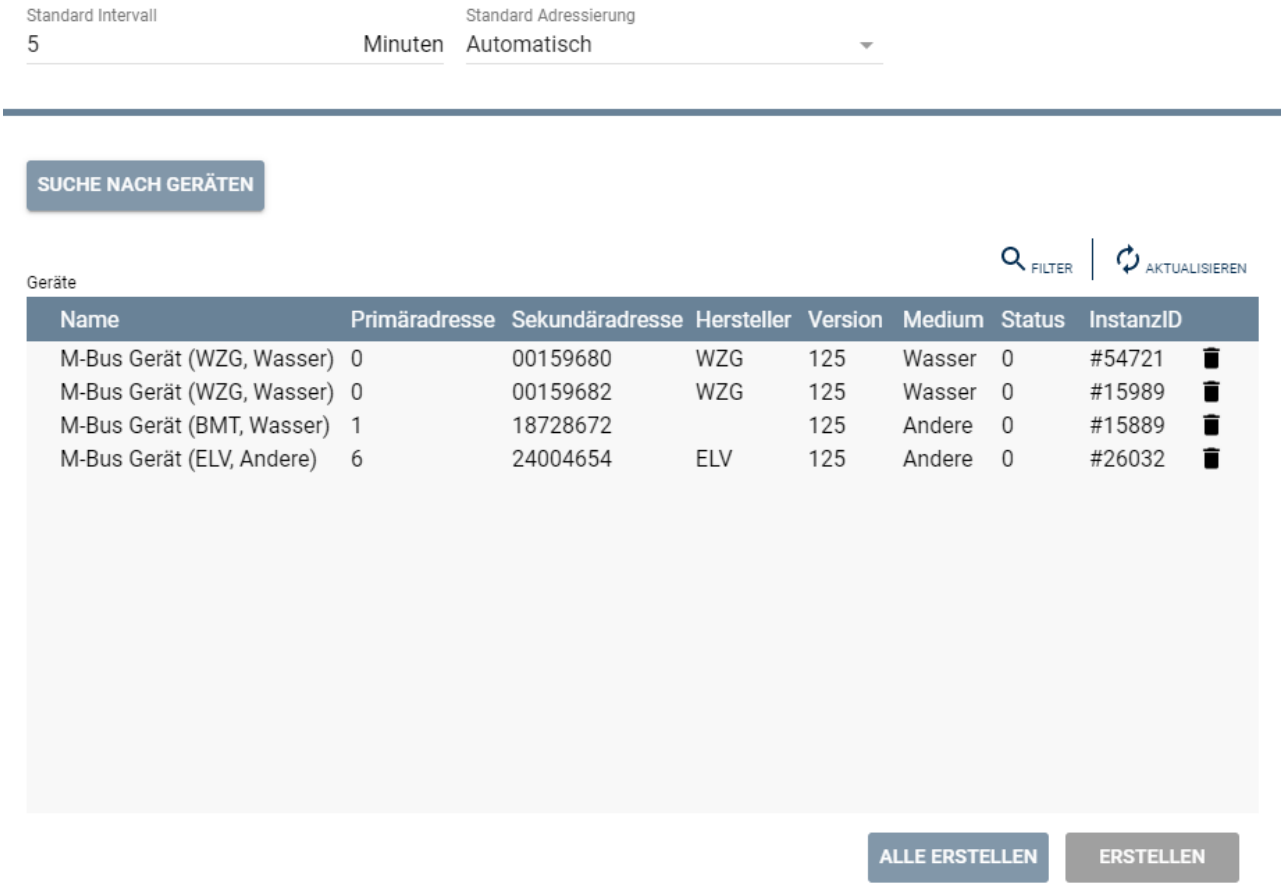

### VERBESSERTER KNX IMPORT

Die Import-Funktion im KNX-Konfigurator wurde erweitert. Zusätzlich zum OPC Import ist es nun auch möglich die modernen DPTs aus einer generierten XML-Datei zu importieren und danach komfortabel im Konfigurator anzulegen. Es stehen dazu folgende Video Tutorials zur Verfügung: XML-Export mithilfe der ETS , XML-Export in IP-Symcon verwenden



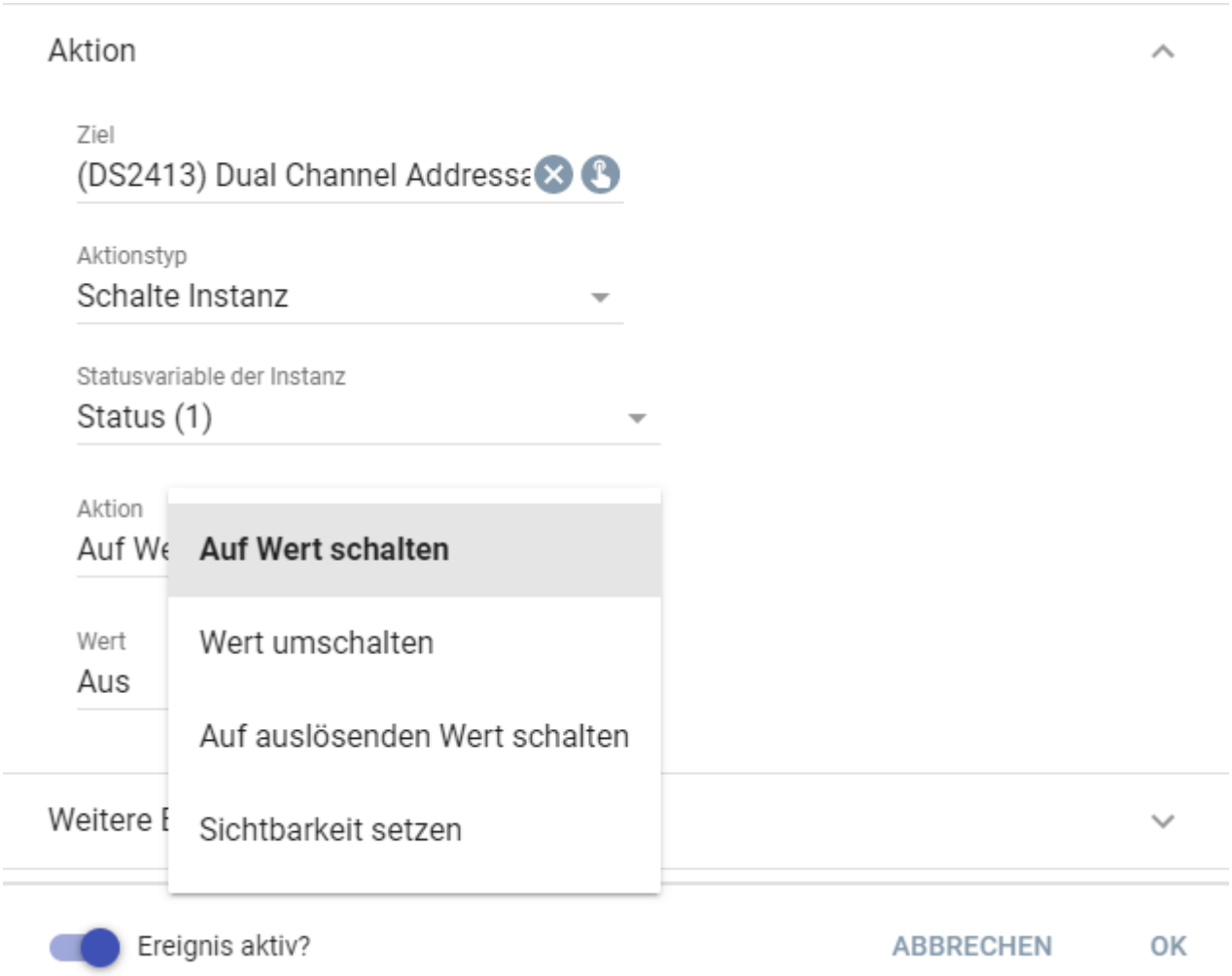

## KONFIGURATOR FÜR M-BUS

Auch für M-Bus gibt es nun einen Konfigurator. Dieser sucht den gesamten Bereich der Sekundäradressen von M-Bus ab und findet so alle angeschlossenen M-Bus-Geräte, damit diese unkompliziert in IP-Symcon erstellt werden können.




## KOMFORTVERBESSERUNGEN IN DER VERWALTUNGSKONSOLE

Diverse kleine Erweiterungen haben den Komfort der Verwaltungskonsole verbessert. So ist es nun möglich die Reihenfolge der Tabs via Drag & Drop anzupassen. Der Browsertitel passt sich an den Namen der Hauptkategorie an. Die Aktionen für Variablen in Ereignissen wurden erweitert. Insbesondere ist es möglich bei ausgelösten Ereignissen den auslösenden Wert direkt weiterzuleiten. So können unkompliziert diverse automatisierte Verhalten implementiert werden.




## MIGRATION

Bitte erstellen Sie auf jeden Fall ein Backup Ihres gesamten IP-Symcon Ordners, um auf die Version 5.2 zurück zu wechseln, sollten wider erwarten Probleme mit der Version 5.3 auftreten.

Falls Sie von der Version 3.4 updaten, lesen Sie bitte vorher die Migrationsanleitung zur Version 4.0: Migration auf Version 4.0 , Version 4.1: Migration auf Version 4.1 , Version 4.2: Migration auf Version 4.2 , Version 4.3: Migration auf Version 4.3 , Version 4.4: Migration auf Version 4.4 , Version 5.0: Migration auf Version 5.0 , Version 5.1: Migration auf Version 5.1 und Version 5.2: Migration auf Version 5.2 .

Beim Upgrade in Windows wird die Konfiguration von IP-Symcon analog zu anderen Betriebssystemen in den ProgramData-Ordner verschoben. Dies geschieht automatisch.
Es wird Raspbian Buster für Raspberry Pi erfordert! (Generelle Versionenübersicht)
Es wird Ubuntu 18.04 für Linux 64 Bit erfordert! (Generelle Versionenübersicht)
Für Microsoft Windows steht ausschließlich eine 64-Bit Version zur Verfügung.

# LISTE DER ÄNDERUNGEN

## ALLGEMEIN

Neu: Windows Version speichert alle Daten ab sofort unter C:\ProgramData\Symcon (Mehr Informationen )
Neu: Wenn IP-Symcon abstürzt wird unter Ubuntu/RPi/MacOS versucht automatisch ein Stacktrace im /var/lib/symcon/minidump Ordner zu speichern.
Neu: MiniDumps unter Windows sind jetzt testweise signifikant kleiner
Neu: HasAction Funktion zum Prüfen, ob Variable eine gültige Aktion besitzt
Neu: SSDP Control (Beispiel: YC_SearchDevices($id, "upnp:rootdevice"))
Neu: DNS-SD kann Geräte suchen
Neu: DNS-SD nutzt unter Linux die native Avahi API
Neu: UDP Socket unterstützt nun korrekt Broadcasting über alle vorhandenen Interfaces
Neu: Datenfluss angepasst, sodass mehrere alternative Pakete versendet werden können und nur das "beste" an die Instanz ausgeliefert wird. (Bisher nicht für PHP-Module verfügbar)
Neu: Update-Kanal wird unter Linux auch unter /etc/apt/sources.list.d/* geprüft
Neu: WFC_PushMessageEx um eine WebSocket Nachricht an einen bestimmten verbundenen Client via (RemoteIP/RemotePort) zu senden.
Neu: Die Docker, Ubuntu und Raspberry Pi Versionen nutzen für DNS-SD nun Avahi Core anstatt Avahi vom Betriebssystem. (Dadurch entfällt die letzte große Paket-Abhängigkeit)
Neu: Ubuntu/Raspberry Pi zeigt nach der Installation den Pfad zur Konsole an
Neu: MQTT: Retain
Neu: MQTT: Session Management (Siehe Timeout und Queue Limit Einstellungen im Server)
Neu: MQTT: Last Will/Testament
Neu: MQTT: Publish eines Clients wir nur in IP-Symcon ausgewertet und nicht an die anderen Clients weitergeleitet
Neu: Im Tray unter Windows gibt es einen Button zum Öffnen des ProgramData Pfades
Fix: Connect Dienst hat ein Timeout von 120 Sekunden anstatt von 10 Sekunden (Das sollte die Abbrüche der Verwaltungskonsole bei langen Operationen lösen)
Fix: MessageQueue wird beim Beenden vollständig abgearbeitet (Löst Problem mit fehlenden Werten in der settings.json die direkt vor dem Beenden geschrieben wurden)
Fix: "Enable Broadcast" aus UDP Socket entfernt, das dies nur für den Multicast Socket anwendbar war
Fix: Bei Modul-Updates wurde der Instanz-Status nicht korrekt zurückgesetzt, sodass diese weiterhin als Aktiv gemeldet wurden, obwohl diese gar nicht verfügbar waren
Fix: Potenzieller Fix für seltenes Crash-Problem, wenn man oft I/Os schließt/öffnet unter Linux
Fix: MQTT: Unvollständige/Fehlerhafte Pakete konnten IP-Symcon zum Absturz bringen
Fix: VoIP: VoIP_SendDTMF erlaubt nun auch * und #
Fix: IntervalBox'en und 'label' auf NumberSpinner und 'caption' aktualisiert
Fix: MQTT leitet Nachrichten nicht an den Ursprung weiter um Endlosschleifen zu verhindern
Fix: Fehlerhafte Module konnten nicht über den Store gelöst werden
Fix: MQTT: Geräte ohne ClientID bekommen eine passende ClientID zugewiesen
Fix: MQTT: Last Will mit QoS > 0 wird aktuell mit QoS = 0 versendet

## GERÄTE

Neu: Discovery Instanzen für alle Symcon LAN Gateways (IPS-868, M-Bus, 1-Wire, Z-Wave, EnOcean)

Neu: HomeMatic Discovery
Neu: IPS-868 Konfigurator
Neu: EnOcean D2-11-*
Neu: EnOcean EEP D2-05-00, EEP D2-05-01, D2-05-02
Neu: M-Bus Konfigurator, welcher alle angeschlossenen M-Bus Geräte suchen kann
Neu: M-Bus Geräte können über Sekundär-Adressierung angesprochen werden
Neu: M-Bus: Anzahl der gelesenen Pakete kann limitiert werden
Neu: M-Bus Konfigurator: Standard Intervall / Standard Adressierung kann vorgegeben werden
Neu: KNX Konfigurator kann XML Exports der ETS Gruppenadressen einlesen (Damit werden auch die neuen DPT Instanzen erstellt)
Neu: KNX Konfigurator in der Legacy Konsole entfernt, da dieser nicht mehr mit IP-Symcon 5.3 kompatibel ist
Neu: KNX Konfigurator versucht GA's mit dem Status-Suffix automatisch als Rückmelde GA zuzuordnen, sofern gewünscht
Neu: KNX/DPT Instanzen haben im Namen zusätzlich den generischen Datentyp
Neu: LCN: Setzen vonn Schwellwerten (1x5 über Register 0, 4x4 über Register 1-4)
Neu: Z-Wave: Es können benutzerdefinierte Werte bei Parameters aus der ZWDB angegeben werden
Neu: VoIP: VoIP_SendDTMF zum Senden von DTMF Tönen
Fix: VoIP: Besserer Fix für das 401 Problem sobald die Session abläuft
Fix: M-Bus: Position wird über mehrere Pakete hinweg korrekt gesetzt
Fix: Z-Wave: Öffnen der Z-Wave Instanzen bei vielen angelernten Knoten sollte wesentlich schneller gehen
Fix: Z-Wave: Problem beim Öffnen von Instanzen mit Zipato RFID/Lock gelöst
Fix: Z-Wave: Workaround für fehlerhafte Fibaro Geräte, welche eine schlampige Implementation der Multi Channel Association Klasse haben. Wir entfernen diese Klasse entsprechend, sodass IP-Symcon die alten simplen Association verwendet
Fix: Z-Wave: Option für "Simple Association erzwingen" wird nur noch angezeigt, wenn diese auch sinnvoll ist (Benötigt vorhanden sein der Multi Channel Klasse und die Multi Channel Association Klasse ab Version 3)
Fix: LCN Display: Anzeigedauer wird nicht in Sekunden angeben, sondern ist von der Modul-Konfiguration abhängig
Fix: LCN: Ampere-Werte werden nun korrekt in Milliampere dargestellt
Fix: xComfort 1-Kanal Heizaktor kann nun auch korrekt ausgeschaltet werden
Fix: xComfort: Router wird vom Konfigurator korrekt erkannt

## KONSOLE

Neu: Die Glocke oben rechts in der Verwaltungskonsole zeigt nun erstellbare Geräte von Discovery und Konfigurator Instanzen an
Neu: Im Module Control kann die Migration auf den Store für beliebige Module deaktiviert werden
Neu: Ausgelöste Ereignisse können den Ereignis-Wert direkt an das Ziel-Objekt weiterleiten.
Neu: Zeige im Skripteditor einen Hinweis, wenn Teile der Skriptausgabe durch den HTML-Filter versteckt werden
Neu: Wird der Aggregationstyp einer Variablen geändert, dann wird gefragt, ob diese reaggregiert werden soll
Neu: Das Schnellausführungs-Widget speichert das Skript der letzten Ausführung
Neu: Browsertitel passt sich an den Namen der Top-Level-Kategorie an
Neu: Schlägt die Erstellung einer Instanz via Konfigurator fehl, dann räumt der Konfigurator auf

Neu: Tabs können nun per Drag & Drop neu angeordnet werden
Neu: Icons im Objektbaum haben wieder einen Bubble-Effekt beim Klicken
Neu: Lade Konfiguration nicht neu, wenn das Übernehmen der Änderungen fehlschlägt
Neu: Verwaltungskonsole kann via Browser als "App" installiert werden
Neu: Beim Hovern über einem Variablenwert und innerhalb des Variable-bearbeiten-Dialogs werden Rohwert und formatierter Wert angezeigt
Neu: Widget um Absturzberichte zu verschicken
Neu: Update-Buttons für die verbleibende Subskriptionsdauer im Info-Dialog und im Lizenz-Widget
Neu: Tabs mit gleichem Namen erhalten Teile des Pfades im Tabnamen, damit die Tabs differenziert werden können
Neu: Tutorial an neue Gerätesuche Mechanismen angepasst
Neu: Geräte können in Konfiguratoren als Gesehen/Ungesehen markiert werden
Neu: Als gesehen markierte Geräte werden in Konfiguratoren initial nicht angezeigt
Neu: "Springe zum Auslöser" für auslösende Ereignisse per Rechtsklick im Objektbaum
Fix: In der Legacy Konsole können Wochentage bei monatlichen Ereignissen korrekt gesetzt werden
Fix: Spalten gehen nicht mehr kaputt, wenn man die Positionsspalte einblendet, im Gegenzug eine andere Spalte ausblendet und dann neu Lädt (Dies ist der Webinar Bug!)
Fix: Neu angelegte Ereignisse konnten unter bestimmten Bedingungen nicht mehr geöffnet werden
Fix: Progressbar mit "indeterminate" Zustand wird korrekt angezeigt
Fix: Übersetze suffix des NumberSpinner wie vorgesehen
Fix: Lade Konfiguration beim Verwerfen der Änderungen neu und lädt somit auch die Einträge von Feldern ohne Eigenschaft neu
Fix: Widget für statistische Daten kann sich nicht mehr mehrfach öffnen, wenn noch keine Entscheidung getroffen wurde
Fix: Korrigiere Dialoge für Microsoft Edge
Fix: Zoomen sollte auch in iOS 13 wie vorgesehen deaktiviert sein
Fix: Der mobile Safari von iOS 13 wird nun wie vorgesehen korrekt erkannt
Fix: In der Timerübersicht werden die Statusicons nun wie vorgesehen eingefärbt
Fix: In der Timerübersicht werden die Timer wie vorgesehen nach Kategorien gruppiert
Fix: Blockiere die Auswahl eines Links als Elternobjekt
Fix: Skripte, die kein Variablentemplate sind, werden nun wie vorgesehen in der Ereignisübersicht einer Instanz angezeigt
Fix: Wird der Zeitzonen-Hinweis auf Wunsch des Benutzers nicht angzeigt, wird dennoch wie vorgesehen die Zeitzone des Servers eingestellt
Fix: closeCaption bei Popups wird nun wie vorgesehen übersetzt, dies betrifft insbesondere Z-Wave
Fix: Der Ausklapp-Button einer eingeklappten Kategorie im Objektbaum verschwindet nun wie vorgesehen wenn das letzte Kind gelöscht wird
Fix: Bestimmte Benutzername/Passwort Kombinationen führen im Chrome nicht mehr zu einem Login-Fehler
Fix: Eigenschaften in Expansion Panels werden wie vorgesehen geladen
Fix: Beim Versuch ein IPSStudio-Formular zu editieren kommt ein Hinweis, dass dies nur mit IPSStudio möglich ist
Fix: Download-Button im Dokument-Dialog wird nicht mehr unterstrichen
Fix: Die Beschreibung eines Objektes wird nun einheitlich als Beschreibung bezeichnet, nicht mehr als Info
Fix: Der Inhalt von Toolbars kann nicht mehr überlaufen. Dies betraf insbesondere die Suchfelder im Kompaktmodus
Fix: Die caption des Formularelements Label wird nun korrekt aktualisiert, auch wenn im Text Links enthalten waren
Fix: Jedes SelectDateTime-Feld hat seinen eigenen Wert und stellt nicht mehr die Werte von vorherigen SelectDateTime-Feldern dar

Fix: Zeige Mehrfachoptionen in einem Konfigurator auch bei mehrfachem Öffnen korrekt an
Fix: location wird nun auch bei Mehrfachoptionen im Konfigurator wie vorgesehen ausgewertet
Fix: value kann nun im Formularfeld Select wie vorgesehen verwendet werden
Fix: SelectColor lädt nun wieder wie vorgesehen Wertevorgaben
Fix: Nach dem Suchen&Ersetzen lädt die Konsole die betroffenen Skripte neu
Fix: Beim Schließen eines Tabs wird nun nicht mehr gelegentlich auf den ersten Tab gewechselt
Fix: SelectLocation überarbeitet, nun verschwindet der Marker nicht mehr nach einem Neuladen und die Bedienung ist flüssiger
Fix: In Firefox wird die Größe des Editieren-Dialogs von Listen nun korrekt bestimmt
Fix: Listen mit einer Spalte können wie vorgesehen im elements-Bereich verwendet werden
Fix: Achte bei der Sortierung in Listen wie vorgesehen nicht auf Groß- und Kleinschreibung

### WEBFRONT

Fix: RTSP Streams wurden bei bestimmten WebFront Kennwörtern nicht geladen
Fix: Dokumente wurden nicht automatisch aktualisiert, sofern diese z.B. über IPS_SetMediaContent verändert wurden

### PHP-SDK

Neu: minimum, maximum und onChange sind beim HorizontalSlider nun optional
Neu: Neuer Parameter stepsize für den HorizontalSlider
Neu: Im Konfigurator können individuelle Namen für Statusvariablen definiert werden
Neu: Neues Formularfeld OpenObjectButton
Neu: Validierung für NumberSpinner, ValidationTextBox und PasswordTextBox
Neu: ExpansionPanel und PopupButton unterstützen nun onClick
Neu: List und Tree unterstützen nun onEdit, onDelete und onAdd
Neu: Skripte in der Instanzkonfiguration können nun die gesamte Elemente vom Typ List oder Tree komplett iterieren und nicht nur auf den aktuell gewählten Wert zugreifen
Neu: Parameter "link" für Label kann einstellen ob URLs in der Caption als Links dargestellt werden oder nicht
Neu: Unterstütze die Option width für fast alle Formularfelder
Fix: Slider, welcher initial per value gesetzt wurde, sendet diesen Wert nicht mehr direkt per onChange wieder raus

## ÄNDERUNGEN 5.3.X

### ALLGEMEIN

Neu: KNX DPT 9.029 hinzugefügt
Neu: SSDP Announce beinhaltet die genutzte Platform
Neu: Performance vom Datenfluss verbessert
Neu: HomeMatic: Keep Alive Timeouts können konfiguriert werden (0 = Aus)
Neu: M-Bus Konfigurator: Standardeinstellung für die Paketlimiterung verfügbar
Neu: Neue Spezialschalter zum Steuern der WebSocket/RTSP Limitationen (ServerSoftQueueSizeLimit, ServerSoftQueueBytesLimit,

ServerSoftQueueTransferTimeout, ServerHardQueueSizeLimit, ServerHardQueueBytesLimit, ServerHardQueueTransferTimeout) *
Neu: Unterstützung für neue xComfort Fensterkontakte (CDWA/*)
Fix: PHP zurück auf 7.3.5, um ggf. Abstürze mit PHP 7.3.8 auszuschließen. (Nur Linux, SymBox, Ubuntu, Docker)
Fix: M-Bus: Sofern zwei Geräte mit der selben Sekundäradresse am Bus waren, konnte IP-Symcon abstürzen
Fix: SSDP: Auf einigen Systemen konnte die Instanz nicht korrekt erstellt werden
Fix: LCN: LCN Konfigurator (Module) hat einige Instanzen nicht korrekt erkannt
Fix: Z-Wave: Dialog für unterstützte Kommando-Klassen hat unterstützte Version falsch angezeigt
Fix: KNX: Profil bei DPT 1.x korrigiert
Fix: KNX: Profil bei DPT 9.028 korrigiert
Fix: MQTT: Problem bei der Kommunikation mit MQTT Client auf ESP Basis korrigiert
Fix: Discovery Instanzen zeigen nun auch eingerichtete Konfiguratoren (welche in IP-Symcon sind, jedoch nicht im Netzwerk)
Fix: Reaggregation aller Variablen konnte manchmal zwischendurch abbrechen
Fix: Z-Wave Geräte, die defekte SerienNummern liefern, sollten die webbasierte Konsole nicht mehr beeinflussen (Neu „Laden“ ist erforderlich)
Fix: Potenzieller VoIP Crash korrigiert
Fix: Potenzieller Connect-Dienst (Reconnect) Crash korrigiert
Fix: KNX Device mit EIS10, 16 Bit, Signed kann wieder eingestellt werden
Fix: KNX XML Import nutzt nun " Status", "-Status" und ".Status" in den GAs als Trigger für Status GAs
Fix: Der Windows Installer lässt bei Problemen kein Abbrechen zu und verweist auf unseren Support
Fix: Der Windows Installer stellt sicher, dass das IP-Symcon Tray und IP-Symcon Verwaltungskonsole vor der Migration beendet wurden
Fix: Archiv: Bei IP-Symcon Neustarts konnte bei der Rekonstruierung der Zähleraggregation in seltenen Fällen eine fehlerhafte Berechnung auftreten
Fix: Archiv: Werte, die im selben Zeitstempel passieren wurden nicht korrekt im Minimum/Maximum berücksichtig
Fix: Z-Wave: "Werte auslesen" Button bei "Benutzerdefinierten Parametern" funktioniert wie vorgesehen
Fix: Z-Wave: Sporadisch wurde beim Hinzufügen/Entfernen von Geräte nicht angezeigt, dass das Gateway bereit ist
Fix: Z-Wave: Z-Wave Instanzen, welche nicht mehr auf dem Gateway vorhanden sind, können direkt gelöscht werden (ohne Test-Pakete zu versenden)
Fix: Der Update-Kanal konnte aus dem sources.list.d Ordner nicht korrekt ermittelt werden
Fix: Die Migration der Daten unter Windows ignoriert die setup.exe, welche bis IP-Symcon 4.2 erstellt wurde
Fix: HomeMatic: Wenn eine "fremde" HomeMatic Instanz keine "Protocol" Property besaß konnte der Konfigurator nicht geöffnet werden
Fix: M-Bus: Suchfunktion im Konfigurator weiter verbessert
Fix: M-Bus: Der vorgeschlagene Name beinhaltet die Sekundäradresse
Fix: Z-Wave: Sofern ungültige Secure-Pakete empfangen wurden konnte IP-Symcon abstürzen
Fix: KNX: ~Switch Profil wird überall konsistent verwendet
Fix: Fix für die Abstürze in Verbindung mit dem WebSocket Rückkanal / RTSP Streams
Fix: SSDP: Anfrage auf 5 Sekunden erweitert, da einige Geräte zu langsam antworten
Fix: DNS-SD: Wartezeit auf 500ms erhöht um Zuverlässigkeit zu verbessern

Fix: EnOcean EEP D2-01-xx mit Leistungsmessung aber ohne Energieausmessung wird nun korrekt ausgewertet
Fix: Multicast Socket sollte wieder wie erwartet funktionieren, auch wenn nur ein Interface in der BindIP gewählt ist
Fix: Query String wird in $_GET Variable korrekt kodiert
Fix: Connect: Verbindungsabbrüche sollten nun schneller erkannt werden
Fix: PHP Module: Type Hints zu "array/object" sind erlaubt und werfen keine Warning
Fix: Sys_GetMemoryInfo unter Linux korrigiert (Available Memory wurde falsch berechnet)
Fix: M-Bus: Negative Werte werden nun nach ISO EN13757-3 ausgewertet
Fix: HasAction war nicht korrekt vorhanden
Fix: KNX: DPT 11 hat beim Senden vom Datum den Tag um 1 versetzt versendet
Fix: PHP Inkludepfad enthälte immer den Pfad zum "scripts" Ordner, damit auch in Ereignissen Dateien korrekt per include geladen werden können
Fix: Z-Wave: Empfange von einigen Kommandoklassen (z.B. CRC_16) auch Nachrichten im Untergerät, wenn nur das Hauptgerät diese als unterstützt melden.
Fix: Z-Wave: Bessere Überprüfung, ob die empfangenen Daten vollständig sind
Fix: Z-Wave: Erkennung von "fehlerhaften" Knoten verbessert
Fix: Z-Wave: Abfrage der Manufacturer Specific Klasse reduziert (fast alle Geräte antworten nicht korrekt und wir brauchen die Informationen eigentlich nicht)
Fix: Z-Wave: Fehlermeldungen im Debug bzgl. falscher Länge (sofern Security genutzt wurde) korrigiert
Fix: Abstürze bei RTSP Streams über eine SSL Verbindung korrigiert
Fix: Aktualisierung-Widget zeigt neue Versionen wieder korrekt an
Fix: DNS-SD Modul limitiert die Anzahl der Einträge auf 60.

## WEBFRONT

Fix: WebFront Editor zeigt unsichtbare Elemente an, sodass diese auch wieder sichtbar geschaltet werden können
Fix: WebFront Editor konnte nicht mehr geöffnet werden

## KONSOLE

Neu: Verbesserte Performance bei vielen Nachrichten in Meldungen oder im Debug Log
Neu: Google Maps wird erst geladen, wenn erforderlich
Neu: Fonts werden lokal geladen
Neu: Im Skript-Editor werden PHP-Schlüsselwörter als Vorschläge für die Autovervollständigung angeboten
Neu: Suchen/Ersetzen-Dialog sortiert die angezeigten Skripte
Neu: Dialog um in allen Skripten zu ersetzen fragt vorm Ersetzen, ob ungespeicherte Skripte gespeichert werden sollen
Neu: Mitteilungen können markiert und somit auch kopiert werden
Neu: Fortschrittsdialog beim Löschen von Objekten
Neu: Die Variablenaktionen "Addieren" und "Subtrahieren" verwenden für den Summanden/Subtrahenden nun die Nachkommastellen des Profils
Fix: Ladenachrichten beim Start der Konsole werden nun wie vorgesehen auf englisch übersetzt
Fix: Ladeicon ist wieder animiert ohne den Chrome einzufrieren
Fix: Ladeicon ist nun auch im Firefox durchgängig animiert

Fix: SVG Animation entfernt - Dies sollte das Einfrieren von Chrome verhindern
Fix: "Befehle testen"-Dialog gibt nun wie vorgesehen Fehlermeldungen aus
Fix: Typo im Adventskalender
Fix: Status Widget war seit der letzten Beta defekt
Fix: Schnelles Duplizieren von Konsolen (Tabs im Browser) konnte die Verbindung abbrechen lassen
Fix: Fonts wurden seit der letzten Beta nicht korrekt geladen
Fix: Speichern von Änderungen in HID und KNX Device (EIS) Instanzen funktioniert wieder
Fix: Prüfe Kompatibilität bei Upgrade vom Module Control zum Module Store
Fix: Sortiere Statusdialog wie vorgesehen
Fix: Speichere Login-Passwort wie vorgesehen
Fix: Modulbeschreibungen mit langen Wörtern werden umgebrochen und blenden keine Scrollbar mehr ein
Fix: Prüfe ob der zurückgegebene Link eines Buttons Zeilenumbrüche enthält und gebe ggf. passende Fehlermeldungen
Fix: Lade Icons für Icon-Auswahl iterativ, dies verhindert mögliche Verbindungsabbrüche beim Laden von sehr vielen Icons
Fix: Korrekte HEX-Darstellung im Debug-Log - Es wird nicht versucht gelegentlich Zeichen zusammenzuziehen, da diese als UTF-8 interpretiert werden
Fix: Module Control filtert die installierten SymconMisc-Module wieder wie vorgesehen
Fix: Wird in einem Konfigurator eine Instanz mit falscher ModulID geprüft kommt wie vorgesehen eine passende Meldung

# V5.1->V5.2 (Q3/2019)

## WESENTLICHE ÄNDERUNGEN

Voice over IP (VoIP)
Dynamische Instanzkonfiguration
Dokumentation direkt öffnen
Performanceverbesserung der Verwaltungskonsole

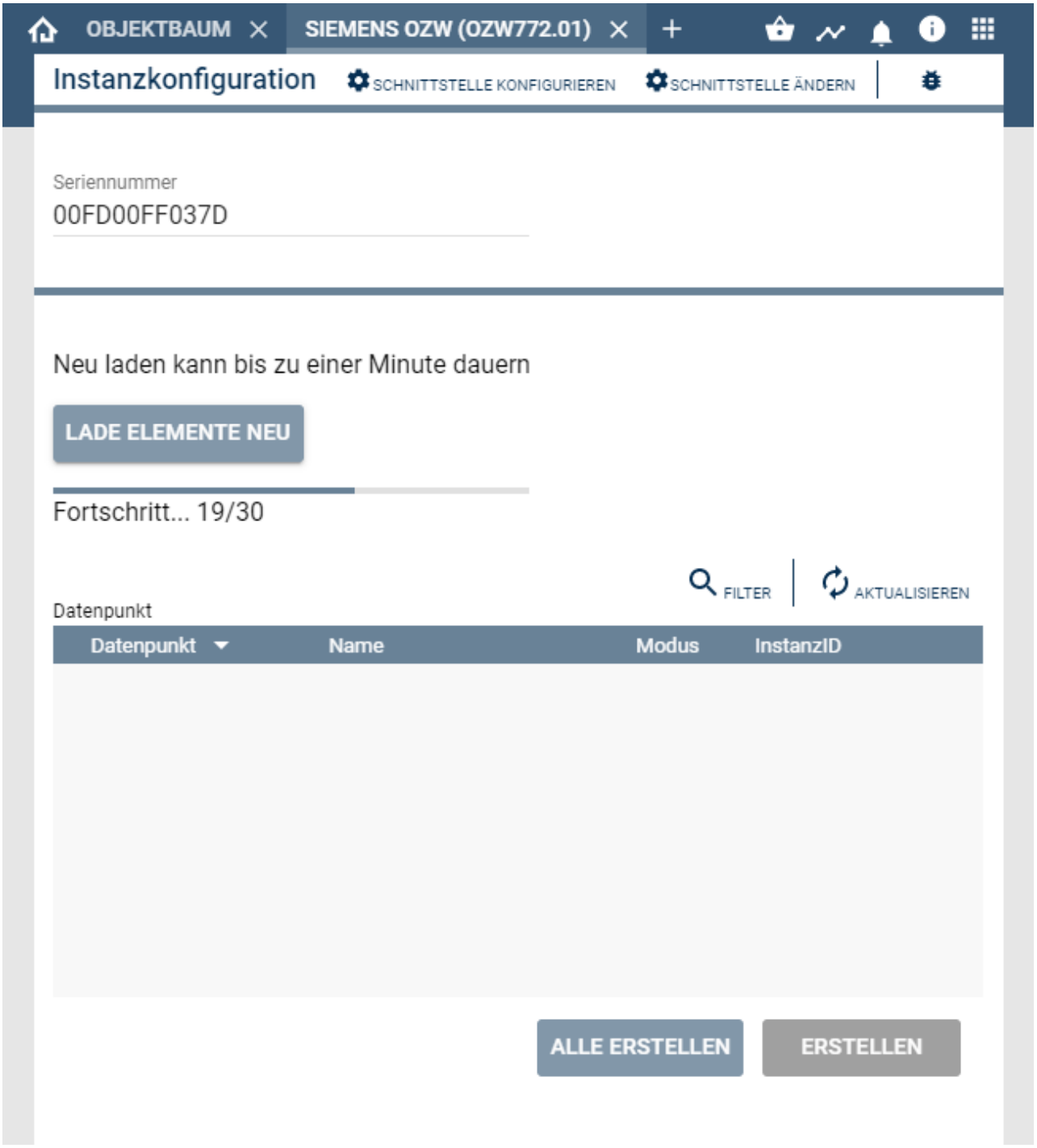
### VOICE OVER IP (VOIP)

Das VoIP Modul kann Telefonverbindungen aufbauen und entgegennehmen. Es können DTMF Signale verarbeitet und Sounds abgespielt werden. So ist es möglich sich beispielsweise in einem Alarmfall automatisch durch einen Telefonanruf informieren zu lassen. Bei einem Anruf können die Zahlentasten ausgewertet und so beispielsweise Geräte geschaltet werden.




## DYNAMISCHE INSTANZKONFIGURATION

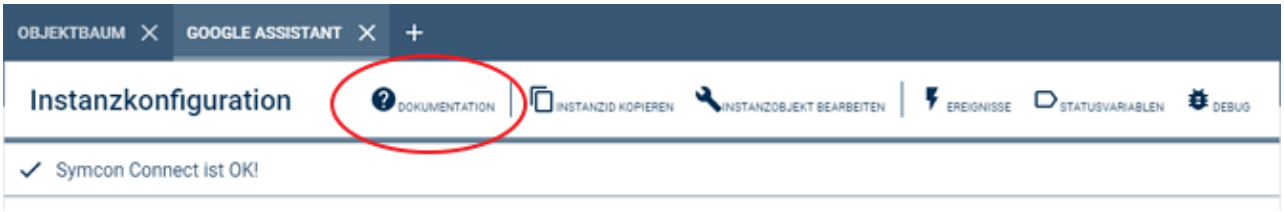
Die Konfigurationsseiten von Instanzen können sich nun dynamisch anpassen. So können unter anderem längere Ladezeiten eines Gerätes durch einen Ladebalken dargestellt oder individuell auf diverse Eingaben reagiert werden. Verwendung der Dynamik kann beispielsweise in den Z-Wave-Modulen, dem Notification Control und verschiedenen Konfiguratoren gesehen werden.




All diese Funktionen stehen selbstverständlich auch Modulentwicklern zur Verfügung. Zur dynamischen Anpassung wurden die Funktionen UpdateFormField und ReloadForm eingebaut und die möglichen Formularfelder um eine Vielzahl von Parametern erweitert.

## DOKUMENTATION DIREKT ÖFFNEN

Die Dokumentation eines Moduls kann nun direkt aus der Instanzkonfiguration geöffnet werden. Kommen also während der Konfiguration eines umfangreichen Moduls Fragen auf, so können diese mit einem einzelnen Klick auf den Dokumentations-Button beantwortet werden.




### PERFORMANCEVERBESSERUNG DER VERWALTUNGSKONSOLE

Die Performance der webbasierten Verwaltungskonsole wurde weiter verbessert. Durch die Verwendung von WebSockets werden Änderungen an IP-Symcon nun nahezu sofort in der Verwaltungskonsole wiedergespiegelt. Da verschachtelte Inhalte nun erst bei Bedarf geladen werden, ist die initiale Ladezeit der Verwaltungskonsole sowie das Öffnen einer Instanz beschleunigt.

## MIGRATION

Bitte erstellen Sie auf jeden Fall ein Backup Ihres gesamten IP-Symcon Ordners, um auf die Version 5.1 zurück zu wechseln, sollten wider erwarten Probleme mit der Version 5.2 auftreten.

Falls Sie von der Version 3.4 updaten, lesen Sie bitte vorher die Migrationsanleitung zur Version 4.0: Migration auf Version 4.0 , Version 4.1: Migration auf Version 4.1 , Version 4.2: Migration auf Version 4.2 , Version 4.3: Migration auf Version 4.3 , Version 4.4: Migration auf Version 4.4 , Version 5.0: Migration auf Version 5.0 und Version 5.1: Migration auf Version 5.1 .

Es wird Raspbian Buster für Raspberry Pi erfordert! (Generelle Versionenübersicht)
Es wird Ubuntu 18.04 für Linux 64 Bit erfordert! (Generelle Versionenübersicht)
Für Microsoft Windows steht ausschließlich eine 64-Bit Version zur Verfügung.

## MIGRATION (FÜR ENTWICKLER)

WebSocket Rückkanal von /api/wfc/{5-stellige-ID} zu /wfc/{5-stellige-ID}/api/ geändert
RTSP Proxy von /proxy/wfc/{5-stellige-ID} zu /wfc/{5-stellige-ID}/proxy/ geändert

## LISTE DER ÄNDERUNGEN

### ALLGEMEIN

Neu: PHP 7.3.8 (Löst insbesondere unter Windows 10 (1903) Abstürze beim Starten)
Neu: VoIP Modul (SIP)
Neu: EnOcean: Konfigurator für die Geräteübersicht
Neu: EnOcean: Geräte, die eine eindeutige, verfügbare Geräte-ID zum Anlernen brauchen,

können diese direkt per Knopfdruck bestimmen lassen. (Es wird automatisch die nächste freie ID gewählt)
Neu: Z-Wave: Kommandoklasse BARRIER_OPERATOR
Neu: Siemens OZW: Laden der verfügbaren Datenpunkte wird im Formular als Fortschrittsbalken angezeigt
Neu: Siemens SPS: Konfigurator für Siemens Logo 7 / Siemens Logo 8 VM-Adressen
Neu: LCN: Konfigurator lädt Module und Modulnamen mit einem schicken Fortschirttsbalken
Neu: LCN: Schwellwerte 4x4 können empfangen werden
Neu: LCN: Schwellwerte 1x5 und 4x4 können umgerechnet werden (wie bereits von den Variablen 1-12 gekannt)
Neu: xComfort: Neue Heizungsaktoren werden unterstützt
Neu: xComfort: Konfigurator unterstützt alle Kanäle vom 12-Kanal Heizungsaktor
Neu: WebSocket Support für WebHooks (Für Entwickler: WC_PushMessage zum Senden)
Neu: IRTrans Konfigurationsseite nutzt die neue Dynamik
Neu: Sys_GetNetworkInfo, Sys_GetBattery, Sys_GetProcessInfo sind jetzt überall verfügbar
Neu: Dynamik wird im Notification Control und WebFront Konfigurator genutzt
Fix: LCN: Konfigurator hat erstellte LCN Host-Kommando und Fingerprint Instanzen nicht korrekt erkannt

## KONSOLE

Neu: Performance Verbesserungen bei Instanzseiten. Insbesondere bei z.B. Alexa, HomeKit oder Z-Wave merkbar
Neu: Das Aktualisierungswidget prüft nun auf neue IP-Symcon-Versionen
Neu: Dauert das Laden der Parameter eines Musters zur Variablenbearbeitung (in Ereignissen oder "Befehle testen") länger, kommt eine Ladeanimation
Neu: Gebe eine schöne Meldung aus, wenn in einem Ereignis oder in "Befehle testen" eine Funktion mit Parameter vom Typ Variant oder Array gewählt wird
Neu: Schöner Beschreibungstext für Aktionen im Ereignisdialog von Instanzen
Neu: Bei Konfiguratoren kann 'position' und 'info' mit angegeben werden
Neu: Bei Konfiguratoren wird eine Warnmeldung angezeigt, wenn die 'location' an der falschen Stelle definiert wurde
Neu: Wochenplan und Ereignisdialog von Instanzen zeigt in den Aktionen eine schönere Beschreibung an, sofern möglich
Neu: Tabs der letzten Sitzung werden erst bei Bedarf geladen. Dies sollte die initiale Ladezeit, insbesondere bei vielen Tabs, deutlich reduzieren
Neu: Aktualisierungswidget wurde überarbeitet und prüft nun auch auf Aktualisierungen von IP-Symcon
Neu: Sind ein oder mehrere Typen im Objektbaum ausgeblendet, so wird der "Typen"-Button eingefärbt
Neu: Konsole kommuniziert über WebSockets
Neu: Ist ein Kontextmenü offen wird bei Betätigung von Escape das Menü und nicht zusätzlich der darunterliegende Tab/Dialog geschlossen
Neu: Enthält ein Modul im Module Store keine Discovery- oder Konfigurator-Instanzen, so biete die Erstellung von Kern-Instanzen an bzw. Geräte-Instanzen, falls auch keine Kern-Instanzen existieren
Neu: Dialog zum Stream erstellen wurde erweitert. Nun können Benutzername und Passwort in seperaten Feldern eingegeben werden, außerdem kann der Streamtyp (RTSP oder MJPEG) ausgewählt werden
Neu: Die Position eines Objektes kann im Objektbaum durch einen Doppelklick auf die Positionsspalte bearbeitet werden
Neu: Dialog zum Instanz erstellen enthält nun die Auswahl "Kein Hersteller" und

"(Sonstige)" wurde zu "(Gerät)" umbenannt
Neu: Verbesserte Performance bei vielen Einträgen pro Sekunde ins Meldungsfenster
Neu: Instanz Debug kann ins Logfile weitergeleitet werden
Neu: Im Instanz-Erstellen Dialog gibt es nun eine Auswahl für den Modultyp
Neu: Die Liste im Status-Widget erlaubt das Expandieren von langen Nachrichten
Neu: Module-Control schlägt Alexa als Standard-Modul vor
Neu: In Konfiguratoren können nun mehrere Instanzen zum Erstellen ausgewählt werden
Neu: Im RowLayout wird ein Zeilenumbruch gemacht, wenn zu viele Elemente enthalten sind
Fix: Alle erlaubten Dokumenttypen können nun wie vorgesehen im Dokument-Dialog ausgewählt werden
Fix: Lange Labels mit Links werden nun korrekt formatiert
Fix: Browsererkennung funktioniert nun auch bei iOS 13
Fix: Einige verwendete Bibliotheken wurden aktualisiert
Fix: Im Demo-Widget konnte "Abgelaufen" stehen, obwohl die Demo noch lange gültig ist
Fix: Konfiguratoren die keine Namensspalte haben funktionieren nun korrekt
Fix: Warte mit dem Springen zu neu erstellten Objekten bis diese korrekt einsortiert wurden
Fix: Aufräumen im Archiv erfolgt nun iterativ und nicht für alle Variablen parallel, was bei größeren Archiven zu Problemen führen konnte
Fix: Kann ein HTML-Link nicht direkt geöffnet werden, so öffne einen Dialog, der dies anbietet
Fix: Nachrichten vom Typ EM_CHANGESCHEDULEGROUPPOINT werden nun wie vorgesehen ausgewertet
Fix: Wird im Archiv eine Zeitspanne gelöscht, so erfolgt eine schöne Fehlermeldung, wenn die "Von"-Zeit nach der "Bis"-Zeit liegt
Fix: Slider in der Instanzkonfiguration als HorizontalSlider oder als Teil des Testcenters skalieren nun in Ihrer Schrittgröße, damit beispielsweise Variablen mit dem Profil ~Intensity.1 gut schaltbar sind
Fix: Es wird verhindert, dass durch die Verwendung der Browserhistorie Tabs zu entfernten Objekten geöffnet werden
Fix: Wird ein Skript von ausserhalb des Skripteditors ausgeführt, wird dies wie vorgesehen nicht mehr im Skripteditor angezeigt
Fix: Bei Konfiguratoren kann die 'location' korrekt auch bei Ketten verwendet werden
Fix: Ereignisse auf Statusvariablen werden wie vorgesehen erlaubt und nicht mit einer Nachricht kommentiert
Fix: Lange Beschreibungen von Parametern im Skripteditor können nicht mehr überlaufen
Fix: Das verwendete Framework Polymer wurde auf die neueste Version aktualisiert
Fix: Uhrzeiten für Aktionen im Wochenplan werden korrekt ausgewertet
Fix: Aktualisiere, die Aktionsauswahl, wenn wiederholt das gleiche Ziel für ein Ereignis gewählt wird, da sich die Schaltbarkeit von Variablen ändern könnte
Fix: Skins über das Aktualisierungswidget zurücksetzen funktioniert nun wie vorgesehen
Fix: Module aus dem Module Store können wie vorgesehen über das Aktualisierungswidget aktualisiert werden
Fix: Farbauswahl von Transparent aus funktioniert wieder wie vorgesehen in allen Browsern
Fix: Kompaktmodus hat den Switch nun links vom Text (wurde manchmal mit aktivierbaren Spezialschaltern verwechselt)
Fix: Dialoggröße von Auswahlbäumen passt sich dynamisch an die Anzahl dargestellter Objekte an
Fix: Fehlende Systemvariablen wurden der Autovervollständigung im Skripteditor hinzugefügt
Fix: Bei Modulen ohne Hersteller wird der Typ im Instanz-erstellen-Dialog wieder wie vorgesehen mit Klammern dargestellt
Fix: Die Größe von Spalten mit prozentualer Breite, z.B. im Module Control, kann wieder wie vorgesehen angepasst werden

Fix: Ausklappmenüs im Skripteditor werden wieder wie vorgesehen geschlossen, wenn daneben geklickt wird

## WEBFRONT

Neu: Graphen in einem Multi-Chart können aus-/eingeblendet werden
Neu: Medien vom Typ "Dokument" (z.B. PDFs) können direkt heruntergeladen werden
Neu: Editor kann nun auch korrekt auf Tablets verwendet werden
Fix: Sichtbarkeit der einzelnen Elemente konnte nicht verändert werden im Gegensatz zur Legacy Konsole
Fix: Im WebFront Konfigurator kann HD/Extrema nicht mehr als Standard aktiviert werden, solange Rohdaten aktiv ist
Fix: Beim Öffnen eines WebFront über den Connect Dienst welcher ein Kennwort benötigt gibt es keine unnötige 10 Sekunden Wartezeit mehr
Fix: WebFront: Schaltbefehle konnten beim Nutzen von Slidern doppelt versendet werden
Fix: WebFront: HD/Extrema wurden ggf. aktiviert, obwohl die Rohdaten Ansicht aktiv sein sollte
Fix: Sortierung wurde für neue Objekte nicht korrekt übernommen
Fix: Sortierung wurde für neue Kategorien nicht korrekt übernommen
Fix: Kategorie wurde nicht korrekt abgewählt wenn diese versteckt/gelöscht wurde

## PHP-SDK

Neu: Formular-Dynamik
Neu: In der module.json gibt es ein neues "url" Feld, welches einen String annimmt. Dies kann ein Link direkt zur Moduldokumentation sein und wird in Zukunft in der Instanzkonfiguration verfügbar sein, sofern angegeben.
Neu: In der library.json im Feld Versionen können nun beliebige Versionen (z.b. 1.0b) angegeben werden
Neu: Die IPS_GetLibrary Funktion liefert im Feld Version nun einen String anstatt einem Integer Wert

## SONSTIGES

Neu: Erweiterte Bedingungen unterstützen den Vergleich auf einen Wochentag
Neu: Im Location Control kann das Interval für Azimuth/Altitude eingestellt werden
Neu: Im Location Control können für "IstTag" auch die Offsets verwendet werden
Neu: EnOcean Gateway liest das Chipset aus (z.B. TCM310 oder TCM515)
Neu: Duplizieren von Objekten übernimmt jetzt Konfiguration der Status-Variablen
Neu: Web-Server kann nun z.B. aus dem /webfront/user/ Ordner mp4 Dateien korrekt in Safari abspielen
Neu: IPS_EnableDebugFile($id) aktiviert das Schreiben vom Debug-Logs in eine Datei (z.B. Linux: /var/log/symcon/debug_id.log)
Neu: DiscoveryServer versendet Connect-Adresse (sofern verfügbar), sodass die Apps direkt den Zugriff über Connect einrichten können *
Neu: Util Control erlaubt das automatische Konvertieren von PHP Short Tags (<?) in die normalen Tags (<?php)
Neu: Ereignisse welche per IPS_SetScriptTimer erstellt werden sind initial versteckt
Fix: WebHook Aufruf wird an den "besten" gefundenen Hook weitergeleitet
Fix: KNX Dicovery setzt auch die BindIP korrekt
Fix: Authentifizierung für RTSP Streams war unvollständig (benötigt Symcon Mobile 5.2)

Fix: Bei der Weiterleitung von z.B. /console auf /console/ wurden ggf. vorhandene Parameter verworfen
Fix: Es konnte ein bad_variant_access Fehler auftreten, sofern intern nicht alle Variablentypen unterstützt wurden
Fix: Modifizierte Dateirechte von Git Submodulen werden nicht mehr als Lokale Änderung betrachtet

## ÄNDERUNGEN 5.2.X

Neu: PHP 7.3.5
Neu: IPS_ConnectInstance prüft, ob die Verbindung sinn ergibt (z.B. Kern Instanzen haben keinen Datenfluss) und ob die Datenfluss GUIDs passen
Neu: Konsole: In Ereignissen gibt es ein neues Template zum Umschalten von Boolean Variablen
Neu: Konsole: Aktualisierungs-Widget berücksichtigt nun auch Skins und den Store
Neu: Konsole: Falsch konfigurierte Ereignisse werden im Baum markiert
Neu: WebFront: Per Parameter ?password=xxxx kann ein Kennwort-geschütztes WebFront direkt geöffnet werden
Neu: WebFront: "Nicht unterstützt" Meldung wird angezeigt, sofern iOS 9.x oder älter verwendet wird
Neu: WebFront: Reverse-Proxy mit Suffixen im Pfad sollte nun korrekt funktionieren
Fix: Docker: DBus und Avahi wurden nicht automatisch gestartet, wodurch das HomeKit Modul nicht genutzt werden konnte
Fix: KNX: Shutter wirft bei aktivierter Lamellensteuerung keine Fehlermeldung, wenn Stop mehrmals betätigt wird
Fix: KNX: Wenn im Shutter falsche EIS Typen empfangen wurden, wurde eine kryptische Fehlermeldung generiert. Die neue orientiert sich an der EIB Geräte Instanz
Fix: MQTT Server wirft bessere Fehlermeldung wenn die noch nicht unterstützten QoS > 0 Pakete empfangen werden
"Memory" Option nicht berücksichtigt wurde
Fix: Z-Wave: Control Klasse "SCENE_ACTIVATION" wird mit zu den unterstützten Klassen gewertet (Löst Probleme mit Scenen Activiation)
Fix: WebFront: CircleSlider und Farbrad können wieder genutzt werden, wenn vorher Werte gespeichert wurden
Fix: Konsole: Abstände der Elemente beim Test-Center verbessert
Fix: Konsole: Module Store überspringt beim Prüfen fehlerhafte alte Module
Fix: WebFront: IMAP/POP3 Postfächer werden wieder geladen
Fix: SymconMisc korrekt in "Misc Modules" beim Migrieren umbenannt
Fix: Skriptausgabe ist im Firefox wieder kopierbar
Fix: Module Store prüft auf Updates in der korrekten Sprache
Fix: Konsole: Einige Ereignisse konnte nicht bearbeitet werden
Fix: Konsole: Im Store kann nach Begriffen mit Leerzeichen gesucht werden
Fix: WebFront: Wenn ein ReverseProxy verwendet wird, versuchen wir den Pfad in der WebSocket Verbindung korrekt abzubilden
Fix: location kann nun wie vorgesehen auf für Ketten im Configurator-Element verwendet werden
Fix: Uhrzeiten für Aktionen im Wochenplan werden korrekt ausgewertet* Fix: Skins über das Aktualisierungswidget zurücksetzen funktioniert nun wie vorgesehen
Fix: Module aus dem Module Store können wie vorgesehen über das Aktualisierungswidget aktualisiert werden

Fix: dS Expert Instanz konnte nicht erstellt werden
Fix: WebFront: RTSP Streams werden zentriert und skaliert
Fix: WebFront: Skin Builder ignoriert alle @eaDir Verzeichnisse
Fix: Konsole: Sofern falsche Elemente genutzt werden, wird nun eine bessere Fehlermeldung geworfen
Fix: Konsole: Hintergrund bei "Befehl ausführen" Dialog wird korrekt vergrößert
Fix: Konsole: Kontextmenü auf Firefox unter MacOS funktioniert nun korrekt
Fix: Konsole: Wird ein Skript von ausserhalb des Skripteditors ausgeführt, wird dies wie vorgesehen nicht mehr im Skripteditor angezeigt
Fix: Abstürze unter Linux/Mac beim Aktualisieren von Modulen über das Module Control/ Store gelöst
Fix: Es wird geprüft, ob eine Datenfluss-Verbindung eine Endlosschleife ergibt. Sollte dies der Fall sein, wird die Verbindung nicht erlaubt.
Fix: PHP-SDK: HasActiveParent liefert false, wenn es keinen Parent gibt.
Fix: WebFront Konfigurator zeigt wieder korrekt die Geräte in der Geräteliste an
Fix: _generated.inc.php sollte beim Aktualisieren von Modulen nicht mehr kaputt gehen
Fix: Beim Entladen der Module werden die Funktionsverweise auch gelöscht, sodass diese nicht mehr doppelt und dreifach in "Befehle testen" auftauchen
Fix: Konsole: Revisionsnummer war 11 statt 12 Zeichen lang
Fix: Konsole: Farbauswahl von Transparent aus funktioniert wieder wie vorgesehen in allen Browsern
Fix: WebFront: Bessere Fehlermeldung, wenn ein falscher Platzhalter verwendet wird
Fix: "Gerät reaktivieren" ist für Z-Wave auch in der webbasierten Konsole verfügbar
Fix: xComfort Konfigurator kann wieder die neuen "Schalter" hinzufügen
Fix: Z-Wave: Absturz beim Empfang von defekter BATTERY_REPORT Pakete wird abgefangen
Fix: Legacy Konsole: Öffnen des neuen WebFront Editors ist jetzt auch über Connect Dienst möglich
Fix: Wenn ein Modul in den "modules" Ordner initial kopiert wurde, konnte es nicht per MC_ReloadModule geladen werden
Fix: Beim Duplizieren von vielen Objekten über die Verwaltungskonsole konnte IP-Symcon hängen bleiben
Fix: Unterkanäle von Z-Wave Geräten können in der webbasierten Konsole korrekt bearbeitet werden
Fix: M-Bus: "stol" Fehler beim Auswerten von M-Bus Zählern korrigiert
Fix: Legacy Konsole: Der neue Editor kann korrekt geöffnet werden, wenn der Connect Dienst genutzt wird
Fix: Konsole: Beim Hinzufügen und Bearbeiten von Gruppen im Wochenplan betreffen Änderungen der neuen Gruppe wie vorgesehen keine anderen Gruppen
Fix: Konsole: Meldungen zu Statusvariablen ohne Aktion bei Ereignissen angepasst
Fix: Konsole: Wird eine Variable, die nie geändert wurde, im Archiv geöffnet, dann wird eine entsprechende Meldung angezeigt
Fix: Konsole: Die Bearbeiten-Dialoge im Archiv können nun wie vorgesehen erneut geöffnet werden, nachdem geloggte Werte hinzugefügt wurden
Fix: Race Condition beim Timer Pool gelöst, welcher ggf. endlose "Warte auf Timer" Fehler auslösen konnte
Fix: Duplizieren beachtet jetzt auch deaktivierte Standardaktionen
Fix: Suchen/Einrichten von D2-* EnOcean Geräten wieder möglich
Fix: MQTT Server (Broker) hat Eingabe für Benutzername/Passwort
Fix: Z-Wave: Wenn direkt beim Senden ein Fehler aufgetreten ist, wurden die Retries nicht berücksichtigt

Fix: KNX Shutter: Wenn auf der GA 0/0/0 Pakete gesendet wurde hat der Shutter diese fälschlicherweise ausgewertet
Neu: Module Store informiert über Module, welche eine neuere IP-Symcon Version erfordern
Fix: Fehlermeldung beim UDP Socket, sofern der Host leer gelassen wurde
Fix: Weiteres Absturzproblem bei unvollständigen Z-Wave Battery Nachrichten gelöst
Fix: Z-Wave: Schalter für "Erzwinge Simple Assoziationen" fehlte in der Legacy Konsole
Fix: LCN Modul Konfigurator: Instanzen für Fingerprint/Host Kommando werden korrekt erkannt
Fix: WebFront: Schaltbefehle konnten beim Nutzen von Slidern doppelt versendet werden
Fix: WebFront: HD/Extrema wurden ggf. aktiviert, obwohl die Rohdaten Ansicht aktiv sein sollte
Fix: VoIP: Erlaube Anrufer Liste wurde nicht korrekt ausgewertet
Fix: VoIP: VoIP_Disconnect kann nun "klingelnde" (=nicht angenommene) Anrufe abbrechen
Fix: Z-Wave: Einige Befehle sollte nicht als gesendete Pakete zählen
Fix: Z-Wave: Für einige Befehle sollte es keine Retries geben (insbesondere NIF-Anfragen)
Fix: Konsole: Erweiterte Bedingungen können nicht mehr unbeabsichtigt gelöscht werden
Fix: Beim Beenden konnte IP-Symcon Abstürzen, sofern OPcache aktiviert war
Fix: Konsole erkennt Safari auf iOS 13 korrekt und zeigt keinen Warnhinweis mehr an
Fix: Spalten gehen nicht mehr kaputt, wenn man die Positionsspalte einblendet und dann neu Lädt (Dies ist der Webinar Bug!)
Fix: Neu angelegte Ereignisse konnten unter bestimmten Bedingungen nicht mehr geöffnet werden

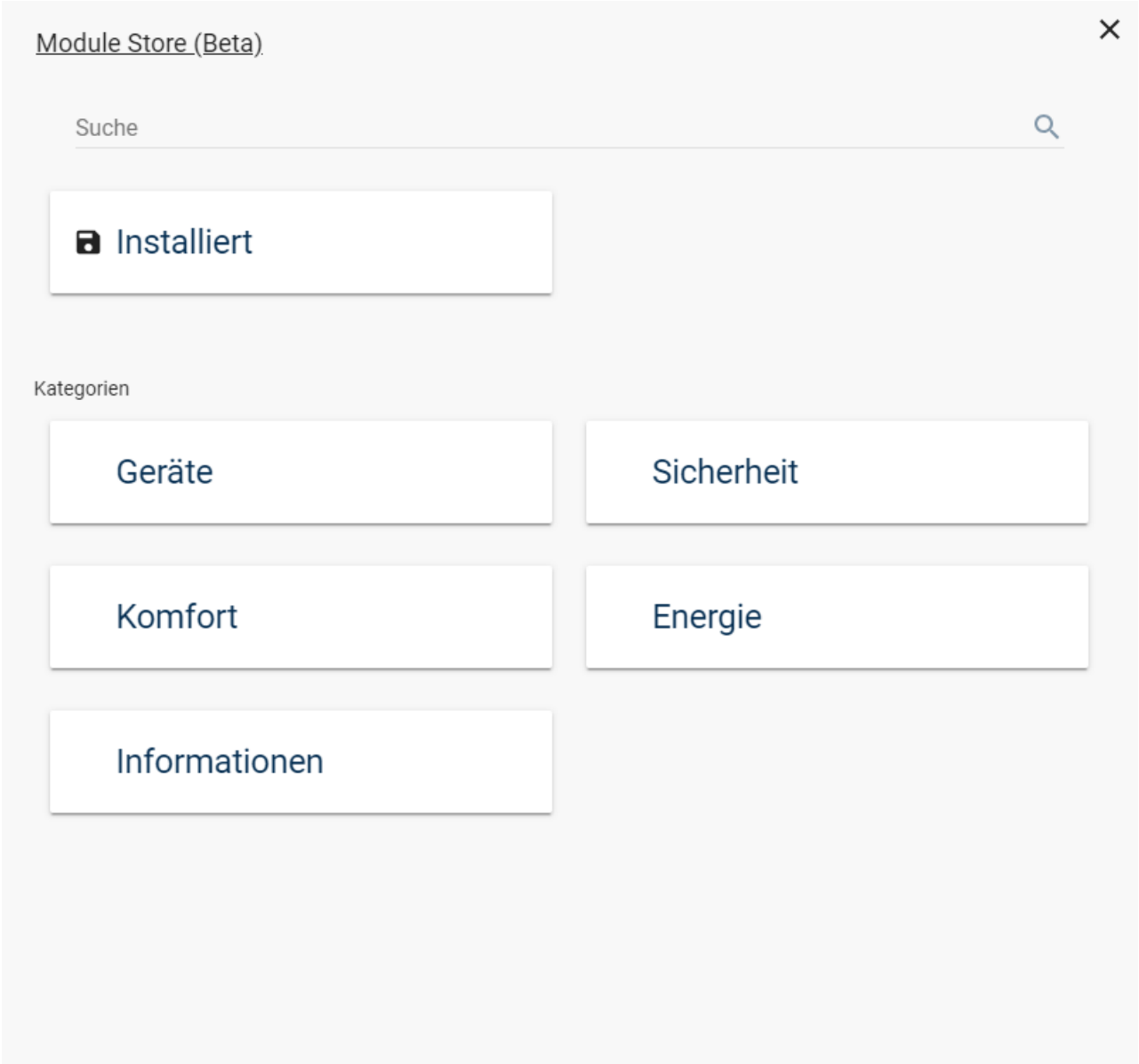
# V5.0->V5.1 (Q2/2019)

## WESENTLICHE ÄNDERUNGEN

Module Store (Beta)
RTSP-Streams
Z-Wave überarbeitet
MQTT (Beta)
Datenimport im Archiv
PHP 7.3
Performance der Konsole

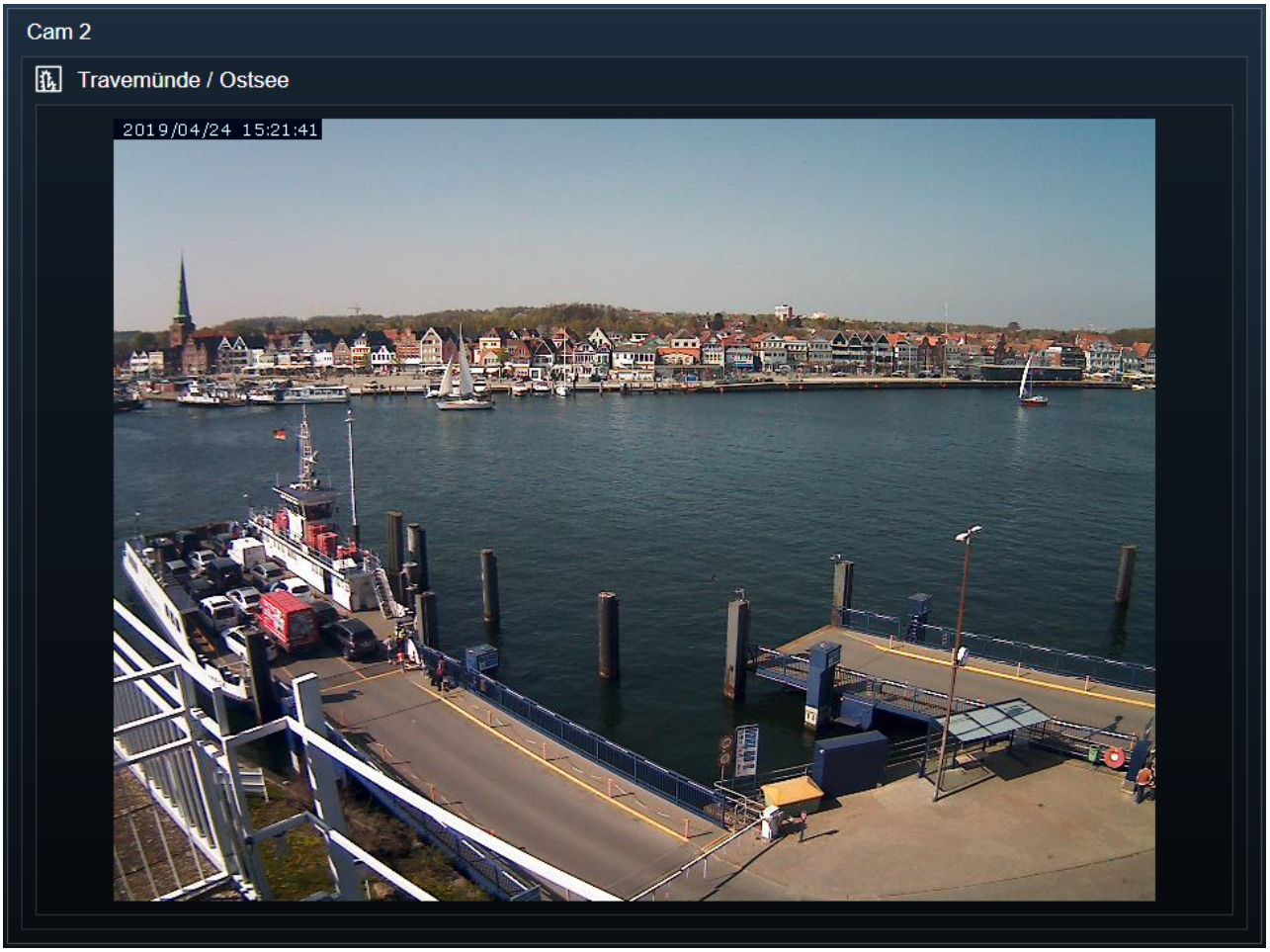
### MODULE STORE (BETA)

In der webbasierten Konsole kann nun auf den Module Store zugegriffen werden. Hier können angebotene Module von anderen Benutzern komfortabel durchsucht und installiert werden. Aktuell ist der Module Store als Beta verfügbar und wird im Laufe der kommenden Versionen noch um diverse spannende Features erweitert.




## RTSP STREAMS

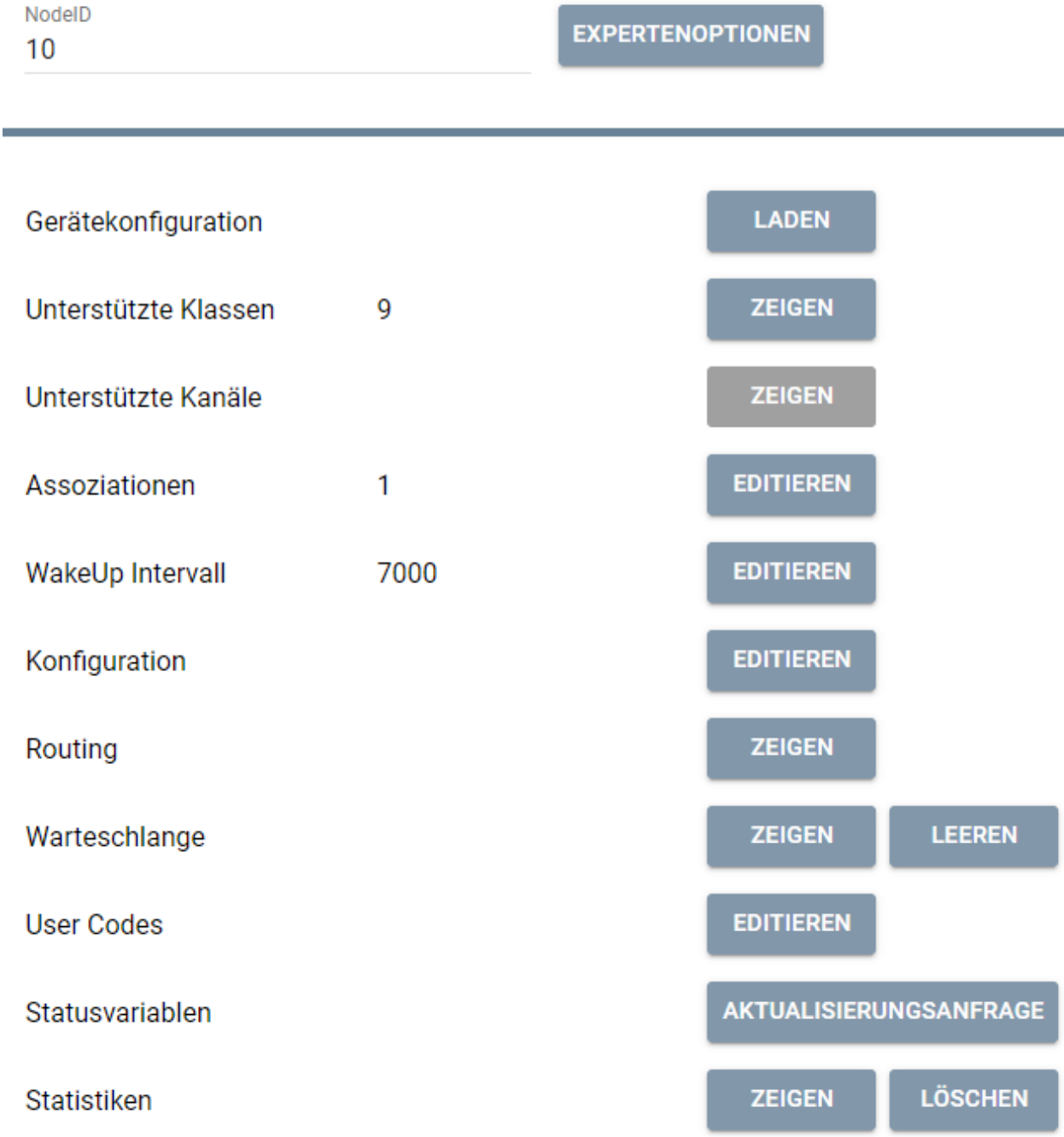
Über RTSP Streams können nahezu alle handelsüblichen Kameras in IP-Symcon eingebunden werden. Da IP-Symcon hier als Verteiler agiert, können auch bei Kameras, die eigentlich nur einen Empfänger zur Zeit unterstützen, mehrere Benutzer gleichzeitig auf den Stream zugreifen.




Aktuell funktionieren die RTSP Streams im Chrome, Firefox, Opera und Android (inkl. Apps ab Version 5.1). Zusätzlich müssen RTSP Streams aktuell mit h264 kodiert werden. Die Anzahl von RTSP Streams, die gleichzeitig angeboten werden können, hängen von der verwendeten Version ab, siehe Funktionsübersicht .

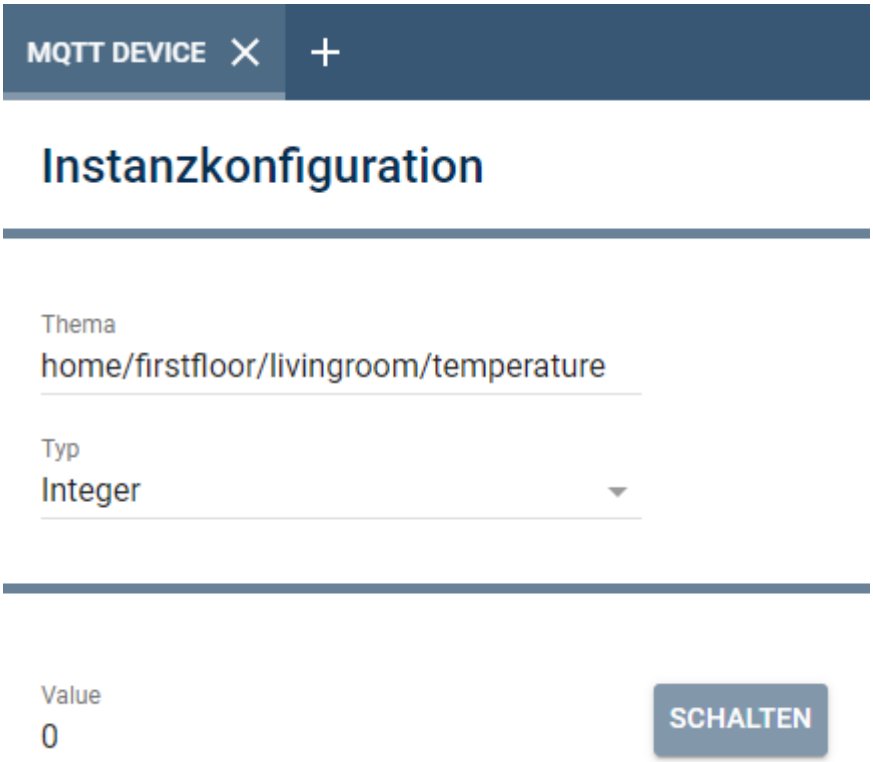
## Z-WAVE ÜBERARBEITET

Die Unterstützung von Z-Wave wurde komplett überarbeitet. Beim Laden der Gerätekonfiguration werden nun alle Daten des Gerätes eingelesen und stehen direkt zur Verfügung. Zusätzlich kann nun eine noch größere Vielfalt von Z-Wave-Geräten eingebunden werden. Mehr Details zu den unterstützten Geräten finden Sie hier .



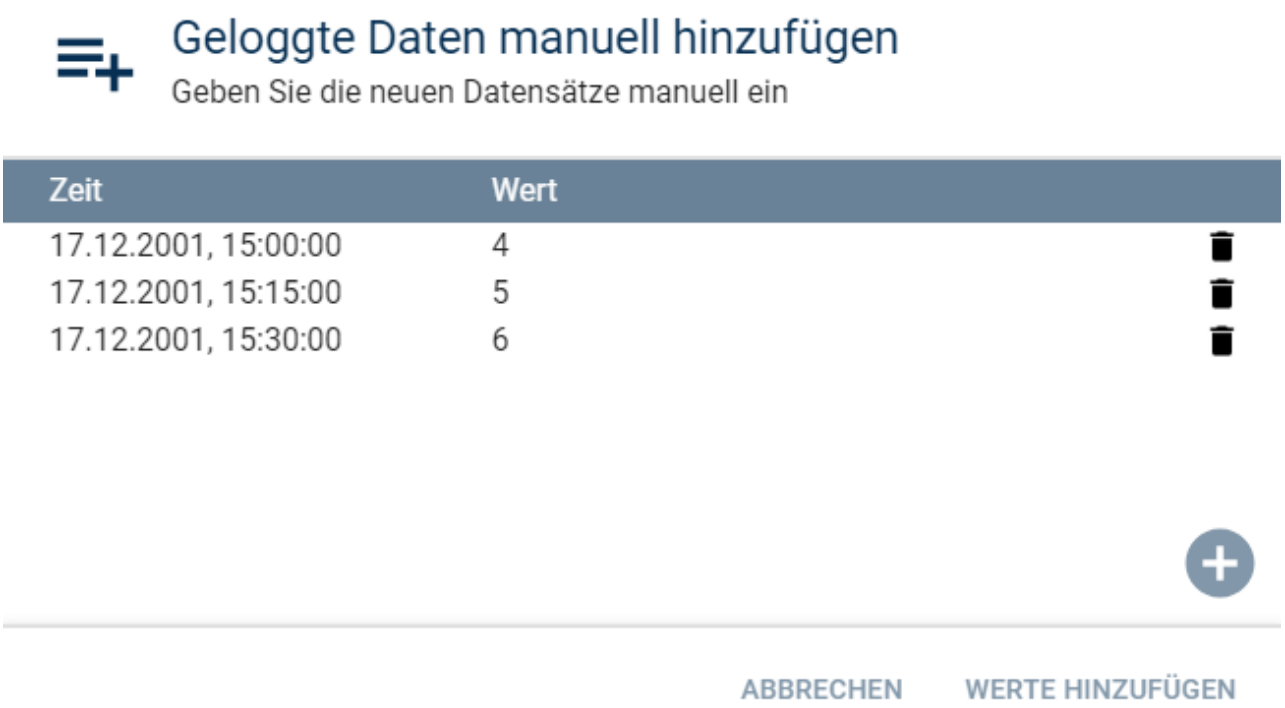

## MQTT (BETA)

MQTT wird nun von IP-Symcon nativ unterstützt. Mit Version 5.1 startet die Open Beta der MQTT Anbindung, welche die grundlegenden Funktionen von MQTT anbietet und damit die meisten MQTT-Geräte unterstützt. Die vollständige Anbindung von MQTT kommt mit Version 5.2.




## DATENIMPORT IM ARCHIV

Es ist nun möglich, die archivierten Werte einer Variablen im Nachhinein zu erweitern. So können beispielsweise Verbräuche eingetragen werden, welche ermittelt wurden bevor sie als Variable in IP-Symcon verwaltet wurden oder Daten nachgetragen werden, falls die Verbindung zwischen Sensor und IP-Symcon kurzzeitig gestört wurde. Es ist sowohl möglich die neuen Daten als Datei einzulesen als auch die Werte direkt in IP-Symcon einzutragen.



### PHP 7.3

Die verwendete PHP-Version wurde von 7.2 auf 7.3 aktualisiert. Dies bietet verschiedene neue Funktionen. Details zur neuen Version können hier gefunden werden.

### PERFORMANCE DER KONSOLE

Die Performance vom Objektbaum und vergleichbaren Elementen wurde deutlich verbessert. Nun können auch Kategorien mit mehr als 1000 Kindern in weniger als einer Sekunde ausgeklappt werden und auch das wiederholte automatisierte Löschen und Erstellen von Objekten stört nicht beim flüssigen Arbeiten.

## MIGRATION

Bitte erstellen Sie auf jeden Fall ein Backup Ihres gesamten IP-Symcon Ordners, um auf die Version 5.0 zurück zu wechseln, sollten wider erwarten Probleme mit der Version 5.1 auftreten.

Falls Sie von der Version 3.4 updaten, lesen Sie bitte vorher die Migrationsanleitung zur Version 4.0: Migration auf Version 4.0 , Version 4.1: Migration auf Version 4.1 , Version 4.2: Migration auf Version 4.2 , Version 4.3: Migration auf Version 4.3 , Version 4.4: Migration auf Version 4.4 und Version 5.0: Migration auf Version 5.0 .

Es wird Raspbian Stretch für Raspberry Pi erfordert! (Generelle Versionenübersicht)
Es wird Ubuntu 18.04 für Linux 64 Bit erfordert! (Generelle Versionenübersicht)
Für Microsoft Windows steht ausschließlich eine 64-Bit Version zur Verfügung.
Upgrade von PHP 7.2 auf PHP 7.3. (Upgrade auf 7.3 )

## MIGRATION (FÜR PHP-MODUL ENTWICKLER)

Das WebFront benötigt mindestens iOS 10.x aufgrund der Migration auf Dart 2.
Mit Version 5.1 werden die neuen Funktionen GetStatus und HasActiveParent angeboten.
Hier kann es bei gleichnamigen Funktionen zu Konflikten kommen.

## LISTE DER ÄNDERUNGEN

### ALLGEMEIN

Neu: PHP 7.3.0
Neu: MQTT Server (ehemals Broker genannt) (Beta)
Neu: RTSP Streams können hinzugefügt werden
Neu: Z-Wave komplett überarbeitet

Neu: M-Bus: Es werden viele neue Geräte (insbesondere mit weiteren Telegrammen) unterstützt
Neu: Archiv kann Daten importieren
Neu: HomeMatic Konfigurator liest die Gerätenamen / Räume per HomeMatic Script aus (Bitte Firewall in der CCU anpassen!)
Neu: HomeMatic unterstützt Authentifizierung der CCU3
Neu: HomeMatic unterstützt Verschlüsselung der CCU3
Neu: KNX DPT 249.*
Neu: WebServer unterstützt WebSockets
Neu: digitalStrom (Expert Modul)
Neu: Presence Control
Neu: Connect Dienst Backend von libssh2 auf libssh gewechselt (Mehr Performance, Löst Probleme mit DS-Lite Anschlüssen)
Neu: Connect Dienst überprüft periodisch, ob das Eigenheim wirklich über ipmagic.de verfügbar ist.
Neu: Wir kompilieren fast alle Abhängigkeiten direkt in IP-Symcon rein, damit diese von der Versionen für alle IP-Symcon Betriebssysteme gleich sind. Damit sollte übrigens die Installation auf andere Linux-Derivate einfacher werden (auch wenn wir es nicht offiziell unterstützen)
Neu: Event Control verbindet fehlerhafte Instanzen nicht mehr jede Minute sondern mit Abstand (2^Fehlversuche Minuten). Mindestens jedoch einmal am Tag.
Neu: Profil für pH Werte
Neu: Status Code für Instanzen geändert: 201 wurde auf 105 geändert, da der 200+ Raum für benutzerdefinierte Fehler reserviert war
Neu: .well-known Ordner wird in den beschreibbaren Bereich gemappt (IPS_GetKernelDir())
Neu: Undokumentierte Funktion IPS_GetOptionList
Neu: IPS_GetReferenceList für alle Instanzen
Neu: Spezialschalter für Basis Authentifizierung des WebFront User Ordners, welche auch für den 3777 Server (inkl. Connect) gelten.
Neu: Skripte haben neue Felder: ScriptUpdated, ScriptCRC, ScriptSize (Neue Nachricht SM_UPDATE)
Neu: Neue Spezialschalter: ServerMaxPostSize, MaxLoginAttemptsBeforeSlowdown, MaxLoginAttemptsBeforeLockdown, MaxLoginAttemptsSlowdownDuration, MaxLoginAttemptsLockdownDuration, MaxLoginAttemptsSlowdownWaitTime (Doku)
Neu: Interne Übersetzungen sind nun aufgeteilt in einzelne Dateien im "locales" Ordner
Neu: LCN Module-Konfigurator legt Instanzen in einer passenden Kategorie an
Neu: Systemvariablen $_IPS['OLDUPDATED'] / $_IPS['OLDCHANGED'] für Ereignisse hinzugefügt
Fix: KNX DPT 13.002 korrigiert (verhält sich jetzt wie die ETS)
Fix: KNX Shutter: Stop konnte nicht immer ausgeführt werden
Fix: KNX DPT 242: Bad Variant Access korrigiert
Fix: KNX wirft keine Fehler mehr, wenn "leere" cEMI Pakete empfangen werden
Fix: DPT_WriteDPT17 lässt IP-Symcon nicht mehr abstürzen
Fix: Leerzeichen um eingefügte Adressen im Module und Skin Control können keine Fehler mehr verursachen
Fix: Multicast Modul zeigt direkt nach Erstellen den Wert für 0.0.0.0 zur Auswahl
Fix: INF/NaN wird bei SetValue* mit Fehlermeldung bemängelt
Fix: Probleme mit ALL2x Modul korrigiert
Fix: EIB/KNX-Formulare in der webbasierten Konsole speichern zusätzliche Adressen nun korrekt als Zahlen ab
Fix: Legacy Konsole zeigt im Location Control korrekt die Offset-Namen an

Fix: Wenn der Editor aktiviert war, konnte Objekte nicht geschaltet werden
Fix: Alte XML-Settings Relikte von IP-Symcon 2.5 und älter werden entfernt
Fix: HomeMatic KeepAlive differenziert je nach Rückkanal und kann somit Abbrüche für jeden Kanal einzeln erkennen
Fix: HomeMatic Geräte Instanzen speichern nun einige Meta-Daten, um die Aktion einer Variable nach dem Neustart korrekt widerherzustellen
Fix: Siemens OZW hat bei Datenpunkten mit Schreibmöglichkeit diese nicht korrekt eingerichtet
Fix: LCN Gateway ist nicht mehr im Fehlerzustand, sofern der I/O geschlossen ist
Fix: GUIDs werden bei Funktionsaufrufen auf Gültigkeit geprüft
Fix: GUIDs von KNX RGB/KNX RGBW korrigiert (Es fehlten die {})
Fix: LibraryID/ModuleID werden automatisch nach "UpperCase" konvertiert. (Wird mit einer netten Warning bestätigt)
Fix: Noch nicht unterstützte EnOcean Paket Typen werfen keine Fehler mehr, sondern werden im Debug Log protokolliert
Fix: xComfort: Der Anzeige der Firmware fehlen führende Nullen

## KONSOLE

Neu: Der Module Store (Beta)
Neu: Umbau des Objektbaums und anderen Listen-/Baumstrukturen für deutlich bessere Performance
Neu: Im Archive Control können die Einstellungen zu einer Variablen alternativ durch einen Doppelklick geöffnet werden
Neu: Abbrechbarer Fortschrittsdialog beim Erstellen von allen Instanzen in einem Konfigurator oder beim Reaggregieren aller Variablen
Neu: Sicheres Löschen (Prüfen von Referenzen vorm Löschen) ist nun auswählbar
Neu: Variablen mit ungültiger Aggregation werden entsprechend im Archiv markiert
Neu: Lokale Sprachen wie de_AU werden nun bei der Übersetzung von Modulen beachtet und primär angewandt, falls vorhanden
Neu: Bessere Fehlermeldungen, wenn es beim Verbinden Probleme gibt, insbesondere bei deaktiviertem Fernzugriff
Neu: Von der Instanzkonfiguration und dem Skripteditor aus kann das entsprechende Objekt bearbeitet werden
Neu: Stelle Anzahl an gelöschten Datensätzen nach Löschen einer Zeitspanne im Archiv dar
Neu: Initialisiere den "In allen Skripten suchen"-Dialog bei Betätigung von Strg+Shift+F mit dem aktuell ausgewählten Text
Neu: Beim Sprung zum Skript aus dem "In allen Skripten suchen"-Dialog wird der gesuchte Text ausgewählt
Neu: Mitteilung beim Abschluss des Aufräumens im Archiv
Neu: Spaltengrößen sind nun auch in der webbasierten Konsole änderbar.
Neu: "Tab hinzufügen" Button ist jetzt immer sichtbar - auch bei vielen Elementen
Neu: Zeige SkriptID in PHP Information an
Neu: Wird im Wochenplan eine Aktion für ein nicht schaltbares Objekt hinzugefügt oder eine mit leerem Skript geöffnet, wird automatisch PHP ausgewählt
Neu: Wochenpläne wurden umgebaut, so dass beim Hovern die Uhrzeit angezeigt wird und das Verhalten bei Klicks dem WebFront entspricht
Neu: Ladebildschirme wurden an das neue Homepagedesign angepasst. Es wird nun auch vor dem Login-Dialog ein Ladebildschirm gezeigt
Neu: Der aktuell ausgewählte Tab wird hervorgehoben

Neu: Füllen die Tabs die Leiste, so werden sie dünner skaliert. Die Pfeile zum Scrollen erscheinen erst, wenn die Tabs sonst eine gewisse Mindestbreite unterschreiten würden
Neu: Autovervollständigung im Skripteditor unterstützt nun auch Standard-PHP-Funktionen
Neu: Enthält eine Variable über ihr Profil Assoziationen, hat aber einen Wert unter allen Assoziationswerten, wird nun als Wert '-' anstatt eines Fehlers angezeigt
Neu: Links werden (wieder) standardmäßig ohne Namen erstellt, da diese nun in allen Anwendungen mit dem Namen des Ziels dargestellt werden
Neu: Mittels Shift + Pfeiltasten können mehrere Knoten in Bäumen ausgewählt werden
Neu: Neues Konfigurationselement "TestCenter" zum Schalten aller Statusvariablen
Neu: Icon editieren Dialog zeigt Fortschritt beim Laden der Icons und Cached diese.
Neu: Skripteditor zeigt nach dem Ausführen die Laufzeit an
Neu: Beim Öffnen des Module Controls wird geprüft, ob installierte Bibliotheken durch Module aus dem Module Store ersetzt werden können
Neu: Im Demo-Widget wird nun beim Text zwischen Demos mit und ohne Lizenz differenziert
Fix: Konsole: Im Prüfen-Dialog des Konfiguratorelements wird nicht mehr "undefined" angezeigt
Fix: Der Tab öffnen Dialog und der Archiv Operations Dialog können per Escape geschlossen werden
Fix: Fehler in Suchfeldern (beispielsweise ID nicht vorhanden) werden korrekt dargestellt
Fix: Die Darstellungsoption im Debug Log wurde umgestaltet und sollte die Funktionalität nun besser verdeutlichen
Fix: Variablen im Archiv werden nun nach Position gruppiert
Fix: Prüfe ob aktuelle Filter ein Sprungziel bei "Zu Referenz springen" im Referenzdialog oder "Zum Quellobjekt springen" bei Links verbergen und frage, ob diese deaktiviert werden sollen
Fix: In Zwischenablage kopieren kann auch mit iOS benutzt werden
Fix: Titelzeile der Instanzkonfiguration auf kleinen Displays optimiert
Fix: In ausgelösten Ereignissen Instanzaktionen laden funktioniert nun auch mit Strings und wechselt auf PHP Code, sofern die Aktion nicht geparst werden kann
Fix: Sollte eine Demoversion geöffnet werden, so erscheint das Demowidget nun als letztes, nicht als erstes, Widget
Fix: Gesperrte Variablen über dem Variablenlimits werden korrekt markiert
Fix: Benutzernamen mit E-Mails, welche einen Suffix von mehr als 4 Zeichen haben, werden nun korrekt unterstützt

## WEBFRONT

Neu: WebFront auf Dart 2 migriert (Entfernt Unterstützung von iOS 9.x)
Neu: WebFront fügt die IDs im DOM hinzu, sodass Skins gezielter greifen können
Neu: WebFront nutzt den neuen WebSocket Rückkanal
Neu: WebFront mit deaktiviertem Editor lädt bei fehlenden Objekten nicht mehr neu, sondern lädt diese passend nach
Neu: WebFront WYSIWYG Editor kann die in der Basic Version vorhandenen Elemente bearbeiten (wie zur Zeit in der Legacy Konsole möglich)
Neu: WebFront WYSIWYG Editor kann per Knopfdruck sicher aus der Verwaltungskonsole geöffnet werden
Neu: UnixTimestamp* Profile können im Info-Widget verwendet werden
Neu: Button im Konfigurator um WebFronts neu zu laden wurde entfernt. Stattdessen werden die WebFronts bei Änderungen automatisch neu geladen

Neu: Timeout Parameter für Logout Widget zum Automatischen Logout nach Inaktivität
Fix: UTF-8 kodierte Zeichen konnten in der WebFront Legende nicht angezeigt werden
Fix: Wenn RAW bei Graphen aktiviert ist, werden die Zeiten korrekt im Titel angezeigt
Fix: WebFront kann nun String-CSV Daten anzeigen
Fix: WFC_OpenCategory funktioniert wieder korrekt
Fix: Wenn das Laden einer Medien Datei fehlschlägt wird die Ladeanimation korrekt versteckt
Fix: Der Slider springt auf die alte Position zurück, wenn das Setzten des neuen Wertes fehlschlägt
Fix: Slider mit % korrigiert, wenn Min > 0 war (Fix für die Apps kommt)
Fix: WebFront Skin-Builder ignoriert .DS_Store Dateien
Fix: WebFront Konfigurator nutzt bei Skins den Namen und nicht mehr den Titel
Fix: Scrollen funktioniert bei vielen Konfiguratoren wieder korrekt
Fix: Die linken Achsen bei Multigraphen werden genauso sortiert wie die Legende
Fix: Achsen von Bool Graphen innerhalb von Line Charts werden dargestellt
Fix: WebFront: Bei UnixTimestampTime konnte exakt 01:00 nicht als Wert angezeigt/ausgewählt werden

## PHP-SDK

Neu: Attribute, welche direkt ohne ApplyChanges gespeichert werden
Neu: Konstanten sind nun definiert
Neu: Funktionen zum Registrieren der benutzten/referenzierten IDs
Neu: Parameter "link" für das Konfigurationselement Button ermöglicht die Angabe, ob die Ausgabe als Link interpretiert werden soll oder nicht
Neu: Das Formularelement "Image" kann durch das Flag "center" nun zentriert werden
Neu: GetStatus
Neu: HasActiveParent

## ÄNDERUNGEN 5.1.X

Neu: PHP 7.3.5
Neu: IPS_ConnectInstance prüft, ob die Verbindung sinn ergibt (z.B. Kern Instanzen haben keinen Datenfluss) und ob die Datenfluss GUIDs passen
Neu: Konsole: In Ereignissen gibt es ein neues Template zum Umschalten von Boolean Variablen
Neu: Konsole: Aktualisierungs-Widget berücksichtigt nun auch Skins und den Store
Neu: Konsole: Falsch konfigurierte Ereignisse werden im Baum markiert
Neu: WebFront: Per Parameter ?password=xxxx kann ein Kennwort-geschütztes WebFront direkt geöffnet werden
Neu: WebFront: "Nicht unterstützt" Meldung wird angezeigt, sofern iOS 9.x oder älter verwendet wird
Neu: WebFront: Reverse-Proxy mit Suffixen im Pfad sollte nun korrekt funktionieren
Neu: Module Store informiert über Module, welche eine neuere IP-Symcon Version erfordern
Fix: DBus und Avahi wurden nicht automatisch gestartet, wodurch das HomeKit Modul nicht genutzt werden konnte
Fix: KNX: Shutter wirft bei aktivierter Lamellensteuerung keine Fehlermeldung, wenn Stop mehrmals betätigt wird
Fix: KNX: Wenn im Shutter falsche EIS Typen empfangen wurden, wurde eine kryptische Fehlermeldung generiert. Die neue orientiert sich an der EIB Geräte Instanz
Fix: MQTT Server wirft bessere Fehlermeldung wenn die noch nicht unterstützten QoS > 0

Pakete empfangen werden
"Memory" Option nicht berücksichtigt wurde
Fix: Z-Wave: Control Klasse "SCENE_ACTIVATION" wird mit zu den unterstützten Klassen gewertet (Löst Probleme mit Scenen Activiation)
Fix: WebFront: CircleSlider und Farbrad können wieder genutzt werden, wenn vorher Werte gespeichert wurden
Fix: Konsole: Abstände der Elemente beim Test-Center verbessert
Fix: Konsole: Module Store überspringt beim Prüfen fehlerhafte alte Module
Fix: WebFront: IMAP/POP3 Postfächer werden wieder geladen
Fix: SymconMisc korrekt in "Misc Modules" beim Migrieren umbenannt
Fix: Skriptausgabe ist im Firefox wieder kopierbar
Fix: Module Store prüft auf Updates in der korrekten Sprache
Fix: Konsole: Einige Ereignisse konnte nicht bearbeitet werden
Fix: Konsole: Im Store kann nach Begriffen mit Leerzeichen gesucht werden
Fix: WebFront: Wenn ein ReverseProxy verwendet wird, versuchen wir den Pfad in der WebSocket Verbindung korrekt abzubilden
Fix: location kann nun wie vorgesehen auf für Ketten im Configurator-Element verwendet werden
Fix: Uhrzeiten für Aktionen im Wochenplan werden korrekt ausgewertet* Fix: Skins über das Aktualisierungswidget zurücksetzen funktioniert nun wie vorgesehen
Fix: Module aus dem Module Store können wie vorgesehen über das Aktualisierungswidget aktualisiert werden
Fix: dS Expert Instanz konnte nicht erstellt werden
Fix: WebFront: RTSP Streams werden zentriert und skaliert
Fix: WebFront: Skin Builder ignoriert alle @eaDir Verzeichnisse
Fix: Konsole: Sofern falsche Elemente genutzt werden, wird nun eine bessere Fehlermeldung geworfen
Fix: Konsole: Hintergrund bei "Befehl ausführen" Dialog wird korrekt vergrößert
Fix: Konsole: Kontextmenü auf Firefox unter MacOS funktioniert nun korrekt
Fix: Konsole: Wird ein Skript von ausserhalb des Skripteditors ausgeführt, wird dies wie vorgesehen nicht mehr im Skripteditor angezeigt
Fix: Abstürze unter Linux/Mac beim Aktualisieren von Modulen über das Module Control/ Store gelöst
Fix: Es wird geprüft, ob eine Datenfluss-Verbindung eine Endlosschleife ergibt. Sollte dies der Fall sein, wird die Verbindung nicht erlaubt.
Fix: PHP-SDK: HasActiveParent liefert false, wenn es keinen Parent gibt.
Fix: WebFront Konfigurator zeigt wieder korrekt die Geräte in der Geräteliste an
Fix: _generated.inc.php sollte beim Aktualisieren von Modulen nicht mehr kaputt gehen
Fix: Beim Entladen der Module werden die Funktionsverweise auch gelöscht, sodass diese nicht mehr doppelt und dreifach in "Befehle testen" auftauchen
Fix: Konsole: Revisionsnummer war 11 statt 12 Zeichen lang
Fix: Konsole: Farbauswahl von Transparent aus funktioniert wieder wie vorgesehen in allen Browsern
Fix: WebFront: Bessere Fehlermeldung, wenn ein falscher Platzhalter verwendet wird
Fix: "Gerät reaktivieren" ist für Z-Wave auch in der webbasierten Konsole verfügbar
Fix: xComfort Konfigurator kann wieder die neuen "Schalter" hinzufügen
Fix: Z-Wave: Absturz beim Empfang von defekter BATTERY_REPORT Pakete wird abgefangen
Fix: Legacy Konsole: Öffnen des neuen WebFront Editors ist jetzt auch über Connect Dienst möglich
Fix: Wenn ein Modul in den "modules" Ordner initial kopiert wurde, konnte es nicht per MC_ReloadModule geladen werden
Fix: Beim Duplizieren von vielen Objekten über die Verwaltungskonsole konnte IP-Symcon hängen bleiben
Fix: Unterkanäle von Z-Wave Geräten können in der webbasierten Konsole korrekt

bearbeitet werden
Fix: M-Bus: "stol" Fehler beim Auswerten von M-Bus Zählern korrigiert
Fix: Legacy Konsole: Der neue Editor kann korrekt geöffnet werden, wenn der Connect Dienst genutzt wird
Fix: Konsole: Beim Hinzufügen und Bearbeiten von Gruppen im Wochenplan betreffen Änderungen der neuen Gruppe wie vorgesehen keine anderen Gruppen
Fix: Konsole: Meldungen zu Statusvariablen ohne Aktion bei Ereignissen angepasst
Fix: Konsole: Wird eine Variable, die nie geändert wurde, im Archiv geöffnet, dann wird eine entsprechende Meldung angezeigt
Fix: Konsole: Die Bearbeiten-Dialoge im Archiv können nun wie vorgesehen erneut geöffnet werden, nachdem geloggte Werte hinzugefügt wurden
Fix: Race Condition beim Timer Pool gelöst, welcher ggf. endlose "Warte auf Timer" Fehler auslösen konnte
Fix: Duplizieren beachtet jetzt auch deaktivierte Standardaktionen
Fix: Suchen/Einrichten von D2-* EnOcean Geräten wieder möglich
Fix: MQTT Server (Broker) hat Eingabe für Benutzername/Passwort
Fix: Z-Wave: Wenn direkt beim Senden ein Fehler aufgetreten ist, wurden die Retries nicht berücksichtigt
Fix: KNX Shutter: Wenn auf der GA 0/0/0 Pakete gesendet wurde hat der Shutter diese fälschlicherweise ausgewertet
Fix: Fehlermeldung beim UDP Socket, sofern der Host leer gelassen wurde
Fix: Weiteres Absturzproblem bei unvollständigen Z-Wave Battery Nachrichten gelöst
Fix: Z-Wave: Schalter für "Erzwinge Simple Assoziationen" fehlte in der Legacy Konsole
Fix: LCN Modul Konfigurator: Instanzen für Fingerprint/Host Kommando werden korrekt erkannt
Fix: WebFront: Schaltbefehle konnten beim Nutzen von Slidern doppelt versendet werden
Fix: WebFront: HD/Extrema wurden ggf. aktiviert, obwohl die Rohdaten Ansicht aktiv sein sollte

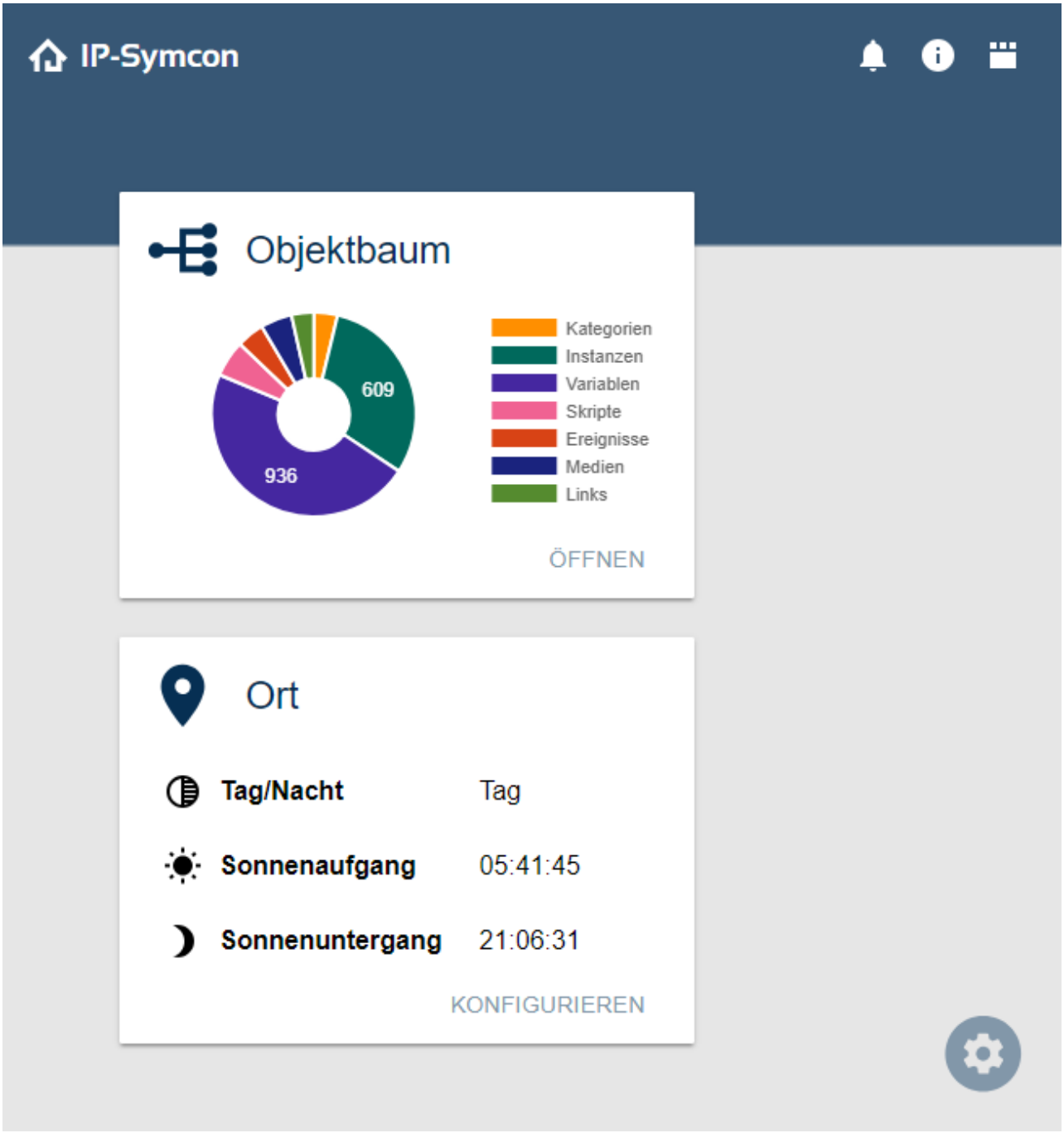
# V4.4->V5.0 (Q3/2018)

## WESENTLICHE ÄNDERUNGEN

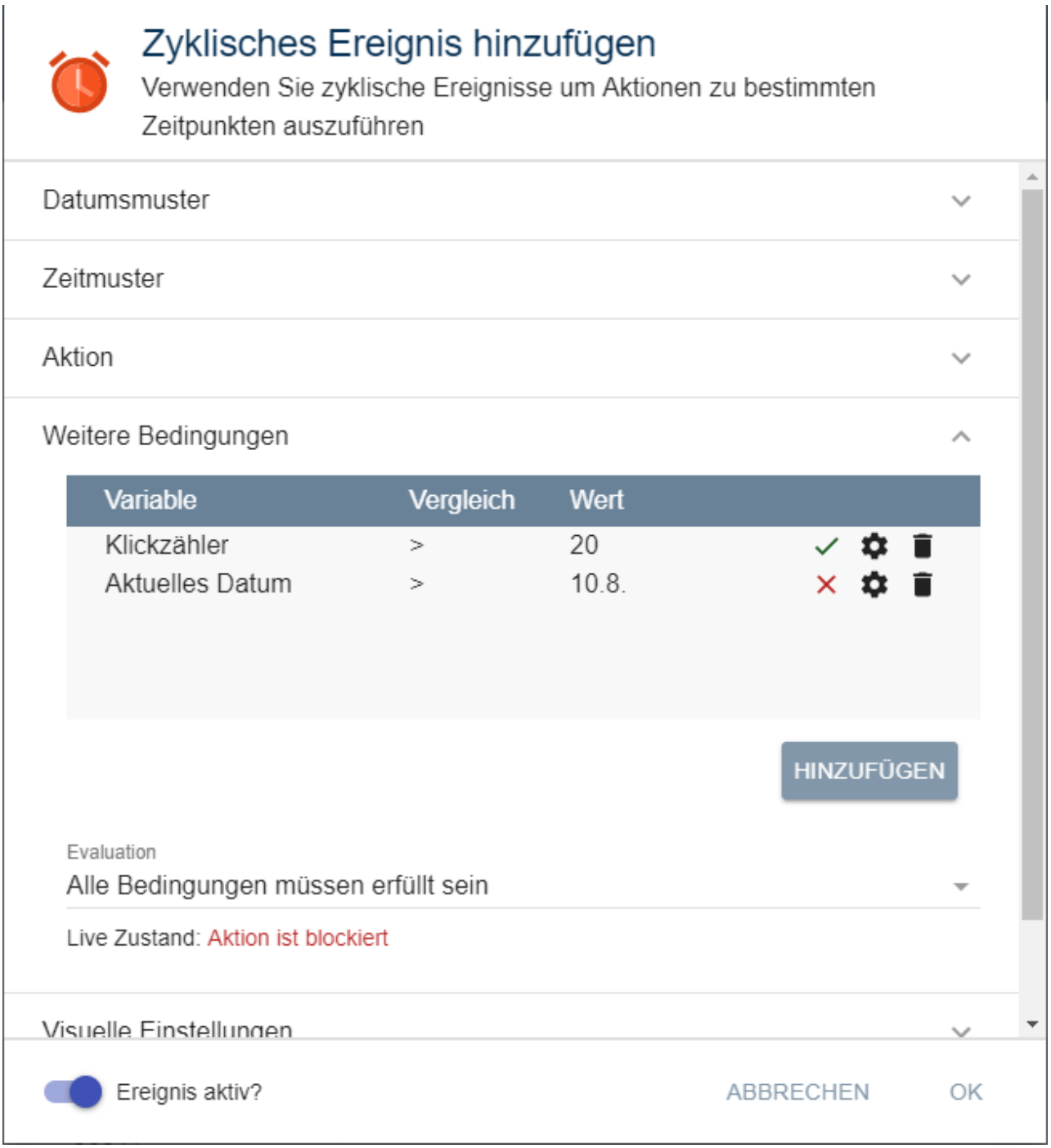

Webbasierte Verwaltungskonsole
Amazon Alexa und Google Home
WebFront: Neuer Light Skin
PHP 7.2
KNX: Erweiterung für alle DPTs
Hardwareunterstützung für ekey Fingerprint Lösungen
Docker Support
Verbesserung der mobilen Apps
Erweiterung vom Location Control
Neue Funktionen im WebFront


### WEBBASIERTE VERWALTUNGSKONSOLE

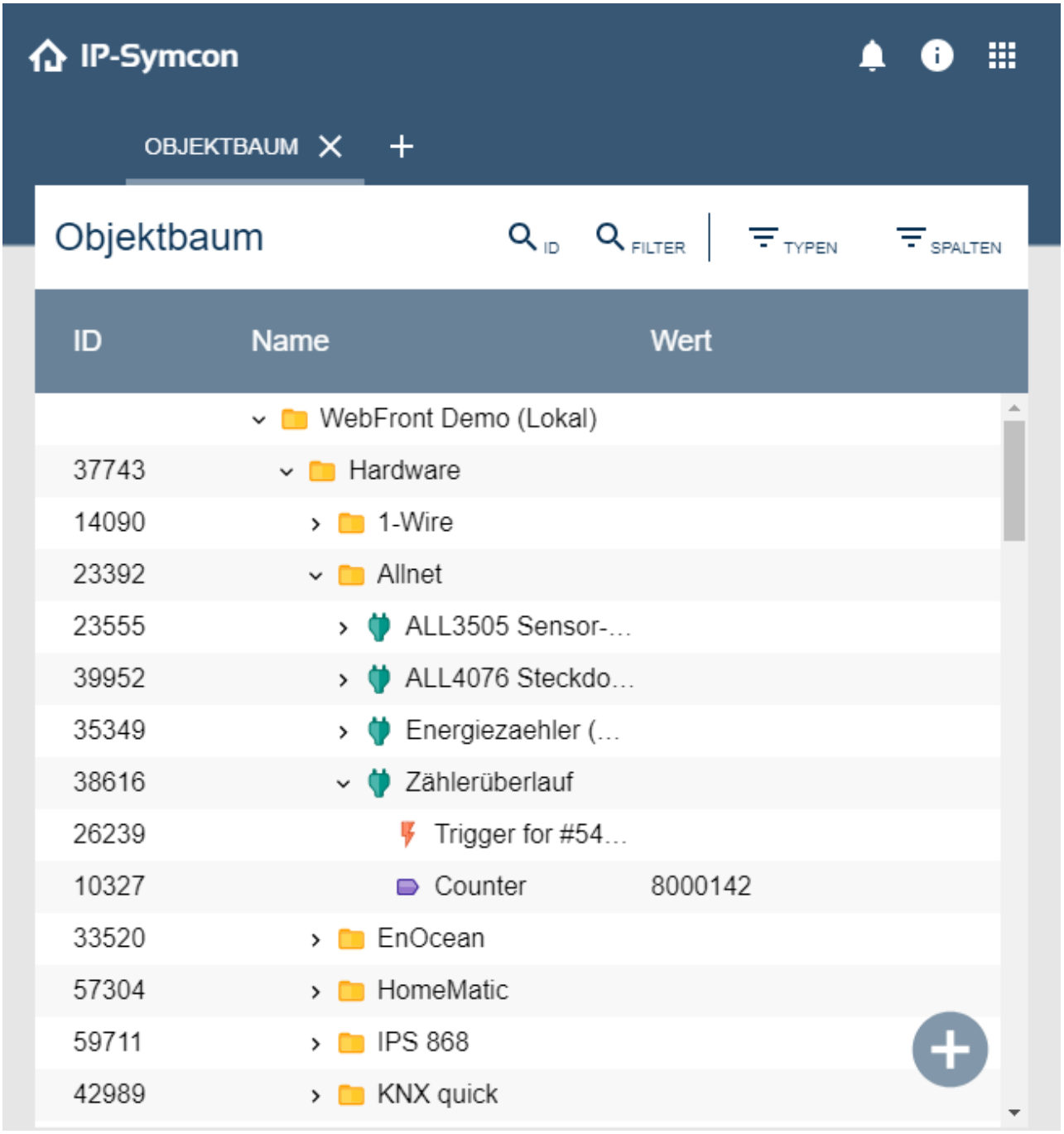
Die webbasierte Verwaltungskonsole ersetzt zu IP-Symcon 5.0 standardmäßig die alte Verwaltungskonsole. Sie bietet diverse Vorzüge und macht die Konfiguration von IP-Symcon um einiges komfortabler. Unterstützt werden die Browser Chrome, Opera, Safari, Firefox und Edge.
So bietet die neue Startseite der Verwaltungskonsole diverse individuell einstellbare Widgets, welche die verschiedenen Aspekte des Systems darstellen können.




Die Dialoge zur Erstellung von Objekten wurden komplett überarbeitet und bieten nun alle Einstellungen mit ausklappbaren Reitern auf einer Seite. So ist alles an einem Ort zu finden und bleibt dennoch übersichtlich.

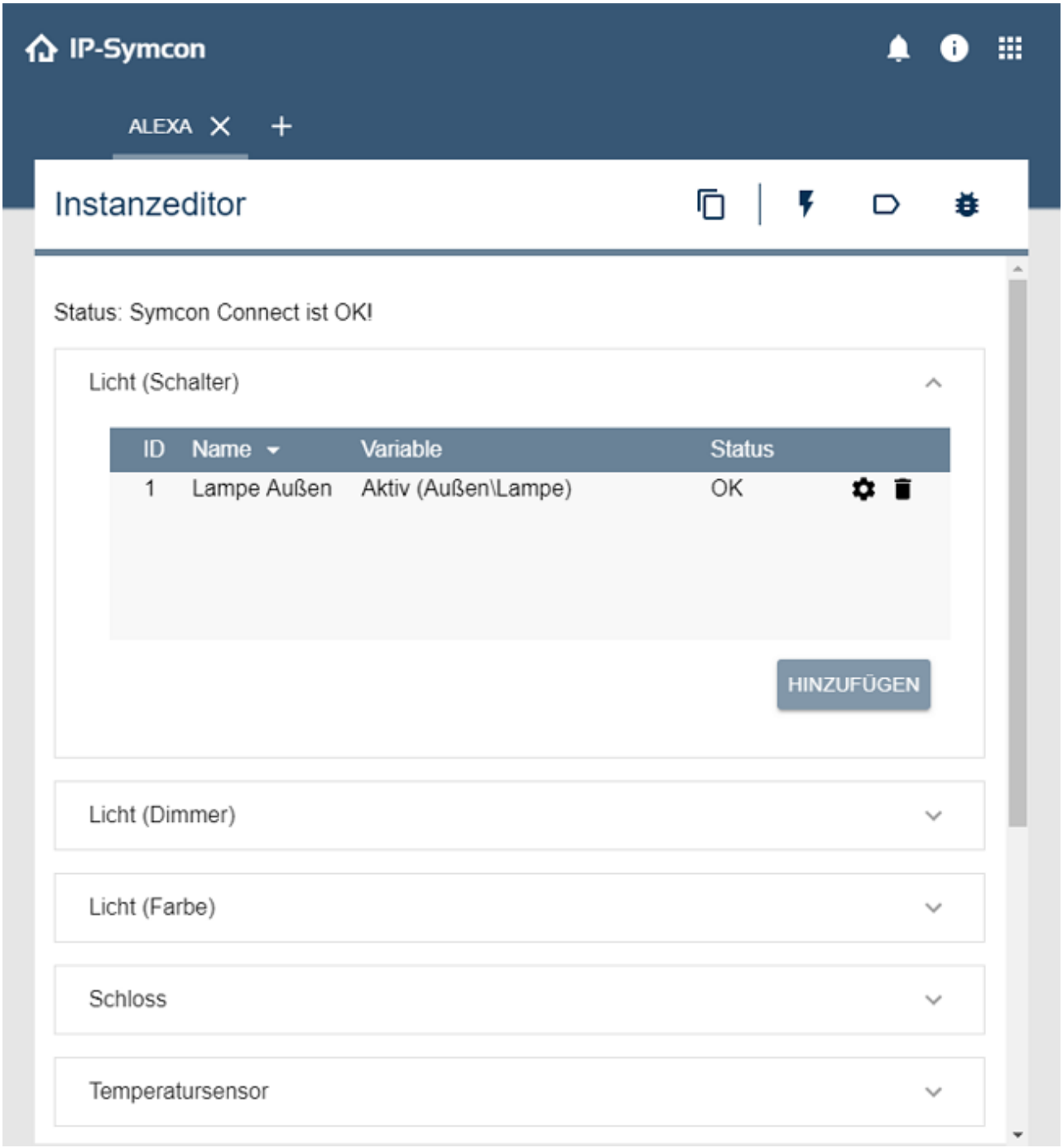



Darüber hinaus ist die Verwaltungskonsole betriebssystemunabhängig. So kann die Konfiguration von IP-Symcon auf jedem üblichen Betriebssystem vorgenommen werden. Zusätzlich muss die Konsole nicht mehr installiert oder heruntergeladen werden, denn sie ist immer Bestandteil von der IP-Symcon Installation. Über Symcon Connect ist die Verwaltungskonsole ebenfalls erreichbar, sodass ein Zugriff jederzeit und von jedem Ort leicht möglich ist.



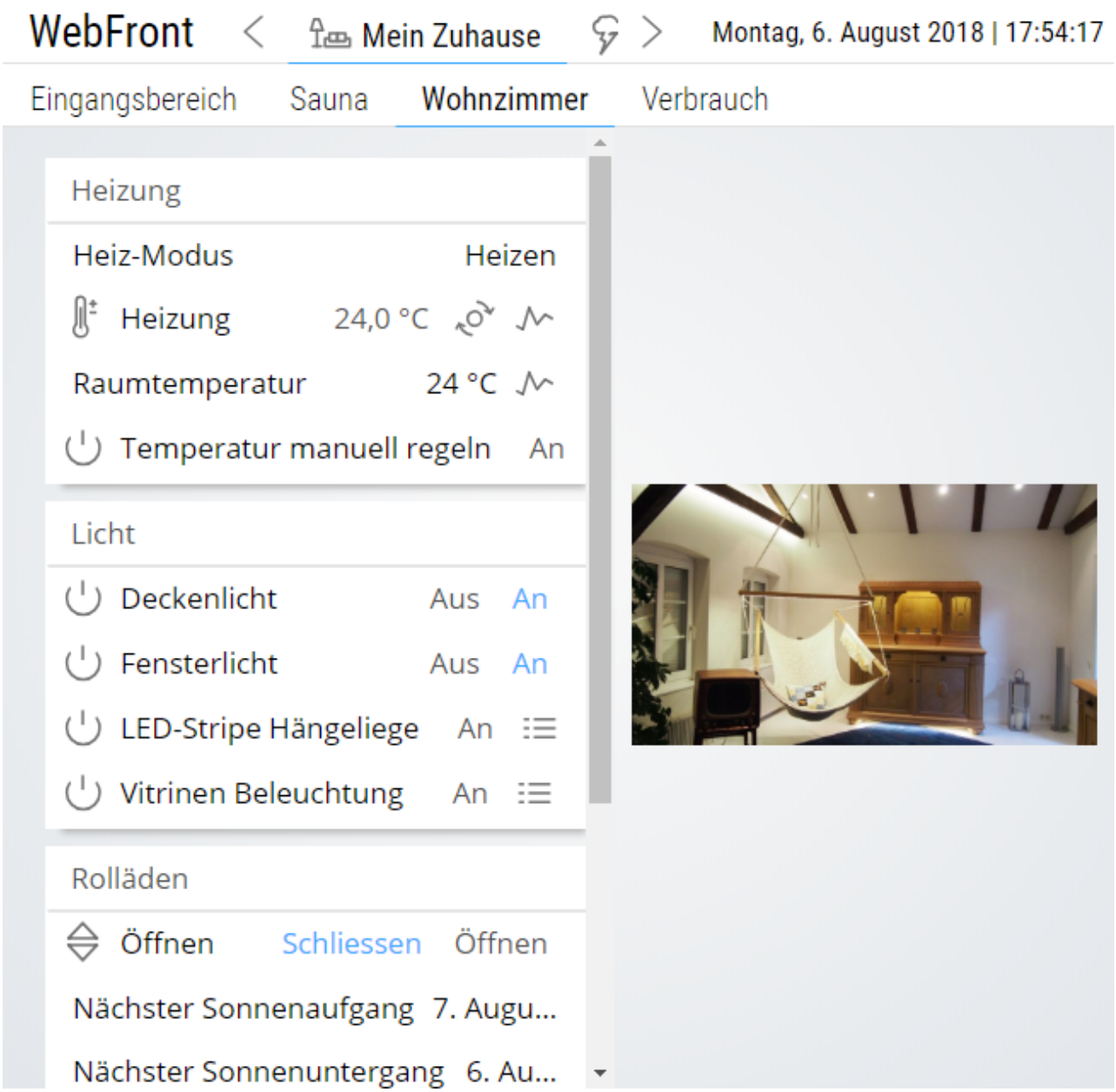

## AMAZON ALEXA UND GOOGLE HOME

Mit IP-Symcon 5.0 kommen zwei neue Module, welche die Sprachsteuerung via Amazon Alexa oder Google Assistant ermöglichen. Wir möchten uns hier noch einmal bei unserem Communitymitglied Dave bedanken, welcher mit uns das Alexa-Modul IQL4SmartHome entwickelt hat. Das neue Alexa-Modul verwendet die Alexa Skill API in der aktuellen Version 3, welche einige neue Features ermöglicht, wie beispielsweise die Szenensteuerung.




## WEBFRONT: NEUER LIGHT SKIN

Ein neuer moderner Skin findet Einzug. Neben dem “Standard” und “DarkSkin” bieten wir zusätzlich den “LightSkin” an. Dieser ist in seinem Aufbau und Optik bewusst modern und hell gehalten um neusten Designs zu entsprechen und stellt ein farbiges Komplement zu den vorhanden Skins dar. Ihr könnt den Light Skin ab sofort auf unserer WebFront Demo ausprobieren: IP-Symcon WebFront. Zur Installation könnt ihr euch die Beispiele im Skin Control ansehen.




### PHP 7.2

Die verwendete PHP-Version wurde von 5.6 auf 7.2 aktualisiert. Dies bietet verschiedene neue Funktionen und einen spürbaren Performancegewinn. Details können hier gefunden werden.

### KNX: ERWEITERUNG FÜR ALLE DPTS

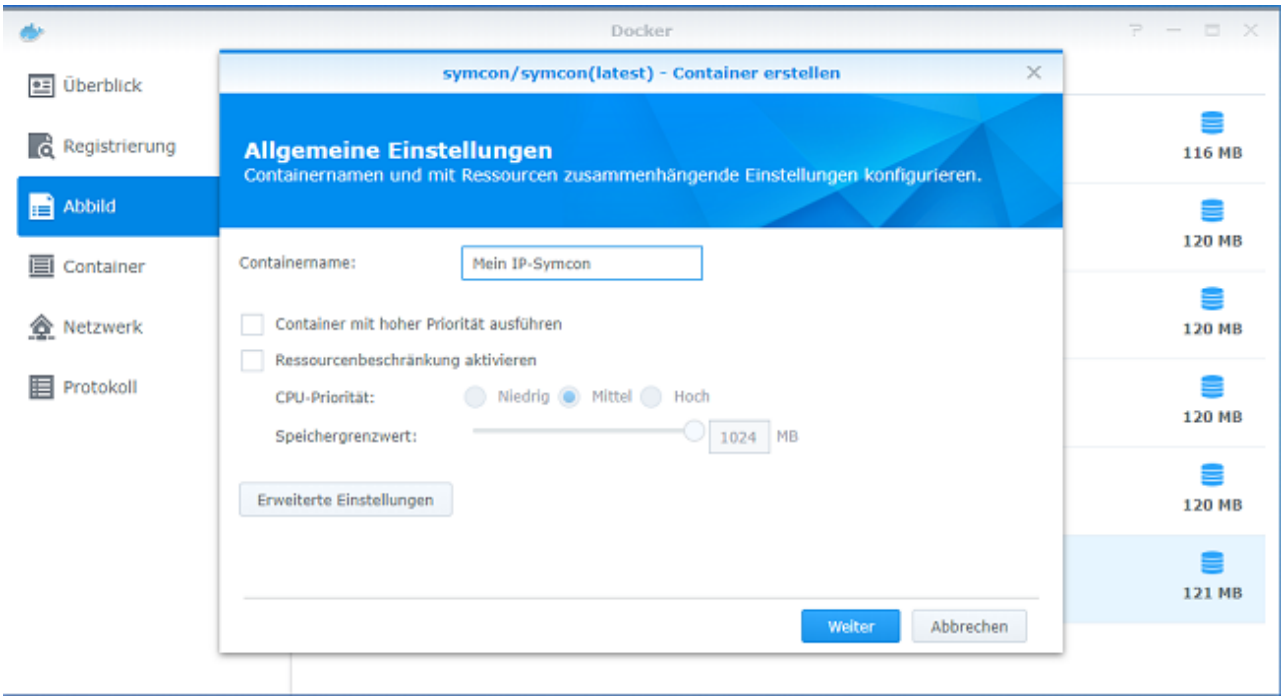
Zusätzlich zu den jeher unterstützten 15 EIS Klassen wird mit der Version 5.0 eine Unterstützung von über 370 Datentypen in 77 DPT Klassen integriert. Dadurch wird die Integration von allen KNX Geräten ermöglicht.

### HARDWAREUNTERSTÜTZUNG FÜR EKEY FINGERPRINT LÖSUNGEN

Es wurden neue Module entwickelt, welche nativ die ekey home/multi/net Zutrittslösungen einbinden. Die eingebundenen Geräte stellen die gerätespezifischen Werte in IP-Symcon zur Verfügung. Dazu gehören z.B. die bekannten Fingerscanner.

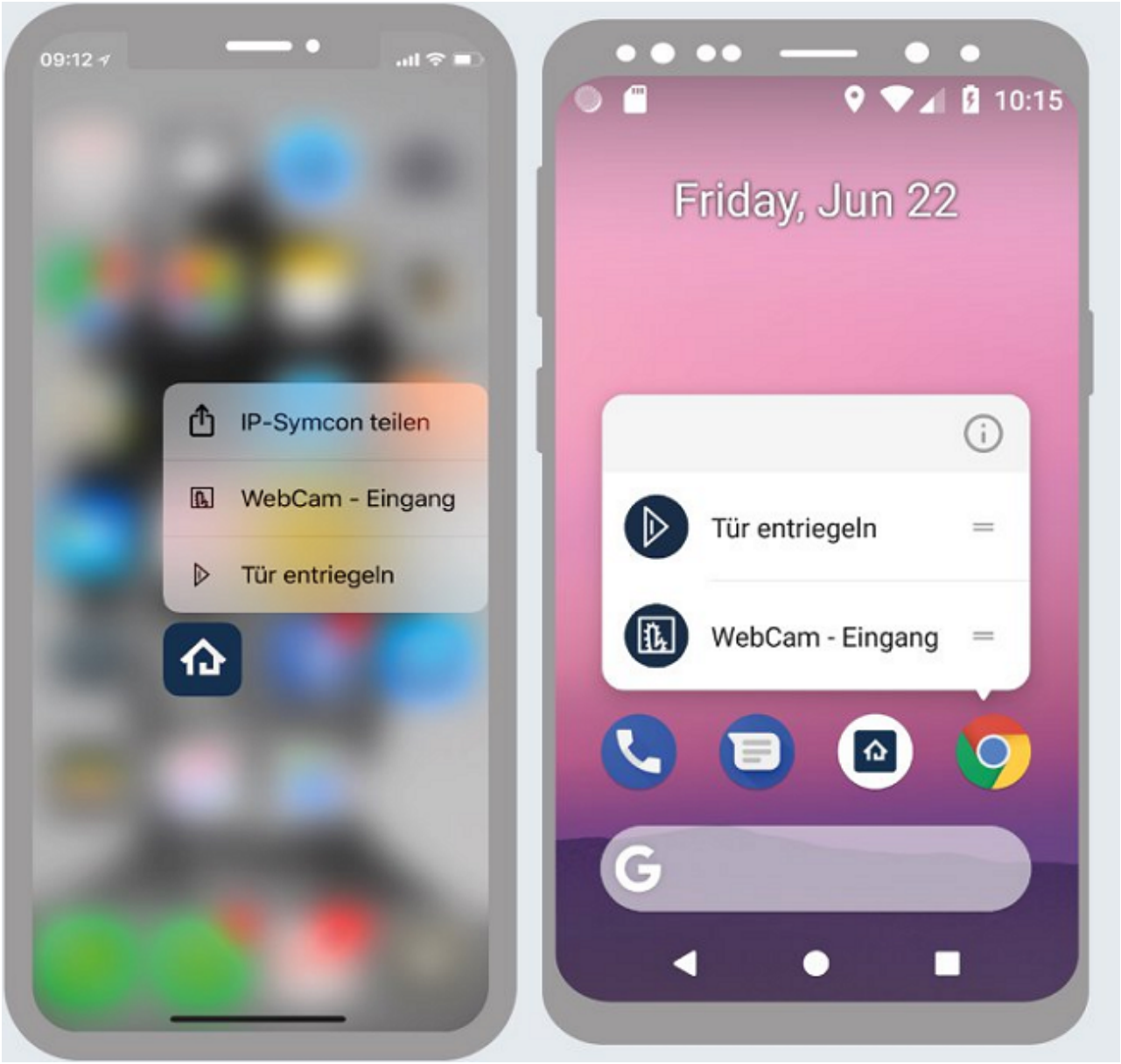
## DOCKER SUPPORT

Zusätzlich zu den bisherigen unterstützten Systemen Windows, MacOS, Ubuntu, RaspberryPi und unserer SymBox wird IP-Symcon “Docker” fähig.
Dies bedeutet, dass IP-Symcon auf allen x86 64bit Systemen lauffähig wird, auf denen Docker funktioniert. Dies ist im Rahmen von NAS Systemen wie Qnap und Synology interessant, welche zum Großteil “Docker” unterstützen. Das Docker Image basiert auf der Alpine Distribution und ist somit schlanke 40MB groß.



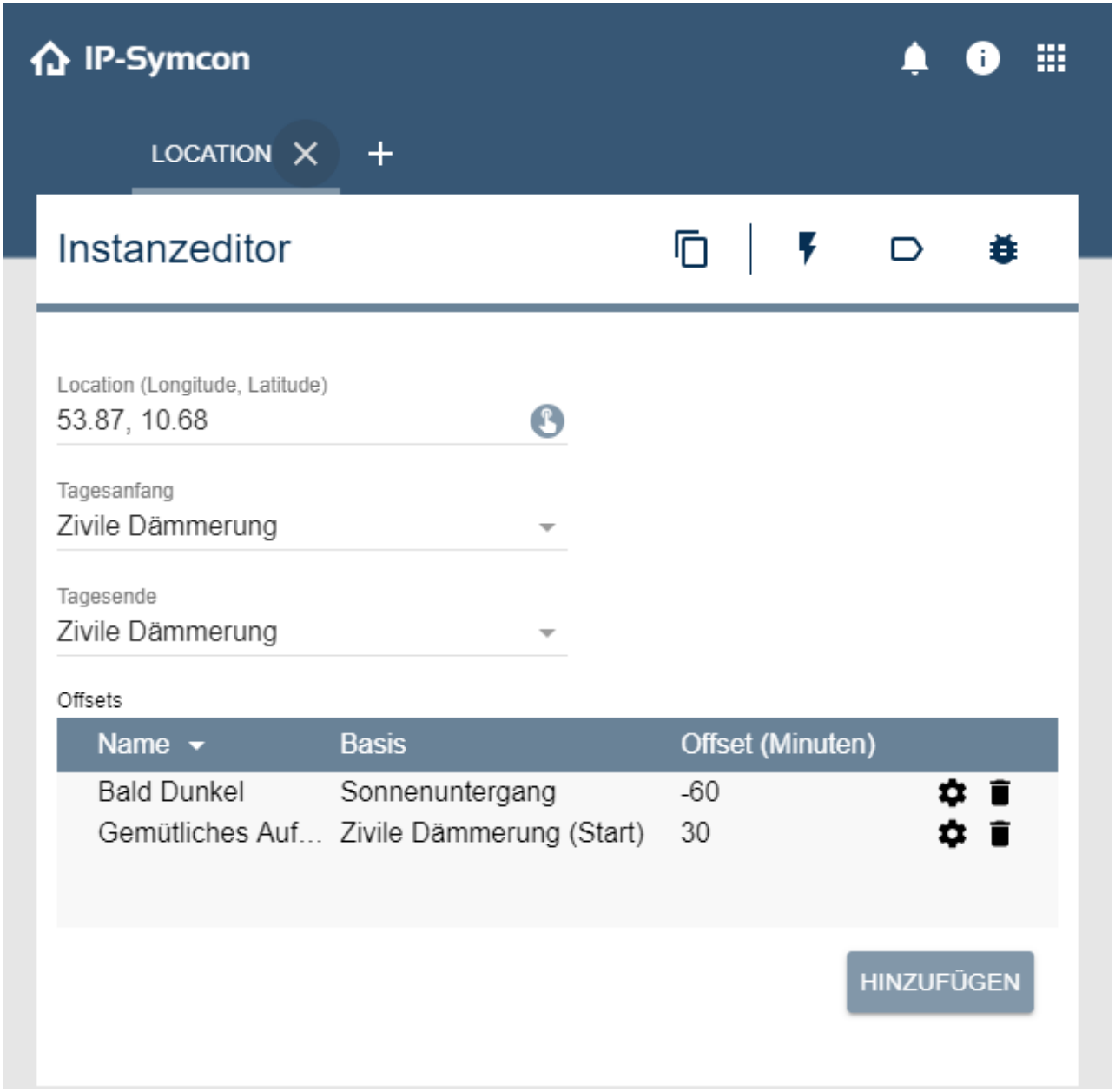

## VERBESSERUNG DER MOBILEN APPS

Mit der Version 5.0 von IP-Symcon bekommen auch die mobilen Apps weitere Features. Beim Öffnen von Push-Nachrichten können nun beliebige objektbezogene Funktionalitäten aufgerufen werden. Sei es das Öffnen einer Kategorie, das Anzeigen eines Bildes oder Skripte die gestartet werden sollen. Durch die “Force Touch Geste” (iOS) oder “Long Press” (Android) kann ein Schnellzugriffsmenü geöffnet werden, wodurch ebenfalls Aktions ausgeführt werden können.



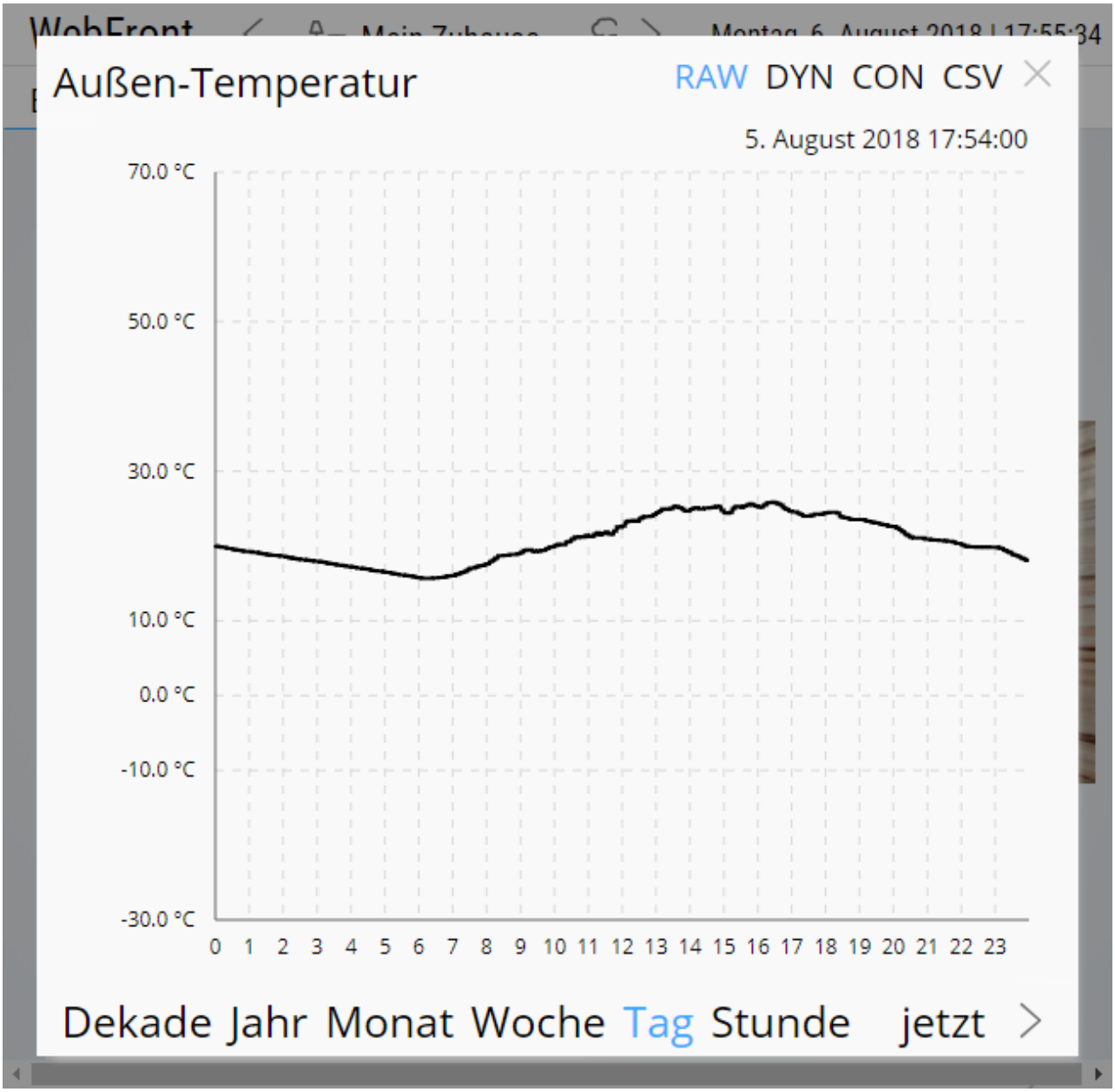

## ERWEITERUNG VOM LOCATION CONTROL

Mit der neuen Version des Location Control ist es möglich zu jedem zur Verfügung gestellten Zeitpunkt, sei es Sonnenaufgang, Beginn der Dämmerung, etc., einen Offset in Minuten hinzuzufügen um so individuelle Wünsche und persönliche Vorlieben zu verwirklichen. Somit können z.B. immer 30 Minuten vor Sonnenuntergang die Rollläden heruntergefahren werden




## NEUE FUNKTIONEN IM WEBFRONT

Das WebFront bekommt einige neue Features.
Die angezeigten Zeiten beziehen sich ab der IP-Symcon Version 5.0 standortunabhängig immer einheitlich auf die Serverzeit. Ist man beispielsweise im Urlaub in New York und möchte schon einmal seinen Wecker für den ersten Arbeitstag nach dem Urlaub einstellen, so kann man sich darauf verlassen, dass man an diesem Tag zur richtigen Zeit geweckt wird und nicht etwa um 8:00 nach der Zeitzone in New York.
Ebenso bekommen Graphen die Möglichkeit Rohdaten anzuzeigen, um eine absolut authentische Darstellung zu verwirklichen. Desweiteren können Multi-Graphen jetzt auch Bool- und Liniengraphen in einem Graphen vereinen und Übersichtlich darstellen.




## MIGRATION

Bitte erstellen Sie auf jeden Fall ein Backup Ihres gesamten IP-Symcon Ordners, um auf die Version 4.4 zurück zu wechseln, sollten wider erwarten Probleme mit der Version 5.0 auftreten.

Falls Sie von der Version 3.4 updaten, lesen Sie bitte vorher die Migrationsanleitung zur Version 4.0: Migration auf Version 4.0 , Version 4.1: Migration auf Version 4.1 , Version 4.2: Migration auf Version 4.2 , Version 4.3: Migration auf Version 4.3 und Version 4.4: Migration auf Version 4.4 .

Es wird Raspbian Stretch für Raspberry Pi erfordert! (Generelle Versionenübersicht)
Es wird Ubuntu 18.04 für Linux 64 Bit erfordert! (Generelle Versionenübersicht)
Für Microsoft Windows steht ausschließlich eine 64-Bit Version zur Verfügung.

Upgrade von PHP 5.6 auf PHP 7.2. (Upgrade auf 7.0 , Upgrade auf 7.1 , Upgrade auf 7.2 )
Profilnamen sind Case Sensitive. Ggf. müssen Skripte die Profile automatisch zuweisen korrigiert werden.
Ziele für Links werden beim Setzen auf Vorhandensein geprüft
Eigene Profile und eigene Aktionen für Variablen werden beim Setzen auf Vorhandensein und Gültigkeit geprüft
~String Profil entfernt, da es keinerlei nützliche Funktion hatte. Einfach kein Profil setzen ergibt die selbe Darstellung
RegisterVariable unterstützt für Server Sockets ein neues TYPE Feld, welches Verbunden/ Getrennt Zustände angibt. Diese werden ggf. bei alten Skripten als leere Datenpakete empfangen
Skins werden aus dem /webfront/ anstatt aus dem /webfront/user/ Ordner geladen. Dies löst ein Problem mit der Basis Authentifizierung bei eigenen WebServern, bei der man denken könnte das WebFront wäre Passwort-Geschützt, obwohl nur der Skin per Passwort gesichert war und dieser die Abfrage bewirkt hat.
Property im Location Control von "Latitude" und "Longitude" zu einer "Location" Property zusammengefasst.
Property im WebFront Konfigurator von IPAutoStart zu AutoStart umbenannt und vom Format angepasst, sodass es Kompatibel zum "List" Element ist.
Property im WebFront Konfigurator von LANPasswordNeeded zu IgnorePasswordOnLAN umbenannt, sodass es passend zur vorhandenen Checkbox genutzt werden kann
Bei dS Shutter, xComfort Shutter und Z-Wave Shutter wurde das Profil zur Ansteuerung auf ~ShutterMoveStop geändert
PHP-Funktion setlocale wurde aufgrund von Nebenwirkungen auf das Restsystem (siehe Warnung in PHP-Dokumentation) deaktiviert

## MIGRATION (FÜR PHP-MODUL ENTWICKLER)

In Konfigurationsformularen ist das "label" Feld mit IP-Symcon 5.0 veraltet und sollte für neue Module nicht mehr verwendet werden. Stattdessen gibt es überall das "caption" Feld, welches Konsistent verfügbar ist.

## LISTE DER ÄNDERUNGEN

### ALLGEMEIN

Neu: Webbasierte Konsole zur plattformübergreifenden Konfiguration
Neu: PHP 7.2.5 inkl. GMP und Sodium Erweiterung
Neu: Zusätzliche Icons: Bath, Bike, Book, Cloudy, Doll, Door, Download, EnergyProduction, EnergySolar, EnergyStorage, Favorite, Fitness, FloorLamp, Gas, Handicap, Heart, Help, Link, Menu, Pants, Party, People, Presence, Raffstore, Remote, Sink, Sleet, Sunny, Teddy, Tee, Thunder, Umbrella (Übersicht)
Neu: Zusätzliche Profile: Euro, Dollar, ShutterMoveStop, ShutterMoveStep
Neu: Ereignisse mit ungültiger Variable werden mit rotem Ausrufezeichen dargestellt
Neu: Limit für Profil-Assoziationen auf 128 erhöht
Neu: Genutzte Zeitzone wird automatisch erkannt und in der php.ini angepasst
Neu: WebServer leitet Links ohne Slash am Ende automatisch weiter, sofern möglich
Neu: Skins können nun Titel, Favicon und Geräte Icons (img) Ordner überschreiben

Neu: Summary bei I/O Instanzen wieder hinzugefügt (Fehlendes Feature von IP-Symcon 3.4)
Neu: Ereignis Port bei HomeMatic wird automatisch gesetzt (relevant bei mehreren CCUs)
Neu: Auslesen vom Subskriptionsdatum kann im Utils Control aktiviert werden
Neu: Multicast Socket kann auf allen Interfaces gleichzeitig lauschen (0.0.0.0)
Neu: UDP Socket/Multicast Socket können nun über die Server Socket GUIDs beim Datenfluss auf ClientIP/ClientPort lauschen und auf explizite Zieladressen senden.
Neu: Spezialschalter "ServerLogging", welcher die access.log auch für den internen WebServer (Port 3777) aktiviert
Neu: access.log zeigt bei Zugriffen über Proxy die Remote-IPs an
Neu: WebFront Konfigurator liefert bei der Suche nach Referenzen alle genutzten Objekte (z.B. Startkategorien)
Neu: Windows Installer ist bei Upgrades wesentlich kompakter strukturiert
Neu: Spezialschalter OPcache wird unterstützt (Experimentell)

## SYSTEME & GERÄTE

Neu: KNX DPTs für fast alle von der KNX Association definierten Typen
Neu: Modul für eKey (ASCII Protokolle von Home, Multi, Net)
Neu: Eltako FHK14 kann die Soll Temperatur direkt übermitteln
Neu: EnOcean EEP A5-14-07
Neu: EnOcean EEP A5-14-08
Neu: EnOcean EEP A5-14-09
Neu: EnOcean EEP A5-14-0A
Neu: EnOcean FGW14 kann GeräteIDs bis 255 nutzen
Neu: xComfort: Unterstützung für xComfort Room Controller Touch (CRCA-00/08, CRCA-00/09, CRCA-00/10, CRCA-00/11)
Neu: Location Control unterstützt Offsets
Neu: Location Control hat Variablen für Sonnenhöhe und Sonnenrichtung, die alle 5 Minuten aktualisiert werden
Neu: Location Control kann mit Längengrad/Breitengrad auf 0 deaktiviert werden, um Variablen zu sparen
Neu: Neues Shutter Control, welches bei manuellem Fahren die Position nachführen kann
Neu: Neues Heating Control, welches auch Kühlen kann
Neu: ALLNET Geräte mit Analogausgang (z.B. RGBW) werden unterstützt
Neu: DNS-SD Control für Bonjour/Avahi
Neu: NAT Support (insbesondere für Docker wichtig!)
Neu: Für Web/Hook/OAuth Requests sind in der $_SERVER Variable alle Header verfügbar
Neu: Legacy Konsole zeigt Listen in voller Breite an
Neu: LCN Shutter, EnOcean Shutter, KNX Shutter nutzen die neuen Profile
Neu: Z-Wave Shutter, dS Shutter, xComfort Shutter an die neuen Profile angepasst
Neu: digitalStrom: Neue Funktionen DS_ShutterStepDown und DS_ShutterStepUp

## MOBILE APP (IOS)

Neu: Favoriten, welche per 3D-Touch direkt über den Home-Screen geöffnet werden können (ab iPhone 6s, iOS 10)
Neu: Neue Icons für IP-Symcon 5.0 hinzugefügt
Neu: Unterstützung für Rohdaten-Darstellung in Graphen
Neu: Push-Nachrichten können Ziel (TargetID) enthalten

Fix: Watch App hat beim Betreten von Kategorien Aktionen ausgeführt

## MOBILE APP (ANDROID)

Neu: Bessere Unterstützung für Android Oreo
Neu: Favoriten, welche direkt über den Home-Screen geöffnet werden können (ab Android 7.1)
Neu: Neue Icons aus IP-Symcon 5.0 hinzugefügt
Neu: Unterstützung für Rohdaten-Darstellung in Graphen
Neu: Push-Nachrichten können Ziel (TargetID) enthalten

## WEBFRONT

Neu: Datum/Uhrzeit bei Graphen ist auf der rechten Seite für eine bessere Darstellung bei kleineren Auflösungen
Neu: Boolean/Integer/Float Variablen ohne Profil, jedoch mit Aktionen können über Freitext verändert werden
Neu: unterstützt Links ohne Namen, wodurch der Name vom Zielobjekt geerbt wird
Neu: zeigt in den Ereignis Dialogen den vollständigen Namen an. (Funktioniert nun auch mit verlinkten Ereignissen)
Neu: erkennt nun eine Abweichung in der Zeitzone (zwischen Browser + Server) und zeigt eine entsprechende Meldung
Neu: zeigt alle Zeiten und Graphen (inkl. CSV Export) immer in der Server-Zeitzone an
Neu: Logout Widget (zur Zeit nur über WebFront Editor hinzufügbar)
Neu: Graphen können in der Aggregationsstufe Raw, also Rohdaten dargestellt werden
Neu: Idle Widget (Erlaubt das zurückspringen auf eine Seite nach Inaktivität)

## SYMOS

Neu: Kernel auf 4.9.73 aktualisiert
Neu: Willkommensseite bietet direkten Link zur neuen webbasierten Verwaltungskonsole an

## PHP-MODULE

Neu: Neue Elemente für Eigenschaftenseiten (ausschließlich für neue webbasierte Verwaltungskonsole)
Tree
Configurator
ExpansionPanel
Image
PopupButton
RowLayout
SelectTime
SelectDate
SelectDateTime
Neu: Kernel-Konstanten (Nachrichten) sind nun automatisch definiert
Neu: FM_CHILDADDED, FM_CHILDREMOVED um zu erkennen, wenn neue "Kinder" am

Datenfluss angemeldet wurden
Neu: SetValue, GetValue Funktion zum einfacheren Lesen/Schreiben von Werten
Neu: GetMessageList Funktion zum Auslesen der registrierten Nachrichten
Neu: GetBufferList Funktion zum Auslesen der verfügbaren Buffer
Neu: ServerSocket unterstütztein neues "Type" Feld mit Wert (0=Data, 1=Connect, 2=Disconnect)
Neu: LogMessage zum Senden von benutzerdefinierten Nachrichten ins Log

## FIXES

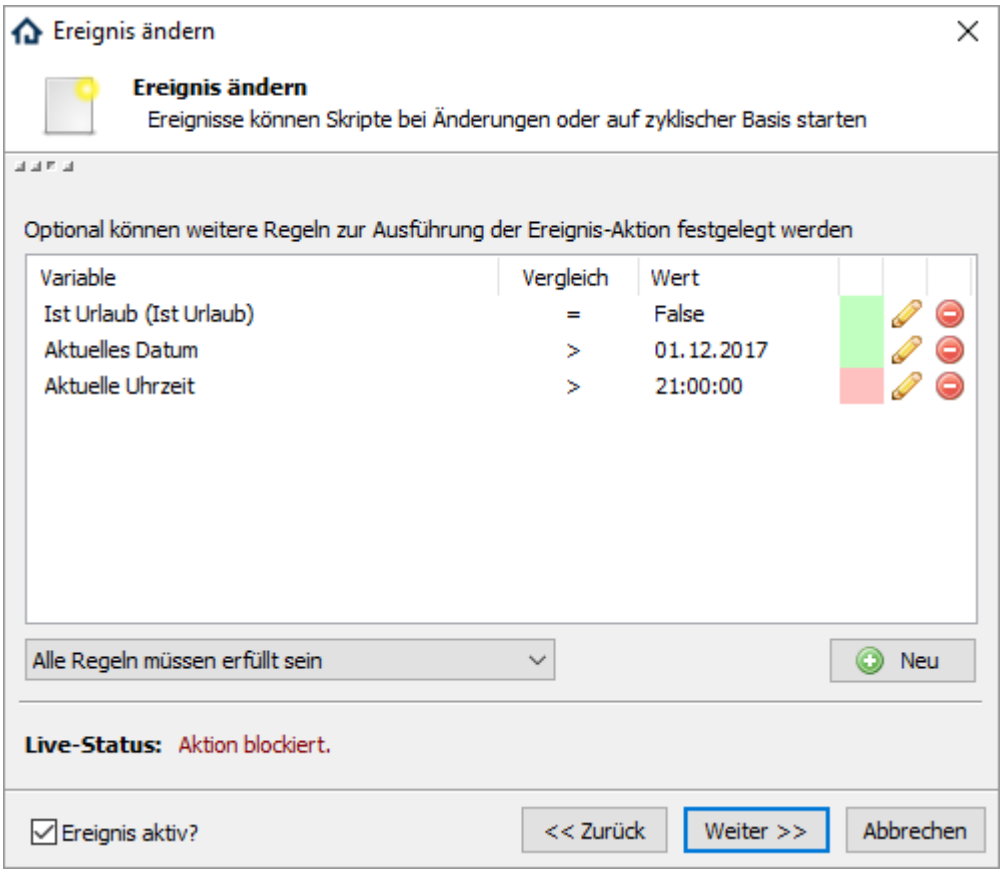
Fix: Funktion setlocale ist in PHP komplett deaktiviert
Fix: ML_LOAD/ML_UNLOAD Nachrichten werden korrekt versendet
Fix: Sofern in der Demo-Version das WebFront modifiziert wurde, wurden die Basic-Lizenz Restriktionen auf das WebFront nach dem Einspielen der Lizenz nicht korrekt angewandt.
Fix: Gleichzeitige Updates von Modules über das Module Control könnten einen Deadlock verursachen
Fix: WebFront: Circle Slider Wert wurde bei kleinen Auflösung nicht korrekt zentriert
Fix: WebFront: Circle Slider Editier-Animation korrigiert
Fix: WebFront: Color Wheel wird besser zentriert bei kleinen Auflösungen
Fix: Beim Duplizieren werden ungültige Inhalte (z.b. ungültiges Profil bei einer Variable) nicht dupliziert
Fix: Fehlermeldung bei JSON-RPC Anfragen gegen Funktionen von PHP Modulen korrigiert
Fix: Absturz bei LCN_SetDisplayText mit leerem Inhalt korrigiert
Fix: DMX: 16-Bit Farbwerte wurden bei 100% im Colorwheel falsch angezeigt
Fix: DMX Instanzen konnten bei Kanal < 0 abstürze verursachen
Fix: Dashboard: Einige Speicherlecks korrigiert
Fix: Legacy Konsole: Einige Speicherlecks korrigiert
Fix: M-Bus Werte, welche ungültige Werte enthielten konnte weiteres Auslesen verhindern
Fix: Fernzugriff konnte unter MacOS nicht korrekt gesetzt werden
Fix: Dashboard konnte in der Demo-Version nicht getestet werden
Fix: Links auf Links sind nicht möglich und werden nun korrekt verhindert
Fix: IPS_GetVariableEventList gibt nun die korrekte aktuelle Liste von Ereignissen zurück
Fix: Wenn die settings.json korrupt war und eine aus dem backup Ordner geladen wurde, wurden die Änderungen in die Backup-Settings gespeichert
Fix: Wenn bei PHP Module im Destroy eine Fehlermeldung geworfen wurde, konnte sich IP-Symcon aufhängen
Fix: Skripttext wurde in PHP Informationen ohne Limitierung angezeigt (Nun auf 100 Zeichen beschränkt)
Fix: MacOS: Fehlende SSL-Zertifikate hinzugefügt für cURL/PHP
Fix: MacOS: Beim Beenden des Dienste wurde ggf. die Settings nicht korrekt geschrieben
Fix: S7-Anbindung wertet vorhandene Fehlerinformationen besser aus
Fix: IMAP/POP3 Mail mit "kaputten" Headern werden besser eingelesen
Fix: Falsche Größenangaben in Sys_GetHardDiskInfo() korrigiert
Fix: Legacy Konsole: Standardwert beim Inhaltswechsler für Scaling war falsch
Fix: SE_EXECUTE Fehlermeldung, wenn ein Ereignis sich selbst gelöscht hat

# V4.3->V4.4 (Q4/2017)

## WESENTLICHE ÄNDERUNGEN

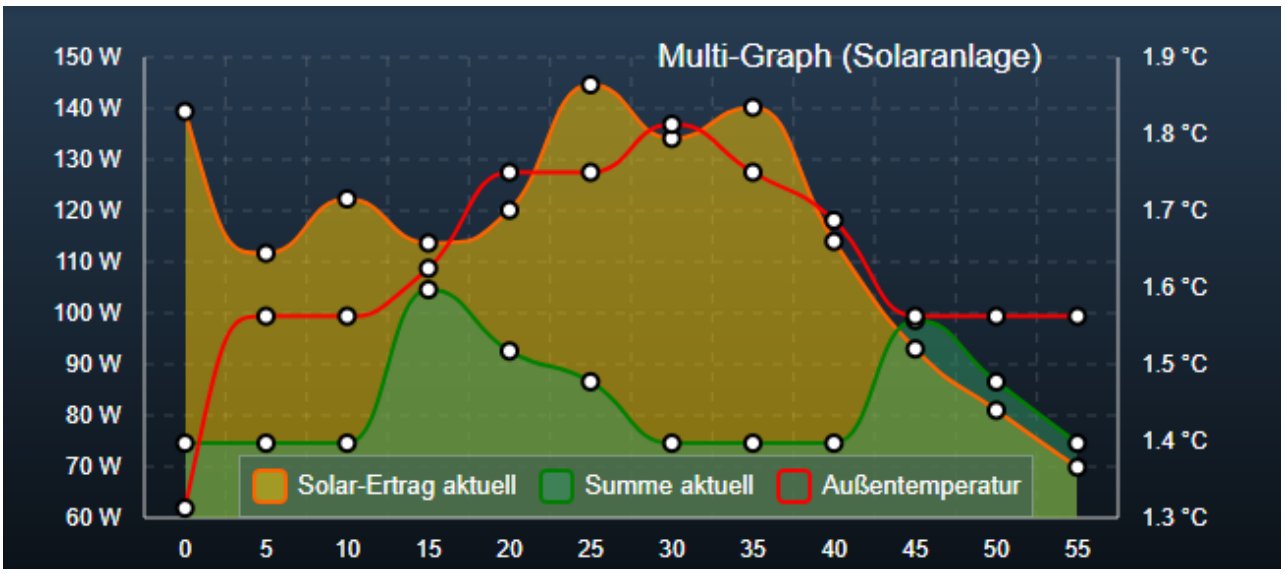
### Bedingte Ereignisse

Ereignisse können noch einfacher mit Bedingungen versehen werden ohne auch nur eine Zeile Code zu schreiben. Soll beim Betreten eines Raumes das Licht angeschaltet werden, aber nur wenn es draußen schon dunkel ist? Oder soll die Heizung automatisch auf den Komfortmodus wechseln, aber nicht wenn Sie im Urlaub sind? Diese Beispiele und mannigfaltige andere Möglichkeiten sind nun durch bedingte Ereignisse komplett ohne Skripte darstellbar.




### Graphen mit mehreren Achsen im WebFront

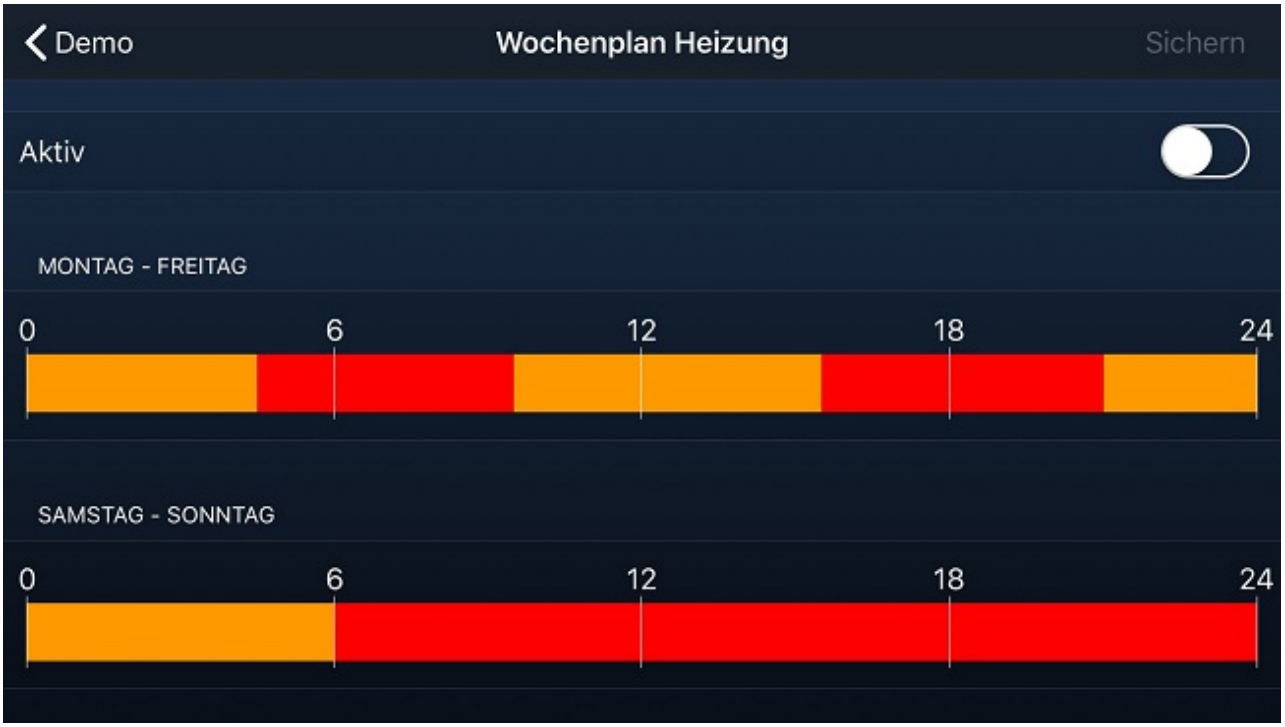
Graphen können nun mehreren Achsen dargestellt werden. So können komfortabel Variablen mit verschiedenen Profilen verglichen werden, wie beispielsweise Außentemperatur und gewonnene Solarenergie.




Graphen-Optionen in den mobilen Apps
Die mobilen Apps unterstützen ebenfalls die neuen Graphen mit mehreren Achsen. Zusätzlich sind alle Optionen wie dynamische Skalierung, hohe Dichte, Extrema, ect. verfügbar




Verbesserter Wochenplan in den mobilen Apps
Die Wochenplandarstellung wurde überarbeitet, sodass die Wochenpläne wie im WebFront intuitiv nutzbar sind. Durch Ziehen und Wischen kann der gewünschte Modus im Handumdrehen angepasst werden.




**Verbesserte Geräteunterstützung**
Verbesserungen bei EnOcean, HomeMatic, KNX, Siemens OZW, Technische Alternative CMI, Siemens S7, Z-Wave

## MIGRATION

Bitte erstellen Sie auf jeden Fall ein Backup Ihres gesamten IP-Symcon Ordners, um auf die Version 4.3 zurück zu wechseln, sollten wider erwarten Probleme mit der Version 4.4 auftreten.

Falls Sie von der Version 3.4 updaten, lesen Sie bitte vorher die Migrationsanleitung zur Version 4.0: Migration auf Version 4.0 , Version 4.1: Migration auf Version 4.1 , Version 4.2: Migration auf Version 4.2 und Version 4.3: Migration auf Version 4.3 .

Es wird Raspbian Stretch für Raspberry Pi erfordert! (Generelle Versionenübersicht)
Die 32-Bit Version für Microsoft Windows ist abgekündigt und wird zu IP-Symcon 5.0, welches 2018 erscheint, nicht mehr angeboten.
Die 64-Bit Version wird automatisch installiert, sofern möglich.

### GEÄNDERTE FUNKTIONEN

S7_WriteRegisterShortInt umbenannt zu S7_WriteRegisterChar
S7_WriteRegisterSmallInt umbenannt zu S7_WriteRegisterShort
ModBus_WriteRegisterShortInt umbenannt zu ModBus_WriteRegisterChar
ModBus_WriteRegisterSmallInt umbenannt zu ModBus_WriteRegisterShort
Rückgabeparamteter von AC_DeleteVariableData auf Integer geändert, welcher die Anzahl der betroffenen Datensätze zurückgibt.

IPS_SetEventTriggerValue validiert den Wert, sodass dieser zur Trigger-Variable passt

## LISTE DER ÄNDERUNGEN

### ALLGEMEIN

Neu: PHP 5.6.32
Neu: Multi-Charts können mehrere Achsen (Links/Rechts) mit verschiedenen Profilen haben
Neu: Bedingungen (Variable, Datum, Uhrzeit) für ausgelöste Ereignisse
Neu: Brute-Force Attacken auf den Fernzugriff werden nach 5 Versuchen verlangsamt und nach 15 Versuchen blockiert
Neu: Raspberry Pi Version benötigt/basiert auf Raspbian Stretch
Neu: Erstellen-Button beim Erstellen/Editieren von Variablen für einfaches anlegen von Standardaktionsskripten
Neu: Rekursives duplizieren von Objekten
Neu: Sollte die settings.json defekt sein, wird nach einer funktionierenden settings.json im Backup Ordner gesucht
Neu: Standardprofile für Energie, Wasser, Gas haben nun Icons zugewiesen bekommen
Neu: "Im Lizenz anzeigen" Dialog wird die genutzt Plattform angezeigt

### SYSTEME & GERÄTE

Neu: HomeMatic Anbindung sollte nun korrekt funktionieren, wenn zwei IP-Symcon auf die selbe CCU zugreifen
Neu: KNX: Unterstützung für Weinzierl kBerry
Neu: Siemens Instanzen unterstützt nun Strings (Array of Char)
Neu: Siemens OZW unterstützt nun Verbindungen per SSL
Neu: LCN GFPS wird unterstützt
Neu: LCN: Limit Variable für LCN Shutter Motor
Neu: EnOcean D2-32-*
Neu: EnOcean Eltako FGW14 kann bis zu 255 Geräte ansprechen
Neu: AC_DeleteVariableData liefert als Rückgabe die Anzahl der betroffenen Datensätze
Neu: Z-Wave: ZW_DimUp, ZW_DimDown, ZW_DimSetEx, ZW_DimUpEx, ZW_DimDownEx
Neu: W&T ThermoHygro unterstützt nun Geräte mit mehr als drei Kanälen

### PHP-MODULE

Neu: PHP Module dürfen nun auch einen "tests" Ordner für automatisierte Tests enthalten.

### FIXES

Fix: Unter frischen Windows 10 Systemen konnte ggf. SSL nicht korrekt genutzt werden (insbesondere Push-Nachrichten/Connect ect.)
Fix: Funktionen in Siemens und ModBus Instanzen umbenannt.* Siehe Migration
Fix: Die settings.json wird in den Backup Ordner kopiert anstatt verschoben, sodass zu jedem Zeitpunkt garantiert ist, dass eine settings.json vorhanden ist.
Fix: CSV Export konnte nicht im Firefox korrekt kopiert werden

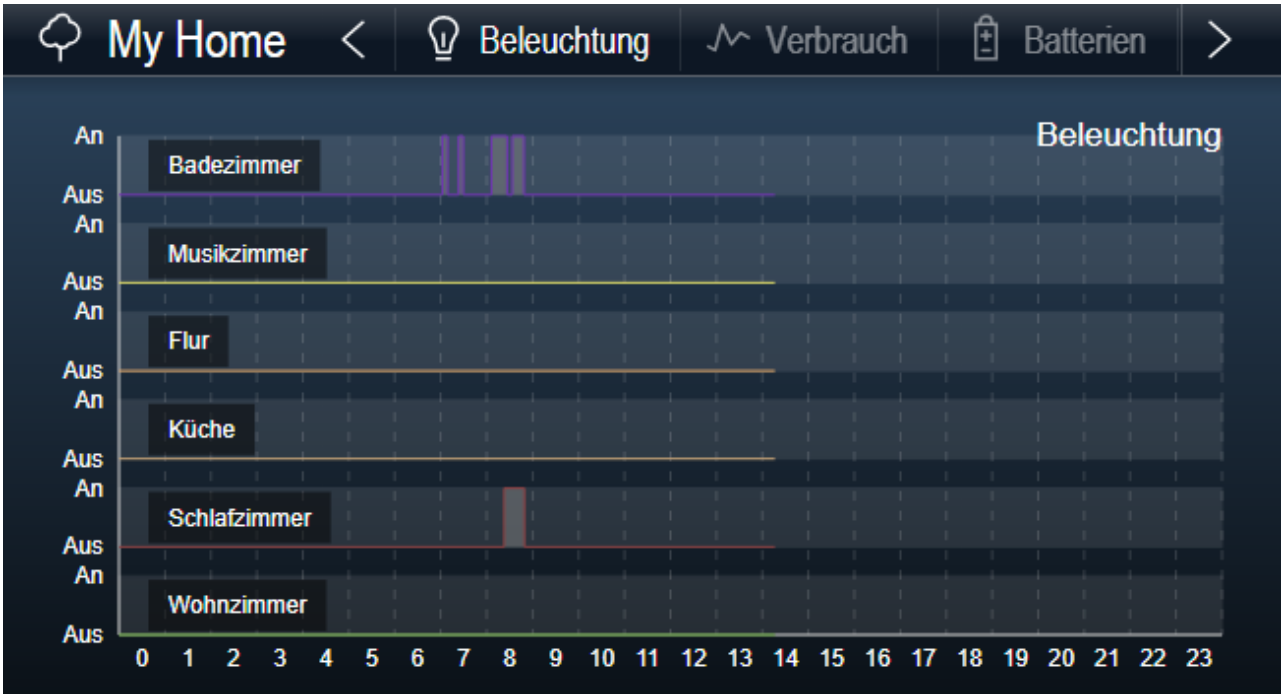
Fix: Problem unter MacOS mit Auto-Update gelöst
Fix: Verwaltungskonsole kann auf dem Mac wieder über CrossOver gestartet werden (über das Menü)
Fix: Charts werden nun in die Referenzsuche einbezogen
Fix: Sollte bei einem Multichart die Variable nicht mehr existieren, kann der Chart trotzdem editiert werden
Fix: CMI Abfrage der Ausgänge konnte nur deaktiviert werden, wenn auch die Eingänge deaktiviert waren
Fix: CMI Profile hatten kein Leerzeichen bei den Suffxen, wodurch diese "geklebt" haben
Fix: WebFront: Wochenplan Aktionen konnten nicht direkt von 0:00 bis 0:00 gesetzt werden
Fix: Access Violation beim Rechtsklick im Message-Log sofern die ObjektID nicht vorhanden ist
Fix: dSS11 braucht bei größeren Gebäuden mehr Zeit, um die Informationen zu liefern (nun max. 30 Sekunden)
Fix: Auf der SymBox wurden nicht alle Festplatten für Sys_GetHardDiskInfo angezeigt
Fix: Heating Control hat Ausgänge von LCN Unit Instanzen nicht korrekt angeschaltet
Fix: Fehlerhafte fortlaufende Zähleraggregation, wenn Werte nur einmal pro Stunde auf fast genau der vollen Stunde geloggt wurden

# V4.2->V4.3 (Q3/2017)

## WESENTLICHE ÄNDERUNGEN

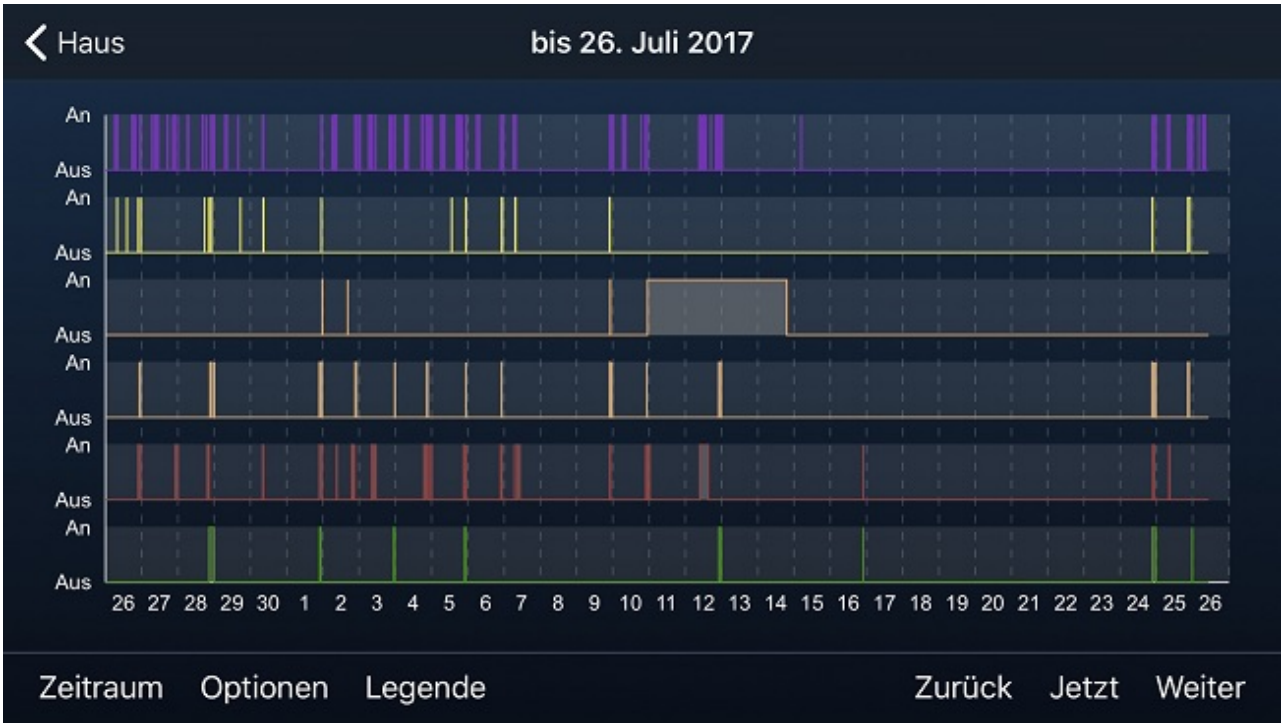
### Graphen im WebFront

Die Graphen im WebFront wurden komplett überarbeitet, sodass diese direkt vom IP-Symcon Server erzeugt werden. Das WebFront ist dadurch um ~15% kleiner, wodurch es über das Internet schneller lädt. Zusätzlich wurde die Darstellung verbessert, sodass die Graphen weniger S-Kurven-artig aussehen.



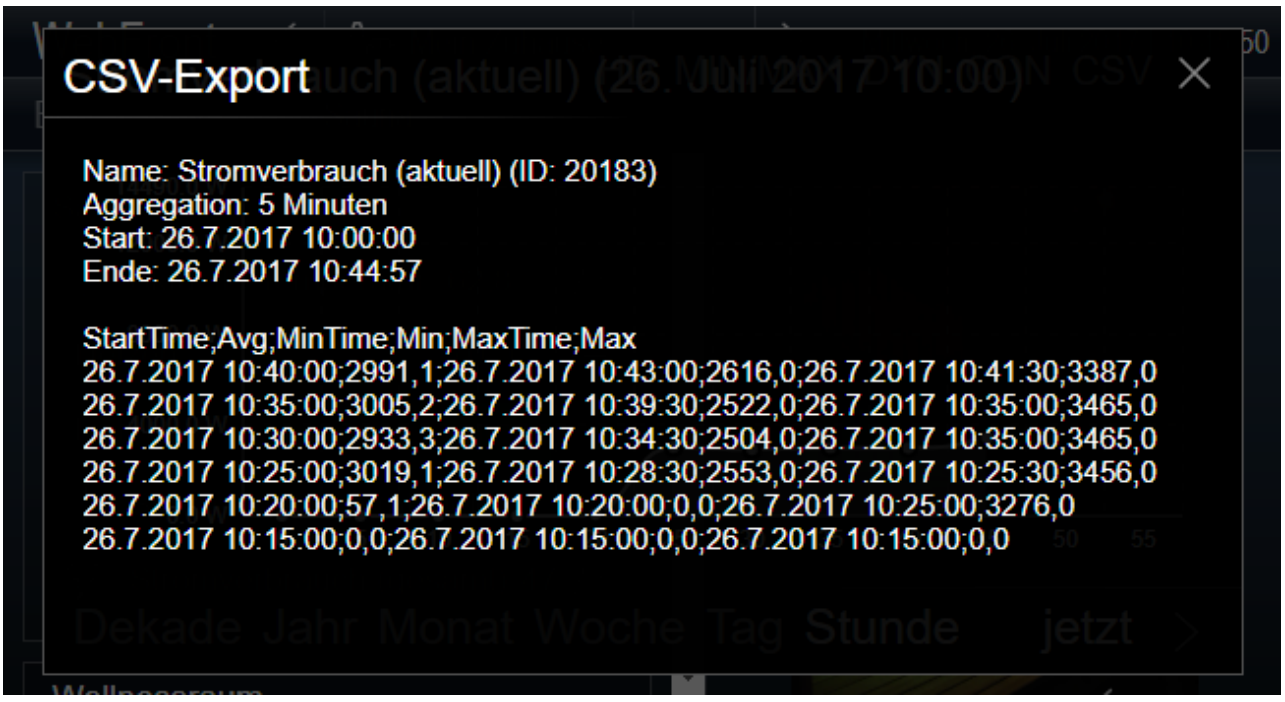

### Graphen in den mobilen Apps

Die mobilen Apps unterstützen alle Optionen für Graphen wie die dynamische Skalierung, hohe Dichte und das Anzeigen von Extrema. Zusätzlich werden auch die Multi-Graphen unterstützt, sodass Auswertungen direkt auf dem mobilen Gerät durchgeführt werden können.




CSV-Export im WebFront

Der CSV-Export unterstützt nun Multi-Graphen und zeigt die einzelnen Datenreihen nacheinacher an. Es wird dabei jeweils Name, Datum, Uhrzeit und Aggregationsstufe angezeigt. Die Daten sind in einem Microsoft Excel kompatiblen Format und können direkt kopiert werden.




Bessere Unterstützung der Geräte

- Module für EnOcean D2-01-, *D2-06-*
- Module für Eltako FGW14
- Module für LCN Display und LCN RGBW

Verbesserung für PHP-Module

- Beliebige Strings können in Modulen übersetzt werden
- Module überprüfen vor dem Update auf Kompatibilität
- Module können lokale Änderungen automatisch entfernen vor einem Update

*(Um den WebFront-Editor nutzen zu können ist mindestens eine IP-Symcon Professional Edition nötig. Alle anderen Funktionen sind in der Basic, Professional und Unlimited Edition von IP-Symcon verfügbar und können mit einer gültigen Subskription bezogen werden. Ein spezielles "Upgrade" muss nicht erworben werden. Sofern Ihre Subskription abgelaufen ist, können Sie diese einfach erneuern und direkt ohne zusätzliche Mehrkosten auf die Version 4.3 updaten. Bitte beachten Sie, dass das Nutzen der Push-Benachrichtigungen und dem Symcon Connect Dienst eine Verbindung zum Internet und eine gültige Subskription voraussetzt. (Shop ))*

## MIGRATION



Bitte erstellen Sie auf jeden Fall ein Backup Ihres gesamten IP-Symcon Ordners, um auf die Version 4.2 zurück zu wechseln, sollten wider erwarten Probleme mit der Version 4.3 auftreten.



Falls Sie von der Version 3.4 updaten, lesen Sie bitte vorher die Migrationsanleitung zur Version 4.0: Migration auf Version 4.0 , Version 4.1: Migration auf Version 4.1 und Version 4.2: Migration auf Version 4.2 .

Die 32-Bit Version für Microsoft Windows ist abgekündigt und wird zu IP-Symcon 5.0, welches 2018 erscheint, nicht mehr angeboten.
Die 64-Bit Version wird automatisch installiert, sofern möglich.

## MIGRATION (FÜR PHP-MODUL ENTWICKLER)

Für Übersetzungen sollte ab sofort die neue Translate Funktion eingesetzt werden.

## LISTE DER ÄNDERUNGEN

### GRAPHEN IM WEBFRONT

Neu: WebFront nutzt die neuen Graphen, welche auch die Apps nutzen
Neu: WebFront ist nun ~100kb kleiner, da die Graphen vom Backend erzeugt werden
Neu: Splines swingen nicht mehr so sehr (Vermeidet die typischen S-förmigen Kurven)
Neu: CSV-Export verbessert, sodass die Daten besser kopierbar sind
Neu: Kontinuierliche Graphen haben im Titel das "bis" wie die mobilen Apps
Neu: Legende bei Multi-Graphen ist kompakter, sofern kein Offset benutzt wird

### GRAPHEN IN DEN MOBILEN APPS

Neu: Multi-Graphen werden in den mobilen Apps angezeigt

## SYSTEME & GERÄTE

Neu: Unterstützung für EnOcean Gateway "Eltako FGW14"
Neu: Unterstützung für EnOcean EEP D2-01-*
Neu: Unterstützung für EnOcean EEP D2-06-*
Neu: Technische Alternative CMI Modul unterstützt die Netzwerk Eingänge/Ausgänge
Neu: LCN_SwitchRelayTimer
Neu: LCN Display Modul
Neu: LCN RGBW Modul
Neu: Z-Wave: Zipato Mini Keypad zeigt nun in einer Variable die ID vom Benutzer an, welcher die Tür zuletzt geöffnet/geschlossen hat

## PHP-MODULE

Neu: Module prüfen die Kompatibilität zur IP-Symcon Version vor einem Update
Neu: Translate Funktion für Module
Neu: Sofern das Modul keine HTTP/HTTPS Adresse hat wird gefragt, ob die aktuelle Adresse in die Zwischenablage kopiert werden soll
Neu: Module können aktualisiert werden, auch wenn lokale Änderungen vorgenommen wurden (Änderungen ggf. werden verworfen)
Neu: Module unterstützen das Überschreiben von gleichen Funktionen mit dem selben Prefix durch andere Module bei gleichen Parametern
Fix: Module: Editierbare Listen im Aktionsbereich funktionieren korrekt
Fix: Modulliste wird nun ohne Beachtung der Groß- und Kleinschreibung sortiert
Fix: Module: Select* Elemente können im Aktionsbereich ausgewertet werden
Fix: Wenn bei Listen kein gültiges "values" Array geschickt wurde, gab es eine AccessViolation
Fix: Fehlende Farben bei Modulen mit Listen, sofern die Liste im "elements" Bereich war und vorhandene Einträge überschrieben hat
Fix: Module mit sauberem Type Hinting warfen einen Fehler sofern diese aus ProcessHookData genutzt wurden
Fix: Leere Strings wurden für rowColor nicht korrekt ausgewertet
Fix: Module: Rot+Blau waren beim Wert für SelectColor vertauscht
Fix: Module welche beim Erstellen Fehler hatten wurden nicht sauber aufgeräumt wenn bereits Unterobjekt erstellt wurden
Fix: Module: Access Violation bei einigen Buttons im Aktionsbereich

## SONSTIGES

Neu: Wir würden gerne eine paar nicht personalisierte Benutzerdaten über Ihr System sammeln (Details )
Neu: Neuer Medientyp: Dokument, welcher zur Zeit txt, doc, xml und pdf Uploads erlaubt
Neu: Bibliothek zum Formatieren von Strings (z.B. %f, %d) wurde gegen eine schnellere getauscht
Neu: Instanz hinzufügen Dialog zeigt den Author des Moduls an
Neu: Konsole: WebFront Konfigurator wählt beim Duplizieren das selektierte Element automatisch vor
Neu: DWD Wetterdaten werden über neues CDN geladen

## FIXES

Fix: Z-Wave Multisensor wählt korrekte Profile aus
Fix: Z-Wave Parameterliste liest "Nur Schreiben" Parameter nicht mehr aus
Fix: Z-Wave: Besseres Handling von vielen gleichzeitigen verschlüsselten Nachrichten
Fix: Z-Wave: Konsole: WAKE_UP wurde nicht korrekt erkannt, wenn vorher ein Timeout beim Auslesen eines Parameters passiert ist
Fix: Z-Wave: Beim Laden von Parametern kann dies korrrekt erneut versucht werden, sofern eine Unterbrechung stattgefunden hat
Fix: Parameterinformationen für IPSModule Klasse
Fix: Filter im Instanz-Debug wurde für neue Nachrichten nicht angewendet
Fix: KNX Konfigurator überprüft ESF Datei auf dreistufige GAs
Fix: KNX Repeat Flag wird nicht mehr gesetzt
Fix: Rundungsfehler im WebFront
Fix: WebFront hat mit einer Fehlermeldung quittiert, sofern ein eingebetteter Graph mit einer nicht existenten Variable vorhanden war
Fix: Für EnOcean Thanos konnte die Temperatur aus dem WebFront nicht korrekt eingestellt werden
Fix: Kommentare zu Ereignissen werden im SkriptEditor korrekt erstellt
Fix: WebFront: CSS Spezialklassen wurden nicht korrekt an verlinkte Objekte angefügt
Fix: Bei Ereignissen musste ein Ziel angegeben werden, obwohl dies gar nicht notwendig war
Fix: Objektnamen waren Leer wenn diese per Skript erstellt wurden und kein IPS_SetName folgte
Fix: Seltenes Problem beim Auslesen von dS Joker Klemmen korrigiert
Fix: Problem gelöst, wenn Gerätetaster von dS Klemmen im SceneResponder genutzt wurden
Fix: Für nicht existente Variablen konnte Logging aktiviert werden
Fix: Multicast Pakete konnten auf Linux, Raspberry Pi und MacOS nicht empfangen werden
Fix: Berechnung von LCN Motor war noch falsch
Fix: Ein Problem beim Starten mit der Aggregation gelöst
Fix: Wir löschen curl.cacert Einträge in der php.ini, wenn diese für die Plattform unnötig sind
Fix: Die Vorschläge im Circle Slider beachten nun die Schrittweite
Fix: Beim Öffnen von Variablen im Archiv wurde der Baum neu geladen und die Scrollposition wurde zurückgesetzt
Fix: LCN_SwitchRelayTimer hat auf dem ersten Kanal falschen Wert gesendet
Fix: Multicast Socket sendet unter Windows nun korrekt über die ausgewählte Schnittstelle (BindIP)
Fix: Speicherleck bei PHP-Modulen, welche den Datenfluss verwendet haben
Fix: Kleines Speicherleck beim Aktualisieren von PHP-Modulen
Fix: Potentielle Race Condition im Discovery Server
Fix: Potenzieller Deadlock beim Laden von Modulen
Fix: Cutter kann HID als I/O auswählen
Fix: %d funktionierte in GetValueFormatted nicht bei Assoziationen
Fix: Windows Installer erstellt keine neuen Einträge pro Version
Fix: Graphen: Bei Graphen, welche Boolean oder HD sind, wird die X-Achsen-Beschriftung korrekt links am Startpunkt gezeichnet
Fix: Graphen: Boolean-Graphen mit Wert-Wechseln auf der selben Sekunde werden korrekt angezeigt
Fix: Graphen: Boolean-Graphen bei dem der erste Wert exakt auf den Startwert fiel,

werden korrekt angezeigt
Fix: Graphen: Einzelne Boolean-Graphen haben keinen "hellen" Hintergrund mehr, um den Kontrast zu verbessern
Fix: WebFront: Internet Explorer 11 funktioniert wieder korrekt
Fix: WebFront: CSV-Export zeigt bei Boolean Variablen Min/Max nicht mehr
Fix: LCN RGBW kann über WebFront angesteuert werden
Fix: LCN Shutter hat bei Ausgängen nicht korrekt geschaltet
Fix: LCN Shutter Motor verändert nun auch die Aktionsvariable je nach Richtung
Fix: Konsole: Shortcut (STRG+C) zum Kopieren von Objekt / ObjektID ist nun sichtbar
Fix: Konsole: Sortierung im Calendar Control korrigiert
Fix: KNX Group: Checkboxen waren nicht breit genug für deutsche Übersetzung
Fix: Instanz für eine Test-Modul wird nicht mehr erstellt, wodurch Fehlermeldungen im Log auftraten
Fix: Archiv war defekt für Zeitzonen westlich vom Nullmeridian
Fix: Archivgröße wurde bei >2GB Größe als negativ angezeigt
Fix: Sofern das Archiv defekt ist, wird dies beim Editieren der Variable angezeigt
Fix: OZW Instanzen konnten nicht korrekt erstellt werden (seit IP-Symcon 4.3 Release)
Fix: Module mit lokalen Änderungen konnten teilweise nicht zurückgesetzt/aktualisiert werden, obwohl die Sicherheitsabfrage bestätigt wurde
Fix: Problem mit EnOcean D2 Geräten, welche nur einen Kanal hatten und nicht auf KanalID 0 angesprochen wurden
Fix: IPS_GetLiveUpdateVersion ist wieder kompatibel zu IP-Symcon 4.0-4.2
Fix: Möglicher Absturz nach Allnet-Update verhindert und durch entsprechende Fehlermeldung ersetzt, wenn Remote Control nicht aktiviert wurde
Fix: IPS_Execute hat unter Linux den geforkten Prozess nicht abgebrochen, wenn die neue Applikation nicht gestartet werden konnte
Fix: EIS5 bei KNX konnte nicht abgefragt werden
Fix: Viele Fehlermeldungen wenn unter Linux Locales vom Typ de_DE@euro (ohne .utf-8) verwendet wurden
Fix: Es sollte nun zuverlässig möglich sein mehrere IP-Symcon Systeme an einer CCU anzumelden
Fix: WebFront: Bei Boolean-Variablen mit Aktion und zu wenig Platz konnte der Button für das List-Popup nicht angeklickt werden
Fix: WebFront: Bei Variablen mit Aktion und zu wenig Platz wurde manchmal ein Scrollbalken im Element angezeigt
Fix: Fehlerhafte Daten von der ELV FHZ konnten den Dienst zum Absturz bringen
Fix: Z-Wave: Zipato Light Bulb funktioniert nun korrekt
Fix: Z-Wave Regression seit IP-Symcon 4.2 bei der Kommunikation mit FLIRS Geräten (Bitte Instanz nach dem Update neu "Laden", um Problem zu lösen!)
Fix: Bei 2 Kanal W&T Geräten wurde der Status nicht korrekt aktualisiert
Fix: Wenn bei HomeMatic Werte ohne Datentyp gesendet werden, loggen wir diese nicht mehr als Fehlerhaft im Log (z.B. SUBMIT)
Fix: Clickatell SMS funktioniert wieder korrekt
Fix: Seit dem Clickatell Fix konnte der digitalStrom dSS nicht mehr korrekt angesprochen werden
Fix: Fehlerhafte FHZ Pakete konnte IP-Symcon zum Absturz bringen
Fix: CMI Abfrage der Ausgänge konnte nur deaktiviert werden, wenn auch die Eingänge deaktiviert waren
Fix: CMI Profile hatten kein Leerzeichen bei den Suffxen, wodurch diese "geklebt" haben

# V4.1->V4.2 (Q2/2017)

## WESENTLICHE ÄNDERUNGEN

IP-Symcon 4.2 gibt es für Microsoft Windows zusätzlich als 64-Bit Version. Dadurch können größere Projekte mit mehr Variablen und Archivdaten realisiert werden
PHP wurde auf die aktuellste Version (5.6.30) aktualisiert
Neues Modul für das CMI der Technischen Alternative
Neues Modul für den LCN-BS4 (LCN Shutter Motor)
Verbesserung vom Eltako Shutter, sodass die Fahrzeiten angegeben werden können
Viele Verbesserungen beim SDK für PHP Module wie z.B. native Timer, bessere Validierung und GetConfigurationForParent
Viele Verbesserungen bei den Konfigurationsmöglichkeiten von PHP Modulen wie z.B. List, SelectColor, SelectFile
*(Um den WebFront-Editor nutzen zu können ist mindestens eine IP-Symcon Professional Version nötig. Alle anderen Funktionen sind in der Basic, Professional und Unlimited Version von IP-Symcon verfügbar und können mit einer gültigen Subskription bezogen werden. Ein spezielles "Upgrade" muss nicht erworben werden. Sofern Ihre Subskription abgelaufen ist, können Sie diese einfach erneuern und direkt ohne zusätzliche Mehrkosten auf die Version 4.2 updaten. Bitte beachten Sie, dass das Nutzen der Push-Benachrichtigungen und dem Symcon Connect Dienst eine Verbindung zum Internet und eine gültige Subskription voraussetzt. (Shop ))*

## MIGRATION



Bitte erstellen Sie auf jeden Fall ein Backup Ihres gesamten IP-Symcon Ordners, um auf die Version 4.1 zurück zu wechseln, sollten wider erwarten Probleme mit der Version 4.2 auftreten.



Falls Sie von der Version 3.4 updaten, lesen Sie bitte vorher die Migrationsanleitung zur Version 4.0: Migration auf Version 4.0 und Version 4.1: Migration auf Version 4.1 .

Die 32-Bit Version für Microsoft Windows ist abgekündigt und wird zu IP-Symcon 5.0, welches 2018 erscheint, nicht mehr angeboten.
Die 64-Bit Version steht nach dem Update auf IP-Symcon 4.2 direkt über "Auf Update suchen..." zur Verfügung.

## MIGRATION (FÜR PHP-MODUL ENTWICKLER)

Es stehen viele neue Funktionen zur Verfügung. z.B. GetConfigurationForParent, List, SelectColor, SelectFile
Die IPSModule Klasse ist jetzt nativ in IP-Symcon implementiert.
Breaking Change: Es stehen nur bestimmte Unterordner zur freien Verfügung. Siehe: Struktur

Breaking Change: Die Datentypen von RegisterProperty *und ReadProperty* müssen übereinstimmen.

## LISTE DER ÄNDERUNGEN

### NEUE FUNKTIONEN

64 Bit Version für Microsoft Windows verfügbar!
PHP 5.6.30
Modul für das CMI der Technischen Alternative (erfordert Firmware 1.25.1 oder neuer)
Modul für LCN Shutter (Motor)
MacOS Tray-Anwendung überprüft korrekt die Subskription beim Updatevorgang
Eltako Shutter kann direkt die Fahrzeit angeben über ENO_ShutterMoveUpEx, ENO_ShutterMoveDownEx

### NEUE FUNKTIONEN (PHP-MODULE)

SelectColor in Eigenschaftenseiten für PHP Module (Doku)
SelectFile in Eigenschaftenseiten für PHP Module (Doku)
List in Eigenschaftenseiten für PHP Module (Doku)
GetConfigurationForParent für PHP Module (Doku)
Die internen Timer (Millisekundenauflösung) werden in PHP Module genutzt
Überprüfung, ob bei PHP Modulen die library.json / module.json korrekt vorhanden ist und geparst werden kann (Ausnahme sind die Ordner: docs, libs, imgs)
Im Modul Control werden fehlerhafte Module markiert

### BEHOBENE FEHLER

Fix: OS X: Beim Neustart des Rechners wurde ggf. bemängelt, dass IP-Symcon nicht gestartet ist, obwohl es am Starten war
Fix: Bei PHP Modulen wurden JSON Daten (SendData*) nicht korrekt einbezogen, welche ein Array waren
Fix: Bei PHP Modulen wird jetzt korrekt das Standardprofil/Standardaktion gesetzt (Löst Problem mit eigenen Actionen/Profilen)
Fix: Bei PHP Modulen werden alle PHP Magic Funktionen nicht nach IP-Symcon exportiert
Fix: Bei PHP Modulen kann auf IPS_KERNELSTARTED/IPS_KERNELSHUTDOWN reagiert werden
Fix: Bei PHP-Modulen kann RegisterTimer nur noch im Create aufgerufen werden
Fix: Multigraphen welche Graphen ohne Datensätze anzeigen sollten, haben diese gar nicht angezeigt (auch nicht in der Legende)
Fix: Color-Wheel und Circle-Slider Presets werden nun pro ObjektID gespeichert
Fix: Lernen Button bei EnOcean Geräten sind nur nach dem Übernehmen anklickbar
Fix: M-Bus unterstützt korrekt Real32 Werte
Fix: LCN Module Eigenschaften Seite kann nun auch korrekt Instanzen löschen
Fix: RegisterVariable konnte hängenbleiben, wenn die Skriptausführung z.B. wegen zu vielen Skriptausführungen abgebrochen wurde
Fix: DMX_FadeRGBDelayed fadete bei vielen nacheinanderfolgeden Fadings falsch
Fix: ModBus 64Bit Real schreiben funktioniert nun korrekt

Fix: Im SkriptEditor wurde von Wochenplänen der Name nicht korrekt als Kommentar eingefügt
Fix: ALT+R funktionierte im Objektbaum nicht
Fix: Start-/Enddatum und Start-/Endzeit werden bei zyklischen Ereignissen korrekt überprüft
Fix: Z-Wave zeigt nun alle unterstützten Klassen an (auch die, die eher kryptisch/lowlevel sind)
Fix: Beim Ändern der Farbe im Multi-Chart wird die Transparenz automatisch deaktiviert
Fix: Speichern Button im Skript-Editor wurde beim Ändern einer ObjektID über STRG+O nicht aktiviert
Fix: Edit-Typen werden korrekt validiert, ob diese erlaubt sind
Fix: HomeMatic Socket hat KeepAlive durchgeführt, obwohl der Socket Inaktiv war

# V4.0->V4.1 (Q4/2016)

## WESENTLICHE ÄNDERUNGEN

Funktionserweiterung für diverse Hardware-Systeme (Z-Wave, HomeMatic IP, digitalStrom, EnOcean, DMX, ModBus, LCN, Siemens OZW)
Vollständig überarbeitete Verwaltungskonsole mit verbessertem Skript-Editor und Suche nach Querverweisen für beliebige ObjektIDs
Viele neue Funktionen für die Entwicklung der beliebten PHP Module
Neue PHP Module aus der Community u. a. Module von tommi, FireTV, Möhlenhoff Alpha 2, Alarmierung, Anwesenheitssimulation u.v.m. (Link zur Übersicht )
Skins für die Login Seite vom WebFront
Dialoge für Circle Slider und Auswahllisten im WebFront
Dialoge für Datum/Uhrzeit im WebFront
SDKs für Microsoft Excel und WebFront Skins

*(Um den WebFront-Editor nutzen zu können ist mindestens eine IP-Symcon Professional Version nötig. Alle anderen Funktionen sind in der Basic, Professional und Unlimited Version von IP-Symcon verfügbar und können mit einer gültigen Subskription bezogen werden. Ein spezielles "Upgrade" muss nicht erworben werden. Sofern Ihre Subskription abgelaufen ist, können Sie diese einfach erneuern und direkt ohne zusätzliche Mehrkosten auf die Version 4.1 updaten. Bitte beachten Sie, dass das Nutzen der Push-Benachrichtigungen und dem Symcon Connect Dienst eine Verbindung zum Internet und eine gültige Subskription voraussetzt. (Shop ))*

## MIGRATION



Bitte erstellen Sie auf jeden Fall ein Backup Ihres gesamten IP-Symcon Ordners, um auf die Version 4.0 zurück zu wechseln, sollten wider erwarten Probleme mit der Version 4.1 auftreten.



Falls Sie von der Version 3.4 updaten, lesen Sie bitte vorher die Migrationsanleitung zur Version 4.0: Migration auf Version 4.0 .

Es wird Raspbian Jessie für Raspberry Pi erfordert! (Generelle Versionenübersicht)
Es wird Ubuntu 16.04 für Linux 64 Bit erfordert! (Generelle Versionenübersicht)

## LISTE DER ÄNDERUNGEN

### NEUE FUNKTIONEN

PHP 5.6.26 (löst insbesondere das microtime() Speicherleck und enthält offiziell den cURL Crashfix)
Z-Wave unterstützt mehr Klassen (SECURITY, COLOR , CENTRAL_SCENE, DOOR_LOCK ,

USER_CODE, TIME, SWITCH_ALL )
Z-Wave zeigt List der Geräteparameter an
Z-Wave hat Möglichkeit die interne WAKE_UP Warteschlange einzusehen
Z-Wave optimiert jede Nacht alle Nodes automatisch (zur Zeit im 5 Minuten Abstand)
Z-Wave kann die Erreichbarkeit der (auch batteriebetriebenen) Geräte direkt aus dem Konfigurator prüfen
Z-Wave Funktion ZW_ConfigurationSetValueEx kann wieder für batteriebetriebene Geräte ausgeführt werden
HomeMatic IP (Vollständiger Umbau von Bin-RPC auf XML-RPC)
digitalStrom: Unterstützung für App-Taster über Joker Klemmen
digitalStrom: Unterstützung für Sensoren an z.B. iSens200
digitalStrom: Unterstützung für binäre Eingänge an z.B. Automatisierungsklemmen
KNX-Shutter unterstützt weitere Rückmeldeadressen und Stepping für Lamellen
KNX-RGB/KNX-RGBW unterstützt weitere Rückmeldeadressen
DMX unterstützt 16 Bit Adressen
xComfort Dimmaktoren (Neue Gerätegeneration mit speziellen Rückmeldungen wird unterstützt)
ModBus Module unterstützt explizite Angabe des Funktionscodes (01, 02, 03, 04, 05, 06, 15, 16)
ModBus RTU/TCP Gateway-Instanzen synchronisieren sich korrekt, sofern mehrere an einer I/O Instanz dran sind
Unterstützung weiterer Eltako EnOcean Geräte (FKC, FKF, FRW, FTR65DS, FTR65HS, FUTH65D, FZS)
Unterstützung von EnOcean EEP F6-10-00
LCN unterstützt die Einstellung für 200 Dimmschritte
ServerSocket mit ClientIP/ClientPort Adressierung
MulticastSocket neu hinzugefügt (erlaubt Empfangen von z.B. UPNP/SSDP)
Siemens OZW unterstützt den Typ RadioButton

## VERWALTUNGSKONSOLE

Upgrade auf neuste Delphi Version (Mit korrekter Unicode Unterstützung)
Verbesserter Skripteditor
Suchen & Ersetzen wird direkt im Editor angezeigt
Besseres Highlighting bei der Suche
Code Folding
Nach Querverweisen suchen (Im Kontextmenü eines Objekts und im Skript-Editor)
Konsistentere Icon-Auswahl
Assistent zum Validieren auf UTF-8 Konformität (über Utils Control) (Bitte Settings + Skripte sichern!)

## WEBFRONT

Login-Seite kann einen Skin laden (über Skin Control einstellbar) (Beispiel auf GitHub )
Profile mit Schrittweite werden als Circle Slider (wie in den mobilen Apps) dargestellt
Profile mit sehr vielen Aufzählungen werden reduziert und können über einen Dialog eingestellt werden
Dialoge für UnixTimestamp, UnixTimestampDate, UnixTimestampTime, sofern Aktionsskript definiert ist

WFC_OpenCategory($id);

## PHP MODULE

Auswahl von Branches im Module Control
Anzeige der Repository URL im Module Control
Übersetzung von PHP-Module (Beispiel auf GitHub )
Dynamische Konfigurationsformulare (Beispiel auf GitHub )
Liste der verfügbaren Nachrichten im MessageSink
$this->MessageSink($TimeStamp, $SenderID, $Message, $Data); (Beispiel auf GitHub )
$this->SetSummary($Summary);
$this->RegisterMessage($SenderID, $Message); (Beispiel auf GitHub )
$this->UnregisterMessage($SenderID, $Message); (Beispiel auf GitHub )
$this->SetBuffer($Name, $Data); (Beispiel auf GitHub )
$this->GetBuffer($Name); (Beispiel auf GitHub )
$this->SetReceiveDataFilter($RequiredRegexMatch); (Beispiel auf GitHub )
$this->SetForwardDataFilter($RequiredRegexMatch);
$this->Destroy();
MC_ReloadModule($module_control_id, "ModulName"); (Lädt das PHP Modul neu und erstellt auch die Instanzen neu)

## CHANGELOG

Neu: OAuth Control für die neue Symcon Connect + OAuth2 Brücke
Neu: Ein Klick auf die rote "Verbindung verloren" Meldung startet die Konsole neu und verbindet automatisch
Neu: PHP Module können direkt Hook/OAuth Daten empfangen
Neu: Konsole maskiert Kennwörter im Module Control
Neu: Z-Wave unterstützt korrekt NOTIFICATION V4
Neu: Z-Wave Optimierung kann im Gateway für alle Geräte und zusätzlich für jedes Gerät einzeln konfiguriert werden
Neu: Z-Wave: Im Konfigurator kann das Einlernen ohne SECURITY erzwungen werden
Neu: Konsole überprüft, ob mindestens Wine 1.8 genutzt wird
Neu: Z-Wave Parameterdaten werden online abgefragt, anstatt auf die Offline Daten zuzugreifen. (Dadurch ist die ZWDB täglich aktuell)
Neu: Clickatell Modul für neue REST API
Neu: EnOcean EEP A5-04-02
Neu: EnOcean EEP A5-04-03
Neu: Mehr Debug Informationen, wenn SendNonce bei Z-Wave fehlgeschlagen ist
Neu: Profil und Aktion wird für HMIP-eTRV korrekt gesetzt
Neu: WebServer unterstützt Cookies (z.B. Adminer läuft dadurch im /user/ Ordner korrekt)
Fix: Sortierung der Info Spalte ist möglich
Fix: Variable ändern Dialog vergrößert das Eingabefeld mit
Fix: Korrekte Erkennung von abgesteckten Serial Ports (z.B. USB)
Fix: Datum wird bei SMTP E-Mail-Versand korrekt mitgesendet
Fix: Arbeitsverzeichnis ist nun auch bei Skript-Ausführung von Ereignissen korrekt
Fix: DiscoveryServer sollte nicht mehr die set_option Probleme bemängeln
Fix: OnClick bei Formularbuttons konnte keine Selects auswerten (Beispiel: GitHub - symcon/SymconLJ: Lingg&Janke KNX quick)

Fix: Sofern ganz bestimmte ungültige UTF-8 Zeichenketten in String Variablen gespeichert wurden, konnte die Konsole nicht mehr geöffnet werden
Fix: Sofern beim Starten von IP-Symcon keine Internet Verbindung vorhanden ist, markiert das Connect Control dies korrekt als Fehler und versucht eine neue Verbindung aufzubauen.
Fix: WebFront zeigt überlange String korrekt mit "..." an
Fix: WebFront nutzt für Selector/Circle im CSS keine !important Attribute, um besser Skins zu erlauben (Beispiel)
Fix: Cursor im Editor springt bei Mausklick zum Zeilenende
Fix: Hohe-CPU-Last der Konsole verbessert (ist noch nicht perfekt!)
Fix: Wine Version Überprüfung erkennt auch korrekt RC Versionen
Fix: In Ereignissen können wieder Fließkommawerte bei "Bei Wert" eingestellt werden
Fix: Für einige Aktionen darf nur ein Element im Baum ausgewählt sein
Fix: EnOcean TCM120 Gateways wurden seit dem letzten Update nicht korrekt ausgewertet
Fix: LCN Load/Save Scene Befehle waren seit 4.x defekt
Fix: Eltako FWS61 Modul berechnet korrekt die Nachkommastellen
Fix: UnixTimeStampDate/UnixTimeStampTime wird auch in GetValueFormatted unterstützt
Fix: digitalStrom Module haben sich nicht an Sendelimits gehalten (max 1 Request/Sekunde)
Fix: UVR1611 hat für einige Sensortypen den Wert falsch berechnet
Fix: Beim Ausführen von Skripten das Timeout entfernt (wie bei der 3.4/4.0)
Fix: Beim Ausführen von Skripten den "Es wird ausgeführt" Dialog wiederhergestellt
Fix: Übersetzungsfehler im Skript-Editor beim Ersetzen korrigiert
Fix: Vorhandene Enums wurden beim Hinzufügen von Funktionen bei Ereignissen nicht angezeigt
Fix: Im Meldungsfenster bringt ein Doppelklick auf Einträge von gelöschten Objekte eine korrekte Fehlermeldung
Fix: Abbrüche beim HomeMatic Rückkanal sollten nun zu 100% erkannt werden (Nur CCU2)
Fix: Absturz unter nicht Windows-Systemen bei Zeitumstellung korrigiert
Fix: Absturz beim Setzen von Variablen-Profilen, wenn die Groß-/Kleinschreibung falsch war
Fix: Verzögerung beim Starten bzgl. DiscoveryServer korrigiert
Fix: WebFront: Rundungsfehler
Fix: EnOcean Eltako Dimmer sendet korrektes Lern-Telegramm
Fix: WebFront Retro/Mobile hat manchmal einen Redeclare Fehler angezeigt
Fix: AccessViolation in der Konsole, wenn ein Tab ohne Fokus geschlossen wurde und dann wieder geöffnet wurde
Fix: Importieren von Skripten funktioniert im Editor wieder
Fix: Editieren durch langsamen Doppelklick geht in der Baumansicht wieder
Fix: Bei Redo/Undo wurde der Speichern-Button manchmal fälschlicherweise ausgegraut, obwohl Änderungen vorhanden waren
Fix: Der Z-Wave Testen Dialog hat die WAKE_UP Klasse bei SECURITY Geräten nicht ausgewertet.
Fix: Das Connect Control hat sich selber zu schnell deaktiviert bei Fehlern
Fix: Image Grabber konnte Dateityp nicht automatisch erkennen, wenn der Charset mitgesendet wurde
Fix: Bei einigen falsch kodierten Strings (nicht UTF-8 kompatibel) konnten die Konsole den Snapshot nicht laden
Fix: Visual Studio 2015 Redistributable wurde nicht in der neusten Version installiert
Fix: Bei PHP Modulen wurden JSON Daten (SendData*) nicht korrekt einbezogen, welche ein Array waren

Fix: Anpassung des Working Dir für Ereignisse, sollte nicht bei WebServer-Ausführung greifen

# V3.4->V4.0 (Q1/2016)

Um die Migration auf 4.0 oder neuer durchführen zu können ist ein Upgrade auf die letzte IP-Symcon 3.4 Version (Stand: 17.07.2016) erforderlich. Erst danach kann das Upgrade auf 4.0 oder neuer ausgeführt werden:

JETZT IP-SYMCON 3.4 (SETUP) FÜR WINDOWS HERUNTERLADEN



Sofern das Setup nicht korrekt installiert werden kann ist es möglich eine vorhandene Installation manuell auf die 3.4 zu aktualisieren. Dazu muss der IP-Symcon Dienst benendet werden und alle Dateien im IP-Symcon Order mit denen aus der ZIP-Datei überschrieben werden.

JETZT IP-SYMCON 3.4 (ZIP) FÜR WINDOWS HERUNTERLADEN

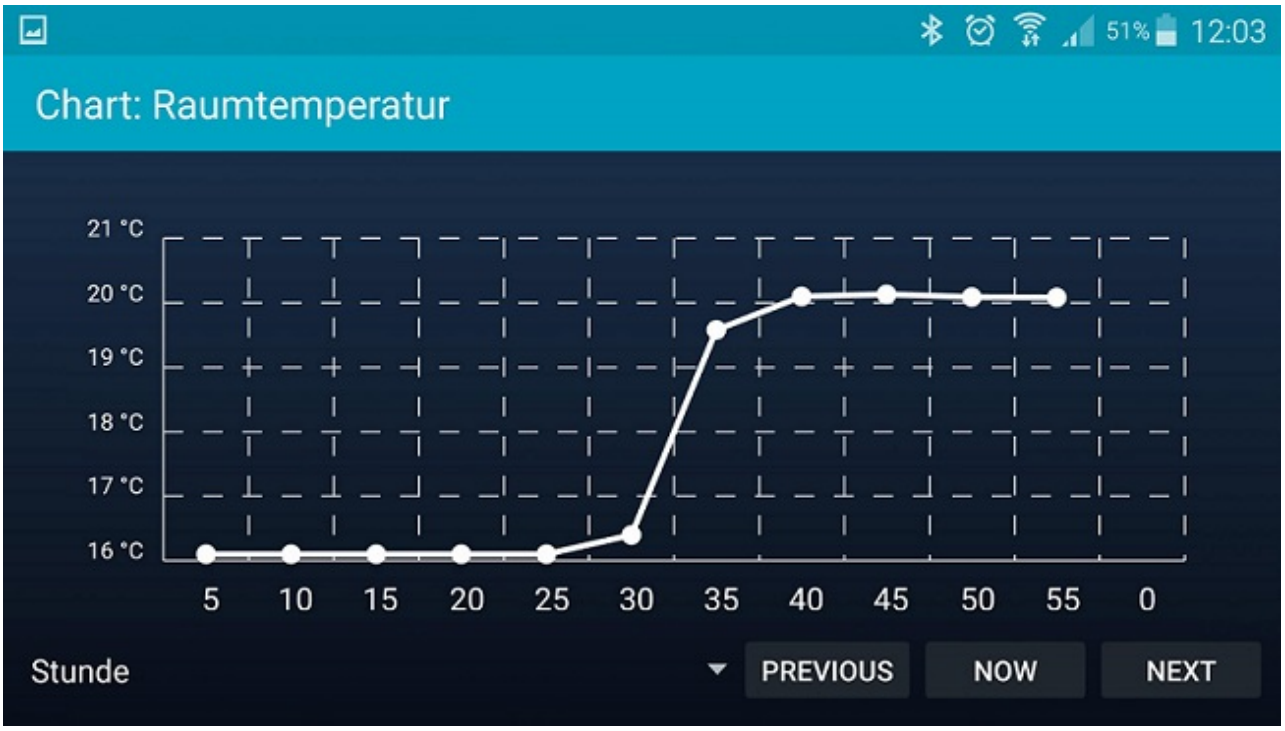
## WESENTLICHE ÄNDERUNGEN

Für IP-Symcon 4.0 wurde insbesondere Wert auf die Unterstützung unterschiedlicher Betriebssysteme gelegt. Dadurch steht ab sofort der IP-Symcon Dienst für den Raspberry Pi, Ubuntu, MacOS und Windows zur Verfügung. Der Einstieg in die Gebäudeautomation wird somit effizienter, flexibler und kostengünstiger.
IP-Symcon 4.0 enthält zudem ein SDK auf PHP Basis. Dadurch kann die Community plattformunabhängige Module bereitstellen, welche jeder verwenden und einfach in seine Hausautomation installieren kann. Bis heute sind bereits über 50 PHP-Module z.B. für die Integration von SONOS, Philips Hue oder Netatmo entstanden.

**Weitere neue Funktionen im Überblick**

- Neues Datenbankformat mit bis zu 70% Platzeinsparung. Dank dem einfachen CSV-Format gibt es bessere Möglichkeiten beim Backup/Restore und Modifikation der geloggten Daten. Ein einfacher Import in Microsoft Excel ist ebenfalls möglich.
- WebFront Visualisierung mit fortlaufenden Graphen, deaktivierbaren Objekten, Popup Instanzen, neuen Icons mit Zuständen uvm.
- Mobile Graphen/Charts in den mobilen Apps
- Verwaltungskonsole wurde optisch für Windows 10 überarbeitet und mit vielen neuen Details verbessert.
- Symcon Connect erlaubt die kinderleichte Verbindung zwischen IP-Symcon und der mobile App ohne extra den Router zu konfigurieren. Zum Einrichten genügt das Abscannen eines QR-Codes.
- Neue Module für astronomische Ereignisse , Kalendereinträge , WebHooks , 1-Wire ohne TMEX, RGB/RGBW für KNX, native Anbindung von Siemens S7 Ethernet, verbesserte Zuverlässigkeit und Funktionalität bei Z-Wave, Unterstützung weiterer EnOcean Eltako

Geräte , uvm.
*(Um den WebFront-Editor nutzen zu können ist mindestens eine IP-Symcon Professional Edition nötig. Alle anderen Funktionen sind in der Basic, Professional und Unlimited Edition von IP-Symcon verfügbar und können mit einer gültigen Subskription bezogen werden. Ein spezielles "Upgrade" muss nicht erworben werden. Sofern Ihre Subskription abgelaufen ist, können Sie diese einfach erneuern und direkt ohne zusätzliche Mehrkosten auf die Version 4.0 updaten. Bitte beachten Sie, dass das Nutzen der Push-Benachrichtigungen und dem Symcon Connect Dienst eine Verbindung zum Internet und eine gültige Subskription voraussetzt. (Shop ))*




## MIGRATION

Bitte erstellen Sie auf jeden Fall ein Backup Ihres gesamten IP-Symcon Ordners, um auf die Version 3.4 zurück zu wechseln, sollten wider erwarten Probleme mit der Version 4.0 auftreten.

Falls Sie von der Version 3.3 updaten, lesen Sie bitte vorher die Migrationsanleitung zur Version 3.4: Migration auf Version 3.4 .

Kritisch: PHP nutzt jetzt bei Ausgaben das lokale Dezimaltrennzeichen (Für DE/AT/CH: 3.4 -> Punkt. 4.0 -> Komma)
Objektnamen (Strings) werden als UTF-8 gespeichert, sodass z.B. Skripte mit Umlauten welche IPS_GetObjectIdByName nutzen ins Leere laufen können. Es wird empfohlen auf die neuere IPS_GetObjectIdByIdent Funktion umzusteigen, die keine Umlaute enthält und besser für den gewünschten Zweck geeignet ist.
Kleinere Änderung bei IPS_GetVariable (VariableType anstatt VariableValue.ValueType)
Kleinere Änderung bei IPS_GetScript (ScriptIsBroken anstatt IsBroken, ScriptExecuted

anstatt LastExecute)
Kleinere Änderung bei IPS_GetLink (TargetID anstatt LinkChildID)
Kleinere Änderung bei IPS_GetMedia (MediaUpdated anstatt LastUpdate, MediaIsAvailable anstatt IsAvailable, IsLinked/SendEvent entfällt)
Kleinere Änderung bei Sys_GetHardDiskInfo (Indexed Array anstatt HDD0 ... HDDn)
Kleinere Änderung bei Sys_GetCPUInfo liefert nun die aktuelle Durchschnittlast, nicht den Mittelwert der letzten 60 Sekunden
SMS-Modul benötigt APIID der HTTP/S API anstatt der SOAP API
Siemens SPS Module basieren von der Idee auf Snap7 und unterstützt ausschließlich Ethernet basierte PLCs
FTDI/USBXpress Instanzen werden zu SerialPort Instanzen konvertiert und müssen ggf. neu konfiguriert werden
Bei FHT_* Befehlen wird eine Fehlermeldung erzeugt, wenn der FHT Buffer im FHZ Splitter aufgebraucht wurde
SOAP Server nicht vorhanden. Am besten gleich auf JSON-RPC wechseln
PHP 5.6 unterstützt Windows XP/Windows 2003/Windows Home Server 2007 nicht mehr. IP-Symcon somit auch nicht.
Image Grabber nutzt die neue Cache Funktion für Medien, und speichert diese nur noch beim Beenden auf die Festplatte/Flash (Dadurch wird die Lebensdauer des Flash-Speichers enorm verbessert!) Ggf. auf IPS_GetMediaContent umstellen.
WebServer unterstützen keine eigenen Ordner mehr. Es wird nur noch webfront/user Ordner unterstützt.
Der "media" Ordner wird nicht mehr überwacht. d.h. neu hinzugefügte oder geänderte Dateien werden nicht bemerkt. Sofern neue Medien über IP-Symcon eingelesen/ aktualisiert werden, kann IP-Symcon über eine Veränderung mit IPS_SendMediaEvent aufmerksam gemacht werden. Bei zyklischen Abfragen, die nicht über den ImageGrabber von extern erfolgen, kann ein Skript, welches zyklisch aufgerufen wird, ebenfalls die IPS_SendMediaEvent Funktion aufrufen.
Für Neuinstallationen wird kein WebServer für Port 82 erstellt. Der interne WebServer ist jetzt unter Port 3777 erreichbar!
Die Datenbank wird während der Installation in ein neues Format konvertiert. Dabei werden die Nachkommastellen mit einer Genauigkeit von 6 Nachkommastellen importiert. Sofern die volle Genauigkeit von 15 Nachkommastellen beim Importieren gewünscht wird (höhere Datenmenge), kann dazu das externe DBConvert Tool genutzt werden.
Einige Module sind entfallen: ISDN, OneWire DS2490, Siemens S7 MPI/PPI, MF420-IR-CTF Klimawächter, EZControl, RFReader, CGEM24
Externe Module/Delphi SDK (*.dll) aus IP-Symcon 3.4 sind nicht mehr kompatibel.
Nach der Installation muss im Archiv unter Kern Instanzen eine Re-Aggregation gestartet werden, damit alle historischen Daten wieder korrekt sichtbar sind.
Skripte in Wochenplan-Aktionen müssen sauber mit einem Semikolon (;) beendet sein.

## LISTE DER ÄNDERUNGEN

### DIENST IST FÜR VERSCHIEDENE PLATTFORMEN VERFÜGBAR

Verfügbar für Windows (ab Vista), RaspberryPi (alle Modelle), Linux (Ubuntu 14.04 x64), MacOS (ab 10.8)

## WEBFRONT ENTHÄLT FOLGENDE NEUE FUNKTIONEN

Fortlaufende Graphen (Link zum Wunsch)
Deaktivierbare Objekte (Link zum Wunsch)
Öffnen von eingebetteten Graphen im Popup (Wurde auch gewünscht)
Voreinstellungen für Graphen (Link zum Wunsch)
Popup-Instanzen (Link zum Wunsch)
Variablenprofil CSS-Hooks zur besseren Individualisierbarkeit über Skins (Link zum Wunsch)
Vereinfachte Ansicht für ~HexColor Profil (Link zum Wunsch)
Neue Vektorbasierte-Icons für noch bessere Retina Kompatibilität (RetroSkin für die alten Icons)
Adaptive Icons (Icons mit Statuszuständen), welche bisher für folgende Icons realisiert wurden: Clock, Garage, Hourglas, Jalousie, Light, Lock, Power, Speaker, Speedo, Temperature, WindSpeed und Window
WebFront Retro optimiert, sodass Blinde dies über JAWS besser bedienen können

## VERWALTUNGSKONSOLE ENTHÄLT FOLGENDE NEUE FUNKTIONEN

Optisch für das Windows 10 Design überarbeitet
Beim "Objekte bearbeiten" Dialog ist die Unterstützung für Aktiv/Inaktiv hinzugekommen
Im WebFront Konfigurator können die Voreinstellungen für Graphdialoge konfiguriert werden
"Suche ID" im Objektbaum enthält die ID aus der Zwischenablage falls vorhanden
Im Debug für Instanzen werden Steuerzeichen entsprechend kodiert. z.B.
Im Debug werden beim Speichern in eine Datei Text und HEX exportiert
Icons für Skripte, Variablen und Wochenpläne getauscht
Alle "Vollbild" Dialog haben nun eine obere Toolleiste mit den relevanten Optionen (Mehr Konsistenz in der Bedienung)
Ereignis/Statusvariablen/Debugfenster werden beim Editieren von Instanzen in eigenen, dockbaren Fenstern geöffnet
Spezialschalter sind aus der Konsole änderbar
Sofern keine Lizenz vorhanden ist, wird diese beim Öffnen der Konsole abgefragt
"Ansicht hinzufügen" ist nun auf der Willkommensseite ganz unten unter "Expertenansichten hinzufügen" platziert
Dokumentation für "Symcon Connect" und "Keine Server gefunden" im Verbindungsassistenten verknüpft
Reaggregation kann im Archiv bequem für mehrere Variablen in Auftrag gegeben werden

## PERFORMANTERES ARCHIV

Die Daten im CSV-Format verbrauchen ca 70% weniger Speicherplatz
Vorteile beim Backup, Restore und Löschen
Hinzufügen von eigenen Datensätzen ist möglich
Öffnen der CSV-Dateien mit Microsoft Excel möglich

## SONSTIGES

Symcon Connect Dienst erspart lästige DynDNS/Routerkonfiguration (benötigt gültige

Subskription) und erhöht über die standardmäßig aktivierte SSL-Verschlüsselung die Sicherheit
Neues Modul für Astronomische Ereignisse
Neues Modul für Kalendereinträge
Neues Modul für WebHooks
Native PHP Erweiterungen (Übersicht der verfügbaren PHP Module )
Native 1-Wire Module (Kein TMEX Treiber wird mehr benötigt)
1-Wire DS2490/DS9490R (blauer USB-Stick). Dieser benötigt noch immer die TMEX Treiber.
KNX RGB/RGBW Modul
KNX Shutter unterstützt Rückmeldegruppenadressen
digitalStrom Konfigurator kann komfortabel Räume anlegen
digitalStrom Klemmen können über DS_RequestStatus nach aktuellem Zustand abgefragt werden
DMX4ALL Gateway wird nun auch über TCP unterstützt
PHP 5.6.17

## ENTFALLENE MODULE

ISDN
Siemens S7 MPI/PPI (Es werden nur noch Ethernet PLCs unterstützt)
MF420-IR-CTF Klimawächter, EZControl, RFReader, CGEM24 (Nur noch als PHP-Module: symcon/SymconBC Â· GitHub)
Alle externen Module der Version 3.x sind nicht mehr mit 4.0 verwendbar. Alternativ gibt es nun die platformunabhängigen PHP-Module.

# V3.3->V3.4 (Q2/2015)

## WESENTLICHE ÄNDERUNGEN

**Verbesserte Nutzbarkeit bei batteriebetriebenen Z-Wave-Geräten**
Die Unterstützung von batteriebetriebenen Geräten wurde wesentlich verbessert, sodass die Einrichtung und Veränderung von Einstellungen leichter möglich ist. Intern wurde die Warteschleife für Nachrichten verbessert, sodass keine gleichen, doppelten Anfragen an das Gerät gesendet werden, wodruch die Lebensdauer der Batterie profitiert. Zusätzlich kann das Abfrageintervall in der Instanz definiert werden, wodurch weitere Einsparungen im Batterieverbrauch möglich sind.
**Benutzerdefinierte Sounds können an mobile Apps gesendet werden**
Push-Benachrichtigungen der mobilen Apps können mit einem benutzerdefinierten Sound ausgeben werden. Ein Blick auf das Smartphone kann somit oftmals entfallen. Weitere Informationen zu den Verfügbaren Sounds sind sind in unserer Dokumentation unter WFC_PushNotification zu finden.
**ALLNET Module aktualisiert**
Die ALLNET Module bevorzugen ab IP-Symcon 3.4 die neue JSON Schnittstelle der Geräte, welche ab Firmware 3.0 angeboten wird. Dadurch wird die Anbindung der Geräte nochmals vereinfacht. Eine aktuelle Liste der unterstützten Geräten finden Sie unter ALLNET .
*(Um den WebFront-Editor nutzen zu können ist mindestens eine IP-Symcon Professional Edition nötig. Alle anderen Funktionen sind in der Basic, Professional und Unlimited Edition von IP-Symcon verfügbar und können mit einer gültigen Subskription bezogen werden. Ein spezielles "Upgrade" muss nicht erworben werden. Sofern Ihre Subskription abgelaufen ist, können Sie diese einfach erneuern und direkt ohne zusätzliche Mehrkosten auf die Version 3.4 updaten. Bitte beachten Sie, dass das Nutzen der Push-Benachrichtigungen eine Verbindung zum Internet und eine gültige Subskription voraussetzt. (Shop ))*

## MIGRATION

Bitte erstellen Sie auf jeden Fall ein Backup Ihres gesamten IP-Symcon Ordners, um auf die Version 3.3 zurück zu wechseln, sollten wider erwarten Probleme mit der Version 3.4 auftreten.

Falls Sie von der Version 3.2 updaten, lesen Sie bitte vorher die Migrationsanleitung zur Version 3.3: Migration auf Version 3.3 .

Sofern batteriebetriebene Z-Wave-Geräte genutzt werden, muss nach dem Update das Abfrageintervall explizit angegeben werden! Andernfalls wird der Status nicht vom Gerät abgefragt, sondern nur die vom Gerät gesendeten Werte aktualisiert.

## LISTE DER ÄNDERUNGEN

Neue Funktionen

-Neu: Verbesserte Nutzbarkeit bei batteriebetriebenen Z-Wave-Geräten
-Neu: Digitale Signaturen (Löst Probleme mit Fehlalarmen in Virenscannern)
-Neu: Im ModBus Modul kann ein Umrechnungsfaktor angegeben werden
-Neu: Im Browser wird der Titel des WebFront in der Titelleiste angezeigt
-Neu: xComfort Konfigurator kann die neuen Geräte verwalten (DeviceID 74)
-Neu: Real64 Datentyp in ModBus hinzugefügt
-Neu: Sound kann an mobile Apps gesendet werden (App-Updates werden benötigt...)
-Neu: Push-Nachrichten können nun 256 Zeichen lang sein (plus zusätzlich 32 Zeichen im Titel) (Erfordert ggf. iOS8)
-Neu: OneWire Module wurden aufgeteilt, um kompatibel zur Linux, RaspberryPi und MacOS Version zu sein
-Neu: Animationen für Graphen können im WebFront über Skins deaktiviert werden
-Neu: ALLUniversal Modul umbenannt zu "alte Firmware 2.x"
-Neu: ALL5000 Modul umbenannt zu ALLUniversal (für Firmware 3.x)

Behobene Fehler

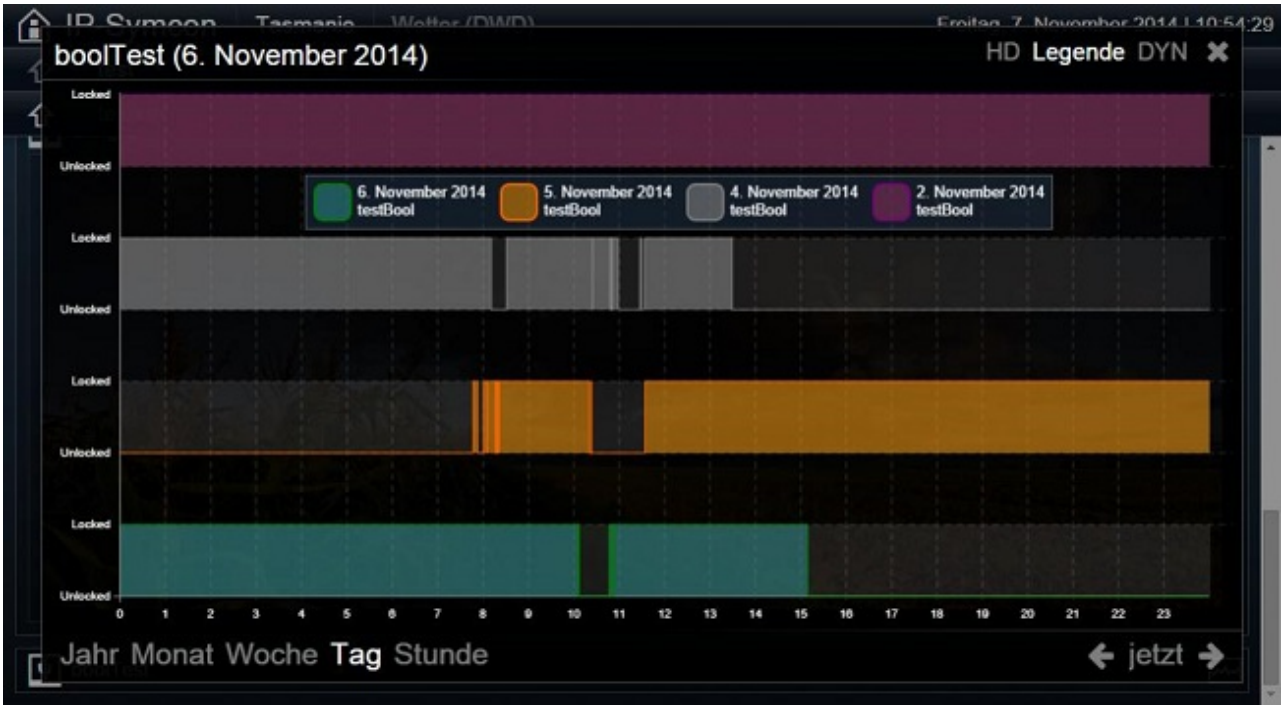
-Fix: Beim Wochenplan wurde unter bestimmten Umständen die Eingabe von Quellcode nicht erlaubt
-Fix: Problem mit Zwischenablage beim Kopieren von ObjektIDs
-Fix: JSON-RPC Klasse wirft nun Exceptions, damit diese korrekt abgefangen werden können
-Fix: Fehlende 1-Wire Kompatibilitätsfunktionen ergänzt
-Fix: Geänderte Idents bei den Modulen werden bei IPS_GetStatusVariableID berücksichtigt
-Fix: Link zum Shop springt direkt zu den Subskriptions
-Fix: Datum wird bei Boolean-Graphen jetzt immer korrekt angezeigt
-Fix: Velleman Board Modul sollte mit VM110N laufen
-Fix: Z-Wave Parameter Dialog überarbeitet
-Fix: WinLIRC kommt mit fragmentierten Paketen zurecht
-Fix: LCN Host Kommando wird korrekt ausgewertet
-Fix: Shortcuts gehen im Objektbaum Popup korrekterweise nicht mehr
-Fix: Ereignisse, die per PHP erstellt werden, werden korrekt als einmalige Ereignisse erstellt
-Fix: Vollbild-Boolean-Charts wurden nicht angezeigt, wenn Extrema aktiviert wurden
-Fix: Kleines Usability-Problem beim Erstellen von Wochenplanaktionen im WebFront gelöst
-Fix: Problem mit EventLimit bei Ereignissen gelöst
-Fix: IPS_VariableProfileExists hat sich bei der Prüfung inkonsistent zu IPS_CreateVariableProfile verhalten
-Fix: Problem mit Symcon-Tray nach Windows-Start gelöst
-Fix: Access.log schreibt bei Vorhandensein die X-Forwarded-IP in Logfile
-Fix: Problem mit xComfort Rückmeldungen gelöst
-Fix: Problem mit Setzen von DS2405 Pins über PHP gelöst (TMEX_F05_SetPin)

# V3.2->V3.3 (Q1/2015)

## WESENTLICHE ÄNDERUNGEN

### Neuer Graphentyp Bool-Chart für Bool-Variablen

Auf vielfachen Wunsch und aufgrund hoher Funktionalität ermöglicht IP-Symcon nun die präzise Darstellung von Bool-Variablen innerhalb von Graphen.
Es ist möglich einen Bool-Chart sowohl im Einzelgraphen als auch Multigraphen darzustellen. Es werden ausschließlich die Rohdaten genutzt, wodurch eindeutige Flanken in den Graphen dargestellt werden. Dies führt zu einer präzisen Veranschaulichung der aufgenommen Daten. Eine weitere Besonderheit zum bisherigen Linien- oder Bar-Multi-Chart ist die Nutzung des jeweils eigenen Profils der Bool-Variablen innerhalb eines Multi-Charts. Dies führt zu einer hohen Übersichtlichkeit.




### Unterstützung für neue EnOcean Geräte und EnOcean EEPs

Neben neuen Geräten wie z.B. Eltako FWS61, Eltako FSB14, Eltako FSB61, Eltako FSB71 können nun auch eine Vielzahl von Geräten eingebunden werden, welche die offiziellen "EnOcean Equipment Profiles" (kurz EEPs) unterstützen. Dadurch sind fast alle neuen EnOcean Geräte mit IP-Symcon kompatibel sein. Zusätzlich wurde auch die Unterstützung für BiDirektionale Geräte, wie zum Beispiel Stellantriebe verbessert. Somit können die Geräte SAB02 von "Thermokon", B2 Drive von "alphaEOS" und das MD15 von "Kieback & Peter" nun genutzt werden. Eine vollständige Liste der aktuell unterstützen Geräte und EEPs kann der EnOcean Geräteliste entnommen werden.

### Verbesserungen der Verwaltungskonsole

Zur Einrichtung von IP-Symcon ist die Verwaltungskonsole besonders wichtig. Aus diesem Grund wurde in diesem Update ein besonderes Augenmerk auf dessen Anwenderfreundlichkeit gelegt. Es wurden, um eine Einrichtung und Bedienung zu beschleunigen, viele Tastenkombinationen hinzugefügt. Beispielsweise ist nun das Hinzufügen, Duplizieren und Verlinken von Objekten mit einem einfachen Tastendruck möglich. Durch das Update des Schnellfilters ist die integrierte Suche im Objektbaum noch einfacher zu bedienen. Damit die Konsole sich konsistent verhält, lassen sich schnell alle Dialoge per ESC schließen und die Dialogfelder per Tabulator durchschalten.

*(Um den WebFront-Editor nutzen zu können ist mindestens eine IP-Symcon Professional Edition nötig. Alle anderen Funktionen sind in der Basic, Professional und Unlimited Edition von IP-Symcon verfügbar und können mit einer gültigen Subskription bezogen werden. Ein spezielles "Upgrade" muss nicht erworben werden. Sofern Ihre Subskription abgelaufen ist, können Sie diese einfach erneuern und direkt ohne zusätzliche Mehrkosten auf die Version 3.3 updaten. Bitte beachten Sie, dass das Nutzen der Push-Benachrichtigungen eine Verbindung zum Internet und eine gültige Subskription voraussetzt. (Shop ))*

## MIGRATION

Bitte erstellen Sie auf jeden Fall ein Backup Ihres gesamten IP-Symcon Ordners, um auf die Version 3.2 zurück zu wechseln, sollten wider erwarten Probleme mit der Version 3.3 auftreten.

Falls Sie von der Version 3.1 updaten, lesen Sie bitte vorher die Migrationsanleitung zur Version 3.2: Migration auf Version 3.2 .

## LISTE DER ÄNDERUNGEN

Neue Funktionen
-Neu: Boolean-Multigraphen (siehe Screenshot )
-Neu: Boolean-Variablen werden in Graphen besser dargestellt
-Neu: EnOcean EEP Module (Nur Empfangen!)
-Neu: Eine Menge Shortcuts für die logische Baumansicht und den Willkommen-Reiter
-Neu: Duplizieren von Objekten (Noch nicht rekursiv!)
-Neu: "Instanz erstellen" Dialog überarbeitet
-Neu: "Variablen erstellen/editieren" Dialog schlanker gestaltet
-Neu: Reaggregieren wird automatisch nach Änderung des Aggregationstyps gestartet
-Neu: "Objekt bearbeiten" Dialog verbessert
-Neu: QuickFilter in der Baumansicht ist nun Case-Insensitive
-Neu: Alle Dialoge sind korrekt mit ESC schließbar
-Neu: Navigieren per Tab wurde überall korrekt korrigiert
-Neu: Alle Zusammenfassungsdialoge (Summary) entfernt
-Neu: Logos angepasst
-Neu: IPS*GetModuleListByType*
*-Neu: COMPort heißt jetzt SPRT_* und WWWReader** heißt jetzt WWW**
-Neu: Neue Module für EnOcean Enegergiezähler (EEP A5-12-00, A5-12-01, A5-12-02, A5-12-03)
-Neu: Neues Modul für Eltako FWS61
-Neu: Eltako Shutter Modul für FSB14, FSB61, FSB71
-Neu: Bool-Charts können nun unterschiedliche Profile bei Variablen anzeigen
-Neu: Auswahl für ModBus TCP über UDP
-Neu: EnOcean EEP A5-20-01 für z.B. Thermokon SAB02, alphaEOS B2 Drive oder Kieback&Peter MD15
-Neu: EnOcean Modul für alphaEOS SENSE TF-H
-Neu: Debugging verlängert sich automatisch
-Neu: TextParser, HeatingControl, ShutterControl, EventControl auf IPS_SetProperty/

IPS_GetProperty umgestellt
-Neu: Geloggte Strings können über den Graphdialog angezeigt werden
-Neu: Geloggte String Variablen können auch als Vollbild "Graph" angezeigt werden

### Behobene Fehler

-Fix: Sortierung im WFC "Element hinzufügen" Dialog
-Fix: AccessViolation bei der Timer-Übersicht, wenn InstanzID Null war
-Fix: Probleme mit Eltako FSS12 Modul gelöst
-Fix: Gelöschte, geöffnete Instanzen werden automatisch als Reiter entfernt
-Fix: Neue Konfiguratoren können aus dem "Konfiguratoren verwalten" Dialog erstellt werden
-Fix: QuickFilter merkt sich die letzte Eingabe
-Fix: Bessere Exception bei fehlerhaften JSONRPC Requests
-Fix: Behebt Sicherheitslücke in Verbindung mit Proxies/ReverseProxies
-Fix: Behebt Fehlermeldung bei JSONRPC Klasse
-Fix: Fehlerhafter Punkt bei Float-Werten im WebFront bei englischsprachigen Systemen
-Fix: Probleme mit leeren HD-Graphen
-Fix: Ventilprosition bei FHT wird bei der Entkalkungsfahrt korrekt ausgewertet
-Fix: Z-Wave Protection Klasse wird nun in Version 2 unterstützt
-Fix: Problem mit Z-Wave SensorAlarm Klasse gelöst
-Fix: Profilmanager kann besser mit Pfeiltasten bedient werden
-Fix: Lade-Dialog ist nun mittig positioniert
-Fix: Fehlende $_IPS Variablen im Skript Editor hinzugefügt
-Fix: Problem mit nicht angezeigten, ersten Kategorien im WebFront korrigiert
-Fix: Kleine Usability-Probleme mit Wochenplan korrigiert
-Fix: Namensänderungen werden auch in Reitern berücksichtigt
-Fix: Fehler beim Schalten mit dem Shutter Control korrigiert

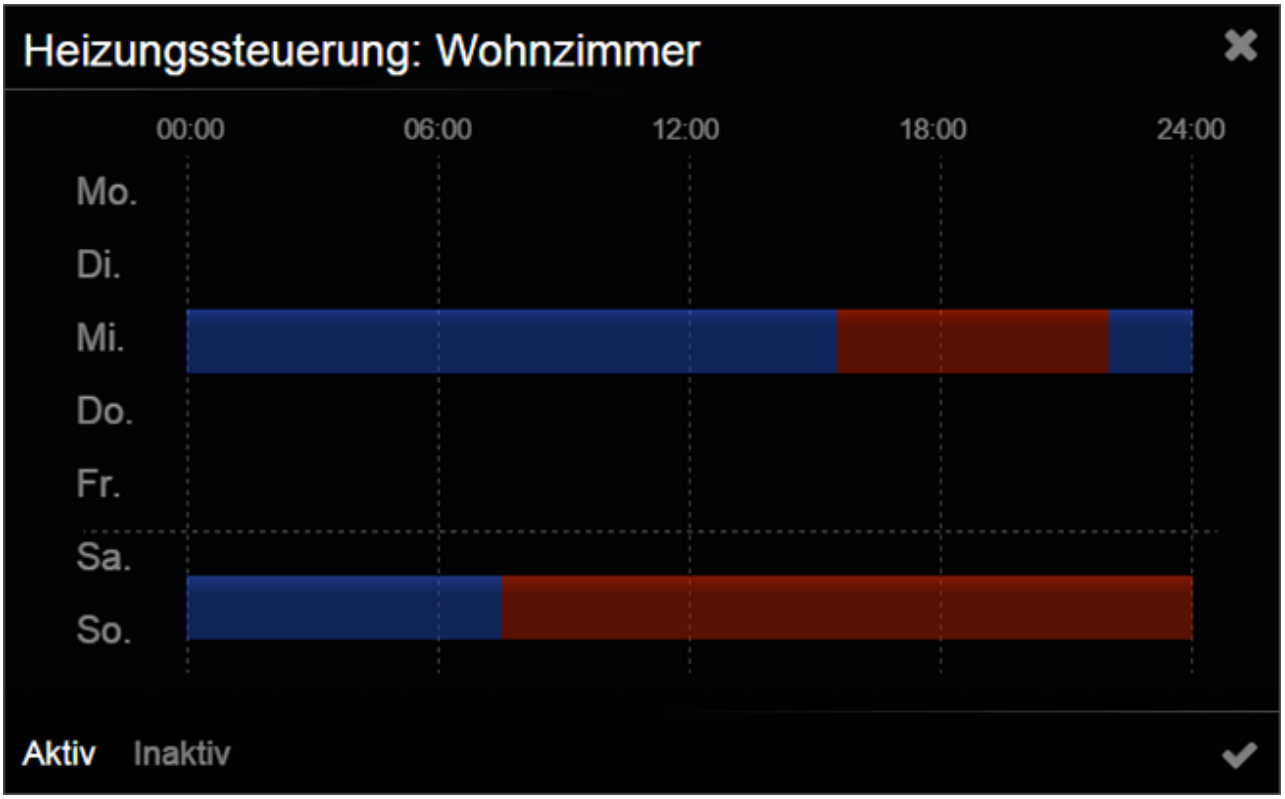
### Änderung Dokumentation

Die Doku befindet sich in einem Status der Generalüberholung und wird stetig aktualisiert.
-Neu: [Schnelleinstieg]
-Neu: Bool Chart
-Neu: Links
-Neu: Gerätelisten (1-Wire , ALLNET , DMX , Eaton xComfort , KNX , EnOcean/EEP , Z-Wave )
-Neu: Befehlsliste vervollständigt (Z-Wave )
-Update: Instanzen
-Update: Ereignisse
-Update: Fernzugriff
-Update: Geräte einbinden
-Update: Links verwenden
-Update: Email Benachrichtigung

# V3.1->V3.2 (Q3/2014)

## WESENTLICHE ÄNDERUNGEN

Neuer Ereignistyp: Wochenplan
Eine der Neuerungen ist der Ereignistyp Wochenplan, mit dem es nun noch schneller möglich ist,
einzelne Zustände z.B. von Licht-Szenarien oder der Heizungssteuerung für jeden Wochentag anzulegen oder zu ändern. Im WebFront werde geplante Zustände durch farbige Balken übersichtlich dargestellt und sind intuitiv anpassbar.




HD für Multi-Charts
Die Multi Charts im WebFront bieten die viel gewünschte HD-Funktion an, mit der es nun möglich ist, die einzelnen Graphen detailliert in hoher Dichte anzuzeigen.
Updates für Systeme
Neue Module für digitalSTROM erlauben das Setzen von Stimmungen direkt aus dem WebFront heraus und endlich den viel gewünschten Rückkanal, der Änderungen der Raumstimmung in IP-Symcon wiederspiegelt. Zusätzlich wurde die Eaton xComfort Unterstützung überarbeitet, sodass die neuen Geräte mit gerouteten Statusmeldungen eingebunden werden können.
Skin-Manager in der Verwaltungskonsole
Der neue Skin-Manager erlaube die bequeme Verwaltung aller Skins und ist nun direkt in der Verwaltungskonsole unter Kern-Instanzen verfügbar.
*(Um den WebFront-Editor nutzen zu können ist mindestens eine IP-Symcon Professional Version nötig. Alle anderen Funktionen sind in der Basic, Professional und Unlimited Version von IP-Symcon verfügbar und können mit einer gültigen Subskription bezogen werden. Ein spezielles "Upgrade" muss nicht erworben werden. Sofern Ihre Subskription abgelaufen ist, können Sie diese einfach erneuern und direkt ohne zusätzliche Mehrkosten auf die Version 3.2 updaten. Bitte beachten Sie, dass das Nutzen der Push-Benachrichtigungen eine Verbindung zum Internet und eine gültige Subskription*

*voraussetzt. (Shop ))*

## MIGRATION



Bitte erstellen Sie auf jeden Fall ein Backup Ihres gesamten IP-Symcon Ordners, um auf die Version 3.1 zurück zu wechseln, sollten wider erwarten Probleme mit der Version 3.2 auftreten.



Falls Sie von der Version 3.0 updaten, lesen Sie bitte vorher die Migrationsanleitung zur Version 3.1: Migration auf Version 3.1 .

## LISTE DER ÄNDERUNGEN

Neue Funktionen
- Neu: Wochenplan Ereignisse können erstellt werden
- Neu: IPS_SetEventScheduleAction
- Neu: IPS_SetEventScheduleGroup
- Neu: IPS_SetEventScheduleGroupPoint
- Neu: HD für MultiCharts
- Neu: dSRoom und dSApartment Module für digitalStrom zur Ansteuerung der Stimmungen
- Neu: Skins werden über die Konsole verwaltet
- Neu: ZW_ConfigurationSetValueEx für Z-Wave Parameter mit variabler Größe (IPSD-86)
- Neu: WinLIRC wertet Rückmeldungen aus und wirft somit auch Fehlermeldungen (IPSD-116)
- Neu: WebFront: Inhaltswechsler reagiert auf Änderungen von Links
- Neu: WebFront: Inhaltswechsler reagiert auf Verstecken/Sichtbar machen von Objekten

Behobene Fehler
- Fix: Aus aktuellem Anlass (POODLE) wird SSLv3 nicht mehr erlaubt
- Fix: IPS_SetEventCyclicTimeFrom/IPS_SetEventCyclicTimeTo überprüft das angegebene Datum
- Fix: Icons der WebFront Menüleiste wurden unter iOS7 leicht verschoben angezeigt
- Fix: WebFront: CSV auch im FF und IE auswählbar
- Fix: WebFront: CSV mit besserer Formatierung
- Fix: WebFront Retro/Mobile können korrekt separat aktiviert/deaktiviert werden
- Fix: Link zur Doku innerhalb der Konsole korrigiert
- Fix: Probleme mit Ereignissen bei Zeitzonen ohne Sommerzeit korrigiert
- Fix: Reloads im WebFront durch z.B. ungültige Float-Werte von ModBus Instanzen korrigiert
- Fix: OneWire DS2413 überprüft komplementär Bits (IPSD-111)
- Fix: Archivhandler Ansicht Zeitraum Auswahl fehlerhaft (IPSD-38)
- Fix: WebFront-Konfigurator überschreibt existierendes Objekt mit gleichem Namen (IPSD-121)
- Fix: Beim Wechsel der Listenansicht des Objektbaums wird ein gesetzter Filter nicht berücksichtigt (IPSD-123)
- Fix: Unnötige Scrollbar bei Inhaltswechslern ohne Navigationsleiste (WEBFRONT-58)
- Fix: Name beim Duplizieren von Profilen werden auf korrektheit überprüft
- Fix: DMX hat ggf. nach 49,7 Tagen ohne Windows Neustart nicht mehr korrekt funktioniert

Fix: Kleine Änderung die bei KNX evtl. die "Not connected to interface" Probleme löst

# V3.0->V3.1 (Q2/2014)

## WESENTLICHE ÄNDERUNGEN

Multi-Charts
Eine der Neuerungen sind die Multi Charts , mit denen es nun einfach möglich ist mehrere Graphen in einer Ansicht darzustellen. Es können so z.B. die Verbrauchskosten von dem laufendem Jahr mit dem des Vorjahrs graphisch gegenübergestellt werden, um so schnell Unregelmäßigkeiten zu entdecken. Dabei kann jeweils ein beliebiger Titel, Linien und Füllfarbe ausgewählt werden. Die Charts können nach der Konfiguration direkt im WebFront angesehen oder über das Graph-Element im Vollbild Modus eingebettet werden.
Fernzugriff auf Konfiguration mit Kennwortschutz
Eine weitere neue Funktion ist der Fernzugriff. Mit ihm ist es nun leicht möglich die Verwaltungskonsole von einem anderen Rechner aus zu nutzen. So lassen sich von überall Änderungen der Konfiguration zu Hause oder beim Kunden durchführen. Durch den neuen, integrierten Kennwortschutz ist dies auch über das Internet sicher möglich.
Komfortabler Profilmanager
Variablenprofile können nun noch schneller und einfacher mit Hilfe des Profil-Manager bearbeitet werden. Dabei ist dieser in einem eigenem Fenster mit viel Platz ausgestattet worden. Änderungen an den Assoziationen sind nun direkt möglich ohne ein zusätzliches Fenster öffnen zu müssen. Die Sortierung nach Variablentyp vereinfacht die Suche nach dem Profil. Zusätzlich können Profile schnell dupliziert und deren Nutzung in den jeweiligen Variablen untersucht werden.
Updates für Systeme
Des Weiteren wurde viele Module überarbeitet. So wurden z.B bei LCN neue Umrechnungsfaktoren, die Unterstützung des S0-Counter vom BU4L hinzugefügt oder bei diversen Z-Wave Geräten die Kompatibilität verbessert. Alle Details können der Liste der Änderungen entnommen werden.
*(Um den WebFront-Editor nutzen zu können ist mindestens eine IP-Symcon Professional Version nötig. Alle anderen Funktionen sind in der Basic, Professional und Unlimited Version von IP-Symcon verfügbar und können mit einer gültigen Subskription bezogen werden. Ein spezielles "Upgrade" muss nicht erworben werden. Sofern Ihre Subskription abgelaufen ist, können Sie diese einfach erneuern und direkt ohne zusätzliche Mehrkosten auf die Version 3.1 updaten. Bitte beachten Sie, dass das Nutzen der Push-Benachrichtigungen eine Verbindung zum Internet und eine gültige Subskription voraussetzt. (Shop ))*

## MIGRATION

Bitte erstellen Sie auf jeden Fall ein Backup Ihres gesamten IP-Symcon Ordners, um auf die Version 3.0 zurück zu wechseln, sollten wider erwarten Probleme mit der Version 3.1 auftreten.

Falls Sie von der Version 2.7 updaten, lesen Sie bitte vorher die Migrationsanleitung zur Version 3.0: Migration auf Version 3.0 .

Sofern die Verwaltungskonsole von einem anderen Rechner aus genutzt wird, muss der Fernzugriff aktiviert werden. Fernzugriff aktivieren
Sofern die Verwaltungskonsole von einem anderen Rechner aus genutzt wird, muss diese einmalig erneut aus unserem Download-Bereich geladen werden
Die Verwaltungskonsole kommuniziert über den WebFront Port (82) anstatt dem SOAP Port (3773)
Die JSON-RPC API ist lokal immer aktiv, und ist von extern ausschließlich über das Fernzugriff-Kennwort und eurem Lizenz-Benutzernamen zugreifbar.
SOAP ist standardmäßig deaktiviert, kann jedoch jederzeit aktiviert werden. Spezialschalter
Die Felder CyclicDateFrom/CyclicDateTo/CyclicTimeFrom/CyclicTimeTo von IPS_GetEvent wurden geändert. Bitte die Dokumentation beachten.

## LISTE DER ÄNDERUNGEN

Neue Funktionen
- Neu: Multi-Charts
- Neu: Profile werden über den neuen Profilmananger bearbeitet
- Neu: Konsole arbeitet über die neue JSON-RPC API (=Fernzugriff, Passwortgeschützt)
- Neu: Konsole benötigt für Fernzugriff einen Benutzernamen/Kennwort
- Neu: Proxy Support für Verwaltungskonsole
- Neu: API-Vorbereitungen für neue App-Versionen
- Neu: Automatische Suche von IP-Symcon Servern via UPNP/SSDP
- Neu: Variablenprofil-Einstellungen aufgeräumt
- Neu: Upgrade auf PHP 5.4.24
- Neu: Unterstützung für Farben bei der Werteauswahl (WebFront)
- Neu: Retina-Support für den Farbauswahl-Dialog
- Neu: Unterstützung für umgerechnete Werte beim W&T Counter Modul
- Neu: Startdatum/Enddatum/Startzeit/Endzeit werden über neue Funktionen gesetzt (Alte Funktionen weiterhin vorhanden)
- Neu: LCN - Neue Umrechnungsfaktoren (wie auch in der LCN Pro vorhanden)
- Neu: LCN - S0-Counter des BU4L werden unterstützt
- Neu: LCN - Unterstützung für Ausgang 4
- Neu: Z-Wave - Profile/Namen werden für METER Klasse besser gesetzt
- Neu: Z-Wave - Battery-Low Anzeige für Z-Wave BATTERY Klasse
- Neu: Z-Wave - Konfigurator kann Untergeräte korrekt einsortieren
- Neu: DMX - RGB Modul für bessere WebFront Integration
- Neu: Module für SERVO-868
- Neu: IsCharging, BatteryRemainingTime, BatteryMaxTime für Sys_GetBattery
- Neu: Unterstützung für "letzter Wochentag im Monat und letzter Tag im Monat" bei Ereignissen
- Neu: Graph-Dialog im WebFront zeigt nun nur gültige Variablen/Charts zum auswählen an
- Neu: Sortierung wird im Objektbaum pro Ansicht gespeichert
- Neu: Suchen/Ersetzen in Dateien merkt sich letzte Begriffe
- Neu: Spalten bei Timer Informationen und Ereignis Informationen können sortiert werden
- Neu: Intern alles auf UnixTime umgestellt, anstatt der proprietären TDateTime von Delphi
- Neu: Modul für ALLNET 5000 / ALLNET 3500 (neue 3.x Firmware)
- Neu: Status wird bei ALL3073 ausgelesen
- Neu: Empfang von Push-Nachrichten kann direkt im Konfigurator/Notification-Control aktiviert/deaktiviert werden
- Neu: Transparente Füllfarbe bei Multi-Charts

Neu: dSS-Splitter kann nun auch mit einem alternativen Port umgehen
Neu: DMX_SetRGB, um die RGB Instanz per PHP anzusteuern
Neu: FS20, FS20EX und FHT intern auf IPS_SetProperty/IPS_GetProperty umgestellt
Neu: HID und MediaPlayer Modul auf IPS_GetProperty/IPS_SetProperty umgebaut-Fix: Update auf OpenSSL 1.0.1g
Neu: Status von ALL3075V2 wird eingelesen
Neu: Status von ALL3075 wird eingelesen
Neu: HTTP_X_FORWARDED_FOR wird ausgewertet
Neu: Lizenz kann direkt über die Verwaltungskonsole geändert werden
Neu: DLL vom MediaPlayer aktualisiert
Neu: ModBus unterstüzt 64bit Werte
Neu: Konsole kann mit Parameter /locationview gestartet werden und zeigt sofort die logische Baumansicht
Neu: DMX_FadeRGB und DMX_FadeRGBDelayed (respektive wurde DMX_SetValue zu DMX_SetChannel ect. umbenannt. Die alten Funktion sind weiterhin als Kompatibilitätsfunktionen verfügbar)
Neu: Basis Authentizifierung greift beim WebFront nur noch für den "user"-Ordner (Sicherheit)
Neu: Mobile/Retro Reiter im WebFront Konfigurator umstrukturiert
Neu: Sicherheitshinweis beim Deaktivieren des Mobil-Zugriffs
Neu: Unterstützung für Pool-Sensor bei Oregon Wetterstationen
Neu: WAC_Reset($id) - Befehl um den MediaPlayer bei einem Ausfall wieder zum Leben zu erwecken. Feedback ist erwünscht!
Neu: Module werden im "Über" Dialog nach Namen sortiert

Behobene Fehler

Fix: Probleme mit der Berechnung von Ereignissen korrigiert
Fix: Mehr Abstand beim Speichern Button im Farbkreis
Fix: EIS8 bei KNX hat nur eine 0 gesendet
Fix: Probleme mit IPS_RunScriptEx/IPS_RunScriptWaitEx über JSON-RPC gelöst
Fix: IMAP/POP3 Mails werden korrekt sortiert
Fix: Probleme mit Schrittweiten < 1 behoben
Fix: Bei RGB Slidern kann nun auch die Prozentzahl geklickt werden
Fix: Konsole stürzte ab, wenn Übergeordnete Instanz auf sich selbst gerichtet wurde
Fix: Übergeordnete Instanz kann nicht mehr auf sich selbst gerichtet werden
Fix: TextBox zeigte fälschlicherweise HTML an
Fix: SMTP funktioniert nun korrekt mit iCloud/MobileMe
Fix: EnOcean Thermokon Instanz kann korrekt konfiguriert werden
Fix: Problem beim Löschen vieler Objekte
Fix: Problem beim Löschen von Objekte, wenn Unterobjekte mit ausgewählt wurden
Fix: Countdown im Skripteditor durch absoluten Zeitpunkt ersetzt.
Fix: Z-Wave - Problem mit fehlerhaften Fibaro FGR222 umgangen
Fix: Z-Wave - Probleme mit SENSOR_ALARM Klasse gelöst
Fix: Z-Wave - Probleme mit SENSOR_MULTILEVEL Klasse gelöst
Fix: Z-Wave - Probleme mit ALARM Klasse gelöst
Fix: Z-Wave - Probleme mit Batterieanzeige korrigiert
Fix: Skalierungsprobleme bei sehr kleinen Graphen gelöst
Fix: Konfiguratoren können nun korrekt per Tastatur/Pfeiltasten bedient werden (#IPSD-61,IPSD-64)
Fix: ArtNet sendet eine korrekte Versionsnummer
Fix: Z-Wave Philio 4-1 Sensor erstellt die Variable nun wirklich korrekt
Fix: Ersetzen in Dateien funktioniert korrekt

Fix: EIS5 wird korrekt abgerundet
Fix: Alle Gruppen Funktionen bei KNX können korrekt ausgewählt werden
Fix: WebFront zeigte keine Fehlermeldung, wenn Aktionsskript einen Fehler geworfen hat
Fix: Probleme beim Erstellen von Instanzen mit KNX Konfigurator behoben
Fix: Skin Manager wird auf dem korrekten Port geöffnet
Fix: WebFront Editor konnte Kategorie nicht auswählen, wenn diese zwischenzeitlich ungültig geworden ist
Fix: AccessViolation beim Laden der Geräte im WebFrontConfigurator, wenn Token/Lizenz unpassend war
Fix: EIS6 wird korrekt gesendet/interpretiert
Fix: xComfort 12-Kanalaktor wird korrekt unterstützt
Fix: Stop Kanal für KNX Shutter wird korrekt gesetzt
Fix: Automatischer Reconnect für Dashboard reagiert nun auch /form Parameter
Fix: IPS_SetScriptTimer kann wieder korrekt mit Werten >=24h umgehen
Fix: Sekündliche Fehlermeldung, wenn ArtNet Gateway nicht verbunden war
Fix: Legende kann nun über den WFC aktiviert/deaktiviert werden
Fix: Konfiguration von monatlichen Ereignissen korrigiert
Fix: Z-Wave: Inclusion/Exclusion sollte nun mit Razberry und Z-Wave.ME Stick korrekt funktionieren.
Fix: StartTime in IPS_GetScriptThread war falsch
Fix: Problem mit Sommerzeitumstellung korrigiert
Fix: FS10 kann wieder konfiguriert werden
Fix: Problem mit OZW Variablenprofilen gelöst
Fix: Fehler bei ausgelösten Ereignisse, welche Werte vom falschen Variablentyp enthielten
Fix: Skriptzuweisungen in Instanzkonfigurationen können entfernt werden
Fix: Bei alten KNX Instanzen konnten zusätzliche GAs nicht gespeichert werden
Fix: Profilmanager hatte verschobene Buttons
Fix: ModBus: Int64 sollte nun korrekt kodiert werden
Fix: Problem mit WAC_* Kompatibilitätsfunktionen
Fix: Alte iOS App sollte wieder korrekt funktionieren
Fix: Profilnamen werden korrekt auf Groß/Kleinschreibung geprüft

# V2.7->V3.0 (Q3/2013)

## WESENTLICHE ÄNDERUNGEN

IP-Symcon WebFront inkl. Editor
Eine besondere neue Funktion ist der WYSIWYG Editor, über den die Einrichtung von komplexen Visualisierungen zu einem Kinderspiel wird. Natürlich ist euer altes WebFront aus IP-Symcon 2.x vollständig dazu kompatibel und kann direkt mit dem neuen Editor bearbeitet werden. Zusätzlich kann das neue WebFront über Skins z.B. farblich umgestaltet werden, welche entweder direkt bei uns im Forum heruntergeladen, oder bei entsprechendem Know-How selbst erstellt werden können. Das neue WebFront bietet außerdem noch Unterstützung für MJPEG-Streams, Audio-Benachrichtigungen, komfortable Konfiguration von zyklischen Ereignissen und einen neuen Farbkreis für RGB-Geräte. Die vollständige Liste mit allen Änderungen ist unterhalb der Migration zu finden.
Neue unterstütze Geräte und Funktionen
Im Bereich DMX unterstützt IP-Symcon nun sämtliche Art-Net-Controller mit verschiedenen Universen.
Im Bereich LCN werden die neuen LCN-Module unterstützt, welche bis zu 12 Variablen zur Verfügung stellen.
Web-Schnittstelle (JSON-RPC)
Die neue, offzielle JSON-RPC Schnittstelle erlaubt eine einfache Anbindung von Dritthersteller-Software, eigenen mobilen Anwendungen und der einfachen Kommunikation zwischen IP-Symcon Servern. Wer Daten zwischen mehreren IP-Symcon Servern austauschen wollte, kann dies jetzt mit wenigen Handgriffen tun. Die Dokumentation liefert dazu unter dem Punkt Datenaustausch einige Beispiele.
*(Um den WebFront-Editor nutzen zu können ist mindestens eine IP-Symcon Professional Version nötig. Alle anderen Funktionen sind in der Basic, Professional und Unlimited Version von IP-Symcon verfügbar und können mit einer gültigen Subskription bezogen werden. Ein spezielles "Upgrade" muss nicht erworben werden. Sofern Ihre Subskription abgelaufen ist, können Sie diese einfach erneuern und direkt ohne zusätzliche Mehrkosten auf die Version 3.0 updaten. Bitte beachten Sie, dass das Nutzen der Push-Benachrichtigungen eine Verbindung zum Internet und eine gültige Subskription voraussetzt. (Shop )*

## VIDEO ZUM WEBFRONT-EDITOR

## MIGRATION



Bitte erstellen Sie auf jeden Fall ein Backup Ihres gesamten IP-Symcon Ordners, um auf die Version 2.7 zurück zu wechseln, sollten wider erwarten Probleme mit der Version 3.0 auftreten.



Falls Sie von der Version 2.6 updaten, lesen Sie bitte vorher die Migrationsanleitung zur Version 2.7: Migration auf Version 2.7 .

Benutzerdefinierte Icons werden in IP-Symcon 3.0 über WebFront-Skins eingebunden.

(Dokumentation )
Die Funktion AC_GetLoggedValues ist nun offiziell dokumentiert, hat sich jedoch leicht in der Funktionsweise geändert. Sofern genutzt, kann das in der Dokumentation zur Verfügung gestellt Mini-Skript die alte Funktionsweise vollständig wiederherstellen.
Der Typ von eigenen Profilen muss für das neue WebFront zum Variablentyp passen. Sofern eigene Profile nur über die Konsole erstellt wurden, ist dieser Hinweis nicht relevant.
Wenn im WebFront Instanzen unterhalb von Instanzen angezeigt werden sollen, dann muss in jedem Konfigurator die Option "Verschachtelung" aktiviert werden.
Um das On-Screen-Keyboard im WebFront unter Windows nutzen zu können, kann folgende Extension für Google Chrome installiert werden: Virtual Keyboard
Für die Graphen im Dashboard gibt es eine Änderung in der Nutzungsweise: Forum

## LISTE DER ÄNDERUNGEN

Komplett für HTML5-Browser neu entwickelt
Der integrierte WebFront-Editor ersetzt den unbequemen WebFront-Konfigurator
Eigene Skins können mit den Skin-Manager erstellt oder fertige direkt aus der Community heruntergeladen werden!
RGB-Slider haben nun zusätzlich einen HSL-Farbkreis
Audio-Notificatations über WFC_AudioNotification
MJPEG-Stream können über das neuen Stream-Mediaobjekt angezeigt werden. Medien
Graphen-Rohdaten können direkt nach CSV exportieren werden
Graphen im WebFront können für jede Variable deaktiviert werden (trotz aktiviertem Logging)
Alle Icons und Grafiken sind für Retina-Displays optimiert.
Das WebFront sieht nun auf dem iPad3+ und hochauflösenden Display einfach toll aus!
Lädt schneller - besonders bei langsamen Internetleitungen. Die schicke Ladeanimation versüßt die Wartezeit
Natives Scrolling auf iOS & Android Geräten
Zusatz Scrollleisten können für ältere Touchscreens alternativ per Skin aktiviert werden (Download )
Das Editieren von Text-Boxen nutzt nun das integrierte OSK des Touch-Geräts
DWD-Wetter benötigt keinen Benutzernamen/Passwort mehr
Inhaltswechsler kommt nun auch mit sehr vielen Objekten zurecht (z.B. Webcam Archiv Funktion)
Ereignisse werden nun direkt untereinander inklusive Konfiguration und nächster Ausführung angezeigt
Die neue JSON-RPC API arbeitet direkt im WebServer ohne kostbare PHP-Threads zu belegen
Über die neue JSON-RPC API können Sie mit wenigen Handgriffen Daten austauschen. Datenaustausch
Für alle Technikbegeisterten: Das neue WebFront ist mit Google Dart erstellt. Kein Dojo mehr.
MIN/MAX Anzeige bei Graphen
Farbauswahl mit persönlich anpassbarem Farbspeicher!
Verbesserungen beim dynamischen Skalieren von Graphen
Skin-Editor hat Option zum Einfärben der Icons
Skin-Icons können in der Konsole ausgewählt werden
Medien (Streams) können editiert werden
Es wird ein Ladekringel beim Klick auf ein Info-Widget angezeigt (Wenn ein OnKlick Skript

definiert wurde) (#WEBFRONT-19)
Wenn mehrere Popups per WFC_SendPop geöffnet werden, gibt es einen "Alle Popups schließen" Button (#WEBFRONT-2)
Unterstützung für Datenpunkte vom neuen HomeMatic Stellantrieb

Neue Systeme/Module

Neu: DMX Splitter unterstützt nun ArtNet (Unicast)
Neu: LCN untersützt die neuen Module mit 12 Variablen
Neu: Native JSON-RPC Schnittstelle

Neue Funktionen

Neu: GZIP-Komprimierung für den WebServer
Neu: X-Forwarded-For wird im WebServer ausgewertet und an PHP weitergeleitet
Neu: Netzwerkbibliothek aktualisiert. Behebt hoffentlich Probleme mit POP3/IMAP
Neu: Silent-Eigenschaft für TIPSWebBrowser um alle Javascript Fehler und Popups zu unterdrücken

# V2.6->V2.7 (Q1/2013)

## WESENTLICHE ÄNDERUNGEN

**Jetzt direkt auf dem Smartphone oder Tablet benachrichtigen lassen!**
Mit der Einführung von IP-Symcon 2.7 und der Apps in der Version 2.0 gibt es endlich die Möglichkeit direkt Push-Benachrichtigungen an Ihre Smartphones und Tablets zu versenden. Lassen Sie sich zum Beispiel bei einer fertigen Waschmaschine, beim Klingen der Tür oder bei Alarmzuständen sofort informieren. Viele weitere Ideen finden Sie auch in unserem Community-Forum
**Neue unterstütze Geräte und Funktionen**
Zur besseren und komfortableren Einrichtung haben wir die Konfiguratoren überarbeitet, sodass diese die volle Größe der Konsole nutzen können. Dadurch haben Sie bei vielen vorhandenen Geräten oder Adressen einen vollständigen Überblick. Zusätzlich haben wir einen Konfigurator für OneWire integriert und den KNX-Konfigurator so erweitert, dass er die Rückmeldeadresse direkt in IP-Symcon importiert.
Ein neues Modul ist für die TermoJET und LevelJET Systeme der Fa. ProJet verfügbar. Die Geräte können in das IPS-868 Funksystem über ein neues Funkgateway (JKM-868) angebunden werden. Dadurch lassen sich die empfangenen Temperaturen oder Füllstände in IP-Symcon auswerten. Alle Geräte des IPS-868 Funksystems können über den ProJET Shop erworben werden.
Im Bereich Z-Wave haben wir die Klasse MULTI_INSTANCE komplett überarbeitet, sodass die neuen Geräte von Fibaro, wie z.B. der 2-fach Aktor oder der vielseitige Binärsensor einfach und mit allen Funktionen genutzt werden können.
Ferner wurden wieder neue Geräte unseres Vertriebspartners ALLNET in IP-Symcon integriert:

- ALL3073WLAN - Netzwerksteckdose zum Schalten über WLAN (16A)
- ALL3075V2 - Netzwerksteckdose zum Schalten über LAN (8A)
- ALL4075 - Netzwerk / WLAN Relais 4-fach 230V/10A
- ALL4076 - IP-Steckdosenleiste 6-fach, schaltbar per Netzwerk (6*10A)
- ALL3418V2 - LAN / WLAN IP Thermometer (ALL3418v2)

*(Alle Funktionen sind in der Basic, Professional und Unlimited Version von IP-Symcon verfügbar und können mit einer gültigen Subskription bezogen werden. Ein spezielles "Upgrade" muss nicht erworben werden. Sofern Ihre Subskription abgelaufen ist, können Sie diese einfach erneuern und direkt ohne zusätzliche Mehrkosten auf die Version 2.7 updaten. Bitte beachten Sie, dass das Nutzen der Push-Benachrichtigungen eine Verbindung zum Internet und eine gültige Subskription voraussetzt. (Shop ))*

## MIGRATION



Bitte erstellen Sie auf jeden Fall ein Backup Ihres gesamten IP-Symcon Ordners, um auf die Version 2.6 zurück zu wechseln, sollten wider erwarten Probleme mit der Version 2.7 auftreten.



Falls Sie von der Version 2.5 updaten, lesen Sie bitte vorher die Migrationsanleitung zur Version 2.6: Migration auf Version 2.6

Falls Sie von Z-Wave einen der Mehrfachaktoren (z.B. Fibaro Dual Switch) verwenden, so müssen Sie die Untergeräte über eigene Instanzen einbinden, da die Unterstützung entsprechend verbessert wurde. Beachten Sie auch, dass die ZW_SwitchModeEx Funktion entfernt wurde, und jedes Untergerät korrekt per ZW_SwitchMode angesprochen wird.

## LISTE DER ÄNDERUNGEN

Jetzt direkt in den Mobilanwendungen benachrichtigen lassen!
- Neu: Push-Benachrichtigungen (Über Kern-Instanzen -> Notification) (Dokumentation Notification Control)
- Neu: WFC_PushNotification($wfcid, $title, $message, $icon, $targetID); (Dokumentation: (link: page://ByMmKKFK2unaOdIE text: WFC_PushNotification ))

Neue Systeme/Module
- Neu: DMX Module komplett erneuert. Es werden intern die nativen IP-Symcon I/O Module genutzt
- Neu: Neue ALLNET Geräte werden unterstützt (ALL3073 WLAN, ALL3075V2, ALL3418V2, ALL4075, ALL4076)
- Neu: KNX-Konfigurator importiert zusätzliche "Empfangs-Gruppenadressen" aus der ESF-Datei
- Neu: OneWire besitzt einen komfortablen Konfigurator. (Dies ersetzt die Suche innerhalb der Instanz!)
- Neu: Z-Wave unterstützt jetzt korrekt Mehrfachgeräte mit der MULTI_INSTANCE Klasse
- Neu: Module für das IPS-868 System (IPS-JKM-868 ThermoJET-, LevelJET-Transmitter)
- Neu: WebFront Konfigurator umgestaltet. Gruppiert. Es gibt jeweils mehr Platz.
- Neu: Alle Konfiguratoren nutzen die komplette Bildschirmgröße

Neue Funktionen
- Neu: Fehlerbehandlung der I/Os wird zentral vom Event Control übernommen
- Neu: WWWReader kann deaktiviert werden. (Praktisch für Systeme wie digitalStrom/OZW)
- Neu: Proxy-Support ist nun IPS-Systemweit einheitlich. Übernimmt Konfiguration aus dem LiveUpdate.
- Neu: Module nutzen nun zu 90% die neuen IPS_SetProperty Befehle zur konfiguration. (Dokumentation: (link: page://pGgwGLaGoWzdLcTX text: IPS_SetProperty ))
- Neu: Warnung, wenn neues Dashboard-Formular mindestens 10% kleiner ist, als das Original. (Workaround für Teilweise fehlerhafte Uploads)
- Neu: Alte, seit 2010 abgekündigte MBT_* Funktionen entfernt
- Neu: Sortierung der Spalten im ArchiveControl
- Neu: Velleman Board liest die AnalogInputs direkt in Variablen ein
- Neu: M-Bus unterstützt nun weitere Spezialcodierungen für Zählerüberlauf/ Zählerunterlauf
- Neu: Unterstützung für 2-fach Bidi-Aktoren von Eltako. (Neue Option im Eltako Switch Modul)
- Neu: Modul für Thermokon Thanos (EnOcean)

Behobene Fehler
- Fix: Timeout bei Abfrage einiger M-Bus Geräte
- Fix: WebFront Ladeanimation bleibt hängen
- Fix: Überprüfung der maximalen WebFront Konfigurationen funktioniert nun korrekt
- Fix: Zeitüberschreitungsfehler bei KNX Sendeanfrage
- Fix: Potenzieller Division durch Null bei der Graphengenerierung
- Fix: AccessViolation beim Erstellen des Objektbaumes über Ansicht hinzufügen
- Fix: AccessViolation beim Hochladen von Dashboard-Formularen

Fix: ALLNET3691 Phasen 4-6 werden korrekt angezeigt
Fix: Potenzielle Sicherheitslücke im WebFront Retro/Retro-Mobile
Fix: Cutter mit #0
Fix: Zeilenumbrüche werden in der Objektbeschreibung korrekt angezeigt
Fix: KNX: Fehler beim Empfangen von Rückmeldungen beim Siemens N146/02 IP-Gateway

# V2.5->V2.6 (Q3/2012)

## WESENTLICHE ÄNDERUNGEN

IP-Symcon startet, lädt und agiert schneller
In der Version 2.6 haben wir besonders die Performance verbessert. Die Startzeit des Dienstes konnte minimiert werden, so dass auch bei großen Projekten ein Aktualisieren der Software nur wenige Minuten in Anspruch nimmt. Da die Konsole zusätzlich nahezu komplett überarbeitet wurde, sind auch dort Start- und Ladezeiten minimal. Das WebFront wurde optimiert und den neusten Sicherheitsanforderungen der aktuellen Browser angepasst. Durch die neuen Verbesserungen wird außerdem die CPU weniger belastet, was sich besonders auf älteren Systemen schnell bemerkbar macht. Auch beim WebFront Konfigurator haben wir auf Feedback reagiert und dessen Bedienung verbessert.
Neue Hardware-Systeme
Darüber hinaus haben wir neue Systeme und Module integriert. Problemlos können nun in IP-Symcon u.a. das gesamte digitalStrom- und das innovative Siemens-Synco™-living-System eingebunden werden, wodurch unsere breite Palette an unterstützter Hardware perfekt ergänzt wurde. Das macht unsere Hausautomationssoftware noch flexibeler, als sie bereits ist.
Noch zuverlässiger
Wir haben zusätzliche die Version unserer integrierten Skriptsprache PHP aktualisiert, um auch in IP-Symcon die neusten Sicherheitspatches, die die PHP-Community veröffentlicht hat, zu nutzen. Wir haben weitere gewünschte Optimierungen vorgenommen, damit IP-Symcon noch besser und leistungsfähiger sein kann. Die vollständige Liste dazu kann unterhalb des Kapitels Migration eingesehen werden.
*(Alle Funktionen sind in der Basic, Professional und Unlimited Version von IP-Symcon verfügbar und können mit einer gültigen Subskription bezogen werden. Ein spezielles "Upgrade" muss nicht erworben werden. Sofern Ihre Subskription abgelaufen ist, können Sie diese einfach erneuern und direkt ohne zusätzliche Mehrkosten auf die Version 2.6 updaten. (Shop ))*

## MIGRATION



Bitte erstellen Sie auf jeden Fall ein Backup Ihres gesamten IP-Symcon Ordners, um auf die Version 2.5 zurück zu wechseln, sollten wider erwarten Probleme mit der Version 2.6 auftreten.



Falls Sie von der Version 2.4 updaten, lesen Sie bitte vorher die Migrationsanleitung zur Version 2.5: Migration auf Version 2.5 .

Sofern der IPSLogger von Brownson verwendet wird, ist ein Update dieser Komponente für den fehlerfreien Betrieb der 2.6 zwingend erforderlich. Bei etwaigen Problemen mit Abstürzen empfiehlt es sich den IPSLogger temporär zu deaktivieren, um dessen Funktion als Fehlerquelle auszuschließen.
Falls auf Ihrem Windows Server 2008 R2 die Installation des VCRedist 2008 in einer Endlosschleife feststecken sollte, überprüfen Sie bitte, ob Ihr System die korrekte Version

vom VCRedist 2008 installiert. Es muss die Version ohne SP1 installiert werden, da nur diese mit PHP korrekt funktioniert!
Falls Sie die COM/.NET Wrapper Funktionen von PHP nutzen, so ist mit Version 2.6 #2554 (und damit verbunden PHP 5.4.8) eine manuelle Installation der Extension erforderlich, da PHP sich entschieden hat diese auszulagern.

## LISTE DER ÄNDERUNGEN

Schneller und besser
- Neu: Performance verbessert, so dass die CPU weniger belastet wird. IP-Symcon sollte sich auf Low-End Systemen wesentlich agiler anfühlen.
- Neu: Performance verbessert, so dass IP-Symcon viel schneller startet. (Bei sehr vielen Objekten reduziert sich die Ladezeit auf <1% der ursprünglichen Ladezeit)
- Neu: Neues Format zum Speichern der Settings, welches wesentlicher schneller lädt und die sporadisch auftretenden Probleme mit der "settings.xml" lösen soll.
- Neu: Konsole wurde komplett überarbeitet. Sie lädt schneller und reagiert besser.
- Neu: WebFront Konfigurator wurde optimiert
- Neu: Upgrade auf die aktuellste PHP-Version 5.4.4

Neue Systeme/Module
- Neu: Module für digitalStrom (Firmware 1.4.0 oder neuer erforderlich)
- Neu: Module für Siemens OZW (OZW Firmware 4.0.4 erforderlich.
- Neu: Z-Wave Klasse: COMMAND_CLASS_SENSOR_ALARM
- Neu: Z-Wave Fibaro Dual Switch
- Neu: Z-Wave Alarm Klasse unterstützt mehr Typen
- Neu: Modul für ALLNET 3691, 4500, 5000

Neue Funktionen
- Neu: Spezialoptionen können nun über die Tray-Anwendung eingestellt werden. (Siehe: (link: page://s6UcBMxs0S8FHhPA text: Spezialschalter ))
- Neu: Bessere Fehlermeldung bei Ausführungsfehlern von PHP-Code innerhalb von Ereignissen
- Neu: Restlaufzeit-Anzeige bei Ereignissen innerhalb vom Skript-Seitenmenü
- Neu: Alle hauptsächlich genutzten $_IPS[*] Systemvariablen werden bei STRG+SPACE angezeigt.
- Neu: SerialPort: DTR/RTS FlowControl Funktionen
- Neu: Das Deaktivieren von Autoscroll gibt einen Hinweis und deaktiviert das Limit der Einträge.
- Neu: PHP Memory-Limit ist jetzt default auf 32MB. Maximal jedoch 64MB. (Siehe php.ini)
- Neu: Maximale PHP-Threadanzahl staffelt sich je nach installiertem RAM (max 40 bei 4GB, 30 bei 3GB, 20 bei 2GB, 10 bei weniger).
- Neu: PHP-Skript Queue meldet sich schon ab zehn Skripten in der Schleife.

Behobene Fehler
- Fix: Benutze Standardaktion kann wieder korrekt genutzt werden
- Fix: WebFront: Timing-Problem bei Ausführung vieler gleichzeitiger IPS_SetHidden Befehle behoben
- Fix: WebFront: Lade-Problem mit aktueller Chrome Version bei aktiviertem SSL und externem Zugriff
- Fix: VCRedist2008 wurde auf dem Server2008 R2 inkorrekt als installiert erkannt.
- Fix: Neue $_IPS Schreibweise in den Ereignis-Templates wird nun genutzt.
- Fix: LCN: Sonderzeichen bei Modulnamen wird korrekt gefiltert.
- Fix: Siemens S5: Verbindungsprobleme gelöst
- Fix: Eaton xComfort: Signalstärke in Verbindung mit dem ECI LAN Gateway

Fix: Eaton Analogeingang: Fehler beim Auswerten von Prozentwerten
Fix: Fehler, dass unter Umständen $IPS_* nicht korrekt definiert war
Fix: WebFront Startkategorie wurden u.U. nicht korrekt übernommen
Fix: Reihenfolge (Objekt-Position) im ContentChanger wird korrekt ausgewertet.
Fix: LCN: R2 Regler wurde nicht korrekt ausgelesen.
Fix: Timing Problem/Umschaltzeiten für LCN Shutter
Fix: Sporadisch auftretendes Floating-Point Problem bei SetValue
Fix: Zu viele Datensätze bei einer AC_* Abfrage konnten das System negativ beeinträchtigen. Limitation auf 10000 Datensätze pro Abfrage. (Siehe: (link: page://CJ7CQ655yiILr2pO text: Limitationen ))
Fix: Unendlich viele Fehlermeldungen (z.B. korrupte for-Schleife) konnten den kompletten RAM des Systems aufbrauchen.
Fix: Widgets konnten in SplitPanes platziert werden.
Fix: False-Positives bei Sys_Ping
Fix: Problematische Werte bei Sys_GetCPUInfo
Fix: Link unter Link konnte fälschlicherweise erstellt werden.
Fix: Alle FHZ Module (FS20, HMS, FHT, KS300) werfen nun direkt Fehlermeldungen, wenn die I/O Instanz nicht verfügbar ist
Fix: OSK übernimmt Zeilensprünge nicht.

Keine Abwärtskompatibilität für folgende Änderungen:
Geändert: IPS_GetFunctionList (Gibt nur noch den Namen zurück. Bitte mit IPS_GetFunction kombinieren.)
Entfernt: FHZ_GetQueue, FHZ_GetFHTQueue
Entfernt: IPS_SendMediaEventEx entfernt

Vollständige Abwärtskompatibilität für folgende Änderungen:
Neu: IPS_FunctionExists, IPS_GetFunction hinzugefügt
Neu: IPS_SetLinkChildID wurde zu IPS_SetLinkTargetID umbenannt. IPS_SetLinkChildID wird zur 3.0 entfernt.
Neu: IPS_GetLink liefert das Feld TargetID, welches identisch zu LinkChildID ist. LinkChildID wird zur 3.0 entfernt.
Neu: IPS_HasInstanceParent, IPS_GetInstanceParentID veraltet. Kann direkt über IPS_GetInstance und ConnectionID abgefragt werden.
Neu: IPS_HasInstanceChildren, IPS_GetInstanceChildrenIDs veraltet. Kann direkt über IPS_GetInstance und ConnectionID abgefragt werden. Ein Beispiel dazu befindet sich auf IPS_GetInstanceChildrenIDs
Neu: LCN_GetStatus wird nur noch emuliert. Auf GetInstance($id)['InstanceStatus'] ausweichen.

Folgende Konfigurationsfunktionen werden nur noch emuliert:
CSCK_SetOpen, CSCK_SetHost, CSCK_SetPort, CSCK_GetOpen, CSCK_GetHost, CSCK_GetPort, SSCK_SetOpen, SSCK_GetOpen, SSCK_SetPort, SSCK_GetPort

# V2.4->V2.5 (Q4/2011)

## WESENTLICHE ÄNDERUNGEN

Mit IP-Symcon 2.5 wurde auf die neue PHP 5.4 Version aktualisiert, die viele Verbesserungen bei Skripten und die Sicherheit dieser mitbringt. Desweiteren wurden neue Funktionen für eine Vielzahl von Hardware-Systemen hinzugefügt und bekannte Fehler beseitigt. Ein Auszug der neuen Funktionen:

- Unterstützung der neuen bidirektionalen EnOcean TCM310 Geräte mit dem EnOcean Serial Protocol 3.
- Unterstützung der neuen bidirektionalen EnOcean Aktoren von Eltako
- Unterstützung des neuen Eaton Xcomfort Thermostats
- Unterstützung für das neue Xcomfort LAN-Gateway (ECI) und deren spezielle DPL Datenpunktliste
- Unterstützung vom OneWire Baustein DS2406
- Komfortables LCN Modul für Rollladen/Jalousien bei Ansteuerung über Ausgänge/Relais
- Unterstützung für das neue LAN-868 Gateway für das IPS Funksystem
- Unterstützung für neue Z-Wave Klassen
  - COMMAND_CLASS_CLOCK
  - COMMAND_CLASS_PROTECTION

Im Kern wurden die Importier- und Kompatibilitätsfunktionen aus IP-Symcon 1.x entfernt, so dass viele Funktionen schneller und effizienter arbeiten können. Desweiteren wurde eine neue Funktion (IPS_SetIdent) geschaffen, mit der Variablen eindeutig per Name identifiziert werden können. Dadurch können Skripte portabel erstellt und mit Nutzern besser ausgetauscht werden. Ein Beispiel dafür ist das Skript zum Auslesen von Miele@Home Geräten (Link ).
Alle Funktionen sind in der Basic, Professional und Unlimited Edition von IP-Symcon verfügbar und können mit einer gültigen Subskription bezogen werden. Ein spezielles “Upgrade” muss nicht erworben werden. Sofern Ihre Subskription abgelaufen ist, können Sie diese einfach erneuern und direkt ohne zusätzliche Mehrkosten auf die 2.5 updaten. (Shop )

## MIGRATION



Bitte erstellen Sie auf jeden Fall ein Backup Ihres gesamten IP-Symcon Ordners, um auf die Version 2.4 zurück zu wechseln, sollten wider erwarten Probleme mit der Version 2.5 auftreten.



Falls Sie von der Version 2.3 updaten, lesen Sie bitte vorher die Migrationsanleitung zur Version 2.4: Migration auf Version 2.4

Die "auto_prepend_file"-Funktion in der "php.ini" wird von IP-Symcon intern genutzt. Falls eine PHP-Datei automatisch geladen werden soll, so muss diese im Skript Ordner liegen und "__autoload.php" heißen.
Die Extension "php_http.dll" ist in PHP 5.4 nicht mehr verfügbar. Falls diese genutzt wurde bitte auf die cURL-Extension ausweichen.

Durch eine Inkompatibilität von PHP 5.4 und OpenSSL 1.0 können über die "file_get_contents" (u.ä.) Funktion keine Anfragen per SSL gestellt werden. Die IP-Symcon interne Funktion Sys_GetURLContent bietet dort passenden Ersatz.
Einige spezielle Funktionen zum Handling von StatusVariablen werden zur 3.0 entfernt werden. Sobald Sie auf 2.5 aktualisiert haben, können Sie dieses Skript zur Überprüfung nutzen: IP-Symcon 2.5 – Hilfsskripte
Falls Skripte im "webfront/user"-Ordner erstellt und genutzt wurden, müssen in PHP 5.4 alle Parameter per "$_GET[...]" abgefragt werden.
Kompatibilitätsfunktionen aus alten IP-Symcon 1.0 Zeiten und die Kompatibilität mit Namen statt ID bei alten Funktionen wurde entfernt. Für alle die mit IP-Symcon 2.x angefangen haben, hat dies keine Bedeutung.

## LISTE DER ÄNDERUNGEN

Neu: PHP 5.4
Neu: PHP Extensions werden per LiveUpdate **aktualisiert** (nur offizielle von PHP. Kein PECL!).
Neu: EnOcean Module können mit dem neuen ESP3 Protokoll umgehen.
Neu: “No configuration” Fehlermeldung verbessert
Neu: Modbus TCP noch robuster gemacht
Neu: xComfort Thermostat
Neu: LCN Shutter Modul
Neu: Neuer Lichtsensor (LCN-LSA) wird unterstützt.
Neu: Typ kann für T, R1, R2 gewählt werden.
Neu: BiDi Auswertung für Eltako Schalter/Dimmer
Neu: Eltako Dimmer kann direkt angesteuert werden. (Muss erneut angelernt werden!)
Neu: xComfort Thermostat kann Variable als Ist-Temperatur verwenden.
Neu: Ports für Wired/RF/Eventkanal können beim HomeMatic Socket eingestellt werden.
Neu: OneWire DS2406
Neu: Z-Wave Rückkanal bei Fibaro Geräten (Über BASIC Kommando Klasse)
Neu: LAN Version vom USBT-868 wird komplett unterstützt.
Neu: $IPS_ *ist nun per $_IPS[]* verfügbar-Neu: RegisterGlobals ist off. Das bedeutet, dass alle Variablen vom WebServer nur noch per $_GET abrufbar sind. (PHP5.4 erfordert dies!)
Neu: OpenSSL 1.0 wird genutzt. Löst Probleme mit SSL-Zugriff von Mobilgeräten
Neu: Importer für V1 Settings wurden entfernt.
Neu: IPS_SetValue*/IPS_GetValue*, IPS_SetScriptTimer, IPS_GetScriptTimer, IPS_RunScript* akzeptieren keine Strings mehr (V1 Kompatibilität).
Neu: IPS_GetVariableID, IPS_GetScriptID entfernt (V1 Kompatibilität). IPS_GetVariableIDByName/IPS_GetScriptIDByName als Ersatz nutzen
Neu: IPS_StatusVariableExists, IPS_GetStatusVariable, IPS_GetStatusVariableID, entfernt. Ersatz: IPS_GetObjectIDByIdent
Neu: IPS_GetStatusVariableIdents entfernt. Ersatz: IPS_GetChildrenIDs mit IPS_GetObject (ObjectIdent)
Neu: IPS_SetStatusVariableUseAction entfernt. Ersatz: Noch keiner.
Neu: IPS_SetIdent. Erlaubt einen internen, auf Kategorieebene eindeutigen, Namen zu vergeben, der dann wieder über IPS_GetObjectIDByIdent abgefragt werden kann. Dieses Feld kann für universelle Skripte hervorragend genutzt werden, um in Skripten auf Variablen per Name/Identifikator zuzugreifen und dem User trotzdem Änderungen an den Namen zu erlauben.
Neu: Links können über das Kontextmenü schneller erstellt werden. Objekt verlinken anklicken. Ziel auswählen. Fertig. (Die ursprüngliche Idee mit den Standard-Windows-Shortscuts erwies sich als viel zu kompliziert in der Verwendung.)
Neu: Sortieren nach Werten in der Listenansicht möglich

Neu: Unterstützung für neue Z-Wave Klassen
  COMMAND_CLASS_CLOCK
  COMMAND_CLASS_PROTECTION
Fix: W&T: Teilweise waren Kanäle nicht auslesbar.
Fix: IPS_Semaphore* Fix. Danke an tgusi74 fürs finden!
Fix: HomeMatic Wired Komponenten wurden nicht korrekt automatisch erstellt.
Fix: ALL3690: Auslesen der Summen wartet nun max. 5min auf die Berechnung
Fix: Konsole konnte per Remote nicht aktualisiert werden, wenn auf dem Server (Win7) die Konsole gestartet war.
Fix: Strings wurden in AC_GetLoggedValues nicht zurückgegeben.
Fix: Werte bei % Profilen wurden bei “MinValue < 0” falsch berechnet
Fix: LCN Modul Konfigurator zeigte TVar/RVars Instanzen nicht korrekt an
Fix: Profil beim KS300 korrigiert
Fix: Installation unter Windows 7 (x86) ist nun im Programme Ordner möglich
Fix: GIF Bilder sollten Dashboard nicht mehr zum Absturz bringen.
Fix: AssertionViolation bei TeamViewer 6 QuickConnect behoben
Fix: FTDI Instanz wurde als fehlerhaft markiert, wenn diese deaktiviert war, aber man schalten wollte.
Fix: Werte bei Boolean Assoziationen können per PHP nur noch modifiziert werden.
Fix: IP-Symcon stürzte ab, wenn der media Ordner nicht existierte.
Fix: Sortieren nach Wert in Listenansicht funktionierte nicht.
Fix: Xcomfort Feuchtigkeit wurde im Sende AN/AUS Modus nicht empfangen.
Fix: DS2438 VAD/VDD wurden unter Umständen nicht korrekt gelesen.
Fix: GetValue versagte nach Aufruf externer Programme oder RRD-Execute.
Fix: LCN Limit auf 5 Telegramme/Sekunde (PCHK 2.1 erfordert dies)
Fix: DS18S20/DS18B20 – Startwert 85 wird nicht mehr gelesen.
Fix: LiveUpdate überprüft, ob CRT Bibliotheken installiert sind.

# V2.3->V2.4 (Q2/2011)

## WESENTLICHE ÄNDERUNGEN

Der Fokus von IP-Symcon 2.4 liegt auf unserer bekannten und einzigartigen Visualisierung – dem WebFront. Die neuste Version hat ein neues, frisches Design und wurde "unter der Haube" nahezu komplett neu entwickelt. IP-Symcon verwendet die neusten Web-Technolgien und ist für die kommenden IP-Symcon-Versionen gerüstet. Wer es noch nicht live gesehen hat, kann jederzeit auf www.webfront.info gehen und sich selbst ein Bild von dem Erscheinungsbild und den neuen Funktionen machen. Einzelne Highlights werden nachfolgend dargestellt.

### Mehrere WebFront-Konfigurationen*

Wer kennst das Problem nicht? Es soll ein eigenes Panel für das Wohnzimmer, für den Service-Techniker oder für das Kind, welches nicht an allen Knöpfen und Reglern drehen soll, geben. Mit diesem Feature bietet IP-Symcon demnächst die volle Kontrolle für verschiedene Bereiche oder Gruppen. Die Auswahl der Konfiguration geschieht, sofern es mehrere gibt, über einen Login-Dialog. Alternativ kann für festgelegte IP-Adressen eine bestimme Konfiguration automatisch geladen werden. Ohne komplizierte Passwort-Eingabe. Die oft nachgefragte Funktion, im sicheren Heimnetzwerk keine Passwörter eingeben zu müssen, gibt es per Mausklick dazu.

### Farbige Zustände

Die schaltbaren Rot/Grün-Buttons haben in IP-Symcon 2.4 ausgedient! Stattdessen können für alle Assoziationen benutzerdefinierte Farben angeben werden. IP-Symcon bedient sich dabei standardmäßig bei Grün für alle Geräte, die aktuell Strom verbrauchen, Rot für akute Warnmeldungen und Blau für Geräte, die sich gerade in Aktion befinden. Neutrale Zustände werden gräulich und dadurch unauffällig dargestellt. Selbstverständlich kann auch diese Funktionsweise völlig umgestaltet werden über die altbekannten Variablenprofile.

### Wecker-Funktion

Alle verfügbaren zyklischen Ereignisse können nun direkt über das WebFront konfiguriert werden. Sei es der Wecker, die Gartenbewässerung oder die zeitabhängige Nachtabsenkung. In IP-Symcon 2.4 kann dies direkt über das Touch-Panel ohne Mehraufwand eingestellt werden.

### Automatische Aktualisierung aller Objekte im WebFront

WFC_Reload? Nicht mehr notwendig. Alle Objekte im Objektbaum werden vollständig im WebFront aktualisiert. Sei es der Name, der Zustand, das Profil, das Icon oder einfach nur die Sichtbarkeit. Neue Objekte werden selbstverständlich auch hinzugefügt und gelöschte entsprechend entfernt.

### Volle Konfigurationsvielfalt für Unterebenen**

Dank der neuen TabPanes (Menüleisten mit Reitern und Widgets) lassen sich bequem neue Unterebenen erstellen. In Kombination mit den SplitPanes (ehemals Container) können so eigene Menüleisten (TabPanes) links und rechts erstellt werden, die jeweils auf neue Kategorien, Inhaltwechsler oder Graphen verweisen können. Ein Beispiel dafür ist unsere dreistufige Gebäude-Navigation im SmartHome Paderborn.

### Balkendiagramme

Für die Darstellung der Zähler-Graphen werden wir in IP-Symcon 2.4 eine Balkenansicht, statt der bekannten Kurven, verwenden, um den Verbrauch besser verständlich darzustellen.

* Diese Funktion ist nur für IP-Symcon Professional und IP-Symcon Unlimited Kunden verfügbar. In der IP-Symcon Professional Version können maximal fünf (5) WebFront Konfiguratoren erstellt werden. IP-Symcon Unlimited beinhaltet, wie der Name schon sagt, keine Limitierung.

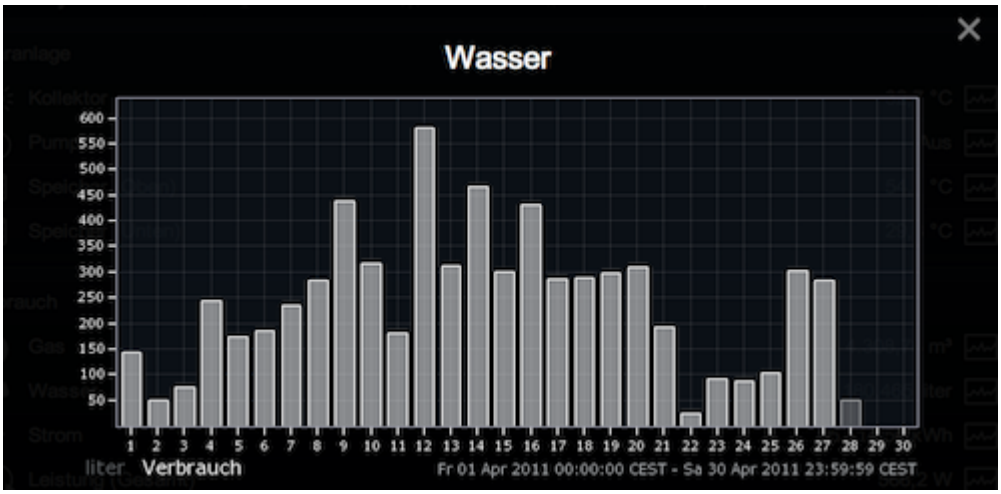
** Diese Funktion ist in IP-Symcon Basic nicht verfügbar.

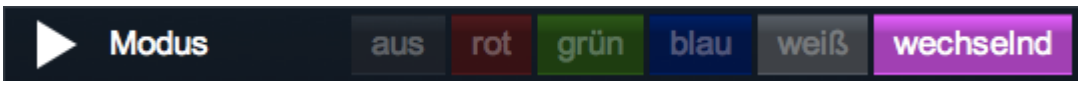
## SCREENSHOTS

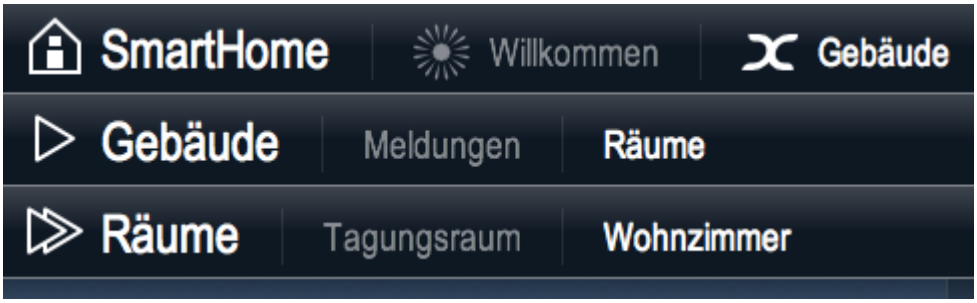











## MIGRATION

Bitte erstellen Sie auf jeden Fall ein Backup Ihres gesamten IP-Symcon Ordners, um auf die Version 2.3 zurück zu wechseln, sollten wider erwarten Probleme mit der Version 2.4 auftreten.

Falls Sie von der Version 2.2 updaten, lesen Sie bitte vorher die Migrationsanleitung zur Version 2.3:
Migration auf Version 2.3

*Der WebFront-Ordner wird automatisch von unbekannten und veralteten Dateien gereinigt. Alle eigenen Dateien, die nicht gelöscht werden sollen, müssen in den user-Unterordner verschoben werden!*
Variablen ohne Profil werden angezeigt. Um alle Variablen ohne Profil auszublenden, kann das folgende Skript genutzt werden. (Link )
Das iFront ist auf Version 1.1 zu aktualisieren, um mit IP-Symcon 2.4 kompatibel zu sein.
Die Module RRDTool (Link ) und XBee Device (bereit 2009 abgekündigt) wurden entfernt.

## LISTE DER ÄNDERUNGEN

Neu: iFront 1.1 wird unterstützt
Neu: Diverse neue Icons
Neu: Module (z.B. Chromoflex) aus IP-Symcon 2.3 können weiterverwendet werden.
Neu: Link auf Ereignis wird erlaubt.
Neu: Sortierung im WebFront Konfigurator / Archiv Modul
Neu: Auswerten von empfangenen Intensity-Werten bei FS20
Neu: Letzter Start von Ereignissen wird gespeichert.
Neu: Entpackvorgang beim LiveUpdate wird angezeigt.
Neu: Dynamische Skalierung bei Graphen jetzt zuschaltbar
Neu: XBee Device Modul entfernt (XBee Splitter als Ersatz verwenden)
Neu: WebFront-Konfigurator mit direkter Konfiguration für Retro, Retro Mobile und iFront
Neu: IPS/ProJet WDT-868 Modul
Neu: ~HM.Motion / ~Presence mit Icon
Neu: Profilliste wird sortiert.
Neu: .Background wird nicht mehr unterstützt.
Neu: Ausklappbare Boxen wurden endgültig entfernt.
Neu: . (Punkt) Notation im WebFront wird nicht mehr unterstützt.
Neu: retro.php/retromobile.php/ifront.php in index.php zusammengeführt

Neu: Parameter für dashboard.php überarbeitet (dynamicScaling=true/false)
Neu: IPS_SetVariableProfileAssociation wurde geändert (neuer Color Parameter).
Neu: Profilassoziationsanzahl auf maximal 32 beschränkt
Neu: KNX: Auslesen von Adressen vom Bus. (EIB_RequestStatus)
Neu: KNX: KNX Geräte können Leseanforderungen an IP-Symcon schicken.
Neu: KNX: FT1.2 kann über TCP/IP getunnelt werden.
Neu: KNX: Spezialmodul für Rollläden/Jalousien
Neu: Eltako FAFH60 (Temp-/Feuchtigkeitssensor)
Neu: Eltako FAH60 (Helligkeitssensor)
Neu: Spezial Eltako Modul für Schalter/Dimmer die Invertiert sind
Neu: ALL3418 wird unterstützt.
Neu: ALL3421 wird unterstützt.
Neu: ALL3690 funktioniert korrekt mit der neuen Firmware 2.0 (Diese wird auch erfordert!).
Neu: Z-Wave: Neue Kommandoklassen werden unterstützt.
  COMMAND_CLASS_METER
  COMMAND_CLASS_ALARM
  COMMAND_CLASS_LOCK
  COMMAND_CLASS_THERMOSTAT
Fix: “Jetzt aktualisieren” sperren, wenn Konfiguration geändert wurde (ImageGrabber/ WWWReader)
Fix: DWD FTP-URL angepasst (outgoing2)
Fix: ThermoJet wird korrekt bei RF-Übertragung ausgewertet.
Fix: Fehlermeldung, wenn Name verändert wurde (ALL3690)
Fix: %f in Assoziationen wurde falsch ausgewertet
Fix: Assoziationen wurden nicht überall korrekt nach dem Muster Präfix %2B Name %2B Suffix dargestellt.
Fix: KNX Zeitüberschreitungen
Fix: OneWire DS2450, Timing-Problem beim Auslesen behoben
Fix: Flackern des Volume-Sliders im WebFront beim MediaPlayer Modul
Fix: Sys_GetMemoryInfo() Limit von 4GB behoben

# V2.2->V2.3 (Q3/2010)

## WESENTLICHE ÄNDERUNGEN

HomeMatic CCU ohne TCPDump Erweiterung
*Funktioniert auch mit dem LAN Adapter, wenn der BidCos Service installiert ist.*
Container-Modul für das WebFront (Horizontale/Vertikale Aufteilung)
WebFront Container
Inhaltswechsler für das WebFront
*Erlaubt die Anzeige von Media Dateien, TextBoxen, HTMLBoxen, die zeitlich umgeschaltet werden können in einem Reiter. (Besonders geeignet für WebCams, die durchgeschaltet werden wollen.) Alle Objekte müssen in einer Kategorie liegen und auf die muss dann die BaseID zeigen.*
KNX Konfigurator
*Bietet das einfache Importieren von Gruppenadressen aus dem ETS Export (*.esf). Über einen Klick werden automatisch alle Gruppenadressen importiert.*
iPhone/iPad Verbesserungen für das WebFront
Splashscreen, Icons fürs iPad, …

## MIGRATION



Bitte erstellen Sie auf jeden Fall ein Backup Ihres gesamten IP-Symcon Ordners, um auf die Version 2.2 zurück zu wechseln, sollten wider erwarten Probleme mit der Version 2.3 auftreten.



Falls Sie von der Version 2.1 updaten, lesen Sie bitte vorher die Migrationsanleitung zur Version 2.2: Migration auf Version 2.2

Spezielle Änderungen sind nicht erforderlich.
Falls Sie im WebFront die DWD Seite nutzen, müssen Sie einen Benutzernamen/Kennwort vom DWD anfordern und im Konfigurator angeben.

## LISTE DER ÄNDERUNGEN

Neu: Container-Modul für das WebFront (Horizontale/Vertikale Aufteilung)
Neu: MediaChanger für das WebFront
Neu: HomeMatic ohne TCPDump
Neu: FS20: Weitere Adresse können angegeben werden. (Gruppen-/Masteradressen)
Neu: ProJet Input Modul
Neu: MediaPlayer Funktion (WAC_SetPosition)
Fix: WebFront DWD (Es muss ein Benutzername/Passwort beim DWD angefordert werden! Dokumentation)
Fix: Mail Modul gibt als Zeitstempel 0 aus, wenn keine Mails vorhanden
Fix: Modbus TCP kann nun auch die DeviceID nutzen.
Fix: Diverse WebFront/iFront fixes

Fix: Assoziationen mit %f funktionieren jetzt korrekt.
Fix: Thermokon: Berechnen der Temperatur war unter Umständen fehlerhaft.
Fix: MediaPlayer: Soundkarten wurden falsch zugeordnet, wenn z.B. ein USB Soundkarte entfernt wurde.
Für Entwickler:
Neu: HTTP Post für PHP/WebServer ($_POST, $_FILES)
Neu: Enumerations in der Rückgabe von IPS_GetFunctionList

# V2.1->V2.2 (Q1/2010)

## WESENTLICHE ÄNDERUNGEN

In der Version 2.2 von IP-Symcon wurde besonders an der grafischen Darstellung von Daten gearbeitet. So erlaubt das WebFront die Darstellung der in der SQLite-Datenbank aggregierten Daten in einer Stunden-, Tages-, Wochen-, Monats- und Jahresansicht. Eine Live-Demo kann unter webfront.info angesehen werden.
Weitere neue Funktionen sind native Module zum Empfangen von E-Mails über POP3 und IMAP, sowie das Versenden von E-Mails (SMTP) und SMS (Clickatel).
Seit der Version 2.2 ist es möglich, Meldungen (Notifications) an das WebFront zu schicken, um sich über Ereignisse im System, direkt auf der Visualisierung informieren zu lassen. Ein eingebautes On-Screen-Display ermöglicht im WebFront die direkte Eingabe von Textdaten in einer Touchscreen-kompatiblen Art und Weise.
Für eine komplette Auflistung aller neuer und geänderter Funktionen lesen Sie bitte die Liste der Änderungen am Ende des Dokuments.

## MIGRATION



Bitte erstellen Sie auf jeden Fall ein Backup Ihres gesamten IP-Symcon Ordners, um auf die Version 2.1 zurück zu wechseln, sollten wider erwarten Probleme mit der Version 2.2 auftreten.



Falls Sie von der Version 2.0 updaten, lesen Sie bitte vorher die Migrationsanleitung zur Version 2.1:
Migration auf Version 2.1

Bedingt durch den Wechsel von PHP 5.1.6 auf PHP 5.3.1 müssen Sie die ggf. genutzte Extensions aktualisieren. Aktuelle Extension DLLs finden sie in diesem Paket: PHP 5.3.1 Extensions
Um Graphen im WebFront sehen zu können, muss eine 'Graphen Seite' im WebFront Konfigurator erstellt werden.

## LISTE DER ÄNDERUNGEN

Neu: Graphen im WebFront (inkl. Archiv Modul auf SQLite Basis)
Neu: Popups/Notifications im WebFront
Neu: Graphen im Dashboard über WebBrowser (http://localhost:82/dashboard.php)
Neu: On-Screen-Keyboard für das Editieren von Strings/TextBoxen
Neu: Schnellerer Rückkanal
Neu: E-Mail versenden (SMTP)
Neu: E-Mail empfangen (POP3&IMAP)
Neu: Image Grabber (Lädt ein Bild direkt in den Medienpool)
Neu: SMS Modul für Clickatel
Neu: M-Bus Modul
Neu: Heating Control Modul

Neu: Ereignisse direkt in der Instanzkonfiguration erstellbar
Neu: Geräte können direkt über Ereignisse angesprochen werden.
Neu: Module für die EKM-868 (Counter) und ViewJet (Display)
Neu: WebFront nativ auf Englisch %2B Deutsche Übersetzung
Neu: ProJet Display (Input Modul)
Neu: Statusveränderungen von I/O's abfangen (Kern Instanzen->Event Handler)
Neu: FS20 Adresse kann auf doppelte Verwendung geprüft werden
Neu: Filter HTML Checkbox bei Skriptausgabe
Neu: Sys_GetURLContentEx -> Erlaubt Zugriff mit BasicAuthentification/Proxy
Neu: Icons können über ein Auswahlfenster ausgewählt werden.
Neu: Locked Eigenschaft für den TIPSLockButton
Neu: AutoLocked Eigenschaft für den TIPSLockButton
Fix: Dimmer im WebFront bei HomeMatic nicht bedienbar
Fix: 32Bit/Real Werte bei ModBus
Fix: SetTargetValue bei LCN
Fix: n wird bei der TextBox im WebFront erkannt
Fix: Werte/Reaktionsbug im Dashboard
Fix: FHT/HomeMatic FHT im WebFront (haken für "Emulation" aktivieren!)
Fix: Leveljet wieder auslesbar
Fix: LCN_StartFlicker fix
Fix: EnOcean kann mit 32bit unsigned IDs umgehen.
Fix: 1-Wire negative Temperaturen bei manuellem polling falsch
Fix: FHT: Negative Ventilpositionen

# V2.0->V2.1 (Q3/2009)

## WESENTLICHE ÄNDERUNGEN

Die neue Version 2.1 von IP-Symcon fügt viele neue Funktion, aber auch Detailverbesserungen, ein, so dass ein Update auf diese Version empfohlen wird. Besonders im Bereich der Visualisierung ist seit der Version 2.0 viel verbessert worden. Aber auch am Kern von IP-Symcon wurde viel Zeit in eine bessere Stabilität und Speichernutzung investiert. Einige besondere Funktionen werden hier vorgestellt:

Variablenprofile, die eine Variable für die Visualisierung beschreiben, wodurch Sie ohne weiteren Aufwand in der Visualisierung (WebFront) angezeigt werden können. Weiterhin ist es möglich Aktionen zu definieren, um virtuelle Geräte zusammenzustellen. Die Dokumentation der Möglichkeiten finden Sie hier: Variablenprofile

Symbolische Links vereinfachen eine mehrfache Verwendung von Objekten in der Visualisierung. Sie müssen Ihr Panikschalter-Skript nur noch einmal anlegen und können es in beliebig vielen Kategorien verknüpfen.

Dadurch sind auch Übersichtsseiten möglich, um z.B. alle Stati der Fenster anzuzeigen.

Die WebFront Konfiguration erspart ihnen eine mühselige Anpassung irgendwelcher Konfigurationsdateien. Fügen Sie der Visualisierung (WebFront) Ihre eigenen Seiten als Reiter hinzu, verändern Sie den Ausgangspunkt Ihrer Visualisierung oder fügen Sie eigene Widgets hinzu, die Ihnen z.B. das Datum der nächsten Müllentsorgung anzeigen. Alles ein paar Mausklicks entfernt. WebFront konfigurieren

Neue Module für:

- den Energiezähler EM24 von Carlo Gavazzi
- generische Module für ModBus TCP und ModBus RTU
- viele neue Funktionen für das Z-Wave System

Als Vorbereitung für die in der Version 2.2 kommende graphische Darstellung von erfassten Daten können Sie bereits mit dieser Version Daten mitloggen und in einer SQLite-Datenbank abspeichern. Sobald die Version 2.2 verfügbar ist, können Sie ihre Daten sofort graphisch auswerten.

Für eine komplette Auflistung aller neuen und geänderten Funktionen lesen Sie bitte die Liste der Änderungen am Ende des Dokuments.

## MIGRATION



Bitte erstellen Sie auf jeden Fall ein Backup Ihres gesamten IP-Symcon Ordners, um auf die Version 2.0 zurück zu wechseln, sollten wider erwarten Probleme mit der Version 2.1 auftreten.

Löschen Sie **vor** der Installation alle Module (*.dll) aus dem ‘modules’ Ordner von IP-Symcon, da diese nicht mehr gebraucht werden, oder falls Sie welche von Drittanbietern haben, diese einfach nicht mehr kompatibel sind.

Falls Sie PHP Extensions nutzen, müssen Sie diese ab der Version 2.1 in den ‘ext’ Ordner von IP-Symcon kopieren. Alle Extensions in diesem Ordner werden automatisch von IP-Symcon geladen. Die "php.ini" muss dazu nicht mehr angepasst werden, sondern wird bei jedem IP-Symcon Start automatisch neu aktualisiert. Ein anderer Ort ist für Extensions nicht mehr möglich.

__Das Error Reporting ist seit der Version 2.1 standardmäßig auf "E_ALL" gesetzt. Das bedeutet, dass Fehler, wie z.B. nicht definierte Variablen, direkt als Fehler angezeigt werden. Außerdem müssen Arrayindexe als String angesprochen werden. z.B. $a['MeinIndex'] statt $a[meinIndex]. (Auslöser dieser Fehlermeldungen sind meist unsauber programmierte Skripte.) Sie können das Error Reporting in der "php.ini" verändern, wovon wir aber abraten.__
Fehler können Sie, wie immer, in der Projektverwaltung im IP-Symcon Forum melden.

## LISTE DER ÄNDERUNGEN

Kernel
- Keine Veränderung der PHP Version (5.1.6 wird benutzt)*
- php.ini wird nur noch aus dem IP-Symcon Order geladen
- Alle passenden Extensions werden automatisch aus dem 'ext' Ordner geladen.
- Alle Module sind direkt in die ips.exe integriert, wodurch der Speicherbedarf sinkt.
- Eigene Variablen Profile ( Dokumentation )
- Objekte können im WebFront versteckt werden ohne dem Punkt (.) Workaround (Objekt bearbeiten -> Optionen).
- Ereignisse können direkt an Variablen/Instanzen erstellt werden.
- Objekte können sortiert/positioniert werden.
- Symbolische Links können für Instanzen, Variablen, Skripte und Medien erstellt werden.
- Neue Funktion GetValueFormatted, die den Variablenwert als formatierten String zurück gibt. Z.B. 17,3°C
- Suche/Ersetze in Dateien implementiert
- Präfixe haben jetzt Leerzeichen, die entsprechend der DIN 5008 erstellt wurden.

**Ein Wechsel auf 5.2.x oder 5.3.x war geplant, aber aufgrund von Problemen wieder verworfen worden. Fixes für einige relevante PHP-Bugs wurden für PHP 5.3.1 angekündigt, welches für Version 2.2/2.3 erneut getestet wird.*

WebFront
- file_get_contents gegen eine Funktion ausgetauscht (Sys_GetURLContent)
- Wetterseiten werden gecacht, für schnellere Ladezeiten/Ausfallsicherheit
- WebFront mit eigenen Profilen kompatibel
- WebFront zeigt alle Variablen mit Variablenprofil, Skripte und Medien direkt unterhalb von Instanzen an.
- Anzahl der Elemente in einem 'enum' / 'selector' kann in der Config definiert werden.
- Slider für Prozentwerte (mit Möglichkeit zum Deaktivieren)
- Media-Dateien können ausgeklappt werden.
- Media-Dateien werden bei Aktualisierung automatisch im WebFront aktualisiert.
- WebFront Retro, iFront den Neuerungen entsprechend angepasst
- $REMOTE_ADDR, $REMOTE_HOST bei Ausführen einer Aktion verfügbar

Module
- Dummy Modul als Platzhalter für eigene Variablen, die im WebFront angezeigt werden können
- Wago: Native Implementation (Wago DLL wird nicht mehr benötigt)
- ModBus: Modbus TCP, Modbus RTU, Modbus RTU over TCP wird unterstützt.
- Carlo Gavazzi: EM24 Modul hinzugefügt
- Virtual I/O Modul zum Testen eigener Device Module (nur für Entwickler interessant!)
- Webserver: OpenSSL Update auf 0.9.8k
- HomeMatic: Standardprofile für die Wetterstation
- Z-Wave

Weitere Geräteklassen werden unterstützt
Assoziationen können konfiguriert werden.
Parameter können an Geräten gesetzt werden.
Routing Tabelle wird ausgelesen.

Archiv Modul auf SQLite Basis zum Loggen von Variablenwerten

Dashboard

Wenn der Parameter/cache angegeben wird, dann werden alle Bilder in einem cache-Ordner gespeichert, wodurch das erneute Laden wesentlich beschleunigt wird.

Neues Control: TIPSSpinEdit für Integer/Float Werte

Dashboard kann Variablenwerte mit Profilformatierung anzeigen.

$REMOTE_ADDR, $REMOTE_HOST, $IPS_FORM bei Ausführen eines Skriptes verfügbar

Geänderte Funktionen

IPS_SetEventScript (Nimmt nun den auszuführenden Text statt der ID des Skriptes.)

IPS_GetEvent (Index: TriggerVariable -> TriggerVariableID)

IPS_GetStatusVariableIDs -> IPS_GetStatusVariableIdents

Änderungen im SDK

KERNEL_VERSION = 0x020A

Konstanten sind jetzt in 100er Schritten definiert.

Kategoriefunktionen vom ObjectManager in den CategoryManager verschoben

LinkManager für die symbolischen Links

Variablenprofile werden über Strings verknüpft und müssen vorher im VariableManager registriert werden.

Mehr Informationen und das neue SDK finde sie im Entwicklerbereich

# V1.0->V2.0 (Q1/2009)

## FEATUREÄNDERUNGEN

Generelle Änderungen
- Server/Client Design, Windows Dienst + Netzwerkfähige Clientapplikation
- Alle Objekte (Instanzen, Variablen, Skripte, Events, Medien) haben eine eindeutige ID.
- Designer (jetzt Dashboard) direkt in die Clientapplikation eingebettet

Serverseitige Änderungen
- Mehr PHP Befehle zum Eingreifen/Konfigurieren von IP-Symcon
- Pro Skript können mehrere Timer (jetzt: Ereignisse) erstellt werden.
- Ereignisse können zu bestimmten Zeitpunkten gestartet werden (Timer Wizard Ersatz!, Kein Importer!).
- SOAP Schnittstelle (Entwicklerbereich )
- Bei Fehlern (IPS Funktionen) werden diese direkt in PHP Fehlermeldungen ausgegeben.

Clientseitige Änderungen
- Clientanwendung funktioniert über das Netzwerk/Internet
- Grafische Instanzenansicht durch Baumansicht ersetzt
- Multilingual (Deutsch/Englisch)
- Wizards zum Einrichten der Objekte
- Mehr Drag&Drop Möglichkeiten
- Docking der einzelnen Fenster
- Quickfilter an den meisten Stellen vorhanden

WebFront
- WebInterface, welches automatisch vorhandene Geräte visualisiert/schalten lässt
- Eine Variante im iPhone / iTouch Design erlaubt schnellen Zugriff

Modulspezifische Änderungen
- Native LCN Module
- HomeMatic Module
- ProJet/LevelJet Module
- SiLabs USBXpress I/O Modul
- Siemens S7 SPS Module
- FS20TX/RX zusammengelegt; Das FS20 Modul kann nun senden/empfangen und die Timer emulieren.
- Aus FS20RX wird FS20EX. Es erlaubt komfortabel eine Mehrkanalfernbedienung einzurichten.
- Alle WAGO Adressen werden als einzelne Instanzen erstellt. Ermöglicht vernünftige Namensvergabe und Anzeige im WebInterface
- I/O Module reconnecten sich automatisch solange, bis Sie eine Verbindunge bekommen.
- WebServer bietet SSL-Verschlüsselung
- MediaPlayer Modul als Ersatz für das WinampControl. Unterstützt mehrere Soundkarten mit direkter Soundkartenauswahl.
- ISDN Modul kann nun mehrere Anrufe gleichzeitig bedienen. Beachten Sie bitte die neue Funktionsweise. (Siehe ISDN Beispiel )
- Das FS20MS Modul ist jetzt nicht mehr an FS20 gebunden, sondern kann für fast alle Systeme verwendet werden.
- Link: Shutter Control

Wegefallende Module
- DirectX MediaPlayer (eignes MediaPlayer Modul als Ersatz)
- COM/LAN (SerialPort/ClientSocket als Ersatz)

memPIO
M34 USB
RedLab
BTKit
IPPower

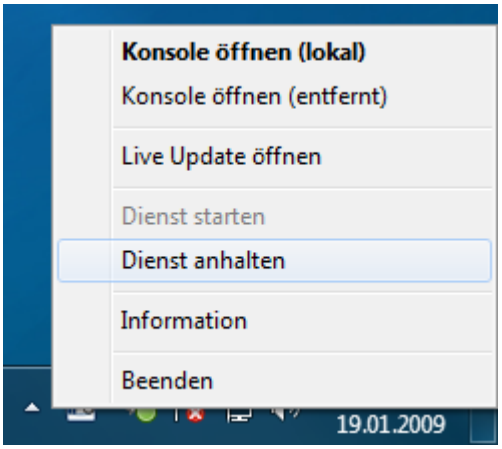
**Weggefallende/Geänderte Funktionen** (nur die wichtigesten Funktionen angegben)

TWZ_ItemExists ( Ersatz: Ereignissfunktionen )
TWZ_DeleteItem ( Ersatz: Ereignissfunktionen )
TWZ_CreateItem ( Ersatz: Ereignissfunktionen )
IPS_UnregisterScript ( Ersatz: Skriptfunktionen )
IPS_RegisterScript ( Ersatz: Skriptfunktionen )
IPS_ScriptExists ( Ersatz: Skriptfunktionen )
IPS_CreateVariable ( Ersatz: Variablenfunktionen )
IPS_VariableExists ( Ersatz: Variablenfunktionen )
IPS_GetVariableType ( Ersatz: s.u. )
IPS_GetVariables ( Ersatz: s.u. )
LOC_GetLocation_Variable ( Ersatz: s.u. )
IPS_GetInstanceIDs ( Ersatz: s.u. )
IPS_GetModuleInstance ( Ersatz: s.u. )
SYS_Ping (2. Parameter entfällt)
IMGP_SendEvent ( Ersatz: IPS_SendMediaEvent )
SNDP_SendEvent ( Ersatz: IPS_SendMediaEvent )
TTS_Speak ( Syntaxänderung + Hinweise: TTS_Speak )

## MIGRATION

Führen Sie bitte ein Backup von Ihrem alten IP-Symcon durch.

Installieren Sie bitte IP-Symcon 2.0 in einen **neuen** Ordner.
Stellen Sie sicher, dass der IP-Symcon Dienst beendet ist. Dies können Sie am grauen TrayIcon erkennen. Einen gestarteten Dienst können Sie durch die Funktion „Dienst anhalten“ beenden.




Sobald der Dienst gestoppt ist, müssen folgende Dateien aus Ihrem alten IP-Symcon Ordner in den neuen IP-Symcon Order kopieren.

scripts*
web* (für WIIPS)
web_data* (für WIIPS)
settings.xml
bricks*
php.ini (Pfade anpassen!)
Alle PHP Extension DLLs, wenn Sie welche installiert haben (z.B. GD2)

Nun können Sie IP-Symcon starten. Alle Variablen, Skripte, Instanzen sollten mit allen Einstellungen importiert werden. Zum Abschluss der Migration sollten Sie Ihre Skripte auf in der neuen Version von IP-Symcon nicht mehr vorhandene Funktionen überprüfen. Ferner müssen Sie alle Zugriffe über Namen durch Zugriffe über IDs ersetzen. Solange Sie aber keine Geräte umbenennen, können Sie diese Aktion auch bei Bedarf durchführen.

## KOMPATIBILITÄTSFUNKTIONEN



Diese Funktionen sollten nur übergangsweise genutzt werden und so schnell wie möglich gegen die neuen, ID basierten Funktionen ausgetauscht werden, um die neuen Möglichkeiten ausnutzen zu können.

```
function IPS_GetVariableType( $varid )
{
if(!is_numeric($varid))
$varid = IPS_GetVariableID($varid);
$var = IPS_GetVariable($varid);
switch($varInvalid Link )
{
case :
return "Boolean";
break;
case 1:
return "Integer";
break;
case 2:
return "Float";
break;
case 3:
return "String";
break;
default:
return "Unknown";
break;
}
}

function IPS_GetUpdateTime($id)
{
$v = IPS_GetVariable($id);
return $v['VariableUpdated'];
}

function FS20_SetPosition($id, $pos)
```

```
{
SC_Move($id, $pos);
}

function IPS_GetVariables()
{
$result = Array();

$varids = IPS_GetVariableList();
foreach($varids as $varid)
{
$var = IPS_GetVariable($varid);
$obj = IPS_GetObject($varid);
$result[] = Array(
"name" => $obj['ObjectName'],
"type" => IPS_GetVariableType($varid),
"value" => GetValue($varid),
"updated" => (int)$var['VariableUpdated']
); }

return $result;
}

function LOC_GetLocation_Variable($varid)
{
if(!is_numeric($varid))
$varid = IPS_GetVariableID($varid);

return 'IP-SYMCON\\'.IPS_GetLocation($varid);
}

function IPS_GetInstanceIDs()
{
return IPS_GetInstanceList();
}

function IPS_GetModuleInstance($instanceid)
{
$instance = IPS_GetInstance($instanceid);
return Array(
"instanceid" => $instance['InstanceID'],
"instancestatus" => $instance['InstanceStatus'],
"moduleguid" => $instanceInvalid Link ,
"packetsreceived" => ,
"packetstransmitted" => ,
"packetsaccepted" => ,
"packetsdropped" => ,
"packetsbuffered" =>
);
}
function IPS_GetModuleInfo ( $mguid ) {
$marray = IPS_GetModule( $mguid );
$m = Array();
$m['name'] = $marray['ModuleName'];

switch ($marray['ModuleType']) {
case :
$m['type'] = 'Core';
break;
```

```
case 1:
$m['type'] = 'I/O';
break;
case 2:
$m['type'] = 'Splitter';
break;
case 3:
$m['type'] = 'Device';
break;
}
$m['guid'] = $mguid;
$larray = IPS_GetLibrary($marray['LibraryID']);
$m['version'] = ($larray['Version'] >> 8) .".". ($larray['Version']
& 0xFF);
return $m;
}
```

# V0.0->V1.0 (Q1/2005)

Erste Version von IP-Symcon wurde veröffentlicht.

# Sicherheit

Es gibt verschiedene Sicherheitsaspekte in IP-Symcon, bei denen ein Passwort oder ein spezieller Dienst für die verschiedenen Abschnitte gesetzt werden kann. Dieser Artikel sollte genau gelesen werden, um etwaigen unerlaubten Zugriff von Außen zu verhindern.



Die Werkseinstellugen sind nur sicher, solange IP-Symcon nicht von extern erreichbar ist. Sollte IP-Symcon von extern erreichbar sein, müssen einige der Einstellungen entsprechend verändert werden!

## FERNZUGRIFF (PASSWORT)

**Werkseinstellung:** *deaktiviert* (sicher)
Der Fernzugriff gestattet den Zugriff auf die Verwaltung von IP-Symcon über die Verwaltungskonsole. Wenn kein Passwort gesetzt ist, ist kein direkter Zugriff außerhalb des lokalen Netzwerks möglich.
Weitere Informationen: Fernzugriff festlegen

## CONNECT DIENST

**Werkseinstellung:** *aktiviert* (sicher)
Bei aktiver Subskription kann der Connect-Dienst via Connect Control genutzt werden, um bequem von extern auf die Verwaltungskonsole als auch die Visualisierungen zugreifen zu können.
Ein Zugriff auf das die Visualisierung ist nur Möglich, wenn vorher ein Passwort vergeben wurde, oder die Visualisierung explizit als öffentlich definiert wurde.
Ein Zugriff auf die Konfiguration mithilfe der Verwaltungskonsole ist nur möglich, wenn ein Fernzugriffspasswort festgelegt wurde.
Weitere Informationen: Connect Control

## VISUALISIERUNG (PASSWORT)

**Werkseinstellung:** *nicht gesetzt* (sicher)
Die Visualisierungen (Kachel-/WebFront-Visualisierung) sind in der Werkseinstellung im lokalen Netzwerk ohne Passwort zugänglich. Wird ein Passwort gesetzt, so ist sind die webbasierte Visualisierung und die mobilen Apps durch dieses Passwort geschützt. Sofern die Visualisierung über den Connect-Dienst oder von extern erreichbar sein soll, muss ein Passwort oder explizit ein Zugriff ohne Passwort konfiguriert worden sein.
Weitere Informationen: Kachel-Visualisierung Sicherheit / WebFront-Visualisierung Sicherheit



Anfragen aus folgenden Netzen werden als intern betrachtet (Private IP Adressen). Diese erfordern somit kein Kennwort, sofern die Option "Erfordere Passwort nur bei externen Zugriff (Internet)" aktiv ist. Wenn jedoch die eingehenden Verbindungen durch einen internen NAT verborgen werden, sollte diese Funktion nicht aktiviert werden.

## USER ORDNER (BENUTZERNAME/PASSWORT)

**Werkseinstellung:** *nicht gesetzt* (unsicher)
Sofern eigene Inhalte in dem User Ordner gelegt wurden, empfiehlt es sich, diese Inhalte zu schützen. Generell sollte die Nutzung vom User Ordner vermieden werden.
Weitere Informationen: "user"-Ordner Sicherheit

## SYMOS (PASSWORT)

**Werkseinstellung:** *nicht gesetzt* (sicher)
Die SymBox besitzt das eigene Betriebssystem SymOS. Der Zugriff auf dessen Weboberfläche kann via Passwort geschützt werden. Dazu kann in der Weboberfläche des SymOS ein Passwort vergeben. Sofern die SymOS Oberfläche von extern verfügbar ist, sollte dringend ein Passwort eingerichtet werden.
Weitere Informationen: SymBox Betriebsanleitung PDF - Kapitel 4.4.6. Sicherheit

## SYMOS CONNECT

**Werkseinstellung:** *nicht gesetzt* (sicher)
Via SymOS Connect kann von außerhalb auf das SymBox Betriebssystem SymOS zugegriffen werden. Der Zugriff von außerhalb ist auch bei aktivierten SymOS Connect nur mit einem gesetzten SymOS Passwort möglich.



Um IP-Symcon den Zugriff auf bestimmte Funktionen außerhalb zu gewähren ist es gegebenenfalls nötig diese in der Firewall einzurichten. Weitere Informationen können unter Firewall gefunden werden.

# Datensicherung

Eine Datensicherung gibt es in zwei verschiedenen Formen.
**Backup:** Ein Backup was alle relevanten Benutzerdaten enthält, sowie auch Mediendateien, Skripte, Datenbank etc.
**settings.json:** Die Konfigurationsdatei ist das Kernelement einer jeden IP-Symconinstallation. Diese wird automatisch gesichert.

## BACKUP

Ein Backup ist eine vollständige Datensicherung. Diese wird nicht automatisch vollzogen. Wie ein Backup erstellt wird und eingespielt wird ist unter Backup erstellen und Backup einspielen nachzulesen.

## AUTOMATISCHE SICHERUNG DER KONFIGURATIONSDATEI (SETTINGS.JSON)

In IP-Symcon werden alle wichtigen Einstellungen in der Datei “settings.json” innerhalb vom IP-Symcon Ordner gesichert. Diese Datei ist also das Kernelement der Konfiguration. Damit im Fehlerfall, auf eine Sicherung zurückgegriffen werden kann, speichert IP-Symcon täglich um 00:00 Uhr eine aktuelle Kopie der Einstellungen im Ordner "backup". Die maximale Anzahl von Sicherungen beträgt 25 Stück. Sofern dieses Limit überschritten wird, wird eine alte Sicherung gelöscht.
Die Sicherungen werden automatisch mit dem Namen “settings1234567890.json” abgespeichert, wobei "1234567890" durch die aktuelle Unix-Timestamp ersetzt wird. Die höchste Zahl ist somit die aktuellste Sicherung.



Die automatischen Sicherungen der Konfigurationsdatei ersetzt kein regelmäßiges Backup des gesamten Systems. Siehe Backup erstellen

## KONFIGURATIONSDATEI (SETTINGS.JSON) IM FEHLERFALL WIEDERHERSTELLEN

Ein Fehlerfall kann mehrere Ursachen haben z.B.:
- Ungewollte Änderungen (z.B. falsche Instanzen, Variablen gelöscht,...)
- Die Einstellungen sind unerwartet “verschwunden”, “gelöscht” oder “leer”. (Dies bei 99 Prozent der Fälle, auf ein unerwartetes Beenden der Software zurückzuführen, wie z.B. einen Stromausfall.)

Um eine Sicherung der Konfigurationsdatei einzuspielen sind folgende Schritte notwendig:
- Beendigung des IP-Symcon Dienstes
- Kopie einer beliebigen (z.B. die mit der höchsten Zahl im Dateinamen) "settings1234567890.json" aus dem "Backup"-Ordner in den Hauptordner
- Entfernen der aktuellen "settings.json"
- Die kopierte "settings1234567890.json" in "settings.json" umbenennen
- Starten des IP-Symcon Dienstes

Falls die aktuelle Wahl der Konfigurationsdatei-Sicherung nicht korrekt funktioniert, kann eine andere Sicherung probiert werden. Sollte keine der Sicherungen korrekt funktionieren, sollte ein Hilfe-Thema im Forum erstellt werden. Hilfegesuche müssen dabei **mindestens ein Logfile** im Anhang haben. Anfragen ohne Logfile können nicht bearbeitet werden.

## WIE OFT WIRD WAS GESCHRIEBEN?

IP-Symcon besitzt seit Version 4.0 verschiedene Mechanismen um die Schreibzyklen zu minimieren.
Dies ist erforderlich, wenn das IP-Symcon auf Speichermedien betrieben wird, die eine Abnutzung aufweisen.
Dies ist z.B. bei einem Raspberry Pi mit einer SD-Karte der Fall.

### MEDIEN UND SETTINGS.JSON

Generell gilt, dass die settings.json standardmäßig zyklisch alle 10 Minuten (siehe Spezialschalter) und beim Start/Stop des Dienstes gespeichert wird.
Medien Dateien (z.B. vom Image Grabber) sind gecached und werden nur beim Start/Stop geschrieben.

### VARIABLEN ROHDATEN UND AGGREGATION

Variablen-Rohdaten, welche im Archiv gespeichert werden, schreibt IP-Symcon jede Minute auf die Festplatte.
Da dies stetig wachsende Datensätze sind, ist dies auch so gewollt. Wiederrum werden die Aggregationsdaten nur geschrieben, wenn ein Aggregationszeitraum vollständig ist. Die jeweiligen unvollständigen Aggregationszeiträume werden beim Start aus den Variablen-Rohdaten wiederhergestellt.
Deswegen dauert der Start bei vielen archivierten Variablen auch etwas länger.

### LOGFILES

Logfiles, welche den größten Teil an Schreibzyklen bewirken, werden je nach System anders behandelt.
Normalerweise landen diese auf der Festplatte.
Das SymOS auf der SymBox mappt jedoch /var/log/symcon auf eine RAM-Drive, wodurch das Speichermedium nicht unnötiog abgenutzt wird und die Langlebigkeit erweitert wird.

# Backup einspielen

Ein Backup kann eingespielt werden, wenn es zu einem nicht reparierbaren Zustand gekommen ist. Ein Backup kann auch genutzt werden, um von einer Plattform auf eine Andere zu wechseln.

## AUF EINE ANDERE PLATTFORM WECHSELN

Bei einem Wechsel auf eine andere Plattform wird die Struktur des Backups genutzt, welches plattformunabhängig für alle gleich ist.



Bei einem Wechsel von einer IP-Symcon Version <= 3.4 auf IP-Symcon 4.0 oder höher, muss die Datenbank über den Windows-Installer in das neue CSV Format konvertiert werden. Erst danach kann auf ein anderes System migriert werden.

## BACKUP EINSPIELEN

SymOS (SymBox)
Windows (Desktop/Server)
MacOS
Linux (Ubuntu)
Raspberry Pi
Docker/Synology/QNAP

### SYMOS - BACKUP EINSPIELEN

Auf der Weboberfläche kann das Backup unter dem Menüpunkt "Einrichten -> Wiederherstellen" ausgewählt werden. Im Anschlusss muss das IP-Symcon (NICHT SymOS) neu gestartet werden.
Weitere Informationen befinden sich in der Anleitung der SymBox unter 4.4.4 Backup

### WINDOWS - BACKUP EINSPIELEN

Sofern die Installation nicht über das Setup durchgeführt wurde, sondern nur auf das System kopiert wurde, ist es möglicherweise notwendig, dass folgende Restributables installiert werden:
Visual Studio 2015/2017/2019 C++ Redistributable x64 (Direktdownload )
Bonjour64 Installer (Direktdownload )
Folgende Schritte sind zu beachten:
Den IP-Symcon Dienst zuerst beenden. (Rechtsklick auf das Tray-Icon -> "Dienst beenden")
Die backup.zip-Datei in den Ordner C:\ProgramData\Symcon entpacken
Der Ordner kann am einfachsten über "Rechtsklick auf das Tray-Icon -> Information -> Serverdaten -> Öffnen" geöffnet werden
Im Anschluss den Dienst wieder starten. (Rechtsklick auf das Tray-Icon -> "Dienst starten")

### MACOS - BACKUP EINSPIELEN

Den IP-Symcon Dienst zuerst beenden.
Die backup.zip-Datei in den Ordner "/Library/Application Support/Symcon" verschieben und via Doppelklicken entpacken.
Das Archiv löschen und im Anschluss einmal den Dienst neustarten.

### LINUX - BACKUP EINSPIELEN

Den IP-Symcon Dienst zuerst beenden.

```
sudo /etc/init.d/symcon stop
```

Die im Home-Verzeichnis liegende backup.zip-Datei in den Ordner "/var/lib/symcon/" entpacken.

```
sudo unzip backup.zip –d /var/lib/symcon/
```

Im Anschluss den Dienst wieder starten.

```
sudo /etc/init.d/symcon start
```

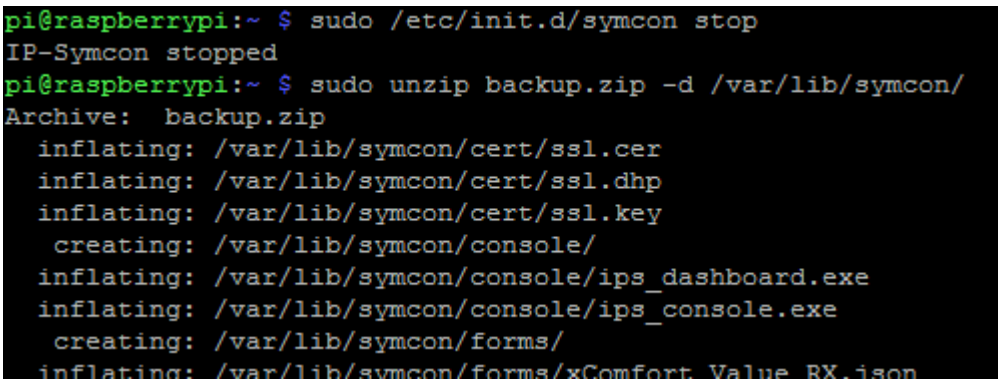
### RASPBERRY PI - BACKUP EINSPIELEN

Den IP-Symcon Dienst zuerst beenden.

```
sudo /etc/init.d/symcon stop
```

Die im Home-Verzeichnis liegende backup.zip-Datei in den Ordner "/var/lib/symcon/" entpacken.

```
sudo unzip backup.zip –d /var/lib/symcon/
```




Im Anschluss den Dienst wieder starten.

```
sudo /etc/init.d/symcon start
```

### DOCKER/SYNOLOGY/QNAP - BACKUP EINSPIELEN

Der IP-Symcon Container muss zuerst gestoppt werden.
Dieser kann über die Containerverwaltung des jeweiligen Systems gestoppt werden.
Anschließend muss in das gemountete IP-Symcon Verzeichnis gewechselt werden, welches auf den Ordner "/var/lib/symcon/" verweist.
Auf welches Volume der Ordner "gemounted" wurde, kann den Einstellungen des Abbilds entnommen werden. Nach Installationsanleitung ist dies standardmässig im Format "/Symcon/[MeinIPSymcon]/Data".
Den Inhalt des Backups entpacken und in das Verzeichnis kopieren.
Im Anschluss den Container wieder starten.

## TIPPS

Folgende Dinge sollten hinterher überprüft und ggf. angepasst werden:

- Für Push-Notifications muss der Server reaktiviert werden
- Absolute Pfade bei Skripten und Mediendateien
- Schnittstellenbelegungen (z.B. COM-Ports) auf dem neuen System
- IP-Adressen, wie z.B. ‘Ereignisserver’ bei der HomeMatic CCU oder ‘WebServer’
- Notwendige Zusatzprogramme, Treiber oder Dienste (z.B. *BidCOS*) installieren

# Backup erstellen

Ein Backup ist eine vollständige Datensicherung aller Benutzerdateien. Dies sollte in regelmässigen Abständen gemacht werden.

## BACKUP ERSTELLEN

SymOS (SymBox)
Windows (Desktop/Server)
MacOS
Linux (Ubuntu)
Raspberry Pi
Docker/Synology/QNAP

### SYMOS - BACKUP ERSTELLEN

Auf der Weboberfläche ist unter dem Menüpunkt "Einstellungen" der Button "Backup". Die Ordnerstruktur wird automatisch erstellt und das backup.zip heruntergeladen. Weitere Informationen befinden sich in der Anleitung der SymBox unter 4.4.4 Backup .

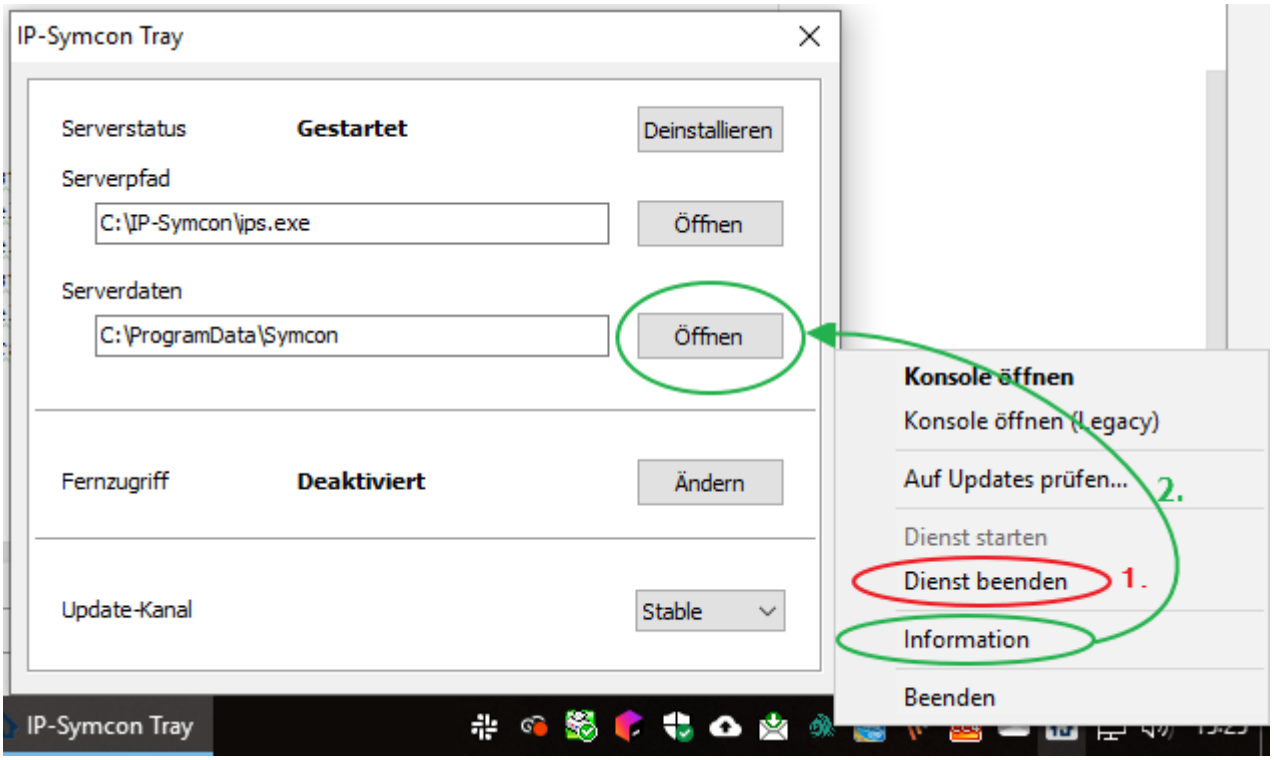
### WINDOWS - BACKUP ERSTELLEN

Es muss der gesamte Inhalt des Ordners C:\ProgramData\Symcon als backup.zip gepackt werden.

Über das Tray Icon den Dienst beenden.
Dann den Dataordner über den "Tray Icon -> Information -> Serverdaten öffnen" öffnen und alle Ordner und Dateien markieren.
Via "Rechtsklick -> Senden an -> zip-komprimierter Ordner" wird im IP-Symcon Ordner eine .zip Datei erstellt, welche noch korrekt in backup.zip umbenannt werden muss.
Diese kann dann zur Sicherung einfach verschoben oder kopiert werden.
Über das Tray Icon den Dienst wieder starten.



Sollte es beim Erstellen der Zip-Datei zu der Meldung kommen "Mindestens ein leeres Verzeichnis [...] nicht hinzugefügt werden", kann diese ignoriert werden. Fehlende leere Ordner erstellt IP-Symcon bei Bedarf selbst.




## MACOS - BACKUP ERSTELLEN

Es muss der gesamte Inhalt des Ordners "/Library/Application Support/Symcon" als backup.zip gepackt werden.
Den IP-Symcon Dienst beenden.
Dann über "Finder -> Gehe zu -> Gehe zum Verzeichnis" auswählen und den Pfad "/Library/Application Support/Symcon" eintragen.
Anschließend in dem geöffneten Ordner alle benötigten Verzeichnisse und Dateien markieren (im Normalfall alle).
Über das Menü "Ablage -> X Objekte komprimieren" wird eine Archiv.zip auf den Schreibtisch erstellt.
Diese muss noch in backup.zip umbenannt werden.
Diese kann dann zur Sicherung einfach verschoben oder kopiert werden.
Den IP-Symcon Dienst starten.

## LINUX - BACKUP ERSTELLEN

Es muss der gesamte Inhalt des Ordners "/var/lib/symcon" gezippt werden.
Falls nötig muss zip installiert werden

```
sudo apt-get install zip
```

Den IP-Symcon Dienst zuerst beenden.

```
sudo /etc/init.d/symcon stop
```

Anschließend muss in das Symcon Verzeichnis gewechselt werden, um zu das Verzeichnis zu packen.

```
cd /var/lib/symcon/ //Wechselt in das IP-Symcon Verzeichnis
zip -r ~/backup.zip * //Packt das gesamte Verzeichnis * in die Datei backup.zip.
```

Die backup.zip befindet sich dann im Home-Verzeichnis. Von dort aus kann diese beliebig kopiert oder verschoben werden.
Den IP-Symcon Dienst wieder starten.

```
sudo /etc/init.d/symcon start
```

## RASPBERRY PI - BACKUP ERSTELLEN

Es muss der gesamte Inhalt des Ordners "/var/lib/symcon" als backup.zip gepackt werden.
Falls nötig muss zip installiert werden

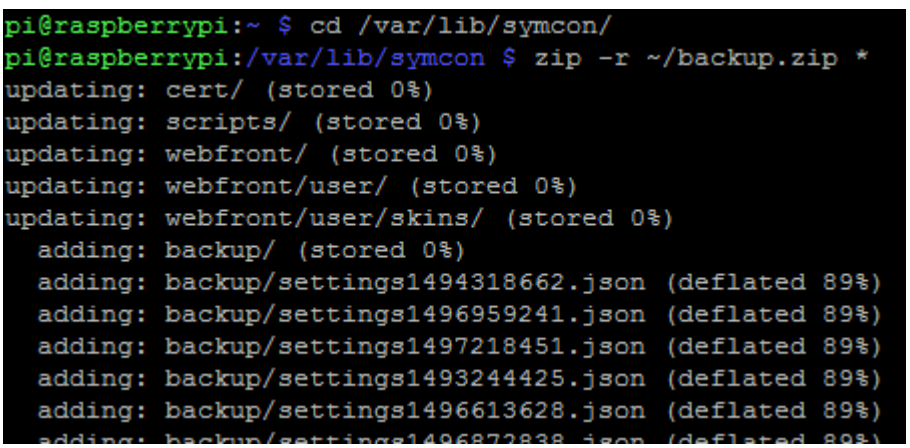
```
sudo apt-get install zip
```

Den IP-Symcon Dienst zuerst beenden.

```
sudo /etc/init.d/symcon stop
```

Anschließend muss in das Symcon Verzeichnis gewechselt werden, um zu das Verzeichnis zu packen.

```
cd /var/lib/symcon/ //Wechselt in das IP-Symcon Verzeichnis
zip -r ~/backup.zip * //Packt das gesamte Verzeichnis * in die Datei backup.zip.
```

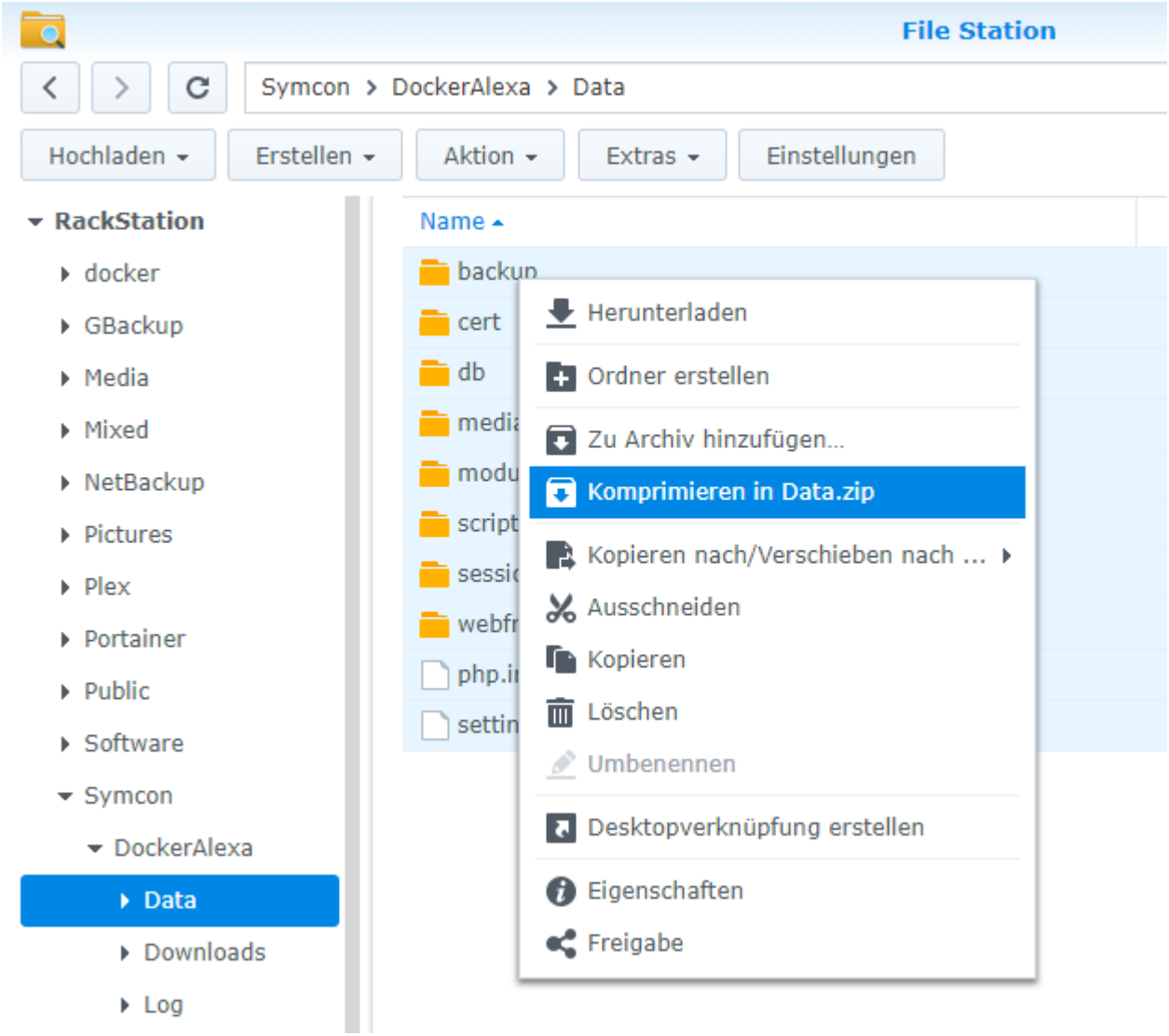


Die backup.zip befindet sich dann im Home-Verzeichnis. Von dort aus kann sie beliebig kopiert oder verschoben werden.
**Den IP-Symcon Dienst wieder starten.**

```
sudo /etc/init.d/symcon start
```

## DOCKER/SYNOLOGY/QNAP - BACKUP ERSTELLEN

Es muss der gesamte Inhalt des Ordners "/var/lib/symcon" als .zip gepackt werden.
Als Beispiel dient hier ein Synology System.
**Den Container von IP-Symcon zuerst stoppen.**
Es muss der Container über die Containerverwaltung des jeweiligen Systems gestoppt werden.
**Anschließend muss in das gemountete Symcon Verzeichnis gewechselt werden, um das Verzeichnis zu packen.**
Auf welches Volume der Ordner "gemounted" wurde, kann den Einstellungen des Abbilds entnommen werden. Nach Installationsanleitung ist dies standardmässig im Format "/Symcon/[MeinIPSymcon]/Data".
Anschließend alle Inhalte markieren und via Rechtsklick "Komprimieren".



Die .zip Datei befindet sich dann auf dem Dateisystem und kann einfach heruntergeladen werden.
Den Container von IP-Symcon wieder starten.

# Grundlagen

## OBJEKTE

Bei IP-Symcon gibt es sieben verschiedene Objektarten.
**Objekte können Kategorien, Instanzen, Variablen, Automationen, Ereignisse, Medien oder Links sein.**

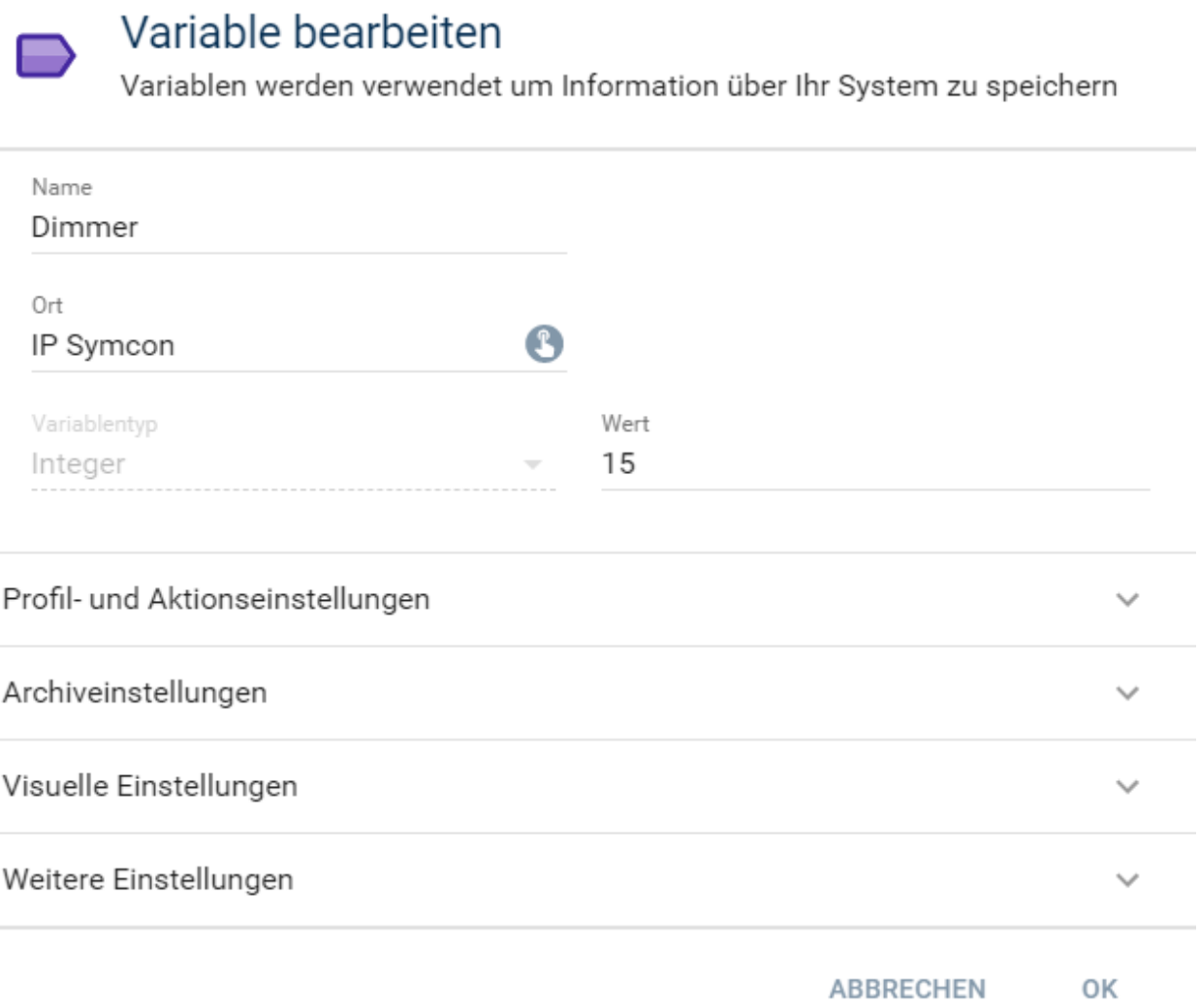


Jedes Objekt besitzt eine feste und nicht änderbare ID (Identifikationsnummer). Dadurch ergeben sich viele Vorteile:

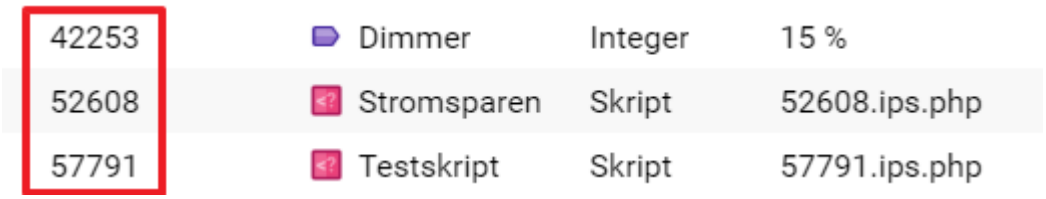
- Jedes Objekt ist einzigartig und eindeutig bestimmbar.
- Der Name kann jederzeit frei verändert werden.
- Größere Projekte können schneller abgearbeitet werden.
- Umfangreiche Funktionen der Objektverwaltung
- Performante Verknüpfung innerhalb der Software

## EIGENSCHAFTEN EINES OBJEKTS

Die Eigenschaften bzw. Einstellungen eines Objekts können durch einen Doppelklick oder via "Rechtsklick" -> "Objekt bearbeiten" aufgerufen werden.




ObjektID
Jedes Objekt bei IP-Symcon besitzt eine Identifikationsnummer (ObjektID), die einmalig und nicht veränderbar ist. Somit ist jedes Objekt eindeutig bestimm- und ansprechbar.




Name
Ist der Name des Objekts. Der Name wird auch zur Darstellung in den Visualisierungen genutzt.
Ort
Der Ort ist das übergeordnete Objekt im Objektbaum und bestimmt sowohl die Position im Objektbaum als auch im WebFront.

Bestimmte Instanzen, wie z.B. I/O, Konfiguratoren, werden in den jeweiligen Bereichen I/O Instanzen, Konfigurator Instanzen angelegt. Diese Instanzen können nicht verschoben werden.

### VISUELLE EINSTELLUNGEN

Diese Einstellungen steuern das Verhalten im WebFront.
**Icon**
Symbol welches in den Visualisierungen genutzt werden soll. Weitere Informationen sind unter Icons einsehbar.
**Objekt anzeigen**
Regelt ob ein Objekt in den Visualisierungen angezeigt werden soll oder versteckt bleiben soll.
**Objekt aktivieren**
Diese Einstellung steuert ob ein Objekt überhaupt aktiv ist oder nicht. Ist es deaktiviert, so wird das Objekt in den Visualisierungen ausgegraut und ist nicht steuerbar.

### WEITERE EINSTELLUNGEN

Zusätzliche Einstellungen, welche optional sind.
**Beschreibung**
Eine Beschreibung des Objekts. Kann genutzt werden um wichtige Notizen zu hinterlassen und somit eine direkte Beschreibung zu liefern. Die Beschreibung kann unter dem Punkt "Weitere Einstellungen" gefunden werden.
**Ident**
Als zusätzlicher Identifikator besitzen Objekte einen Ident. Die Besonderheit des Ident im Gegensatz zum Namen ist, dass innerhalb einer Kategorie in der logischen Baumansicht jeder Ident einmalig ist. Dieser ist z.B. in der Modulentwicklung wichtig.

## KONTEXTMENÜ

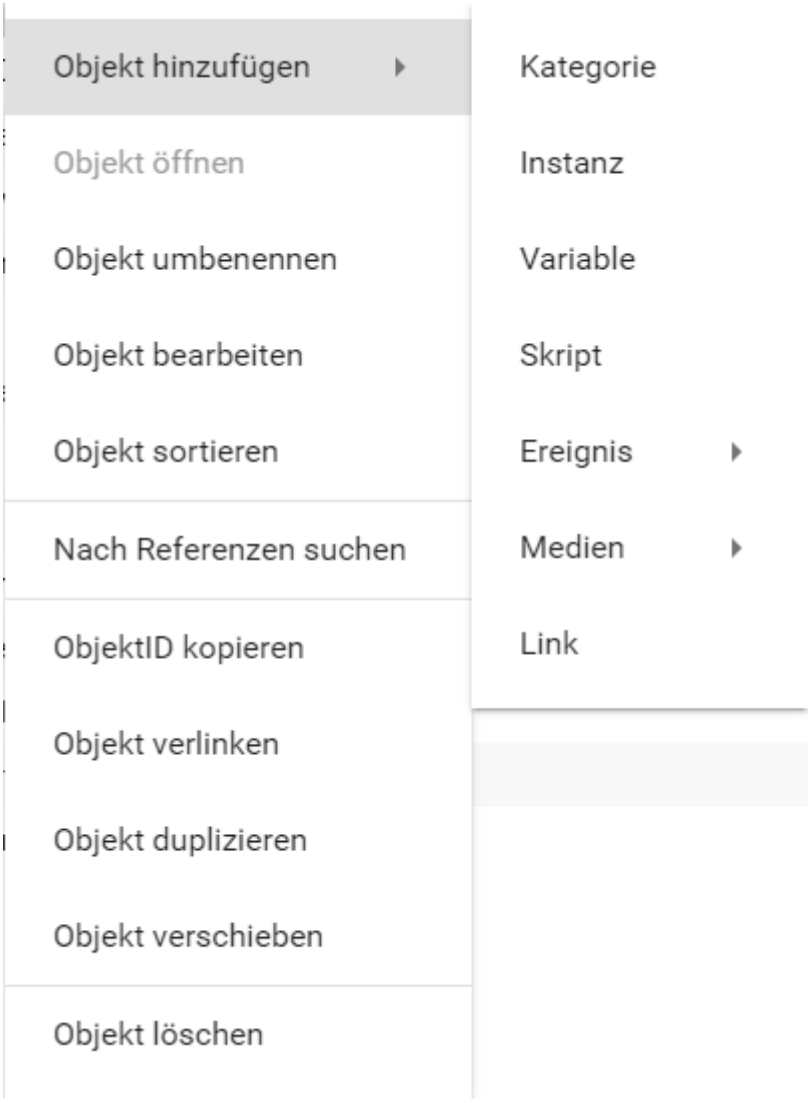
Das Kontextmenü im Objektbaum beinhaltet verschiedene Funktionalitäten.



Jede Objektart hat, zusätzlich zu den beschriebenen Funktionalitäten, typspezifische und erweiternde Optionen. Siehe: Objektartspezifische Kontexteinträge

### OBJEKT HINZUFÜGEN

Fügt ein neues Objekt hinzu. Es öffnet sich ein Dialog bei dem Schrittweise alle wichtigen Einstellungen getätigt werden können.
Weiter Informationen sind auch unter Objekt hinzufügen nachzulesen.




## OBJEKT ÖFFNEN

Sofern das Objekt eine objektspezifische Aktion besitzt, wird diese aufgerufen. Dies bedeutet bei Instanzen wird die Konfiguration geöffnet, bei Skripten der Skripteditor, bei Medien wird eine Vorschau angezeigt. Sollte es keine objektspezifische Aktion geben wird die Option deaktiviert und ausgegraut dargestellt.

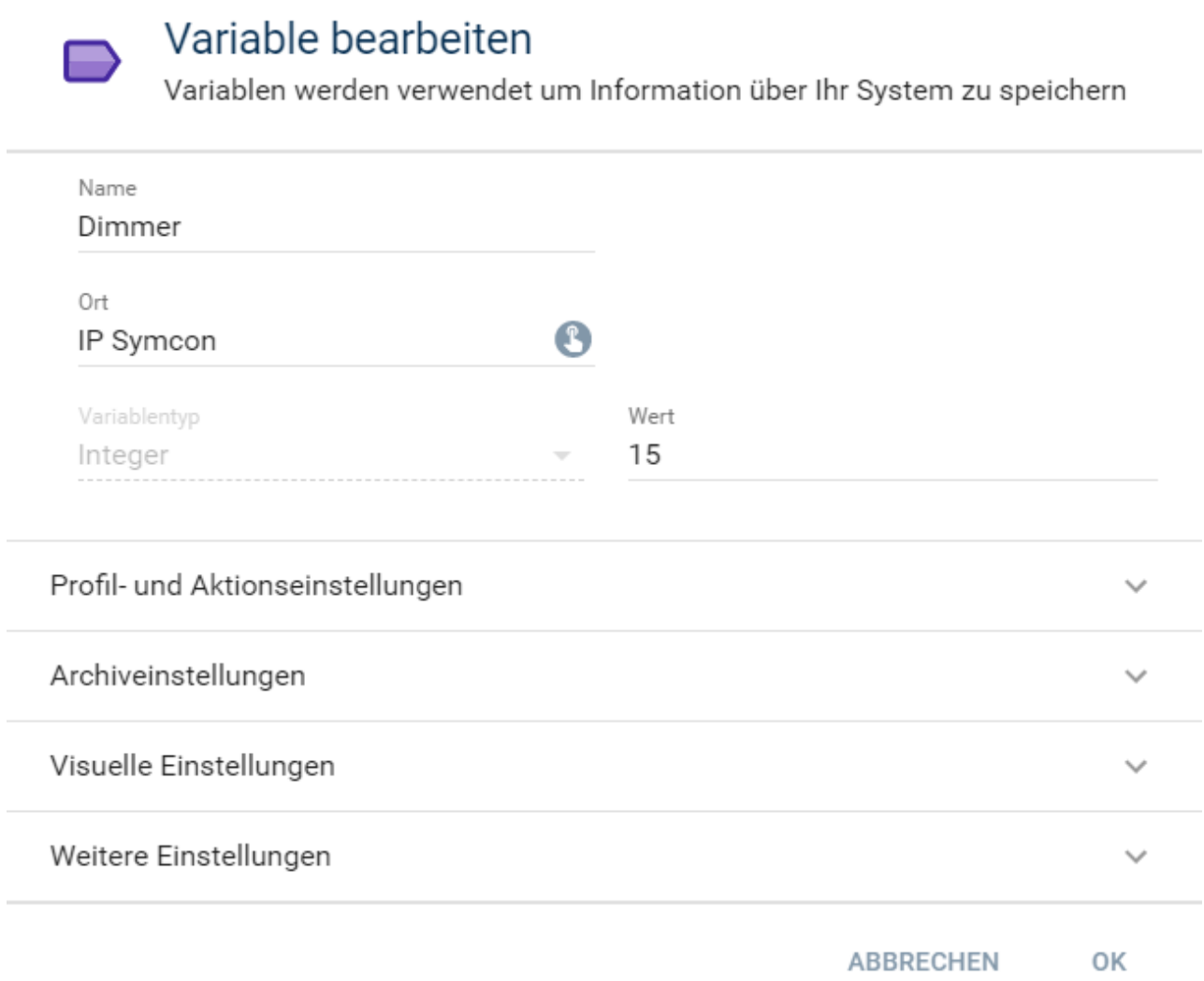
Ein Doppelklick auf ein Objekt öffnet den Dialog "Objekt öffnen". Sollte die Option ausgegraut und somit keine objektspezifische Aktion vorhanden sein, öffnet der Doppelklick den Dialog "Objekt bearbeiten".

## OBJEKT UMBENENNEN

Ändert den Namen des Objekts. Lässt die ObjektID und den Ident unberührt. Es kann unter allen Objekten mehrfach den gleichen Namen geben.

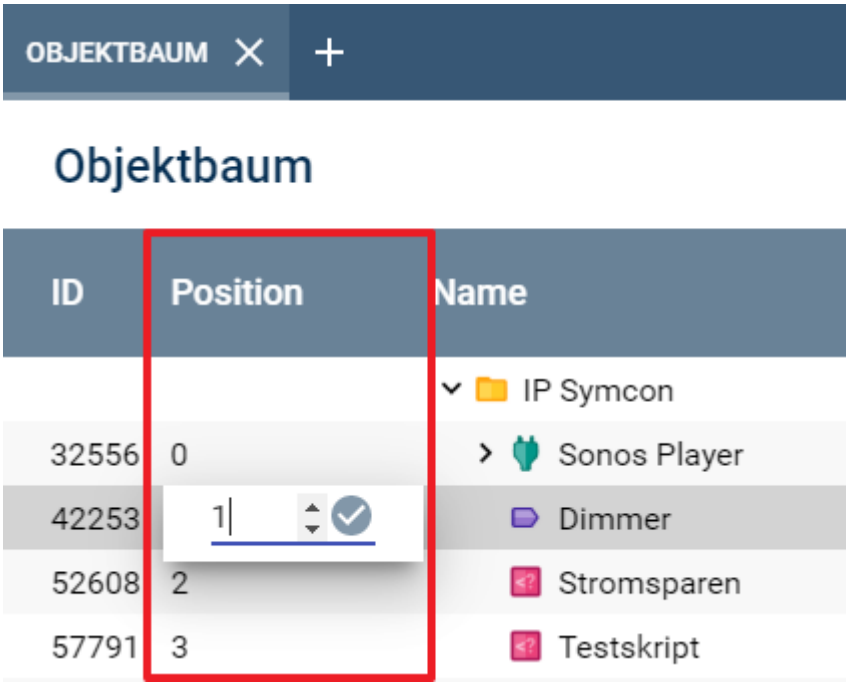
## OBJEKT BEARBEITEN

Jedes Objekt hat die Option "Objekt bearbeiten". Es öffnet den objektabhängigen Dialog. Hier können verschiedene Informationen entnommen und Einstellungen gemacht werden.




## OBJEKT SORTIEREN

Bestimmt die Reihenfolge der Objekte über den Positionswert. Einhergehend wird auch die Position in den Visualisierungen darüber gesteuert.




Über die "Spalten"-Option kann die Positionsspalte dauerhaft eingeblendet werden. Mit dieser können Objekte via Doppelklick auf den Positionswert schneller einsortiert werden, ohne jedesmal den "Objekt sortieren"-Menüpunkt anzuklicken.

### OBJEKTID KOPIEREN

Dies kopiert die ObjektID des Objekts in die Zwischenablage. Dies ist nützlich um ID's in Skripten einzufügen. ("Rechtsklick->einfügen" oder "STRG + V" im Editor).

### OBJEKT VERLINKEN

Erstellt von dem Objekt eine Verlinkung. Es können mehrere Links auf das selbe Objekt verweisen. Im Dialogfenster kann die Position des Link-Objekts ausgewählt werden.




### OBJEKT DUPLIZIEREN

Erstellt ein identisches Objekt mit eigener einmaligen ObjektID.

### OBJEKT VERSCHIEBEN

Öffnet einen Dialog um den Ort des Objekts auszuwählen unter dem es platziert wird.

### OBJEKT LÖSCHEN

Löscht das Objekt und alle seine Unterobjekte, falls vorhanden.

## OBJEKTARTSPEZIFISCHE KONTEXTEINTRÄGE

### NACH REFERENZ SUCHEN (ALLE AUSSER KATEGORIEN)

Öffnet einen Dialog, welcher entweder besagt, dass es keine Referenzen gibt oder alle gegebenen Referenzen auflistet.
Folgende Kriterien werden je nach Objektart durchsucht:

| OBJEKTART | WO/WAS WIRD NACH REFERENZ/ID GESUCHT? |
|---|---|
| Variable | Es werden die ID's für "VariableAction" und "VariableCustomAction" durchsucht. |
| PHP-Skript | Der Inhalt eines PHP-Skriptes wird nach der ID durchsucht. |
| Ereignis | Der Inhalt und der Auslöser werden nach der ID durchsucht. |
| Link | Die TargetID/Wert wird durchsucht. |
| Instanz | IDs von möglichen Objekten, welche auf der Konfigurationsseite ausgewählt wurden. |

### BEFEHLE TESTEN (NUR INSTANZEN)

Öffnet einen Dialog zum Testen der wichtigsten Aktionen einer Instanz.
**Für Experten:** Mit der Tastenkombination Strg+C kann der eingestellte Befehl als PHP-Code kopiert werden und im Skripteditor eingefügt werden.

### VARIABLE VERÄNDERN (NUR VARIABLEN)

Öffnet den Dialog zum Setzen eines Variablenwertes.

### AUTOMATION AUSFÜHREN (NUR AUTOMATIONEN)

Führt das Skript aus ohne es zu öffnen.
Mögliche Ausgaben werden in einem Dialog ausgegeben.

### ZUM QUELLOBJEKT SPRINGEN (NUR LINKS)

Springt innerhalb des Objektbaums zum Ursprungsobjekt des Links.

# Kategorien

Um eine einfache Bedienung der verschieden agierenden Geräte zu gewährleisten und um einen Überblick über die Vielzahl der Instanzen zu behalten, bietet IP-Symcon eine Funktion zum einsortieren. Im Objektbaum können die Instanzen der Geräte verschiedenen Kategorien zugeordnet werden.

## EINRICHTUNG

Über das "+" unten Rechts im Objektbaum oder über "Rechtsklick" -> "**Objekt hinzufügen**" und folgend "Kategorie" kann eine Kategorie erstellt werden. Im folgenden Dialog wird die neue Kategorie konfiguriert und unter anderem Name und Ort definiert. Zur Namensgebung eignet sich beispielsweise der Bereich in dem sich einzusortierende Geräte befinden. *In einem Einfamilienhaus bietet sich z.B. die Bezeichnung des Stockwerks an.*



Es ist unbedingt zu beachten, dass beim Löschen einer Kategorie alle Unterkategorien und alle hier eingeordneten Geräte im Objektbaum ebenfalls gelöscht werden.

## KONFIGURATION

Die erstellte Kategorie befindet sich jetzt im Objektbaum. Auf diese Art und Weise können noch weitere Oberkategorien erstellt werden, z.B. Erdgeschoss, 1. Stock, 2. Stock...
Über einen Rechtsklick im Objektbaum auf eine ausgewählte Oberkategorie und "Objekt hinzufügen -> Kategorie hinzufügen" können Kategorien erstellt werden, bei denen die Oberkategorie standardmäßig als Ort eingestellt ist. Dann kann ein Name (z.B. Wohnzimmer) für die Unterkategorie ausgewählt werden. Dies lässt sich beliebig oft auch für jegliche Unterkategorie wiederholen.
Die Reihenfolge der Kategorien kann durch "Rechtsklick->Objekt sortieren" bestimmt werden.
Die Verschachtelung ineinander ist durch einfaches Drag and Drop möglich.



Es wird empfohlen, die Bezeichnung der **Kategorien/ Unterkategorien/Geräte** wie folgt zu verwenden:
**Etage / Zimmer / Gerät**

## TIPPS & TRICKS

Zum Überblick sollte diese Bezeichnungsart von Anfang an genutzt werden, da im Laufe der Zeit immer mehr Geräte hinzukommen.
Jeder Gerätename sollte nur sich selbst erklären und keine Komponenten enthalten, die vorher schon benannt wurden.
Z.B.: Erdgeschoss / Wohnzimmer / Deckenlampe (Nordseite)

## WEBFRONT

Ein weiterer Vorteil an dieser Struktur ist die automatische Eins zu Eins Visualisierung innerhalb des WebFronts.
Sofern die angezeigte Oberkategorie sichtbare Unterkategorien hat, wird automatisch eine Kategorie-Navigationsleiste angezeigt.
Wird über die Navigation eine Unterkategorie ausgewählt, so wird im Objektbereich deren Inhalt dargestellt.
Besitzt eine so angezeigte Kategorie ebenfalls sichtbare Unterkategorien, so wird die Navigation automatisch erweitert.
Per "Link"-Objekt verknüpfte Kategorien werden ebenfalls in der Navigationsleiste angezeigt. Über die Eigenschaft “Zeige Navigation” lässt sich die Kategorieleiste bei Bedarf ausblenden.

# Instanzen

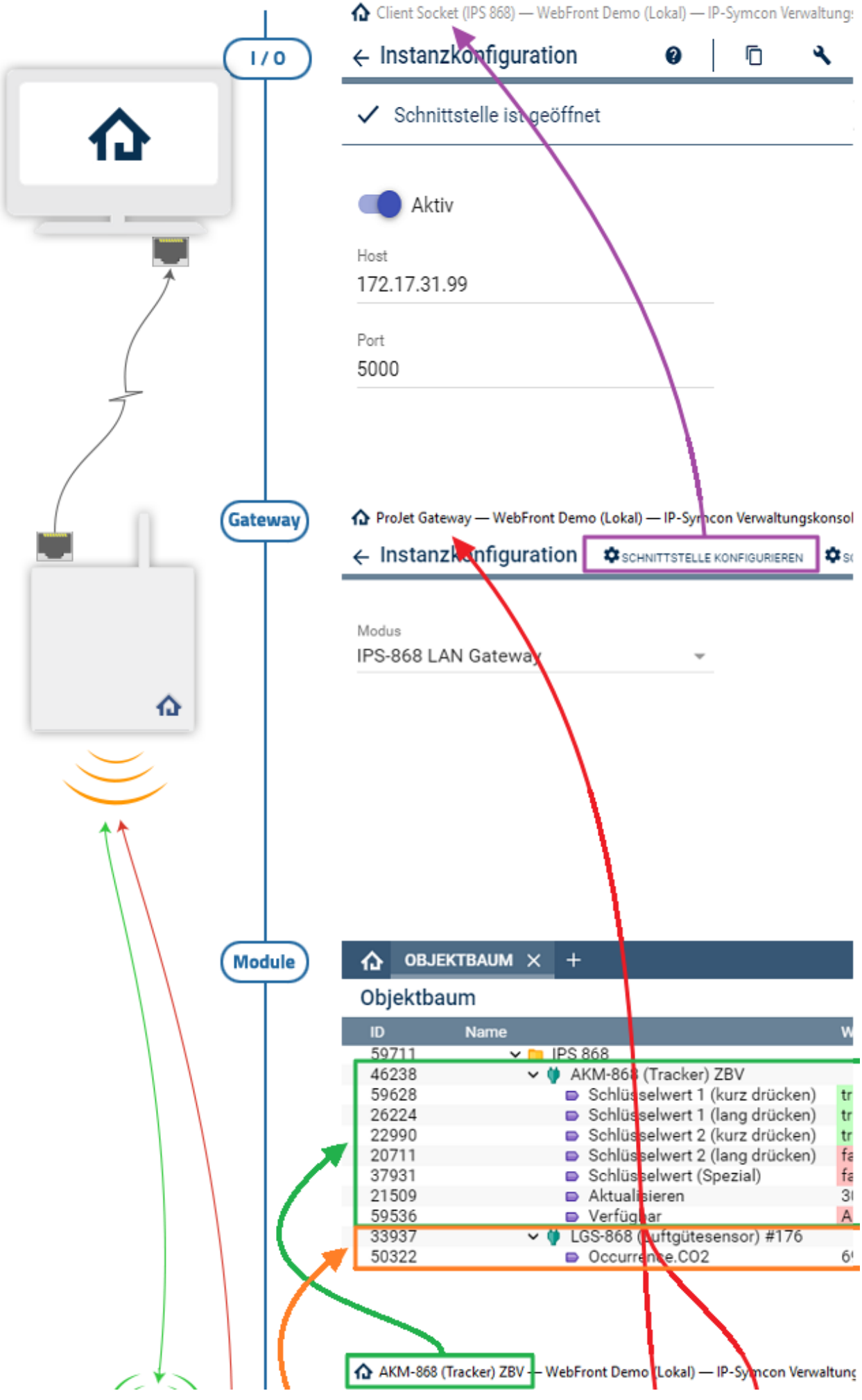
Instanzen repräsentieren im Normalfall Geräte, die an IP-Symcon angeschlossen sind. Zusätzlich können Instanzen auch virtuelle Geräte wie z. B. "Text To Speech" oder "Mediaplayer" repräsentieren, welche nicht physikalisch vorhanden sind. Die Zustände von Geräten werden in der Verwaltungskonsole und im WebFront über Variablen dargestellt.

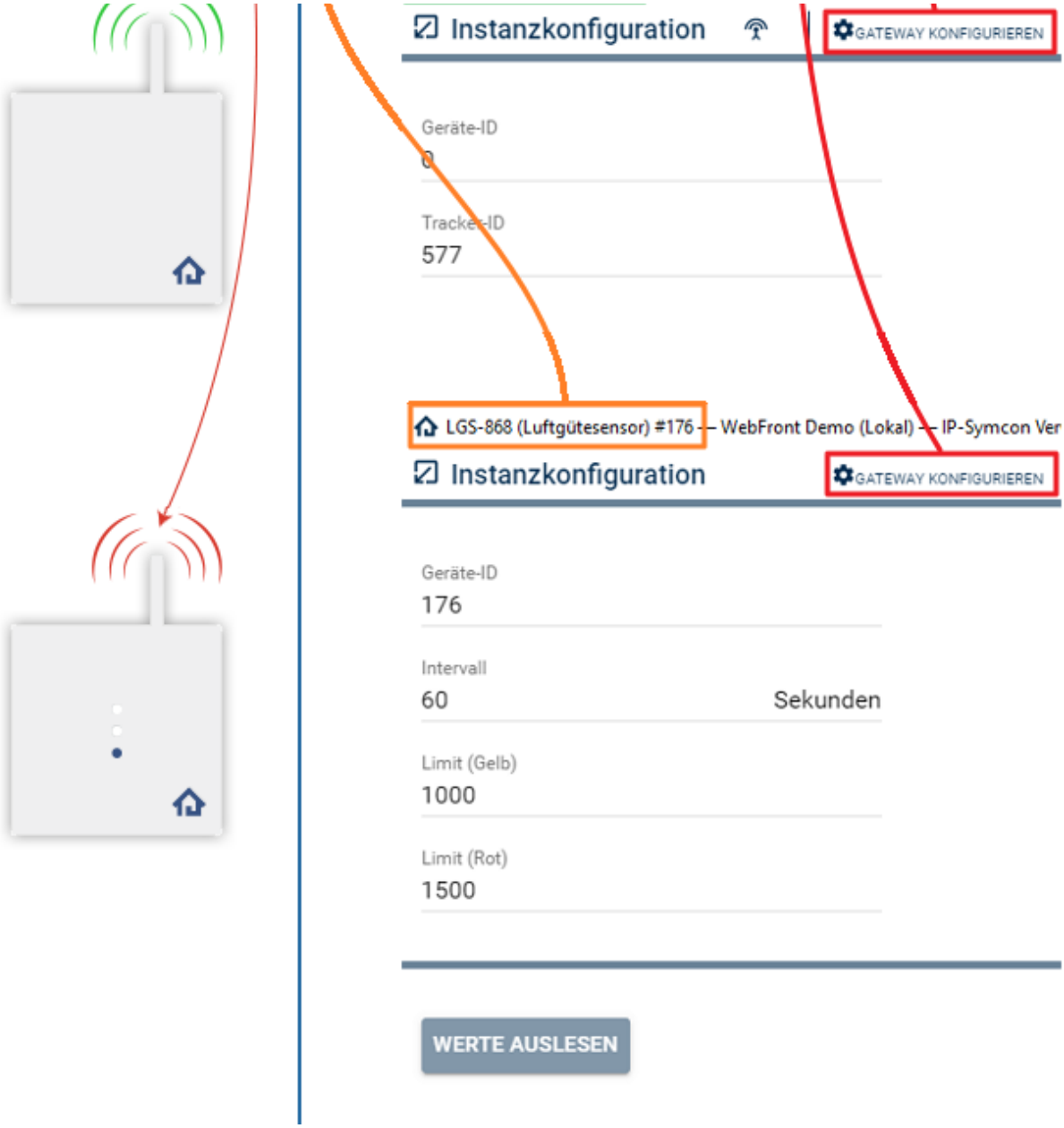
## EINBINDUNG

Über den "Instanz hinzufügen" Dialog oder über die Konfiguratoren in der Verwaltungskonsole können neue Geräte hinzugefügt werden. Zusätzlich zu dem vom Benutzer hinzugefügten Gerät erstellt IP-Symcon, falls benötigt, weitere Instanzen, welche die Kommunikation zwischen Gerät und IP-Symcon aufbauen. Diese Instanzen (auch "übergeordnete Instanzen" genannt) sind dabei austauschbar bzw. umkonfigurierbar, ohne dass das Gerät neu eingebunden werden muss. Soll zum Beispiel das Protokoll mit dem kommuniziert wird oder die Anschlussart des Geräts von Funk auf Seriell geändert werden, kann dies in der passenden Gateway- oder I/O-Instanz umgestellt werden. Diese Änderung gilt anschließend für alle an diese Gateway Instanz angeschlossenen Geräte, beinflusst aber in keiner Weise die Zustandsvariablen, eingerichtete Skripte oder die ursprünglich erstellte Instanz des Geräts selbst.

## ÜBERGEORDNETE INSTANZEN

Übergordnete Instanzen kennen und nutzen sowohl die benötigten Protokolle als auch Eigenschaften für einen Verbindungsaufbau zwischen Geräten und IP-Symcon. Diese Instanzen können somit nur Gateway- oder I/O-Instanzen sein.
Falls übergeordneten Instanzen zur Verbindung benötigt werden, richtet IP-Symcon diese vollkommen automatisch ein. Gegebenfalls wird noch eine Konfiguration dieser benötigt (z.B. Das Einstellen der IP-Adresse).
Die übergeordnete Instanz für ein Gerät/Instanz ist am schnellsten über das Zahnrad innerhalb des jeweiligen Konfigurationsreiters erreichbar.
Wie unten im Beispiel zu sehen, kommunizieren die beiden Geräte "AKM-868" und "LGS-868" via Funk mit der übergeordneten Instanz "ProJet Gateway". Das Gateway kommuniziert seinerseits mit der übergeordneten I/O Instanz "Client Socket IPS 868" via LAN.
Es wäre nun problemlos möglich die I/O Instanz von einer LAN-basierenden zu einer seriellen Kommunikation umzustellen (dies macht natürlich nur Sinn wenn das neue Gateway über ein serielles Kabel mit IP-Symcon kommuniziert). Dazu wird im "ProJet Gateway" der Modus auf "Connection via: Serial" umgestellt. Automatisch richtet IP-Symcon eine Serial Port Instanz ein, welche mit dem Gateway verknüpft ist. Nun wäre über das Zahnrad auf der Gateway-Konfigurationsseite die übergeordnete Instanz "Serial Port IPS 868" aufruf-/konfigurierbar. Dies beeinflusst in keiner Weise die beiden Geräte AKM und LGS.


I / O
Client Socket (IPS 868) — WebFront Demo (Lokal) — IP-Symcon Verwaltung
Instanzkonfiguration
Schnittstelle ist geöffnet
Aktiv
Host
172.17.31.99
Port
5000
Gateway
ProJet Gateway — WebFront Demo (Lokal) — IP-Symcon Verwaltungskonsol
Instanzkonfiguration
SCHNITTSTELLE KONFIGURIEREN
Modus
IPS-868 LAN Gateway
Module
OBJEKTBAUM
Objektbaum
ID
Name
59711 IPS 868
46238 AKM-868 (Tracker) ZBV
59628 Schlüsselwert 1 (kurz drücken)
26224 Schlüsselwert 1 (lang drücken)
22990 Schlüsselwert 2 (kurz drücken)
20711 Schlüsselwert 2 (lang drücken)
37931 Schlüsselwert (Spezial)
21509 Aktualisieren
59536 Verfügbar
33937 LGS-868 (Luftgütesensor) #176
50322 Occurrence.CO2
AKM-868 (Tracker) ZBV — WebFront Demo (Lokal) — IP-Symcon Verwaltung




## EINBINDUNGSARTEN

Instanzen werden immer nach einem ähnlichen Schema in IP-Symcon eingebunden.
Es gibt 3 Typen von Einbindungen.

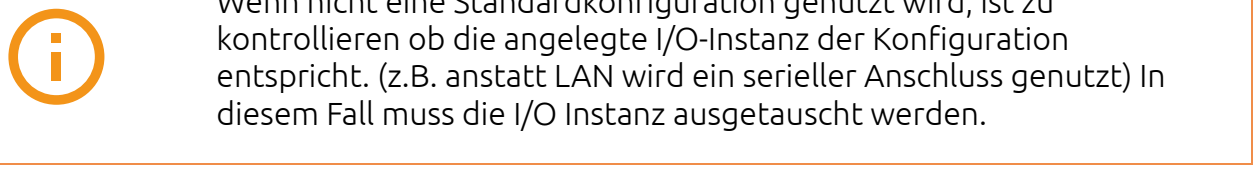
| TYP | BESCHREIBUNG | BEISPIELSYSTEME |
|---|---|---|
| 1:1:n | 1 I/O - 1 Gateway - n Instanzen | EnOcean |
| 1:m:n | 1 I/O - m Gateways - n Instanzen | LCN |
| 1:n | 1 I/O - n Instanzen | Registervariable |

## VERBINDUNGSKOMPONENTEN

I/O:
I/O beschreibt die Kommunikationsart zwischen dem Gateway und dem Server(IP-Symcon).

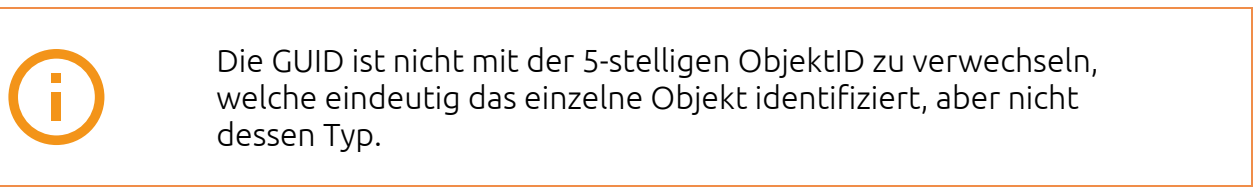
| ART | BESCHREIBUNG |
|---|---|
| Client Socket | TCP-Client basierende Kommunikation |
| HID | HID basierende Kommunikation (USB) |
| Multicast Socket | Multicast basierende Kommunikation |
| Serial Port | Seriell basierende Kommunikation |
| Server Socket | TCP Server basierende Kommunikation |
| TMEX | TMEX basierende Kommunikation |
| UDP Socket | UDP-Client basierende Kommunikation |
| Virtual I/O | Emuliert einen Serial Port, einen Client Socket oder einen UDP Socket |
| WWW Reader | HTTP(Get) Anfragen |

**Gateway/Splitter:**
An ein Gateway können ein oder mehrere Geräte/Instanzen angebunden sein. Das Gateway kümmert sich um die Kommunikation zwischen Gerät und I/O.




## GUID

Jedes Instanz hat eine eindeutige GUID, welche den Typ bestimmt (z.B. Es ist ein AKM-868 oder WDT-868).
Die GUID ist eine UUID und hat das Format 8-4-4-4-12. Die Zahlen geben die Anzahl der Ziffern an. Die Ziffern bestehen aus Zeichen zwischen 0-9 und A-F. Es müssen immer Bindestriche und geschweifte Klammern vorhanden sein und es dürfen nur Großbuchstaben verwendet werden. (Beispiel: {12345678-90AB-CDEF-1234-567890ABCDEF} )



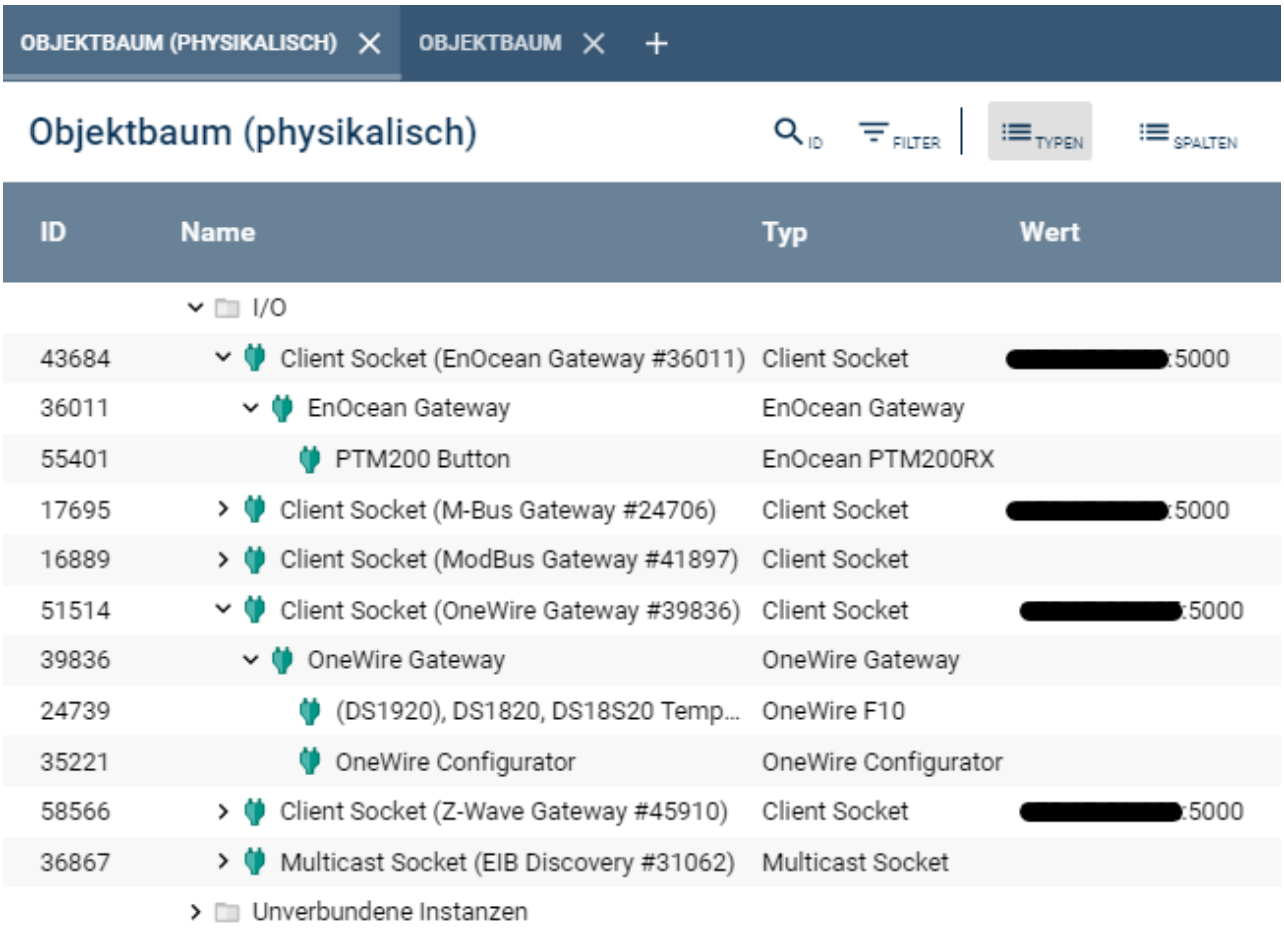

## INSTANZ ERSTELLEN

Das Erstellen von Instanzen bzw. das Einbinden von Geräten erfolgt wiefolgt:
Geräte einbinden

## BEISPIEL

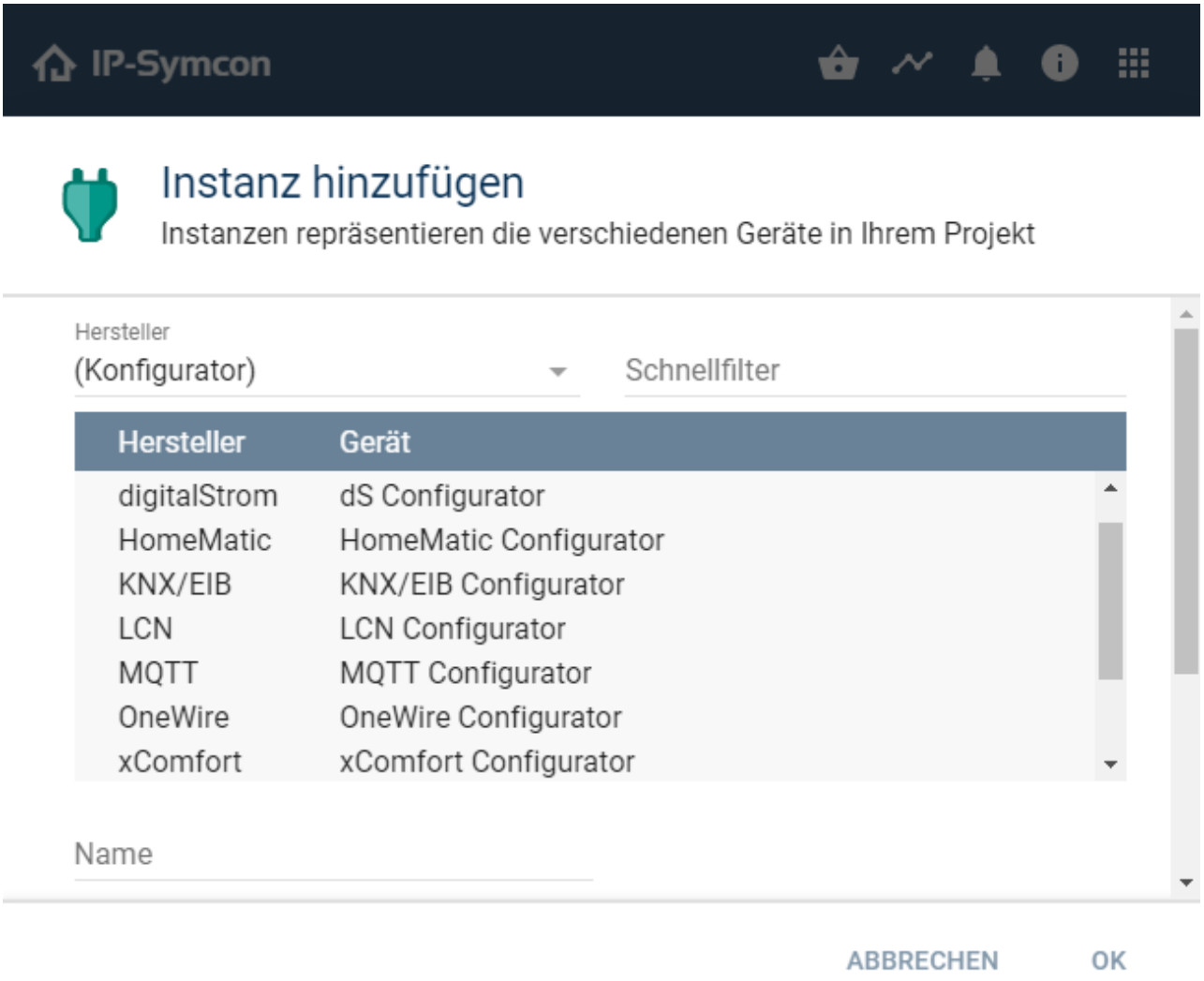
Hier ein Beispiel anhand der physikalischen Baumansicht.
Geräte-Instanz "PTM200 Button"
-> (übergeordnete) Gateway-Instanz "EnOcean Gateway"

-> (übergeordnete) I/O-Instanz "Client Socket (EnOcean Gateway #36011)"



# Konfiguratoren

Konfiguratoren stellen in IP-Symcon eine ganz spezielle Art von Instanzen dar, die, wenn für ein System verfügbar, die Einrichtung in IP-Symcon erheblich erleichtern. Es können für mehrere Stränge/Linien/Gateways jeweils Konfiguratoren erstellt werden. Die Erstellung eines Konfigurators ist im Objektbaum über das hinzufügen einer Instanz möglich.



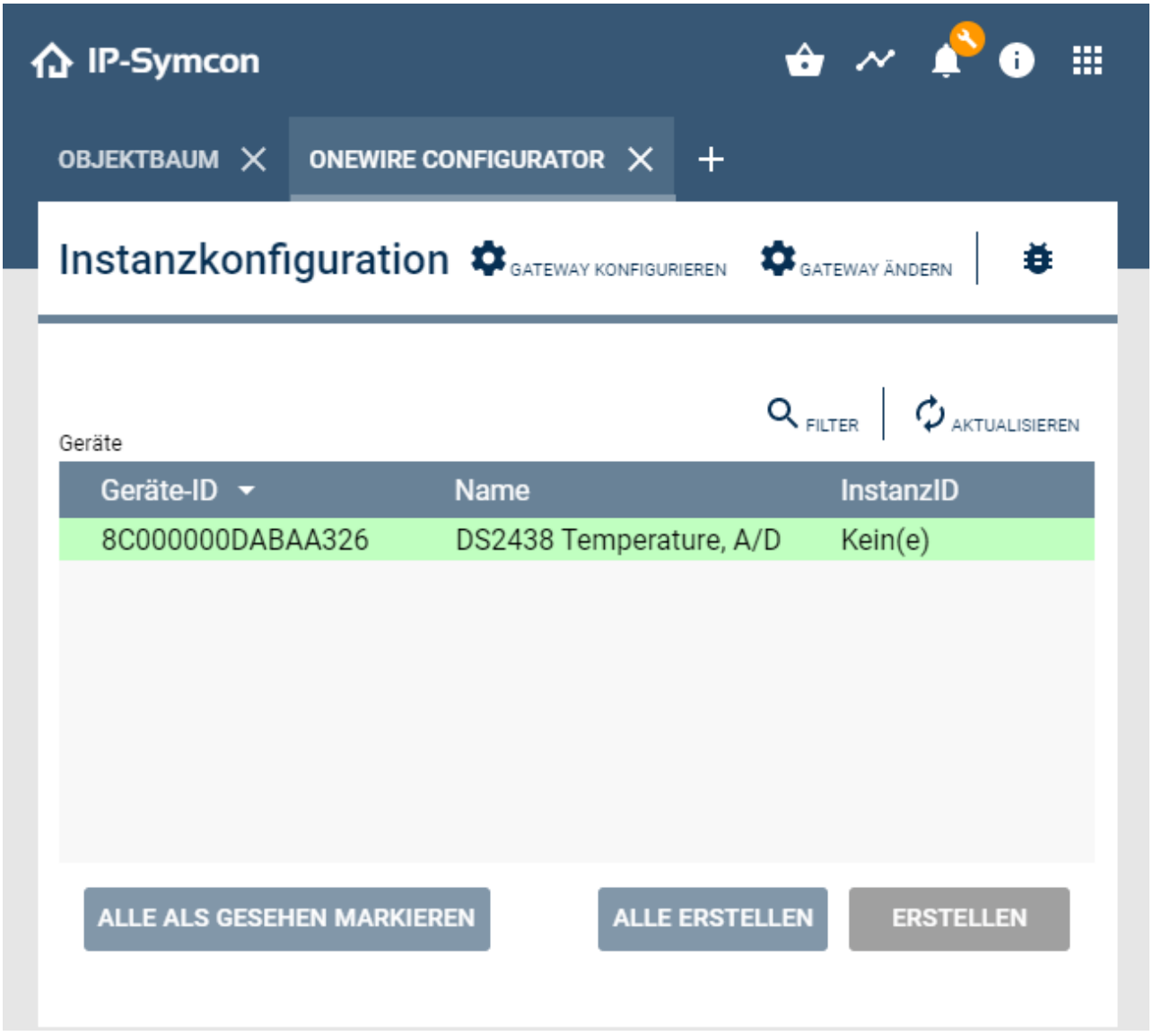

Zur Zeit sind Konfiguratoren für die folgenden Systeme verfügbar:

- 1-Wire
- digitalStrom
- Eaton xComfort
- KNX
- EnOcean
- HomeMatic
- IPS-868
- LCN
- M-Bus
- MQTT
- Siemens Logo
- Siemens OZW
- Z-Wave

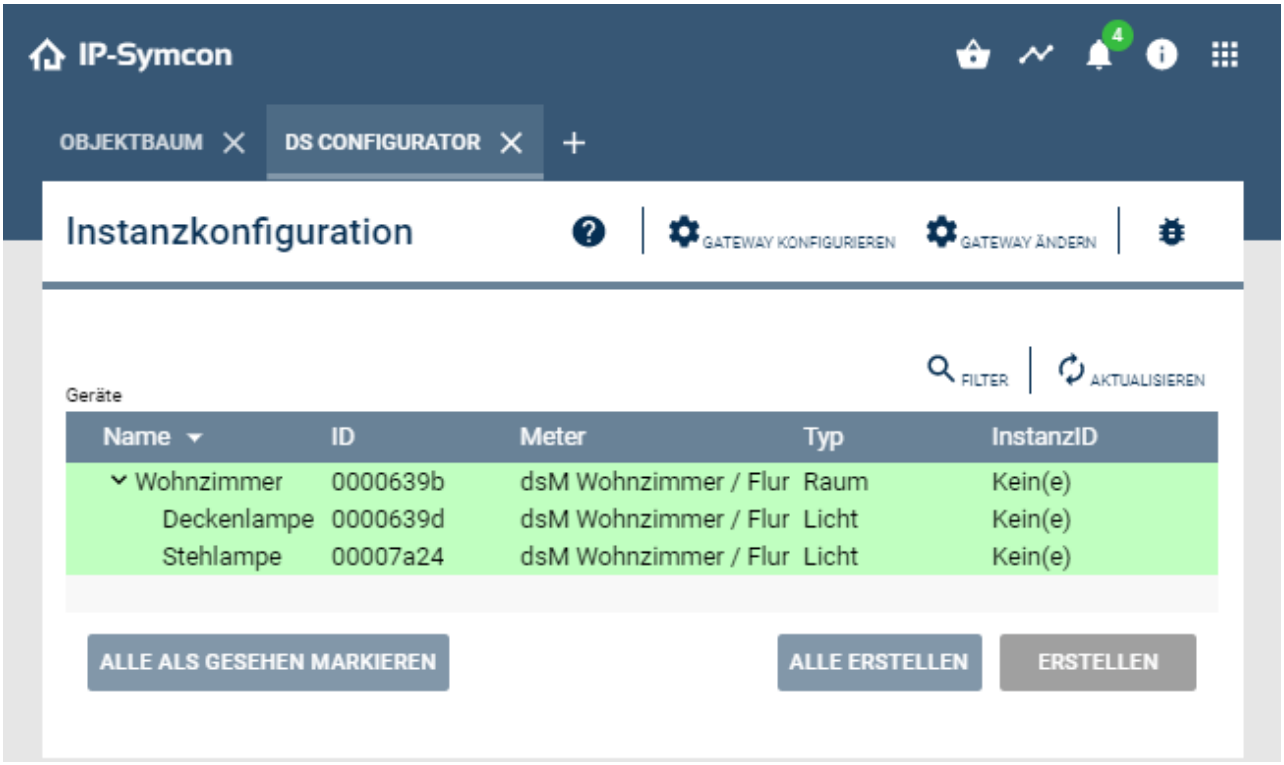
| FARBE | BEDEUTUNG | HINWEIS |
|---|---|---|
| Grün | Neues Gerät, welches noch nicht in IP-Symcon eingerichtet wurde | Gerät kann als Instanz hinzugefügt werden |
| Rot | Gerät nicht mehr vorhanden, aber noch in IP-Symcon eingerichtet | Instanz kann in IP-Symcon gelöscht werden |
| Grau | Das vorhandene Gerät ist nicht mit dem in IP-Symcon eingerichteten Gerät kompatibel | Instanz muss in IP-Symcon entfernt und neu hinzugefügt werden |
| Weiß | Gerät vorhanden und in IP-Symcon eingerichtet | Keine Aktion erforderlich |

## SCREENSHOTS

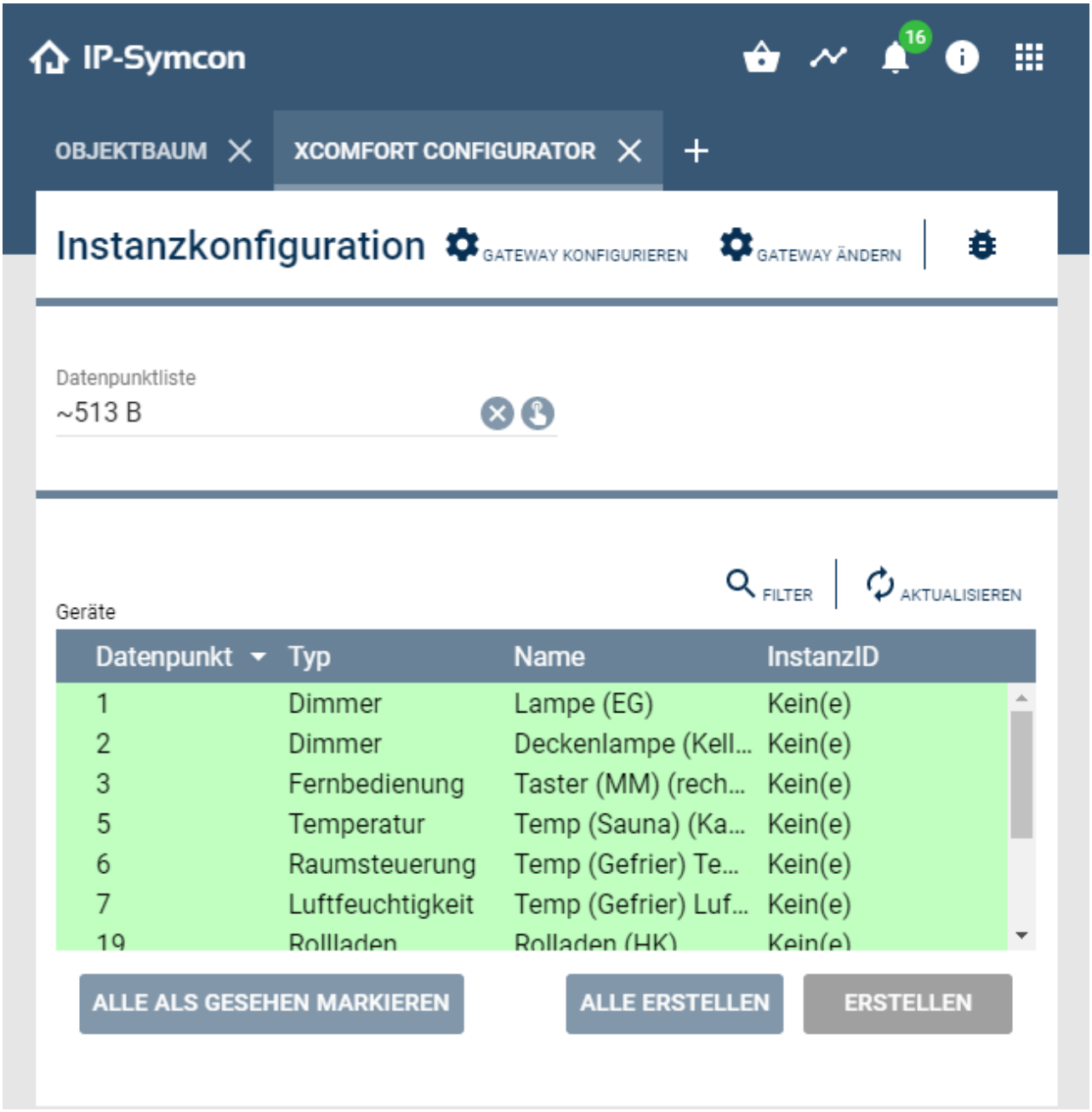
### 1-WIRE KONFIGURATOR



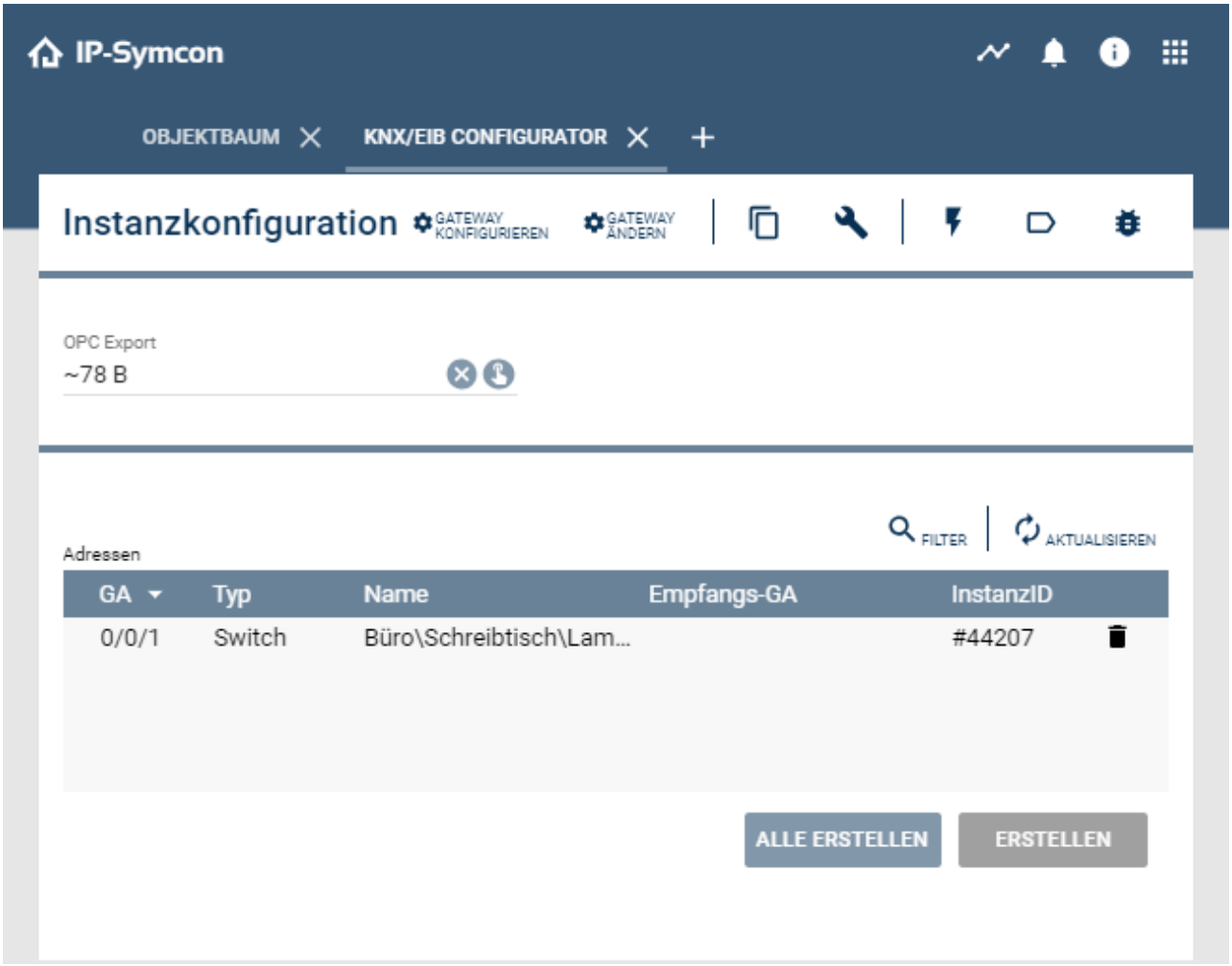

### DIGITALSTROM KONFIGURATOR



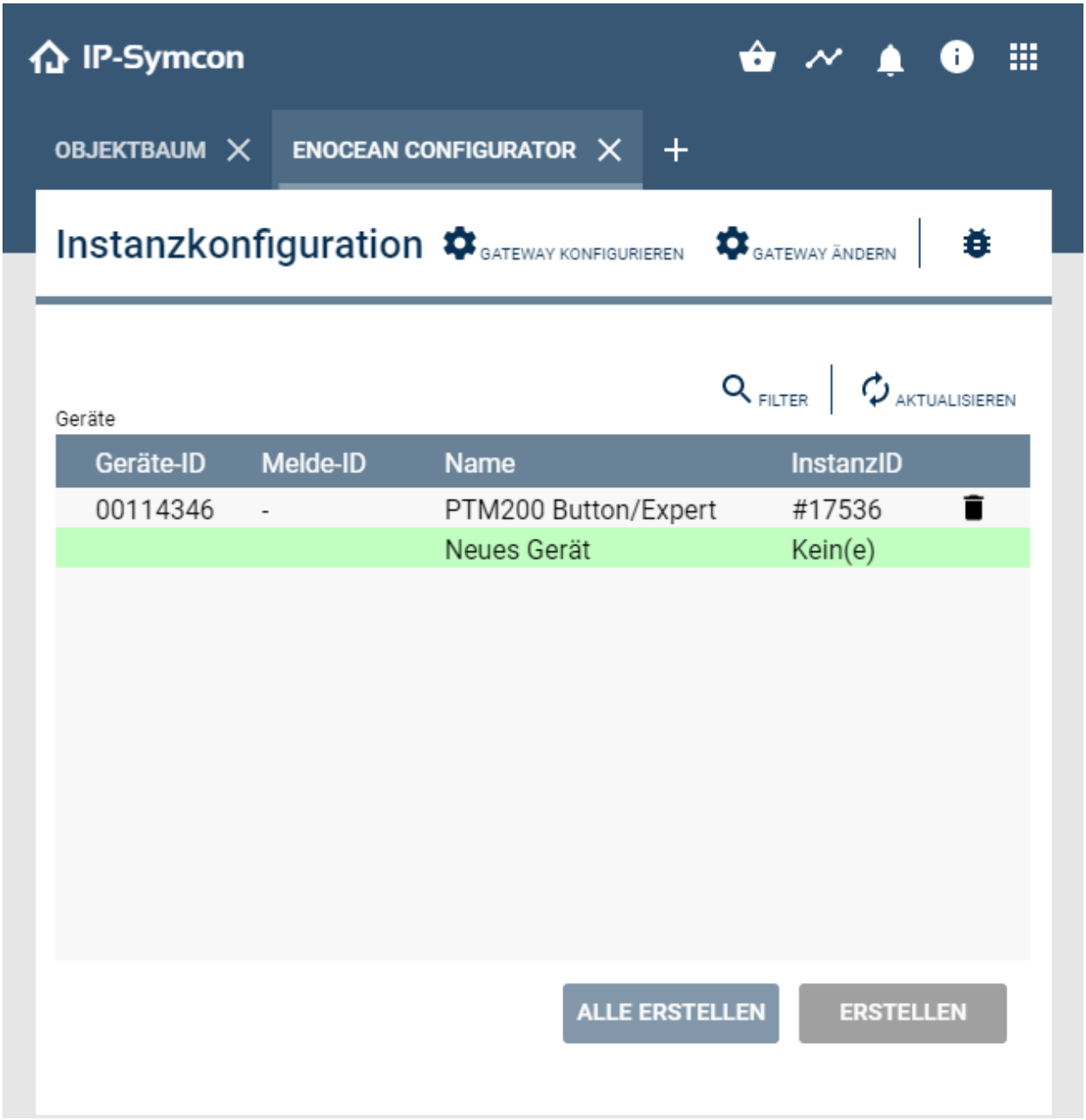

## XCOMFORT KONFIGURATOR



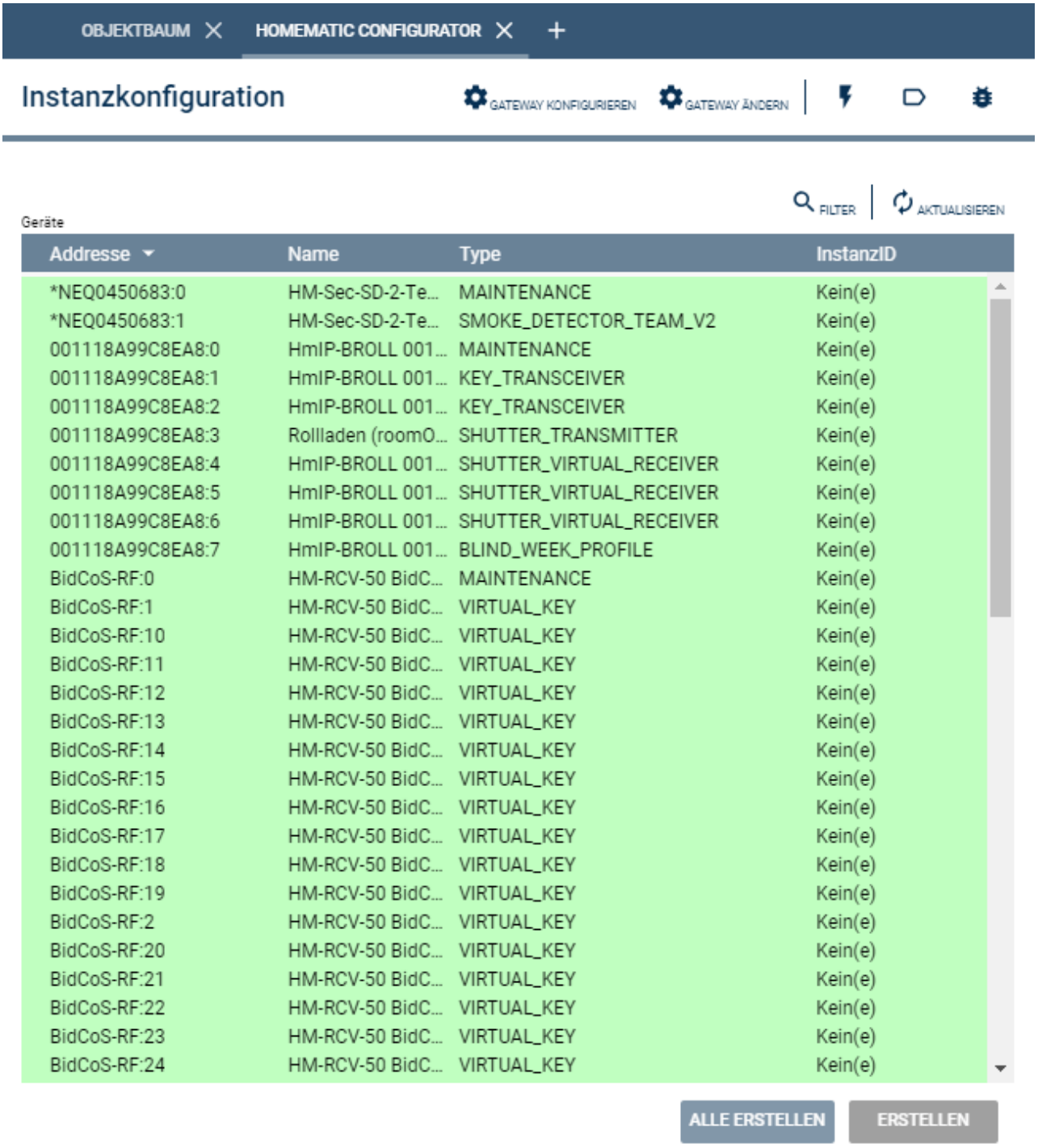

**KNX KONFIGURATOR**



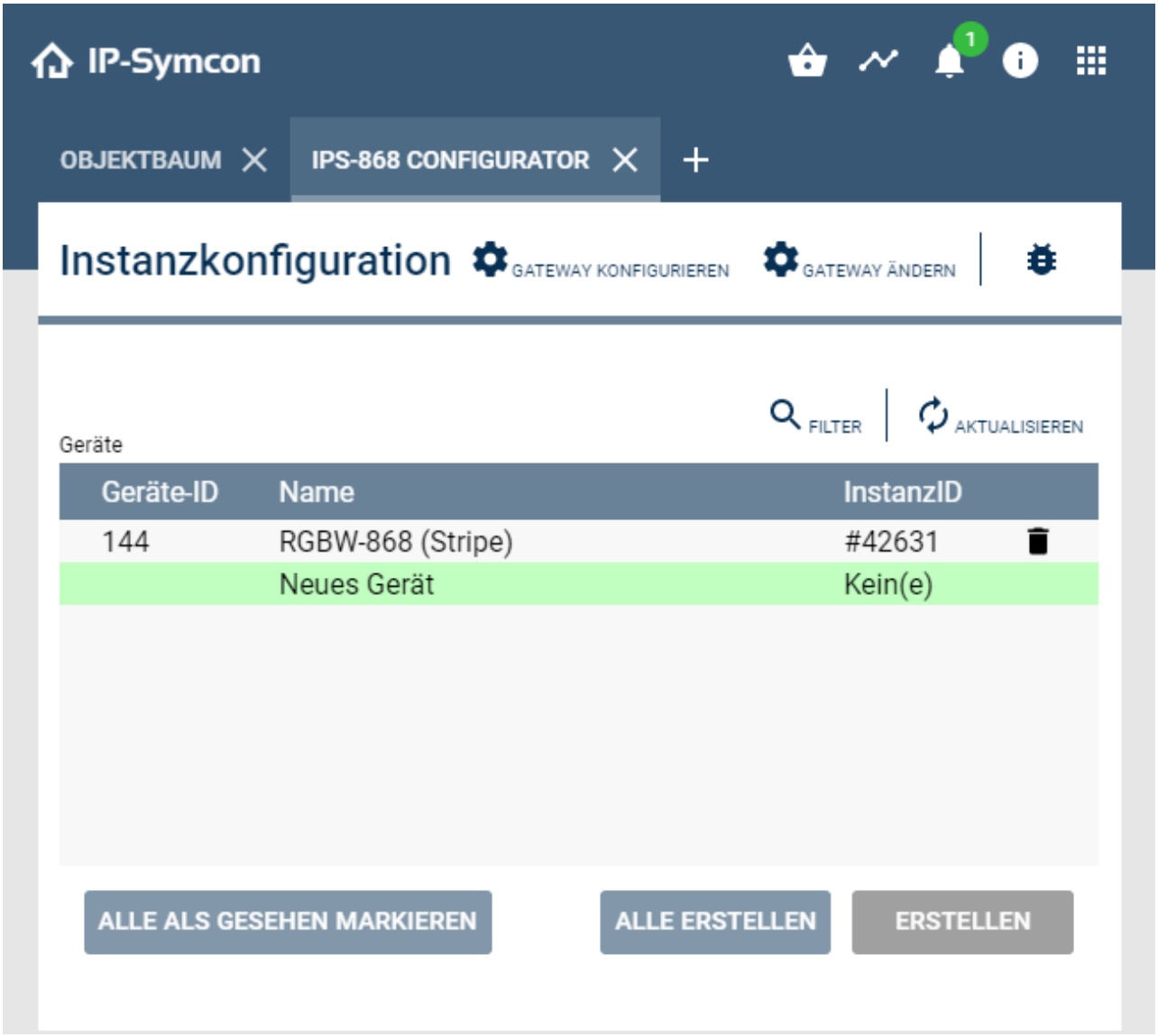

## ENOCEAN KONFIGURATOR



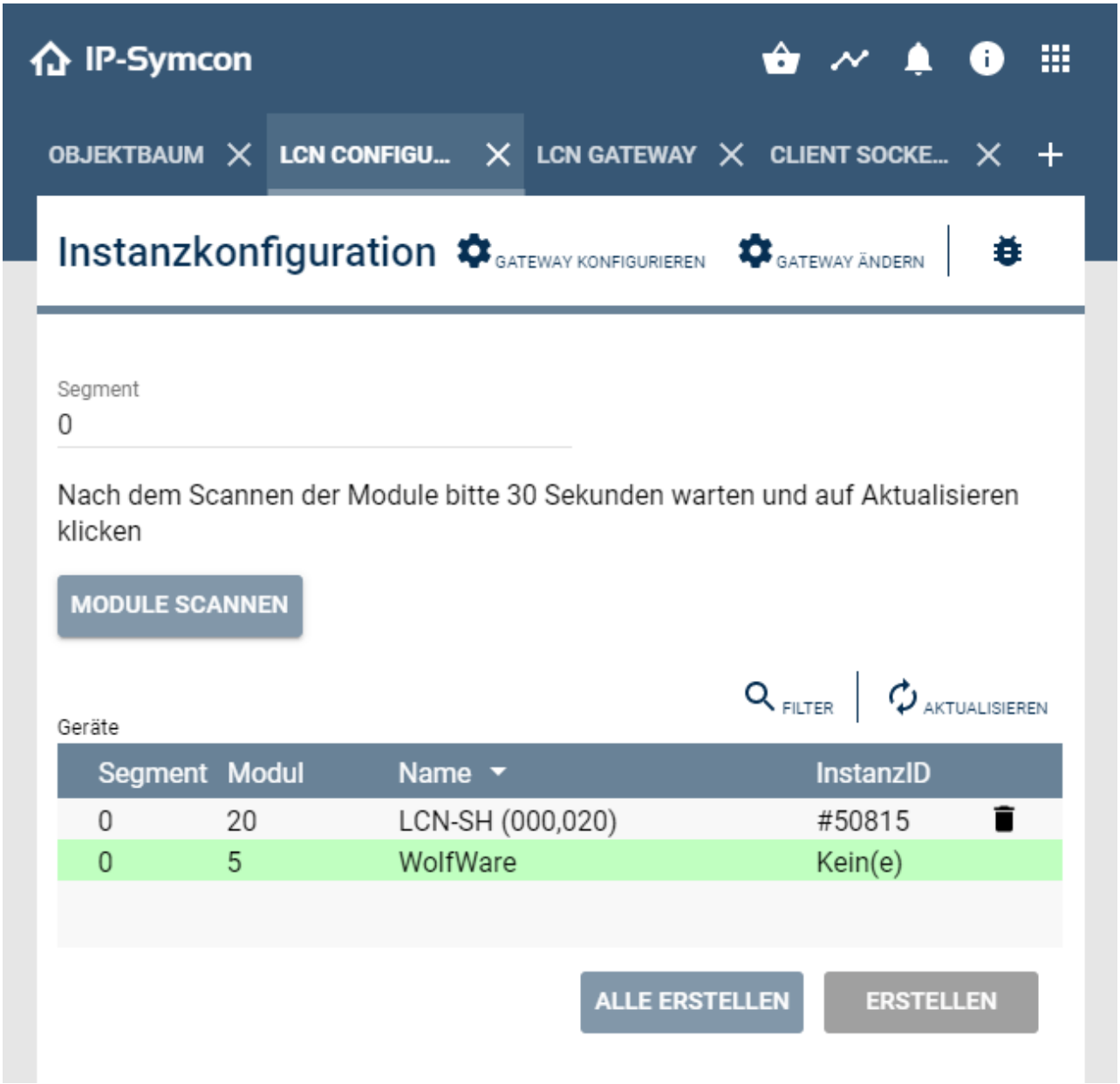

## HOMEMATIC KONFIGURATOR

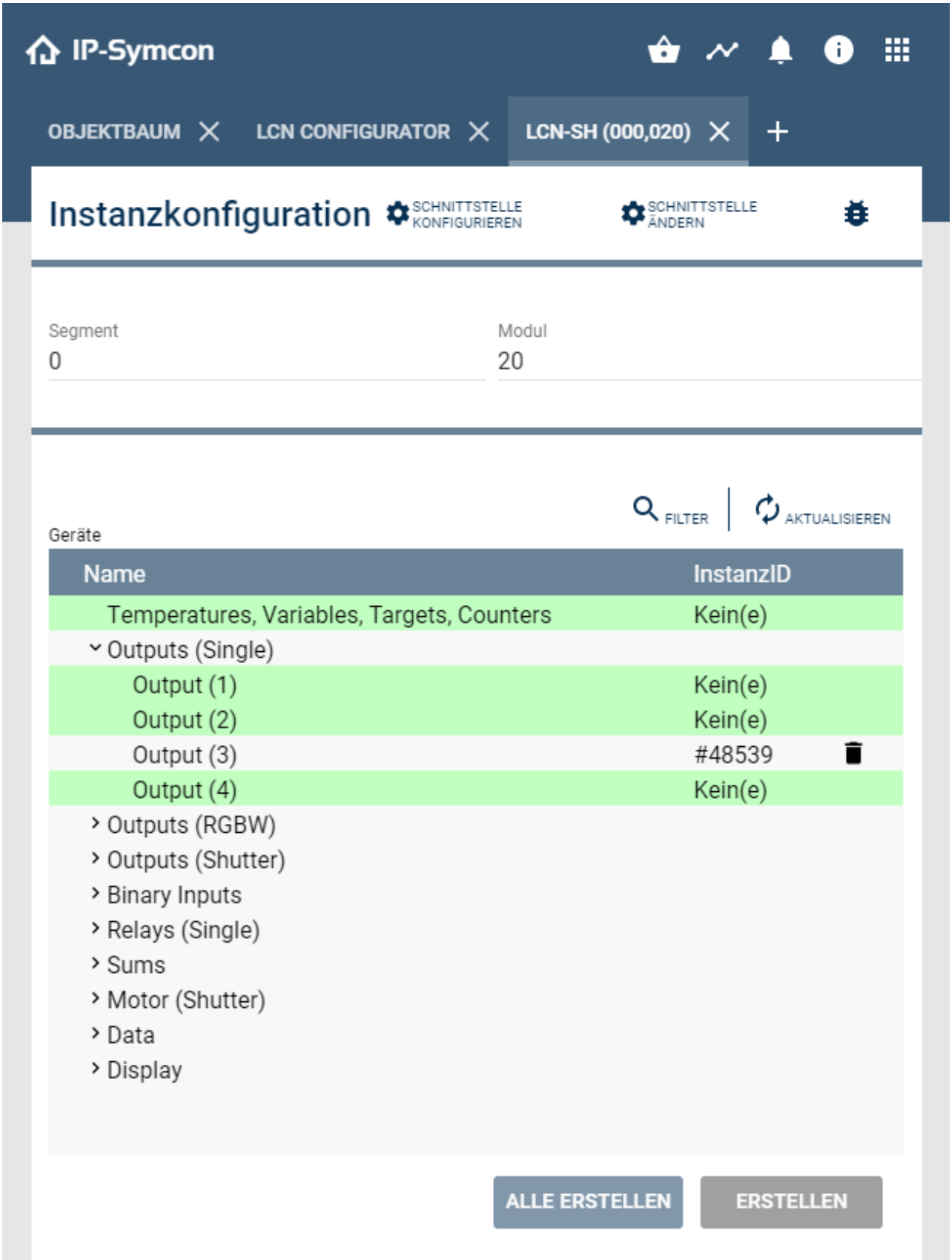



## IPS-868 KONFIGURATOR

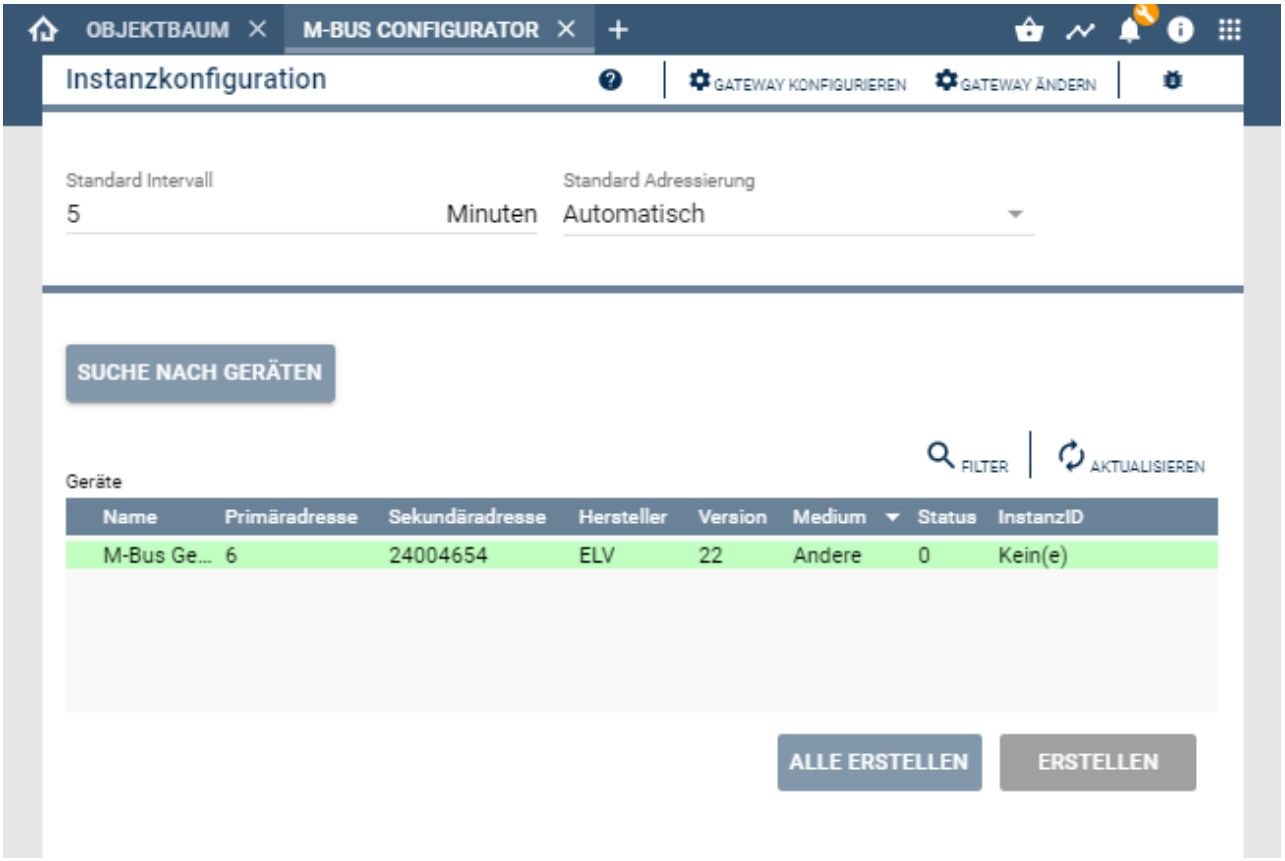



**LCN KONFIGURATOR**

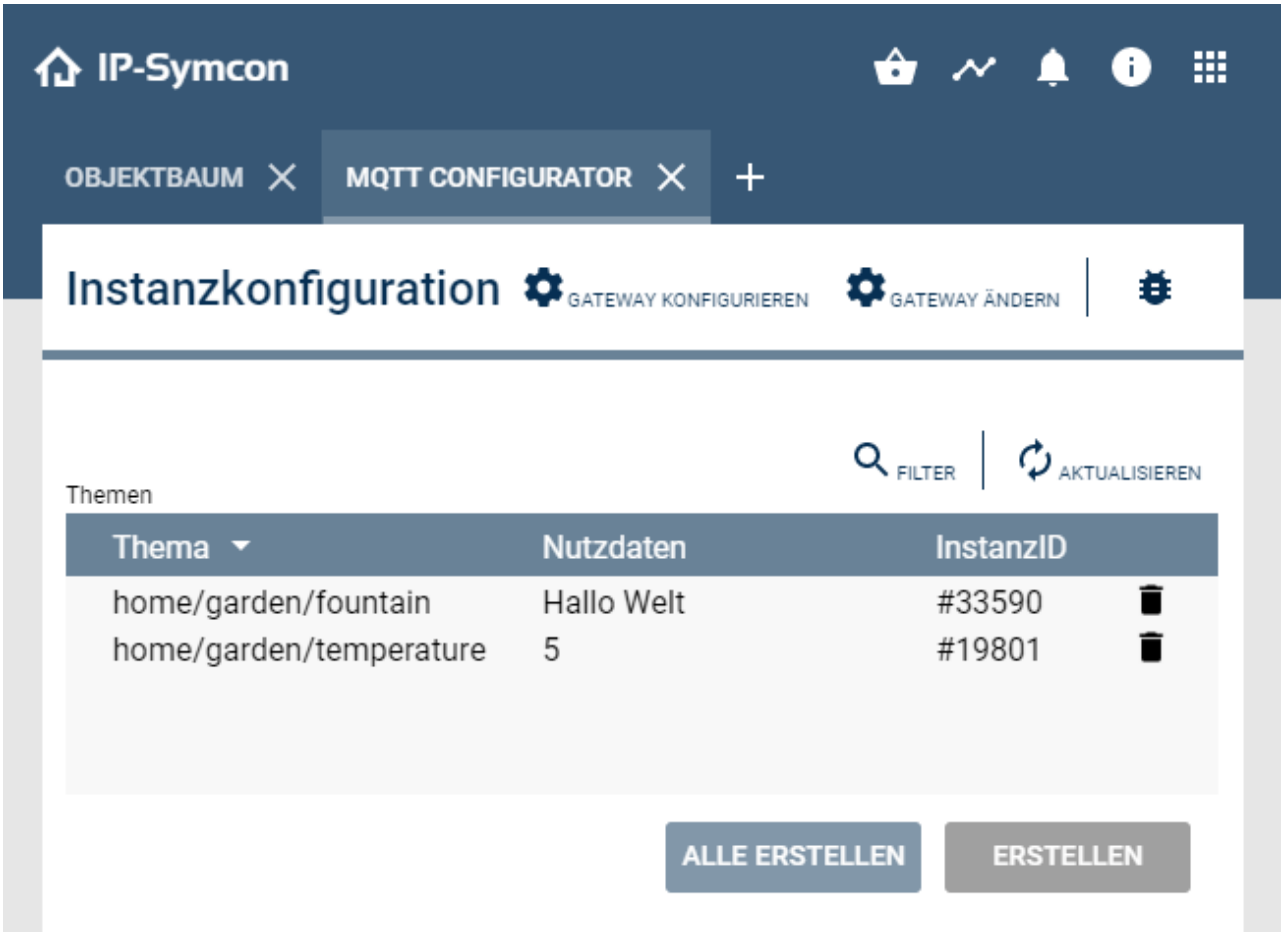

IP-Symcon
OBJEKTBAUM
LCN CONFIGU...
LCN GATEWAY
CLIENT SOCKE...
Instanzkonfiguration
GATEWAY KONFIGURIEREN
GATEWAY ÄNDERN
Segment
0
Nach dem Scannen der Module bitte 30 Sekunden warten und auf Aktualisieren klicken
MODULE SCANNEN
FILTER
AKTUALISIEREN
Geräte
Segment
Modul
Name
InstanzID
0
20
LCN-SH (000,020)
#50815
0
5
WolfWare
Kein(e)
ALLE ERSTELLEN
ERSTELLEN



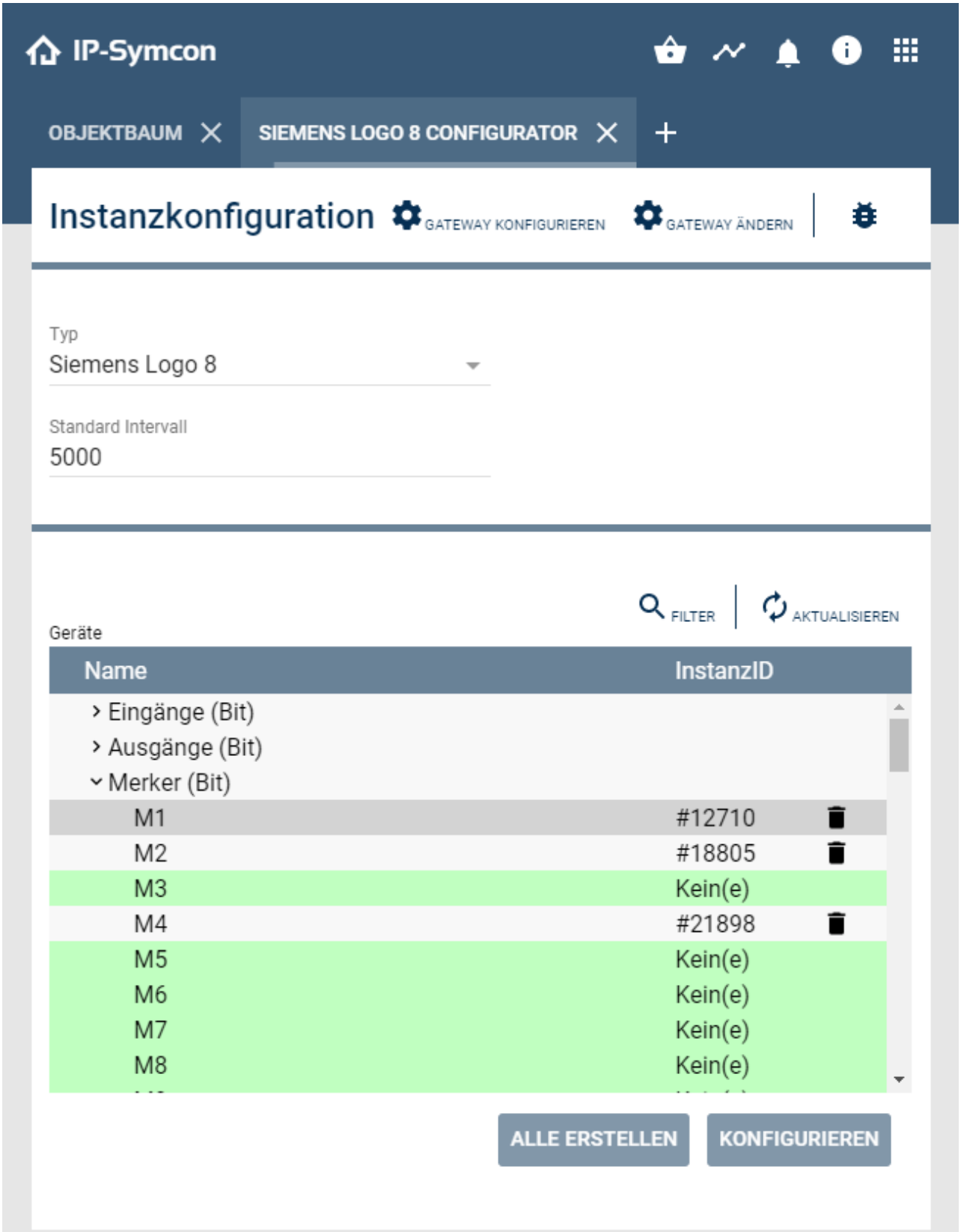

## M-BUS KONFIGURATOR



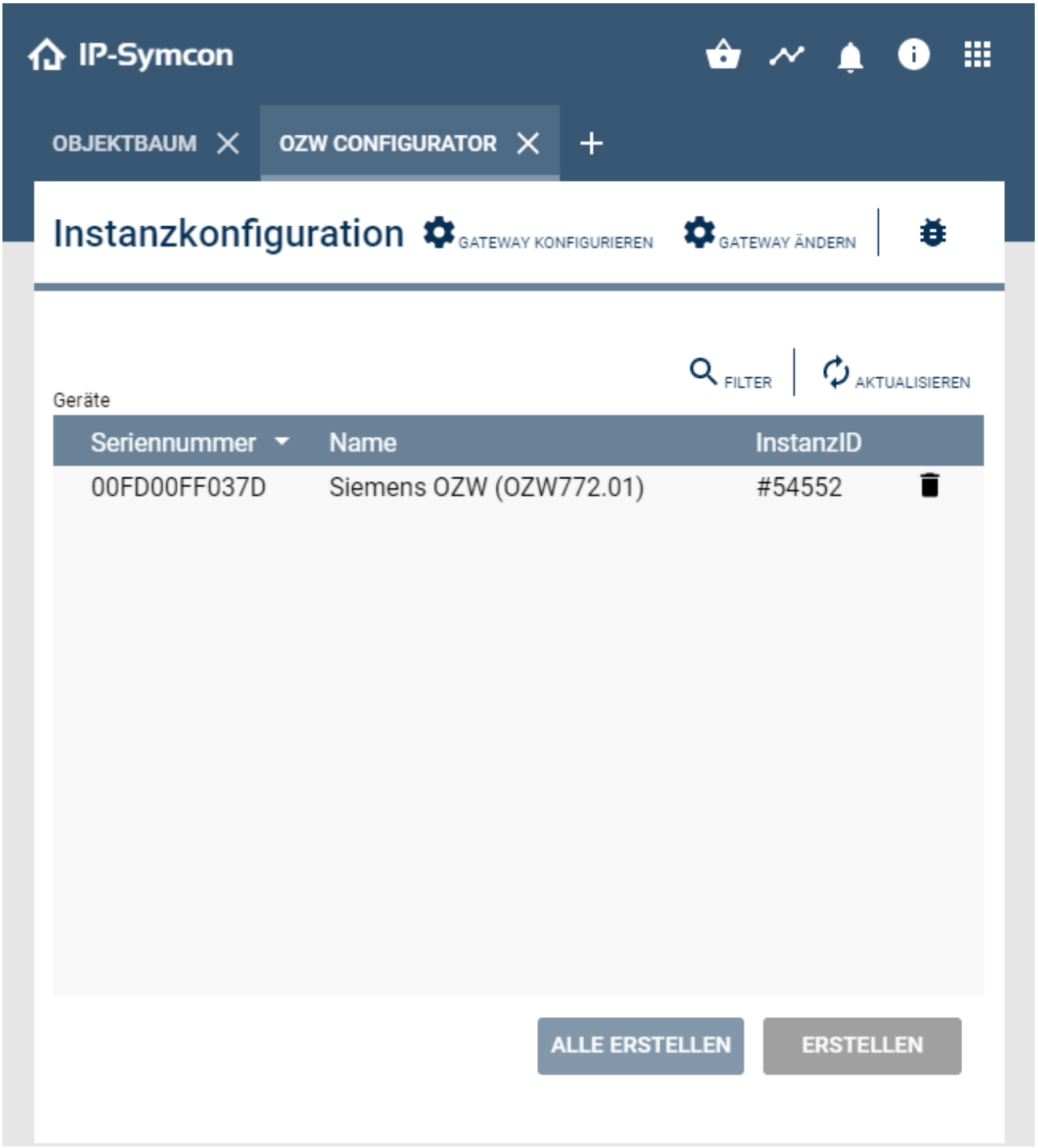

## MQTT KONFIGURATOR



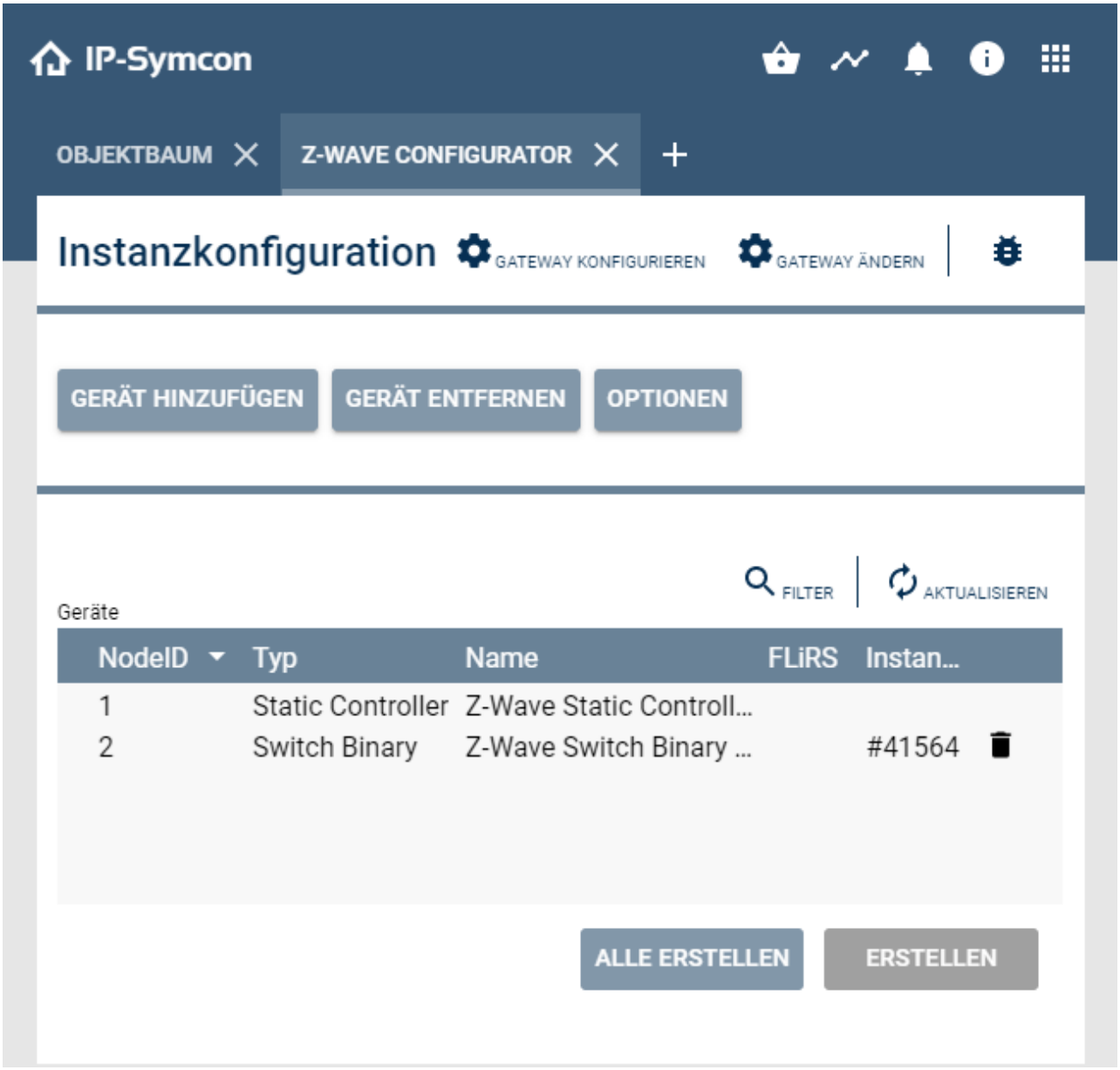
## SIEMENS LOGO




## SIEMENS OZW KONFIGURATOR




### Z-WAVE KONFIGURATOR


IP-Symcon
OBJEKTBAUM
Z-WAVE CONFIGURATOR
Instanzkonfiguration
GATEWAY KONFIGURIEREN
GATEWAY ÄNDERN
GERÄT HINZUFÜGEN
GERÄT ENTFERNEN
OPTIONEN
FILTER
AKTUALISIEREN
Geräte
NodeID
Typ
Name
FLiRS
Instan...
1
Static Controller
Z-Wave Static Controll...
2
Switch Binary
Z-Wave Switch Binary ...
#41564
ALLE ERSTELLEN
ERSTELLEN

# Variablen

Eine Variable ist ein Behälter, der einen Wert enthält. Dieser Wert kann ein Boolean, Float, Integer oder String sein. Im Normalfall beinhalten Variablen Statuswerte von Geräten. Zusätzlich können selbsterstellte Variablen genutzt werden um eigene Werte abzuspeichern, welche z.B. ebenfalls in der Visualisierung dargestellt oder in Skripten verarbeitet werden können.



Systemvariablen haben im engeren Sinn, nichts mit normalen Variablen zu tun, da diese nur innerhalb von Skripten verfügbar sind.

Zum Austausch und zur Speicherung aller Daten benutzt IP-Symcon diese Variablen. Variablen beinhalten Statuswerte von Geräten, die es erst möglich machen, dass z.B. die Temperatur eines Thermometers in der Visualisierung sichtbar wird oder dass ein An- bzw. Ausschalten der Beleuchtung überhaupt funktioniert.

## BESCHREIBUNG

Bei IP-Symcon gibt es zwei Arten von Variablen:

### STATUS VARIABLEN

Jede Instanz eines Gerätes kann eine oder mehrere Status-Variablen besitzen, diese werden automatisch erstellt. In diesen Variablen wird der Status des Gerätes gespeichert (z.B. An / Aus oder Temperatur in °C oder zusätzlich noch die Feuchtigkeit und der Zustand der Batterie).
Wann sich der Status bzw. dessen Wert ändert oder aktualisiert wird, hängt von dem jeweiligen System ab. Eine Wetterstation z.B. sendet periodisch die Umweltdaten an IP-Symcon, während bei anderen Systemen aktiv deren Zustand (z.B. ein Fenster- Türkontakt) abgefragt werden muss. Das (Abfrage-)Intervall wird in der Konfiguration der zugehörigen Instanz eingestellt.
Eine wichtige Besonderheit der Status-Variablen ist, dass deren Wert nicht vom Anwender verändert werden darf. Diese Variablen spiegeln nur den Zustand und die Datensätze des Gerätes wieder.



Status Variablen dürfen nicht gelöscht werden, da dies zu unvorhersehbarem Verhalten der jeweiligen Instanz führen kann.

**Beispiel 1:**
Um die Soll-Raumtemperatur bei einer Heizungssteuerung zu ändern, wird ein Befehl ausgeführt, welcher den Soll-Wert an das Gerät übermittelt. Es wird nicht die Status-Variable "Temperatur" selbst verändert. Die dazu notwendigen Befehle können in der jeweiligen Modulreferenz nachgelesen werden. Alternativ kann auch der Befehl "RequestAction " direkt auf die Variable angewandt werden. IP-Symcon sucht dann den dedizierten Befehl selbst und nutzt diesen.
**Beispiel 2:**

Ein abstrakteres Beispiel in Form eines KFZ: um ein Fahrzeug zu beschleunigen oder abzubremsen, müssen die entsprechenden Fußpedale betätigt werden. Es reicht nicht aus die Tachonadel auf die gewünschte Geschwindigkeit einzustellen.



Fazit: Status Variablen liefern ausschließlich Informationen über das Gerät und sind darum explizit als "Nur Lesen" markiert.

### BENUTZERDEFINIERTE VARIABLEN

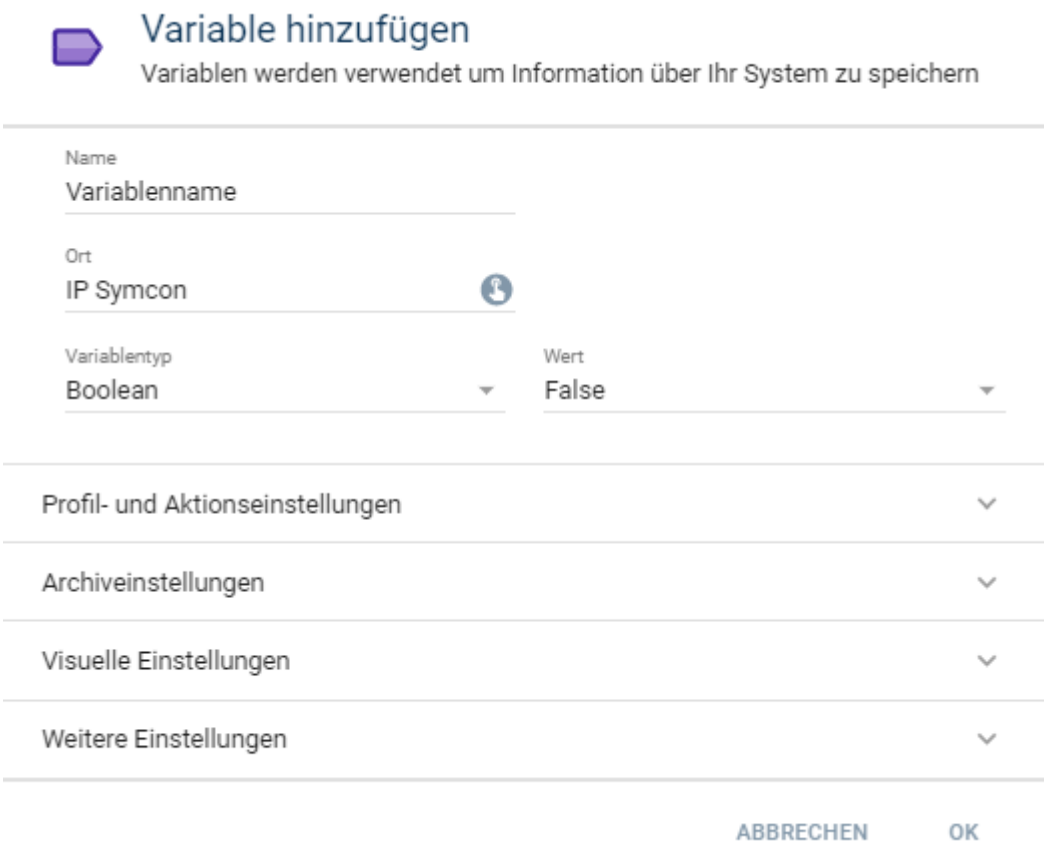
Benutzerdefinierte Variablen können selbst erstellt werden und einen von vier Datenypen (siehe Tabelle Variablentypen ) beinhalten. Benutzerdefinierte Variablen können vollständig Manipuliert werden.
**Beispiel 1:**
Soll ein bisher unbekanntes Gerät (z.B. AV-Verstärker) in IP-Symcon integriert oder ein Wert umgerechnet werden, so ist dies mithilfe der "Benutzerdefinierten Variablen" lösbar.
**Beispiel 2:**
Ein weiteres Beispiel wäre die Umrechnung der Außentemperatur zusätzlich in Grad Fahrenheit (°F).

## VARIABLENTYPEN

| VARIABLENTYP | BESCHREIBUNG | BEISPIEL |
|---|---|---|
| Boolean | True / False | z.B. An oder Aus |
| Float | Gleitkommazahl | z.B. 231,956 |
| Integer | Ganze Zahlen | z-B. -10 … -4 … 0 … 32 … 472 |
| String | Text | “Hallo IP-Symcon. Hallo Welt” |

## NEUE VARIABLEN ANLEGEN

Eine Variable kann im Objektbaum über das "+" oder "Rechtsklick" -> "Objekt hinzufügen" -> "Variable" erstellt werden. Die zweite Variante hat den Vorteil, dass die Variable direkt unterhalb vom ausgewählten Objekt erstellt wird und nicht erst hinterher einsortiert werden muss. Im folgenden Fenster muss ein Variablentyp, Variablendarstellung und ebenso ob die Variablenaufzeichnung für diese Variable aktiv sein soll ausgewählt werden. Die Aggregation gibt dabei den Typ der Variablenaufzeichnung an.
Bevor die Erstellung mit "OK" bestätigt wird, sollte darauf geachtet werden, dass die neu erstellte Variable einen aussagekräftigen Namen hat. Optional kann noch eine Bemerkung hinzugefügt und ein Icon ausgewählt werden. Der Ort der Variable kann mit dem Dialog "Ort" festlegt oder nachträglich im Objektbaum verschoben werden.




## VARIABLENVISUALISIERUNG

Variablen werden in der Visualisierung standardmäßig angezeigt. Dies kann durch die Visuelle Einstellung ein- oder ausgeschaltet werden.
Um Variablenwerte aufzuzeichnen und einen Graphen in der Visualisierung anzuzeigen, muss das Aufzeichnen von Variablenveränderung (Archive Control ) aktiviert werden.
Variablen die aufgezeichnet und in der Visualisierung angezeigt werden, bieten in der Visualisierung ein Diagramm mit dem Verlauf der Variablenwerte an.

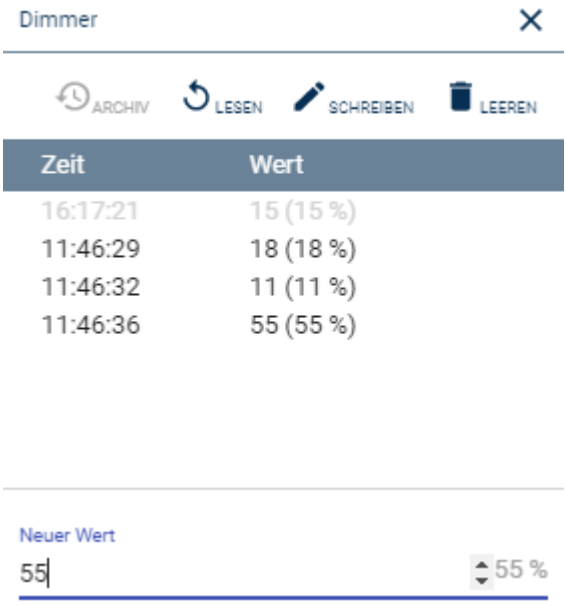
## VARIABLENDARSTELLUNG

Die Darstellung der Variablen in der Visualisierung kann angepasst werden, sodass eine Variable beispielsweise als Auswahl oder Schieberegler dargestellt wird. Diese Optionen kann auch nachträglich verändert werden. Eine genauere Beschreibung der Möglichkeiten befindet sich hier: Variablendarstellungen

## VARIABLENWERTE BEARBEITEN

Mit einem Doppelklick auf den Namen der Variablen kann die Darstellung einer Variable verändert oder die Variablenaufzeichnung in die Datenbank aktiviert werden.
Um die Werte einer Variable mitzuschneiden oder zu verändern, kann per Doppelklick auf den Wert der Variable das Fenster für den "Überwachungsmodus" aufgerufen werden.
Alternativ kann per Kontextmenü auch der Eintrag “Variable verändern" auswählt werden.
Diese Funktion ist primär für Testzwecke konzipiert, um beispielsweise zu überprüfen, ob bei einer simulierten Temperatur die Heizung eingeschaltet wird. Außerdem ist es möglich die Darstellung in der Visualisierung zu kontrollieren.

Das nachfolgende Bild zeigt das Fenster für den "Überwachungsmodus", in dem fortlaufend die Variablenwerte angezeigt werden und währenddessen modifiziert werden können. Ein Veränderung kann entweder über den Button “Schreiben” oder über die Return-Taste bestätigt werden.




Falls die Variable eine Status-Variable ist, kann der Wert modifiziert werden, obwohl dieser als schreibgeschützt markiert ist. Bei einer Wertänderung werden auch alle damit verbundenen Ereignisse ausgeführt. Das ist nützlich zum Testen eigener Ereignisse, ändert aber nicht die tatsächlichen Werte auf dem physikalischen Gerät.

## VARIABLENAKTIONEN

Einer Variable kann via "Eigene Aktion" eine Variablenaktion zugewiesen werden. Diese wird aufgerufen, wenn die Variable via Visualisierung geschaltet wird. Darüber hinaus wird die Variablenaktion ausgeführt, wenn die Variable geschaltet wird, beispielsweise über die Funktion RequestAction .

# Variablenaktionen

Eine Variablenaktion wird angefordert, wenn eine Variable via Visualierung angeklickt wird.
Das hinterlegte Aktionsskript wird ausgeführt und an dieses werden anhand von Systemvariablen Daten wie der neue "gewünschte" Wert und ID übergeben.

## VARIABLENAKTION AUSWÄHLEN

Eine Aktion kann im "Variable bearbeiten"-Dialog über "Eigene Aktion" ausgewählt werden. Der Dialog "Variable bearbeiten" ist beim Erstellen einer Variable oder im Nachhinein via "Objektbaum" -> "Doppelklick" oder über "Objektbaum" -> "Rechtsklick" -> "Objekt bearbeiten" erreichbar.



Einige Variablen von hinzugefügten Modulen beinhalten eine "Standardaktion". Diese kann durch eine "Eigene Aktion" überschrieben werden.




Mit dem "+" kann eine Variablenaktion erstellt werden, welche sich nur darum kümmert, das im WebFront die Variable ihren Wert bei einem anklicken auch ändert. Dieses automatisch erstellte Aktionsskript wird mit "(Automatisch erstellt)" angezeigt.




## VARIABLENAKTION ERSTELLEN

Eine Variablenaktion ist ein ausgewähltes Aktionsskript.
Nähere Informationen sind unter PHP-Skripte und Aktionsskripte nachlesbar.

# Variablendarstellung

Benötigt: IP-Symcon >= 8.0
Jede Variable besitzt eine Variablendarstellung. Variablendarstellungen werden genutzt um Variablenwerte wie True, False oder 0, 1, 2, 3 in eine für den Menschen leserliche Form zu bringen und mit den notwendigen Kontextdaten (z.B. °C) zu versehen. Die Variablendarstellung gibt also an, wie eine Variable in der Visualisierung dargestellt wird. Darüber hinaus bestimmt sie das Bedienelement, mit dem der Variablenwert verändert werden kann.

## DARSTELLUNG AUSWÄHLEN

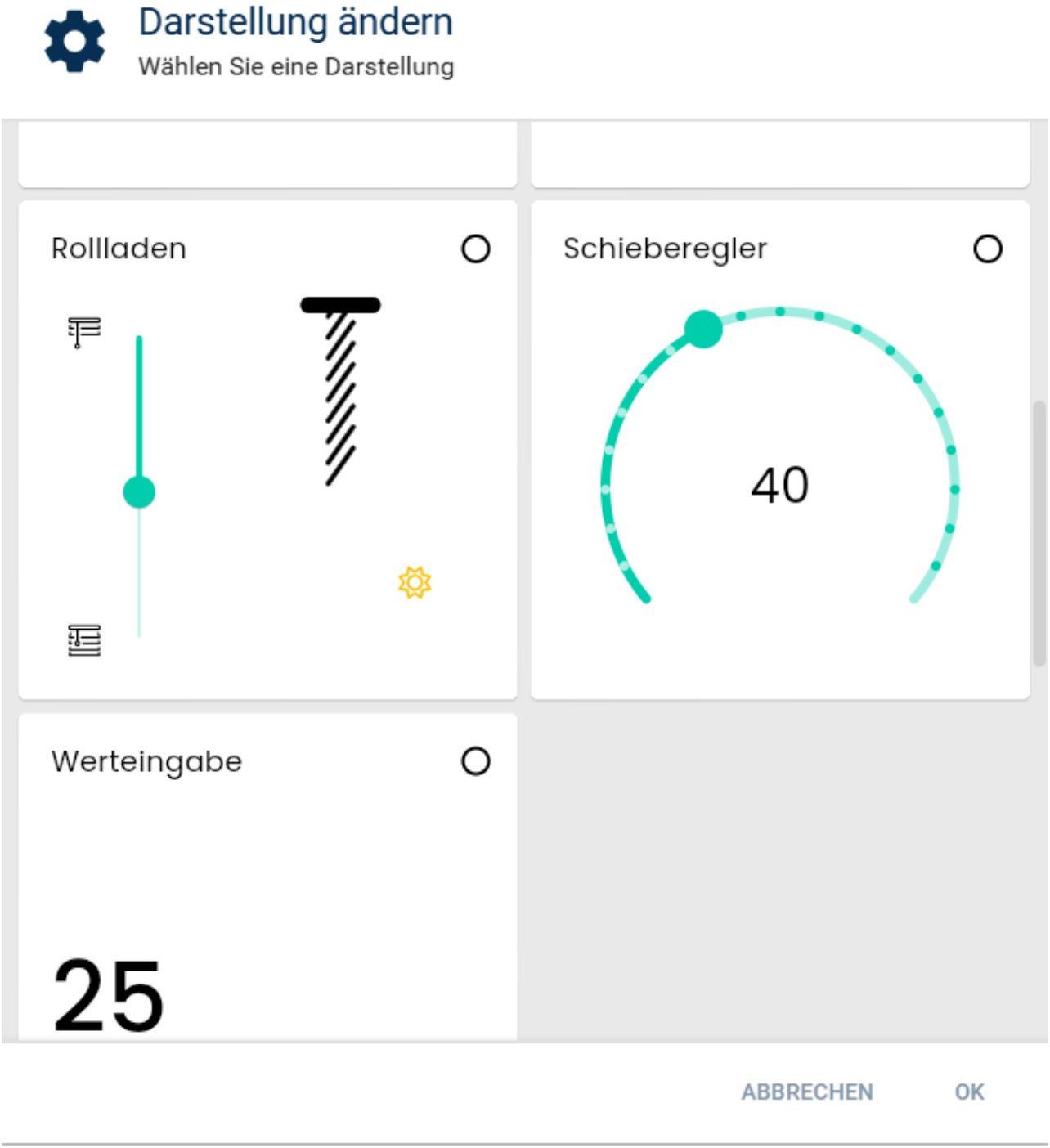
Eine Darstellung kann im "Variable bearbeiten"- Dialog in dem Abschnitt "Variablendarstellung" ausgewählt werden. Dieser Dialog ist beim Erstellen einer Variable oder im Nachhinein via "Objektbaum" -> "Doppelklick" oder über "Objektbaum" -> "Rechtsklick" -> "Objekt bearbeiten" erreichbar.



Einige Variablen von hinzugefügten Modulen beinhalten eine "Standarddarstellung". Diese kann durch einen Klick auf die Schaltfläche "Standard überschreiben" überschrieben werden. Der Originalzustand kann daraufhin über die Schaltfläche "Auf Standard zurücksetzten" wiederhergestellt werden.




Über das Auswahl-Icon beim Eintrag "Darstellung" wird der Darstellungsdialog geöffnet, in welchem eine Darstellung ausgewählt werden kann. In dem Darstellungsdialog werden immer alle Darstellungen angezeigt. Wenn die Konfiguration der aktuellen Variable eine Darstellung nicht zulässt, wird diese ausgegraut. Beim Bewegen der Maus über das ausgegraute Element wird ein Text angezeigt, der beschreibt welche Anforderungen erfüllt werden müssen um die Darstellung zu werden.




Jede Darstellung bietet viele verschieden Parameter um sie den Anforderungen entsprechend anzupassen. Die verfügbaren Parameter und ihre Funktion sind unter den jeweiligen Einträgen unter Objektdarstellung näher beschrieben.

## VORSCHAU

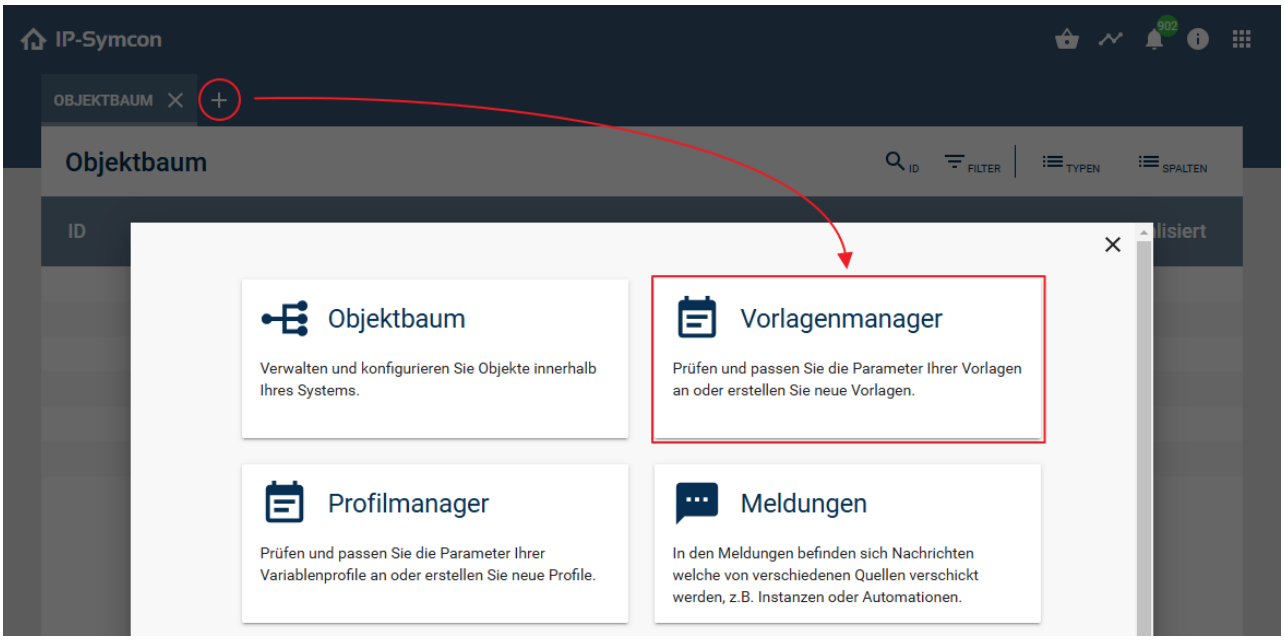
Jede Anpassung kann direkt in der Vorschau neben der Darstellungsauswahl beobachtet werden. Durch Klicken auf das Lupen-Symbol der Vorschau kann diese im Vollbild geöffnet werden. Hier kann, wie in der normalen Visualisierung, über das Stift-Symbol die Größe der Kachel angepasst werden, um zu sehen wie sich die Darstellung in verschiedenen Größen verhält.

## VORLAGEN

Wenn die Parameter einer Darstellung verändert werden, gelten diese nur für die aktuell ausgewählte Variable. Wenn mehrere Variablen die gleichen Parameter nutzen sollen, können Vorlagen genutzt werden. Vorlagen können unter dem Feld für Darstellungen ausgewählt werden. Wurde eine Vorlage gewählt, so verwendet die Darstellung die Parameter der Vorlage, auch wenn diese später noch angepasst werden sollte. Dennoch können Parameter im Variablendialog weiterhin verändert werden. Entsprechen die Parameter dadurch nicht mehr den Werten der Vorlage oder der übergeordneten Darstellung wird in dem Feld Vorlage "(Benutzerdefiniert)" angezeigt und die eingestellten Parameter gelten wieder genau für diese Variable. Eigene Vorlagen können im Vorlagenmanager erstellt und bearbeitet werden.

# Vorlagenmanager

Benötigt: IP-Symcon >= 8.0
Der Vorlagenmanager ist kann über das "+" in der Tabliste aufgerufen werden.

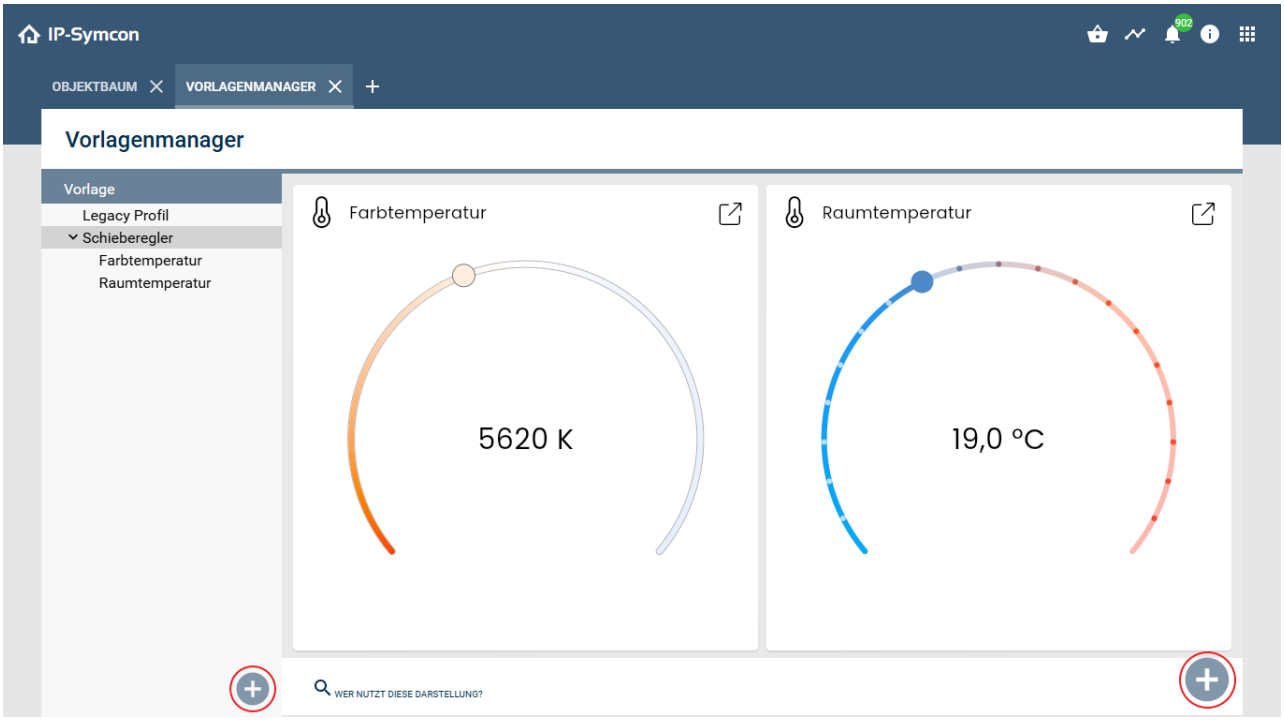



## AUFBAU

Der Vorlagenmanager besteht aus einem Baum auf der linken Seite, in welchem Vorlagen gruppiert nach Darstellungen ausgewählt werden können und einem zentralen Detailbereich, welcher sich an die aktuelle Auswahl des Baumes anpasst. Ist kein Eintrag ausgewählt, so werden im Detailbereich alle Darstellungen als Vorschau angezeigt. Alternativ zur Auswahl im Baum, kann hier eine spezifische Darstellung per Klick auf das "Öffnen"-Symbol oben rechts in jeder Kachel geöffnet werden. Ist eine Darstellung ausgewählt, so zeigt der Detailbereich für jede Vorlage dieser Darstellung eine Vorschau und ermöglicht auch hier ein direktes Öffnen. Über den Button "Wer nutzt diese Darstellung?" am unteren Ende des Detailbereichs werden alle Variablen angezeigt, welche diese Darstellung verwenden. Ist schließlich eine Vorlage ausgewählt, so erscheint ein dazugehöriges Konfigurationsformular, analog zum zur Variablenkonfiguration, in welchem die Parameter der Vorlage konfiguriert werden können. Zusätzlich kann der Name der Vorlage angepasst werden. Da jede Vorlage im Hintergrund über eine eindeutige ID und nicht den Namen identifiziert wird, kann der Name bedenkenlos angepasst werden. Er dient lediglich des besseren Verständnisses für den Benutzer. Im unteren Bereich kann über "Wer nutzt diese Vorlage?" eine Liste von Variablen, welche diese Vorlage verwenden, ausgegeben werden. Über "Duplizieren" kann eine neue Vorlage erstellt werden, welche initial mit den gleichen Parametern konfiguriert ist.

Informationen über die einzelnen Parameter der Darstellungen können den Objektdarstellungen entnommen werden

## VORLAGEN ERSTELLEN




Eine Vorlage kann über das "+" in der Liste auf der linken Seite hinzugefügt werden. In einem Dialog wird die gewünschte Darstellung ausgewählt und danach die Vorlage konfiguriert werden. Falls eine Darstellung ausgewählt ist, kann alternativ auf der rechten Seite auf das "+" geklickt werden um direkt eine Vorlage zur ausgewählten Darstellung zu erstellen.
Um auf einer bestehenden Vorlage aufzubauen, kann diese ausgewählt werden und unten auf "Duplizieren" geklickt werden.

## VORLAGEN LÖSCHEN

Benutzerdefinierte Vorlagen können über das Mülleimersymbol neben dem Namen in der Liste auf der linken Seite gelöscht werden. Die von IP-Symcon bereitgestellten Vorlagen können nicht gelöscht werden.

# Variablenprofile

Variablenprofile werden weiterhin unterstützt. Allerdings gibt es seit der 8.0 Variablendarstellungen als modernere und flexiblere Möglichkeit die Darstellung der Variablen anzupassen. Variablenprofile sind dennoch weiterhin über die Darstellung Legacy Profil nutzbar.

Jede Variable kann ein Variablenprofil besitzen. Variablenprofile werden primär dafür genutzt, um Variablenwerte wie True, False oder 0, 1, 2, 3 in eine für den Menschen leserliche Form zu bringen und mit den notwendigen Kontextdaten (z.B. °C) zu versehen. Das Variablenprofil gibt also an, wie eine Variable visualisiert wird.

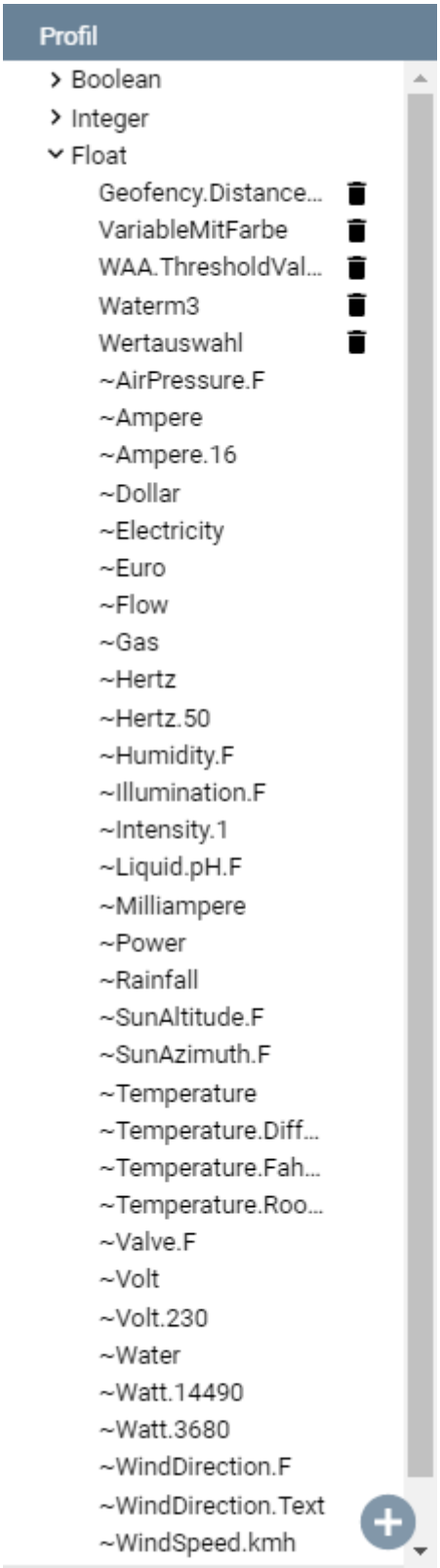
## PROFIL AUSWÄHLEN

Ein Profil kann im "Variable editieren"-Dialog über "Eigenes Profil" ausgewählt werden. Der Dialog "Variable bearbeiten" ist beim Erstellen einer Variable oder im Nachhinein via "Objektbaum" -> "Doppelklick" oder über "Objektbaum" -> "Rechtsklick" -> "Objekt bearbeiten" erreichbar.

Einige Variablen von hinzugefügten Modulen beinhalten ein "Standardprofil". Dies kann durch ein "Eigenes Profil" überschrieben werden.

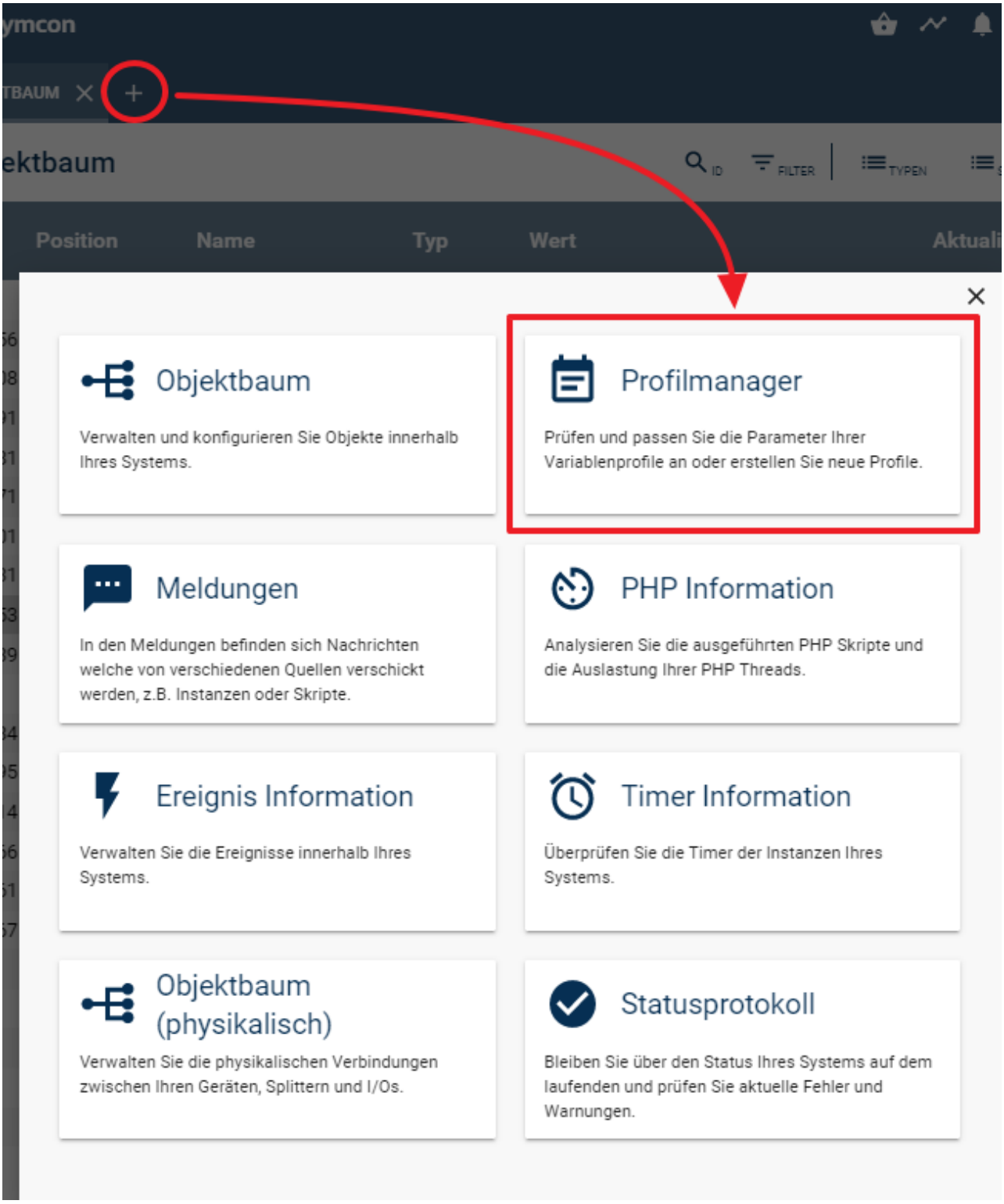



Anschließend eins der Profile auswählen.

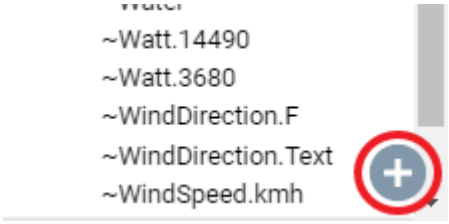


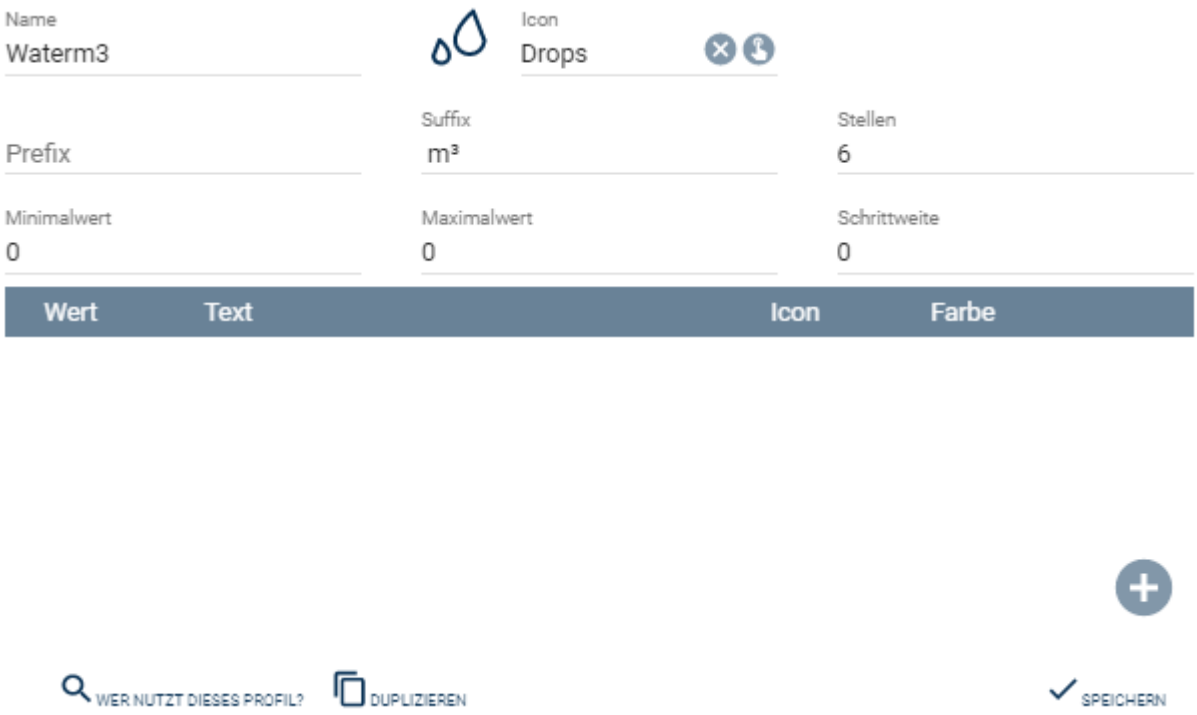

## PROFIL ERSTELLEN

Wenn von den Standardprofilen keins passend für die Variable ist können eigene erstellt werden. Diese können dann für beliebig viele Variablen verwendet werden können.
Um die Variablenprofile zu editieren, muss der Profilmanager geöffnet werden. Dieser kann im "Variable bearbeiten"-Dialog über den Fingerbutton bei "Eigenes Profil" oder über "+" -> "Profilmanager" in der Tabliste aufgerufen werden.




Es besteht die Möglichkeit, ein Profil aus den vorgefertigten Profilen auszuwählen, eines zu duplizieren oder ein Eigenes zu erstellen. Um ein eigenes Profil zu erstellen, muss linksseitig auf “+” geklickt, dem Profil ein Namen gegeben und das Profil gespeichert werden.






Obwohl die Variablenprofile nach Typen (Boolean, Integer, Float, String) unterschieden werden, kann ein Profilname jeweils nur einmalig vergeben werden.

Beispiele für verschiedene Darstellungsweisen für Variablenprofile im WebFront sind unter Objekt-Darstellung sichtbar

Standard-Profile können an der Tilde (~) vor dem Profilnamen erkannt werden. Diese Profile werden vom System erstellt und können nicht editiert werden. Bei eigenen Profilen darf die Tilde nicht im Profilnamen verwendet werden.

Es können folgende Änderungen am Profil vorgenommen werden:

| PARAMETER | TYP | BESCHREIBUNG |
|---|---|---|
| Name | Alle Typen | Name des Profils |
| Präfix | Alle Typen | String vor dem eigentlichen Wert, wie z.B. feste Beschreibung |
| Suffix | Alle Typen | String nach dem eigentlichen Wert wie z.B. °C; Sonderfall: Beim Wert % wird der Wert mit Hilfe von Min/Max in einen Prozentwert umgerechnet. (Wert = (Wert – Min) * 100 / (Max – Min)) |
| Minimalwert | Integer, Float | Kleinstmöglicher Wert der Variable für die Visualisierung |
| Maximalwert | Integer, Float | Größtmöglicher Wert der Variable für die Visualisierung |
| Schrittweite | Integer, Float | Aus Schrittweite berechnet sich die Anzahl der Felder, die in der Visualisierung zum Anklicken erstellt werden (z.B.: Mininmalwert = 0, Maximalwert = 100). Bei einer Schrittweite von 25 würden die Werte 0, 25, 50, 75, 100 zur Auswahl stehen. Ist die Schrittweite auf 0 gesetzt und sind Assoziationen vorhanden, so werden direkt alle Assoziationstexte nacheinander aufgelistet. Hierbei entfallen die sonst gezeigten Pfeile zum Durchklicken. Dieses Feld wird nur ausgewertet, wenn die zu visualisierende Variable durch Zugehörigkeit zu einer Instanz bereits eine Aktion zugewiesen bekommen hat oder ein Aktionsskript zugewiesen wurde. |
| Stellen | Float | Gibt die Anzahl der angezeigten Nachkommastellen an. |
| Icon | Alle Typen | Falls kein Icon über die Assoziationen vorhanden ist, wird das Standard-Icon verwendet. Sollte diese Feld leer bleiben, wird auf das Objekt Icon zurückgegriffen. Alle vorhandenen Icons können hier gefunden werden: WebFront Icons |
| Assoziationen | Alle Typen | **Boolean:**<br>Zu jedem der beiden Möglichen Werte (True, False) kann ein Text und ein Icon angegeben werden, welches statt des eigentlichen Wertes angezeigt wird. Es wird nur die Darstellung beeinflusst. Der Wert der Variable bleibt erhalten.<br>**Integer/Float**:<br>1.Wie bei Boolean ist es möglich, für einen Wert eine Repräsentation über einen Text und ein Icon zu wählen. Dabei muss im einfachsten Fall für jeden Wert ein Text und Icon angegeben werden.<br>2. Die zweite Möglichkeit besteht darin, Werte auszulassen, um z.B. nur wichtige Positionen, die eine Änderung benötigen, anzuzeigen. So könnten für eine Jalousie z.B. nur die Werte 0, 50, 100 interessant sein, da nur für diese Icons verwendet werden. Die Assoziationen können definiert werden, indem für die Werte 0, 50, 100 einen Eintrag hinzufügt werden. Dabei würden die Einträge für folgende Werte gültig sein: 0 (0-49), 50 (50-99) und 100 (100).<br>3. Es gibt die Möglichkeit, einen Ausdruck zu generieren, der den aktuellen Wert enthält. Dafür kann ein %d (int) oder %f (float) Platzhalter genutzt werden. So würde ein "Lampenstufe%d" z.B. als "Lampenstufe4" dargestellt werden.<br>**Hinweis:** Wird eines der beiden Felder (Text / Icon) leer gelassen, so wird der normale Wert oder das Standard Icon verwendet. Für einen leeren Text muss ein Leerzeichen verwendet werden. |

Profile werden nur für die Visualisierung verwendet. Sobald der Variablenwert außerhalb der angegebenen Minimal-/Maximalwert Grenzen liegt oder außerhalb der spezifizierten Assoziationen, kann eine falsche oder gar keine Ausgabe des jeweiligen Wertes erfolgen.

Die maximale Anzahl von gleichzeitiger Assoziationen pro Profil ist 128.

## BEISPIEL

Wenn im Profil Minimalwert=0 und Maximalwert=20 definiert sind und % als Suffix angegeben wird, aber der Variablenwert 40 ist, so wird in der Visualisierung 200% angezeigt. Die Min/Max Werte limitieren nicht den Variablenwert!

Je nach Variablentyp können als Platzhalter %d und %f genutzt werden. Um die Nachkommastellen zu beeinflussen, kann folgendes Format verwendet werden (%.2f für zwei Nachkommastellen).

Um das Prozentzeichen (%) innerhalb eines Assoziationstextes nutzen zu können, muss ein doppeltes Prozentzeichen (%%) eingegeben werden.

Eine Möglichkeit um die Variablen schon in einer formatierten Weise in Skripten zu nutzen beinhaltet die Funktion GetValueFormatted

# Automationen

Automationen werden genutzt um individuell angepasstes Verhalten zu definieren. Von IP-Symcon unterstützte Automationen sind Ablaufpläne und PHP-Skripte. Dabei nutzt ein Ablaufplan die integrierten Aktionen, welche einfach zusammenklickbar sind. Ein Skript ist die Expertenoption, welche die Programmiersprache PHP benutzt.

## AUTOMATIONEN HINZUFÜGEN

Die Anleitung wie die jeweiligen Automationen eingerichtet werden befinden sich in den jeweiligen Unterkategorien.
Es können folgende Automationen eingebunden werden.

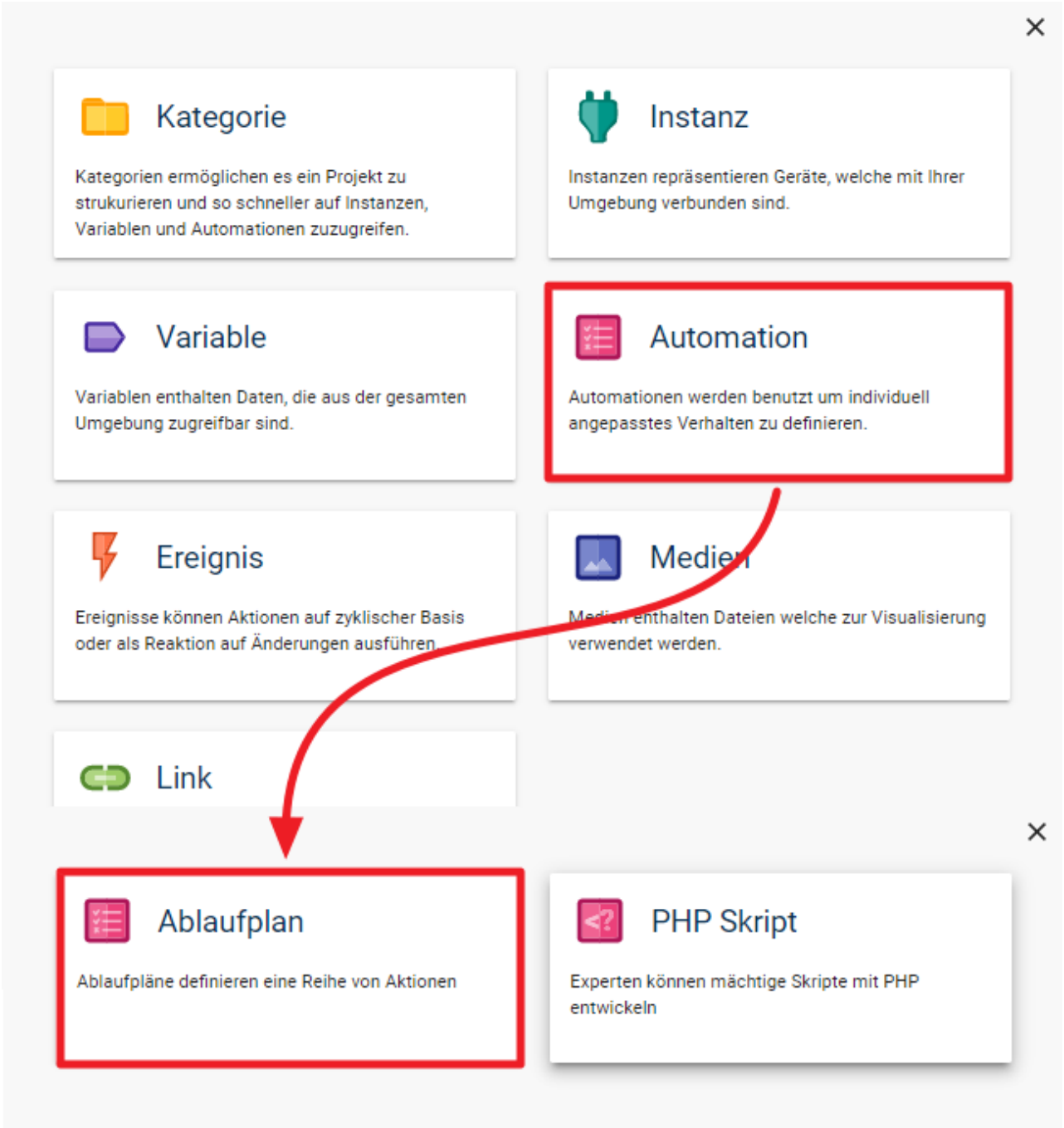
Ablaufpläne
PHP-Skripte (für Experten)

# Ablaufpläne

Ein Ablaufplan ist eine Aneinanderreichungen von Aktionen . Diese kommen ohne eine Zeile Quellcode aus und sind somit ohne irgendwelche Programmierkenntnisse einfach nutzbar.

## ABLAUFPLAN HINZUFÜGEN

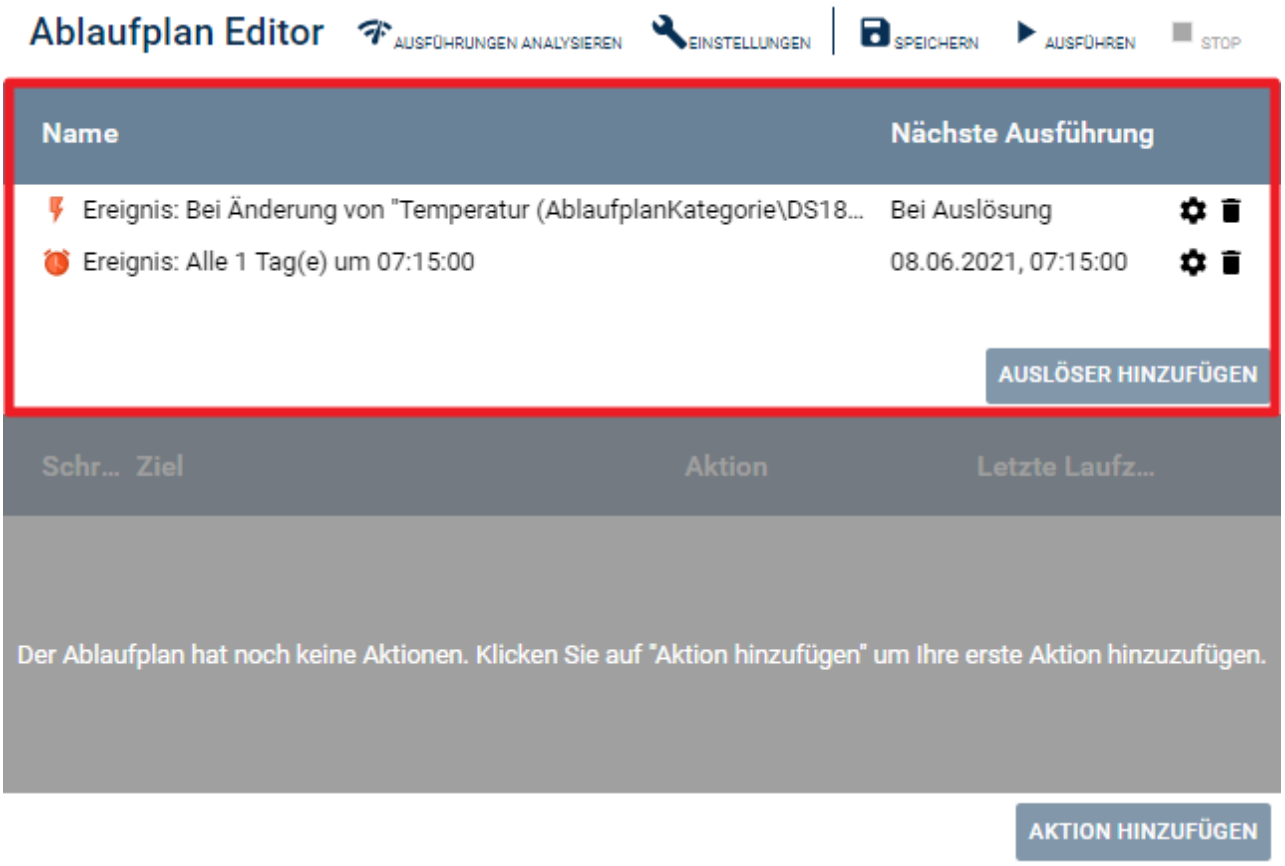
Über "+" -> "Automation" -> "Ablaufplan" kann ein neuer Ablaufplan hinzufügt werden. Es wird empfohlen dem Ablaufplan einen aussagekräftigen Namen zu geben.
Der Name ist jederzeit änderbar und der Ablaufplan kann im Objektbaum per Drag & Drop an eine andere Position verschoben werden. Alternativ kann die Erstellung per "Rechtsklick" -> "Objekt hinzufuegen" -> "Automation" -> "Ablaufplan" direkt an der gewünschten Position erfolgen.



## AUSLÖSER HINZUFÜGEN

Ein Auslöser ist ein Ereignis und ist ausschlaggebend wann der Ablaufplan gestartet wird. Die genaue Kofiguration kann unter Ereignisse nachgeschlagen werden.

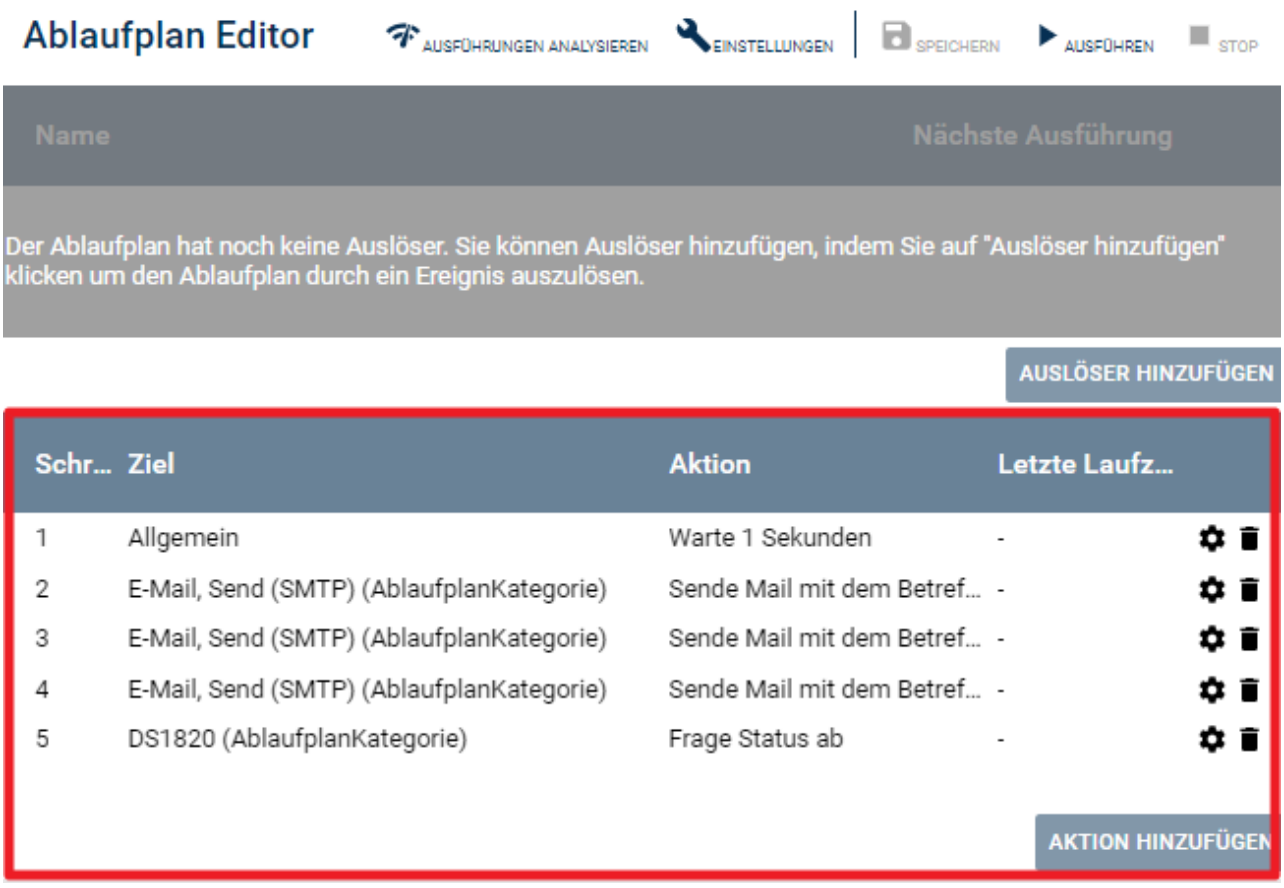
| SPALTE | BESCHREIBUNG |
|---|---|
| Name | Der Name des Auslöser im Objektbaum |
| Nächste Ausführung | Wann oder wodurch der Auslöser auslöst |
| Zahnrad | Öffnet die Konfiguration des Auslösers |
| Mülleimer | Löscht den Auslöser |




## AKTION HINZUFÜGEN

Eine Aktion ist spezifisch auf das ausgewählte Ziel oder Funktion angepasst. Eine Aktion kann das Zielobjekt oder den Fluss des Ablaufplans auf vielfältige Art modifizieren. So kann eine Aktion beispielsweise ein Schaltbefehl oder eine Bedingung sein. Auch ist es per Aktion möglich für eine bestimmte Zeit zu warten. Somit können komplexe Abläufe zusammengestellt werden ohne eine Zeile Quellcode zu schreiben.

| SPALTE | BESCHREIBUNG |
|---|---|
| Schritt | Die Position der Aktion im Ablaufplan. Manche Aktionen nutzen auch Unterschritte |
| Ziel | Gibt an, was das Ziel ist. Dies kann ein Objekt oder eine allgemeine Aktion wie z.B. "Warten" oder eine Bedingung sein |
| Aktion | Eine Beschreibung der Aktion die ausgeführt werden soll |
| Letzte Laufzeit | Gibt an wie lang der Schritt bei der letzten Ausführung gebraucht hat |
| Zahnrad | Öffnet die Konfiguration der Aktion |
| Mülleimer | Löscht die Aktion |






Erscheint ein rotes Ausrufezeichen in der Liste war die letzte Ausführung der Aktion fehlerhaft. Durch ein Klick auf das Ausrufezeichen kann der Inhalt der Fehlermeldung angezeigt werden.

### KONTEXTMENÜ

Mit einem Rechtklicks in die Liste der Aktionen, werden weitere Optionen in einem Kontextmenü angezeigt.

| LISTE | BESCHREIBUNG |
|---|---|
| Aktion davor/danach hinzufügen | Es wird eine Aktion vor oder nach der Zeile, welche angeklickt wurde, hinzugefügt. |
| Duplizieren | Die angeklickte Aktion wird dupliziert, falls vorhanden mitsamt Verschachtelung. Das duplizierte Element wird direkt "danach" einsortiert. |

## ALLGEMEINE ELEMENTE

| MENÜPUNKT | BESCHREIBUNG |
|---|---|
| Ausführungen Analysieren | Dient zur Analyse von Ausführungen bei denen fehlerhafte, abgebrochene und erfolgreiche Aktionen aufgeführt werden. Siehe unten |
| Einstellungen | Hier kann das Verhalten des Ablaufplans bestimmt werden. ( Siehe: Verhalten bei laufender Ausführung und Verhalten bei Fehler) |
| Speichern | Speichert die aktuelle Konfiguration des Ablaufplans ab |
| Ausführen | Dies speichert den Ablaufplan ab und führt ihn aus |
| Stop | Stoppt die aktuelle Ausführung des Ablaufplans |

## AUSFÜHRUNG ANALYSIEREN

Die Ausführung eines Ablaufplans durchläuft verschiedene Status. Diese sind textlich beschrieben, sowie farblich kodiert. Auch können die einzelnen Ausführungen ausgeklappt werden. Dies gibt die Chance einzelne Schritte nachzuvollziehen.

| STATUS | BESCHREIBUNG |
|---|---|
| Eingreiht | Es befindet sich in der Warteschlange (Farbcode: Weiß) |
| Läuft | Die Ausführung des Ablaufplans läuft noch (Farbcode: Weiß) |
| Abgebrochen | Die Ausführung des Ablaufplans wurde abgebrochen (Farbcode: Gelb) |
| Abgeschlossen | Erfolgreich abgeschlossen (Farbcode: Grün) |
| Abgeschlossen (Fehler) | Die Ausführung des Ablaufplans wurde ausgeführt, es kam aber zu Fehler/n (Farbcode: Gelb) |
| Fehler | Die gesamte Ausführung des Ablaufplans hat abgebrochen (Farbcode: Rot) |
| Übersprungen | Wird aufgerufen durch: "Bei Ausführung weitere Ausführung ablehnen" (Farbcode: Gelb) |

## VERHALTEN BEI LAUFENDER AUSFÜHRUNG

Dies beschreibt das Verhalten, sollte die erneute Ausführung des Plans bei bereits laufender Ausführung angestoßen werden.

| VERHALTEN | BESCHREIBUNG |
|---|---|
| Bei Ausführung weitere Ausführung ablehnen | Weitere Ausführungen des Ablaufplans werden ignoriert |
| Bei neuer Ausführung laufende Ausführung abbrechen | Die aktuelle Ausführung des Ablaufplans wird abgebrochen und von vorne gestaret |
| Ausführungen in Warteschlange hinzufügen | Bei mehrfachen Aufruf oder Auslösung wird die Ausführung des Ablaufplans einer Warteschlange hinzugefügt und nacheinander aufgerufen. |

## VERHALTEN BEI FEHLER

| VERHALTEN | BESCHREIBUNG |
| --- | --- |
| Bei Fehler die restliche Ausführung abbrechen | Bei einem Fehler wird die Ausführung des Ablaufplans abgebrochen |
| Bei Fehler die restliche Ausführung fortsetzen | Bei einem Fehler wird dieser protokolliert und es wird der nächste Schritt ausgeführt |
| Bei Fehler den Schritt wiederholen | Bei einem Fehler wird die Ausführung des Ablaufplans wiederholt bis entweder die Ausführung erfolgreich ist oder die Anzahl der eingestellten Wiederholungen erreicht wurde. Die Option "Setze die Ausführung trotz fehlgeschlagener Wiederholungen fort" besagt ob nach erfolglosen Wiederholungen der Schritt übersprungen wird oder die Ausführung des Ablaufplans komplett abgebrochen wird |

# Aktionen

Eine Aktion ist ein Kommando, welches ohne komplexe Programmierung oder Skript auskommt. So werden abhängig vom ausgewählten Ziel verschiedenste Aktionen aufgelistet. Jede Aktion ermöglicht individuelle Einstellmöglichkeiten und Optionen.

## AKTION HINZUFÜGEN

Eine Aktion wird im Ablaufplan, Ereignis, Instanzkonfiguration oder Befehl testen genutzt. Wie diese in den jeweiligen Möglichkeiten hinzugefügt werden ist in den jeweiligen Abschnitten beschrieben. Aber bei allen Möglichkeiten werden die Aktionen nach dem gleichen Schema konfiguriert.

Schritt 1: Auswahl des Ziels - Dadurch werden die möglichen Aktionen geladen und angezeigt

Schritt 2: Auswahl der Aktion - Die tatsächliche Auswahl einer Aktion (Schnellfiltersuche möglich)

** Die Aktionen sind in verschiedene Kategorien einsortiert und ermöglichen so einen schnellen Zugriff

Schritt 3: Konfigurieren der Parameter - Hier werden die tatsächlichen Werte eingestellt für die Aktion eingestellt.

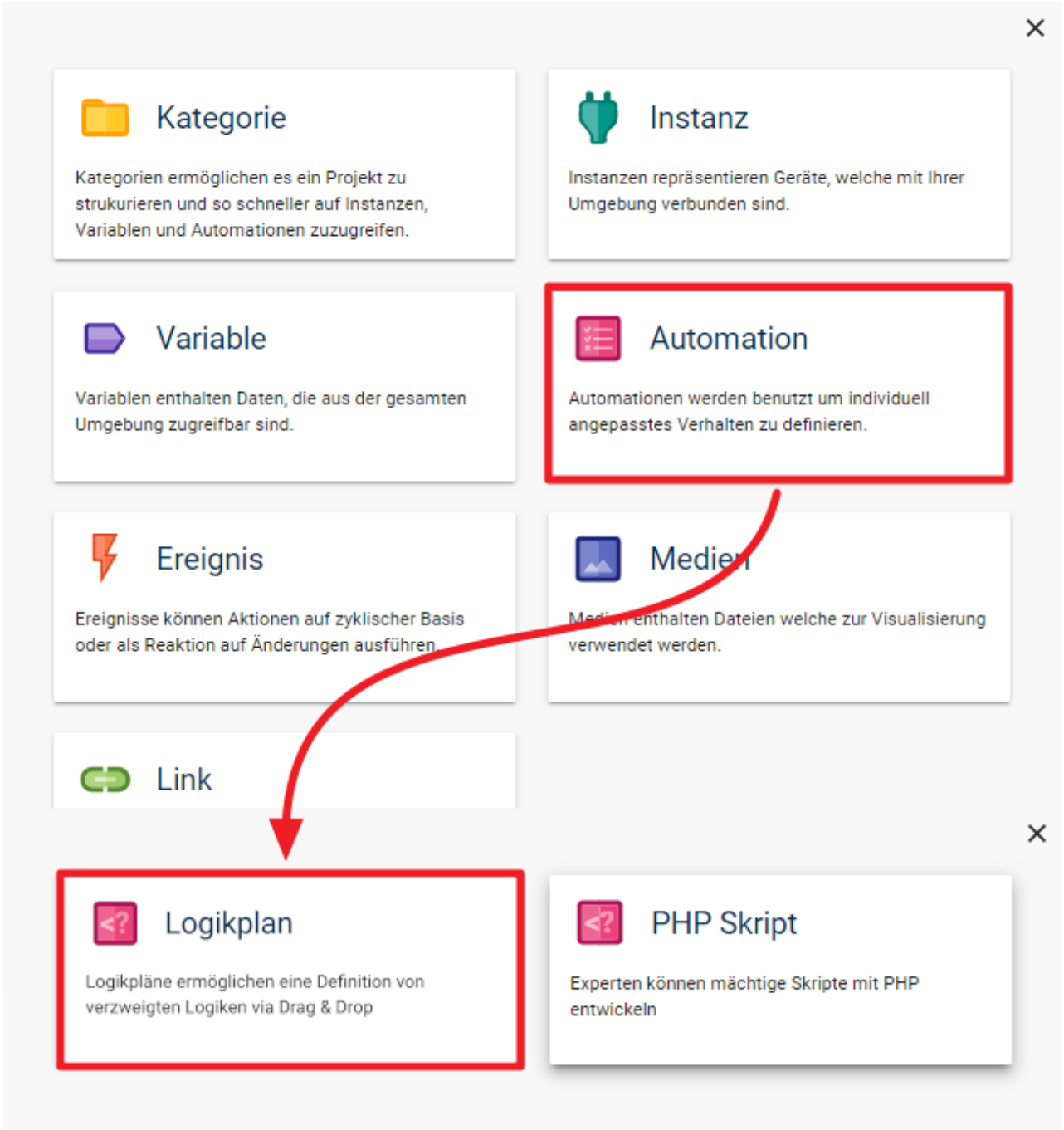
# Logikpläne

Benötigt: IP-Symcon >= 7.0
Logikpläne ist ein Tool, mit dem individuelle Abläufe und Automatisierungen in IP-Symcon erstellen werden können. Typische Anwendungsfälle sind:

- Gruppenschalter für Licht
- Treppenhausautomat
- Sensorauswertung

## LOGIKPLAN HINZUFÜGEN

Über "+" -> "Automation" -> "Logikplan" kann ein neuer Logikplan hinzufügt werden. Es wird empfohlen dem Logikplan einen aussagekräftigen Namen zu geben.
Der Name ist jederzeit änderbar und der Logikplan kann im Objektbaum per Drag & Drop an eine andere Position verschoben werden. Alternativ kann die Erstellung per "Rechtsklick" -> "Objekt hinzufuegen" -> "Automation" -> "Logikplan" direkt an der gewünschten Position erfolgen.




# BESTANDTEILE DES EDITORS

Der Logikplan gliedert sich in folgende Bereiche:

| NAME | BESCHREIBUNG |
|---|---|
| Menu | Das Menu stellt zentrale Funktionen wie das Speichern, das Zoomen oder das Ausführen eines Logikplans zur Verfügung. |
| Panels | Der Logikplan stellt 2 Panels zur Verfügung:<br>- Das Repository mit der Liste aller verfügbaren Module<br>- Den PropertyEditor zur Anzeige und zum Editieren von Eigenschaften<br>Verwenden Sie Drag & Drop um die Position der Panels zu verändern, es stehen 4 vordefinierte Positionen zur Verfügung. Wenn Sie mehrere Panels an einer Position platzieren, werden automatisch Tabs gebildet. |
| Zeichenfläche | Die Zeichenfläche ist der zentrale Bestandteil des Editors und ermöglicht das Platzieren und Editieren der einzelnen Module. |

# Menü

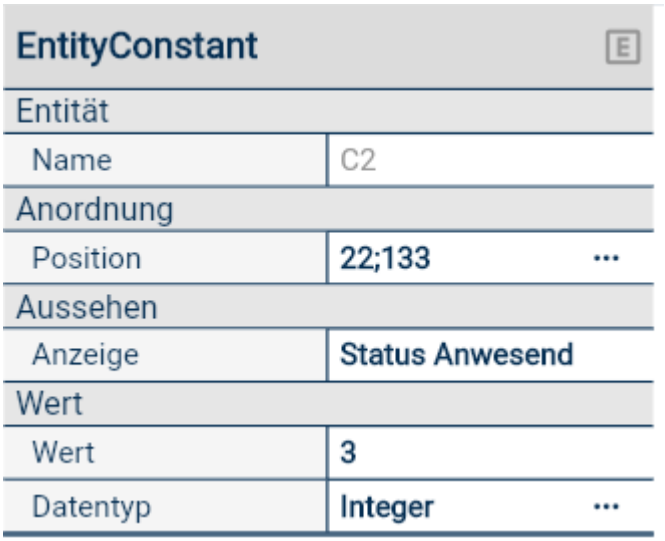



Die folgenden Funktionen stehen zur Verfügung:

| NAME | BESCHREIBUNG |
|---|---|
| Administration | Betätigen Sie den Button Administration um die verfügbaren Aktionen für den Logikplan anzuzeigen. Nach der Aktivierung des Buttons wird das Menu für die Administration geöffnet.<br>Folgende Aktionen stehen zur Verfügung:<br>- Logikplan exportieren<br>- Logikplan importieren |
| Zoom | Der Logikplan bietet die Möglichkeit eine View in Schritten von 10% beliebig zu verkleinern und zu vergrößern, Details dazu finden Sie im Bereich Zeichenfläche. |
| Rückgängig und Wiedererstellen | Menügruppe zum Annullieren bzw. Wiederherstellen von Änderungen, Details dazu finden Sie im Bereich Zeichenfläche. |
| Zurück und Vorwärts navigieren | - Menügruppe zum Navigieren in der Historie, diese Steuerelemente sind nur im Modus Historie verfügbar.<br>- Betätigen Sie den Button Zurück um in der Historie zurück zu navigieren. Nach der Betätigung des Buttons werden die nächst älteren Daten der Historie geladen und angezeigt.<br>- Betätigen Sie den Button Vorwärts um in der Historie vorwärts zu navigieren. Nach der Betätigung des Buttons werden die nächst neueren Daten der Historie geladen und angezeigt. |
| Bearbeiten | Betätigen Sie den Button Bearbeiten um den Editier Modus des Logikplan Editors zu aktivieren. Nach der Aktivierung des Buttons kann der Logikplan bearbeitet werden. |
| Live | Betätigen Sie den Button Live um den Live Modus des Logikplan Editors zu aktivieren. Nach der Aktivierung des Buttons werden die aktuellen Werte der Logikplan Ausführung auf der Zeichfäche dargestellt. |
| Historie | Betätigen Sie den Button Historie um den Historie Modus des Logikplan Editors zu aktivieren. Nach der Aktivierung des Buttons können Sie in der Historie der einzelnen Ausführungen navigieren. |
| Ausführen | Betätigen Sie den Button Ausführen um eine manuelle Ausführung des Logikplans zu starten. Nach der Aktivierung des Buttons wird der Automation Logikplan in IP-Symcon ausgeführt und die aktuellen Werte auf der Zeichenfläche angezeigt. |
| Speichern | Betätigen Sie den Button Speichern um Ihre Änderungen am Logikplan zu speichern. Nach der Aktivierung des Buttons wird der Logikplan am IP-Symcon Server gespeichert. Bei Umschaltung in den Live oder Historie Modus, wird der Logikplan automatisch gespeichert. |
| Info | Betätigen Sie den Button Info um die Programminformationen anzuzeigen. Nach der Aktivierung des Buttons wird der Dialog mit den Lizenz- und Programminformationen angezeigt. |

# Property-Editor

Das Panel Property-Editor zeigt alle verfügbaren Eigenschaften eines Steuerelements an.




Jede Property besteht aus einer Bezeichnung (linke Spalte) und einem dazugehörenden Wert (Für Techniker: Key-Value-Pair; rechte Spalte).
Wenn Sie ein neues Objekt auf der Zeichenfläche anlegen, werden dessen Properties mit Standardwerten aus IP-Symcon belegt. Um Properties an Ihre Bedürfnisse anzupassen:

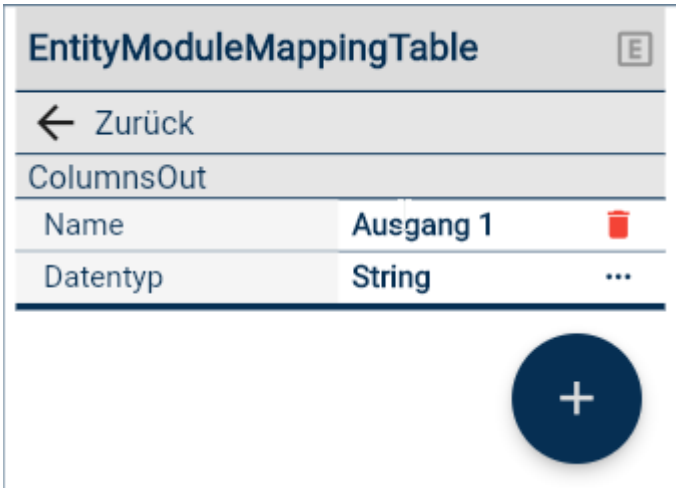
Klicken Sie in die jeweiligen Wertefelder und ändern Sie die Werte.
Speichern Sie Ihre Änderungen von Zeit zu Zeit (dies kann je nach Ihrer Verbindungsgeschwindigkeit einige Sekunden dauern).

## ÜBERSICHT


| | |
|---|---|
| Ausgänge | Ausgänge eines Mapping Moduls definieren |
| Die Properties | Bedeutung der Properties |
| Eingänge | Eingänge eines Mapping Moduls definieren. |
| Mappings | Bearbeiten der Zuordnungen eines Mapping Moduls |

# Ausgänge

Um die Ausgänge eines Mapping Moduls zudefinieren betätigen Sie den Menu Button der Eigenschaft Ausgänge Ihres Mapping Moduls. Die Definition der Eingänge wird geladen und angezeigt:




Um einen Ausgang des Mapping Moduls zu konfigurieren:

Fügen Sie mittels + einen neuen Eintrag zur Auswahl hinzu.
Wählen Sie nacheinander die Einträge aus und bearbeiten Sie deren Properties (siehe unten).
Wenn Sie alle Ausgänge definiert haben, verlassen Sie den Editor mit dem Button Zurück.

## FOLGENDE OPTIONEN STEHEN ZUR VERFÜGUNG

| NAME | BESCHREIBUNG |
|---|---|
| Name | Name des Ausgangs |
| Datentyp | Definiert den Datentyp des Ausgangs, folgende Optionen stehen zur Auswahl: Boolean, Integer, Float, String |

# Die Properties

Die hier beschriebenen Properties sind für Ihre Arbeit mit Logikpläne von Bedeutung. Eine Beschreibung spezifischer Properites eines Moduls finden Sie auch noch im Repository .

## GRUPPE: ENTITY

| NAME | BESCHREIBUNG |
| --- | --- |
| EntityName | Der Name des Objektes im Logikplan. |

## GRUPPE: VARIABLE

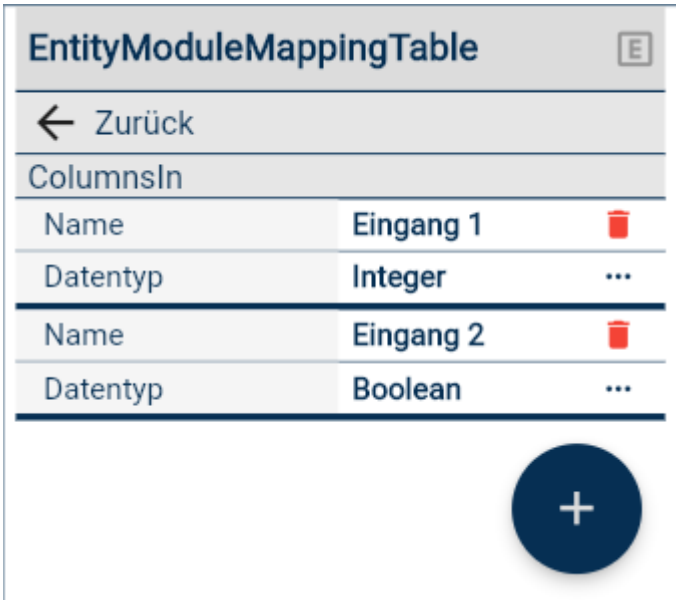
| NAME | BESCHREIBUNG |
| --- | --- |
| ID | ID der Variable in IP-Symcon. |
| Name | Name der Variable in IP-Symcon. |

## GRUPPE: INSTANZ

| NAME | BESCHREIBUNG |
| --- | --- |
| ID | ID der Instanz in IP-Symcon. |
| Name | Name der Instanz in IP-Symcon. |

# Eingänge

Um die Eingänge eines Mapping Moduls zudefinieren betätigen Sie den Menu Button der Eigenschaft Eingänge Ihres Mapping Moduls. Die Definition der Eingänge wird geladen und angezeigt:




Um einen Eingang des Mapping Moduls zu konfigurieren:

Fügen Sie mittels + einen neuen Eintrag zur Auswahl hinzu.

Wählen Sie nacheinander die Einträge aus und bearbeiten Sie deren Properties (siehe unten).

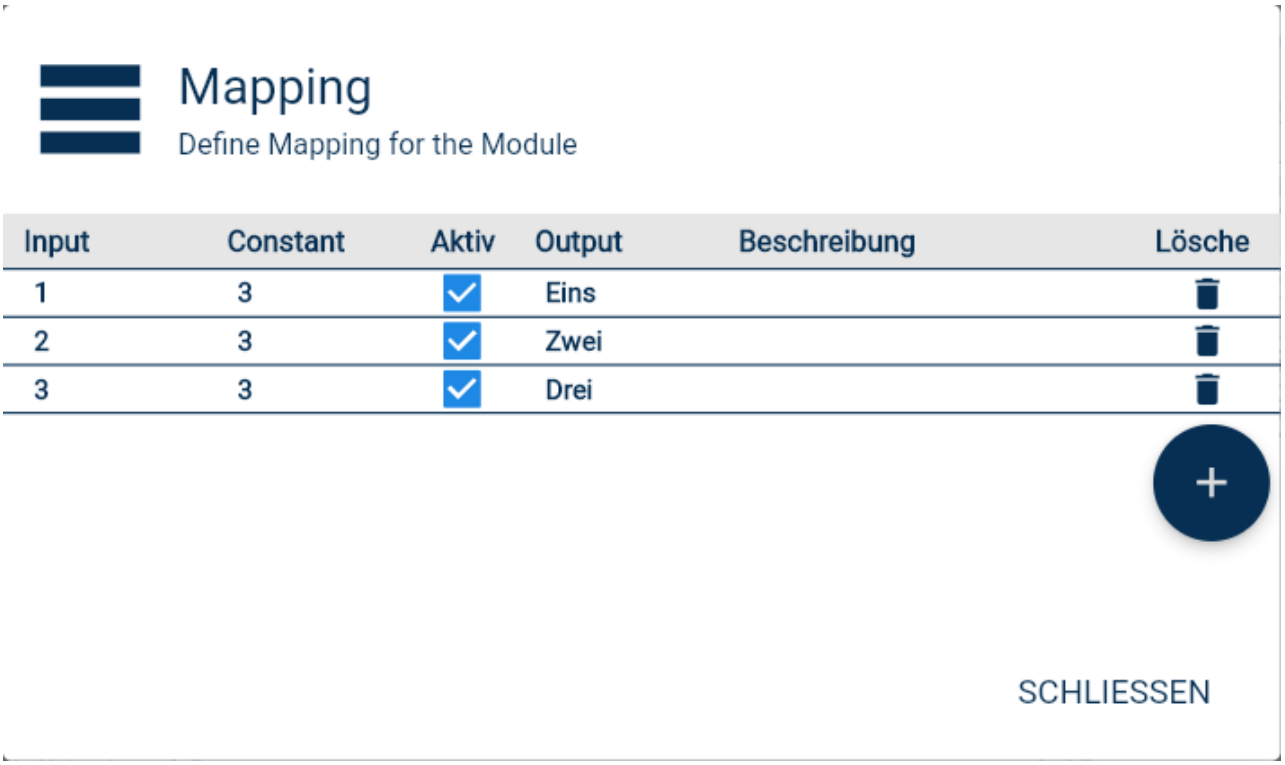
Wenn Sie alle Eingänge definiert haben, verlassen Sie den Editor mit dem Button Zurück.

Folgende Optionen stehen zur Verfügung

| NAME | BESCHREIBUNG |
|---|---|
| Name | Name des Eingangs |
| Datentyp | Definiert den Datentyp des Eingangs, folgende Optionen stehen zur Auswahl: Boolean, Integer, Float, String |

# Mappings

Um die einzelnen Mappings eines Mapping Moduls zudefinieren betätigen Sie den Button Edit Ihres Mapping Moduls. Die Definition der Mappings wird geladen und angezeigt:




Der Editor stellt für jeden Eingang und für jeden Ausgang eine Spalte zur Verfügung, bei der Evaluierung des Moduls wird der erste Eintrag in der Liste verwendet, wo alle Werte am Eingang zusammenpassen.
Um ein Mapping des Mapping Moduls zu konfigurieren:

Fügen Sie mittels + einen neuen Eintrag zur Auswahl hinzu.
Wählen Sie für jeden Eingang den Wert für das Mapping.
Mit "Aktiv" können Sie ein Mapping deaktivieren ohne es sofoft löschen zu müssen.
Wählen Sie für jeden Ausgang den gewünschten Wert.
Beschreibung ermöglicht eine kurze Dokumentation der aktuellen Mapping Zeile. Dieser Eintrag hat keine Auswirkung auf die eigentliche Ausführung des Moduls.
Wenn Sie alle Ausgänge definiert haben, verlassen Sie den Editor mit dem Button Zurück.

# Repository

Das Panel Repository bietet alle verfügbaren Logikplan Module, die man einfach per "Drag and Drop" auf der Zeichenfläche platzieren kann.
Die folgenden Möglichkeiten bietet das Repository um Module zu finden:

## ALPHABETISCHE SORTIERUNG

Wählen Sie die Option Alphabetische Sortierung um alle verfügbaren Module im Repository nach der Bezeichnung zu sortieren.



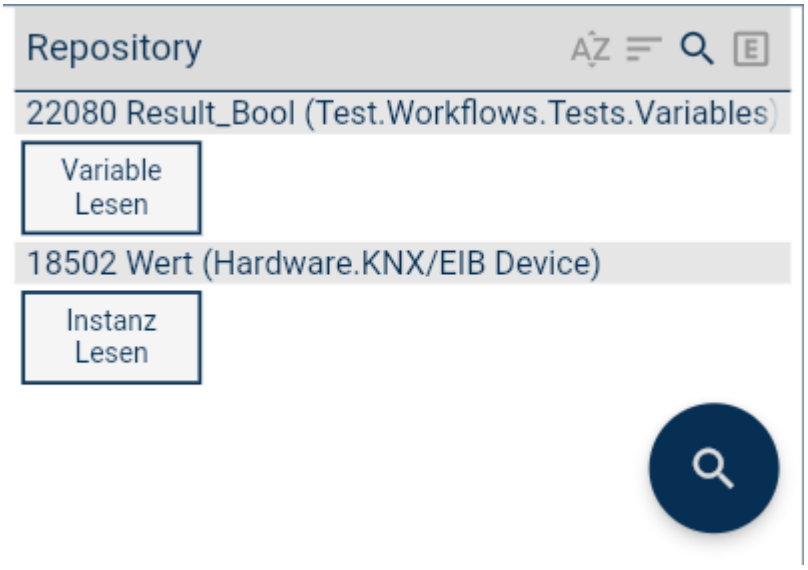

## GRUPPIERT NACH VERWENDUNG

Wählen Sie die Option Gruppiert nach Verwendung () um alle verfügbaren Module im Repository nach ihrer Verwendung zu gruppieren:

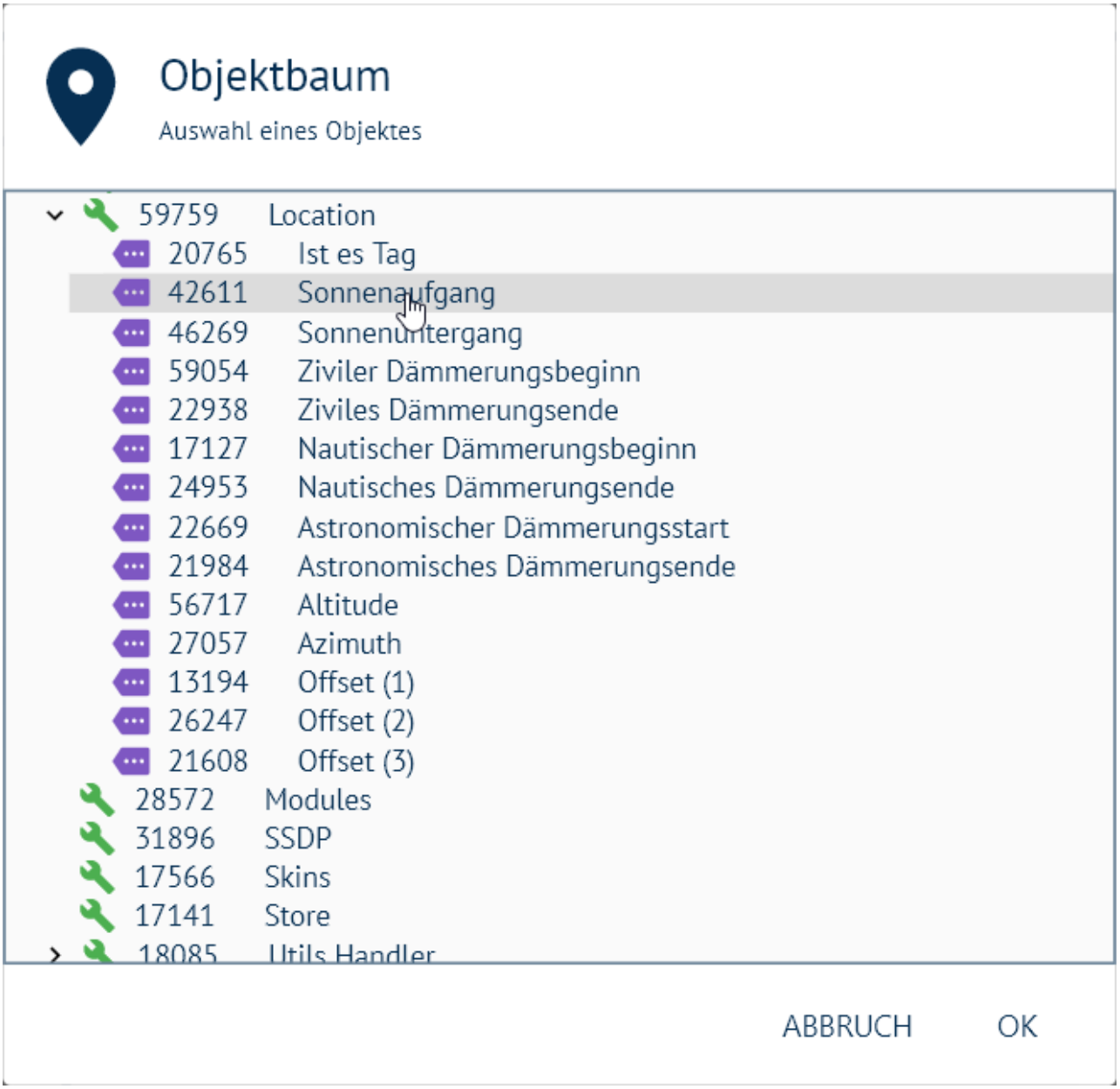



## SUCHE NACH ID

Wählen Sie die Option Suche nach ID um alle verfügbaren Module zu suchen die für eine bestimmte ID in IP-Symcon verwendet werden können.




Klicken Sie auf den Button "Suchen" um Module für ein IP-Symcon Objekt zu suchen, Nach Betätigung des Button "Suchen" wird der Dialog zur Suche einer ID geöffnet:




Nach Auswahl einer ID und der Bestätigung durch OK werden alle verfügbaren Module für das gewählte Objekt angezeigt.

## EXPERTEN MODUS

Wählen Sie die Option Experten Modus () um den "Experten Modus" zu aktivieren, nach dem Aktivieren der Option zeigt das Repository auch selten benötigte Module an.



Einen Überblick über die relevanten IP-Symcon-Properties, unabhängig vom Objekttyp, finden Sie im Abschnitt Die Properties.

## ÜBERSICHT DER MODULE

Zur besseren Übersicht wurden die Module des Editors nach ihrer Verwendung gruppiert.

## AKTIONEN

Ausführen spezieller Aktionen

| | |
|---|---|
| Eigene Action | Ausführen von beliebigen PHP Code |
| Funktionsaufruf | Aufrufen einer IP-Symcon Instanz Funktion. |
| IRTrans | Versenden von IR Befehlen. |
| Log Meldung | Schreiben der Log Meldung. |
| Mail Benachrichtigung | Versenden einer Mail Benachrichtigung. |
| Push Benachrichtigung | Versenden einer Push Benachrichtigung. |
| Skript Ausführung | Aufruf von Skripten |
| Variable mit ID | Modifikation eine Variable mit Referenzierung per ID zur Laufzeit. |

## BEDINGUNG

Auswertung von Bedingungen

| | |
|---|---|
| Bedingter Wert | Wert setzen wenn eine bestimmte Bedingung zutrifft |
| Bedingung | Bedingte Verzweigung |

## DOKUMENTATION: KOMMENTAR

Kommentare in den Logikplan einfügen.

## EREIGNISSE

Ereignisse definieren um die Automation automatisch zu starten

| | |
|---|---|
| Bei Änderung | Ausführung des Logikplans bei Veränderung einer Variable. |
| Bei Aktualisierung | Ausführung des Logikplans bei Aktualisierung einer Variable. |
| Bei bestimmten Wert | Ausführung des Logikplans bei erreichen eines bestimmten Variablen Wertes. |
| Bei Überschreitung | Ausführung des Logikplans bei Überschreitung eines Variablen Wertes. |
| Bei Unterschreitung | Ausführung des Logikplans bei Unterschreitung eines Variablen Wertes. |
| Wochenplan (1 Gruppe) | Wochenplan mit 1 Gruppe. |
| Wochenplan (2 Gruppen) | Wochenplan mit 2 Gruppen. |
| Wochenplan (7 Gruppen) | Wochenplan mit 7 Gruppen. |

## INFORMATIONEN

Informationen auslesen

| | |
|---|---|
| Skript Informationen | Inforamtionen zur aktuellen Skriptausführung |
| System Informationen | Datum und Uhrzeit des IP-Symcon Systems |
| Variablen Inforamtionen | Informationen zu einer Variable |

## KONSTANTE WERTE

Konstante Werte definieren

## LOGIK

Logische Module

| | |
|---|---|
| Nicht | Logisches Nicht |
| Oder | Logisches Oder |
| Und | Logisches Und |

## MAPPING

Komplexe Mapping Logik erstellen.

## MATHEMATIK

Mathematische Operationen ausführen.

| | |
|---|---|
| Addition | Addition zweier Werte |
| Division | Division zweier Werte |
| Formel | Berechnung eines Wertes basierend einer benutzerdefinierten Formel. |
| Maximum | Maximum mehrerer Werte |
| Minimum | Minimum mehrerer Werte |
| Modulo | Berechnung des Modulo zweier Werte |
| Multiplikation | Multiplikation zweier Werte |
| Runden | Runden eines Wertes |
| Subtraktion | Subtraktion zweier Werte |

## SUBLOGIKPLAN

Mehrere Module zu einem Sublogikplan zusammenfassen.

| | |
|---|---|
| Sublogikplan Parameter | Sublogikplan Parameter erstellen. |
| Sublogikplan Ausgang | Sublogikplan Ausgang erstellen. |
| Sublogikplan Eingang | Sublogikplan Eingang erstellen. |
| Sublogikplan Modul | Sublogikplan erstellen. |

## TIMER

Definition von Timern.

| | |
|---|---|
| Einmal am Tag | Zyklischer Timer mit täglicher Auslösung. |
| Einmaliger Timer | Einmalige Ausführung eines Timers. |
| Timer Variable | Timer mit Variable zur Anzeige der Restlaufzeit. |
| Timer Zyklisch Minute | Zyklischer Timer mit Auslösung im Minuten Intervall. |
| Timer Zyklisch Sekunde | Zyklischer Timer mit Auslösung im Sekunden Intervall. |
| Timer Zyklisch Stunde | Zyklischer Timer mit Auslösung im Stunden Intervall. |

## UMWANDLUNG

Konvertierung von Werten in einen anderen Datentyp

| | |
|---|---|
| Datum zu String | Konvertiert einen Unix Timestamp in einen String |
| Datum zu Zeit | Konvertiert eines Unix Timestamp in die einzelnen Zeit Komponenten. |
| Meldung formatieren | Formatiert mehrere Werte zu einer Meldung. |
| Wert inventieren | Invertiert einen Wert. |
| Wert zu Boolean | Konvertiert einen Wert in einen Boolean. |
| Wert zu Float | Konvertiert einen Wert in einen Float. |
| Wert zu Integer | Konvertiert einen Wert in einen Integer. |
| Wert zu String | Konvertiert einen Wert in einen String. |
| Wertebereich konvertieren | Konvertieren eines Werte von einem Wertebereich in einen anderen. |
| Zeit zu Datum | Konvertiert einzelne Zeit Komponenten in einen Unix Timestamp. |

## VARIABLEN LESEN/SCHREIBEN

Lesen und Schreiben von Variablen und Instanzen.

| | |
|---|---|
| Instanz Lesen | Instanzen als Eingang eines Logikplans |
| Instanz Schreiben | Instanz als Ausgang eines Logikplans |
| Variable Lesen | Variable als Eingang eines Logikplans |
| Variable Schreiben | Variable als Ausgang eines Logikplans |
| Logikplan Variable | Variable durch den Logikplan erstellen |

## VERGLEICHEN

Werte vergleichen

| | |
|---|---|
| Gleich | Vergleichen von Werten auf Gleichheit oder Ungleichheit |
| Größer/Kleiner | Vergleichen von Werten auf Größer, Größergleich, Gleich, Kleinergleich oder Kleiner. |
| Im Wertebereich | Überprüft einen Wert auf einen bestimmten Wertebereich |

# Aktionen

Module der Gruppe "Aktionen" ermöglichen das Ausführen von IP-Symcon Automation Skripte oder diverser anderer IP-Symcon Aktionen.

| | |
|---|---|
| Eigene Action | Ausführen von beliebigen PHP Code |
| Funktionsaufruf | Aufrufen einer IP-Symcon Instanz Funktion. |
| IRTrans | Versenden von IR Befehlen. |
| Log Meldung | Schreiben der Log Meldung. |
| Mail Benachrichtigung | Versenden einer Mail Benachrichtigung. |
| Push Benachrichtigung | Versenden einer Push Benachrichtigung. |
| Skript Ausführung | Aufruf von Skripten |
| Variable mit ID | Modifikation eine Variable mit Referenzierung per ID zur Laufzeit. |

# Eigene Action

Das Module "Eigene Aktion" bietet die Möglichkeit zum Ausführen von beliebigen PHP Code.




## PROPERTIES

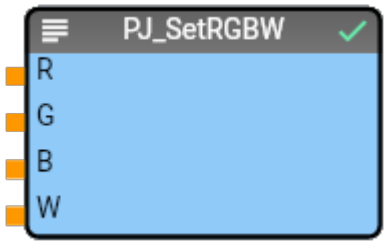
| NAME | BESCHREIBUNG |
| --- | --- |
| Anzahl Eingänge | Anzahl der Eingänge, die das Modul zur Verfügung stellen soll. |
| PHP Code | PHP Code, der ausgeführt werden soll. Die Eingänge können im PHP Code über die Variablen $value1 bis $value8 referenziert werden. |

## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
| --- | --- |
| Wert x | Abhängig von der Eigenschaft "Anzahl Eingänge" stehen 1 bis 8 Datenpunkte am Eingang bereit. |

# Funktionsaufruf

Das Module "Funktionsaufruf" bietet die Möglichkeit um eine IP-Symcon Instanzfunktion aufzurufen.




## PROPERTIES

| NAME | BESCHREIBUNG |
| --- | --- |
| ID | ID der Instanz wo eine Funktion aufgerufen werden soll |
| Funktionsname | Name der Funktion die aufgerufen werden soll. |

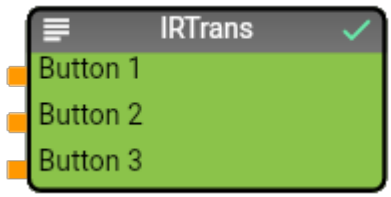
Nach Eingabe einer gültigen ID, stellt die DropDown Funktion bereits alle möglichen Funktionsnamen zur Verfügung.

## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
| --- | --- |
| Parameter | Die Eingänge des Modules variieren je nach gewählter Funktion |

# IRTrans

Das Module "IRTrans" bietet die Möglichkeit zum Ansteuern einer IRTrans Instanz. Es können bis zu 3 Buttons angesteuert werden.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Fernbedienung | Fernbedienung, die für das Senden der Buttons verwendet werden soll. |
| Instanz ID | Instanz, die für das Versenden der IR Befehle verwendet werden soll. |

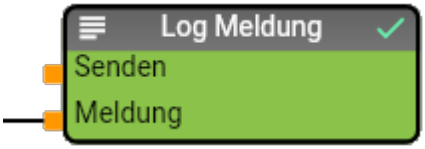
Sollen Befehle zu verschiedenen Geräten versendet werden, müssen mehrere IRTrans Module verwendet werden.

## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
|---|---|
| Button 1 | 1. IR Befehl, der gesendet werden soll. |
| Button 2 | 2. IR Befehl, der gesendet werden soll. |
| Button 3 | 3. IR Befehl, der gesendet werden soll. |

# Log Meldung

Das Module "Log Meldung" bietet die Möglichkeit zum Schreiber einer Meldung in das IP-Symcon Log.




## PROPERTIES

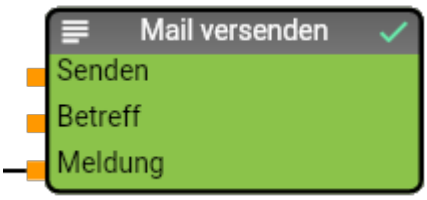
| NAME | BESCHREIBUNG |
|---|---|
| Sender | Text der im Feld Sender der Log Meldung verwendet werden soll. |

## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
|---|---|
| Senden | Durch TRUE auf diesem Eingang wird das Versenden der Nachricht ausgelöst. Wird der Eingang nicht verbunden, wird immer ein Mail Versandt ausgelöst. |
| Nachricht | Text für die Log Meldung. |

# Mail Benachrichtigung

Das Module "Mail versenden" bietet die Möglichkeit zum Versenden von Mail Nachrichten.




## PROPERTIES

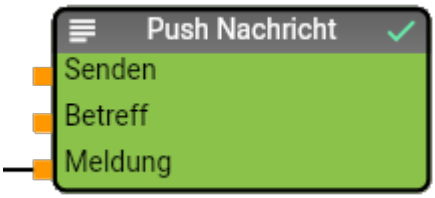
| NAME | BESCHREIBUNG |
|---|---|
| InstanzID Mail | ID der SMTP Instanz, die für das Versenden der Mails verwendet werden soll. |

## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
|---|---|
| Senden | Durch TRUE auf diesem Eingang wird das Versenden der Nachricht ausgelöst.<br>Wird der Eingang nicht verbunden, wird immer ein Mail Versandt ausgelöst. |
| Betreff | Text der im Betreff der Nachricht verwendet werden soll. |
| Nachricht | Text für die Mail Nachricht. |

# Push Benachrichtigung

Das Module "Push Nachricht" bietet die Möglichkeit zum Versenden einer Push Benachrichtigung auf ein mobiles Gerät.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| InstanzID Benachrichtigung | ID der Benachrichtigungs Instanz (Notification Control), die für das Versenden der Push Benachrichtigung verwendet werden soll. |
| ObjektID Benachrichtigungen | ID des Objektes (View oder Visualisierung-Konfigurator) an das die Push Benachrichtigung gesendet werden soll. |

## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
|---|---|
| Senden | Durch TRUE auf diesem Eingang wird das Versenden der Nachricht ausgelöst. Wird der Eingang nicht verbunden, wird immer eine Benachrichtigung ausgelöst. |
| Betreff | Text der im Betreff der Nachricht verwendet werden soll. |
| Nachricht | Text für die Nachricht. |

# Skript Ausführung

Das Module "Skript Ausführung" bietet die Möglichkeit zur Einbindung von anderen Skripten bzw. Automations in den Logikplan.




## PROPERTIES

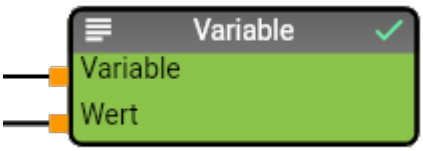
| NAME | BESCHREIBUNG |
|---|---|
| SkriptID | ID des Skriptes. |
| Sender | Wert für die Variable SENDER beim Aufruf des Skriptes. |
| VariableID | Wert für die Variable VARIABLE beim Aufruf des Skriptes. |

## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
|---|---|
| Wert | Wert für die Variable VALUE beim Aufruf des Skriptes. |

# Variable mit ID

Das Module "Variable mit ID" bietet die Möglichkeit zum Verändern einer Variable die erst zur Laufzeit des Logikplans über die ID der Variable referenziert wird.




## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
| --- | --- |
| Variable | ID der Variable, die verändert werden soll. |
| Wert | Wert der gesetzt werden soll. |

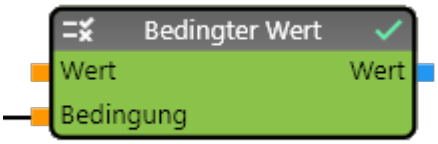
# Bedingung

Module der Gruppe "Bedingung" ermöglichen das bedingte Ausführen von Teilen des Logikplans.

| | |
|---|---|
| Bedingter Wert | Wert setzen wenn eine bestimmte Bedingung zutrifft |
| Bedingung | Bedingte Verzweigung |

# Bedingter Wert

Das Module "Bedingter Wert" setzt einen Wert nur dann wenn eine Bedingung am Eingang zu TRUE evaluiert wurde.




## DATENPUNKTE EINGANG

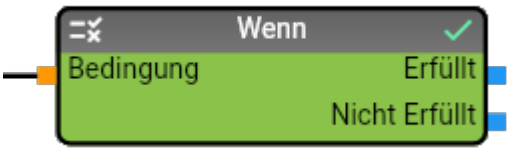
| NAME | BESCHREIBUNG |
|---|---|
| Wert | Wert der gesetzt werden soll. |
| Bedingung | Logischer Wert der als Bedingung überprüft wird. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Wert | Wert der gesetzt werden soll. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn die Bedingung am Eingang zu TRUE evaluiert wurde. |

# Bedingung

Das Module "Bedingung" (Wenn) bietet die Möglichkeit eine bedingte Verzweigung in den Logikplan zu integrieren. Ist die Bedingung am Eingang erfüllt, liefert der Ausgang "Erfüllt" den Wert "True", andernfalls liefert der Ausgang "Nicht erfüllt" den Wert "False".




## DATENPUNKTE EINGANG

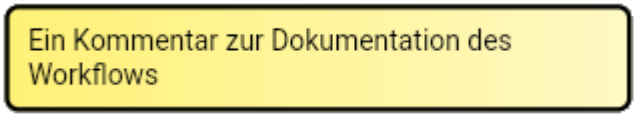
| NAME | BESCHREIBUNG |
|---|---|
| Bedingung | Logischer Wert der als Bedingung überprüft wird. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Erfüllt | Der nachfolgende Zweig wird nur ausgeführt wenn die Bedingung zu TRUE evaluiert wird. |
| Nicht erfüllt | Der nachfolgende Zweig wird nur ausgeführt wenn die Bedingung zu FALSE evaluiert wird. |

# Dokumentation: Kommentar

Das Module "Kommentar" bietet die Möglichkeit Text zur Kommentierung des Logikplans einzufügen.




## PROPERTIES

| NAME | BESCHREIBUNG |
| --- | --- |
| Text | Kommentar Text. |

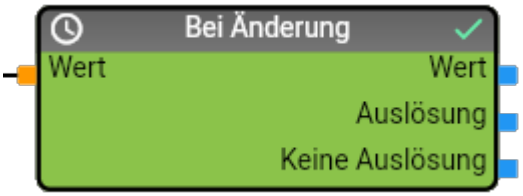
# Ereignisse

Module der Gruppe "Ereignisse" ermöglichen das automatische Ausführen des Logikplans wenn sich der Wert einer Variable ändert.

| | |
|---|---|
| Bei Änderung | Ausführung des Logikplans bei Veränderung einer Variable. |
| Bei Aktualisierung | Ausführung des Logikplans bei Aktualisierung einer Variable. |
| Bei bestimmten Wert | Ausführung des Logikplans bei erreichen eines bestimmten Variablen Wertes. |
| Bei Überschreitung | Ausführung des Logikplans bei Überschreitung eines Variablen Wertes. |
| Bei Unterschreitung | Ausführung des Logikplans bei Unterschreitung eines Variablen Wertes. |
| Wochenplan (1 Gruppe) | Wochenplan mit 1 Gruppe. |
| Wochenplan (2 Gruppen) | Wochenplan mit 2 Gruppen. |
| Wochenplan (7 Gruppen) | Wochenplan mit 7 Gruppen. |

# Bei Änderung

Das Module "Bei Änderung" ermöglicht das Ausführen des Logikplans wenn eine bestimmte Variable verändert (neuer Wert wird der Variable zugewiesen) wurde.




## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
|---|---|
| Wert | Datenpunkt der Variable deren Änderung eine Ausführung des Logikplans auslösen soll. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Wert | Datenpunkt der Variable. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn die zugehörige Variable verändert wurde. |
| Auslösung | Datenpunkt des Modules der TRUE liefert, wenn die verlinkte Variable verändert wurde. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn die Variable verändert wurde. |
| Keine Auslösung | Datenpunkt des Modules der TRUE liefert, wenn die verlinkte Variable nicht verändert wurde. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn die Variable nicht verändert wurde. |

## ANWENDUNGSBEISPIEL

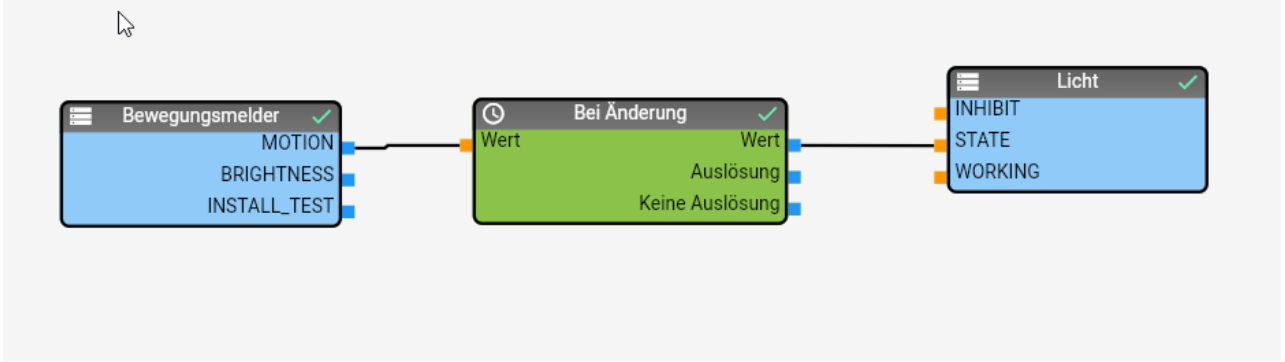
Gegeben sind 1 Homematic Bewegungsmelder und 1 Homematic Aktor zum Schalten eines Lichtes:

| | | |
|---|---|---|
| Bewegungsmelder | HomeMatic Device | |
| BRIGHTNESS | Integer | 160 |
| INSTALL_TEST | Boolean | true |
| MOTION | Boolean | Untätig |
| Licht | HomeMatic Device | |
| INHIBIT | Boolean | false |
| STATE | Boolean | Aus |
| WORKING | Boolean | false |

Ansicht der beiden Aktoren im WebFront



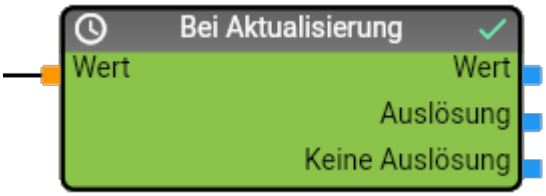

Es soll nun ein Logikplan im Logikpläne erstellt werden, der bei einem Auslösen des Bewegungsmelders automatisch ein Licht ein bzw. wieder ausschaltet.
Sie können dieses Verhalten durch die Verwendung eines Bei Änderung realisieren, vordefinierte Instanz Module für das Setzen der Werte finden Sie im Objektbaum des Logikplandesigners.




Im konkreten Fall wird der Logikplan bei jeder Änderung der Statusvariable des Bewegungsmelders aufgerufen und appliziert dessen Status auf das Licht weiter.

# Bei Aktualisierung

Das Module "Bei Aktualisierung" ermöglicht das Ausführen des Logikplans wenn eine bestimmte Variable aktualisiert (selber Wert oder anderer Wert wird der Variable zugewiesen) wurde.




## DATENPUNKTE EINGANG

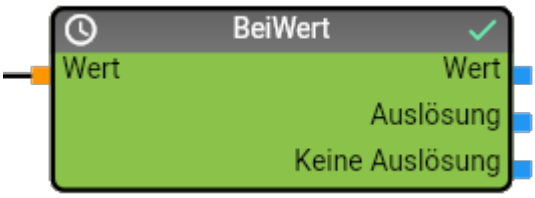
| NAME | BESCHREIBUNG |
|---|---|
| Wert | Datenpunkt der Variable deren Aktualisierung eine Ausführung des Logikplans auslösen soll. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Wert | Datenpunkt der Variable. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn die zugehörige Variable aktualisiert wurde. |
| Auslösung | Datenpunkt des Modules der TRUE liefert, wenn die verlinkte Variable aktualisiert wurde. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn die Variable aktualisiert wurde. |
| Keine Auslösung | Datenpunkt des Modules der TRUE liefert, wenn die verlinkte Variable nicht aktualisiert wurde. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn die Variable nicht aktualisiert wurde. |

# Bei bestimmten Wert

Das Module "Bei bestimmten Wert" ermöglicht das Ausführen des Logikplans wenn eine Variable einen bestimmten Wert erreicht.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Wert für Auslösung | Wert, bei dem das Module auslösen soll. |
| Nachfolgende Ereignisse auslösen | Bei Aktivierung dieser Option, löst das Modul bei jeder Aktualisierung der Variable aus, wenn der Wert dem spezifizierten Limit entspricht. Bei Deaktivierung der Option löst das Modul nur einmalig aus, erst bei überschreiten oder unterschreiten und nochmaligen erreichen des Limits löst das Modul wieder aus. |

## DATENPUNKTE EINGANG

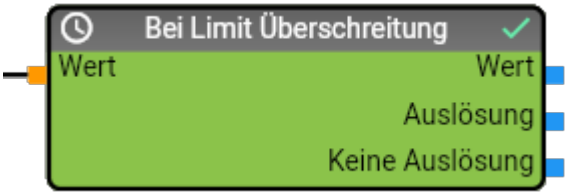
| NAME | BESCHREIBUNG |
|---|---|
| Wert | Datenpunkt der Variable deren Änderung eine Ausführung des Logikplans auslösen soll. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Wert | Datenpunkt der Variable. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn auch der Ausgang "Aktualisierung" TRUE liefert. |
| Auslösung | Datenpunkt des Modules der TRUE liefert, wenn der Wert der verlinkten Variable den spezifizierten Wert erreicht hat. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn der Wert erreicht wurde. |
| Keine Auslösung | Datenpunkt des Modules der TRUE liefert, wenn der Wert der verlinkten Variable den spezifizierten Wert nicht erreicht hat. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn der Wert nicht erreicht wurde. |

# Bei Überschreitung

Das Module "Bei Überschreitung" ermöglicht das Ausführen des Logikplans wenn eine Variable einen bestimmten Wert überschreitet.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Wert für Auslösung | Wert, bei dessen Überschreitung das Module auslösen soll. |
| Nachfolgende Ereignisse auslösen | Bei Aktivierung dieser Option, löst das Modul bei jeder Änderung der Variable aus, wenn der Wert über dem spezifizierten Limit liegt. Bei Deaktivierung der Option löst das Modul nur einmalig aus, erst bei unterschreiten und nochmaligen überschreiten des Limits löst das Modul wieder aus. |

## DATENPUNKTE EINGANG

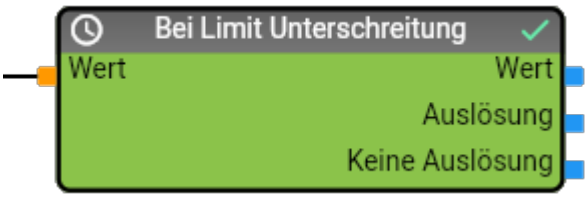
| NAME | BESCHREIBUNG |
|---|---|
| Wert | Datenpunkt der Variable deren Überschreitung eine Ausführung des Logikplans auslösen soll. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Wert | Datenpunkt der Variable. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn auch der Datenpunkt "Aktualisierung" TRUE liefert. |
| Auslösung | Datenpunkt des Modules der TRUE liefert, wenn der Wert der verlinkten Variable den spezifizierten Wert überschritten hat. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn der Wert überschritten wurde. |
| Keine Auslösung | Datenpunkt des Modules der TRUE liefert, wenn der Wert der verlinkten Variable den spezifizierten Wert nicht überschritten hat. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn der Wert nicht überschritten wurde. |

# Bei Unterschreitung

Das Module "Bei Limit Unterschreitung" ermöglicht das Ausführen des Logikplans wenn eine Variable einen bestimmten Wert unterschreitet.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Wert für Auslösung | Wert, bei dessen Unterschreitung das Module auslösen soll. |
| Nachfolgende Ereignisse auslösen | Bei Aktivierung dieser Option, löst das Modul bei jeder Änderung der Variable aus, wenn der Wert unter dem spezifizierten Limit liegt. Bei Deaktivierung der Option löst das Modul nur einmalig aus, erst bei überschreiten und nochmaligen unterschreiten des Limits löst das Modul wieder aus. |

## DATENPUNKTE EINGANG

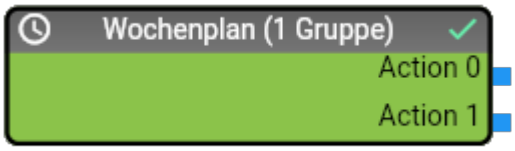
| NAME | BESCHREIBUNG |
|---|---|
| Wert | Datenpunkt der Variable deren Unterschreitung eine Ausführung des Logikplans auslösen soll. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Wert | Datenpunkt der Variable. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn auch der Ausgang "Aktualisierung" TRUE liefert. |
| Auslösung | Datenpunkt des Modules der TRUE liefert, wenn der Wert der verlinkten Variable den spezifizierten Wert unterschritten hat. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn der Wert unterschritten wurde. |
| Keine Auslösung | Datenpunkt des Modules der TRUE liefert, wenn der Wert der verlinkten Variable den spezifizierten Wert nicht unterschritten hat. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn der Wert nicht unterschritten wurde. |

# Wochenplan (1 Gruppe)

Das Module "Wochenplan (1 Gruppe)" bietet die Möglichkeit den Logikplan mit einem IP-Symcon Wochenplan mit 1 Gruppe zu verknüpfen.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Anzahl Aktionen | Anzahl der Aktionen, die für die Konfiguration des Wochenplans in der Visualisierung zur Verfügung stehen. |
| Aktionen | Liste der Wochenplan Aktionen. Jeder Wochenplan Aktion kann ein Name und eine Farbe zugeordnet werden. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Aktion x | Datenpunkt der jeweiligen Wochenplan Aktion, wenn die entsprechende Aktion auslöst, wird der Datenpunkt zu TRUE evaluiert wird. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn dieser Datenpunkt zu TRUE evaluiert wurde. |

# Wochenplan (2 Gruppen)

Das Module "Wochenplan (2 Gruppen)" bietet die Möglichkeit den Logikplan mit einem IP-Symcon Wochenplan mit 2 Gruppe zu verknüpfen.




## PROPERTIES

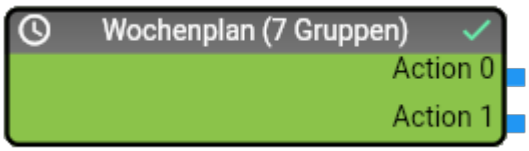
| NAME | BESCHREIBUNG |
|---|---|
| Anzahl Aktionen | Anzahl der Aktionen, die für die Konfiguration des Wochenplans in der Visualisierung zur Verfügung stehen. |
| Aktionen | Liste der Wochenplan Aktionen. Jeder Wochenplan Aktion kann ein Name und eine Farbe zugeordnet werden. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Aktion x | Datenpunkt der jeweiligen Wochenplan Aktion, wenn die entsprechende Aktion auslöst, wird der Datenpunkt zu TRUE evaluiert wird. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn dieser Datenpunkt zu TRUE evaluiert wurde. |

# Wochenplan (7 Gruppen)

Das Module "Wochenplan (7 Gruppen)" bietet die Möglichkeit den Logikplan mit einem IP-Symcon Wochenplan mit 7 Gruppen zu verknüpfen.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Anzahl Aktionen | Anzahl der Aktionen, die für die Konfiguration des Wochenplans in der Visualisierung zur Verfügung stehen. |
| Aktionen | Liste der Wochenplan Aktionen. Jeder Wochenplan Aktion kann ein Name und eine Farbe zugeordnet werden. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Aktion x | Datenpunkt der jeweiligen Wochenplan Aktion, wenn die entsprechende Aktion auslöst, wird der Datenpunkt zu TRUE evaluiert wird. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn dieser Datenpunkt zu TRUE evaluiert wurde. |

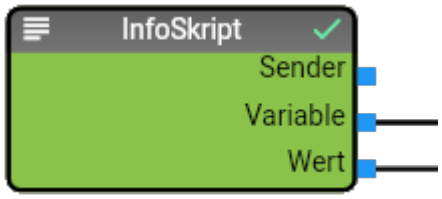
# Informationen

Module der Gruppe "Informationen" ermöglichen die Einbindung von diversen IP-Symcon Informationen in den Logikplan.

| | |
|---|---|
| Skript Informationen | Inforamtionen zur aktuellen Skriptausführung |
| System Informationen | Datum und Uhrzeit des IP-Symcon Systems |
| Variablen Inforamtionen | Informationen zu einer Variable |

# Skript Informationen

Das Module "SkriptInfo" bietet die Möglichkeit zum Einbinden von Informationen zur aktuellen Skriptausführung.

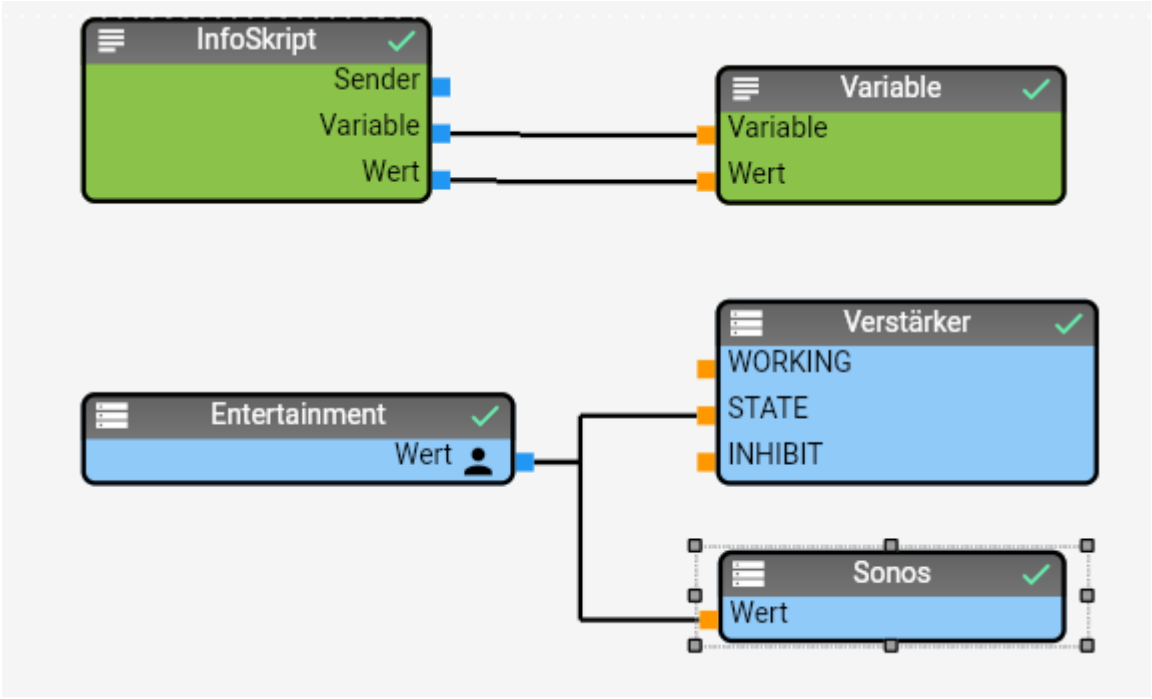



## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Sender | Die Sender Variable der aktuellen Skriptausführung (Wert der IP-Symcon Skript Variable $_IPS['SENDER']). |
| Variable | Die Variable ID der aktuellen Skriptausführung (Wert der IP-Symcon Skript Variable $_IPS['VARIABLE']). |
| Wert | Die Value Variable der aktuellen Skriptausführung (Wert der IP-Symcon Skript Variable $_IPS['VALUE']). |

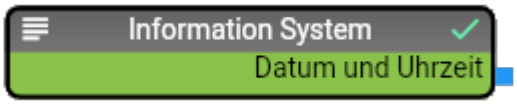
## ANWENDUNGSBEISPIEL

Ein typisches Anwendungsbeispiel für das Modul "SkriptInfo" ist die Realisierung eines Aktionsskriptes für eine Variable.
Dabei nimmt der Logikplan den Wert der Visualisierung entgegen und weist in der betreffenden Variable zu.
Weitergehende Informationen zum Thema Aktionsskript.
Die Zuweisung des Logikplans als Aktionsskript der Variable erfolgt dabei indem man die Variable in den Logikplan einbindet und im Eigenschaft-Editor die Option "Als Aktionsskript" auswählt.
Alternativ kann das Logikplan Skript auch manuell der Variable in der Console zugewiesen werden.
Im vorliegenden Beispiel wird zuerst der Wert der Variable "Entertainment" gesetzt (Bedienung über das WebFront oder IPSView Client) und danach der Wert verwendet um einen Verstärker und einen Sonos Player mittels Automation Skript anzusteuern.


InfoSkript
Sender
Variable
Wert
Variable
Variable
Wert
Verstärker
WORKING
STATE
INHIBIT
Entertainment
Wert
Sonos
Wert

# System Informationen

Das Module "SystemInfo" liefert Informationen zum IP-Symcon System.

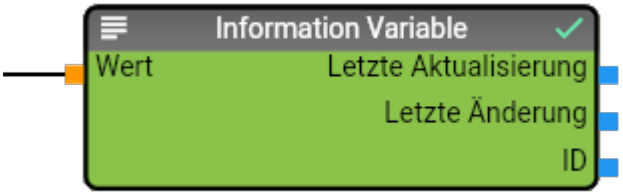



## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
| --- | --- |
| Zeitstempel | Unix Timestamp der aktuellen Systemzeit. |

# Variablen Inforamtionen

Das Module "VariableInfo" liefert Informationen zur einer Variable.




## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
|---|---|
| Wert | Verbindung zum Datenpunkt Wert einer Variable, von der die Informationen angezeigt werden sollen. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Letzte Aktualisierung | Unix Timestamp der letzten Aktualisierung der Variable. |
| Letzte Änderung | Unix Timestamp der letzten Änderung der Variable. |
| ID | ID der Variable. |

# Konstante Werte

Das Module "Konstante" bietet die Möglichkeit zur Einbindung von fixen Werten in den Logikplan, dem Modul kann in den Properties ein Datentyp, ein Wert und ein Name für die Anzeige zugewiesen werden.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Anzeige | Name der im Diagramm für die Konstante angezeigt wird. |
| Wert | Wert der Konstante als String. |
| Datentype | Datentype der Konstante. |

## DATENPUNKTE AUSGANG

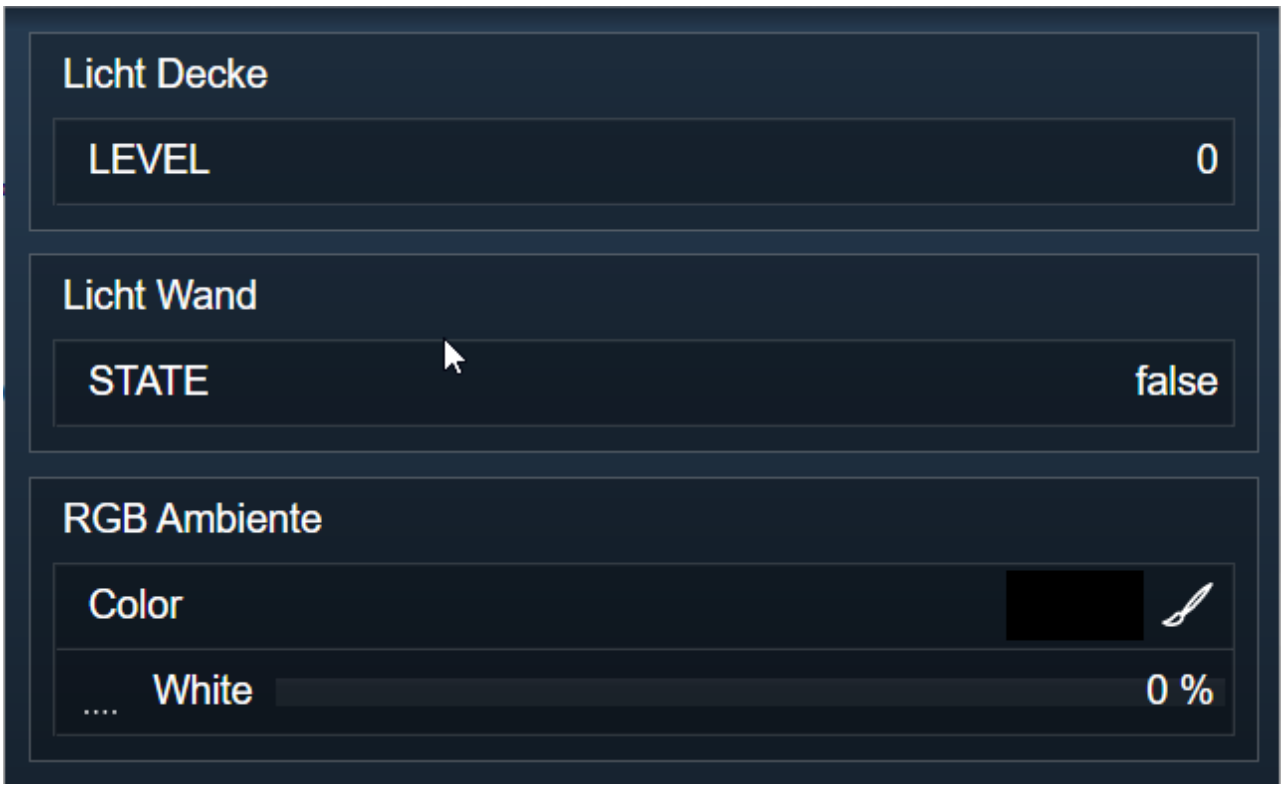
| NAME | BESCHREIBUNG |
|---|---|
| Wert | Der Wert der Konstante |

## ANWENDUNGSBEISPIEL

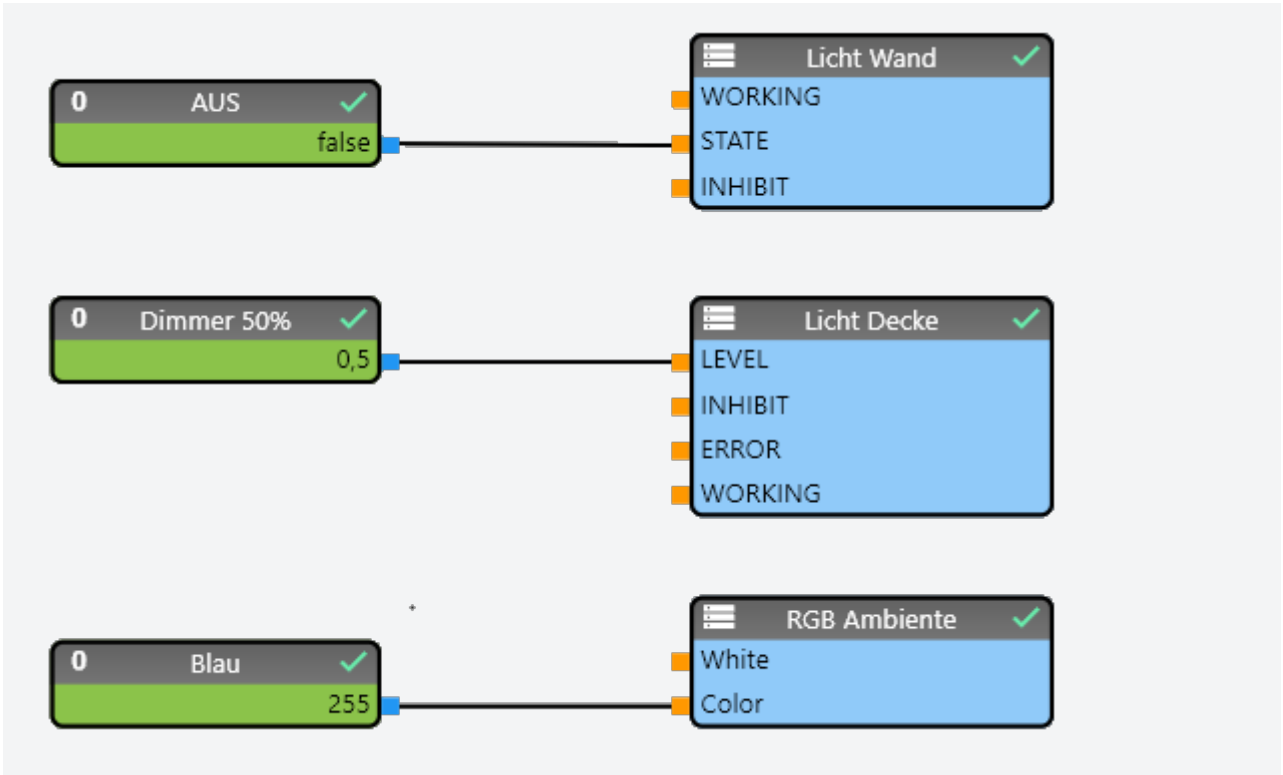
Gegeben sind 1 Homematic Aktor zum Schalten eines Lichtes, 1 Homematic Aktor zum Dimmen eines Lichtes und ein IP-Symcon 868 Aktor zum Betrieb eines RGB Strips:

| | | |
|---|---|---|
| Licht Decke | HomeMatic Device | |
| ERROR | Integer | 0 |
| INHIBIT | Boolean | false |
| LEVEL | Float | 0 |
| WORKING | Boolean | false |
| Licht Wand | HomeMatic Device | |
| INHIBIT | Boolean | false |
| STATE | Boolean | false |
| WORKING | Boolean | false |
| RGB Ambiente | IPS-868 Stripe | |
| Color | Integer | |
| White | Integer | 0 % |

Die Aktoren können über das WebFront getrennt voneinander geschaltet werden:




Es soll nun ein Logikplan im Logikpläne erstellt werden, der ein bestimmtes Lichtszenario einstellt und dabei den einzelnen Beleuchtungskörpern vordefinierte Werte zuweist.
Sie können dieses Werte durch die Verwendung von Konstanten realisieren, vordefinierte Instanz Module für das Setzen der Werte finden Sie im Objektbaum des Logikplandesigners.




Im konkreten Fall wird der RGB Leuchte die Farbe Blau zugewiesen, das Wandlicht wird abgeschaltet und der Dimmer auf 50% gesetzt.

# Logik

Module der Gruppe "Logik" ermöglichen das Einbinden von logischen Modulen.

| | |
|---|---|
| Nicht | Logisches Nicht |
| Oder | Logisches Oder |
| Und | Logisches Und |

# Nicht

Das Module "Nicht" bietet die Möglichkeit zur Realisierung eines logischen NICHT im Logikplan.




## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
|---|---|
| Wert | Datenpunkt für Eingangswert. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Invertiert | Liefert TRUE wenn der Wert am Eingang Wert FALSE ist und liefert FALSE wenn der Wert am Eingang Wert TRUE ist. |

# Oder

Das Module "Oder" bietet die Möglichkeit zur Realisierung einer logischen ODER Verknüpfung im Logikplan.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Anzahl Eingänge | Stellt 2 bis 8 Eingänge des Modules zu Verfügung. |

## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
|---|---|
| Wert 1 | Datenpunkt für Vergleichswert 1. |
| Wert 2 | Datenpunkt für Vergleichswert 2. |

## DATENPUNKTE AUSGANG

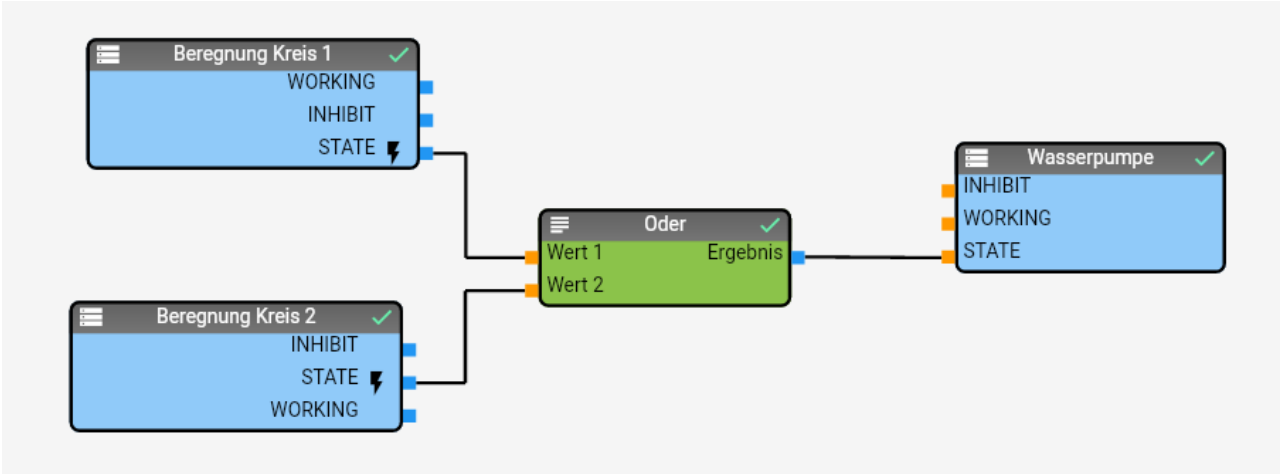
| NAME | BESCHREIBUNG |
|---|---|
| Ergebnis | Liefert TRUE wenn der Wert eines Einganges TRUE ist. |

## ANWENDUNGSBEISPIEL

Gegeben sind 2 Homematic Aktoren zur Steuerung einer Gartenbewässerung und 1 Homematic Aktor, der als Hauptschalter für die Bewässerung agiert:

| | | |
|---|---|---|
| Beregnung Kreis 1 | HomeMatic Device | |
| INHIBIT | Boolean | false |
| STATE | Boolean | true |
| WORKING | Boolean | false |
| Beregnung Kreis 2 | HomeMatic Device | |
| INHIBIT | Boolean | false |
| STATE | Boolean | false |
| WORKING | Boolean | false |
| Wasserpumpe | HomeMatic Device | |
| INHIBIT | Boolean | false |
| STATE | Boolean | true |
| WORKING | Boolean | false |

Es soll nun ein Logikplan im Logikpläne erstellt werden, der eine automatische Ansteuerung der Gartenpumpe realisiert. Wenn einer der beiden Bewässerungsaktoren aktiviert wird, soll der Hauptschalter ebenfalls aktiviert werden. Sind beide Aktoren aus, soll auch der Hauptschalter deaktiviert werden.
Sie können dieses Verhalten durch ein Oder Modul realisieren, vordefinierte Module für die Instanzen finden Sie im Objektbaum des Logikplan-Editors.



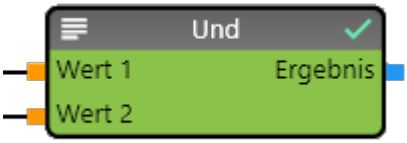

Funktionsweise:
Eine Änderung an den "STATE" Ausgängen der Beregnungskreis rufen den Logikplan automatisch auf, sobald einer der beiden Bewässerungsaktoren geschaltet wird (egal ob über eine Visualisierung, über einen Wochenplan oder ein anderes Skript).
Wurde einer der beiden Aktoren aktiviert, liefert das Oder TRUE und aktiviert den Hauptschalter .
Sind beide Aktoren deaktiviert, liefert das Oder FALSE und deaktiviert den Hauptschalter wieder.

Für eine automatische Ausführung des Logikplans bei einer Änderung der Variable müssen Sie die Eigenschaft "Logikplan Ausführung" des Moduls "Instanz Lesen" auf den Wert "Bei Änderung" setzen.



# Und

Das Module "Und" bietet die Möglichkeit zur Realisierung einer logischen UND Verknüpfung im Logikplan.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Anzahl Eingänge | Stellt 2 bis 8 Eingänge zu Verfügung |

## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
|---|---|
| Wert 1 | Datenpunkt für Vergleichswert 1. |
| Wert 2 | Datenpunkt für Vergleichswert 2. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Ergebnis | Liefert *TRUE* wenn der Wert an allen Eingängen *TRUE* ist. |

# Mapping

Das Module "Mappingtabelle" bietet die Möglichkeit bis zu 8 Eingänge mit einer komplexen Mapping Logik auf bis zu 8 Ausgänge zu mappen. Sie können für verschiedenste Kombinationen von Werten am Eingänge jeweils die Werte für die jeweiligen Ausgänge definieren.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Eingänge | Definition der Eingänge. Es können bis zu 8 Eingänge definiert werden, jedem Eingang muss ein Name und ein Datentyp zugewiesen werden. |
| Ausgänge | Definition der Ausgänge. Es können bis zu 8 Ausgänge definiert werden, jedem Ausgang muss ein Name und ein Datentyp zugewiesen werden. |
| Mappings | Definition der Mappings. |

## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
|---|---|
| Eingang x | Datenpunkte die in der Eigenschaft Eingänge definiert wurden |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Ausgang x | Datenpunkte die in der Eigenschaft Ausgänge definiert wurden |

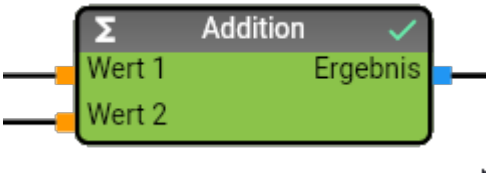
# Mathematik

Module der Gruppe "Mathematik" ermöglichen die Berechnung mathematischer Ausdrücke im Logikplan.

| | |
|---|---|
| Addition | Addition zweier Werte |
| Division | Division zweier Werte |
| Formel | Berechnung eines Wertes basierend einer benutzerdefinierten Formel. |
| Maximum | Maximum mehrerer Werte |
| Minimum | Minimum mehrerer Werte |
| Modulo | Berechnung des Modulo zweier Werte |
| Multiplikation | Multiplikation zweier Werte |
| Runden | Runden eines Wertes |
| Subtraktion | Subtraktion zweier Werte |

# Addition

Das Module "Addition" bietet die Möglichkeit zum Addieren von 2 Werten.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Anzahl Eingänge | Stellt 2 bis 8 Eingänge des Modules zu Verfügung. |

## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
|---|---|
| Wert 1 | Datenpunkt für Wert 1. |
| Wert 2 | Datenpunkt für Wert 2. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Ergebnis | Summe der Eingänge. |

# Division

Das Module "Division" bietet die Möglichkeit zur Division von 2 Werten.




## DATENPUNKTE EINGANG

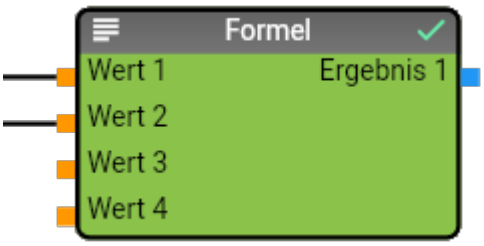
| NAME | BESCHREIBUNG |
|---|---|
| Wert 1 | Datenpunkt für Wert 1. |
| Wert 2 | Datenpunkt für Wert 2. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Ergebnis | Ergebnis aus der Berechnung Wert 1 durch Wert 2. |

# Formel

Das Module "Formel" bietet die Möglichkeit zur Berechnung von bis zu 8 Werten auf Basis einer benutzerdefinierten Formel mit bis zu 8 verschiedenen Eingangswerten.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Formel | PHP Code der beim Aufruf des Moduls ausgeführt werden soll. Die Werte des Eingangs stehen über die Variablen $value1, $value2, $value3, $value4, $value5, $value6, $value7 und $value8 zur Verfügung. Das Ergebnis kann über die Variable $result1, $result2, $result3, $result4, $result5, $result6, $result7 und $result8 an den Logikplan zurückgegeben werden. |
| Anzahl Ausgänge | Anzahl der Datenpunkte am Ausgang (Wertebereich 1-8) |
| Anzahl Eingänge | Anzahl der Datenpunkte am Eingang (Wertebereich 1-8) |

## DATENPUNKTE EINGANG

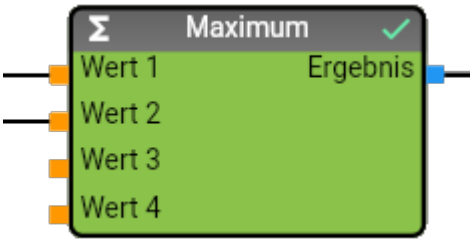
| NAME | BESCHREIBUNG |
|---|---|
| Wert 1 | Datenpunkt für Wert 1. |
| Wert 2 | Datenpunkt für Wert 2. |
| Wert 3 | Datenpunkt für Wert 3. |
| Wert 4 | Datenpunkt für Wert 4. |
| Wert 5 | Datenpunkt für Wert 5. |
| Wert 6 | Datenpunkt für Wert 6. |
| Wert 7 | Datenpunkt für Wert 7. |
| Wert 8 | Datenpunkt für Wert 8. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
| --- | --- |
| Ergebnis 1 | Ergebnis 1 aus der Berechnung mit der benutzerdefinierten Formel. |
| Ergebnis 2 | Ergebnis 2 aus der Berechnung mit der benutzerdefinierten Formel. |
| Ergebnis 3 | Ergebnis 3 aus der Berechnung mit der benutzerdefinierten Formel. |
| Ergebnis 4 | Ergebnis 4 aus der Berechnung mit der benutzerdefinierten Formel. |
| Ergebnis 5 | Ergebnis 5 aus der Berechnung mit der benutzerdefinierten Formel. |
| Ergebnis 6 | Ergebnis 6 aus der Berechnung mit der benutzerdefinierten Formel. |
| Ergebnis 7 | Ergebnis 7 aus der Berechnung mit der benutzerdefinierten Formel. |
| Ergebnis 8 | Ergebnis 8 aus der Berechnung mit der benutzerdefinierten Formel. |

# Maximum

Das Module "Maximum" bietet die Möglichkeit zur Berechnung des Maximum aus bis zu 4 verschiedenen Werten.




## DATENPUNKTE EINGANG

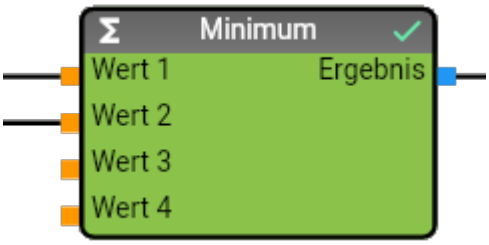
| NAME | BESCHREIBUNG |
|---|---|
| Wert 1 | Datenpunkt für Wert 1. |
| Wert 2 | Datenpunkt für Wert 2. |
| Wert 3 | Datenpunkt für Wert 3. |
| Wert 4 | Datenpunkt für Wert 4. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Ergebnis | Maximum der Eingangswerte Wert 1 bis Wert 4. |

# Minimum

Das Module "Minimum" bietet die Möglichkeit zur Berechnung des Minimum aus bis zu 4 verschiedenen Werten.




## DATENPUNKTE EINGANG

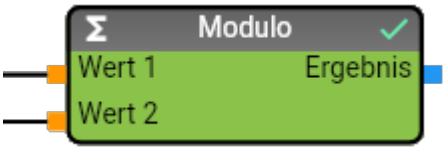
| NAME | BESCHREIBUNG |
|---|---|
| Wert 1 | Datenpunkt für Wert 1. |
| Wert 2 | Datenpunkt für Wert 2. |
| Wert 3 | Datenpunkt für Wert 3. |
| Wert 4 | Datenpunkt für Wert 4. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Ergebnis | Minimum der Eingangswerte Wert 1 bis Wert 4. |

# Modulo

Das Module "Modulo" bietet die Möglichkeit zur Berechnung des Modulo von 2 Werten.




## DATENPUNKTE EINGANG

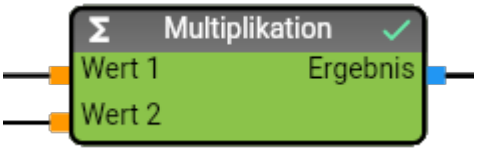
| NAME | BESCHREIBUNG |
|---|---|
| Wert 1 | Datenpunkt für Wert 1. |
| Wert 2 | Datenpunkt für Wert 2. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Ergebnis | Ergebnis aus der Berechnung Wert 1 modulo Wert 2. |

# Multiplikation

Das Module "Multiplikation" bietet die Möglichkeit zum multiplizieren von 2 Werten.




## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
|---|---|
| Wert 1 | Datenpunkt für Wert 1. |
| Wert 2 | Datenpunkt für Wert 2. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Ergebnis | Ergebnis aus der Berechnung Wert 1 mal Wert 2. |

# Runden

Das Module "Runden" bietet die Möglichkeit zum Runden eines Wertes.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Stellen | Anzahl der Stellen auf die gerundet werden soll. |

## DATENPUNKTE EINGANG

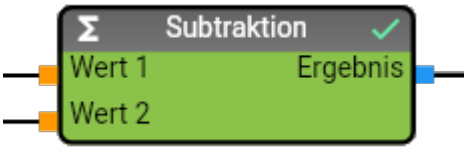
| NAME | BESCHREIBUNG |
|---|---|
| Wert 1 | Datenpunkt für Eingangswert. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Ergebnis | Ergebnis aus der Berechnung Wert gerundet. |

# Subtraktion

Das Module "Subtraktion" bietet die Möglichkeit zum Subtrahieren von 2 Werten.




## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
|---|---|
| Wert 1 | Datenpunkt für Wert 1. |
| Wert 2 | Datenpunkt für Wert 2. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Ergebnis | Ergebnis aus der Berechnung Wert 1 minus Wert 2. |

# Sublogikplan

"Sublogikplan" ermöglichen es, mehrere Module in eine neue Einheit zu Gruppieren und mit Parametern und Datenpunkten zu versehen. Diese Gruppe an Modulen ist dann im "Hauptlogikplan" als eigenständiges Modul anzusprechen, ferner besteht auch die Möglichkeit diese Sublogikplans in ein File zu exportieren bzw. zu importieren.

| | |
|---|---|
| Sublogikplan Parameter | Sublogikplan Parameter erstellen. |
| Sublogikplan Ausgang | Sublogikplan Ausgang erstellen. |
| Sublogikplan Eingang | Sublogikplan Eingang erstellen. |
| Sublogikplan Modul | Sublogikplan erstellen. |

# Sublogikplan Parameter

Das Module "Parameter" bietet die Möglichkeit einen Parameter für einen Sublogikplan zu erstellen. Parameter können in weiterer Folge in den Eigenschaften des Sublogikplan Modules gesetzt werden.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Anzeige | Name des Parameters, spezifiziert den Namen des Parameters wie dieser in der Eigenschaftsliste angezeigt werden soll. |
| Datentype | Datentype des Parameters |
| Defaultwert | Default Wert des Parameters, der nach dem initialen Erstellen des Modules zugewiesen werden soll. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Wert | Aktueller Wert des Parameters der in weiterer Folge im Sublogikplan verwendet werden kann. |

# Sublogikplan Ausgang

Das Module "Ausgang" bietet die Möglichkeit einen Datenpunkt Ausgang für einen Sublogikplan zu erstellen. Ausgänge werden als Datenpunkte des Sublogikplan Modules zur Verfügung gestellt und können dort mit anderen Modulen verbunden werden.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Anzeige | Name des Ausgangs, spezifiziert den Namen des Datenpunktes wie dieser im Hauptlogikplan Modul angezeigt werden soll. |
| Datentype | Datentype des Parameters |

## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
|---|---|
| Wert | Wert der dem Hauptlogikplan über den Ausgang zur Verfügung gestellt werden soll. |

# Sublogikplan Eingang

Das Module "Eingang" bietet die Möglichkeit einen Datenpunkt Eingang für einen Sublogikplan zu erstellen. Eingänge werden als Datenpunkte des SubLogikplan Modules zur Verfügung gestellt und können dort mit anderen Modulen verbunden werden.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Anzeige | Name des Eingangs, spezifiziert den Namen des Datenpunktes wie dieser im Hauptlogikplan Modul angezeigt werden soll. |
| Datentype | Datentype des Parameters |
| Pflicht | Spezifiziert ob der Eingang zwingend verbunden werden muss. |

## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
|---|---|
| Default Wert | Default Wert wenn der Datenpunkt nicht verbunden wurde. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Wert | Aktueller Wert der an den Datenpunkt des Sublogikplan Modules übergeben wurde. |

# Sublogikplan Modul

Das Module "Sublogikplan" bietet die Möglichkeit mehrere Module zu einer Einheit zusammenzufassen und in Ihren Hauptlogikplan einzufügen.




Die folgenden Funktionen stehen zur Verfügung:

| NAME | BESCHREIBUNG |
|---|---|
| Öffnen / Bearbeiten | Klicken Sie auf Öffnen / Bearbeiten, um einen Sublogikplan zur Bearbeitung zu öffnen. Nach Betätigung des Buttons wird der Sublogikplan geöffnet und kann verändert werden. |
| Zurück | Klicken Sie auf Zurück, um den Sublogikplan Editor wieder zu verlassen. Nach Betätigung des Buttons werden die aktuellen Änderungen am Sublogikplan gespeichert und der Hauptlogikplan wieder geladen. |

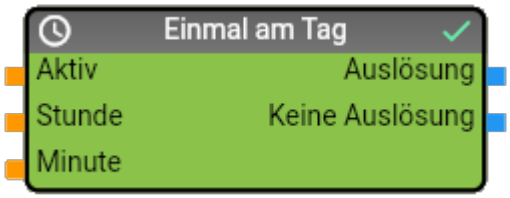
# Timer

Module der Gruppe "Timer" das automatische Ausführen des Logikplans über einen Timer.

| | |
|---|---|
| Einmal am Tag | Zyklischer Timer mit täglicher Auslösung. |
| Einmaliger Timer | Einmalige Ausführung eines Timers. |
| Timer Variable | Timer mit Variable zur Anzeige der Restlaufzeit. |
| Timer Zyklisch Minute | Zyklischer Timer mit Auslösung im Minuten Intervall. |
| Timer Zyklisch Sekunde | Zyklischer Timer mit Auslösung im Sekunden Intervall. |
| Timer Zyklisch Stunde | Zyklischer Timer mit Auslösung im Stunden Intervall. |

# Einmal am Tag

Das Module "Einmal am Tag" ermöglicht die Realisierung eines Timers, der jeden Tag um eine bestimmte Uhrzeit den Logikplan aufruft.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Uhrzeit | Uhrzeit für die Auslösung des Timers. |

## DATENPUNKTE EINGANG

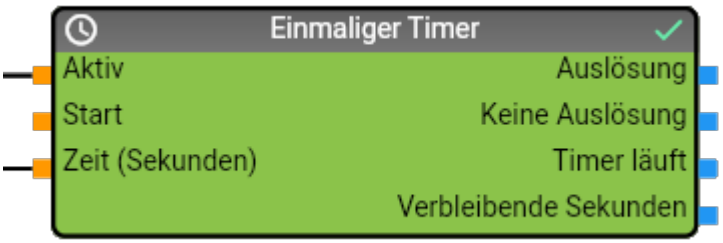
| NAME | BESCHREIBUNG |
|---|---|
| Aktiv | Aktivierung des Timers, bei TRUE wird der Timer aktiviert und löst den Logikplan zu der eingestellten Zeit täglich aus. |
| Stunde | Datenpunkt des Modules der die Stunde der Auslösung steuert. Wenn der Eingang nicht verbunden ist, wird der Wert der Eigenschaft "Uhrzeit" für die Auslösung verwendet. |
| Minute | Datenpunkt des Modules der die Minute der Auslösung steuert. Wenn der Eingang nicht verbunden ist, wird der Wert der Eigenschaft "Uhrzeit" für die Auslösung verwendet. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Auslösung | Datenpunkt des Modules der TRUE liefert, wenn das Modul ausgelöst hat. Dies erfolgt bei aktiven Timer jede Sekunde. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn der Timer ausgelöst hat. |
| Keine Auslösung | Datenpunkt des Modules der TRUE liefert, wenn das Modul nicht ausgelöst hat. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn der Timer nicht ausgelöst hat. |

# Einmaliger Timer

Das Module "Einmaliger Timer" ermöglicht das zeitversetzte Ausführen von bestimmten Teilen des Logikplans.




## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
|---|---|
| Neustart/ Aktiv | Aktivierung des Timers, bei TRUE wird der Timer aktiviert und löst den Logikplan nach einer bestimmten Zeit wieder aus. Sollte der Timer bereits laufen, wird die verbleibende Zeit auf den ursprünglichen Wert zurückgesetzt. |
| Start | Starten des Timers, bei TRUE wird der Timer aktiviert wenn dieser noch nicht aktiv ist und löst den Logikplan nach einer bestimmten Zeit wieder aus. |
| Zeit (Sekunden) | Zeit in Sekunden, nach der Timer auslösen soll. |

## DATENPUNKTE AUSGANG

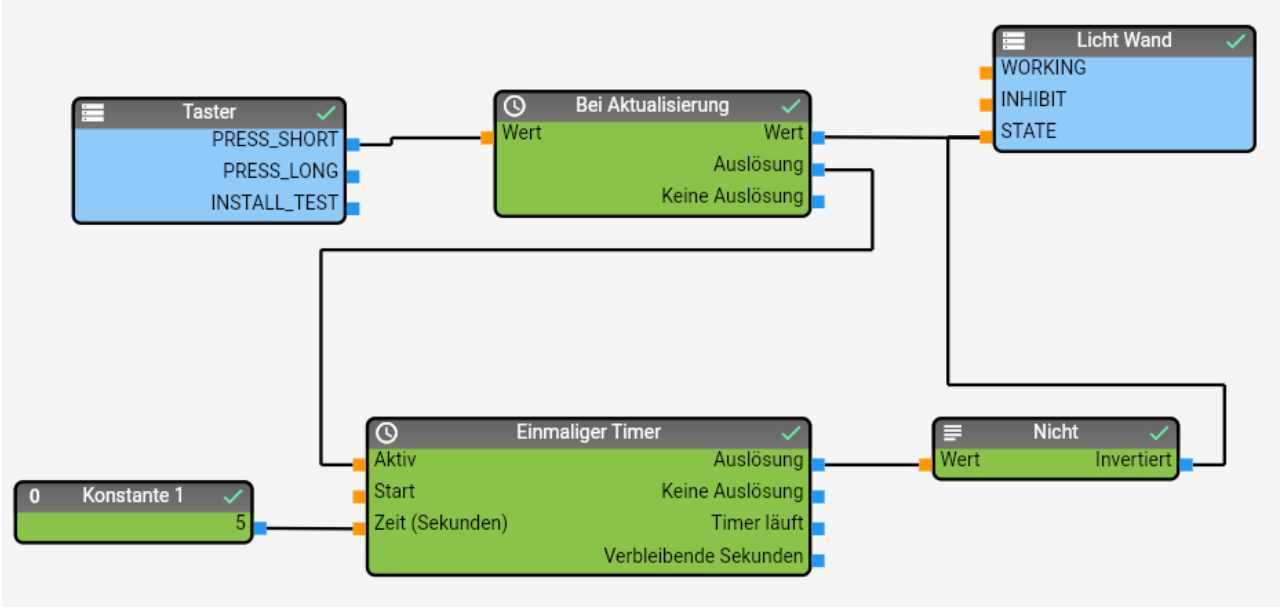
| NAME | BESCHREIBUNG |
|---|---|
| Auslösung | Datenpunkt des Modules der TRUE liefert, wenn das Modul ausgelöst hat. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn der Timer ausgelöst hat. |
| Keine Auslösung | Datenpunkt des Modules der TRUE liefert, wenn das Modul nicht ausgelöst hat. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn der Timer nicht ausgelöst hat. |
| Timer läuft | Datenpunkt des Modules der TRUE liefert, wenn der Timer gestartet wurde, aber noch nicht ausgelöst hat. |
| Verbleibende Sekunden | Liefert die Anzahl von Sekunden die bis zum Auslösen des Timers noch vergehen müssen. |

## ANWENDUNGSBEISPIEL

Gegeben sind 1 Homematic Taster und 1 Homematic Aktor zum Schalten eines Lichtes:

| | | |
|---|---|---|
| Licht Wand | HomeMatic Device | |
| INHIBIT | Boolean | false |
| STATE | Boolean | false |
| WORKING | Boolean | false |
| Taster | HomeMatic Device | |
| INSTALL_TEST | Boolean | false |
| PRESS_LONG | Boolean | true |
| PRESS_SHORT | Boolean | true |

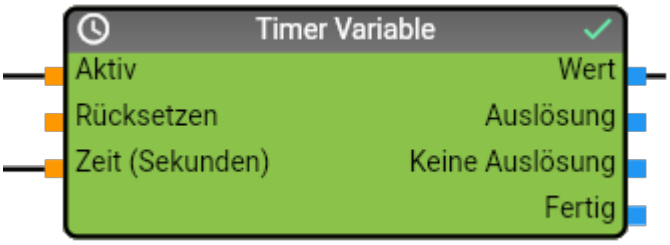
Es soll nun ein Logikplan im Logikpläne erstellt werden, der bei einem Auslösen des Tasters ein Licht einschaltet und nach einer Minute wieder abschaltet.
Sie können diese Anforderung durch die Verwendung eines Modules Einmaliger Timer realisieren, vordefinierte Instanz Module für den Taster bzw. das Aktivieren des Lichtes finden Sie im Objektbaum des Logikplandesigners.




Funktionsweise:
Die Variable PRESS_SHORT eines Homematic Tasters hat immer den Wert TRUE, bei Betätigung des Tasters wird der Wert der Variable aktualisiert und lediglich Datum/Uhrzeit der Variable werden aktualisiert. Diese Aktualisierung kann man mit dem Modul Bei Aktualisierung auswerten und andere Module damit steuern.
Im aktuellen Beispiel wird bei der Aktualisierung der Variable PRESS_SHORT das Licht eingeschaltet und gleichzeitig durch das Setzen des "Aktiv" Eingangs im Modul Einmaliger Timer ein Timer mit 60 Sekunden ausgelöst.
Das LogikplanSkript wird durch diesen Timer nach 60 Sekunden ein weiteres mal ausgeführt und schaltet das Licht über das Nicht Modul wieder aus.

# Timer Variable

Das Module "Timer Variable" ermöglicht die Realisierung eines Timers basierend auf einer Variable. Nach dem Aktivieren des Timers wird die Verbleibende Zeit bis zum Auslösen des Timers heruntergezählt.




## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
|---|---|
| Aktiv | Aktivierung des Timers, bei TRUE wird der Timer aktiviert und löst den Logikplan nach einer bestimmten Zeit wieder aus. |
| Rücksetzen | Deaktivierung des Timers, bei TRUE kann ein aktiver Timer abgebrochen werden. |
| Zeit (Sekunden) | Zeit in Sekunden, nach der Timer auslösen soll. |

## DATENPUNKTE AUSGANG

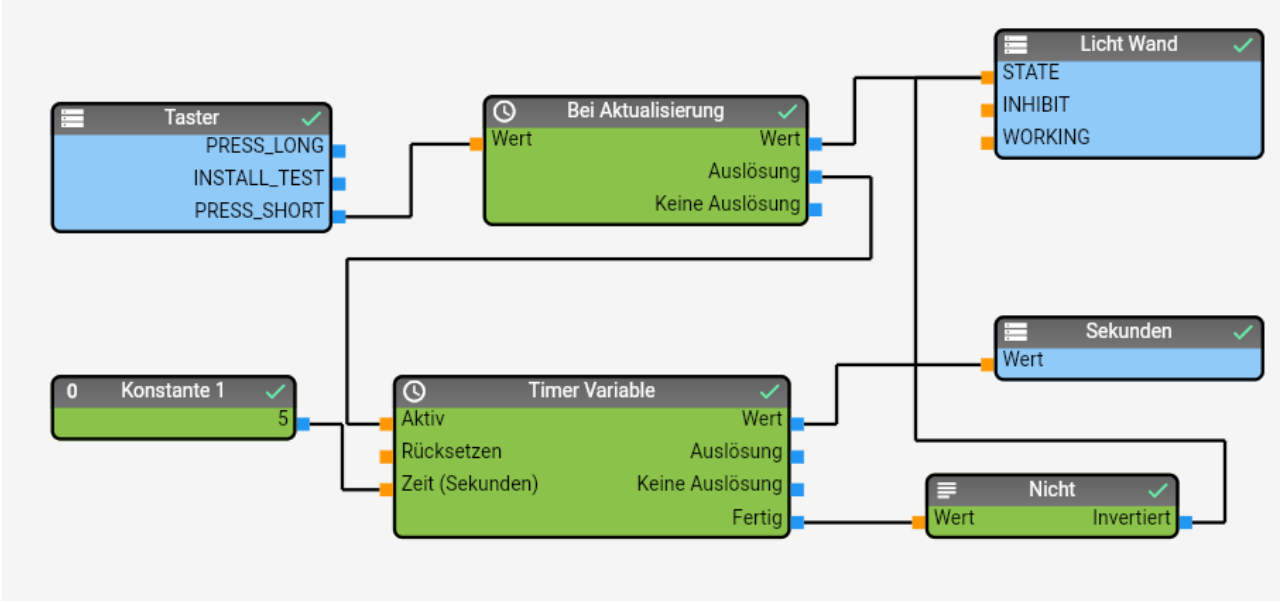
| NAME | BESCHREIBUNG |
|---|---|
| Wert | Dieser Datenpunkt des Modules muss zu einer Variable vom Type Integer verbunden werden. Die Variable realisiert die Anzeige der verbleibenden Timer Zeit. |
| Auslösung | Datenpunkt des Modules der TRUE liefert, wenn das Modul ausgelöst hat. Dies erfolgt bei aktiven Timer jede Sekunde. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn der Timer ausgelöst hat. |
| Keine Auslösung | Datenpunkt des Modules der TRUE liefert, wenn das Modul nicht ausgelöst hat. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn der Timer nicht ausgelöst hat. |
| Fertig | Datenpunkt des Modules der TRUE liefert, wenn der Timer beendet wurde. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn der Timer beendet wurde. |

## ANWENDUNGSBEISPIEL

Gegeben sind 1 Homematic Taster, 1 Homematic Aktor zum Schalten eines Lichtes und eine Variable für die Anzeige der Restlaufzeit eines Treppenhausautomaten (normale Variable vom Typ Integer, die ein Profil mit Suffix "Sec" zugewiesen hat).

| | | |
|---|---|---|
| Licht Wand | HomeMatic Device | |
| INHIBIT | Boolean | false |
| STATE | Boolean | false |
| WORKING | Boolean | false |
| Taster | HomeMatic Device | |
| INSTALL_TEST | Boolean | false |
| PRESS_LONG | Boolean | true |
| PRESS_SHORT | Boolean | true |
| Sekunden | Integer | 0 |

Es soll nun ein Logikplan im Logikpläne erstellt werden, der bei einem Auslösen des Tasters ein Licht einschaltet und nach einer Minute wieder abschaltet. Zusätzlich soll auch noch die Restlaufzeit in Sekunden visualisiert werden
Sie können diese Anforderung durch die Verwendung eines Modules Timer Variable realisieren, vordefinierte Instanz Module für den Taster bzw. das Aktivieren des Lichtes finden Sie im Objektbaum des Logikplandesigners.



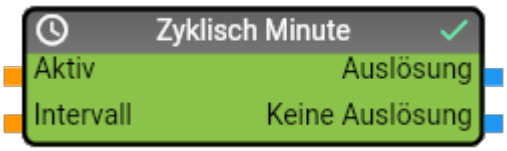

Funktionsweise:

Die Variable PRESS_SHORT eines Homematic Tasters hat immer den Wert TRUE, bei Betätigung des Tasters wird der Wert der Variable aktualisiert und lediglich Datum/Uhrzeit der Variable werden aktualisiert. Diese Aktualisierung kann man mit dem Modul Bei Aktualisierung auswerten und andere Module damit steuern.
Im aktuellen Beispiel wird bei der Aktualisierung der Variable PRESS_SHORT das Licht eingeschaltet und gleichzeitig durch das Setzen des "Aktiv" Eingangs im Modul Timer Variable ein Timer mit 60 Sekunden ausgelöst, der die Variable "Verbleibende Zeit" zum Herunterzählen der Sekunden verwendet.
Das LogikplanSkript wird durch diesen Timer jede Sekunde ausgeführt und aktualisiert den Wert in der Variable "Verbleibende Zeit".

Hat der Timer den Wert 0 erreicht, wird die Lampe über den Ausgang "Fertig" und einem Nicht Modul wieder ausgeschaltet.

# Timer Zyklisch Minute

Das Module "Timer Zyklisch Minuten" ermöglicht die Realisierung eines Timers, der den Logikplan zyklisch mit dem eingestellten Intervall aufruft.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Intervall | Intervall in Minuten für die Auslösung des Timers. |

## DATENPUNKTE EINGANG

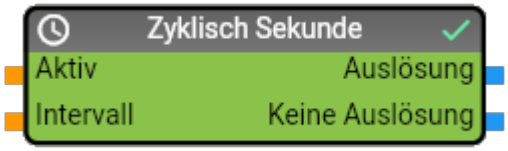
| NAME | BESCHREIBUNG |
|---|---|
| Aktiv | Aktivierung des Timers, bei TRUE wird der Timer aktiviert und löst den Logikplan zum vorgegebenen Intervall minütlich aus. |
| Intervall | Datenpunkt des Modules für das Steuern des Intervalls. Wenn der Eingang nicht verbunden ist, wird der Wert der Eigenschaft "Intervall" für die Auslösung verwendet. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Auslösung | Datenpunkt des Modules der TRUE liefert, wenn das Modul ausgelöst hat. Dies erfolgt bei aktiven Timer jede Sekunde. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn der Timer ausgelöst hat. |
| Keine Auslösung | Datenpunkt des Modules der TRUE liefert, wenn das Modul nicht ausgelöst hat. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn der Timer nicht ausgelöst hat. |

# Timer Zyklisch Sekunde

Das Module "Timer Zyklisch Sekunde" ermöglicht die Realisierung eines Timers, der den Logikplan zyklisch mit dem eingestellten Intervall aufruft.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Intervall | Intervall in Sekunden für die Auslösung des Timers. |

## DATENPUNKTE EINGANG

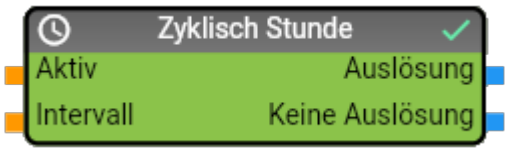
| NAME | BESCHREIBUNG |
|---|---|
| Aktiv | Aktivierung des Timers, bei TRUE wird der Timer aktiviert und löst den Logikplan zum vorgegebenen Intervall sekündlich aus. |
| Intervall | Datenpunkt des Modules für das Steuern des Intervalls. Wenn der Eingang nicht verbunden ist, wird der Wert der Eigenschaft "Intervall" für die Auslösung verwendet. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Auslösung | Datenpunkt des Modules der TRUE liefert, wenn das Modul ausgelöst hat. Dies erfolgt bei aktiven Timer jede Sekunde. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn der Timer ausgelöst hat. |
| Keine Auslösung | Datenpunkt des Modules der TRUE liefert, wenn das Modul nicht ausgelöst hat. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn der Timer nicht ausgelöst hat. |

# Timer Zyklisch Stunde

Das Module "Timer Zyklisch Stunde" ermöglicht die Realisierung eines Timers, der den Logikplan zyklisch mit dem eingestellten Intervall aufruft.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Intervall | Intervall in Stunden für die Auslösung des Timers. |

## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
|---|---|
| Aktiv | Aktivierung des Timers, bei TRUE wird der Timer aktiviert und löst den Logikplan zum vorgegebenen Intervall stündlich aus. |
| Intervall | Datenpunkt des Modules für das Steuern des Intervalls. Wenn der Eingang nicht verbunden ist, wird der Wert der Eigenschaft "Intervall" für die Auslösung verwendet. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Auslösung | Datenpunkt des Modules der TRUE liefert, wenn das Modul ausgelöst hat. Dies erfolgt bei aktiven Timer jede Sekunde. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn der Timer ausgelöst hat. |
| Keine Auslösung | Datenpunkt des Modules der TRUE liefert, wenn das Modul nicht ausgelöst hat. Der nachfolgende Zweig des Logikplans wird nur ausgeführt wenn der Timer nicht ausgelöst hat. |

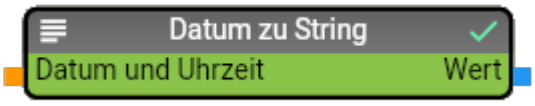
# Umwandlung

Module der Gruppe "Umwandlung" ermöglichen das Einbinden von Modulen zum Konvertieren von Daten in ein anderes Format.

| | |
|---|---|
| Datum zu String | Konvertiert einen Unix Timestamp in einen String |
| Datum zu Zeit | Konvertiert eines Unix Timestamp in die einzelnen Zeit Komponenten. |
| Meldung formatieren | Formatiert mehrere Werte zu einer Meldung. |
| Wert inventieren | Invertiert einen Wert. |
| Wert zu Boolean | Konvertiert einen Wert in einen Boolean. |
| Wert zu Float | Konvertiert einen Wert in einen Float. |
| Wert zu Integer | Konvertiert einen Wert in einen Integer. |
| Wert zu String | Konvertiert einen Wert in einen String. |
| Wertebereich konvertieren | Konvertieren eines Werte von einem Wertebereich in einen anderen. |
| Zeit zu Datum | Konvertiert einzelne Zeit Komponenten in einen Unix Timestamp. |

# Datum zu String

Das Module "Datum zu String" bietet die Möglichkeit um einen Unix Zeitstempel in einen String zu konvertieren.




## PROPERTIES

| NAME | BESCHREIBUNG |
| --- | --- |
| Datumsformat | Format, das für die Konvertierung in einen String verwendet werden soll. |

Es kann jedes beliebige PHP Format für eine Konvertierung eines Datums eingetragen werden. Details zu den möglichen Formaten findet man in der PHP Dokumentation.
Einige allgemeine Formatierung sind bereits vordefiniert und können über das DropDown ausgewählt werden.

## DATENPUNKTE EINGANG

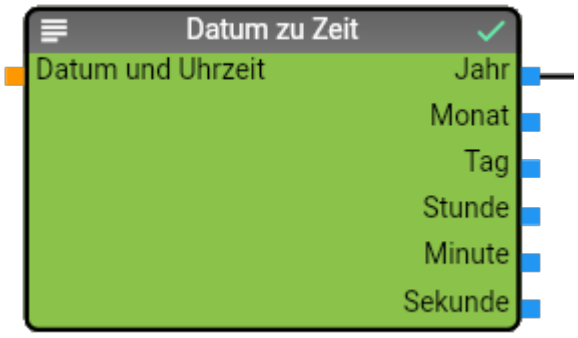
| NAME | BESCHREIBUNG |
| --- | --- |
| Zeitstempel | Unix Zeitstempel, der konvertiert werden soll. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
| --- | --- |
| Wert | Konvertierter Zeitstempel als String. |

# Datum zu Zeit

Das Module "Datum zu Zeit" bietet die Möglichkeit um einen Unix Zeitstempel in die einzelnen Zeit Komponenten zu zerlegen.




## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
|---|---|
| Zeitstempel | Unix Zeitstempel, der konvertiert werden soll. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Jahr | Das Jahr aus dem Unix Zeitstempel. |
| Monat | Das Monat aus dem Unix Zeitstempel. |
| Tag | Der Tag aus dem Unix Zeitstempel. |
| Stunde | Die Stunden aus dem Unix Zeitstempel. |
| Minute | Die Minuten aus dem Unix Zeitstempel. |
| Sekunde | Die Sekunden aus dem Unix Zeitstempel. |

# Meldung formatieren

Das Module "Meldung formatieren" bietet die Möglichkeit um einen String mit einem oder mehreren Variablen Werten zusammen zu bauen. Das Modul unterstützt bis zu 8 verschiedene Variablen.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Anzahl Eingänge | Anzahl der Eingänge, die das Modul zur Verfügung stellen soll. |
| Format String | PHP Format String zur Formatierung des Strings |

## DATENPUNKTE EINGANG

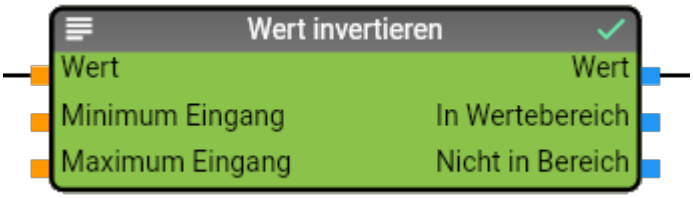
| NAME | BESCHREIBUNG |
|---|---|
| Wert 1 | Wert 1 für das Bilden der Meldung |
| Wert 2 | Wert 2 für das Bilden der Meldung |
| Wert 3 | Wert 3 für das Bilden der Meldung |
| Wert 4 | Wert 4 für das Bilden der Meldung |
| Wert 5 | Wert 5 für das Bilden der Meldung |
| Wert 6 | Wert 6 für das Bilden der Meldung |
| Wert 7 | Wert 7 für das Bilden der Meldung |
| Wert 8 | Wert 8 für das Bilden der Meldung |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Ergebnis | Ergebnis Meldung aus dem Format String und den Eingangs Variablen. |

# Wert inventieren

Das Module "Wert inventieren" bietet die Möglichkeit um einen Wert zu invertieren. Das jeweilige Minimum bzw. Maximum kann entweder über die Eigenschaften des Modules konfiguriert werden oder alternativ bietet das Modul auch Eingänge für die Werte an.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Ausgänge Bereich | Aktiviert die Datenpunkte am Ausgang "In Bereich" und "Nicht in Bereich". Standardmäßig sind diese Datenpunkte nicht aktiviert. |
| Eingänge Limit | Aktiviert die Datenpunkte am Eingang für die Limits ("Minimum Eingang" und "Maximum Eingang"). Standardmäßig sind diese Datenpunkte nicht aktiviert. |
| Maximum Eingang | Spezifiziert das Maximum für die Invertierung des Wertes, wird nur verwendet wenn die Datenpunkte für die Limits nicht aktiviert sind. |
| Minimum Eingang | Spezifiziert das Minimum für die Invertierung des Wertes, wird nur verwendet wenn die Datenpunkte für die Limits nicht aktiviert sind. |

## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
|---|---|
| Wert | Wert der invertiert werden soll |
| Minimum Eingang | Minimum Wert für die Invertierung. Dieser Datenpunkt ist nur vorhanden wenn die Eigenschaft "Eingänge Limit" aktiviert wurde. |
| Maximum Eingang | Maximum Wert für die Invertierung. Dieser Datenpunkt ist nur vorhanden wenn die Eigenschaft "Eingänge Limit" aktiviert wurde. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
| --- | --- |
| Wert | Konvertierter Wert |
| In Bereich | Liefert **TRUE** wenn sich der zu invertierende Wert innerhalb von Minimum und Maximum befand. Dieser Datenpunkt ist nur vorhanden wenn die Eigenschaft "Ausgänge Bereich" aktiviert wurde. |
| Nicht in Bereich | Liefert **TRUE** wenn sich der zu invertierende Wert außerhalb von Minimum und Maximum befand. Dieser Datenpunkt ist nur vorhanden wenn die Eigenschaft "Ausgänge Bereich" aktiviert wurde. |

# Wert zu Boolean

Das Module "Wert zu Boolean" bietet die Möglichkeit um einen Wert in einen Boolean zu konvertieren.




## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
| --- | --- |
| Wert | Wert der konvertiert werden soll. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
| --- | --- |
| Wert | Konvertierter Wert als Boolean. |

# Wert zu Float

Das Module "Wert zu Integer" bietet die Möglichkeit um einen Wert in einen Float zu konvertieren.




## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
| --- | --- |
| Wert | Wert der konvertiert werden soll. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
| --- | --- |
| Wert | Konvertierter Wert als Float. |

# Wert zu Integer

Das Module "Wert zu Integer" bietet die Möglichkeit um einen Wert in einen Integer zu konvertieren.




## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
| --- | --- |
| Wert | Wert der konvertiert werden soll. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
| --- | --- |
| Wert | Konvertierter Wert als Integer. |

# Wert zu String

Das Module "Wert zu String" bietet die Möglichkeit um einen Wert in einen String zu konvertieren.




## DATENPUNKTE EINGANG

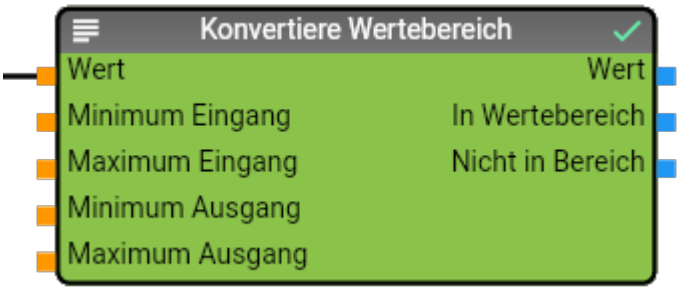
| NAME | BESCHREIBUNG |
| --- | --- |
| Wert | Wert der konvertiert werden soll. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
| --- | --- |
| Wert | Konvertierter Wert als String. |

# Wertebereich konvertieren

Das Module "Konvertiere Bereich" bietet die Möglichkeit um einen Wert von einem Wertebereich in einen anderen Wertebereich zu konvertieren. Beispielsweise ist es mit dem Modul möglich einen Wert der im Bereich 0 bis 1 liegt in den Wertebereich 0 bis 100 zu konvertieren.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Ausgänge Bereich | Aktiviert die Datenpunkte am Ausgang "In Bereich" und "Nicht in Bereich". Standardmäßig sind diese Datenpunkte nicht aktiviert. |
| Eingänge Limit | Aktiviert die Datenpunkte am Eingang für die Limits ("Minimum Eingang" und "Maximum Eingang"). Standardmäßig sind diese Datenpunkte nicht aktiviert. |
| Maximum Eingang | Spezifiziert das Maximum für den Eingangsbereich des Wertes, wird nur verwendet wenn die Datenpunkte für die Limits nicht aktiviert sind. |
| Minimum Eingang | Spezifiziert das Minimum für den Eingangsbereich des Wertes, wird nur verwendet wenn die Datenpunkte für die Limits nicht aktiviert sind. |
| Maximum Ausgang | Spezifiziert das Maximum für den Ausgangsbereich des Wertes, wird nur verwendet wenn die Datenpunkte für die Limits nicht aktiviert sind. |
| Minimum Ausgang | Spezifiziert das Minimum für den Ausgangsbereich des Wertes, wird nur verwendet wenn die Datenpunkte für die Limits nicht aktiviert sind. |
| Rundung auf Stellen | Spezifiziert die Anzahl der Stellen, die für das Runden des konvertierten Wertes verwendet werden soll. |

## DATENPUNKTE EINGANG

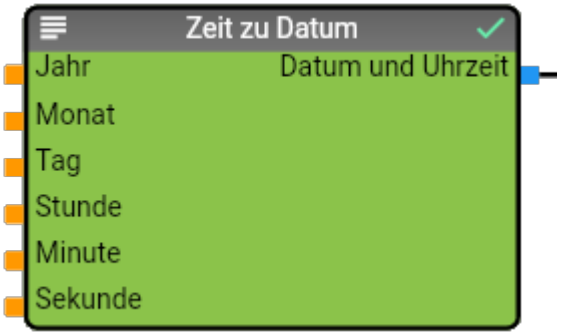
| NAME | BESCHREIBUNG |
| --- | --- |
| Wert | Wert der invertiert werden soll |
| Minimum Eingang | Minimum Wert für den Eingangsbereich. Dieser Datenpunkt ist nur vorhanden wenn die Eigenschaft "Verbinder für Limits" aktiviert wurde. |
| Maximum Eingang | Maximum Wert für den Eingangsbereich. Dieser Datenpunkt ist nur vorhanden wenn die Eigenschaft "Verbinder für Limits" aktiviert wurde. |
| Minimum Ausgang | Minimum Wert für den Ausgangsbereich. Dieser Datenpunkt ist nur vorhanden wenn die Eigenschaft "Verbinder für Limits" aktiviert wurde. |
| Maximum Ausgang | Maximum Wert für den Ausgangsbereich. Dieser Datenpunkt ist nur vorhanden wenn die Eigenschaft "Verbinder für Limits" aktiviert wurde. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
| --- | --- |
| Wert | Konvertierter Wert |
| In Bereich | Liefert "true" wenn sich der zu konvertierende Wert innerhalb von Minimum und Maximum befand. Dieser Datenpunkt ist nur vorhanden wenn die Eigenschaft "Verbinder für Bereichsüberprüfung" aktiviert wurde. |
| Nicht in Bereich | Liefert "true" wenn sich der zu konvertierende Wert außerhalb von Minimum und Maximum befand. Dieser Datenpunkt ist nur vorhanden wenn die Eigenschaft "Verbinder für Bereichsüberprüfung" aktiviert wurde. |

# Zeit zu Datum

Das Module "Zeit zu Datum" bietet die Möglichkeit um einzelne Zeit Komponenten in einen Unix Zeitstempel umzuwandeln.




## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
|---|---|
| Jahr | Das Jahr für den Unix Zeitstempel (wird kein Wert gesetzt, dann wird das aktuelle Jahr verwendet). |
| Monat | Das Monat für den Unix Zeitstempel (wird kein Wert gesetzt, dann wird das aktuelle Monat verwendet). |
| Tag | Der Tag für den Unix Zeitstempel (wird kein Wert gesetzt, dann wird der aktuelle Tag verwendet). |
| Stunde | Die Stunden für den Unix Zeitstempel (wird kein Wert gesetzt, dann wird die aktuelle Stunde verwendet). |
| Minute | Die Minuten für den Unix Zeitstempel (wird kein Wert gesetzt, dann wird die aktuelle Minute verwendet). |
| Sekunde | Die Sekunden für den Unix Zeitstempel (wird kein Wert gesetzt, dann wird die aktuelle Sekunde verwendet). |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Zeitstempel | Unix Zeitstempel, der aus den einzelnen Zeit Komponenten gebildet wurde. |

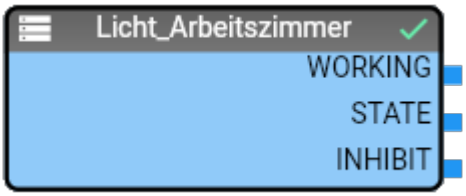
# Variablen lesen/schreiben

Module der Gruppe "Variable Lesen/Schreiben" ermöglichen das Lesen und Setzen von Variablen bzw. Instanzen in IP-Symcon.

| | |
|---|---|
| Instanz Lesen | Instanzen als Eingang eines Logikplans |
| Instanz Schreiben | Instanz als Ausgang eines Logikplans |
| Variable Lesen | Variable als Eingang eines Logikplans |
| Variable Schreiben | Variable als Ausgang eines Logikplans |
| Logikplan Variable | Variable durch den Logikplan erstellen |

# Instanz Lesen

Das Module "Instanz Lesen" bietet die Möglichkeit zur Einbindung von Instanzen als Eingang des Logikplans, Variablen der entsprechenden Instanz können zur Steuerung anderen Module verwendet werden.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| ID | Die ID der Instanz in IP-Symcon. |
| Logikplan Ausführung | Ermöglicht die automatische Ausführung des Logikplans bei einer Änderung einer Variable. |

Die folgenden Werte stehen bei der Logikplan Ausführung zur Auswahl:

Nie - keine automatische Ausführung des Logikplans

Bei Änderung - automatische Ausführung des Logikplans wenn sich der Wert einer verlinkten Variable ändert.

Bei Aktualisierung - automatische Ausführung des Logikplans wenn der Wert einer verlinkten Variable aktualisiert wird.

Als Aktionsskript - das Logikplanskript wird als Aktionsskript für jede in der Instanz verlinkte Variable hinterlegt. Bei einer Bedienung in der GUI (WebFront oder IPSView) wird der Logikplan mit dem neuen Wert aufgerufen.

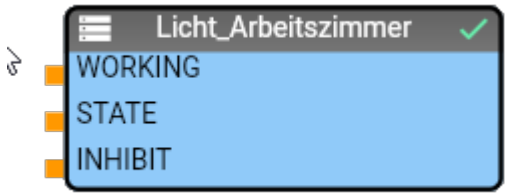
Zusätzliche Informationen zum Thema Aktionsskript.

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Wert Instanzvariable | Die Werte aller Instanzvariablen stehen als Datenpunkte zur Verfügung. |

# Instanz Schreiben

Das Module "Instanz schreiben" bietet die Möglichkeit zur Einbindung von Instanzen als Ausgang des Logikplans.




## PROPERTIES

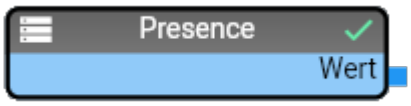
| NAME | BESCHREIBUNG |
| --- | --- |
| ID | Die ID der Instanz in IP-Symcon. |

## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
| --- | --- |
| Wert Instanzvariable | Instanzvariablen, die ein Aktionsskript hinterlegt haben, stehen als Datenpunkte zur Verfügung. |

# Variable Lesen

Das Module "Variable" bietet die Möglichkeit zur Einbindung von normalen Variablen als Eingang des Logikplans.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| ID | Die ID der Variablen in IP-Symcon. |
| Logikplan Ausführung | Ermöglicht die automatische Ausführung des Logikplans bei einer Änderung der Variable. |

Die folgenden Werte stehen bei Logikplan Ausführung zur Auswahl:

Nie - keine automatische Ausführung des Logikplans

Bei Änderung - automatische Ausführung des Logikplans wenn sich der Wert der Variable ändert.

Bei Aktualisierung - automatische Ausführung des Logikplans wenn der Wert der Variable aktualisiert wird.

Als Aktionsskript - das Logikplanskript wird als Aktionsskript für die Variable hinterlegt. Bei einer Bedienung in der GUI (WebFront oder IPSView) wird der Logikplan mit dem neuen Wert aufgerufen.

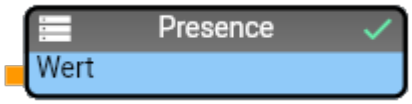
Zusätzliche Informationen zum Thema Aktionsskript.

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Wert | Der Wert der Variable. |

# Variable Schreiben

Das Module "Variable" bietet die Möglichkeit zur Einbindung von normalen Variablen als Ausgang des Logikplans.




## PROPERTIES

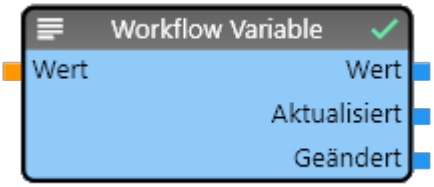
| NAME | BESCHREIBUNG |
|---|---|
| ID | Die ID der Variablen in IP-Symcon. |
| Benutze Aktionsskript | Bei TRUE wird das Aktionsskript zum Setzen des Variable-Wertes verwendet. Wenn kein Aktionsskript vorhanden ist, wird der Wert der Variable direkt gesetzt. |
| Sender | Wert für die Variable SENDER beim Aufruf des Aktionsskriptes. |

## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
|---|---|
| Wert | Der Wert der Variable. |

# Logikplan Variable

Das Module "Logikplan Variable" bietet die Möglichkeit eine Variable durch den Logikplan selbst erstellen zu lassen.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Datentype | Datentype der Variable |
| Defaultwert | Wert der Variable, der nach dem initialen Erstellen der Variable zugewiesen werden soll. |
| Profilename | Name des Profiles, das der Variable nach dem initialen Erstellen der Variable zugewiesen werden soll. |
| Variablenname | Name der Variable |
| Logikplan als Aktionsskript | Bei Aktivierung dieser Option, wird der Logikplan selbst zum Setzen eines Wertes über eine Visualisierung aufgerufen. In diesem Fall müssen Sie sich im Logikplan um die Zuweisung des neuen Wertes kümmern, ansonsten wird der Wert nicht gesetzt. |

## DATENPUNKTE EINGANG

| NAME | BESCHREIBUNG |
|---|---|
| Wert | Wert der Variable. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Wert | Aktueller Wert der Variable |
| Aktualisierung | Signalisiert eine Aktualisierung der Variable über den Eingang |
| Veränderung | Signalisiert eine Änderung der Variable über den Eingang |

# Vergleichen

Module der Gruppe "Vergleichen" ermöglichen das Vergleichen von Werten im Logikplan.

| | |
|---|---|
| Gleich | Vergleichen von Werten auf Gleichheit oder Ungleichheit |
| Größer/Kleiner | Vergleichen von Werten auf Größer, Größergleich, Gleich, Kleinergleich oder Kleiner. |
| Im Wertebereich | Überprüft einen Wert auf einen bestimmten Wertebereich |

# Gleich

Das Module "Gleich" bietet die Möglichkeit zum Vergleich verschiedener Werte auf Gleichheit oder Ungleichheit.




## DATENPUNKTE EINGANG

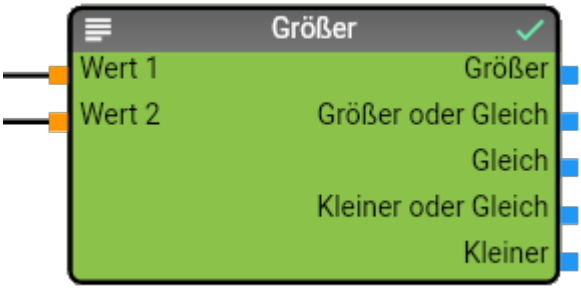
| NAME | BESCHREIBUNG |
|---|---|
| Wert 1 | Datenpunkt für Vergleichswert 1. |
| Wert 2 | Datenpunkt für Vergleichswert 2. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Gleich | Liefert TRUE wenn der Wert am Eingang Wert 1 gleich dem Wert am Eingang Wert 2 ist. |
| Nicht Gleich | Liefert TRUE wenn der Wert am Eingang Wert 1 ungleich dem Wert am Eingang Wert 2 ist |

# Größer/Kleiner

Das Module "Größer/Kleiner" bietet die Möglichkeit zum Vergleich verschiedener Werte auf Größer, Größergleich, Gleich, Kleinergleich oder Kleiner.




## DATENPUNKTE EINGANG

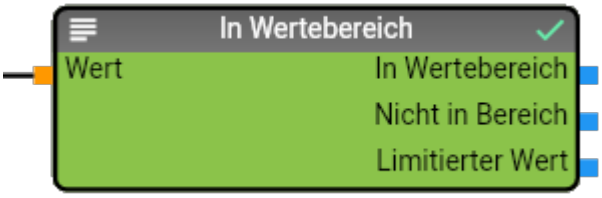
| NAME | BESCHREIBUNG |
|---|---|
| Wert 1 | Datenpunkt für Vergleichswert 1. |
| Wert 2 | Datenpunkt für Vergleichswert 2. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| Größer | Liefert TRUE wenn der Wert am Eingang Wert 1 größer dem Wert am Eingang Wert 2 ist. |
| Größergleich | Liefert TRUE wenn der Wert am Eingang Wert 1 größer oder gleich dem Wert am Eingang Wert 2 ist. |
| Gleich | Liefert TRUE wenn der Wert am Eingang Wert 1 gleich dem Wert am Eingang Wert 2 ist. |
| Kleinergleich | Liefert TRUE wenn der Wert am Eingang Wert 1 kleiner oder gleich dem Wert am Eingang Wert 2 ist. |
| Kleiner | Liefert TRUE wenn der Wert am Eingang Wert 1 kleiner dem Wert am Eingang Wert 2 ist. |

# Im Wertebereich

Das Module "In Wertebereich" bietet die Möglichkeit zur Überprüfung ob ein Wert innerhalb eines bestimmten Wertebereiches liegt. Die Limits für den Wertebereich können über Eigenschaften definiert werden oder auch über eigene Modul Eingänge.




## PROPERTIES

| NAME | BESCHREIBUNG |
|---|---|
| Eingänge Limits | Aktiviert die Datenpunkte am Eingang für die Limits ("Unteres Limit" und "Oberes Limit"). Standardmäßig sind diese Datenpunkte nicht aktiviert. |
| Oberes Limit | Spezifiziert das Maximum für den Eingangsbereich des Wertes, wird nur verwendet wenn die Datenpunkte für die Limits nicht aktiviert sind. |
| Unteres Limit | Spezifiziert das Minimum für den Eingangsbereich des Wertes, wird nur verwendet wenn die Datenpunkte für die Limits nicht aktiviert sind. |

## DATENPUNKTE EINGANG

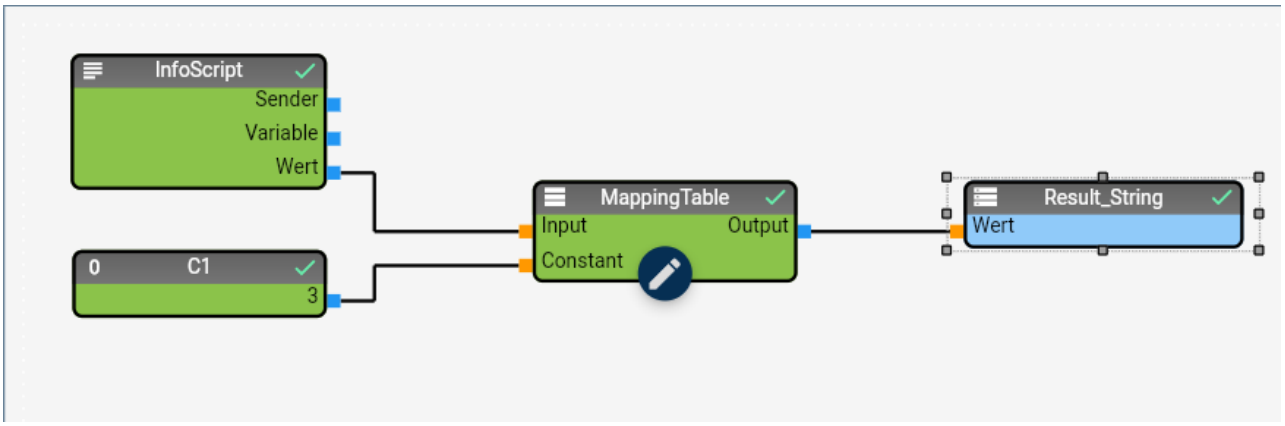
| NAME | BESCHREIBUNG |
|---|---|
| Wert | Wert der evaluiert werden soll |
| Unteres Limit | Minimum Wert für den Eingangsbereich. Dieser Datenpunkt ist nur vorhanden wenn die Eigenschaft "Eingänge Limits" aktiviert wurde. |
| Oberes Limit | Maximum Wert für den Eingangsbereich. Dieser Datenpunkt ist nur vorhanden wenn die Eigenschaft "Eingänge Limits" aktiviert wurde. |

## DATENPUNKTE AUSGANG

| NAME | BESCHREIBUNG |
|---|---|
| In Bereich | Liefert TRUE wenn sich der zu konvertierende Wert innerhalb von Minimum und Maximum befand. |
| Nicht in Bereich | Liefert TRUE wenn sich der zu konvertierende Wert außerhalb von Minimum und Maximum befand. |
| Limitierter Wert | Wenn der Wert außerhalb des oberen oder unteren Limit ist, wird der Wert auf das jeweilige Limit korrigiert. |

# Zeichenfläche

In der Mitte des Logikpläne Designers befindet sich die Zeichenfläche. Auf dieser können Sie die einzelnen Module zu einem Logikplan zusammensetzen.



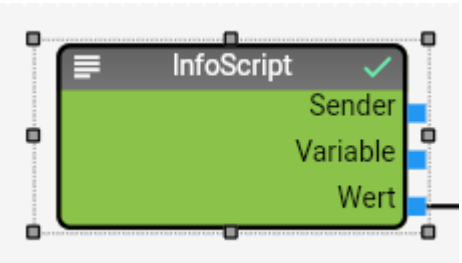

Die Zeichenfläche ist intuitiv bedienbar; die Funktionsweise ist anderen Programmen mit grafischen Komponenten sehr ähnlich. In diesem Abschnitt finden Sie eine Übersicht der verfügbaren Funktionen und Aktionen.

## OBJEKT PLATZIEREN

Um ein neues Objekt auf der Zeichenfläche zu platzieren:
  Wählen Sie das Objekt im Objektbaum oder aus dem Menu Repository aus.
  Ziehen Sie das gewählte Steuerelement auf die Zeichenfläche (Drag & Drop).
Das Objekt wird mit den Standardwerten aus IP-Symcon (Größe, Farbe …) gezeichnet. Sie müssen es anschließend für Ihre Zwecke anpassen (siehe unten).

## OBJEKTE AUSWÄHLEN / AUSWAHL AUFHEBEN

Um ein einzelnes Objekt auszuwählen:
  Klicken Sie das Objekt an.
Um mehrere Objekte gleichzeitig auszuwählen:
  Während des Klicken auf verschiedene Objekte **<Umschalten>** gedrückt halten.
Um alle Objekte auf der Zeichenfläche gleichzeitig auszuwählen:
  Drücken Sie **<Strg>+A**.
Ausgewählte Objekte erkennen Sie an den acht Griffpunkten (Kästchen) am Rahmen:




Um die Auswahl wieder aufzuheben:
  Klicken Sie auf eine leere Stelle der Zeichenfläche.
Die gesamte getätigte Auswahl wird aufgehoben.

# EIGENSCHAFTEN ÄNDERN

Die hinterlegten Werte und Eigenschaften (Properties) des ausgewählten Objekts werden im Property-Editor angezeigt. Dort können Sie diese nach Bedarf ändern.
Haben Sie mehrere Objekte gleichzeitig ausgewählt, können Sie jene Properties, die in allen ausgewählten Objekten vorkommen, ändern (Properties, die nur in einzelnen Objekten vorkommen, werden nicht angezeigt). Ihre Änderung wird dann auf alle derzeit ausgewählten Objekte gleichzeitig angewandt.

# AUSSCHNEIDEN, KOPIEREN, EINFÜGEN, LÖSCHEN

Logikpläne unterstützt die Standardbefehle zum Manipulieren von Objekten und Inhalten:

| NAME | BESCHREIBUNG |
|---|---|
| Ausschneiden | Kontextmenü?Ausschneiden oder die Tastenkombination **<Strg>+X**. |
| Kopieren | Kontextmenü?Kopieren oder die Tastenkombination **<Strg>+C**. |
| Einfügen | Kontextmenü?Einfügen oder die Tastenkombination **<Strg>+V**. |
| Löschen | Kontextmenü?Löschen oder die Taste **<Entf>**. |

# RÜCKGÄNGIG / WIEDERHERSTELLEN

Logikpläne unterstützt "Undo / Redo", also das Annullieren bzw. Wiederherstellen von Änderungen:

| NAME | BESCHREIBUNG |
|---|---|
| Rückgängig | Menü+Rückgängig oder die Tastenkombination **<Strg>+Z**. |
| Wiederherstellen | Menü+Wiederherstellen oder die Tastenkombination **<Strg>+<Umschalttaste>+Z**. |

# ZOOMEN EINES LOGIKPLANS

Der Logikpläne Designer bietet die Möglichkeit eine View in Schritten von 10% beliebig zu verkleinern und zu vergrößern. Dabei wird nur die Anzeige im Designer verändert, eine Veränderung der Darstellung am Client findet dabei nicht statt.

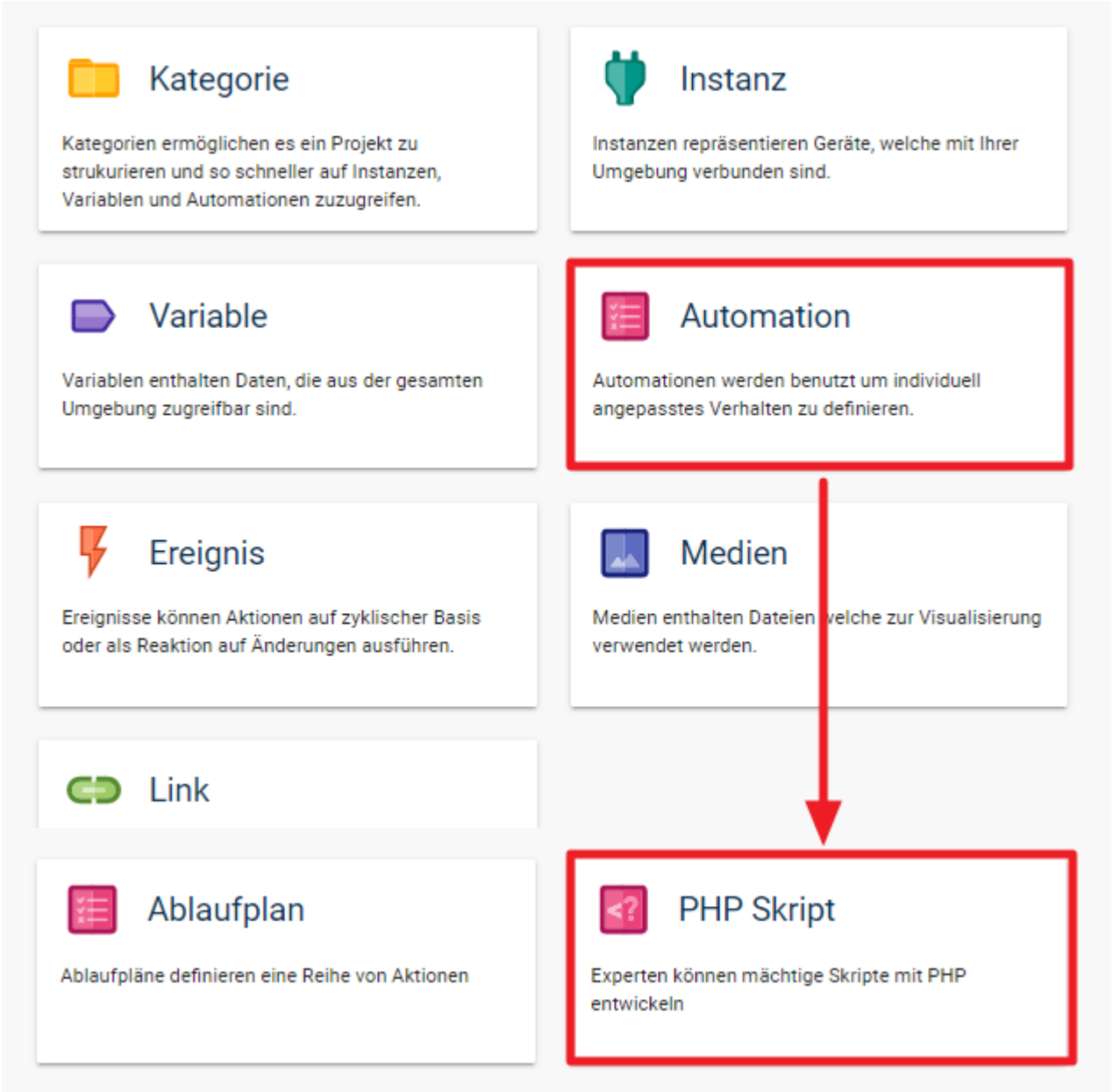
| NAME | BESCHREIBUNG |
|---|---|
| Zoom In | Wählen Sie Menü+Zoom In oder drücken Sie die Tastenkombination ++, um die Anzeige um eine Stufe zu vergrößern. |
| Zoom Out | Wählen Sie Menü+Zoom Out oder drücken Sie die Tastenkombination +-, um die Anzeige um eine Stufe zu verkleinern. |
| Zoom | Wählen Sie Menü+Zoom, um die Zoom Funktionalität auf einen bestimmten Wert zu setzen. Nach Betätigung des Buttons wird ein DropDown Fenster angezeigt, das die Auswahl der Zoomstufe erlaubt. |

# PHP-Skripte

Ein PHP-Skript ist eine Aneinanderreihung von mehr oder weniger komplexen Befehlen. IP-Symcon verwendet und unterstützt jegliche Funktionen der Programmiersprache PHP. In unserem Community-Forum oder Internet können zahllose Beispiele und weiterführende Dokumentationen gefunden werden.

## PHP-SKRIPTE HINZUFÜGEN

Über "+" -> "Automation" -> "PHP Skript" kann ein neues PHP-Skript hinzufügt werden. Es wird empfohlen dem PHP-Skript einen eindeutigen und aussagekräftigen Namen zu geben.
Der Name ist jederzeit änderbar und das PHP-Skript kann in der Baumansicht per Drag & Drop an eine andere Position verschoben werden. Alternativ kann die Erstellung per "Rechtsklick" -> "Objekt hinzufuegen" -> "Automation" -> "PHP-Skript" direkt an der gewünschten Position erfolgen.



## SKRIPTE EDITIEREN & AUSFÜHREN

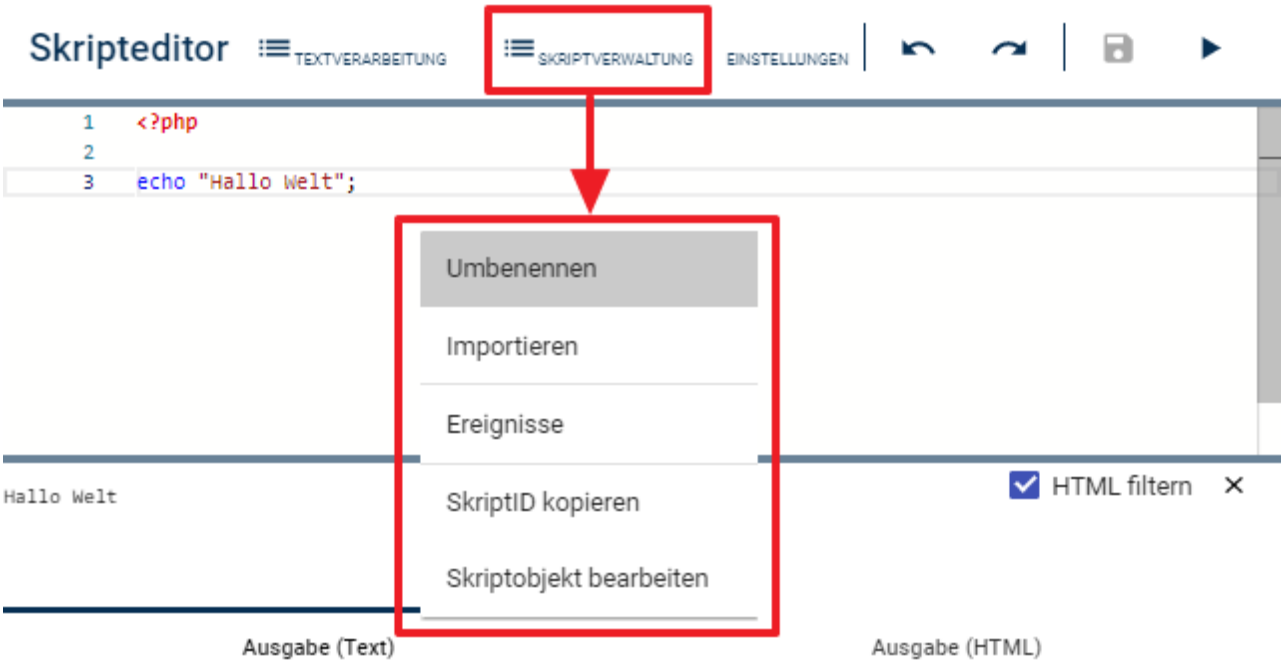
Grundsätzlich gilt: Ein PHP-Skript verlangt, dass dieses mit einem PHP-Tag (<?php) beginnt und am Ende jedes Statements/Befehl mit einem Semikolon beendet wird.

> Wird ein externer Editor genutzt muss das Skript in UTF8 ohne BOM abgespeichert werden.




Via "Ausführen" kann das Skript zu Testzwecken manuell gestartet werden.
Dabei wird es automatisch gespeichert und sowohl der Zeitstempel der letzten Ausführung als auch die Länge der Laufzeit aktualisiert. Die Ansicht kann zwischen [Text] und [HTML] umgeschaltet werden.
Fehlerhafte Eingaben können mit „Rückgängig“ oder “Wiederherstellen“ korrigiert werden.
Mit „Nach HTML exportieren“ wird das gesamte Skript im Browser angezeigt.

## SKRIPTE IMPORTIEREN & UMBENENNEN



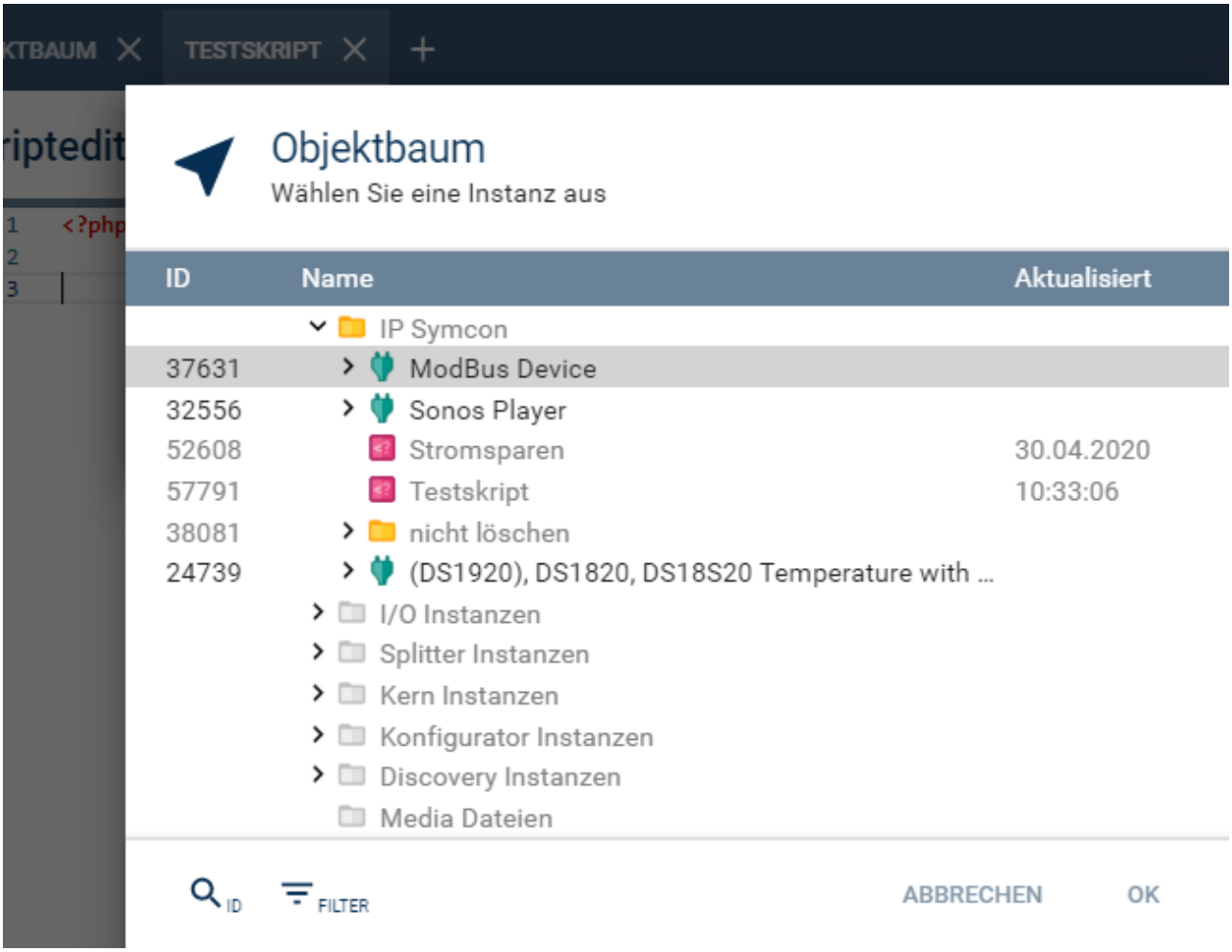

Wenn bereits ein Skript (z.B. aus dem Forum) heruntergeladen wurde, so kann dieses schnell und unkompliziert über den Menüknopf "Importieren" einfügt werden. Beim Drücken von "Importieren" öffnet sich ein Menü, über welches die gewünschte Datei ausgewählt werden kann. Standardmäßig werden in IP-Symcon alle Skripte mit der ID des Skriptes als Dateiname erstellt. Wenn zur besseren Übersicht ein eigener Dateinamen verwendet werden soll, kann jederzeit über den Menüknopf "Umbenennen" dem Skript ein anderer Name gegeben werden.

## BEFEHLE HINZUFÜGEN

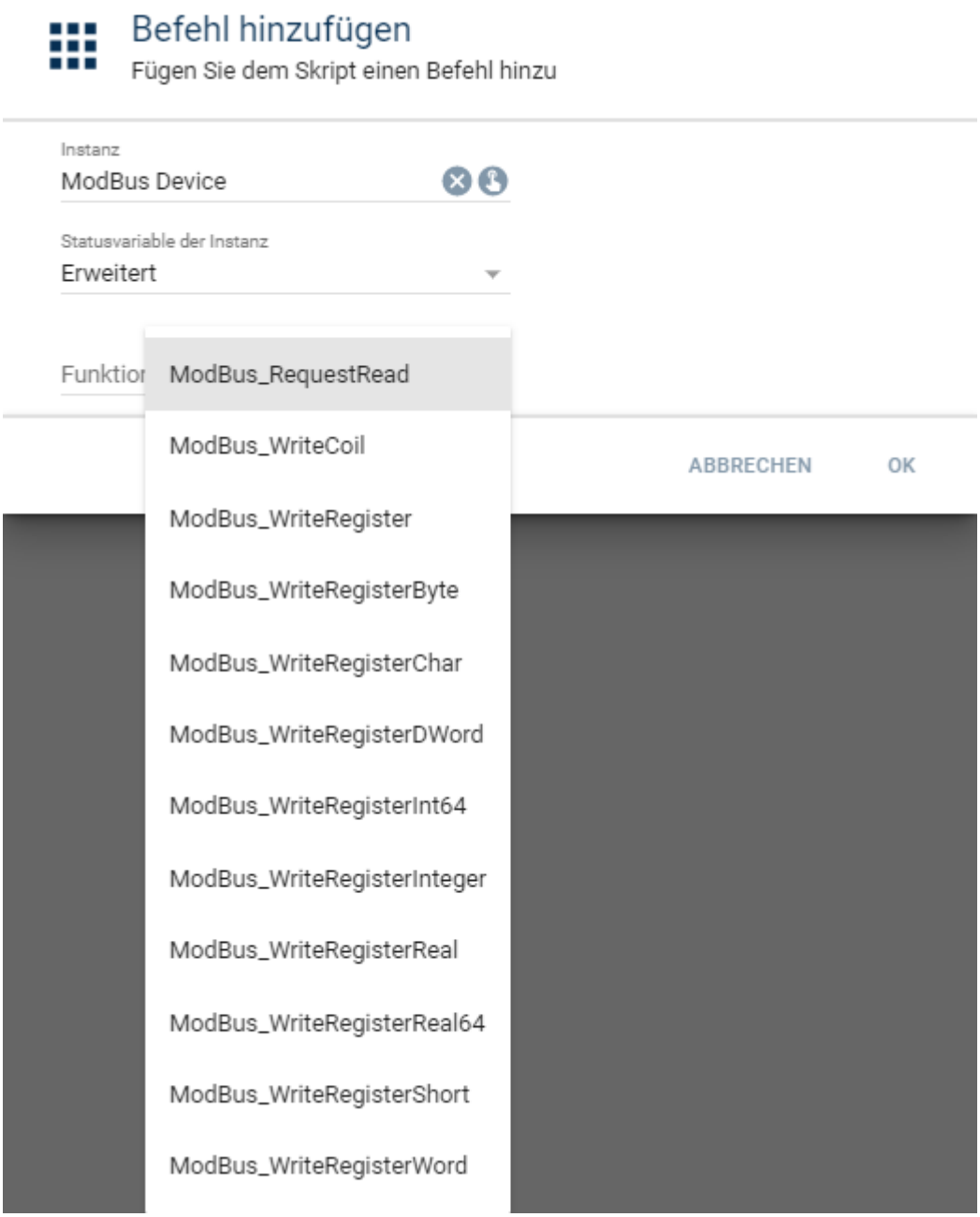
Über „Befehl hinzufügen“ können an der Cursor-Position im Skript neue Befehle eingefügt werden.




Um die Vielfalt der Befehle einzugrenzen muss das betreffende Gerät ausgewählt werden.
Hier als Beispiel für eine Statusabfrage bei einem "ModBus Device":




Als Auswahlmöglichkeit werden dann nur die gerätespezifischen Funktionen angezeigt:







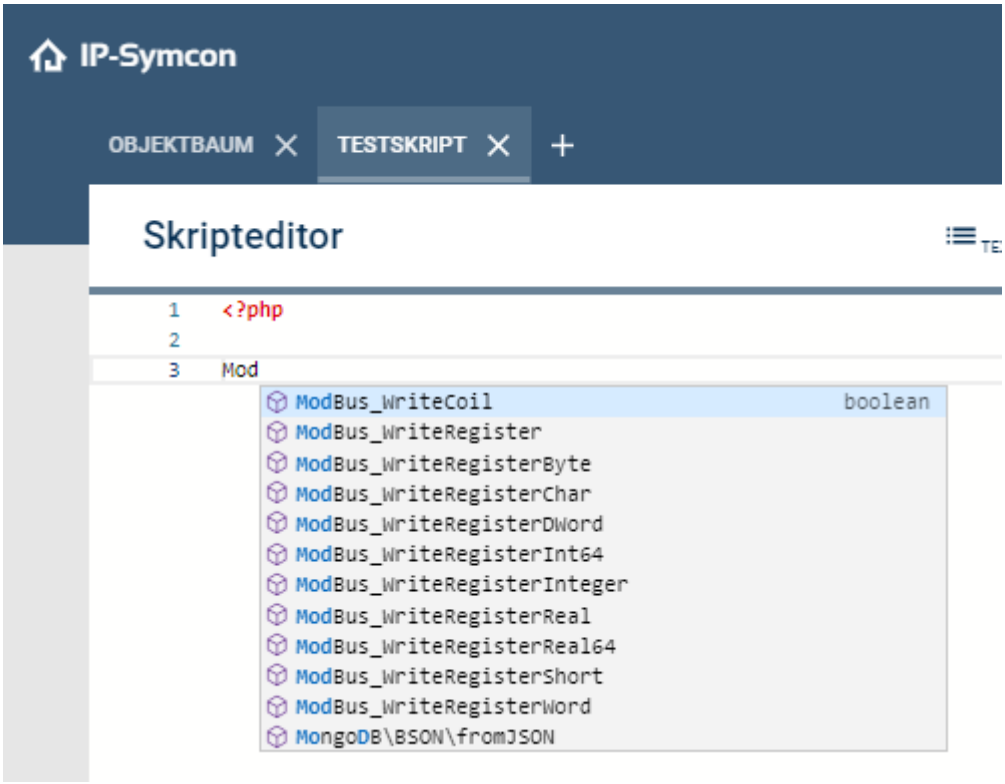
Das Ergebnis ist der Befehl mit der passenden ID-Nummer.



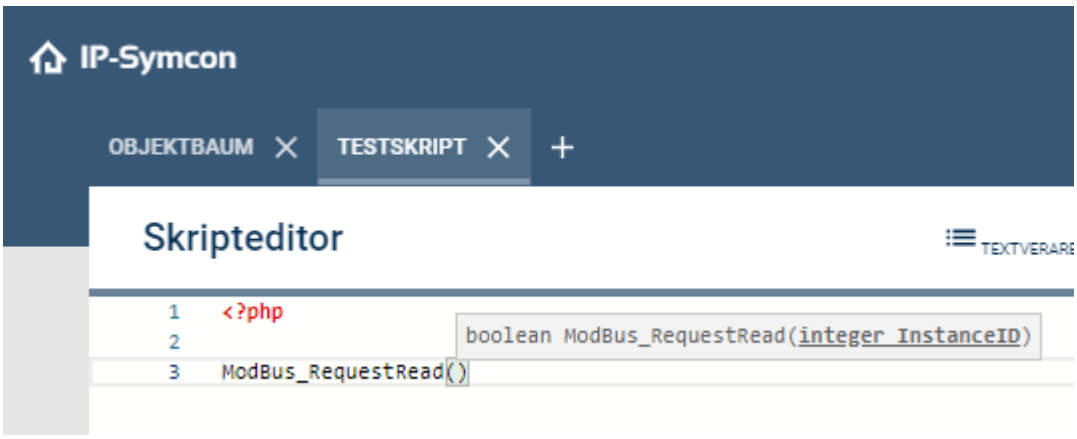

Die Dokumentation zu allen Befehlen die zur Auswahl stehen, befindet sich in der Befehlsreferenz und Modulreferenz. Eine andere Möglichkeit bietet die Suche im oberen rechten Bereich der IP-Symcon Internetseite. Diese bietet einen schnellen Weg die Dokumentation und mögliche Beispiele zu einem Befehl zu durchsuchen.

## BEFEHLSLISTE ANZEIGEN UND NUTZEN

Um einen beliebigen Befehl direkt eingeben zu können muss die Tastenkombination "Strg + Leertaste" innerhalb eines Skripts gedrückt werden. Die Liste beinhaltet dann alle IP-Symcon-Funktionen. Auch hier kann die Auswahl eingeschränkt werden indem die ersten Buchstaben eines Befehls eingegeben werden. Z.B. zeigt die Eingabe "Modbus" alle Modbus Befehle. Der gewünschte Befehl kann dann per Doppelklick oder per Pfeiltasten + Eingabe-Taste ausgewählt werden.




Wenn der Befehl vollständig ist und die Klammern gesetzt werden, können ebenfalls über "Strg + Leertaste" die benötigten Parameter angezeigt werden.




## SYSTEMVARIABLEN IN SKRIPTEN

Eine komplette Auflistung kann unter Systemvariablen gefunden werden.

## TIPPS & TRICKS

Eine Auflistung aller Tastenkombinationen.

# Aktionsskripte

Aktionsskripte sind solche Skripte, die bei einer Variable via anklicken in einer Visualisierung (z.B. WebFront, Mobile) aufgerufen werden.
Das Aktionsskript muss in der jeweiligen Variable als "Eigene Aktion" verknüpft werden.
Unter Systemvariablen können die speziellen Variablen für aufgerufene Skripte eingesehen werden.
Ein Aktionsskript kann von mehreren Variablen genutzt werden. Häufigste Variante siehe unten "Nur Variable setzen".

## AKTIONSSKRIPT AUSWÄHLEN

Das Auswählen eines Aktionsskript ist dasselbe wie das Auswählen einer Variablenaktion .

> Einige Variablen von hinzugefügten Modulen beinhalten eine "Standardaktion". Diese kann durch ein "Eigene Aktion" überschrieben werden.




> Aus einer Kombination von Dummy Modulen (Instanz), einer Variable mit verknüpften Variablenprofil und einem verknüpften Aktionsskript, kann die Funktion eines nativen IP-Symcon Moduls nachgebildet werden.

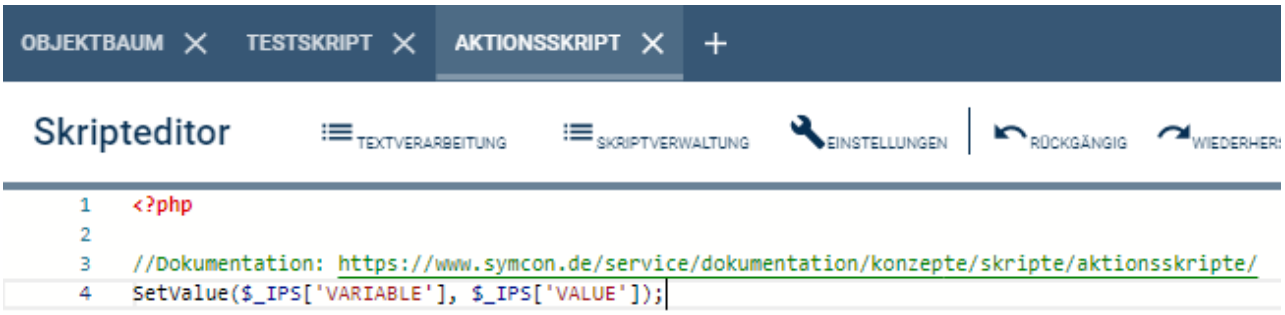
## AKTIONSSKRIPT ERSTELLEN

Ein Aktionsskript ist ein normales PHP-Skript. Erst durch das Einbinden in einer Variable wird es zum Aktionsskript.
In einem Aktionsskript sollte zumindest der Schaltvorgang und die Anweisung stehen, welche die Variable auf den angeforderten Zustand setzt:

### BEISPIEL 1

Dies ist das Beispiel, welches durch das "+" bei "Eigene Aktion" erstellt wird.

```
//Nur Variable setzen
SetValue($_IPS['VARIABLE'], $_IPS['VALUE']);
```




### BEISPIEL 2

```
//Variable nur passend setzen wenn Schaltvorgang erfolgreich
if (FS20_SwitchMode(12345, (boolean)$_IPS['VALUE'])) {
SetValue($_IPS['VARIABLE'], $_IPS['VALUE']);
}
```

## SCHLECHTE AKTIONSSKRIPTE

Es gibt Kriterien, welche unbedingt beim Erstellen eines Aktionsskriptes beachtet werden müssen. Werden diese nicht befolgt, kann es zu unvorhergesehenen Verhalten kommen.

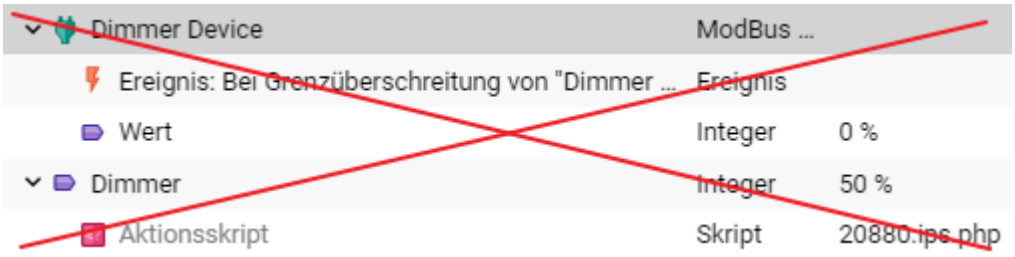
### SETZEN DER VARIABLE OHNE AUF ERFOLGREICHES SCHALTEN ZU ACHTEN

Die meisten Funktionen geben bei Erfolg ein "True" zurück und auf dies sollte auch geprüft werden.
Wird dies nicht getan ist der Status in IP-Symcon und die des Geräts nicht mehr konsistent zueinander. Somit kann es passieren das die eingerichtete automatische Steuerung nicht korrekt arbeitet.

```
//Variable wird auf neuen Wert gesetzt, selbst wenn der
Schaltvorgang nicht erfolgreich war. Es kann zu Konsistensfehlern
kommen.
FS20_SwitchMode(12345, (boolean)$_IPS['VALUE']);
SetValue($_IPS['VARIABLE'], $_IPS['VALUE']);
```

### BEI VARIABLENÄNDERUNG VIA EREIGNIS EIN GERÄT SCHALTEN

Wenn der Wert einer Gerätevariable via Aktionsskript gesetzt wurde und aufgrund der Variablenänderung ein Ereignis versucht das Gerät auf diesen Wert zu setzen.
Durch diese Handhabe hat man keine Möglichkeit etwaige Schaltprobleme des Geräts zu bemerken. Dies bedeutet, dass sowohl Fehlermeldung verdeckt werden, als auch die gesetzte Variable nicht mit dem tatsächlichen Gerätewert übereinstimmt.
Ein Beispiel wäre ein Dimmer der im WebFront von 0% auf 50% gestellt wird.
Die Variable wird sofort auf 50% gesetzt und der Schaltbefehl für den Dimmer wird asynchron über das Ereignis abgearbeitet. Ist dieser Schaltbefehl nicht erfolgreich, wird das nicht im Aktionsskript erkannt und somit bleibt die Variable in IP-Symcon auf 50% eingestellt, obwohl der tatsächliche Wert noch immer 0% ist.



# Systemvariablen

Systemvariablen sind Variablen, die in jedem PHP-Skript verfügbar sind. Der Benutzer kann diese nutzen um effizient PHP-Skripte erstellen zu können, welche generisch Aufgaben lösen können. Je nach Absender, durch den das PHP-Skript ausgeführt wurde, werden automatisch passende Systemvariablen erstellt und zur Verfügung gestellt. Die nachfolgenden Tabellen geben einen Überblick bei welchem PHP-Skript welche Variablen automatisch zur Verfügung stehen.



**Die Groß- und Kleinschreibung der Variablen ist zu beachten (Case-Sensitive)!**

## GENERELLE SYSTEMVARIABLEN

Folgende Systemvariablen stehen immer zur Verfügung.

| SYSTEMVARIABLE | BESCHREIBUNG |
|---|---|
| $_IPS['SELF'] | PHP-SkriptID des aktuellen PHP-Skriptes |
| $_IPS['SENDER'] | Auslöser des PHP-Skriptes. Abhängig vom Wert können sich weitere globale Variablen ergeben. Mögliche Werte werden unter Zusätzliche_Systemvariablen genauer beschrieben. |
| $_IPS['THREAD'] | ThreadID des aktuellen PHP-Skriptes |

## ZUSÄTZLICHE SYSTEMVARIABLEN

Folgende Systemvariablen stehen in Abhängigkeit des $_IPS['SENDER'] zur Verfügung.

Action
Execute
HeatingControl
RegisterVariable
RunScript
Shutdown
ShutterControl (legacy)
Startup
StatusEvent
TimerEvent
Variable
VoIP ( ab Version 5.2 )
Watchdog
WebFront
WebHook ( ab Version 4.0 )
WebInterface
WebOAuth ( ab Version 4.0 )

### ACTION

Wenn das PHP-Skript durch ein RequestAction ausgeführt wurde.

| SYSTEMVARIABLE | BESCHREIBUNG |
|---|---|
| $_IPS['SENDER'] | Standardmäßig ist dieser "Action" kann aber durch RequestActionEx Individuell gesetzt werden |
| $_IPS['VALUE'] | Wert mit dem die Variable aufgerufen wurde |
| $_IPS['VARIABLE'] | ID der Variable, welche geschaltet werden soll |

### EXECUTE

Wenn das PHP-Skript aus der Verwaltungskonsole ausgeführt wurde.
Keine zusätzlichen Systemvariablen.

### HEATINGCONTROL

Wenn das PHP-Skript durch ein Ereignis vom HeatingControl ausgeführt wurde.

| SYSTEMVARIABLE | BESCHREIBUNG |
|---|---|
| $_IPS['INSTANCES'] | IDs der Sende-Instanzen |
| $_IPS['INVERTS'] | Beinhaltet ob ein Gerät invertiert ist (true/false) |
| $_IPS['VALUE'] | Enthält ob das Gerät heizen soll (true) oder nicht heizen soll (false) |

### REGISTERVARIABLE

Wenn das PHP-Skript durch eine RegisterVariable Instanz ausgeführt wurde.

| SYSTEMVARIABLE | BESCHREIBUNG |
|---|---|
| $_IPS['CLIENTIP'] (nur bei I/O ServerSocket) | Empfangene IP-Adresse des Clients |
| $_IPS['CLIENTPORT'] (nur bei I/O ServerSocket) | Empfangsport des Clients |
| $_IPS['INSTANCE'] | ID der auslösenden RegisterVariable Instanz |
| $_IPS['VALUE'] | Wert des Puffers, der vom Splitter oder I/O Modul empfangen wurde |
| $_IPS['TYPE'] | Status des Empfangsport (0 = Data; 1 = Verbunden; 2 = Getrennt |

### RUNSCRIPT

Wenn das PHP-Skript durch eine IPS_RunScript Funktionen ausgeführt wurde. Erweiternd kann auch IPS_RunScriptEx genutzt werden um zusätzliche Parameter zu übergeben.
Keine zusätzlichen Systemvariablen.

### SHUTDOWN

Beim IP-Symcon Shutdown (siehe EventControl).
Keine zusätzlichen Systemvariablen.

### SHUTTERCONTROL

Wenn das PHP-Skript durch ein ShutterControl Modul (legacy) ausgeführt wurde.

| SYSTEMVARIABLE | BESCHREIBUNG |
|---|---|
| $_IPS['DIRECTION'] | Zu fahrende Richtung:<br>**0** = Stop<br>**1** = Hoch<br>**2** = Runter |
| $_IPS['DURATION'] | Fahrdauer in Millisekunden |
| $_IPS['INSTANCE'] | InstanzID, die im ShutterControl eingestellt ist |
| $_IPS['INSTANCE2'] | InstanzID #2, die im ShutterControl eingestellt ist |

### STARTUP

Beim IP-Symcon Startup (siehe EventControl).
Keine zusätzlichen Systemvariablen.

### STATUSEVENT

Wenn das PHP-Skript durch eine Statusveränderung einer Instanz ausgeführt wurde.
Nähere Informationen sind unter EventControl und IPS_GetInstance zu finden.

| SYSTEMVARIABLE | BESCHREIBUNG |
|---|---|
| $_IPS['INSTANCE'] | InstanzID zur Statusänderung |
| $_IPS['STATUS'] | Status der Instanz. Eine Auflistung ist hier zu finden: IPS_GetInstance |
| $_IPS['STATUSTEXT'] | Kurztext passend zum Status |

### TIMEREVENT

Wenn das PHP-Skript durch ein zyklisches- oder Wochenplan-Ereignis ausgeführt wurde.

| SYSTEMVARIABLE | BESCHREIBUNG |
|---|---|
| $_IPS['ACTION'] | ID der aufzurufenden Aktion (nur Wochenplan Ereignis) |
| $_IPS['EVENT'] | ID des auslösenden Ereignisses |
| $_IPS['TARGET'] | ID des übergeordneten Objektes |

### VARIABLE

Wenn das PHP-Skript durch ein ausgelöstes Ereignis ausgeführt wurde.

| SYSTEMVARIABLE | BESCHREIBUNG |
|---|---|
| $_IPS['EVENT'] | ID des auslösenden Ereignisses |
| $_IPS['OLDCHANGED'] (ab Version 5.1) | Zeitstempel der letzten Änderung des alten Wertes |
| $_IPS['OLDVALUE'] | Wert der auslösenden Variable vor Auslösezeitpunkt |
| $_IPS['OLDUPDATED'] (ab Version 5.1) | Zeitstempel der letzten Aktualisierung des alten Wertes |
| $_IPS['TARGET'] | ID des übergeordneten Objektes |
| $_IPS['TRIGGER'] | Typ des auslösenden Ereignisses |
| $_IPS['VALUE'] | Wert der auslösenden Variable zum Auslösezeitpunkt |
| $_IPS['VARIABLE'] | ID der auslösenden Variable |

### VOIP ( AB VERSION 5.2 )

Wenn das PHP-Skript durch ein Ereignis vom VoIP-Modul ausgeführt wurde.

| SYSTEMVARIABLE | BESCHREIBUNG |
|---|---|
| $_IPS['CONNECTION'] | KanalID der Verbindung |
| $_IPS['DATA'] | Beinhaltet die gedrückte Taste, wenn das $_IPS['EVENT'] gleich DTMF ist |
| $_IPS['EVENT'] | Die Variable kann folgende Werte beinhalten:<br>**Incoming:** Bei eingehendem Anruf<br>**Connect:** Bei aufgebauter Verbindung<br>**Disconnect:** Bei beendeter Verbindung<br>**PlayFinish:** Sobald Audiodatei zuende abgespielt wurde<br>**DTMF:** Sobald ein DTMF Ton empfangen wurde |
| $_IPS['INSTANCE'] (ab Version 5.4) | InstanzID der VoIP-Instanz |

### WATCHDOG

Wenn das PHP-Skript durch den Watchdog ausgeführt wurde.

| SYSTEMVARIABLE | BESCHREIBUNG |
|---|---|
| $_IPS['STATUSTEXT'] | Der Statustext der übergeben wird |
| $_IPS['VALUE'] | Wert der Variable, welcher außerhalb der Grenzen ist |
| $_IPS['VARIABLE'] | ID der Variable, welche die Grenzen überschritten hat |

### WEBFRONT

Wenn das PHP-Skript durch das WebFront ausgeführt wurde.

| SYSTEMVARIABLE | BESCHREIBUNG |
| --- | --- |
| $_IPS['CONFIGURATOR'] | ID des aktuell genutzten Konfigurators |
| $_IPS['VALUE']<br>(nur bei Aktionsskript) | Neuer Wert der Variable |
| $_IPS['VARIABLE']<br>(nur bei Aktionsskript) | ID der zu ändernden Variable |

### WEBHOOK ( AB VERSION 4.0 )

Wenn das PHP-Skript durch einen WebHook ausgeführt wurde.

| SYSTEMVARIABLE | BESCHREIBUNG |
| --- | --- |
| $_SERVER['HOOK'] | Vollständige URL des aufgerufenen WebHooks |

Zusätzlich stehen beim Typ WebHook auch alle Variablen des WebInterface zur Verfügung.

### WEBOAUTH ( AB VERSION 4.0 )

Wenn das PHP-Skript durch einen OAuth Control ausgeführt wurde.

| SYSTEMVARIABLE | BESCHREIBUNG |
| --- | --- |
| $_SERVER['OAUTH'] | Vollständige URL des aufgerufenen WebOAuth |

Zusätzlich stehen beim Typ WebOAuth auch alle Variablen des WebInterface zur Verfügung.

### WEBINTERFACE

Wenn das PHP-Skript durch den Webserver (z.B. im 'user' Ordner) ausgeführt wurde.

| SYSTEMVARIABLE | BESCHREIBUNG |
|---|---|
| $_GET | Empfangene Daten über GET |
| $_POST | Empfangene Daten über POST |
| $_SERVER['DOCUMENT_ROOT'] | Enthält das Document Root-Verzeichnis, unter dem das aktuelle PHP-Skript ausgeführt wird, so wie es in der Konfiguration des Servers festgelegt wurde |
| $_SERVER['PHP_AUTH_PW'] | Wenn HTTP authentication verwendet wird, wird diese Variable mit dem vom Benutzer angegebenen Passwort gefüllt |
| $_SERVER['PHP_AUTH_USER'] | Wenn HTTP authentication verwendet wird, wird diese Variable mit dem vom Benutzer angegebenen Usernamen gefüllt |
| $_SERVER['PHP_SELF'] | Datei/Pfad des gestarteten PHP-Skriptes |
| $_SERVER['QUERY_STRING'] | Sofern vorhanden, der Querystring, mittels dessen auf die Seite zugegriffen wurde |
| $_SERVER['REMOTE_ADDR'] | Die IP-Adresse, von der aus der Benutzer die aktuelle Seite ansieht |
| $_SERVER['REMOTE_PORT'] | Der Port, von der aus der Benutzer die aktuelle Seite ansieht |
| $_SERVER['REQUEST_METHOD'] | Enthält die für den Zugriff auf die Seite verwendete Requestmethode, z. B. 'GET', 'HEAD', 'POST' oder 'PUT' |
| $_SERVER['REQUEST_URI'] | Der URI, der angegeben wurde, um auf die aktuelle Seite zuzugreifen, beispielsweise '/index.html' |
| $_SERVER['SCRIPT_NAME'] | Enthält den Pfad zum aktuellen PHP-Skript. Dies ist nützlich für Seiten, die auf sich selbst verweisen sollen |

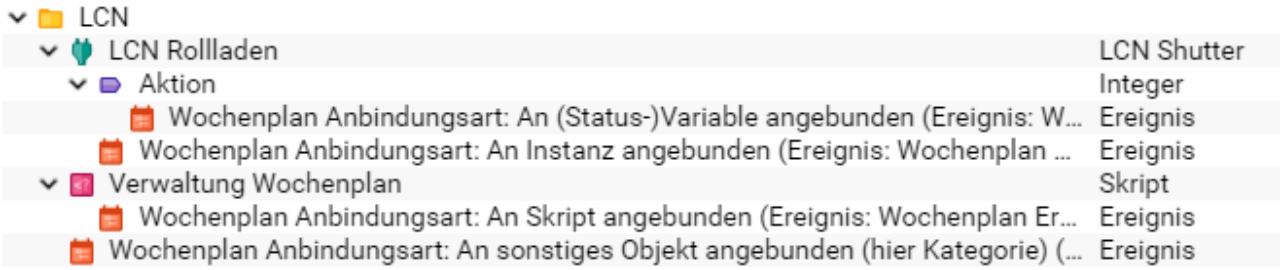
Zusätzlich werden alle Request Header in Großbuchstaben mit dem HTTP* *Prefix hinzugefügt, wobei ein Minus (-) zu einem Unterstrich (*) geändert wird. z.B. der Header 'User-Agent' wird als $_SERVER['HTTP_USER_AGENT'] kodiert.

# Ereignisse

Ein Ereignis ist eine automatisierte und an eine Bedingung geknüpfte Aktion.
Eintretende Ereignisse modifizieren angebundene Objekte. Die Anbindung geschieht durch das “Unterordnen” innerhalb der IP-Symcon Verwaltungskonsole. ( siehe: Anbindungsarten)
Ereignisse treten abhängig von ihrem Ereignistyp und eingerichteten Bedingung ein. ( siehe: Ereignistypen)

## ANBINDUNG AN AKTIONSZIEL

Das angebundene Objekt wird hierbei als Ziel für eine Aktion definiert. Wie bei Aktionen üblich, kann eine zu dem Ziel passende Aktion gewählt und konfiguriert werden.
Die Aktion eines Ereignisses bestimmt was geschehen soll, wenn das Ereignis geschaltet wird. Weitere Informationen gibt es unter Aktionen .




## EREIGNISTYPEN

Man unterscheidet zwischen drei Ereignistypen.

| TYP | BESCHREIBUNG |
|---|---|
| Zyklisch | Zu einem bestimmten und einmaligen oder wiederholenden Zeitpunkt eintretendes Ereignis. |
| Ausgelöst | An eine bestimmte Variable gebundenes Ereignis. |
| Wochenplan | Mit dem Wochenplan im WebFront konfiguriertes Ereignis. |

Zugriff auf Systemvariablen:
IP-Symcon liefert automatisch grundlegende Variablen, auf welche innerhalb von Skripten zugegriffen werden kann.
Dies funktioniert allerdings nur sofern das Skript von einem Ereignis aufgerufen wurde.
Siehe auch:
Systemvariablen

## BEDINGUNGEN

Bedingungen erweitern Ereignisse um weitere Optionen, die erfüllt sein müssen, damit das Ereigniss eintritt. Somit können Mehrfachbedingungen realisiert werden.
Weitere Bedingungen können im Abschnitt "Weitere Bedingungen" -> "Hinzufügen"

hinzugefügt werden.
Als Typ kann Variable, Datum, Wochentag oder Uhrzeit ausgewählt werden. Weitere Regeloptionen sind zum einen der Wert auf den verglichen werden soll und der angewandte Vergleichsoperator (Siehe Tabelle Vergleichsoperatoren).
In der Übersicht sind alle Regeln aufgelistet. Ein "Haken" bzw. "Kreuz" zeigt an ob eine Regel momentan erfüllt ist oder nicht. Über das Zahnradsymbol kann die die jeweilige Regel angepasst werden. Das DropDown- Menü "Evaluation" bestimmt, ob alle angegebenen Regeln erfüllt sein müssen oder schon eine erfüllte Regel ausreicht um die Ereignissaktion zu starten.
Der Live Zustand zeigt an ob das Ereignis unter den momentanen Regeln schalten würde oder nicht.

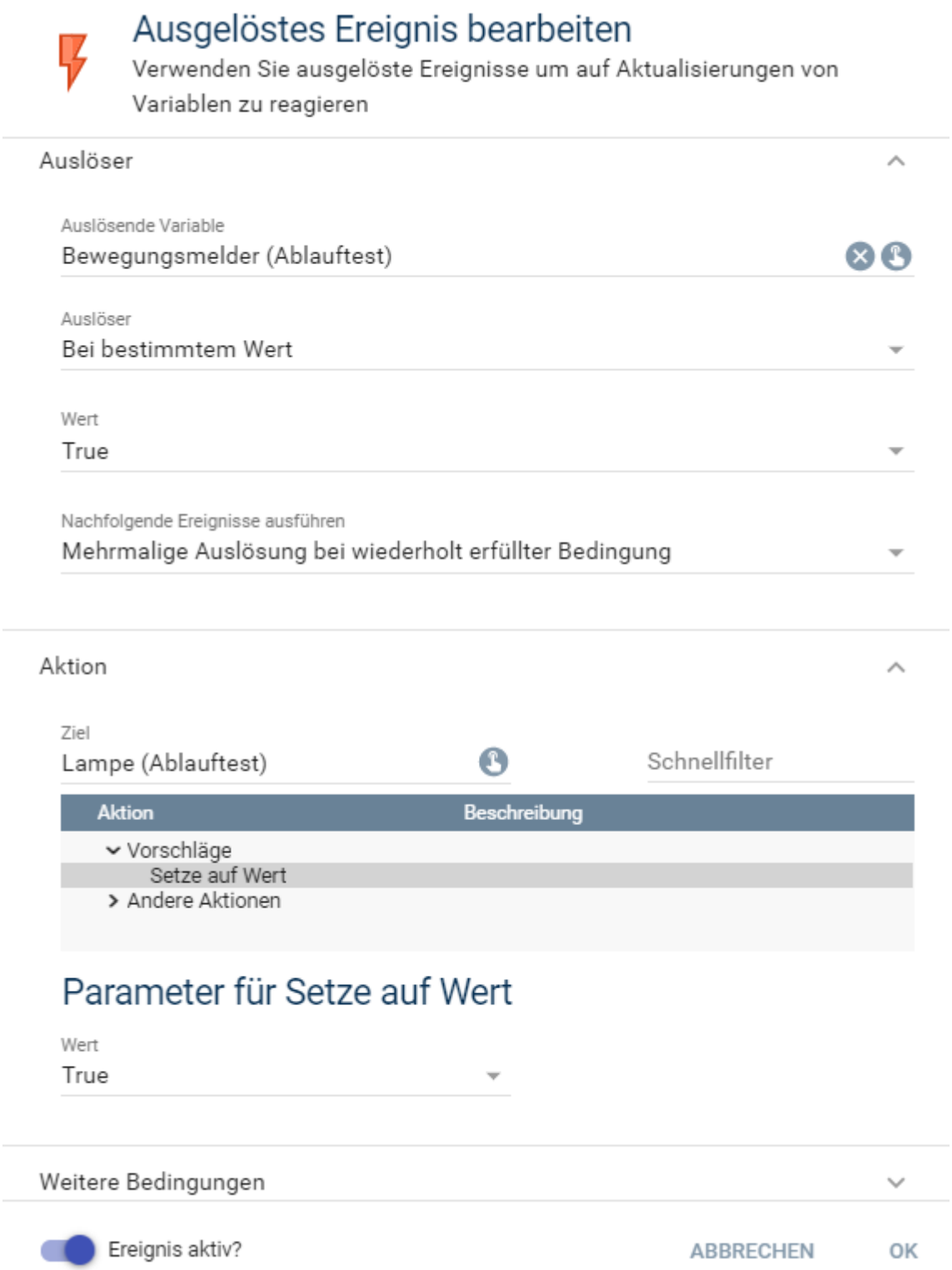
## VERGLEICHSOPERATOREN

Vergleichsoperatoren beschreiben das Verhalten, wann ein Vergleich erfüllt ist oder nicht.

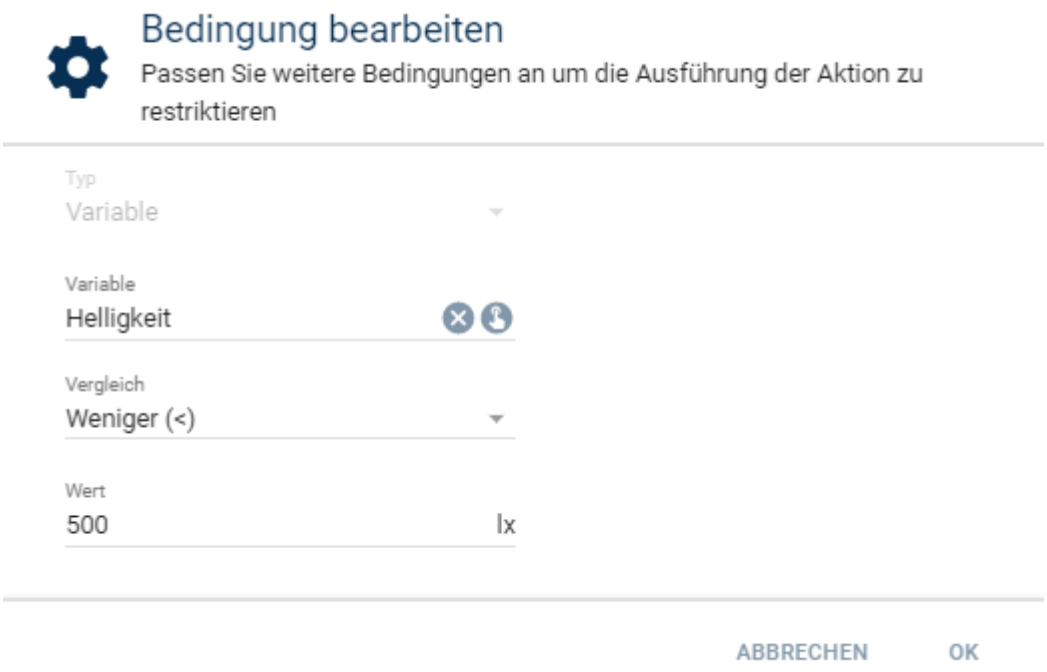
| OPERATOR | ERKLÄRUNG |
|---|---|
| Gleich (=) | Ist erfüllt, wenn der Regelwert genau dem übergebenen Wert entspricht. (Z.B. true = true, 123 = 123) |
| Ungleich (≠) | Ist erfüllt, wenn der Regelwert nicht dem übergeben Wert entspricht. (Z.B. false ≠ true, 123 ≠ 122 od. 124) |
| Größer (>) | Ist erfüllt, wenn der Regelwert echt größer als der übergebene Wert ist. (Z.B. 123 > 122, 09:00 > 08:59) |
| Größer gleich (≥) | Ist erfüllt, wenn der Regelwert größer als oder gleich dem übergebenen Wert ist. (Z.B. 123 ≥ 122 od. 123, 09:00 ≥ 08:59 od. 09:00) |
| Kleiner (<) | Ist erfüllt, wenn der Regelwert echt kleiner als der übergebene Wert ist. (Z.B. 123 < 124, 09:00 < 09:01) |
| Kleiner gleich (≤) | Ist erfüllt, wenn der Regelwert kleiner als oder gleich dem übergebenen Wert ist. (Z.B. 123 ≤ 124 od. 123, 09:00 ≤ 09:01 od. 09:00) |

## BEISPIEL

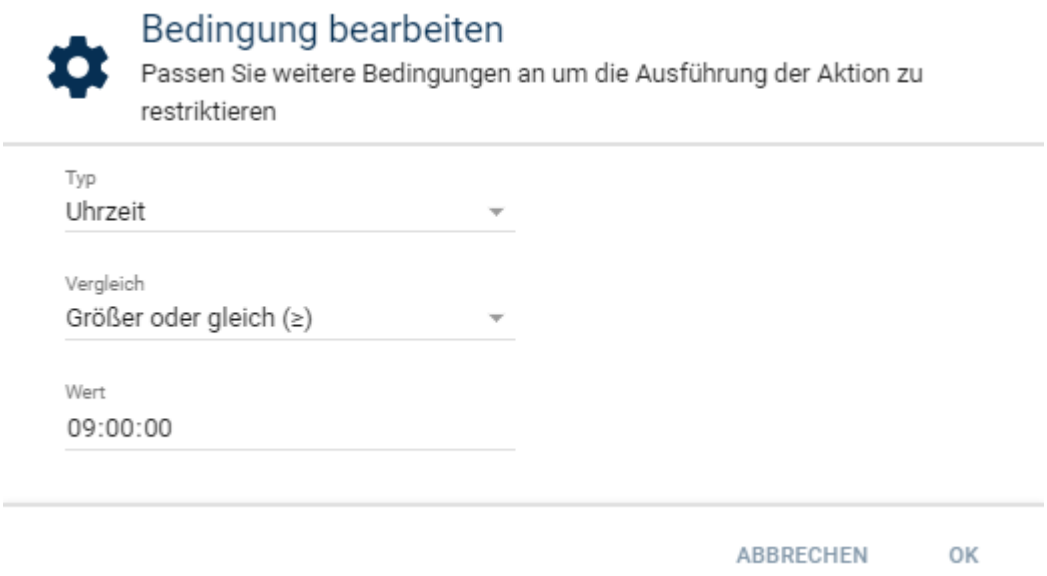
Ein Lampe soll nur angeschaltet werden, wenn der Bewegungsmelder aktiviert wird. Zusätzlich soll dies aber nur geschehen wenn es zu dunkel (Helligkeitswert unter 500) und es zwischen 09:00 und 17:00 Uhr ist.
Zunächst ein ausgelöstes Ereignis hinzufügen. Dabei den Bewegungsmelder als Auslöser bei "Anwesenheit" und die Lampe als Zielinstanz mit dem Befehl den Wert auf "An" setzen.



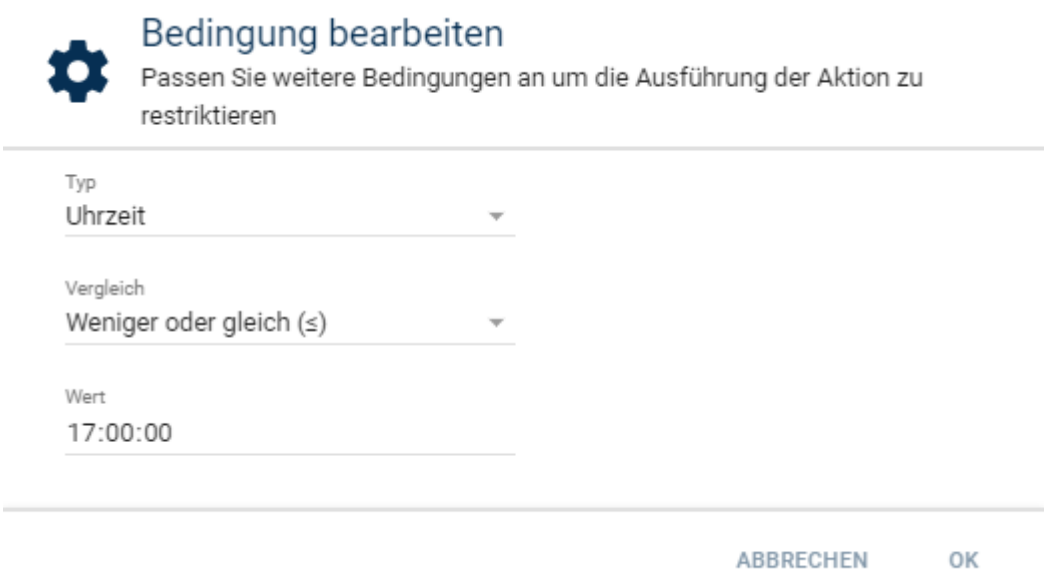

Draufhin unter "Weitere Bedingungen" in der Regelübersicht weitere Regeln über "Hinzufügen" erstellen.
Den Lichtwert "Helligkeit" mit der Regel "kleiner" als 500 einrichten.




Die Uhrzeit mit der Regel "größer gleich" 9 Uhr einrichten.




Die Uhrzeit mit der Regel "kleiner gleich" 17 Uhr einrichten.




In der Übersicht sind nun alle drei Regeln aufgeführt und der Live-Zustand gibt an, ob die momentanen Regeln bei einem Auslösen das Schalten zulassen würden.

# Ausgelöstes Ereignis bearbeiten

Verwenden Sie ausgelöste Ereignisse um auf Aktualisierungen von Variablen zu reagieren

## Aktion

Ziel
Lampe

Aktionstyp
Schalte Instanz

Statusvariable der Instanz
Wert

Aktion
Auf Wert schalten

Wert
An

## Weitere Bedingungen

| Variable | Vergleich | Wert |
|---|---|---|
| Helligkeit | < | 500 lx |
| Aktuelle Uhrzeit | ≥ | 09:00:00 |
| Aktuelle Uhrzeit | ≤ | 17:00:00 |

HINZUFÜGEN

Evaluation
Alle Bedingungen müssen erfüllt sein

Live Zustand: Aktion wird ausgeführt

Ereignis aktiv?

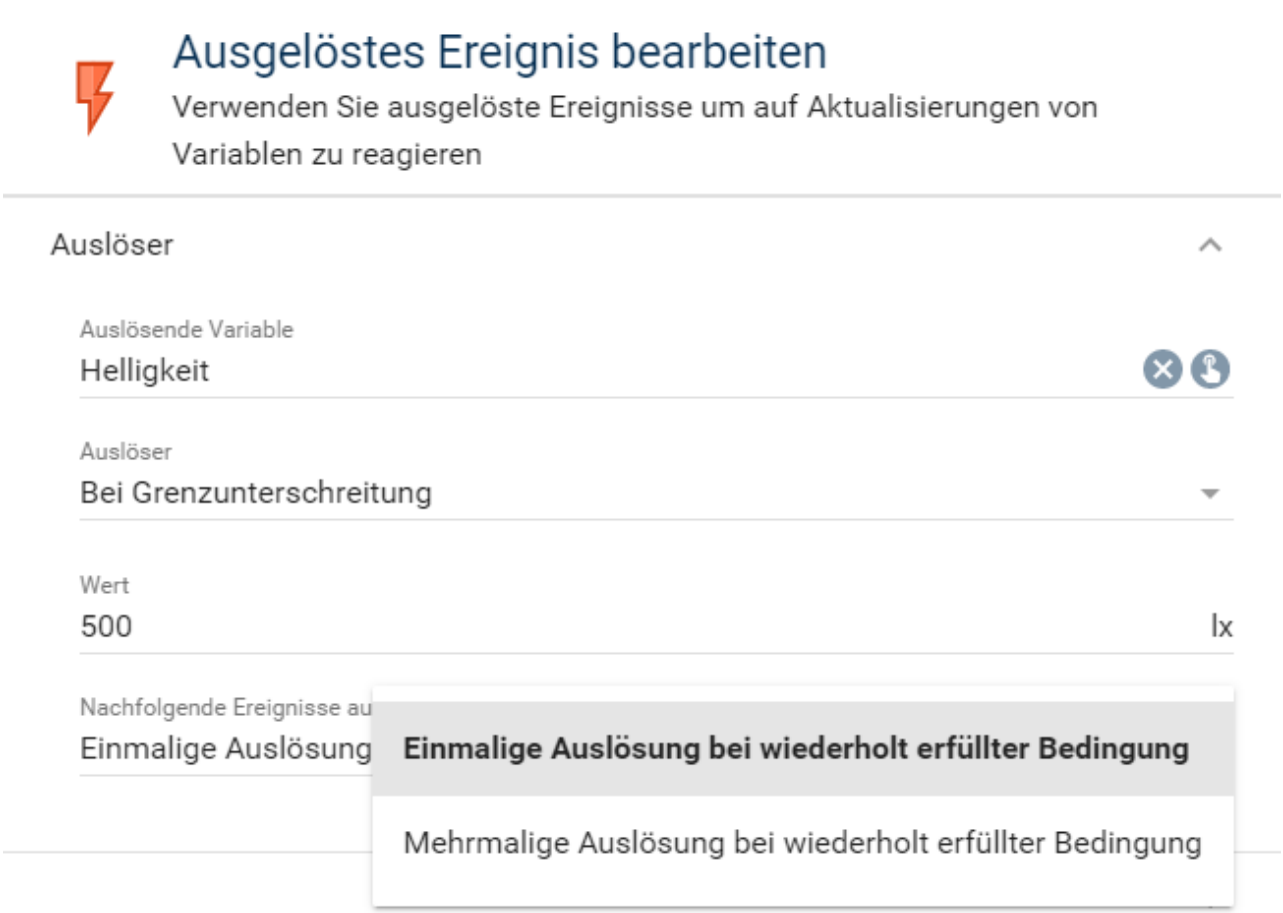
ABBRECHEN OK

# Ausgelöst

Dieser Ereignistyp löst durch Aktualisierung einer angebundenen Variable aus, unabhängig von dem Zeitpunkt der Auslösung.
Dazu stehen 5 Auslöser zur Auswahl.

Skripte, die durch das Variablenereignis ausgeführt werden, beinhalten diese Systemvariablen

## AUSLÖSER



| ART DES AUSLÖSERS | BESCHREIBUNG |
| --- | --- |
| Bei Änderung: | Das Ereignis löst bei jeglicher Werteänderung der gewählten Variable aus. |
| Bei Aktualisierung: | Das Ereignis löst bei jedem empfangenen Variablenwert aus. Dies gilt auch wenn der empfangene und der bereits vorhandene Wert identisch sind. |
| Bei Grenz- unter/überschreitung: (Option “Nachfolgende Ereignisse ausführen”) | Das Ereignis löst aus, wenn der Wert der gewählten Variable einen gesetzten Wert über- oder unterschreitet. Der "Wert" setzt den Grenzwert bei dem Auslösung erfolgt. |
| Bei bestimmten Wert: (Option “Nachfolgende Ereignisse ausführen”) | Das Ereignis löst aus, wenn die gewählte Variable genau einen bestimmten Wert erreicht. Der "Wert" setzt den genauen Wert bei dem Auslösung erfolgt. |

## OPTIONEN

### NACHFOLGENDE EREIGNISSE AUSFÜHREN

**Einmalige Auslösung bei wiederholt erfüllter Bedingung:** Das Ereignis löst einmalig aus. Erst wenn die Bedingung im Auslöser zwischenzeitlich nicht erfüllt war löst ein erneutes über- / unterschreiten der Grenze oder erreichen des bestimmten Wertes das Ereignis wieder aus.
**Mehrmalige Auslösung bei wiederholt erfüllter Bedingung:** Das Ereignis löst bei jeder Aktualisierung aus, solange die Bedingung erfüllt ist.

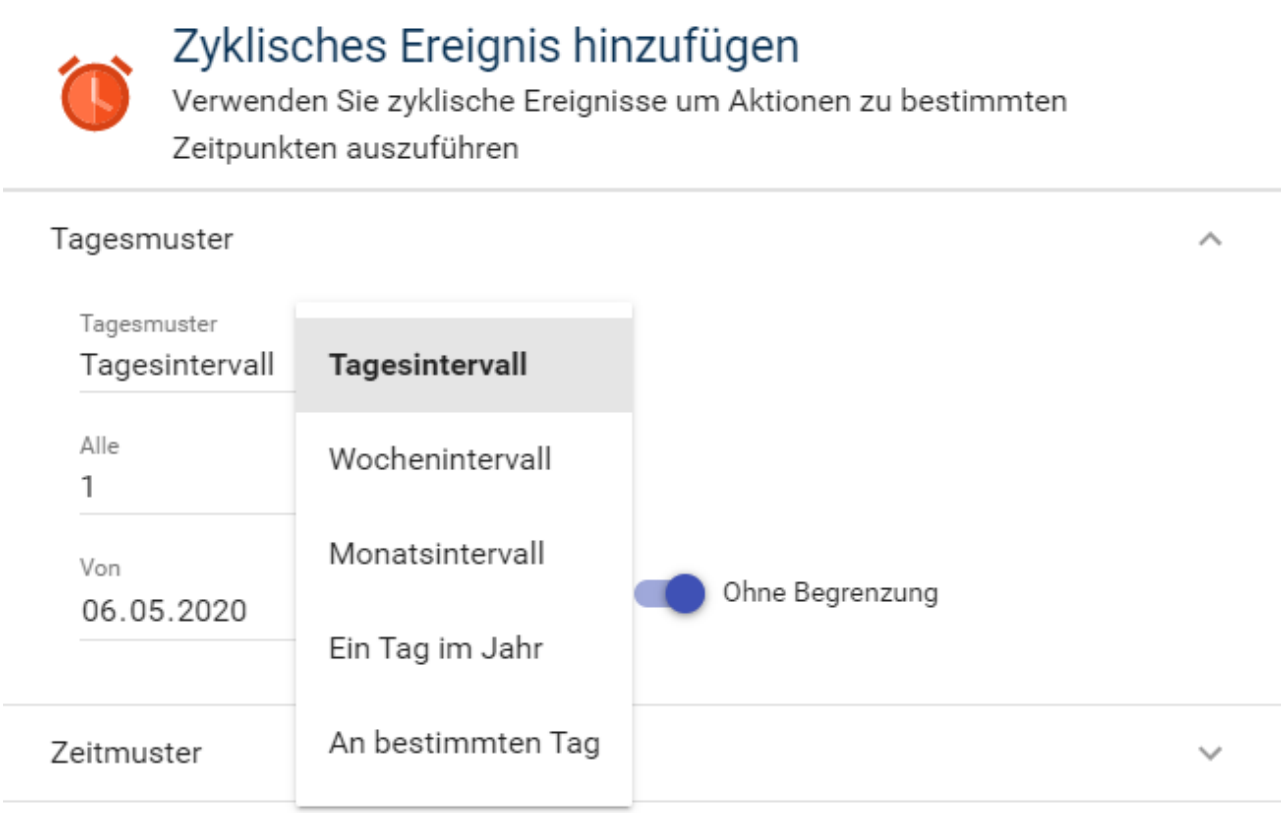
# Zyklisch

Dieser Ereignistyp wird zu einem bestimmten Zeitpunkt ausgelöst und kann sich zyklisch wiederholen.
Der Zeitpunkt wird durch die Kombination von “Tagessmuster” und “Zeitmuster” definiert.

Skripte, die durch das zyklische Ereignis ausgeführt werden, beinhalten diese Systemvariablen

## TAGESSMUSTER

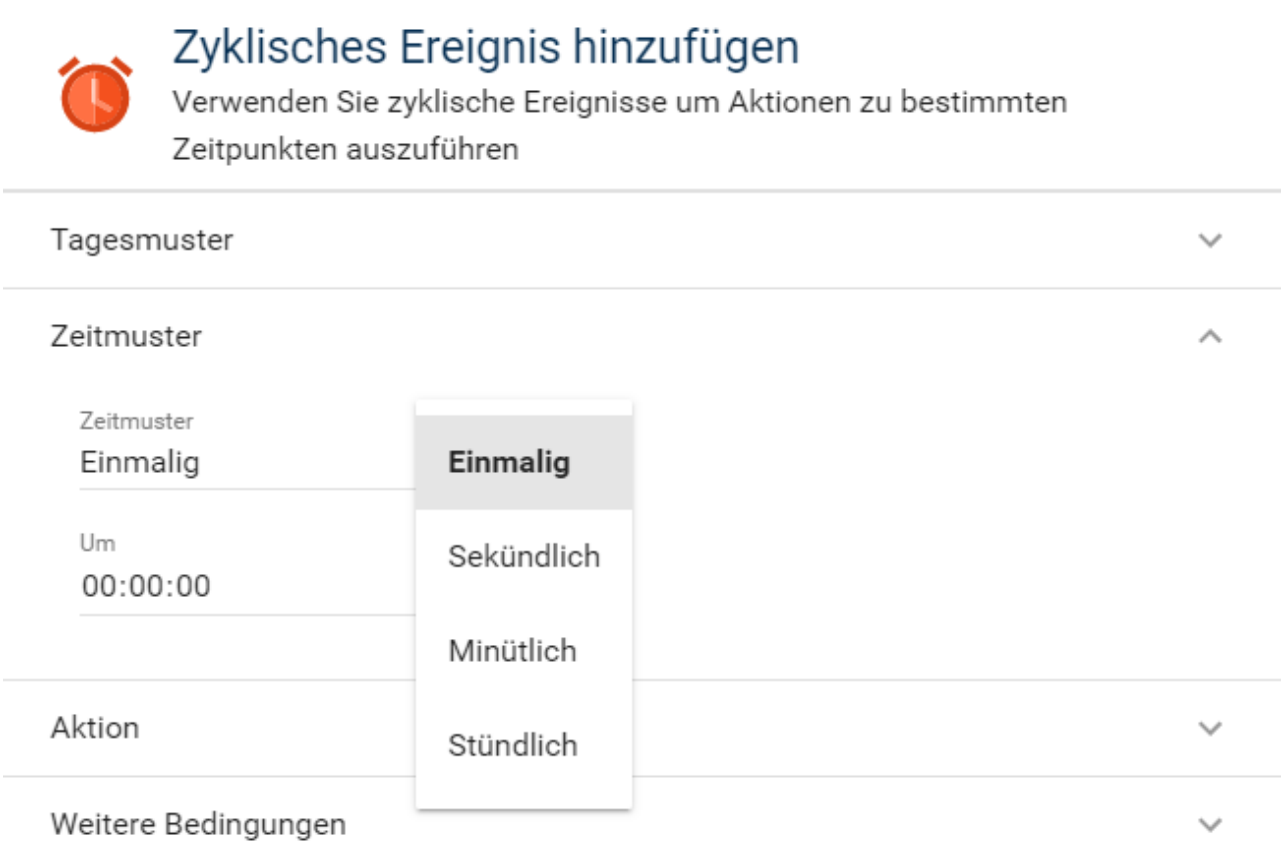
Hat 5 Optionen



| TAGESSMUSTER | BESCHREIBUNG |
|---|---|
| Tagesintervall | Gibt den Tageintervall an (1 = jeden Tag; 2 = jeden zweiten Tag ; …) in denen das Ereignis ausgelöst werden soll. |
| Wochenintervall | Gibt den Wochenintervall an (1 = jede Woche; 2 = jede zweite Woche; …) und an welchen Tagen einer Woche das Ereignis ausgelöst werden soll. |
| Monatsintervall | Gibt den Monatsintervall an (1 = jeden Monat; 2 = jeden zweiten Monat; …) und an welchem Tag im Monat das Ereignis ausgelöst werden soll. |
| Ein Tag im Jahr | Gibt einen bestimmten Tag im Jahr an, an dem das Ereignis ausgelöst werden soll. |
| An bestimmten Tag | Gibt ein bestimmtes Datum an, an dem das Ereignis ausgelöst wird. |

## ZEITMUSTER

Hat 4 Optionen




| ZEITMUSTER | BESCHREIBUNG |
|---|---|
| Einmalig | Gibt eine bestimmte Uhrzeit an dem das Ereignis ausgelöst werden soll. |
| Sekündlich | Gibt den Sekundenintervall an, in dem das Ereignis ausgelöst werden soll. |
| Menütlich | Gibt den Minutenintervall an, in dem das Ereignis ausgelöst werden soll. |
| Stündlich | Gibt den Stundenintervall an, in dem das Ereignis ausgelöst werden soll. |

## SONDEROPTIONEN

### VON/BIS

Start und Endzeitpunkt bestimmbar

> Der "Bis"-Wert muss größer sein als "Von"-Wert

| EINSTELLMÖGLICHKEITEN | BESCHREIBUNG |
|---|---|
| Bei Tagessmuster | Gibt den Datumszeitraum an in der das Ereignis aktiv ist. |
| Bei Zeitmuster | Gibt den Tageszeitraum an in der das Ereignis aktiv ist. |
| Von | Startzeitpunkt = Gibt an ab wann das Intervall anfängt zu zählen. |
| Bis | Endzeitpunkt = Letzter Zeitpunkt bei dem ein Ereignis auslösen kann. |
| "Ohne Begrenzung" | Kein Endzeitpunkt. Wenn deaktiviert, wird die "Bis" Option angezeigt. |

### WEITERE BEDINGUNGEN

Die Funktion der weiteren Bedinungen kann unter Bedingungen nachgelesen werden.

# BEISPIELE

## JEDEN FREITAG UM 09:00UHR:

Tagessmuster = Wöchentlich mit Haken bei Freitag
Zeitmuster = einmalig mit 09:00:00

## ALLE 5 MINUTEN IM ZEITRAUM VON 12:02:10 BIS 16:00:00 UHR

Tagessmuster = Täglich mit dem Wert 1
Zeitmuster = Minütlich mit dem Wert 5
Von: 12:02:10 Bis: 16:00:00
Kein Haken bei “Ohne Begrenzung”
Erste Ausführung Freitags um 12:02:10
Zweite Ausführung Freitags um 12:07:10
...
--fortlaufend bis-->
...
Letzte Ausführung Freitags um 15:57:10

## TÄGLICH UND DORT MINÜTLICH IM ZEITRAUM VON 23:00 BIS 06:00

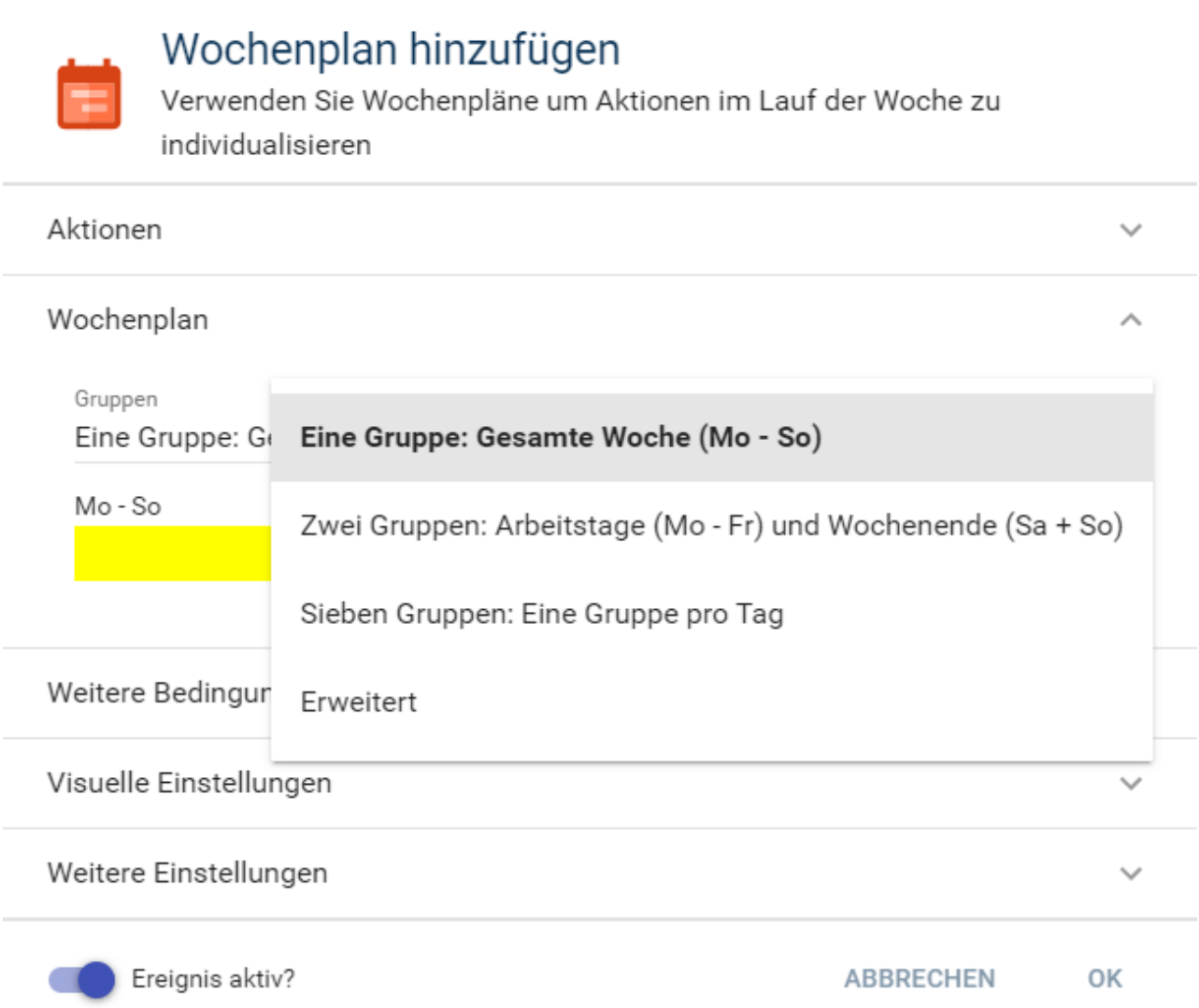
Es müssen 2 Timer erstellt werden.
Timer 1:
Tagessmuster = Täglich mit dem Wert 1
Zeitmuster = Minütlich mit dem Wert 1
Kein Haken bei "Ohne Begrenzung"
Von: 00:00:00 Bis: 06:00:00
Timer 2:
Tagessmuster = Täglich mit dem Wert 1
Zeitmuster = Minütlich mit dem Wert 1
Kein Haken bei "Ohne Begrenzung"
Von: 23:00:00 Bis: 23:59:59

# Wochenplan

Der Wochenplan ist ein grafisches Werkzeug zur Konfiguration von wöchentlichen Abläufen.
In der Verwaltungskonsole erstellt man den Wochenplan, den Gruppentyp, die Aktionen und die Anbindungsart.
Innerhalb der Visualisierung "WebFront" und Wochenplankonfiguration können die Aktionen und somit Aktionswechsel definiert werden.

## GRUPPENTYPEN

Hat 4 Optionen.



| TYP | BESCHREIBUNG |
| --- | --- |
| Eine Gruppe:<br>Gesamte Woche (Mo - So) | Alle Wochentage sind zu einer Gruppe zusammengefasst. |
| Zwei Gruppen:<br>Arbeitstage (Mo - Fr) und<br>Wochenende (Sa + So) | Unterteilt die Woche in zwei Gruppen um somit die Tage Mo-Fr und Sa-So unabhängig voneinander konfigurieren zu können. |
| Sieben Gruppen:<br>Eine Gruppe pro Tag | Jeder Wochentag kann einzeln konfiguriert werden. |
| Erweitert | Die Wochentage können in beliebig kombiniert werden. Jede Kombination wird in einer Gruppe zusammengefasst und konfiguriert werden. |

## AKTIONEN

Der Wochenplan bietet Aktionen passend zum ausgewählten Ziel.
Es müssen mindestens zwei Wochenplanaktionen hinzugefügt werden.
Als Teil der Wochenplanaktion wird eine Aktion für das Ziel definiert.

### WOCHENPLANAKTION HINZUFÜGEN

Um eine Wochenplanaktion hinzuzufügen muss auf "Hinzufügen" geklickt werden.
Es kann der Wochenplanaktion ein Name, eine Farbe und eine Aktion zugewiesen werden.




## KONFIGURATION DER WOCHENPLANAKTIONEN

Die Zeiträume von Aktionen können innerhalb der Verwaltungskonsole und WebFront konfiguriert werden.
Dabei gilt innerhalb der grafischen Darstellung:
**Einfaches Klicken** bearbeitet einen vorhanden Zustand.
**Klicken & Ziehen** erstellt einen neuen Zustand.

### IN DER VERWALTUNGSKONSOLE



## Wochenplan bearbeiten

Verwenden Sie Wochenpläne um Aktionen im Lauf der Woche zu individualisieren

Aktionen

Wochenplan

Gruppen

Zwei Gruppen: Arbeitstage (Mo - Fr) und Wochenende (Sa + So)

Mo - Fr



Sa + So



Weitere Bedingungen

Visuelle Einstellungen

Weitere Einstellungen

Ereignis aktiv?

ABBRECHEN OK

### IM WEBFRONT





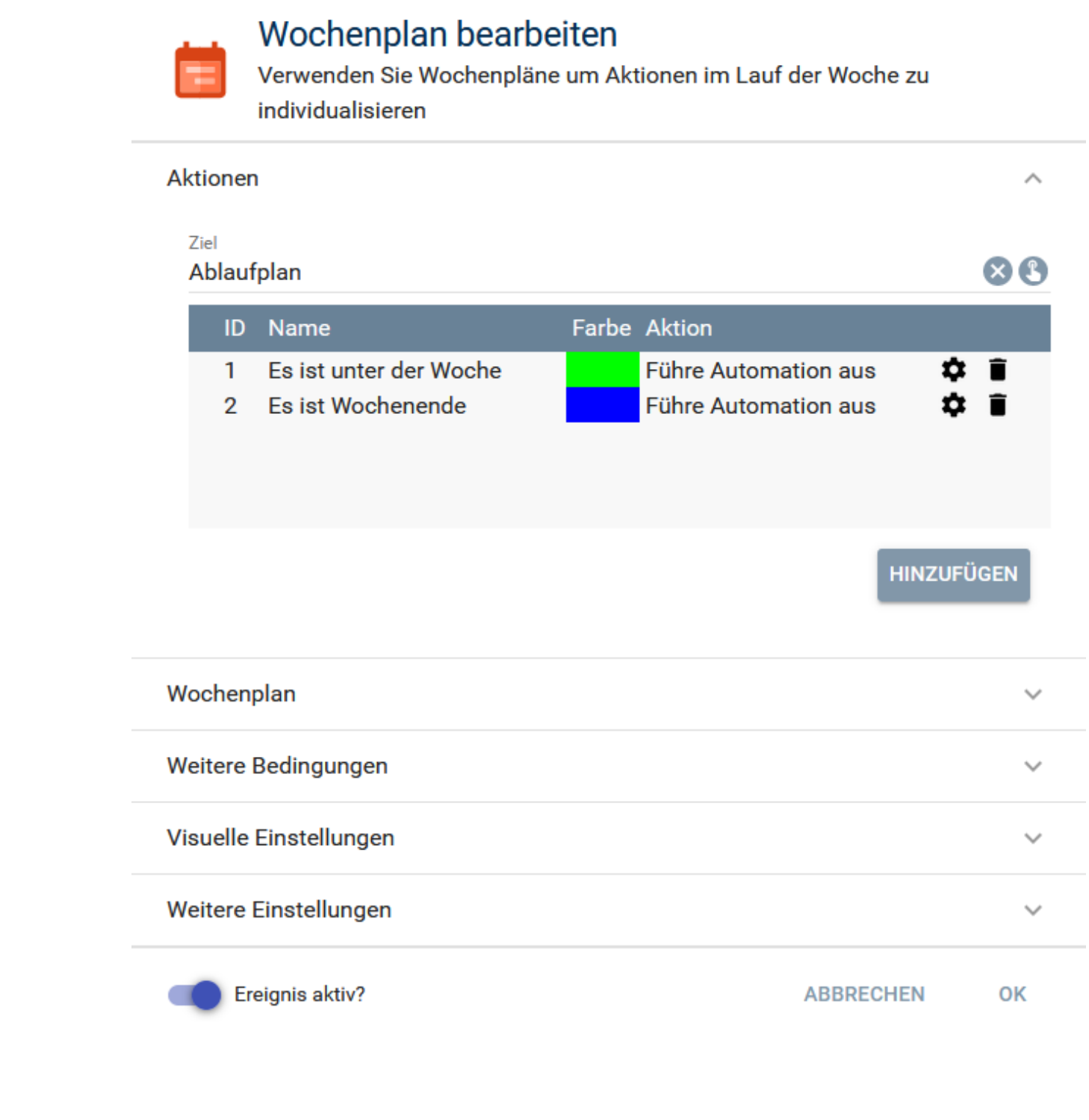


Nur bei einem Aktionswechsel wird die ausgewählte Aktion ausgeführt.

Wichtiger Hinweis:
Zuvor gesetzte Aktionen werden durch einen Aktionswechsel nicht aufgehoben.
D.h. Wenn man in einer Aktion das Licht eingeschaltet hat, wird bei einem Wechsel in die nächste Aktion nicht automatisch das Licht ausgeschaltet, außer man beschreibt dies explizit.

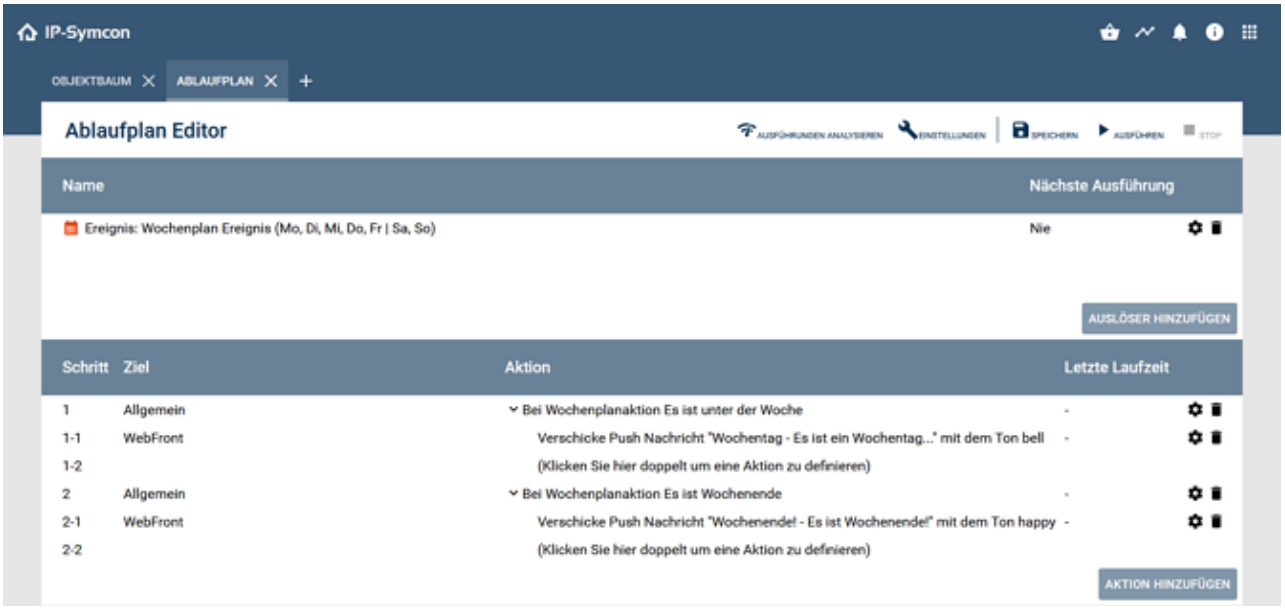
## AN ABLAUFPLAN ANGEBUNDEN

Wird auf einem Ablaufplan die Aktion "Führe Automation aus" gewählt, so kann über die Aktion "Bei Wochenplanaktion" bei einer bestimmten Wochenplanaktion eine Reihe von Aktionen ausgeführt werden. Wird der erste Wochenplan einem Ablaufplan als Auslöser hinzugefügt, so werden die "Bei Wochenplanaktion"-Aktionen automatisch hinzugefügt.



## BEISPIEL MIT ABLAUFPLAN

Dies ist ein Ablaufplanbeispiel, welches über einen Wochenplan ausgeführt wird.




### AN PHP-SKRIPT ANGEBUNDEN

Wird auf einem PHP-Skript die Aktion "Führe Automation aus" gewählt, so steht die ID der Wochenplanaktion als Systemvariable $_IPS['ACTION'] zur Verfügung.

## BEISPIEL MIT SKRIPT

Dies ist ein PHP-Skript-Beispiel, welches über einen Wochenplan ausgeführt wird.

```
//switch über die ID's der Aktionen
switch ($_IPS['ACTION']) {
case 1: //ID 1
SetValueBoolean(39540 /*[Testumgebung\Arbeitstag]*/, true);
echo "Hallo Welt, Unter der Woche";
break;
case 2: //ID 2
SetValueBoolean(39540 /*[Testumgebung\Arbeitstag]*/, false);
echo "Hallo Welt, Yay Wochenende!!";
break;
}
```

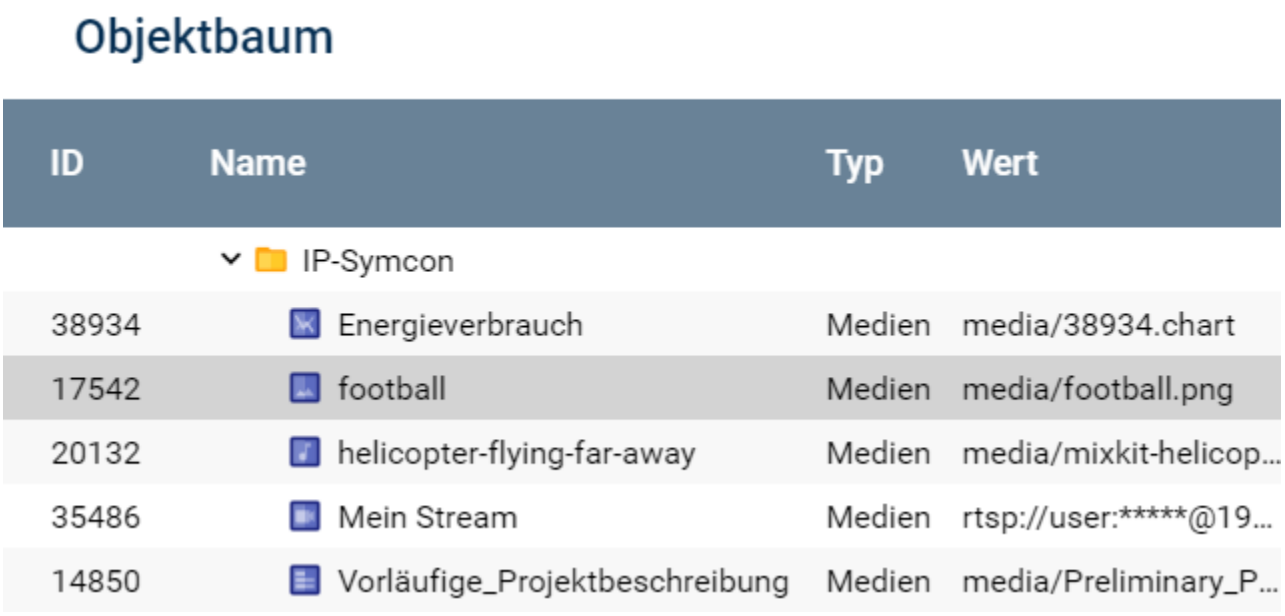
### TIPP FÜR EXPERTEN

Um herauszufinden, welche Wochenplanaktion momentan aktiv ist, kann dies über ein Skript abgefragt werden.

```
$e = IPS_GetEvent($id);
$actionID = false;
//Durch alle Gruppen gehen
foreach($e['ScheduleGroups'] as $g) {
//Überprüfen ob die Gruppe für heute zuständig ist
if($g['Days'] & date("N") > 0) {
//Aktuellen Schaltpunkt suchen. Wir nutzen die Eigenschaft, dass die
Schaltpunkte immer aufsteigend sortiert sind.
foreach($g['Points'] as $p) {
if(date("H") * 3600 + date("i") * 60 + date("s") >= $p['Start']
['Hour'] * 3600 + $p['Start']['Minute'] * 60 + $p['Start']
['Second']) {
$actionID = $p['ActionID'];
} else {
break; //Sobald wir drüber sind, können wir abbrechen.
}
}
break; //Sobald wir unseren Tag gefunden haben, können wir die
Schleife abbrechen. Jeder Tag darf nur in genau einer Gruppe sein.
}
}
var_dump($actionID);
```

# Medien

Von IP-Symcon unterstützte Medien sind Bild und Ton-Dateien, Diagramme, Dokumente, Streams sowie Views von IPSStudio.




## MEDIEN HINZUFÜGEN

Die Anleitung wie die jeweiligen Medien eingerichtet werden befinden sich in den jeweiligen Unterkategorien.
Es können folgende Medien eingebunden werden.
Bild und Ton
Diagramme
Dokumente
Streams
Views von IPSStudio

# Bild/Ton

## BILD- ODER TONDATEI HINZUFÜGEN

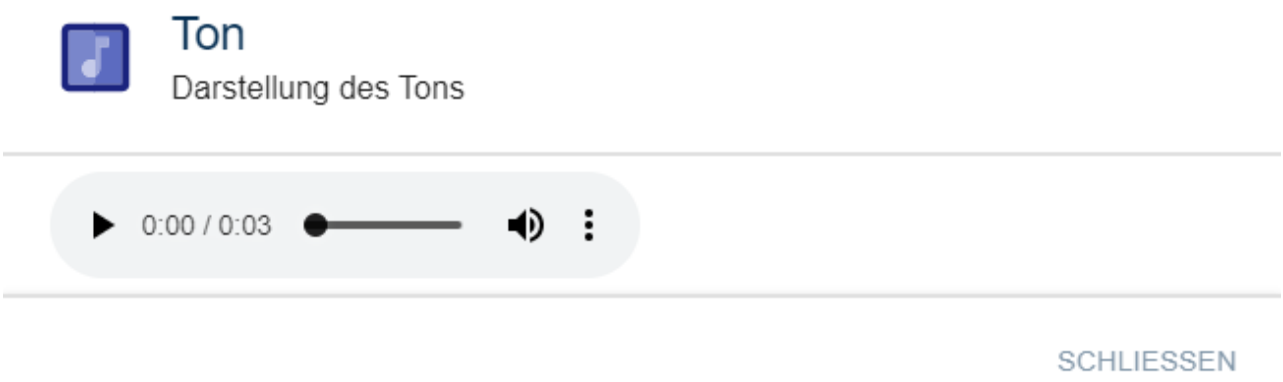
Um eine Bild- oder Ton-Datei dem Media-Pool hinzuzufügen, muss der Dialog "Objekt hinzufügen -> Medien -> Bild oder Ton" genutzt werden.
Es kann eine beliebige Ton- oder Bilddatei von Festplatte, Netzwerk, Internet usw. ausgewählt werden, welche dann von IP-Symcon in den Media-Pool kopiert wird.






Hinzugefügte Bild/Ton-Objekte können im Objektbaum eingesehen werden.
Eine Vorschau kann mit Doppelklick auf das Medienobjekt im Objektbaum angezeigt werden.






## ERLAUBTE DATEIENDUNGEN

| ENDUNG | AB VERSION |
|---|---|
| .jpg | 3.4 |
| .jpeg | 3.4 |
| .gif | 3.4 |
| .png | 3.4 |
| .ico | 3.4 |
| .webp | 8.1 |

Es können alle Dateien hinzugefügt werden - die korrekte Darstellung in der Kachel Visualisierung wird aber nur für diese Typen garantiert

# Diagramme

In einem Diagramm können ein oder mehrere Variablen als Graphen dargestellt werden. Voraussetzung hierfür ist, dass zuvor bei den betreffenden Variablen das Logging aktiviert wurde. Das Einrichten vom Logging wird im Archiv Control beschrieben.

## EINRICHTUNG IN IP-SYMCON

Über den Rechtsklick-Dialog "Objekt hinzufügen" -> "Medien" -> "Diagramm" kann ein neues Diagramm an gewünschter Position angelegt werden. Im Folgenden werden die Konfigurationsmöglichkeiten erklärt. Die dort ausgewählten (geloggten) Variablen dienen als Datenquelle für die einzelnen Graphen.

### DIAGRAMM BEARBEITEN

Allgemeine Eigenschaften des Diagramms.

| OPTION | BESCHREIBUNG |
|---|---|
| Hinzufügen | Hinzufügen der einzelnen Variablen -> Öffnet den "Graph konfigurieren" Dialog. Bei schon verhandenen Graphen wird dieser über das Zahnrad in der Liste aufgerufen |
| Name | Der Name des Diagramm |
| Ort | Auswahl des Speicherorts für das Diagramm |
| Typ | Auswahl der Darstellung als Linien-, Balken- oder Bool-Diagramm |
| Visuelle Einstellungen | Hier könnnen ein Icon ausgewählt, die Anzeige und Aktivierung konfiguriert werden |
| Weitere Einstellungen | Bietet die Möglichkeit eine Beschreibung anzugeben |

### GRAPH KONFIGURIEREN

Ein Graph repräsentiert ein Datenset für das Diagramm.

| FELD | BESCHREIBUNG |
|---|---|
| Achsenprofil | Auswahl des Variablenprofils. Die darin angegeben Min- und Max-Werte dienen der Skalierung der Y-Achse. (Auswahlmöglichkeit entfällt bei Bool-Diagramm) |
| Achsenseite | Auswahl ob die Beschriftung für diesen Graphen Links oder Rechts sein soll. (Auswahlmöglichkeit entfällt bei Bool-Diagramm) |
| Füllfarbe | Auswahl der gewünschten Füllfarbe - Transparenz möglich |
| Linienfarbe | Auswahl der gewünschten Linienfarbe |
| Titel | Aussagekräftiger Titel des Graphen, welcher in der Legende dargestellt wird. Bei einem leeren Titel wird der Name der Variable genutzt |
| Variable | Auswahl der darzustellenden Variablen |
| Zeitversatz | Verschiebung der jeweiligen Zeitachse um die eingestellte Zeiteinheit. (Beispiele siehe unten) |

## DARSTELLUNG

**Darstellung als Popup Graph**
Um einen Popup Graph anzuzeigen muss das Diagramm einfach in die gewünschte Kategorie im Objektbaum verschoben werden.
**Darstellung als Vollbild Graph**
Um einen Vollbild Graphen darzustellen muss das Diagramm im WebFront als Graphenelement hinzugefügt werden.

## DARSTELLUNGSARTEN

Aktivierbare Darstellungsarten, welche innerhalb des Graphen geschaltet werden können.

| ABKÜRZUNG | BEDEUTUNG | BESCHREIBUNG |
|---|---|---|
| CON | Kontinuierlich | Die X-Achse wird so angepasst, dass der aktuellste Wert an der rechten Graphengrenze dargestellt wird. Dies dient dazu die größtmögliche Menge an aktuellen Daten darzustellen |
| CSV | Datensätze | Zeigt die verwendeten Datensätze und Zeitstempel für die jeweiligen Werte an |
| DYN | Dynamisch | Wenn aktiviert: Wird der dargestellte Ausschnitt an die Mininmal-/Maximalwerte des Graphen auf der Y-Achse angepasst.<br>Wenn deaktiviert: Werden die Grenzen des eingestellten Variablenprofils genutzt |
| HD | Hohe Dichte | Wenn aktiviert: Wird eine höhere Genauigkeit der Durchschnittswerte zur Darstellung des Graphen genutzt. Siehe Tabelle **Durchschnittswertgrundlage** |
| Legende | Legende | (nur Diagramme) De-/Aktiviert die Darstellung der Legende |
| MIN/MAX | Min/Max | (nur automatische Variablendarstellung) Es werden 2 zusätzliche Graphen mit den Minimal-/Maximalwerten dargestellt |
| RAW | Rohdaten | Wenn aktiviert: Werden die Rohdaten dargestellt. Wenn deaktiviert: Werden die aggregierten Werte genutzt |

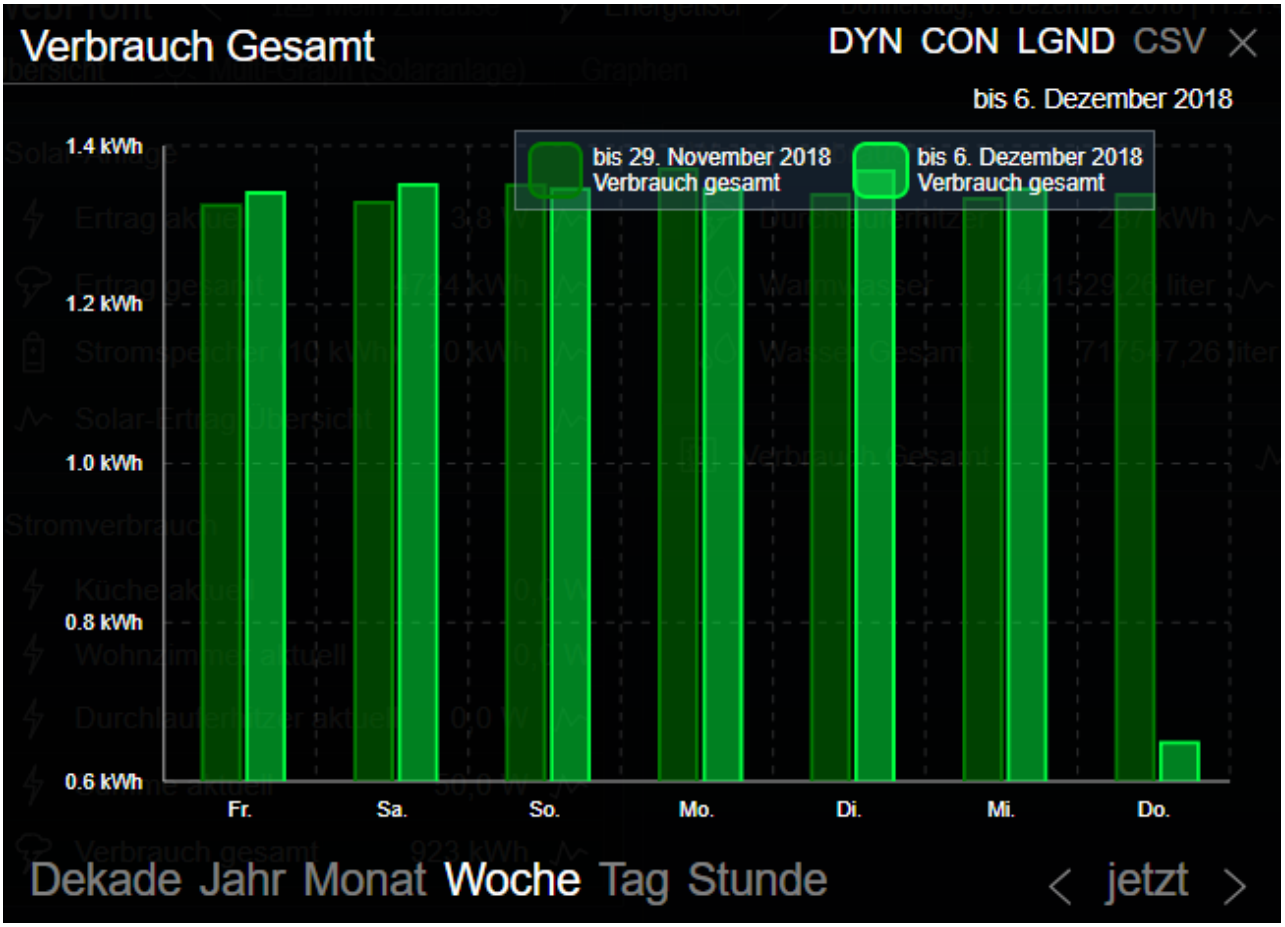
**Durchschnittswertgrundlage**

| ZEITRAUM | STANDARD | HOHE DICHTE (HD) |
|---|---|---|
| Dekade | Jahre | Monate |
| Jahr | Monate | Tage |
| Monat | Tage | Stunden |
| Woche | Tage | Stunden |
| Tag | Stunden | 5-Minuten |
| Stunde | 5-Minuten | 1-Minuten |

## BEISPIEL 1 (TYP: BALKEN)

Dieses Beispiel zeigt die Gegenüberstellung von Verbrauchswerten einer Steckdose. Hier wurde zweimal die selbe Variable ausgewählt, wobei eine (hier dunkelgrün) den "Zeitversatz" von -1 hat.

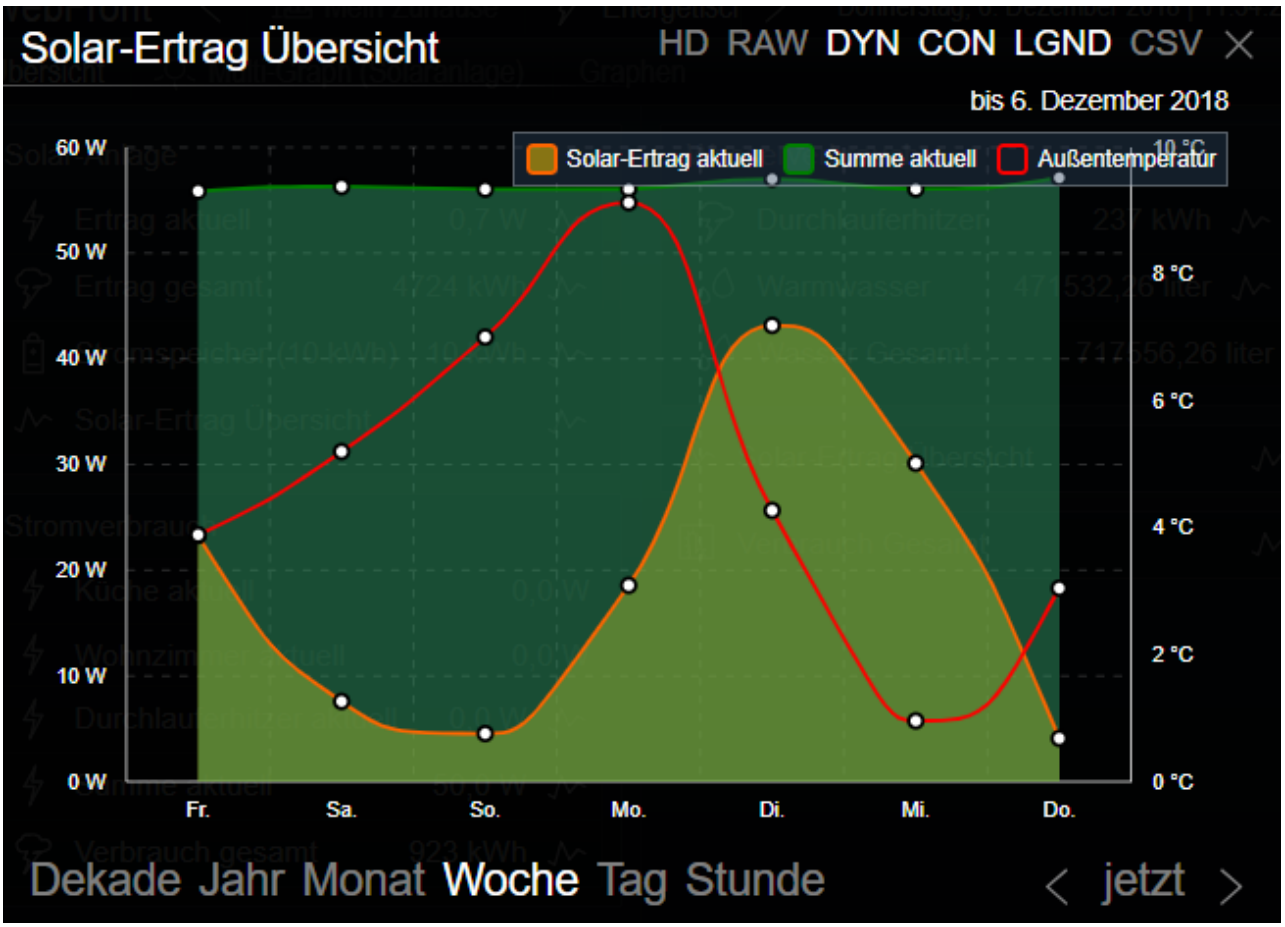
In der Wochen-Darstellung sieht man einen Vergleich zur Vorwoche; in der Jahres-Darstellung wäre demzufolge ein Vergleich zum Vorjahr.






## BEISPIEL 2 (TYP: LINIEN)

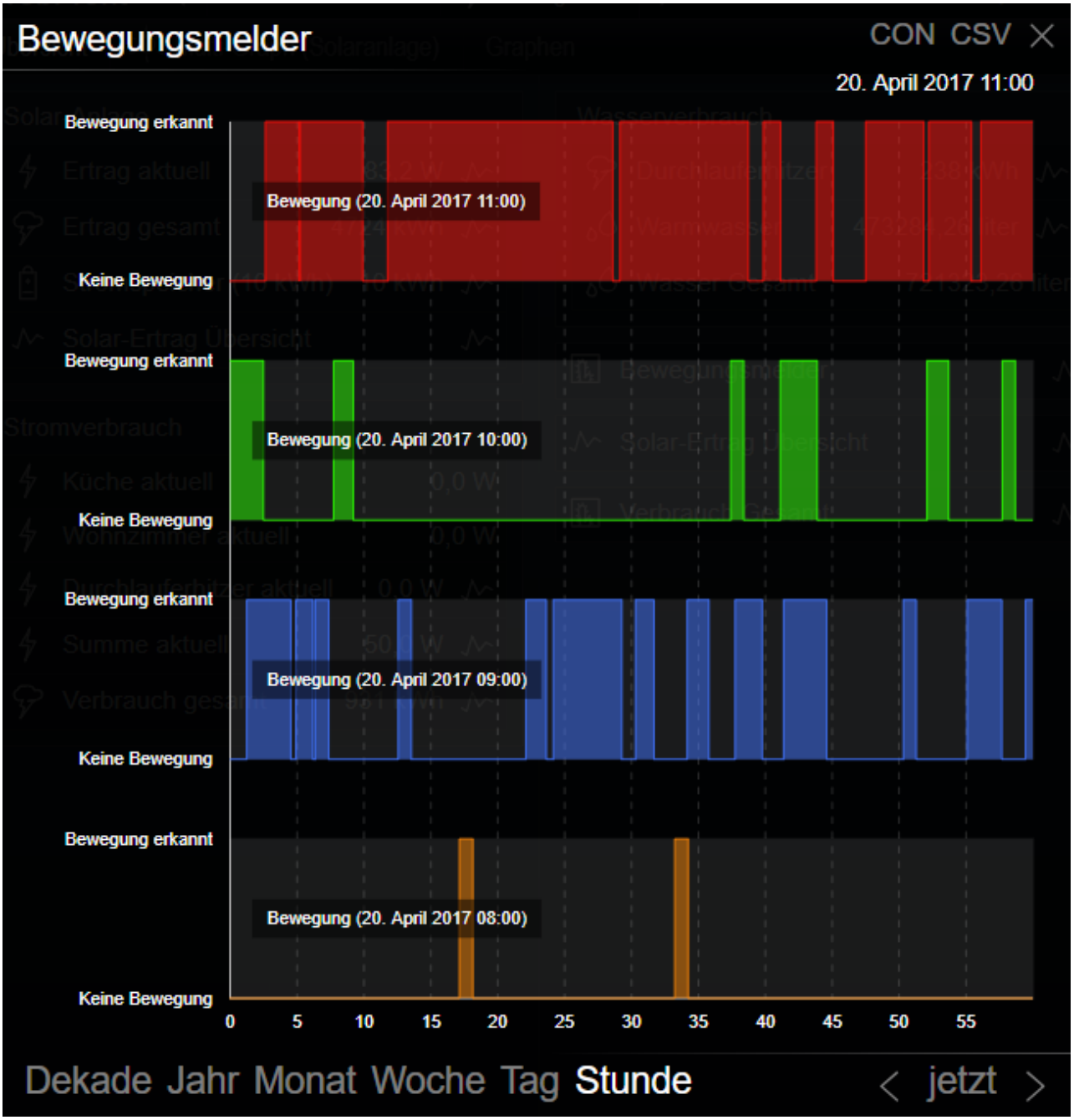
Dieses Beispiel zeigt den Ertrag einer Solaranlage, sowie ein Summenwert für den aktuellen Verbrauch. Die Werte für diese beiden Graphen sind linksseitig ablesbar. Zusätzlich wurde noch die Außentemperatur mit integriert, dessen Werte an der rechten Seite abzulesen sind.


Solar-Ertrag Übersicht
HD RAW DYN CON LGND CSV
bis 6. Dezember 2018
Solar-Ertrag aktuell
Summe aktuell
Außentemperatur
60 W
50 W
40 W
30 W
20 W
10 W
0 W
10 °C
8 °C
6 °C
4 °C
2 °C
0 °C
Fr.
Sa.
So.
Mo.
Di.
Mi.
Do.
Dekade Jahr Monat Woche Tag Stunde
jetzt




## BEISPIEL 3 (TYP: BOOL)

Dieses Beispiel zeigt einen Bool-Diagramm, welches den Zustand von einem Bewegungsmelder im Verlaufe von 4 aufeinanderfolgenden Stunden darstellt.


Bewegungsmelder
CON CSV
20. April 2017 11:00
Bewegung erkannt
Bewegung (20. April 2017 11:00)
Keine Bewegung
Bewegung erkannt
Bewegung (20. April 2017 10:00)
Keine Bewegung
Bewegung erkannt
Bewegung (20. April 2017 09:00)
Keine Bewegung
Bewegung erkannt
Bewegung (20. April 2017 08:00)
Keine Bewegung
0
5
10
15
20
25
30
35
40
45
50
55
Dekade Jahr Monat Woche Tag Stunde
jetzt

# Diagramm bearbeiten

Benutzen Sie Charts um den Verlauf Ihrer Variablen zu visualisieren

Name

Bewegungsmelder

Ort

Eltako FBH65TFB (Helligkeit/Bewegu

Typ

Bool

| Variable | Titel | Profil | Seite | Zeitver... | Farbe |
|---|---|---|---|---|---|
| Bewegun... | Bewegung | (Automatisch) | Links | 0 | |
| Bewegun... | Bewegung | (Automatisch) | Links | -1 | |
| Bewegun... | Bewegung | (Automatisch) | Links | -2 | |
| Bewegun... | Bewegung | (Automatisch) | Links | -3 | |

HINZUFÜGEN

Visuelle Einstellungen

Weitere Einstellungen

ABBRECHEN OK

# Dokumente

Benötigt: IP-Symcon >= 4.3

## MEDIEN-DOKUMENT HINZUFÜGEN

Über den Dialog "Objekt hinzufügen -> Medien -> Dokument" kann eine neue Mediendatei vom Typ Dokument hinzugefügt werden.
Es kann eine neue Datei erstellt oder eine vorhande von Festplatte, Netzwerk usw. ausgewählt werden, welche dann von IP-Symcon in den Media-Pool kopiert wird.
Hinzugefügte Dokumente können im Objektbaum unter "Media Dateien" eingesehen und geöffnet werden.




### ERLAUBTE DATEIENDUNGEN

| ENDUNG | AB VERSION |
|---|---|
| .doc | 4.3 |
| .docx | 5.2 |
| .pdf | 4.3 |
| .txt | 4.3 |
| .xls | 4.3 |
| .xlsx | 5.2 |



Es können alle Dateien hinzugefügt werden, aber ein späterer Support via WebFront wird nur für die angegeben Dateitypen vorhanden sein

# Streams

## MEDIEN-STREAM HINZUFÜGEN

Über den speziellen Medientyp Stream können MJPEG und RTSP Stream zum WebFront hinzugefügt werden. Weitere Informationen und die Adresse kann der Dokumentation der genutzten Web-Kamera entnommen werden. Die Größe des Streams kann ebenfalls über die Konfiguration der Web-Kamera parametriert werden. Innerhalb des Dialogs von IP-Symcon muss die vollständige Adresse inklusive des Benutzernamen/Passworts eingetragen werden.



Diese Funktion steht nicht im "Internet Explorer" zur Verfügung.



RTSP Streams können aktuell in Chrome, Firefox, Opera, Edge, Android (inkl. Apps ab Version 5.1), Safari (ab Version 5.5) und iOS (inkl. Apps ab Version 5.5) dargestellt werden.



RTSP Streams müssen h264 codiert sein. Da IP-Symcon bei RTSP Streams als Verteiler agiert, werden hier Limitierungen in Abhängigkeit der Edition angewandt.

# Stream hinzufügen

Verwenden Sie Streams um Ihre RTSP/MJPEG Videostreams zu verwalten

Name
WebCam Haustür

Ort
IP-Symcon

Protokoll
RTSP

Stream
rtsp://192.168.0.25

Benutzername
admin

Passwort
•••••

Visuelle Einstellungen

Weitere Einstellungen

ABBRECHEN OK

# Links

## BESCHREIBUNG

Links sind verknüpfte Objekte und dienen dazu, bereits vorhandene Objekte an anderer Stelle im Objektbaum wiederzuspiegeln. Dies geschieht ohne vorhandene Strukturen oder Sortierungen zu ändern.
Es werden alle Eigenschaften des ursprünglichen Objekt wiedergespiegelt.
Einzig folgende Eigenschaften können bei Links geändert werden:

- Name
- Icon für das WebFront
- Sichtbarkeitseinstellung für das WebFront.

Verlinkt werden können alle Objekte aus dem Objektbaum:

- Kategorien
- Instanzen
- Variablen
- Skripte
- Ereignisse
- Medien



Links von Links sind nicht möglich. Mehrfache Links von ein und demgleichen Objekt allerdings schon.

## ANWENDUNG

Links dienen primär dazu im WebFront Objekte an der gewollten Stelle positionieren zu können. Das heißt anstatt das Objekt selbst an die Position im WebFront zu setzen, wird ein Link von diesem Objekt an die Position gesetzt.
Eine weitere Anwendungsmöglichkeit wären Links auf "verstreute" Objekte um eine Übersichtsseite zu erstellen, welche zentral alle wichtigen Informationen abbildet.
Ein Objekt kann wiefolgt verlinkt werden: Rechtsklick auf das zu verlinkende Objekt -> "Objekt verlinken" -> Zu verlinkendes Objekt auswählen.



Für die iOS und Android App sind mehrfach Verschachtelung nicht möglich. Per Link können verschachtelte Objekte wieder an die "Oberfläche" geholt und somit auch dargestellt werden.

## BEISPIEL

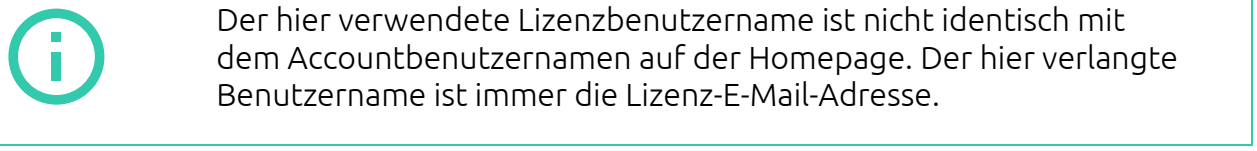
Ein mögliches Beispiel befindet sich unter dem Punkt Vorgehensweisen -> Links verwenden

# Komponenten

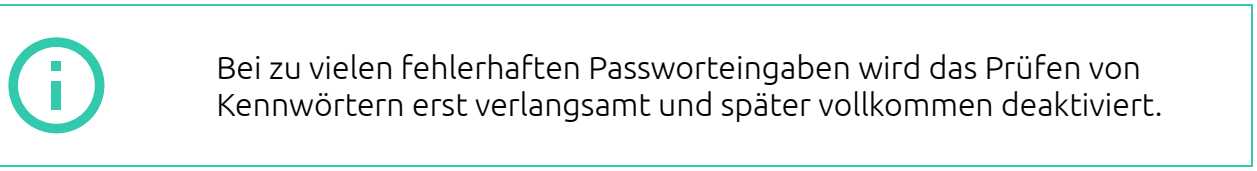
IP-Symcon besteht aus mehreren Software-Komponenten, die in den nachfolgenden Unterpunkten genauer erklärt werden.

# Fernzugriff

Standardmäßig ist der Zugriff im lokalen Netzwerk ohne Benutzername/Kennwort möglich. Externe Zugriff (z.B. über Symcon Connect) sind vollständig deaktiviert. Um den Zugriff von extern zu erlauben oder im lokalen Netzwerk per Kennwort zu schützen, kann der Fernzugriff aktiviert werden. Dabei sollte ein sicheres Kennwort (mindestens acht Zeichen mit Sonderzeichen) vergeben werden. Sobald der Fernzugriff aktiviert wurde, kann über die Verwaltungskonsole durch die Eingabe von Benutzernamen (Lizenz-Benutzername) und dem vorher gesetzten Kennwort auf das System zugegriffen werden.

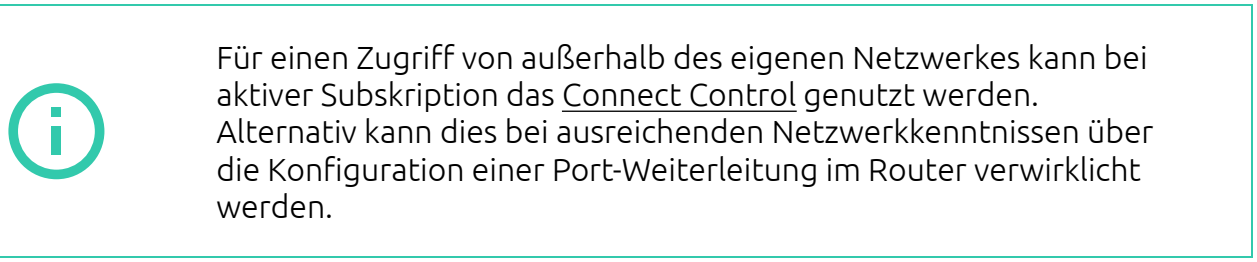


Der hier verwendete Lizenzbenutzername ist nicht identisch mit dem Accountbenutzernamen auf der Homepage. Der hier verlangte Benutzername ist immer die Lizenz-E-Mail-Adresse.



Bei zu vielen fehlerhaften Passworteingaben wird das Prüfen von Kennwörtern erst verlangsamt und später vollkommen deaktiviert.

Um von einem anderen Rechner per Fernzugriff auf IP-Symcon zuzugreifen wird die Verwaltungskonsole benötigt. Sobald die Verwaltungskonsole gestartet wurde, erscheint der Verbindungsassistent der automatisch alle IP-Symcon Server im eigenen Netzwerk findet. Nach der Serverauswahl muss der Benutzername und das Kennwort eingetragen werden, um eine Verbindung aufzubauen.



Für einen Zugriff von außerhalb des eigenen Netzwerkes kann bei aktiver Subskription das Connect Control genutzt werden. Alternativ kann dies bei ausreichenden Netzwerkkenntnissen über die Konfiguration einer Port-Weiterleitung im Router verwirklicht werden.

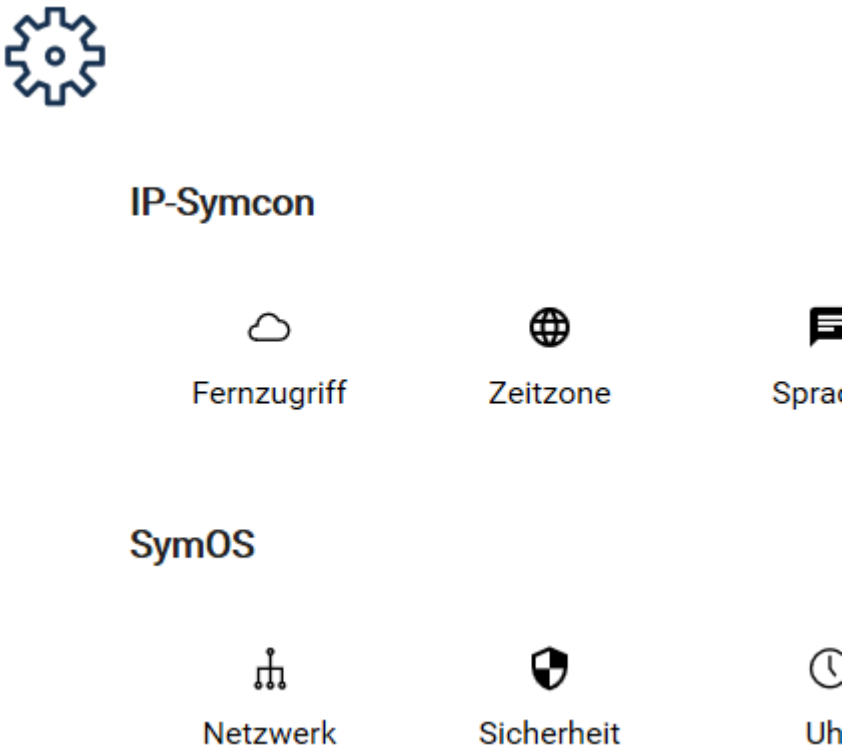
## FERNZUGRIFF AKTIVIEREN/DEAKTIVIEREN

Um den Fernzugriff zu aktivieren muss ein Passwort vergeben sein. Soll der Fernzugriff wieder deaktiviert werden, muss das Fernzugriffpasswort einfach "leer" gelassen und gespeichert werden.
SymOS (SymBox)
Windows (Desktop/Server)
MacOS
Linux (Ubuntu)
Raspberry Pi
Docker
QNAP
Synology

Ab Version 4.4 kann auch innerhalb einer Demo-Lizenz der Fernzugriff aktiviert werden. Bei einem Zugriff in der Demo-Version via IPSStudio oder Mediola muss der Benutzername leer gelassen und nur das Passwort eingetragen werden.

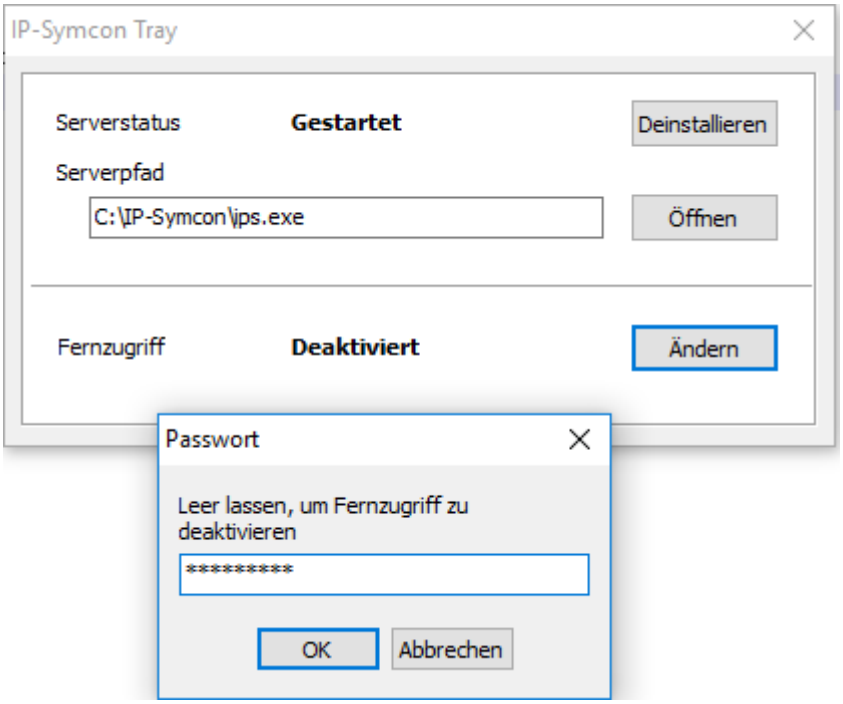
### SYMOS

Über die Weboberfläche kann via "Einstellungen -> Fernzugriff" ein Passwort eingerichtet werden.



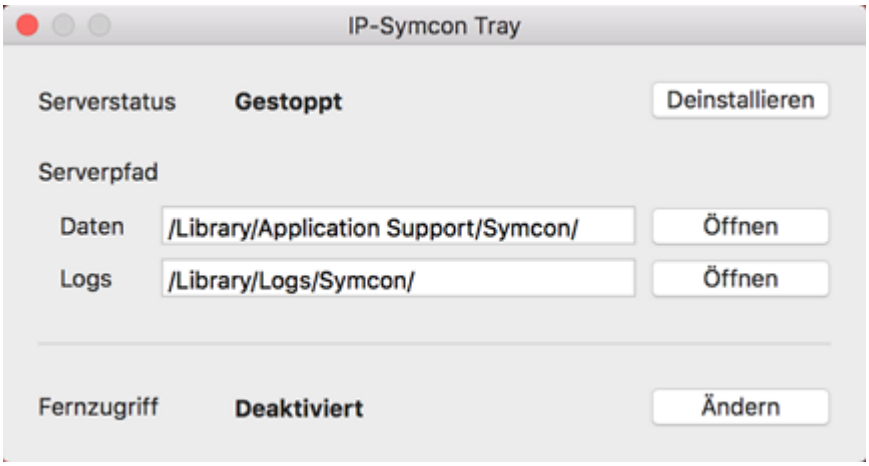

### WINDOWS

Über die Tray-Anwendung kann via "Rechtsklick -> Information" der Dialog aufgerufen werden, bei dem unten der Punkt Fernzugriff anklickbar ist. Dort kann das Passwort gesetzt werden.




## MACOS

Über die Tray-Anwendung kann via "Klick -> Information" der Dialog aufgerufen werden, bei dem unten der Punkt Fernzugriff anklickbar ist. Dort kann das Passwort gesetzt werden.




## LINUX

Die Lizenz muss in der Verwaltungskonsole korrekt eingerichtet sein. Im Anschluss muss folgender Befehl ausgeführt werden:

```
sudo nano /root/.symcon
```

Es sollten mehrere Einträge vorhanden sein. In eine neue Zeile muss folgendes eingefügt werden. (Keine Leerzeilen lassen!)

```
Password=c3ltY29u
```

Danach Speichern und Beenden. Der Dienst muss neugestartet werden.
Das Kennwort c3ltY29u ist "symcon" in Base64 kodiert. Das Kennwort muss Base64 kodiert werden. Dazu kann dieses Tool genutzt werden: base64encode.org

### RASPBERRY PI

Die Lizenz muss in der Verwaltungskonsole korrekt eingerichtet sein. Im Anschluss muss folgender Befehl ausgeführt werden:

```
sudo nano /root/.symcon
```

Es sollten mehrere Einträge vorhanden sein. In eine neue Zeile muss folgendes eingefügt werden. (Keine Leerzeilen lassen!)

```
Password=c3ltY29u
```

Danach Speichern und Beenden. Der Dienst muss neugestartet werden.
Das Kennwort c3ltY29u ist "symcon" in Base64 kodiert. Das Kennwort muss Base64 kodiert werden. Dazu kann dieses Tool genutzt werden: base64encode.org

### DOCKER

Die Lizenz muss in der Verwaltungskonsole korrekt eingerichtet sein. Im Anschluss muss folgender Befehl ausgeführt werden:

```
sudo nano /opt/symcon/.symcon
```



Der Pfad bezieht sich auf den in der Installation von IP-Symcon angegeben Pfad. Dieser kann abweichen.

Es sollten mehrere Einträge vorhanden sein. In eine neue Zeile muss folgendes eingefügt werden. (Keine Leerzeilen lassen!)

```
Password=c3ltY29u
```

Danach Speichern und Beenden. Der Dienst muss neugestartet werden.
Das Kennwort c3ltY29u ist "symcon" in Base64 kodiert. Das Kennwort muss Base64 kodiert werden. Dazu kann dieses Tool genutzt werden: base64encode.org

## QNAP

Die Lizenz muss in der Verwaltungskonsole korrekt eingerichtet sein. Im Anschluss muss mit einem Editor die .symcon Datei angepasst werden. Diese befindet sich in dem bei der Installation verknüpften Root-Volume.

```
/Symcon/MeinIPSymcon/.symcon
```



Der Pfad bezieht sich auf den in der Installation von IP-Symcon angegeben Pfad. Dieser kann abweichen.

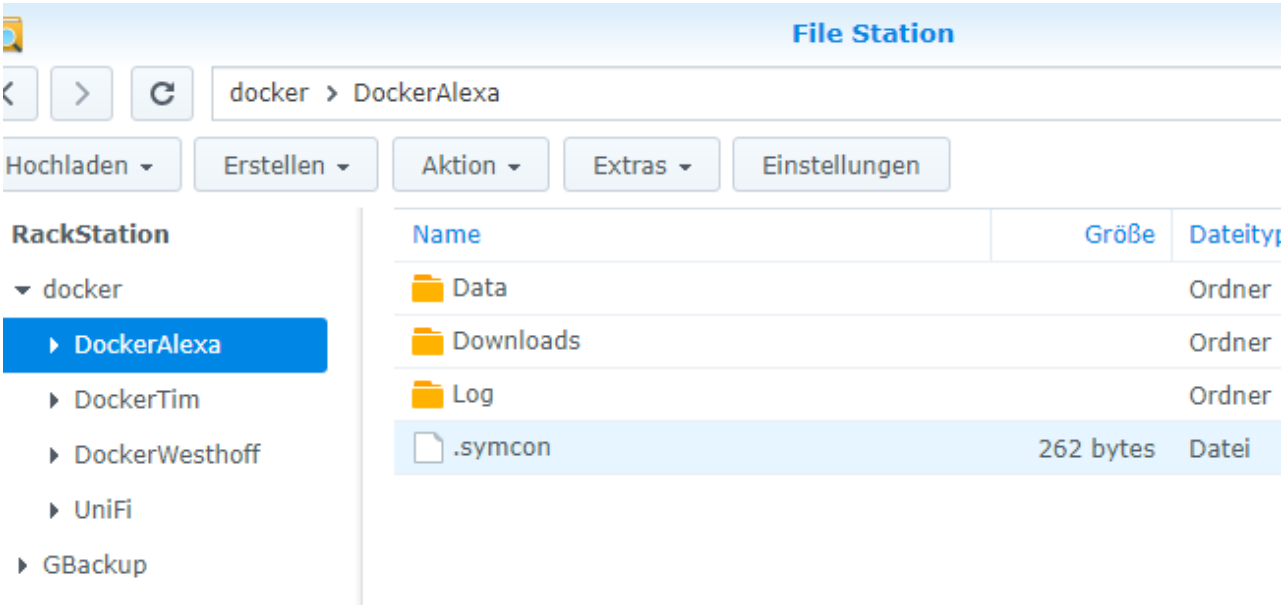
Es sollten mehrere Einträge vorhanden sein. In eine neue Zeile muss folgendes eingefügt werden. (Keine Leerzeilen lassen!)

```
Password=c3ltY29u
```

Danach Speichern und Beenden. Der Dienst muss neugestartet werden.
Das Kennwort c3ltY29u ist "symcon" in Base64 kodiert. Das Kennwort muss Base64 kodiert werden. Dazu kann dieses Tool genutzt werden: base64encode.org

## SYNOLOGY

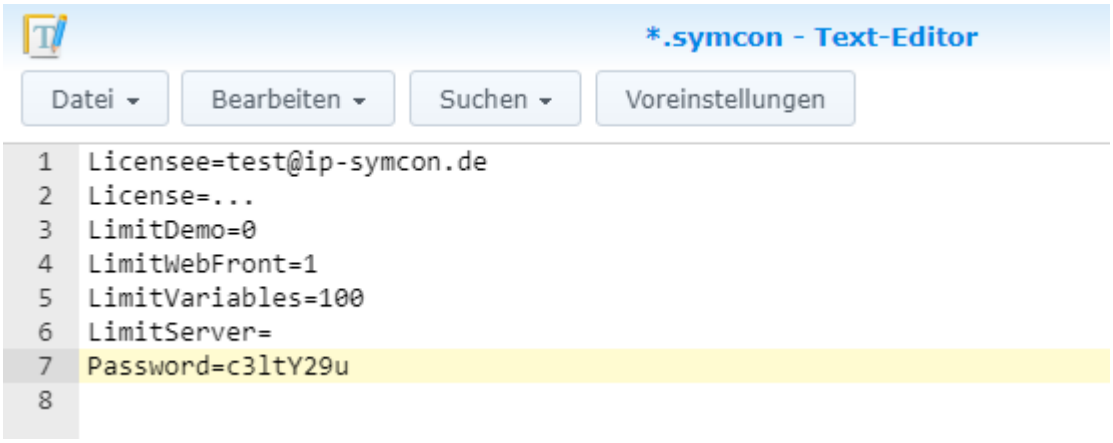
Die Lizenz muss in der Verwaltungskonsole korrekt eingerichtet sein. Im Anschluss muss mit einem Editor die .symcon Datei angepasst werden. Diese befindet sich in dem bei der Installation verknüpften Root-Volume.



Der Pfad bezieht sich auf den in der Installation von IP-Symcon angegeben Pfad. Dieser kann abweichen.

Es sollten mehrere Einträge vorhanden sein. In eine neue Zeile muss folgendes eingefügt werden. (Keine Leerzeilen lassen!)

```
Password=c3ltY29u
```




Danach Speichern und Beenden. Der Dienst muss neugestartet werden.
Das Kennwort c3ltY29u ist "symcon" in Base64 kodiert. Das Kennwort muss Base64 kodiert werden. Dazu kann dieses Tool genutzt werden: base64encode.org

# Tray

Die IP-Symcon Tray-Anwendung ist die Schnittstelle zum IP-Symcon Dienst, der im Hintergrund läuft. Über das Kontextmenü können bestimmte Einstellungen administriert werden. Dazu gehören z.B. das Starten und Stoppen des Dienstes oder das Öffnen der Verwaltungskonsole.



Der Tray steht nur für MacOS und Windows Betriebssysteme zur Verfügung.

Folgende weitere Einstellungen können über die Tray-Anwendung getätigt werden:

- Update starten
- Dienst deinstallieren
- Lizenz ändern
- Fernzugriff ändern

# Dienst

Der Dienst ist das Herzstück von IP-Symcon. Er läuft immer im Hintergrund und wird üblicherweise beim Start des Betriebssystems automatisch mit gestartet. Dadurch muss auch kein Benutzer am Rechner/Server eingeloggt sein, um eine einwandfreie Funktion zu gewährleisten.

## KONFIGURATIONSDATEI

Die Konfiguration (settings.json) von IP-Symcon wird alle 10 Minuten oder bei ordnungsgemäßen Herunterfahren gespeichert. Sollte während des Speicherns die Stromversorgung zusammenbrechen, kann es zu einer korrupten bzw. defekten Konfigurationsdatei kommen.
Um dies zu korrigieren, geht IP-Symcon alle im Backup Ordner vorhandenen automatisch generierten Sicherungen der Settings.json durch, bis eine funktionale und nicht defekte gefunden wird. Sollte keine funktionale Konfigurationsdatei gefunden werden, startet der Dienst nicht und gibt eine Fehlermeldung aus.

# PHP

IP-Symcon verwendet als Skriptsprache PHP. Durch die Tatsache, dass PHP komplett integriert ist, können alle Vorteile von PHP genutzt werden. Die Ausführung der Skripte kann dabei parallel erfolgen.

### VERSIONSÜBERSICHT

| IP-SYMCON VERSION | INTEGRIERTE PHP VERSION |
|---|---|
| ab 1.0 | 5.1.x (x86 Thread Safe) |
| ab 2.2 | 5.3.x (x86 Thread Safe) |
| ab 2.5 | 5.4.x (x86 Thread Safe) |
| ab 4.0 | 5.6.x (x86 Thread Safe) |
| ab 5.0 | 7.2.x (x64 Thread Safe) |
| ab 5.1 | 7.3.x (x64 Thread Safe) |
| ab 5.5 | 7.4.x (x64 Thread Safe) |
| ab 7.0 | 8.2.x (x64 Thread Safe) |

Zusätzlich zu den normalen PHP Funktionen sind spezielle IP-Symcon Funktionen vorhanden, über die auf IP-Symcon-spezifische Einstellungen (siehe Befehlsreferenz) oder über die in IP-Symcon eingerichteten Geräte zugegriffen werden kann (siehe Modulreferenz).
Die aus PHP bekannte "php.ini" ( php.net/manual/de/ini.php ) ist in IP-Symcon ebenfalls vorhanden. IP-Symcon konfiguriert dabei die “extension” Einträge automatisch, anhand der vorhandenen Erweiterungen, die im "IP-Symcon/ext" Ordner abgelegt wurden.

## GLOBALES INCLUDE

Um Funktionen, Konstanten usw. global über alle Skripte hinweg zur Verfügung zu stellen müssen diese in der Datei "__autoload.php" definiert werden. Diese muss sich im "IP-Symcon/scripts"-Ordner befinden.
Innerhalb der "__autoload.php" können dann weitere Dateien eingelesen werden.



Befehle oder Dateien via "__autoinclude.inc.php" einzulesen funktioniert auch, wird aber jedesmal bei einem Update oder Neuinstallation überschrieben.



Die PHP Funktion "auto_prepend_file" ist nicht nutzbar, da diese schon von IP-Symcon aufgerufen und genutzt wird. Diese soll/kann nur einmal Systemweit genutzt werden.

### BEISPIEL

```
__autoload.php
require_once(IPS_GetKernelDir() . "/scripts/
globalfunction.ips.php");
```

## KONFIGURIEREN

Mögliche Konfigurationsparameter können dem PHP Handbuch entnommen werden. Eine nützliche Einstellung ist die Verlängerung der maximalen Skriptlaufzeit, falls etwaige Skripte (z.B. ShutterControl) dies notwendig machen sollten. Bei Anpassung dieser Einstellung muss beachtet werden, dass nur eine begrenzte Anzahl von PHP Skripten in IP-Symcon parallel ausgeführt werden können. Wenn durch eine lange Skriptlaufzeit alle Slots besetzt sein sollten, werden andere Skripte in eine Warteschlage abgelegt und erst verzögert aufgerufen. "Sleep"-Anweisungen sollten deshalb immer vermieden und gegen "Timer"-Anweisungen getauscht werden.



IP-Symcon muss nach einer Änderung der "php.ini" neu gestartet werden!

### BEISPIEL

Maximale Skriptlaufzeit auf 5 Minuten erhöhen.

```
[PHP]
max_execution_time=300
```

# Schnittstellen

## HARDWARE MODULE

sortiert nach Anschlüssen und Alphabet

| KOMPONENTEN | HERSTELLER | ANSCHLÜSSE | UNTERSTÜTZTE GERÄTE |
|---|---|---|---|
| 1-Wire | Maxim Dallas | Kabelgebunden | Geräteliste |
| ALLNET | ALLNET | Kabelgebunden | Geräteliste |
| digitalSTROM | AIZO | Kabelgebunden | Geräteliste |
| DMX / Artnet | u.a. DMX4ALL | Kabelgebunden | Geräteliste |
| KNX | u.a. Siemens AG, EIBMarkt, Gira, Jung, Merten, Weinzerl | Kabelgebunden | Geräteliste |
| eKey | eKey | Kabelgebunden | Geräteliste |
| IR Fernbedienung | u.a. IRTrans, WinLIRC | Kabelgebunden | Geräteliste |
| LCN | Issendorff | Kabelgebunden | Geräteliste |
| ModBus TCP , ModBus RTU , ModBus RTU over TCP | z.B. Wago/ Beckhoff SPS | Kabelgebunden | Geräteliste |
| M-Bus | z.B. ALLMESS | Kabelgebunden | Geräteliste |
| SPS Siemens Vipa Logo | Siemens AG | Kabelgebunden | Geräteliste |
| Siemens OZW | Siemens AG | Kabelgebunden | Geräteliste |
| UVR1611 | Technische Alternative | Kabelgebunden | Geräteliste |
| Velleman Board | Velleman | Kabelgebunden | ---------- |
| WuT | W&T (Wiesemann & Theis) | Kabelgebunden | ---------- |
| EnOcean | EnOcean Alliance | Funkbasiert (868Mhz) | Geräteliste |
| FHZ1X00PC (FS20, HMS, FHT, KS300) | ELV | Funkbasiert (868Mhz) | Geräteliste |
| HomeMatic | ELV | Funkbasiert (868Mhz) | Geräteliste |
| IPS-868 | IP-Symcon/ ProJet | Funkbasiert (868Mhz) | IPS-868 |
| XBee | ZigBee | Funkbasiert (868Mhz) | ---------- |
| xComfort | Eaton | Funkbasiert (868Mhz) | Geräteliste |
| Z-Wave | u.a. ACT, Popp, Düwi, Merten, Innovus, Fibaro | Funkbasiert (868Mhz) | Geräteliste |
| FS10 Wetter | ELV | Funkbasiert (433Mhz) | ---------- |

# VIRTUELLE MODULE

sortiert nach Alphabet

| BEZEICHNUNG | FUNKTIONEN | BESONDERHEITEN |
|---|---|---|
| Archive Control | Speichert, verwaltet und stellt geloggte Variablen zur Verfügung. | |
| Calendar Control | Verwaltet geplante An-/Abwesendheit und setzt bereitgestellte Variablen. | |
| Connect Control | Öffnet einen verschlüsselten Tunnel zum Connect Service, welcher einen gesicherten Fernzugriff erlaubt. | Benötigt eine aktive Subskription |
| Cutter | Verarbeitet Binärdaten und zerschneidet/ synchronisiert diese. Kann mit den Schnittstellenmodulen verwendet werden. | |
| DNSSD Control | Verwaltet eingerichtete Bonjour Services. | |
| Event Control | Reagieren auf spezielle Ereignisse wie Start/ Stop des Systems, um spezielle Systemzustände zu speichern/einzulesen | |
| Heating Control | 2-Punkt-Regler, welcher auf Basis von Ist- und Soll-Temperaturen Heizungsaktoren schaltet, um auf voreingestellte Temperaturen zu regeln. | |
| IMAP | Dient zur IMAP Abfrage von E-Mails. | |
| Location Control | Berechnet verschiedene Dämmerungszeitpunkte anhand von Längen- und Breitengrad. | |
| MediaPlayer | WAV, MP3, WMA, Shoutcast, IceCast, WMALive | Mehrere Soundkarten gleichzeitig |
| Module Control | Stellt verschiedene Module, eingebunden über ein Git-Repository, zur Verfügung. | |
| MQTT | MQTT ist ein offenes Nachrichtenprotokoll für Machine-to-Machine-Kommunikation (M2M). IP-Symcon agiert in diesem Zusammenhang als Server (Broker). | |
| Notification Control | Verwaltet alle für Push-Nachrichten registrierten Geräte. | Push Notifications benötigen eine aktive Subskription |
| POP3 | Dient zur POP3 Abfrage von E-Mails. | |
| Presence Control | Bietet die Möglichkeitkeit eine Anwesenheit in einem Bereich auszuwerten | |
| RegisterVariable | Stellt eine Datenweiterleitungs- und -verarbeitungsschnittstelle zur Verfügung | |
| SMTP | Dient zum Versenden von E-Mails via SMTP. | |
| SMS | Dient zum Versenden von SMS mithilfe eines externen SMS Anbieters. | |
| System Informationen | CPU Last, Festplattenauslastung, Batteriestärke, Druckerwarteschlange | |
| Shutter Control | Positioniert ihre Rollladen auf einen genauen % Wert, auch wenn ihr Modul eigentlich nur Hoch/Runter fahren kann. | Muss einmalig angelernt werden |
| TextParser | Extrahiert Daten aus einem Text, der z.B. | |

## SPRACHASSISTENTSYSTEME

sortiert nach Alphabet

| SYSTEM | HERSTELLER | BESONDERHEITEN |
|---|---|---|
| Amazon Alexa | Amazon | Benötigt Symcon Connect |
| Google Assistant | Google | Benötigt Symcon Connect |

## VISUELLE MODULE

sortiert nach Alphabet

| BEZEICHNUNG | FUNKTIONEN |
|---|---|
| Dummy Modul | Erstellt optisch eine Instanz im WebFront. |
| Image Grabber | Erstellt automatisch Webcam-Bilder als Medien-Objekte und stellt diese somit direkt dem WebFront zur Verfügung. |
| Popup Modul | Erstellt ein Popup im WebFront und zeigt verlinkte Objekte an. |
| Skin Control | Dieses Modul verwaltet vorhandene Skins und weißt diese verschiedenen WebFronts zu. |
| WebFront Visualisierung | Verwaltet ein WebFront und bietet verschiedene Einstellmöglichkeiten zur Verfügung. |

## SCHNITTSTELLEN MODULE

sortiert nach Alphabet

| BEZEICHNUNG | FUNKTIONEN | BESONDERHEITEN |
|---|---|---|
| Client Socket | ---------- | TCP-Client |
| HID | ---------- | ---------- |
| HTTPClient | Ruft/liest Webseiten aus | Proxy, Basic Authentication Support |
| Multicast Socket | ---------- | Multicast-Client |
| Serial Port | ---------- | RS232/RS485 |
| Server Socket | ---------- | TCP-Server |
| UDP Socket | ---------- | UDP-Client |

# Verwaltungskonsole

Die Verwaltungskonsole ist das Tool zum Einrichten und Konfigurieren der gesamten Logik des IP-Symcon Servers. Die Verwaltungskonsole gibt es in zwei Versionen: Der webbasierten Verwaltungskonsole und der Pro Konsole.

## WEBBASIERTE VERWALTUNGSKONSOLE

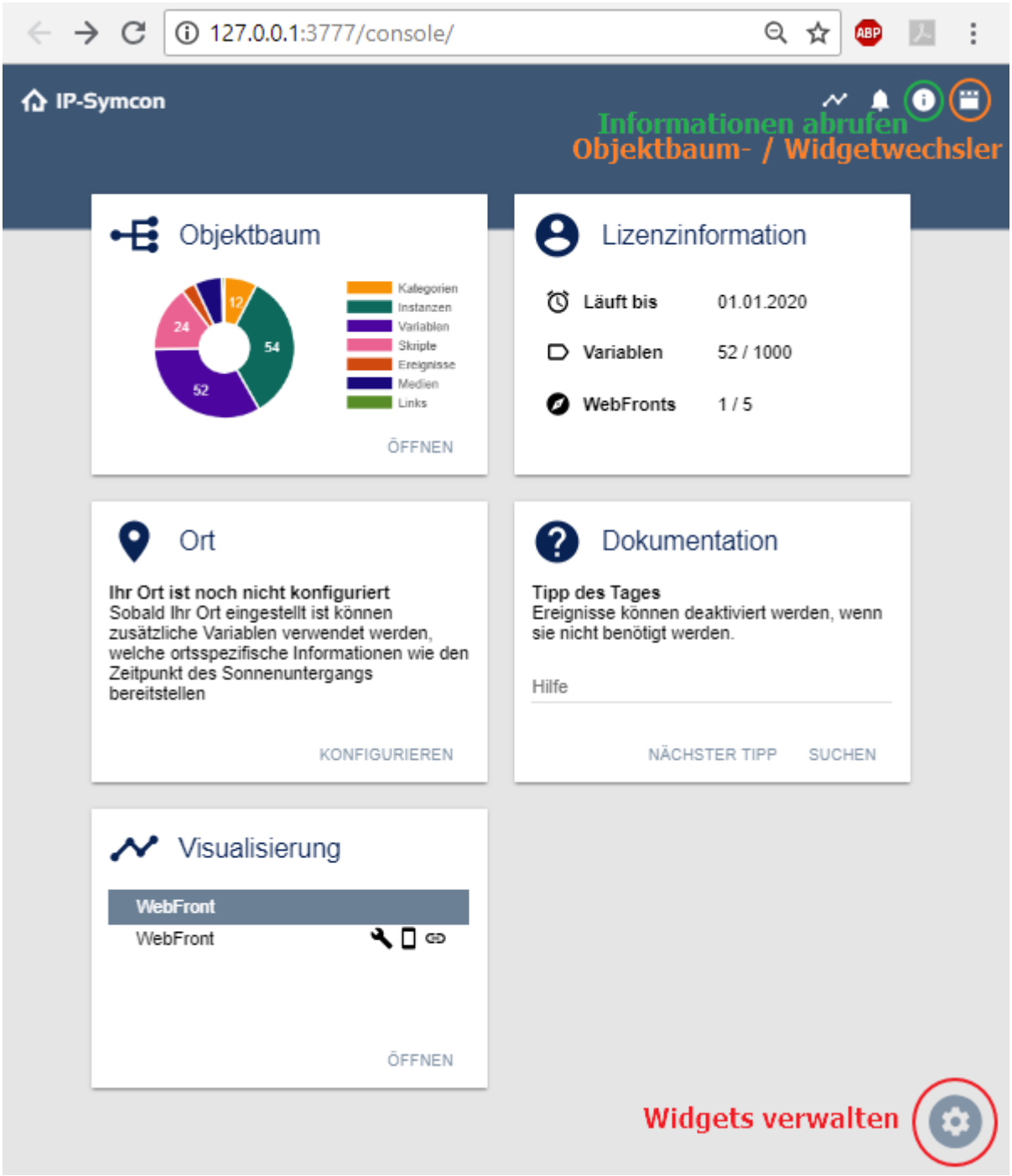
ab IP-Symcon 5.0
Die webbasierte Verwaltungskonsole ist Plattform unabhängig und wird über einen Browser aufgerufen. Offiziell empfohlene Browser sind: Google Chrome, Apple Safari und Opera

### VERBINDEN MIT DEM IP-SYMCON SERVER

Die webbasierte Verwaltungskonsole kann via "[ServerIP]:3777/console/" in einem Browser aufgerufen werden.

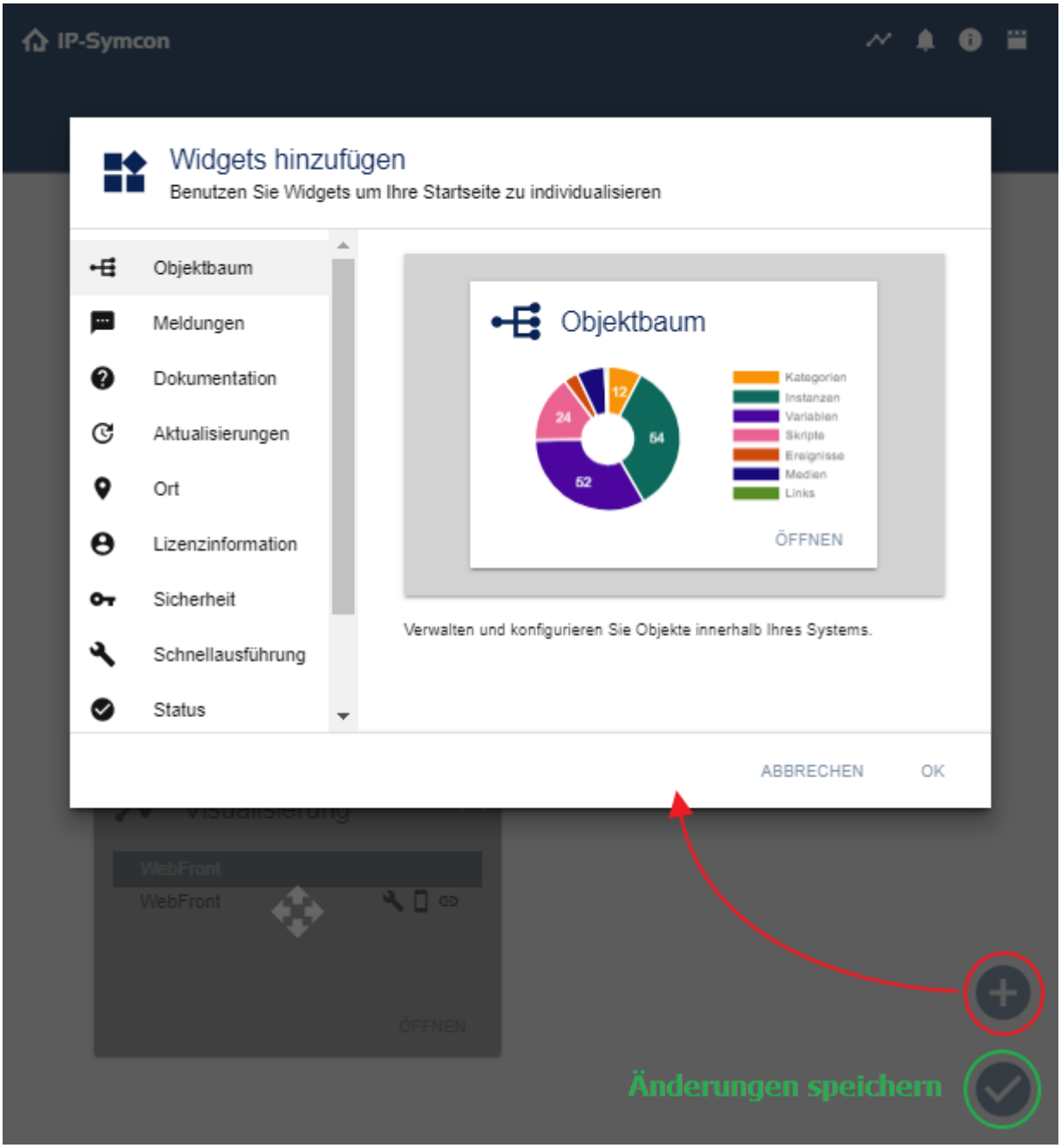
### DIE STARTSEITE

Auf der Startseite sind die Widgets sichtbar. Diese sind individuell konfigurierbar und bieten einen ersten Überblick über viele Aktivitäten des Systems.



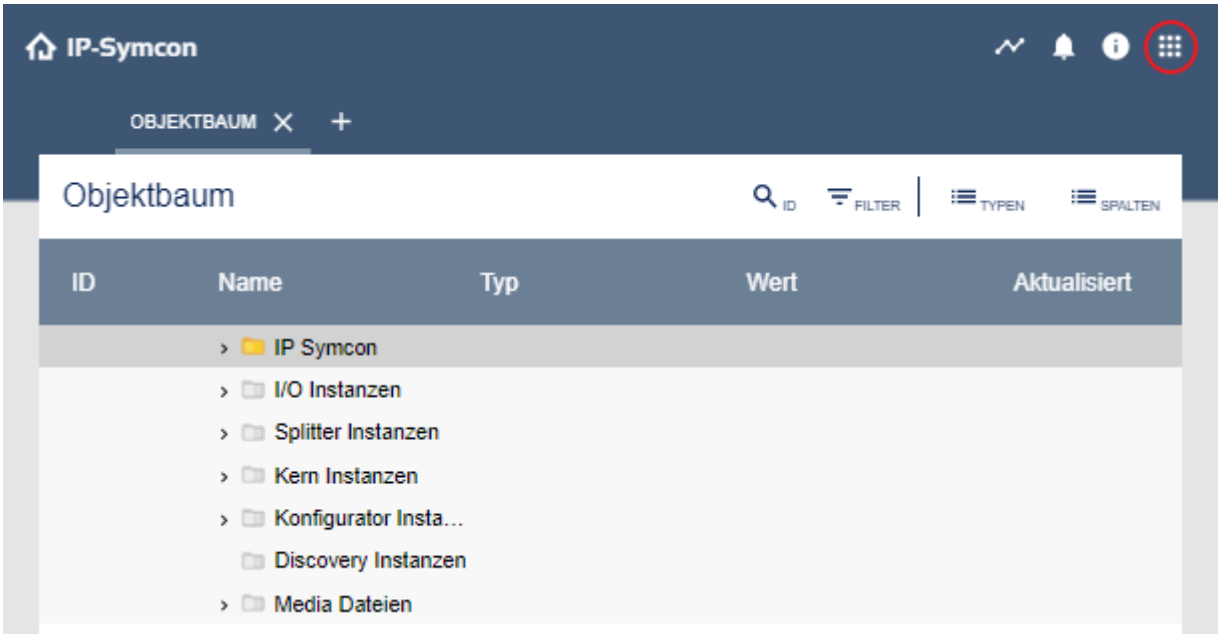

## WIDGETS KONFIGURIEREN

Auf der Startseite können via dem Zahnrad-Button unten rechts die Widgets ein-/ ausgeblendet werden.




## ANSICHTWECHSLER

Über den Button oben rechts kann schnell zwischen Startseite und Tabliste gewechselt werden.



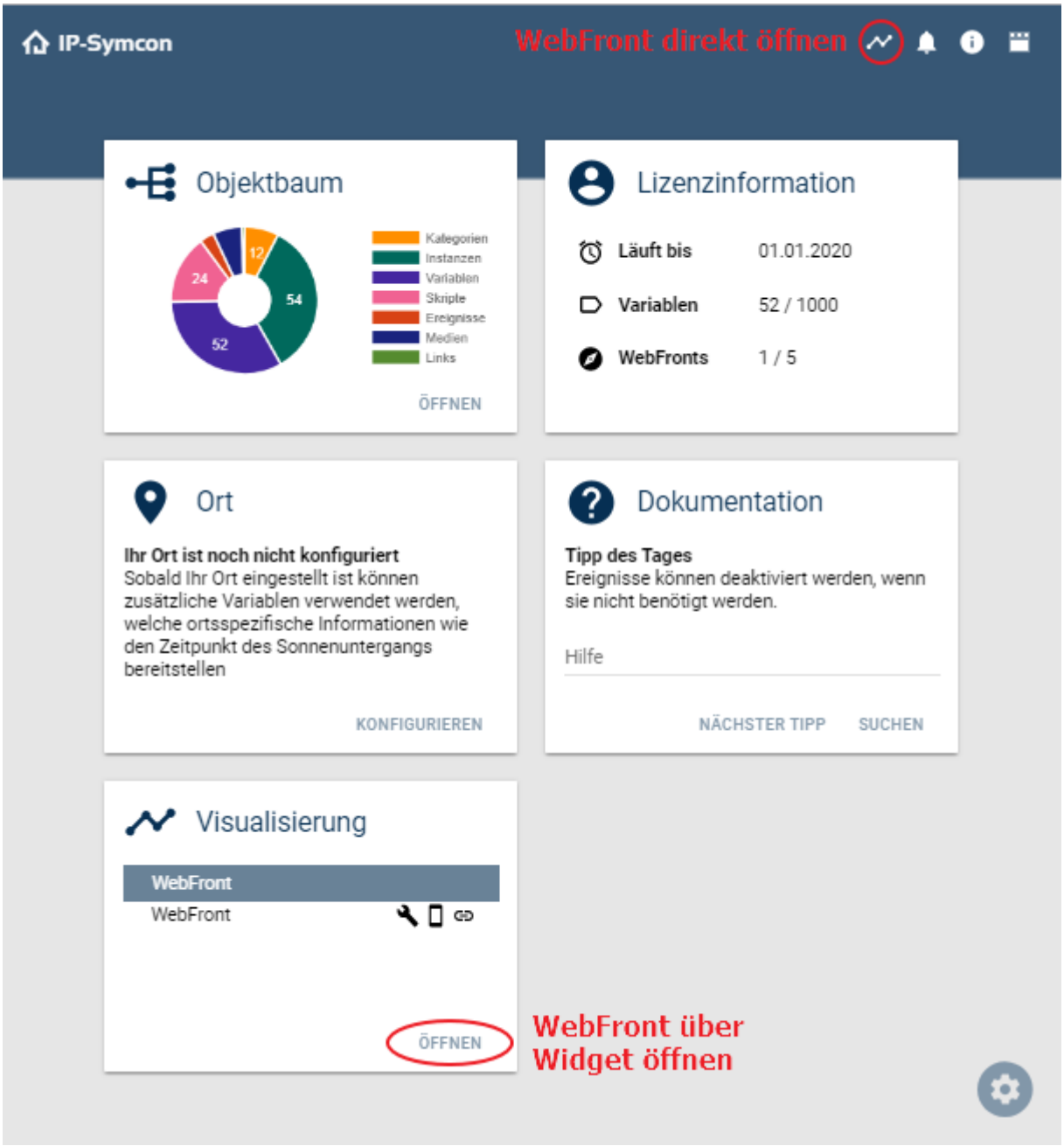

## TABLISTE

In der Tabliste können diverse Tabs geöffnet werden. Dies beinhaltet unter anderem den Objektbaum, die Meldungen oder die Konfiguration von Instanzen oder Automationen. Über das Plus-Icon rechts von der Tabliste können neue Tabs geöffnet werden. Sind mehrere Tabs von Objekten mit gleichem Namen geöffnet, so wird zusätzlich der Pfad zu diesen Objekten angezeigt, so dass die Tabs zugewiesen werden können.



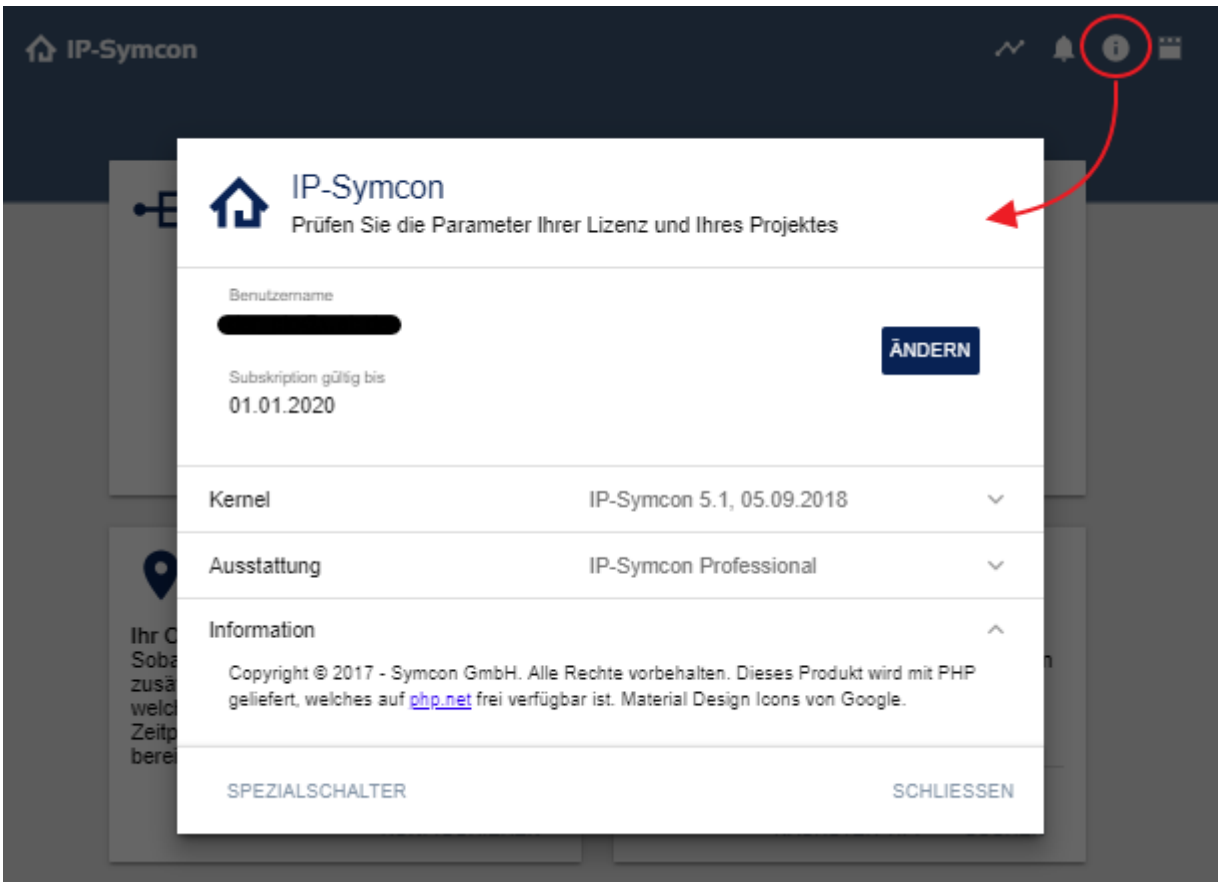

## VISUALISIERUNG

Über den Button oben rechts kann direkt das WebFront geöffnet werden.



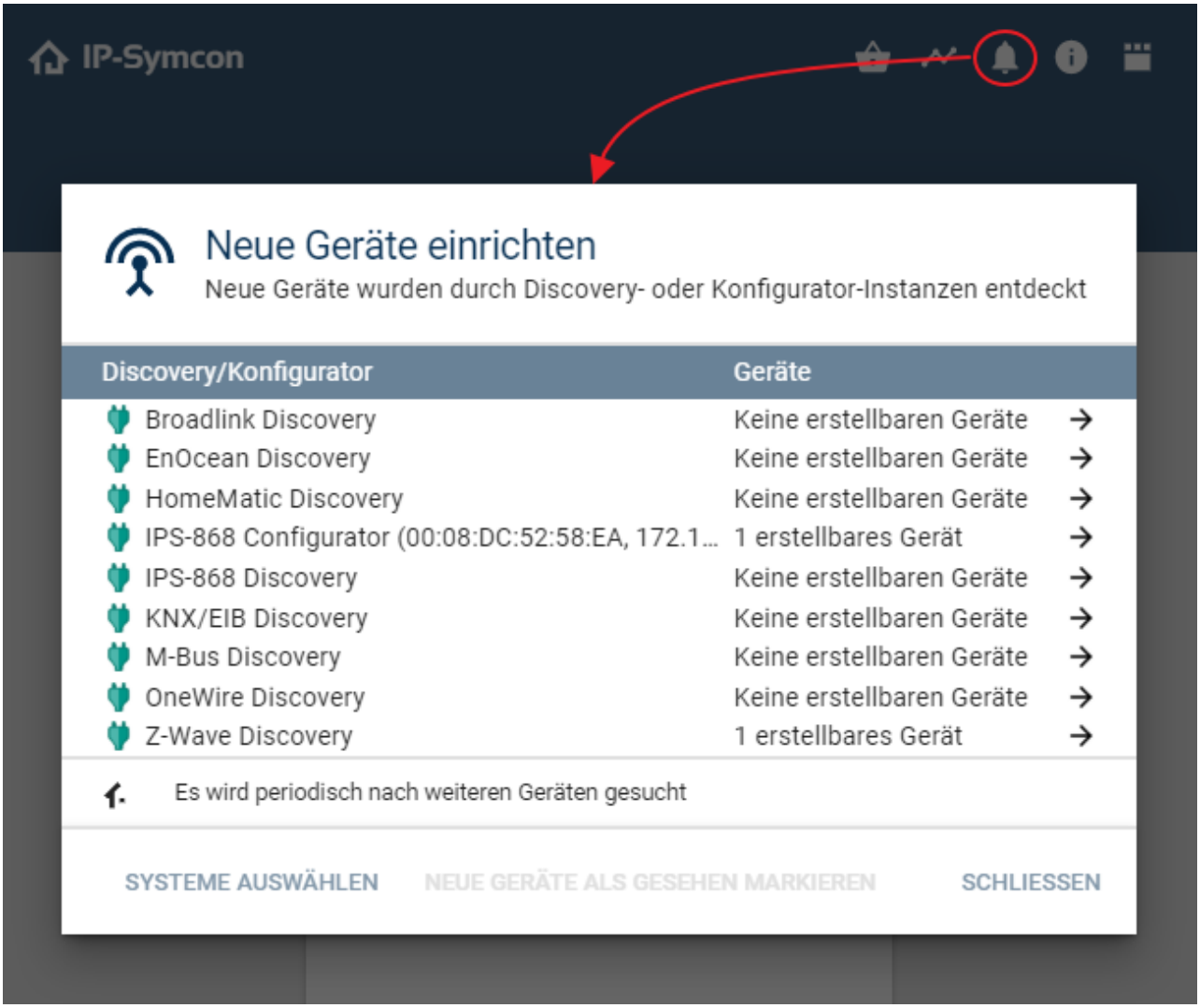

## INFORMATIONEN

Über den Informationen-Button können diverse Informationen abgefragt werden. Dazu zählen Kernel- und Ausstattungsinformationen, sowie einen Button zu den Spezialschaltern.



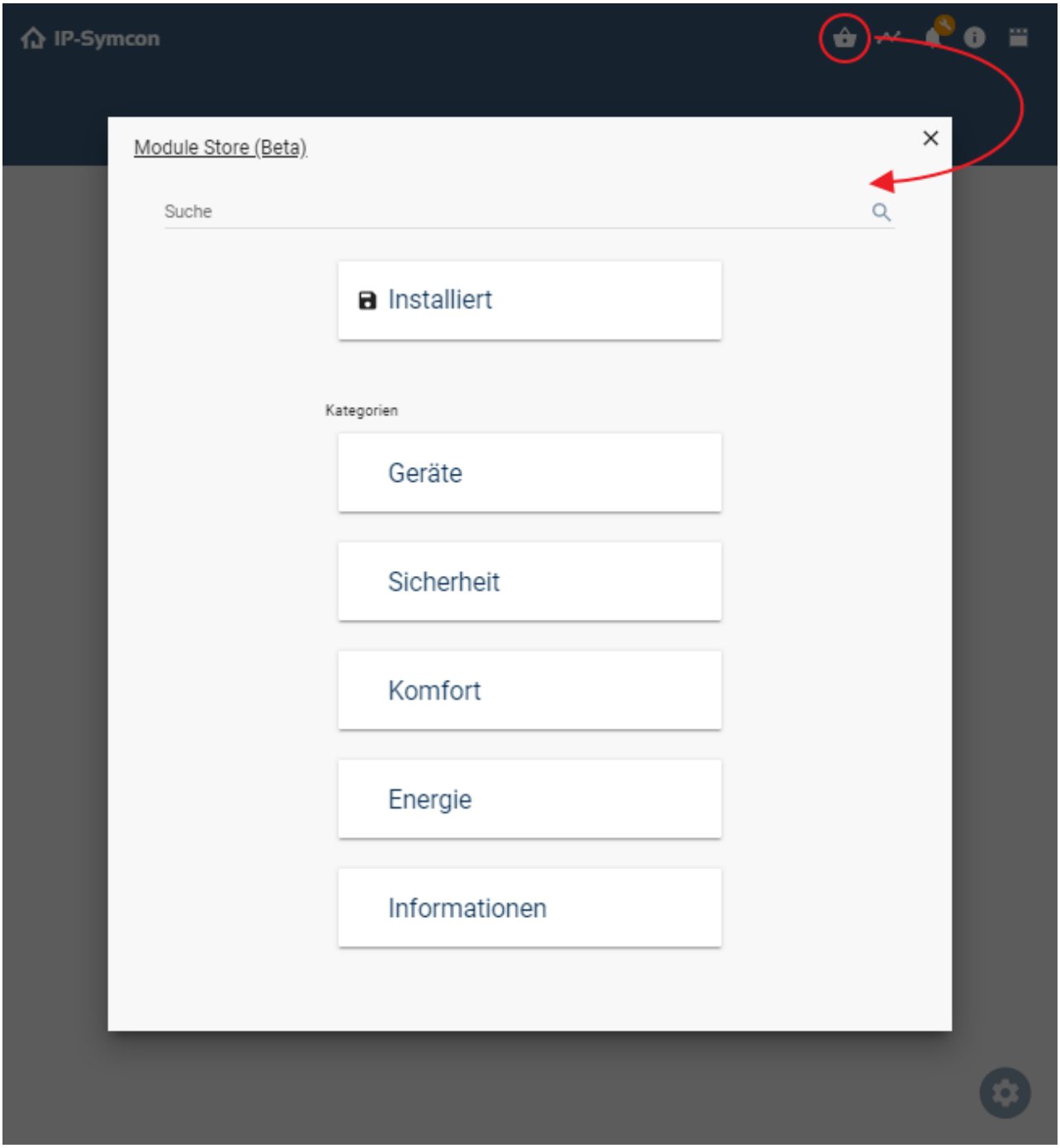

## GERÄTESUCHE

Über den Gerätesuche-Button können Discoverys eingerichtet werden.
Diese informieren über neu angeschlossene Geräte und vereinfachen die Einrichtungen neuer Geräte.




## MODULE STORE

Über den Module Store können neue Module installiert werden. Diese Module erweitern den Funktionsumfang und ermöglichen so die Verwendung von neuen Funktionen oder weiteren Geräten.


IP-Symcon
Module Store (Beta)
Suche
Installiert
Kategorien
Geräte
Sicherheit
Komfort
Energie
Informationen

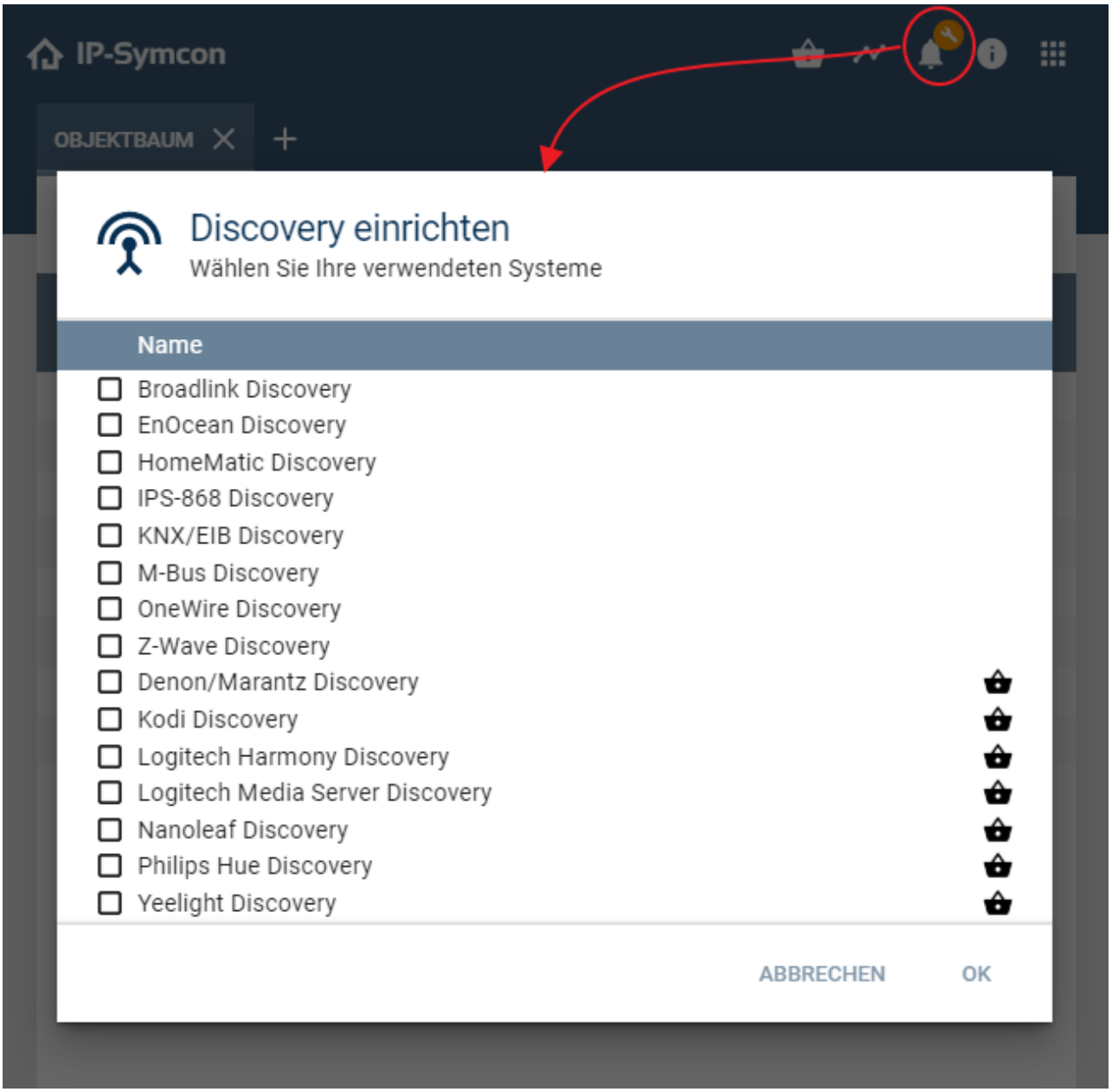
# Gerätesuche

Über den Gerätesuche-Button können Discoverys eingerichtet und Konfiguratoren aufgerufen werden. Diese informieren über neu angeschlossene Geräte und vereinfachen die Einrichtungen neuer Geräte.

## DISCOVERY EINRICHTEN

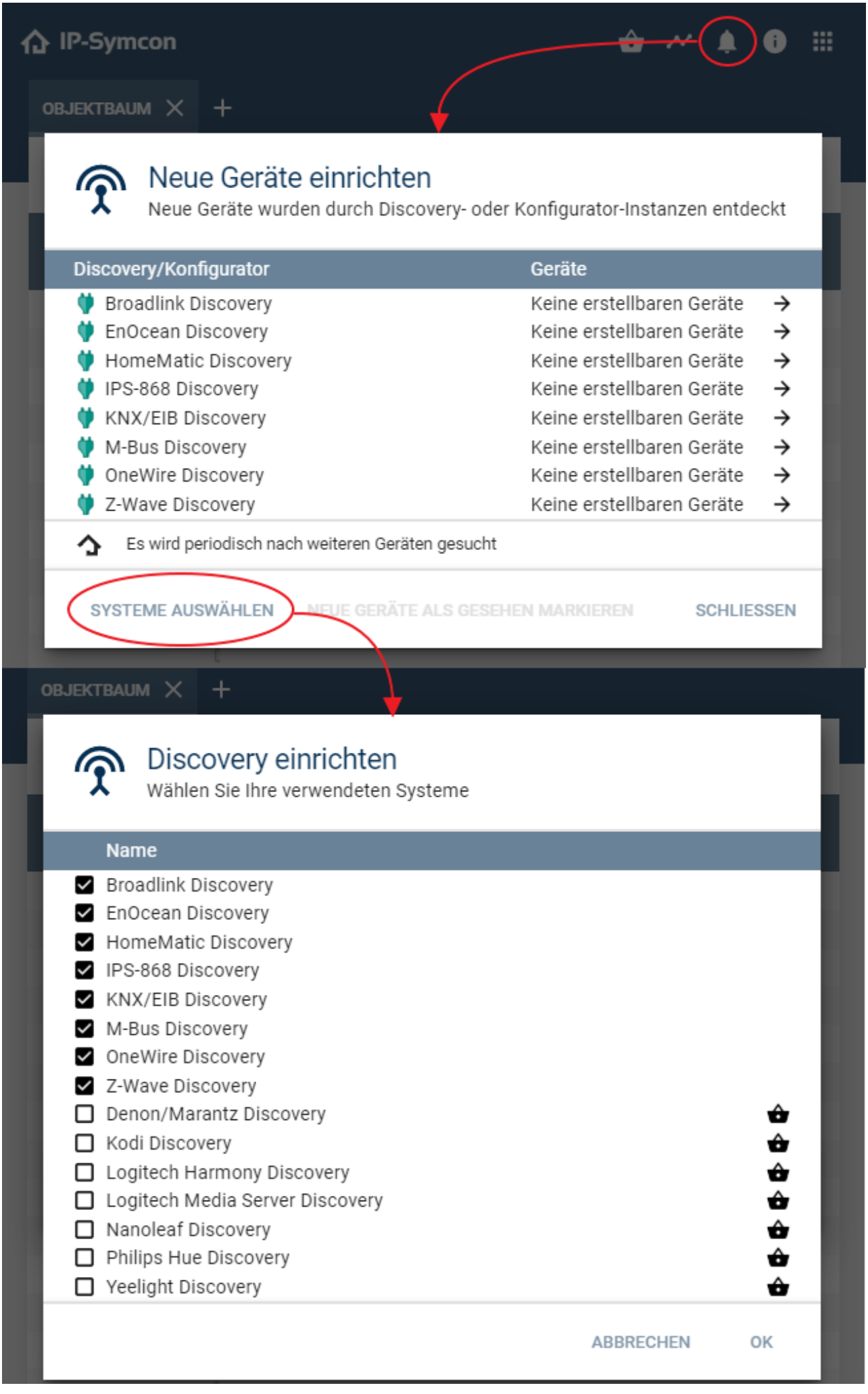
Die Gerätesuche ist über die Glocke oben rechts in der Konsole aufrufbar.

### ERSTMALIGE EINRICHTUNG




Über das Anhaken am linken Rand werden neue Discoverys eingerichtet.
Ist ein Korb am rechten Rand, so handelt es sich um ein Discovery aus dem Module Store.
Diese können durch einen Klick auf den Korb geöffnet werden.
Beim Einrichten eines Discoverys werden automatisch benötige IO-/Splitter-Instanzen erstellt.

## ZUSÄTZLICHE EINRICHTUNG


IP-Symcon
OBJEKTBAUM
Neue Geräte einrichten
Neue Geräte wurden durch Discovery- oder Konfigurator-Instanzen entdeckt
Discovery/Konfigurator
Geräte
Broadlink Discovery
Keine erstellbaren Geräte
EnOcean Discovery
Keine erstellbaren Geräte
HomeMatic Discovery
Keine erstellbaren Geräte
IPS-868 Discovery
Keine erstellbaren Geräte
KNX/EIB Discovery
Keine erstellbaren Geräte
M-Bus Discovery
Keine erstellbaren Geräte
OneWire Discovery
Keine erstellbaren Geräte
Z-Wave Discovery
Keine erstellbaren Geräte
Es wird periodisch nach weiteren Geräten gesucht
SYSTEME AUSWÄHLEN
NEUE GERÄTE ALS GESEHEN MARKIEREN
SCHLIESSEN
OBJEKTBAUM
Discovery einrichten
Wählen Sie Ihre verwendeten Systeme
Name
Broadlink Discovery
EnOcean Discovery
HomeMatic Discovery
IPS-868 Discovery
KNX/EIB Discovery
M-Bus Discovery
OneWire Discovery
Z-Wave Discovery
Denon/Marantz Discovery
Kodi Discovery
Logitech Harmony Discovery
Logitech Media Server Discovery
Nanoleaf Discovery
Philips Hue Discovery
Yeelight Discovery
ABBRECHEN
OK

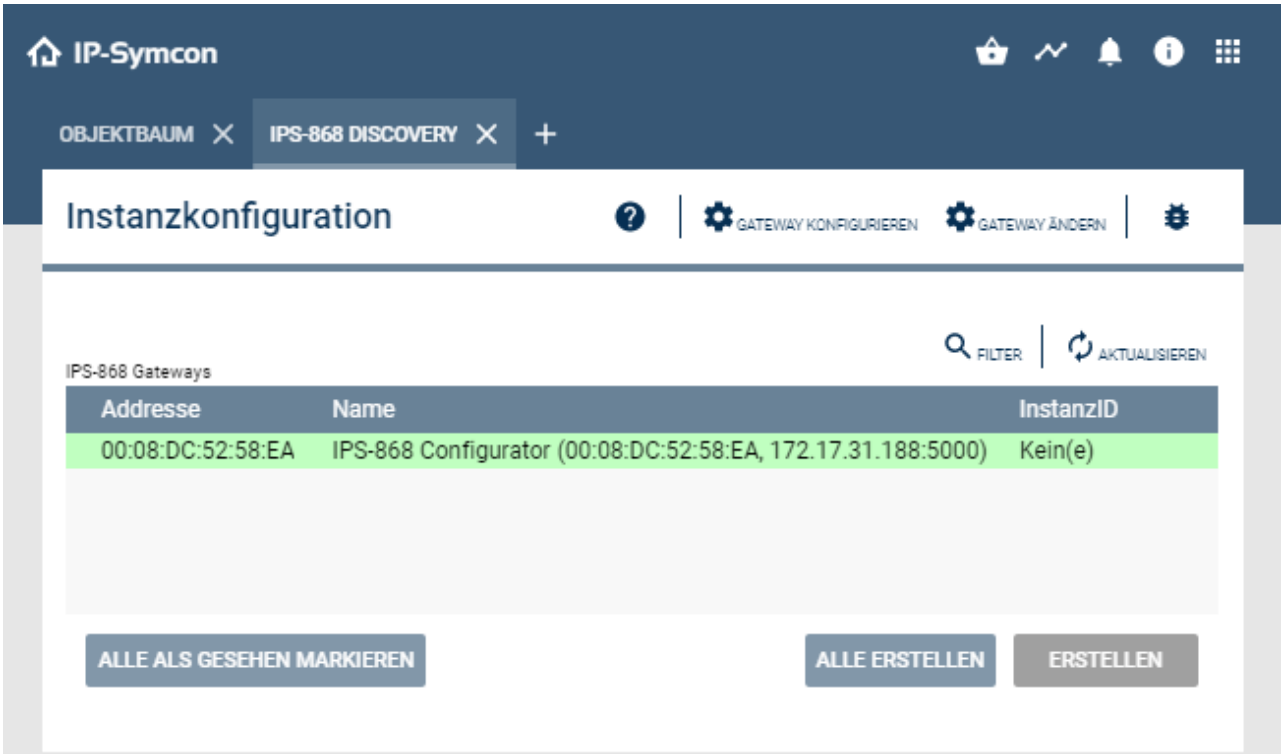
Über "Systeme auswählen" können weitere Discovery Systeme hinzugefügt werden.




Über die eingerichteten Discoverys können Konfiguratoren erstellt und automatisch konfiguriert werden. Erstellte Konfiguratoren werden in der Gerätesuche angezeigt. Über die Pfeile am rechten Rand kann die Konfiguration zu eingerichteten Discoverys/ Konfiguratoren geöffnet werden.

## KONFIGURATOR ERSTELLEN

Im jeweiligen Discovery können systemspezifische Konfiguratoren erstellt werden. Dazu muss der passende Eintrag aus der Liste ausgewählt und auf "Erstellen" geklickt werden. Das Discovery übernimmt automatisch die vollständige Konfiguration des Konfigurators und dieser kann sofort genutzt werden.




## VERSCHIEDENE STATUS

Das Icon der Gerätesuche zeigt direkt an, ob zum Beispiel ein neues Gerät erstellt werden kann.
Dazu gibt es drei Status, welche direkt am Icon innerhalb der Konsole ersichtlich sind.

### STATUS UNVERÄNDERT

Sollten keine neuen Geräte erkannt, alle erstellbaren Geräte erstellt oder als "gesehen" markiert worden sein, wird nur das Symbol an sich angezeigt.



### KEIN DISCOVERY EINGERICHTET

Es wurden noch keine Discovery Instanz eingerichtet.



### NEUE GERÄTE ERSTELLBAR

Wenn neue Geräte erkannt werden, wird dessen Anzahl in einer grünen Markierung erkenntlich gemacht.

Ist keine Einrichtung erwünscht, kann diese Meldung im Gerätesuchefenster mit "Neue Geräte als Gesehen markieren" ausgeblendet werden



# Meldungen

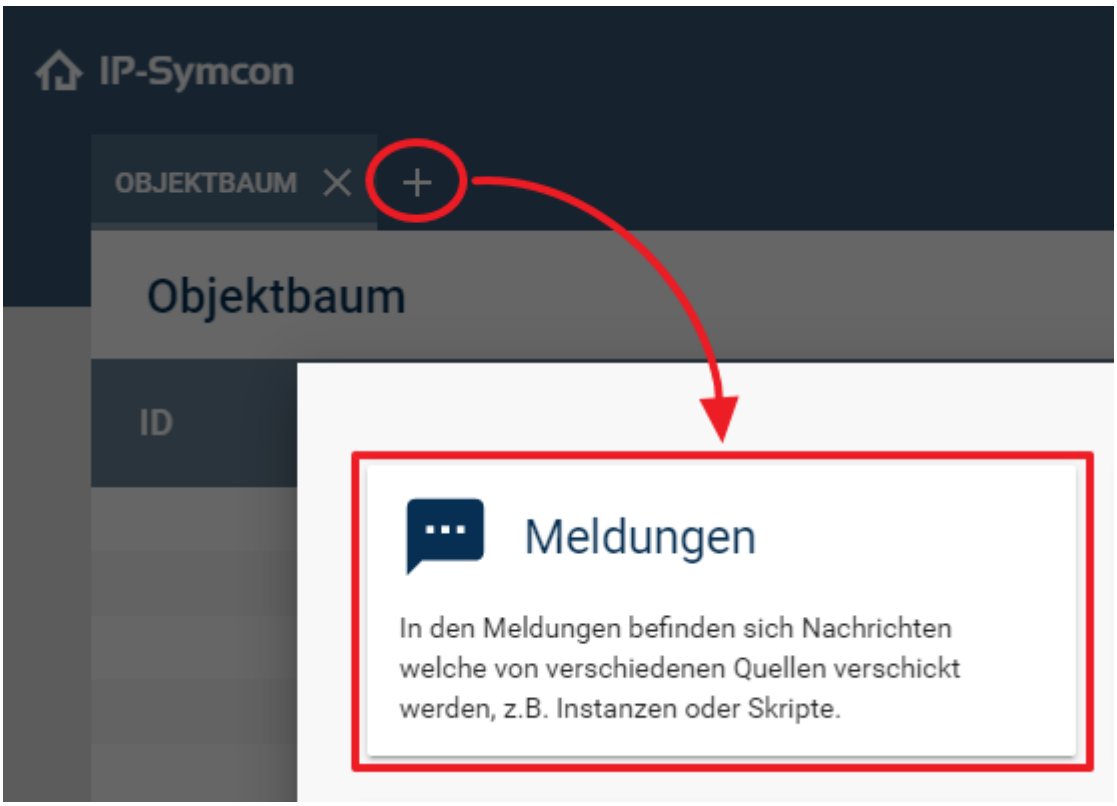
Im Meldungenfenster können Informationen abgelesen werden, was alles auf dem IP-Symcon Server gerade passiert. Hier werden Änderungen, Aktivitäten und auch Fehlermeldungen gesammelt dargestellt und können auch gefiltert werden.

> Durch einen Doppelklick oder “Rechtsklick -> Öffnen” auf eine Variable innerhalb des Meldungsfensters, kann diese verändert werden.

## EINRICHTUNG

Die Meldungen können über das Meldungen Widget oder über das Plus in der Tableiste geöffnet werden.
Hierzu im Widget einfach auf "Öffnen" klicken oder in der Tableiste mit dem Plus die Meldungen öffnen.






## FARBCODES

Im Meldungsfenster werden die unterschiedlichen Meldungen farblich dargestellt.
**Transparenter Hintergrund:**
Eine Standardnachricht. (MESSAGE)
**Grauer Hintergrund:**
Eine benutzerdefinierte Nachricht. (CUSTOM)
**Grüner Hintergrund:**
Eine Erfolgsmeldung. (SUCCESS)
**Lila Hintergrund:**
Eine Benachrichtigung. (NOTIFY)
**Gelber Hintergrund:**
Eine Warnungsmeldung. (WARNING)
**Roter Hintergrund:**
Eine Fehlermeldung. (ERROR)

## MENÜPUNKTE

### START/STOP

Startet und stoppt das Aufzeichnen der Meldungen. Somit können die bisherigen Meldungen in Ruhe analysiert werden.

### LEEREN

Löscht alle bisherigen Meldungen.

## NACHRICHTEN LIMITIEREN

Die maximale Anzahl von angezeigten Nachrichten. Wenn das Limit erreicht wird, werden die ältesten Nachrichten entfernt, bevor neue hinzugefügt werden. Dadurch kann die Liste automatisch scrollen, auch wenn die AutoScroll Funktion deaktiviert wurde.

## AUTOSCROLL

Wenn aktiviert, scrollt das Fenster automatisch mit den aktuellsten Meldungen.

## FILTER

Hier kann ausgewählt werden, welche Art von Datensatz angezeigt werden soll.

## SCHNELLFILTER

Es wird die Spalte "Absender" und "Nachricht" nach dem, was im Eingabefeld steht, durchsucht. Die Suche aktiviert sich bei Eingabe ins Suchfeld selbst, kann aber durch Klick auf "Schnellfilter" de-/aktiviert werden.
Es ist keine Suche nach mehreren Suchbegriffen möglich.

# Fehlermeldungen

Eine Fehlermeldung wird entweder direkt oder im Message Log angezeigt.
Fehlermeldungen werden im Meldungsfenster rot dargestellt.
Alle Fehlermeldungen, welche ins Message Log kommen, werden auch im Logfile im "logs"-Ordner gespeichert.



Es gilt: Wird eine Fehlermeldung nicht direkt angezeigt, wird diese im Message Log angezeigt.

### BEISPIELE FÜR EINE DIREKTE AUSGABE DER FEHLERMELDUNGEN:

Ausführen im Skript-Editor
Ausführen im WebFront und der App
Ausführen einer eigenen Aktion (Skript oder PHP-Modul) per WebFront und App
Ausführen von Web Hooks und OAuth Hooks

### BEISPIELE FÜR FEHLERMELDUNGEN IM LOG:

Ereignisse (Zyklisch, Ausgelöst)
Skript Timer
Timer von Instanzen (inkl. PHP-Module)
Fehler beim Start- und Stop-Skript im Event Control
Fehler beim Auswerten einer Register Variable

# Module Store

Über den Module Store können neue Module installiert werden. Diese Module erweitern den Funktionsumfang und ermöglichen so die Verwendung von neuen Funktionen oder weiteren Geräten.

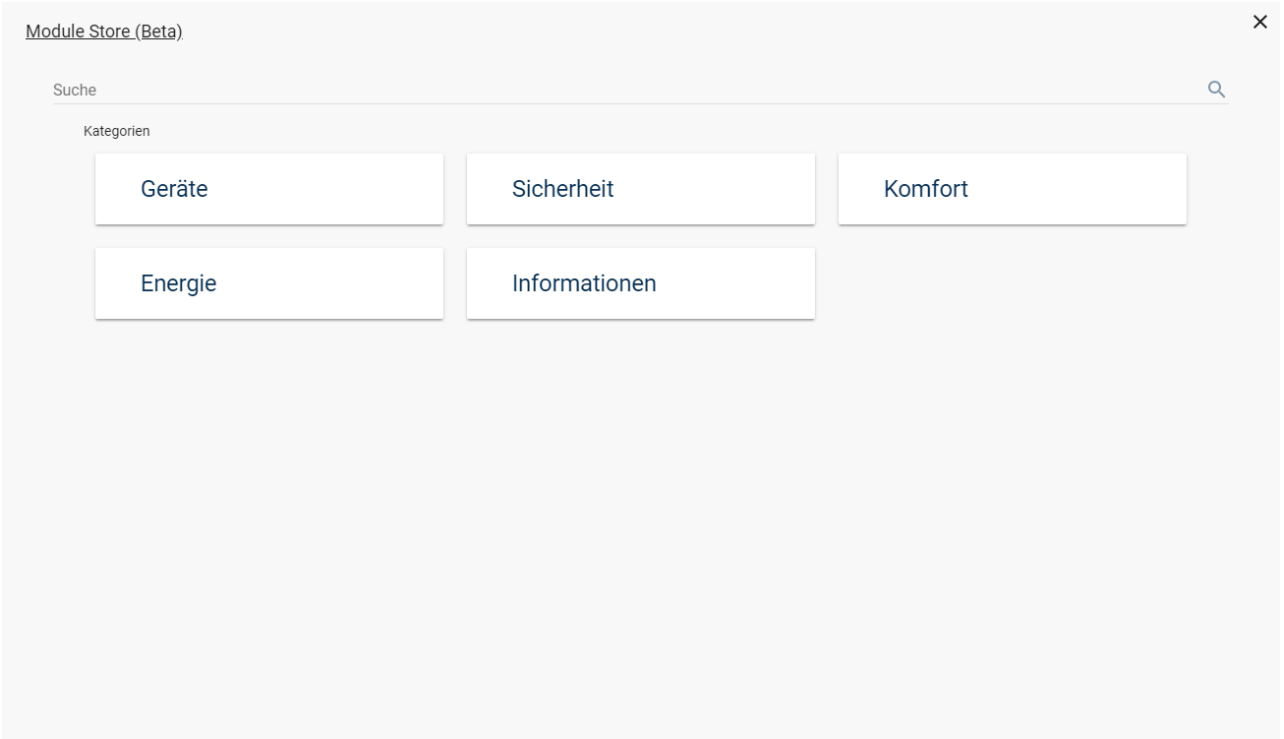
> Alle verfügbaren Module im Module Store im Browser durchsuchen.

## MODULE STORE ÖFFNEN

Der Module Store ist über den Korb oben rechts in der Konsole erreichbar.



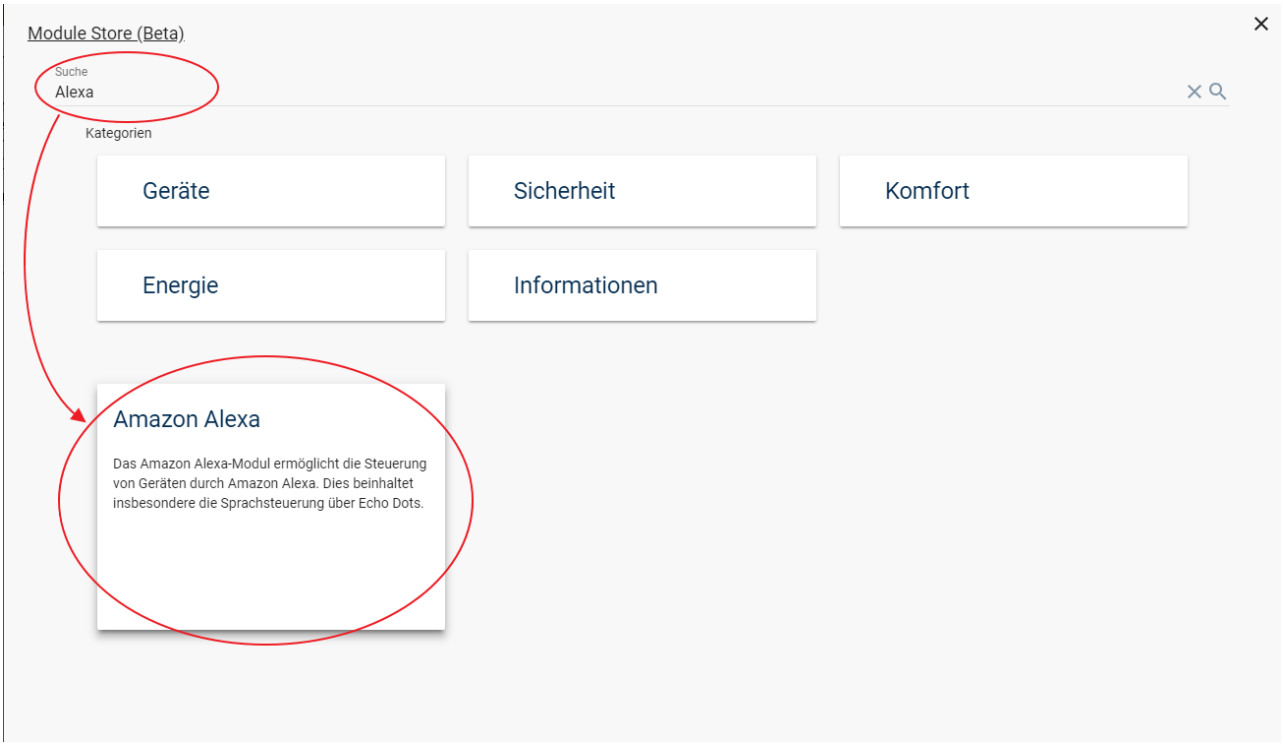

Initial zeigt der Module Store verschiedene Kategorien in denen Module angeboten werden und bietet eine Suchfunktion an. Wurden bereits Module über den Module Store installiert, so sind diese über eine entsprechende Spezialkategorie erreichbar.



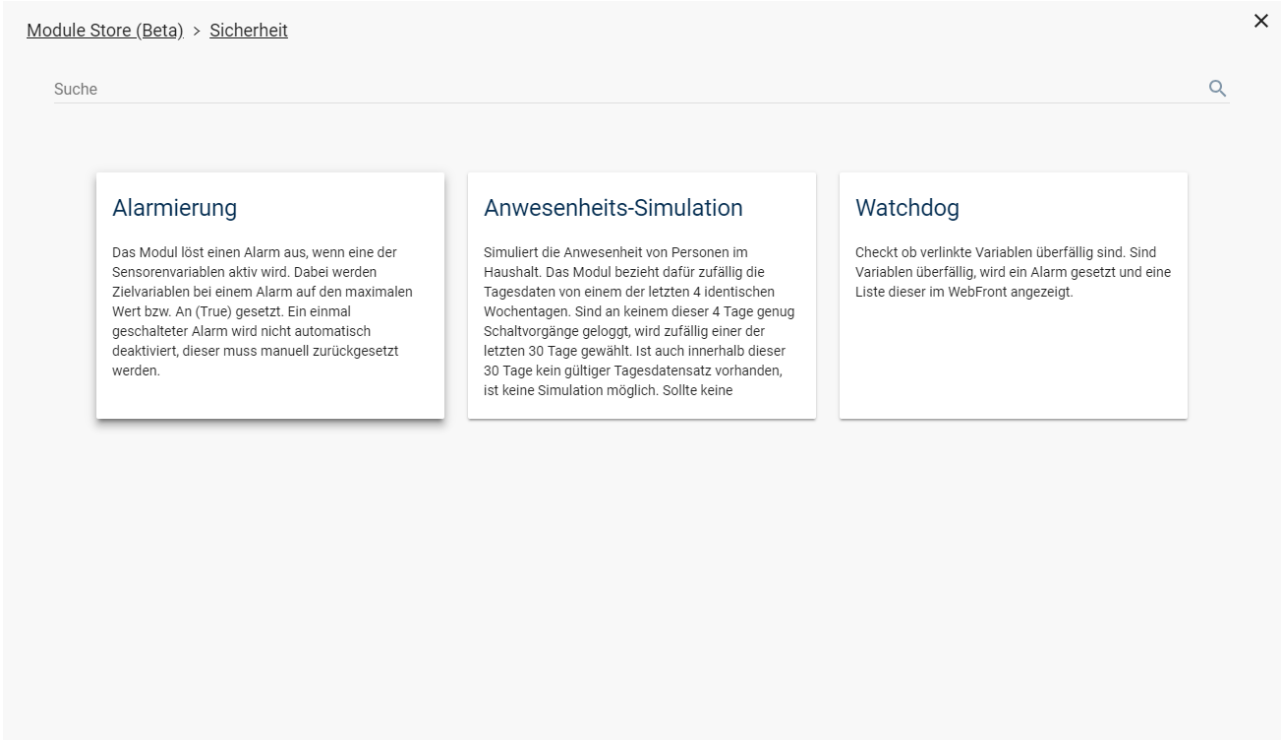

## MODULE FINDEN

Um ein passendes Modul zu finden, gibt es verschiedene Möglichkeiten. Zum einen kann die Suchfunktion verwendet werden. Hier kann ein Suchbegriff eingegeben werden und

durch die Eingabetaste oder das Klicken auf die Lupe bestätigt werden. Daraufhin werden passende Module angezeigt.




Alternativ kann eine Kategorie durch einen Klick ausgewählt werden um die darin enthaltenen Module anzuzeigen. Kategorien können Unterkategorien enthalten, welche dann ebenfalls ausgewählt werden können. Durch einen Klick auf eine vorherige Kategorie in der oberen Leiste kann man dorthin zurückkehren. Ein Klick auf "Module Store" zeigt wieder den Startbildschirm an.

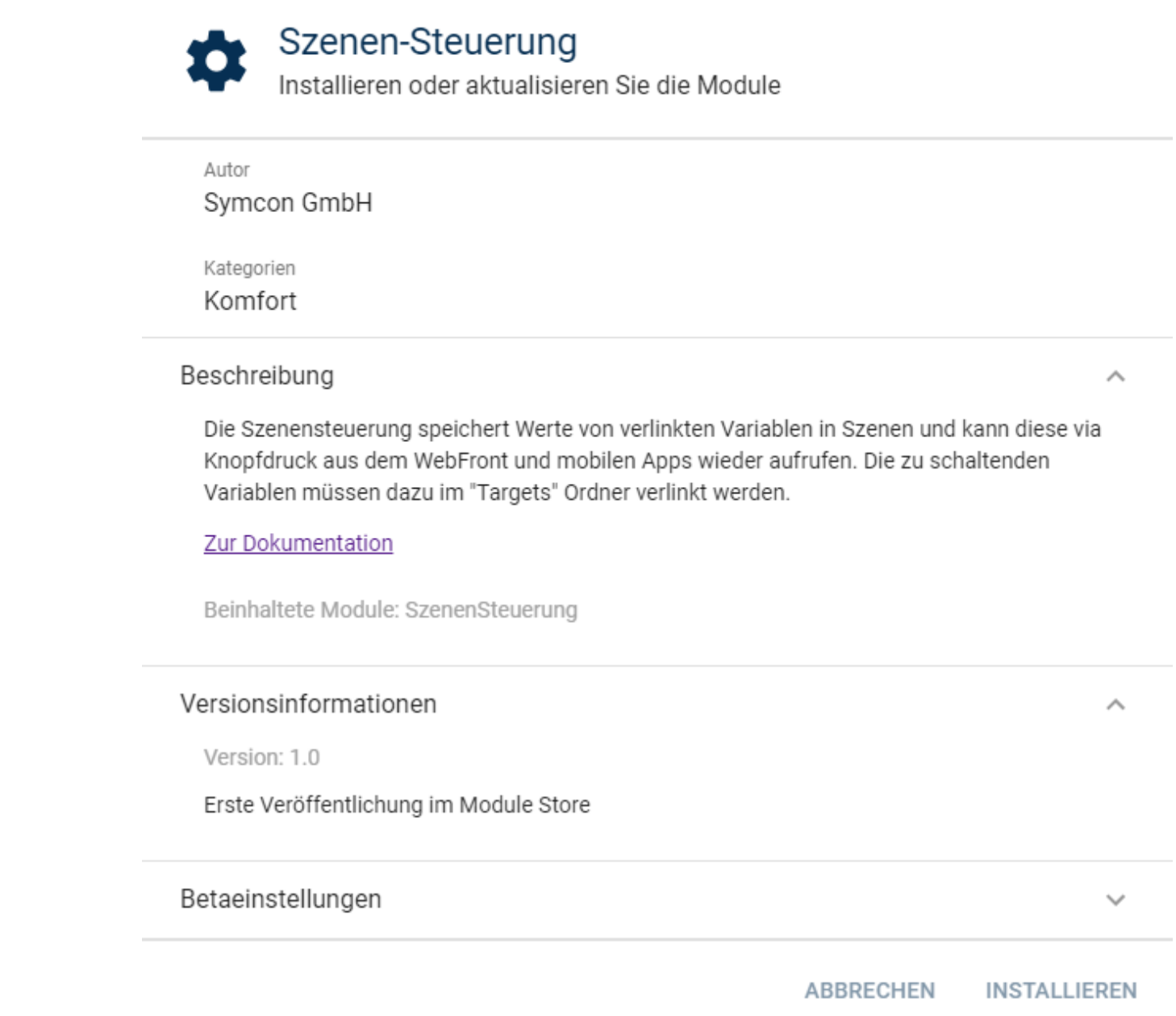



Die beiden Methoden können auch kombiniert werden. So kann in einer bestimmten Kategorie nach einem Begriff gesucht werden oder ein Suchergebnis im Nachhinein durch Kategorien eingeschränkt werden.

Wird ein Suchbegriff verwendet, so werden auch Unterkategorien durchsucht. Ohne Suchbegriff werden nur Module aus der aktuellen Kategorie angezeigt, nicht aber deren Unterkategorien.

## MODULE INSTALLIEREN

Sollen Module installiert werden, so muss zuerst die dazugehörige Karte angeklickt werden. Daraufhin öffnet sich ein Dialog, welcher die aktuell gewählten Module erörtert.



| ELEMENT | BESCHREIBUNG |
| --- | --- |
| Autor | Der Autor der Module |
| Kategorien | Die Kategorien in welchen die Module enthalten sind |
| Beschreibung | Die Beschreibung der Module. Zusätzlich ist hier ein Link zur Dokumentation und eine Liste der beinhalteten Module zu finden |
| Versionsinformation | Die aktuelle Versionsnummer und eine Beschreibung, was zu dieser Version neu ist |
| Betaeinstellungen | Falls die Module über eine Beta-Version verfügen, kann hierüber auf die Beta-Version gewechselt werden |

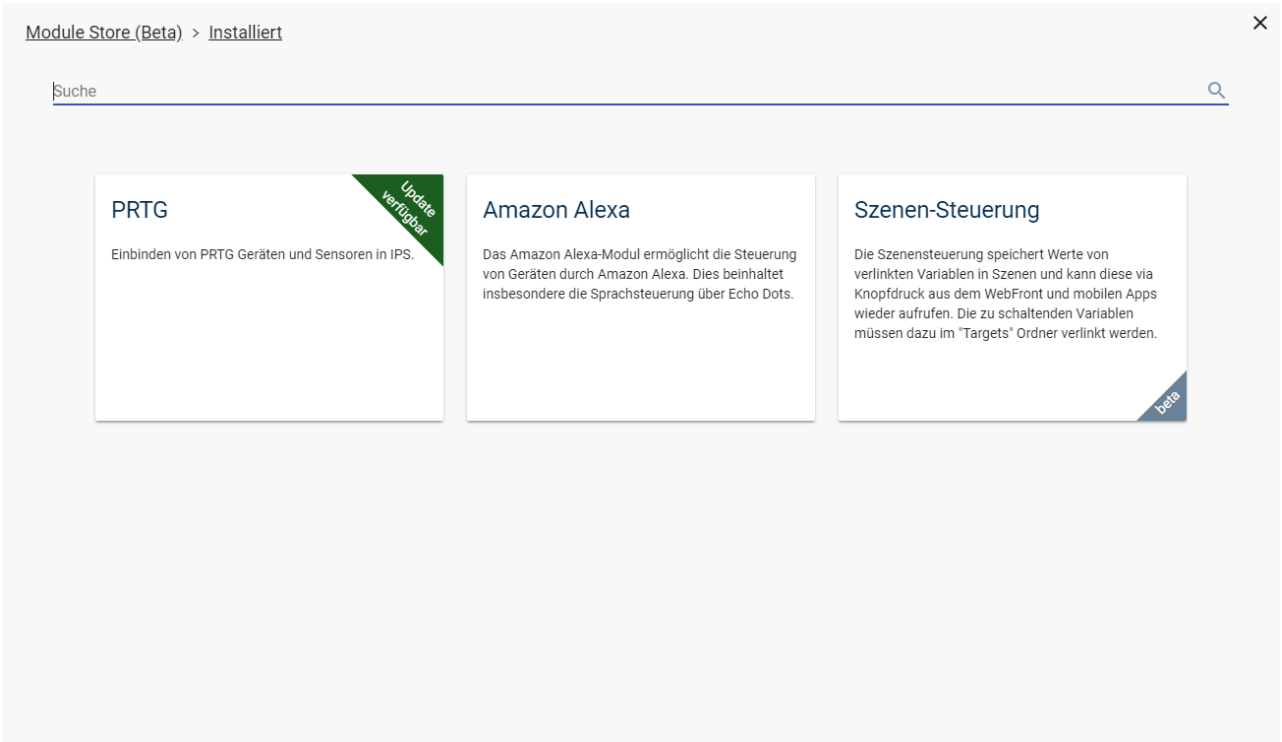
Am unteren Ende des Dialogs befindt sich der Button "Installieren" um die Module zu installieren und "Abbrechen" um den Dialog wieder zu schließen.
Soll eine Beta-Version installiert werden, so können die Betaeinstellungen ausgeklappt werden und auf "Zu Beta wechseln" geklickt werden. Daraufhin werden die Modulinformationen auf die Beta-Version angepasst und diese kann durch einen Klick auf "Installieren" installiert werden.



Beta-Versionen werden nicht durch das Symcon-Team geprüft.

## MODULE AKTUALISIEREN UND ENTFERNEN

Wurden Module installiert, so erscheint im Startbildschirm des Module Stores eine Spezialkategorie "Installiert". Hier werden alle installierten Module aufgelistet. Gibt es für Module neue Versionen, wird dies auch hervorgehoben.




Wird ein installiertes Modul angeklickt, so erscheint wieder der Installationsdialog. Nun hat man zusätzlich die Möglichkeit die Module durch Klick auf "Entfernen" zu löschen.

Gibt es eine neue Version, kann man diese per "Aktualisieren" installieren. Möchte man im Nachhinein den Kanal wechseln, also von der regulären Stable-Version auf Beta oder anders herum, so kann dies über die Betaeinstellungen getan werden und per "Kanal wechseln" bestätigt werden. Ist die aktuelle Version installiert, so kann man diese per "Erneut installieren" noch einmal herunterladen.

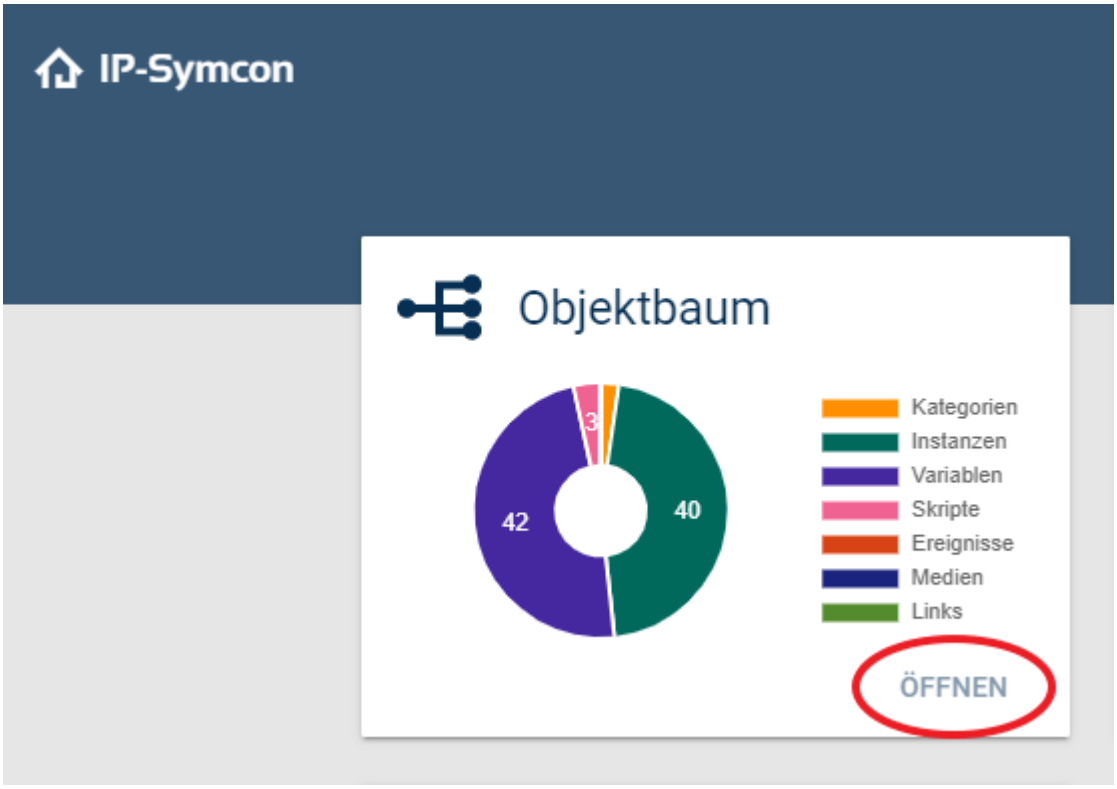
# Objektbaum

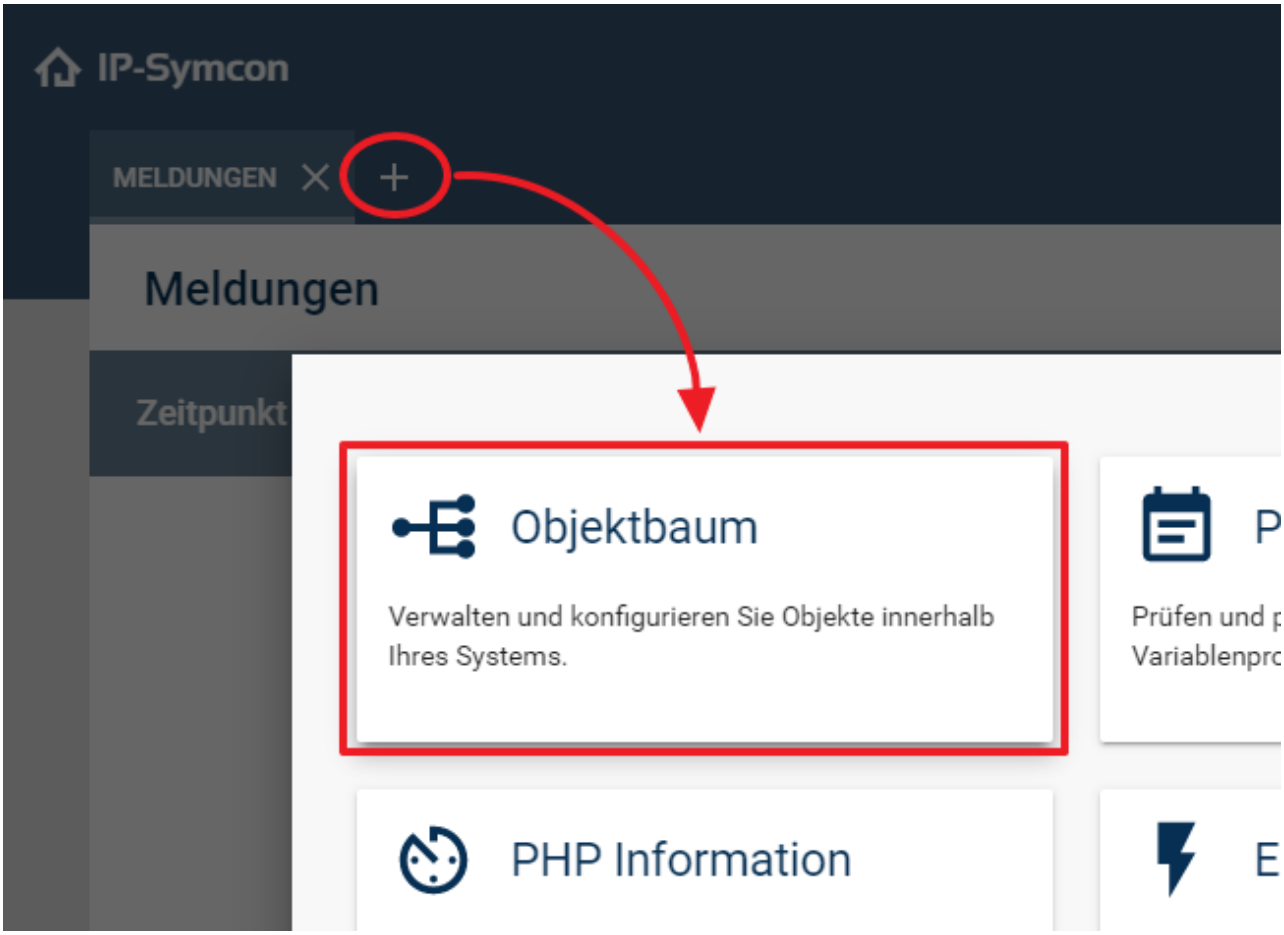
Der Objektbaum dient zur Übersicht und Verwaltung aller Objekte und Instanzen. Die hier angelegte Struktur wird später auch als Grundlage zur Darstellung für das WebFront genutzt. Das Verschieben von Objekten geschieht via Drag&Drop.

Die Erklärung zum Kontextmenü der verschiedenen Instanzen kann in den Grundlagen nachgeschlagen werden.

## EINRICHTUNG

Der Objektbaum kann über das Objektbaum Widget oder über das Plus in der Tableiste geöffnet werden.
Hierzu im Widget einfach auf "Öffnen" klicken oder in der Tableiste mit dem Plus den Objektbaum öffnen.






## FARBCODES

Im Objektbaum wird der Status der Objekte mit Farbcodes dargestellt.
**Schwarze Schrift:**
Alles normal und das Objekt wird im WebFront angezeigt.
**Hellgraue Schrift:**
Alles normal und das Objekt wird nicht im WebFront angezeigt.
**Graues Ausrufezeichen:**
Das Objekt ist inaktiv.
**Hellgraues Ausrufezeichen:**
Eine übergeordnete Instanz des Objekts ist inaktiv.
**Rotes Ausrufezeichen:**
Das Objekt hat einen Fehlerzustand.
**Hellrotes Ausrufezeichen:**
Eine übergeordnete Instanz hat einen Fehlerzustand.

## WEITERE FUNKTIONALITÄTEN

Wenn im Objektbaum über eine Uhrzeit oder ein Statusicon gehovert wird, erscheint ein Tooltip, welcher z.B. das gesamte Datum oder die Art der Störung anzeigt.

## MENÜPUNKTE

### PLUS-KNOPF

Über den Dialog "+" unten Rechts können neue Objekte erstellt werden. Alternativ kann durch ein Rechtsklick an der gewünschte Position im Objektbaum ein Objekt an dieser Position hinzugefügt werden. Die jeweiligen Objekttypen werden in den Grundlagen genauer beschrieben.

### ID

Mithilfe des Eingabefelds kann eine ObjektID gesucht werden. Bei Fund springt der Objektbaum automatisch zu der gesuchten ID und wählt diese aus.

### FILTER

Bei Eingabe in das Suchfeld wird der Filter automatisch aktiviert. Dieser kann durch Klick auf "Filter" wieder ein- und zugeschaltet werden. Es wird in der Spalte “Name” gesucht. Falls die Spalte "Beschreibung" sichtbar ist, wird auch in dieser gesucht.

### TYPEN

Hier kann die Sichtbarkeit bestimmter Objekttypen im Objektbaum, an- und ausgeschaltet werden, um die Übersicht oder den Fokus auf bestimmte Typen zu setzen. Ist etwas ausgeblendet wird dieser Button grau unterlegt.

### SPALTEN

Übersichtsfunktionalität, um die angezeigten Spalten zu aktivieren und deaktivieren.

# Objektbaum (Physikalisch)

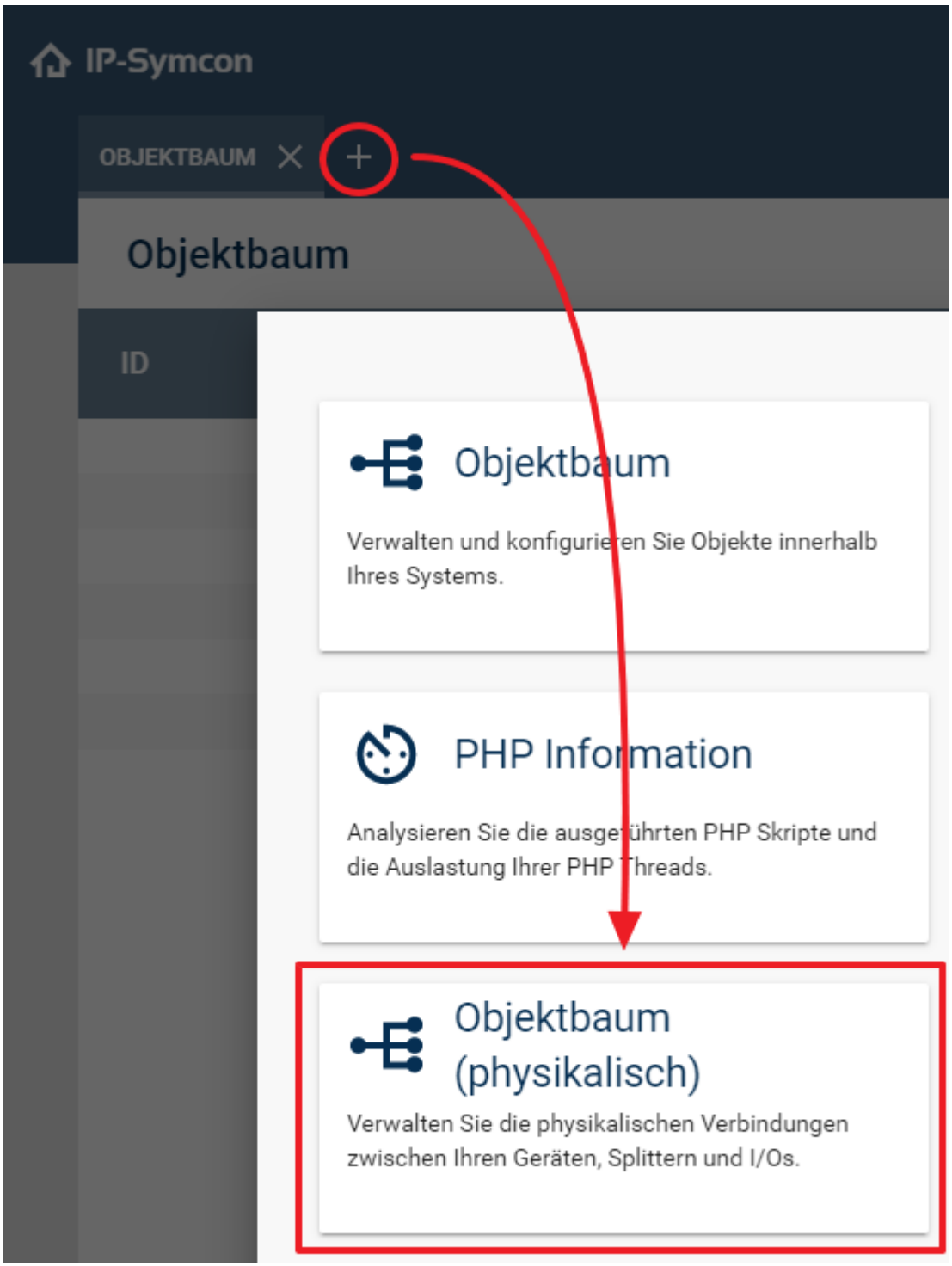
Die physikalische Ansicht des Objektbaums zeigt auf der obersten Ebene nur unverbundene und I/O Instanzen an. Im I/O Bereich werden die Geräte in der Struktur der tatsächlichen physikalischen Verbindung dargestellt. Die Struktur wird unter Übergeordnete Instanzen veranschaulicht. (I/O -> Gateway ->Gerät). Es werden nur Instanzen angezeigt.
Unverbundene Instanzen beinhaltet solche, welche nicht physisch vorhanden sind.



Die Erklärung zum Kontextmenü der verschiedenen Instanzen kann in den Grundlagen nachgeschlagen werden.

## EINRICHTUNG

Der Objektbaum kann über das Plus in der Tableiste geöffnet werden.




## FARBCODES

Im Objektbaum wird der Status der Objekte mit Farbcodes dargestellt.
**Schwarze Schrift:**
Alles normal und das Objekt wird im WebFront angezeigt.
**Hellgraue Schrift:**
Alles normal und das Objekt wird nicht im WebFront angezeigt.
**Graues Ausrufezeichen:**
Das Objekt ist inaktiv.
**Hellgraues Ausrufezeichen:**
Eine übergeordnete Instanz des Objekts ist inaktiv.
**Rotes Ausrufezeichen:**
Das Objekt hat einen Fehlerzustand.

Hellrotes Ausrufezeichen:
Eine übergeordnete Instanz hat einen Fehlerzustand.

## WEITERE FUNKTIONALITÄTEN

Wenn im Objektbaum über eine Uhrzeit oder ein Ausrufezeichen gehovert wird, erscheint ein Tooltip, welcher z.B. das gesamte Datum oder die Art der Störung anzeigt.

## MENÜPUNKTE

### ID

Mithilfe des Suchdialogs kann eine ObjektID gesucht werden. Bei Fund springt der Objektbaum automatisch zu der gesuchten ID und wählt diese aus.

### FILTER

Bei Eingabe in das Suchfeld wird der Filter automatisch aktiviert. Dieser kann durch Klick auf "Filter" wieder ein- und zugeschaltet werden. Es wird nur in der Spalte “Name” gesucht.

### TYPEN

Hier kann die Sichtbarkeit bestimmter Objekte im Objektbaum, an- und ausgeschaltet werden, um die Übersicht oder den Fokus auf bestimmte Typen zu setzen. Ist etwas ausgeblendet wird dieser Button grau unterlegt.

### SPALTEN

Übersichtsfunktionalität, um die angezeigten Spalten zu aktivieren und deaktivieren.

# Pro Konsole

Die Pro Konsole ist das Experten-Tool zum Einrichten und Konfigurieren der gesamten Logik des IP-Symcon Servers. Diese besitzt alle Features, welche auch die Webbasierte Verwaltungskonsole hat, und noch weiterführende Expertenoptionen.

## INSTALLATION

Die Pro Konsole kann für Windows, MacOS oder Linux Ubuntu installiert werden.



Die Pro Konsole wurde mit der IP-Symcon Version 5.4 eingeführt, funktioniert aber mit Server Versionen ab 5.0. Bei älteren Versionen stehen allerdings nicht alle Funktionen der Pro Konsole zur Verfügung.

### WINDOWS

Die aktuelle Pro Konsole für Windows kann über diesen Downloadlink heruntergeladen werden.

### MACOS

Die aktuelle Pro Konsole für MacOS kann über diesen Downloadlink heruntergeladen werden.

### UBUNTU

```
echo "deb [arch=amd64] http://apt.symcon.de/ stable ubuntu" | sudo
tee /etc/apt/sources.list.d/symcon.list
```

GPG-Schlüssel hinzufügen:

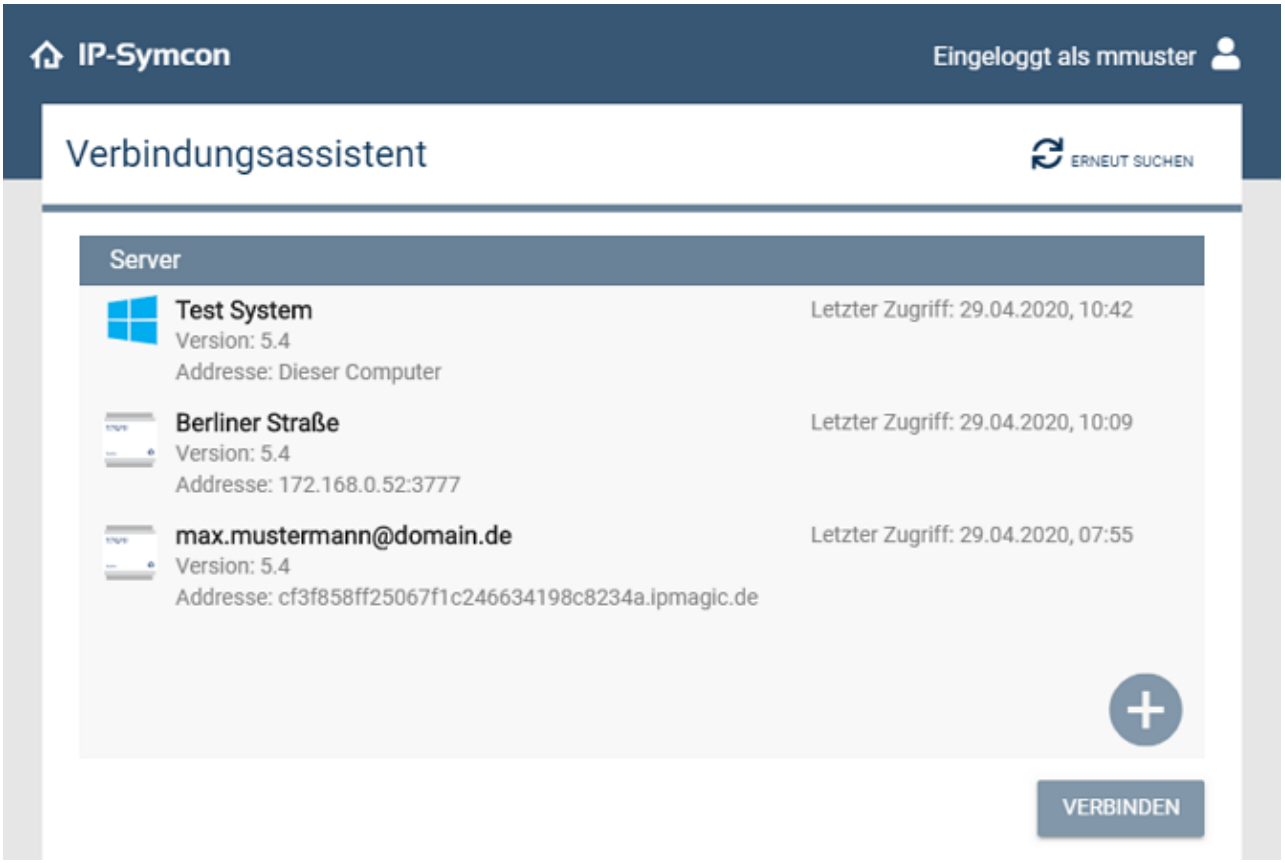
```
wget -qO - https://apt.symcon.de/symcon.key | sudo apt-key add -
```

Folgende Befehle ausführen:

```
sudo apt-get update
sudo apt-get install symcon-console
```

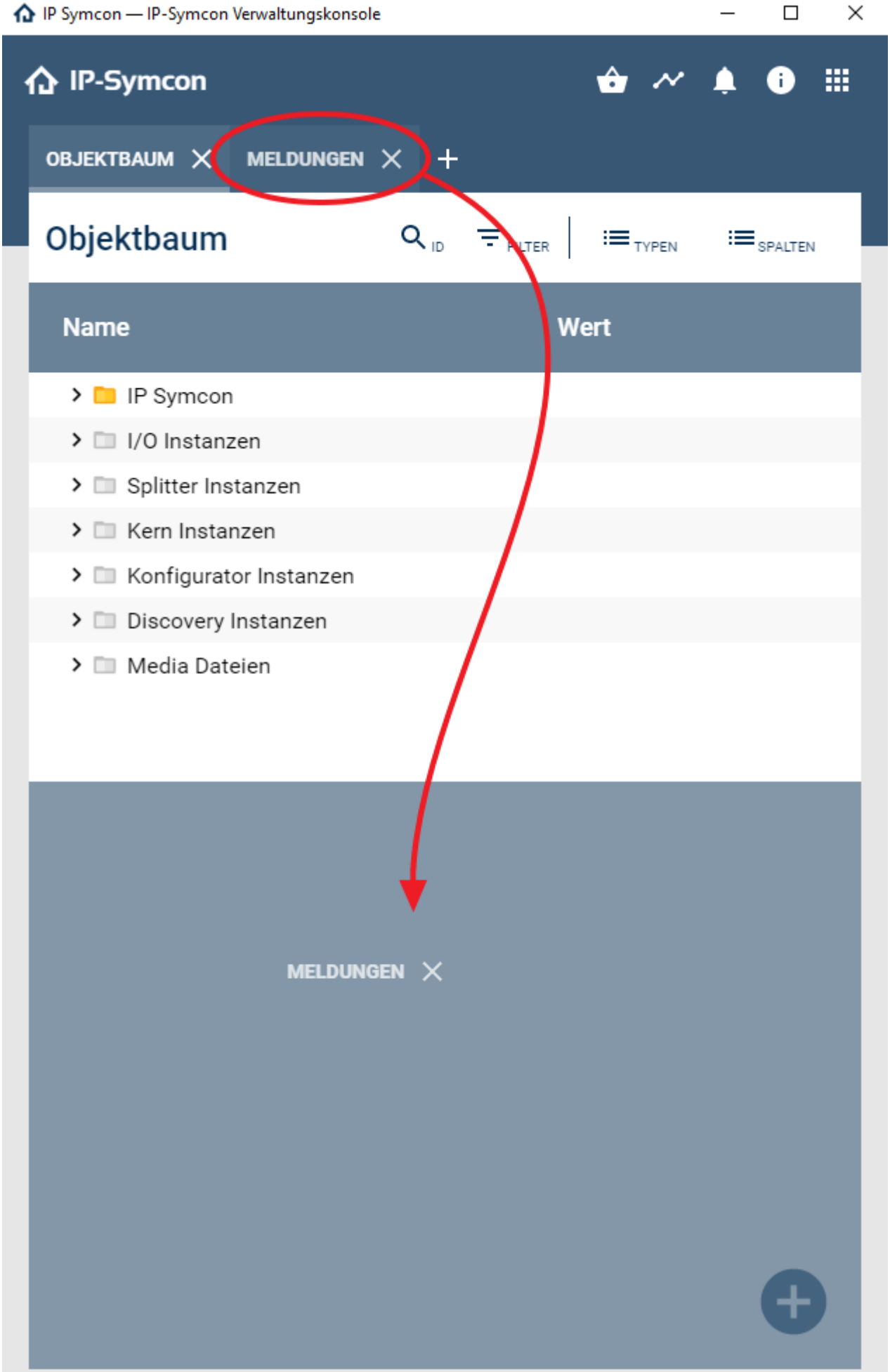
## VERBINDEN MIT DEM IP-SYMCON SERVER

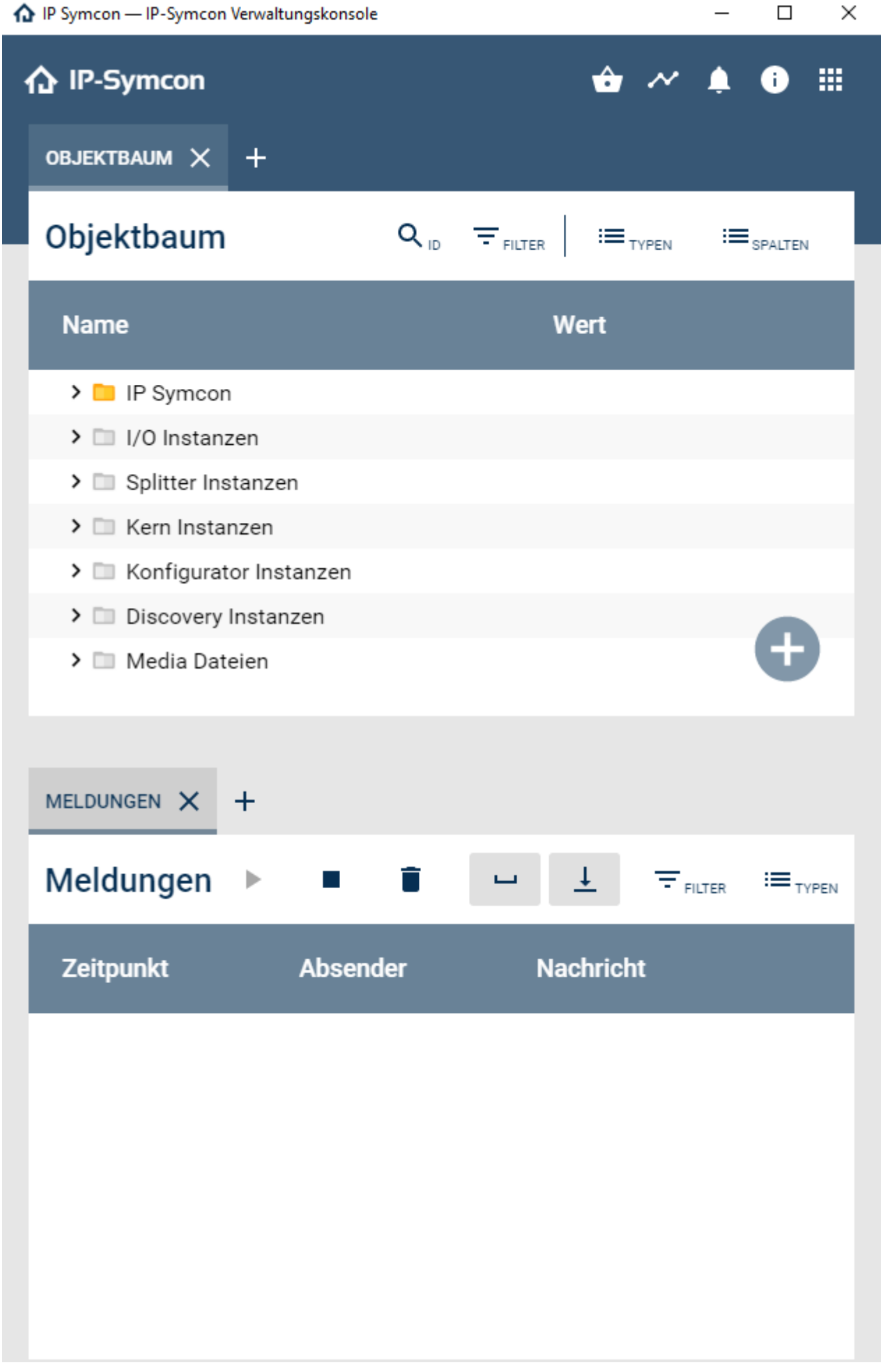
Die Pro Konsole nutzt den Verbindungsassistenten. Mit diesem kann ein Server ausgewählt und mit "Verbinden" oder Doppelklick geöffnet werden.




## AN- UND ABDOCKEN

In der Pro Konsole können einzelne Tabs beliebig an- und abgedockt werden. So kann ein individuell angepasstes Arbeitsfeld aufgebaut und alles in einem Blick behalten werden. Zusätzlich können einzelne Tabs mit einem Rechtsklick und dann "In neuem Fenster öffnen" separat angezeigt werden.


IP Symcon — IP-Symcon Verwaltungskonsole
IP-Symcon
OBJEKTBAUM
MELDUNGEN
Objektbaum
ID
FILTER
TYPEN
SPALTEN
Name
Wert
IP Symcon
I/O Instanzen
Splitter Instanzen
Kern Instanzen
Konfigurator Instanzen
Discovery Instanzen
Media Dateien
MELDUNGEN


IP Symcon — IP-Symcon Verwaltungskonsole
IP-Symcon
OBJEKTBAUM
Objektbaum
ID
FILTER
TYPEN
SPALTEN
Name
Wert
IP Symcon
I/O Instanzen
Splitter Instanzen
Kern Instanzen
Konfigurator Instanzen
Discovery Instanzen
Media Dateien
MELDUNGEN
Meldungen
FILTER
TYPEN
Zeitpunkt
Absender
Nachricht

## START MIT PARAMETERN

Üblicherweise öffnet die Pro Konsole beim Aufruf direkt den Verbindungsassistenten. Dieses Verhalten kann durch Parameter angepasst werden.

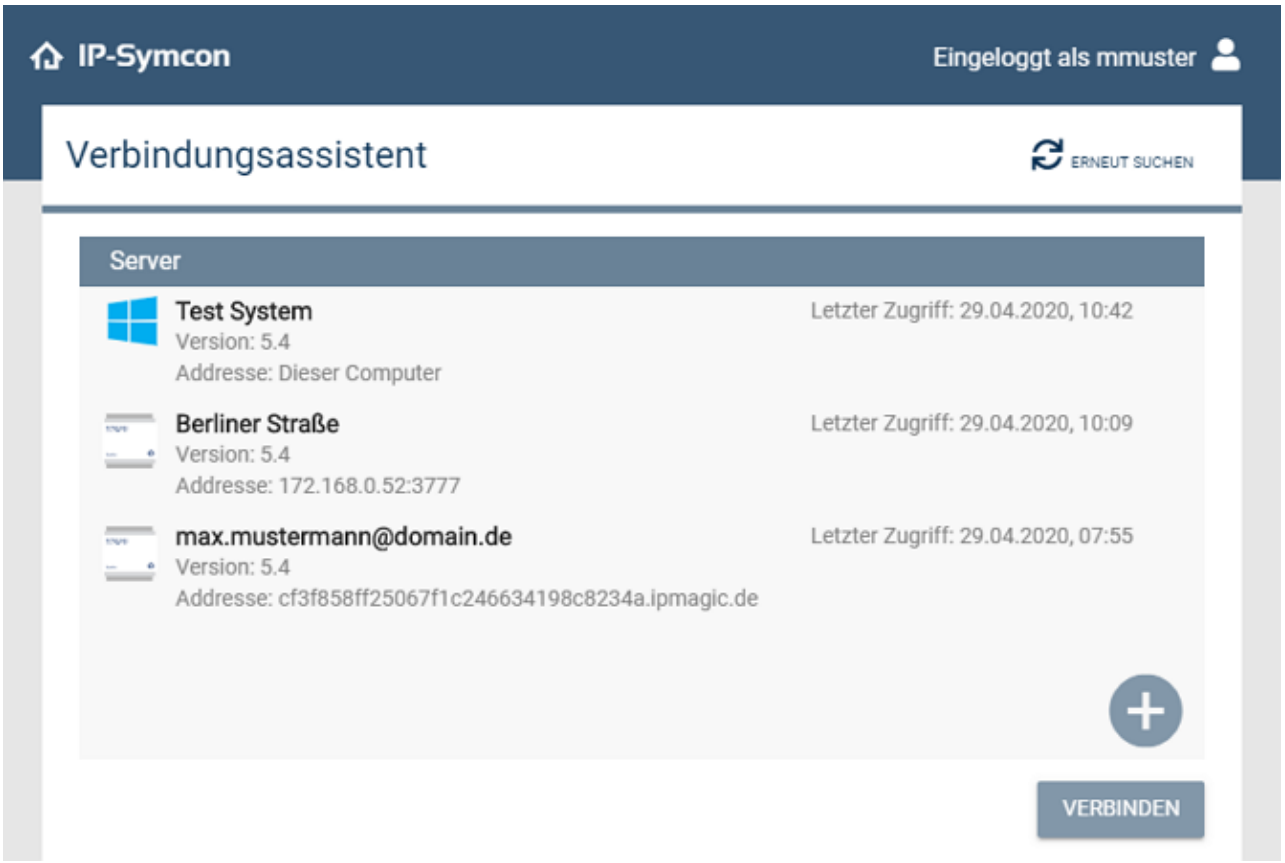
| PARAMETER | BESCHREIBUNG | BEISPIEL FÜR WINDOWS |
|---|---|---|
| --server=[Konsole] | Verbindet direkt mit einem Server. Für [Konsole] muss hierbei die Adresse der webbasierten Verwaltungskonsole des Servers verwendet werden. | "IP-Symcon Management Console.exe" --server=http://127.0.0.1:3777/console/ |

# Verbindungsassistent

Der Verbindungsassistent hilft bei der Verbindung der Pro Konsole zu einem IP-Symcon Server. Die verfügbaren Server werden dabei in einer Liste angezeigt.

## SERVER

Es werden alle Server, die im lokalen Netzwerk gefunden werden oder in der Lizenzverwaltung eingetragen sind, dargestellt. Für die Anzeige, der in der Lizenzverwaltung eingetragenen Servern muss sich mit dem Symcon Account eingeloggt werden. Die Anmeldung erfolgt über einen Klick auf das Personenicon oben rechts. Dort müssen Benutzername und Kennwort des persönlichen Bereichs der Symcon-Homepage eingegeben werden. Es ist zu beachten, dass dies Konto separat vom Communityforum und dem Fernzugriff ist. Mit "Erneut suchen" kann nach neuen Servern im lokalen Netzwerk gescannt werden.




## SERVER HINZUFÜGEN

Es können auch manuell Server hinzugefügt werden. Dazu muss auf das "+" geklickt werden und Name, Adresse und Port angegeben werden. Dies sollte genutzt werden, wenn zum Beispiel ein DynDNS genutzt wird. Sollte der Connect Control genutzt werden ist die Verwendung der Lizenzverwaltung empfohlen.

## KEINE SERVER GEFUNDEN

Sollten keine Server in der Serverliste sichtbar sein kann dies mehrere Gründe haben.

### IP-Symcon Server aus der Lizenzverwaltung tauchen nicht auf

Es sollte überprüft werden, ob der korrekte Benutzer eingeloggt ist. Ist ein Benutzer angemeldet, so steht neben dem Personenicon "Eingeloggt als ...". Steht dies nicht dort, so ist kein Benutzer angemeldet und demnach können keine Lizenzen der Lizenzverwaltung angezeigt werden.

### IP-Symcon Server ist nicht eingeschaltet/gestartet

Bitte IP-Symcon am Server starten. Überprüfen, ob der Server korrekt angeschlossen und gestartet ist.

### IP-Symcon Server ist nicht korrekt mit dem Netzwerk verbunden

Es sollte überprüft werden, ob eine Netzwerkverbindung zwischen beiden Computern verfügbar ist.

### Firewall Einstellungen nicht korrekt

Sowohl Client-, als auch Serverseitig muss der Zugriff in der Firewall für Port 1900/UDP erlaubt sein.

Für die Verbindung mit der Verwaltungskonsole muss zusätzlich, im Normalfall, Port 82/TCP bzw. 3777/TCP erlaubt sein.

# Kachel Visualisierung

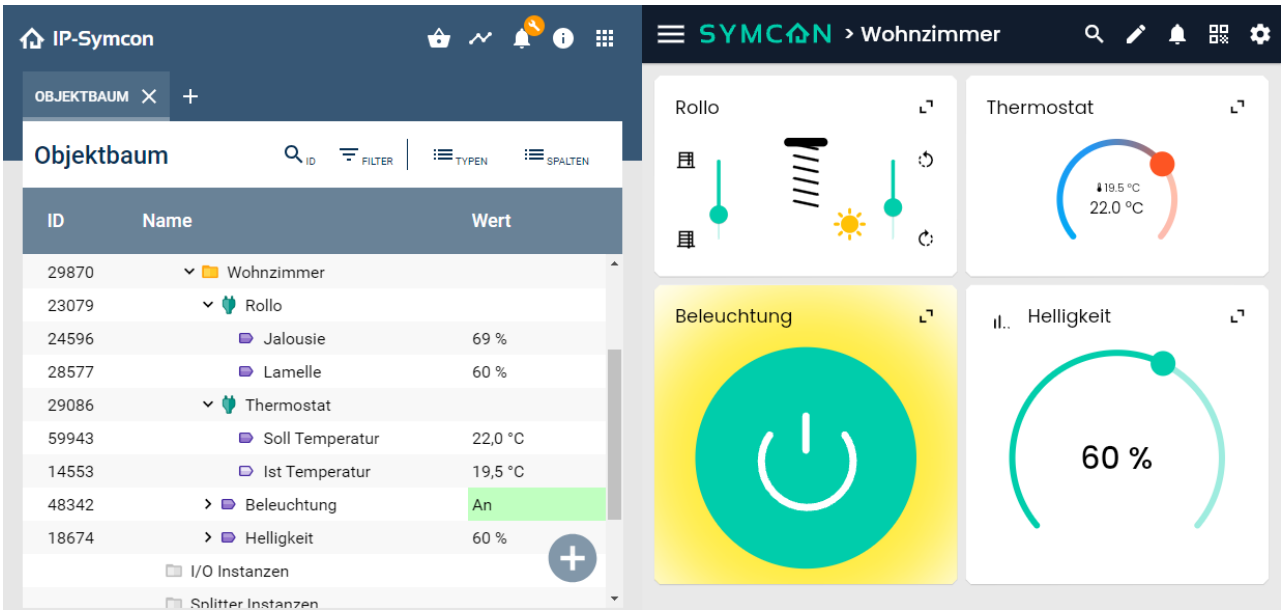
Benötigt: IP-Symcon >= 7.0
Die Kachel-Visualisierung bietet eine Darstellung Ihres Systems, welche den gesamten Objektbaum oder auch nur einen Teil davon visualisiert. Die Visualisierung kann via Browser oder via Apps genutzt werden.

## VERBINDEN MIT DER VISUALISIERUNG

Das Verbinden ist einer der wenigen Punkte, in denen sich Apps und Browser unterscheiden:
Verbinden mit Apps
Verbinden mit Browser

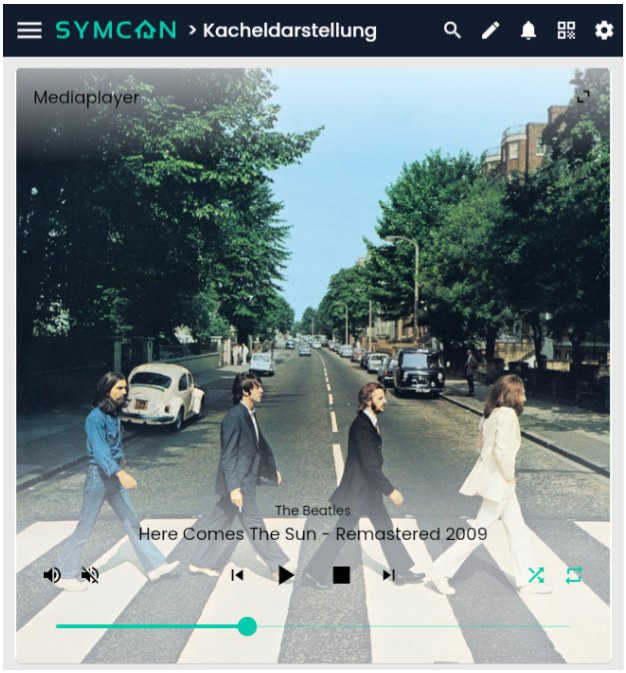
## HAUPTANSICHT

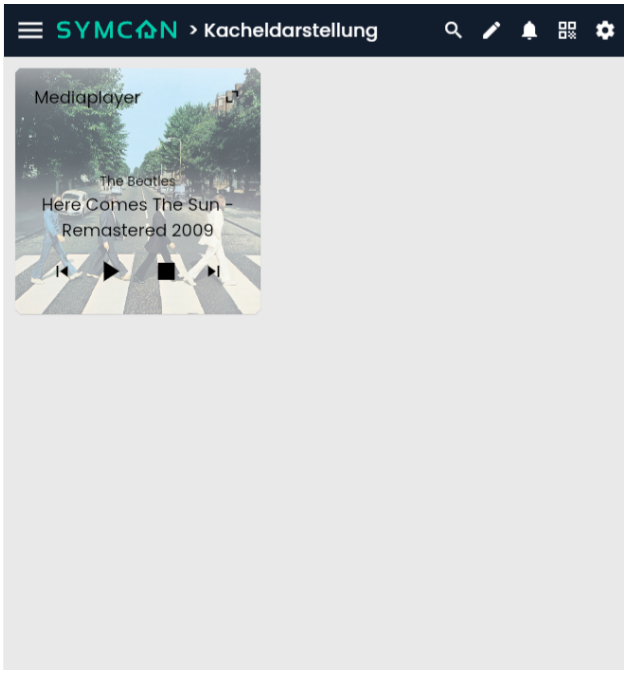
Die Visualisierung stellt Objekte aus IP-Symcon als Kacheln dar und orientiert sich dabei an dem Objektbaum. Eine Seite der Visualisierung entspricht somit dem Inhalt einer Kategorie. Wie die einzelnen Objekttypen angezeigt werden ist in der Objekt-Darstellung beschrieben




## KACHELN

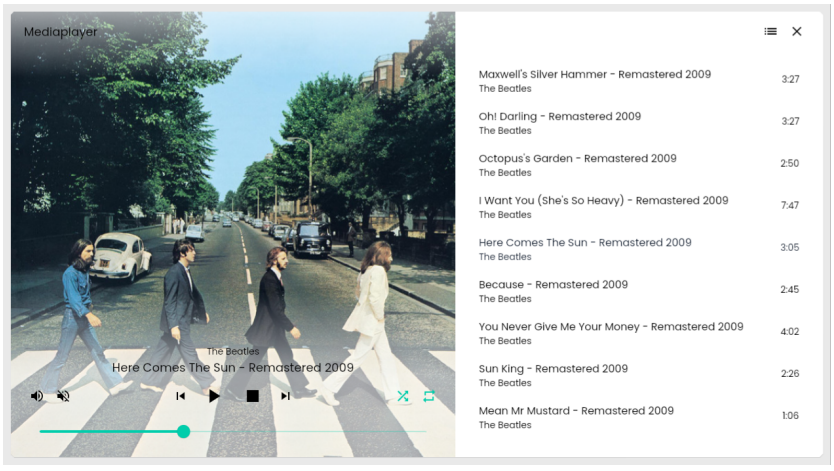
Kacheln haben verschiedene Darstellungsarten.
Minimierte Ansicht
Wenn Kacheln unter einer Kategorie neben anderen angezeigt werden, nutzen diese die minimierte Darstellung. Diese sind in der Objekt-Darstellung näher beschrieben. Die Größe Der Kachel kann ebenfalls einen Einfluss auf den Inhalt der Kachel haben.

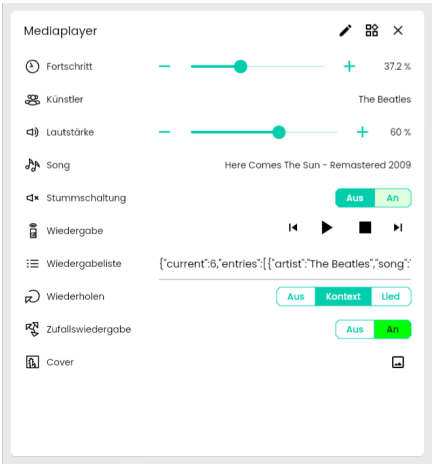



### Maximierte Ansicht

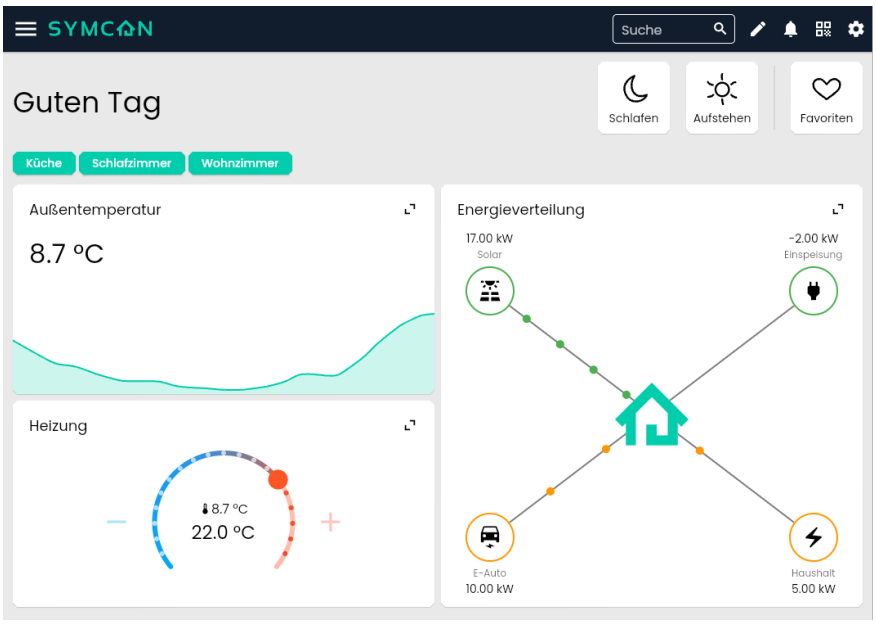
Jede Kachel kann über die zwei Pfeile in der oberen rechten Ecken maximiert werden. Je nach Objekttyp und Konfiguration steht eine erweiterte Darstellung zur Verfügung. Gibt es keine werden das Objekt selbst oder alle untergeordneten Objekte in einer Liste angezeigt. Auch wenn es eine erweiterte Darstellung gibt kann jederzeit über das Listen Icon in der oberen rechten Ecke zur Listendarstellung gewechselt werden.

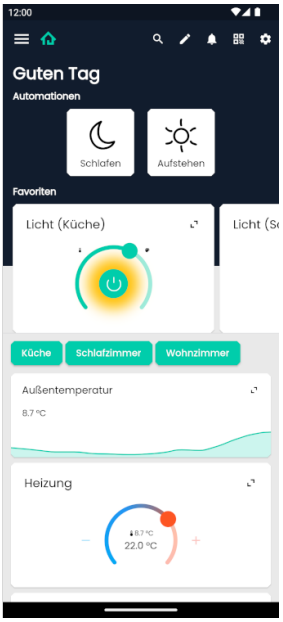



## ELEMENTE DER VISUALISIERUNG

Die Visualisierung ist in verschiedene Bereiche unterteilt:




### Appbar

Am oberen Bereich des Bildschirms befindet sich die Appbar. Auf der linken Seite kann über das Navigationsmenü eine Übersicht der Kategorien aufgerufen werden. Auf der rechten Seite befindet sich standardmäßig die Suche und Schaltflächen für den Editiermodus, Benachrichtigungen, das Teilen der Visualisierung und die Einstellungen. In der Instanzkonfiguration können diese Optionen deaktiviert oder noch weitere Darstellungen wie etwa das aktuelle Datum hinzugefügt werden.

### Startbereich

Unter der Appbar ist der Startbereich. Der Inhalt kann in der Instanzkonfiguration angepasst werden. Hier wird standardmäßig ein Begrüßungssatz angezeigt. Zusätzlich können eine Reihe von Automationen und Favoriten hier angezeigt werden. Bei schmaleren Bildschirmen wie etwa Smartphones wird die Darstellung leicht angepasst, um den verfügbaren Platz besser zu nutzen.

### Kategorieleiste

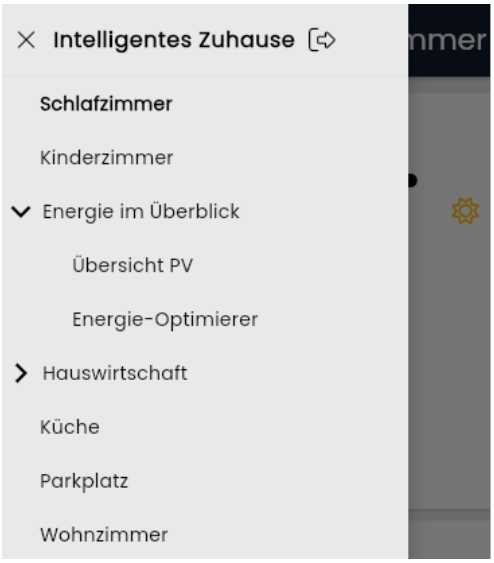
Die Kategorieleiste fasst alle Kategorien, die sich in der aktuellen Kategorie befinden zusammen. Bei Bedarf bricht die Leiste um, damit alle Kategorien angezeigt werden können. In der Instanzkonfiguration kann die Kategorieleiste ausgeblendet werden, wodurch Kategorien mit den anderen Objekten darunter im Objektbereich angezeigt werden.

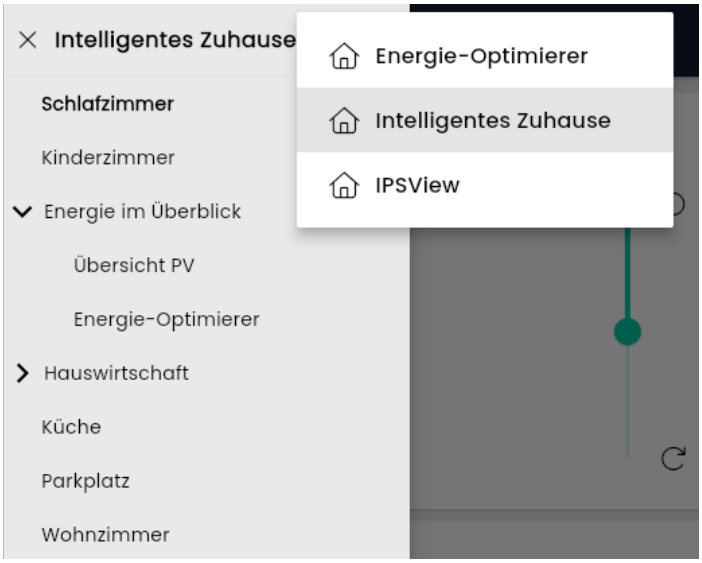
### Objektbereich

Hier werden alle Objekte aus IP-Symcon wie etwa Variablen und Instanzen als Kacheln dargestellt. Die Position und Größe dieser kann jederzeit verändert werden. Ebenso kann der Inhalt der einzelnen Kacheln angepasst werden. Diese Informationen werden für verschiedene Bildschirmgrößen in Darstellungprofilen gespeichert. Näheres hierzu ist in Abschnitt Individualisierung beschrieben.

### Seitenleiste

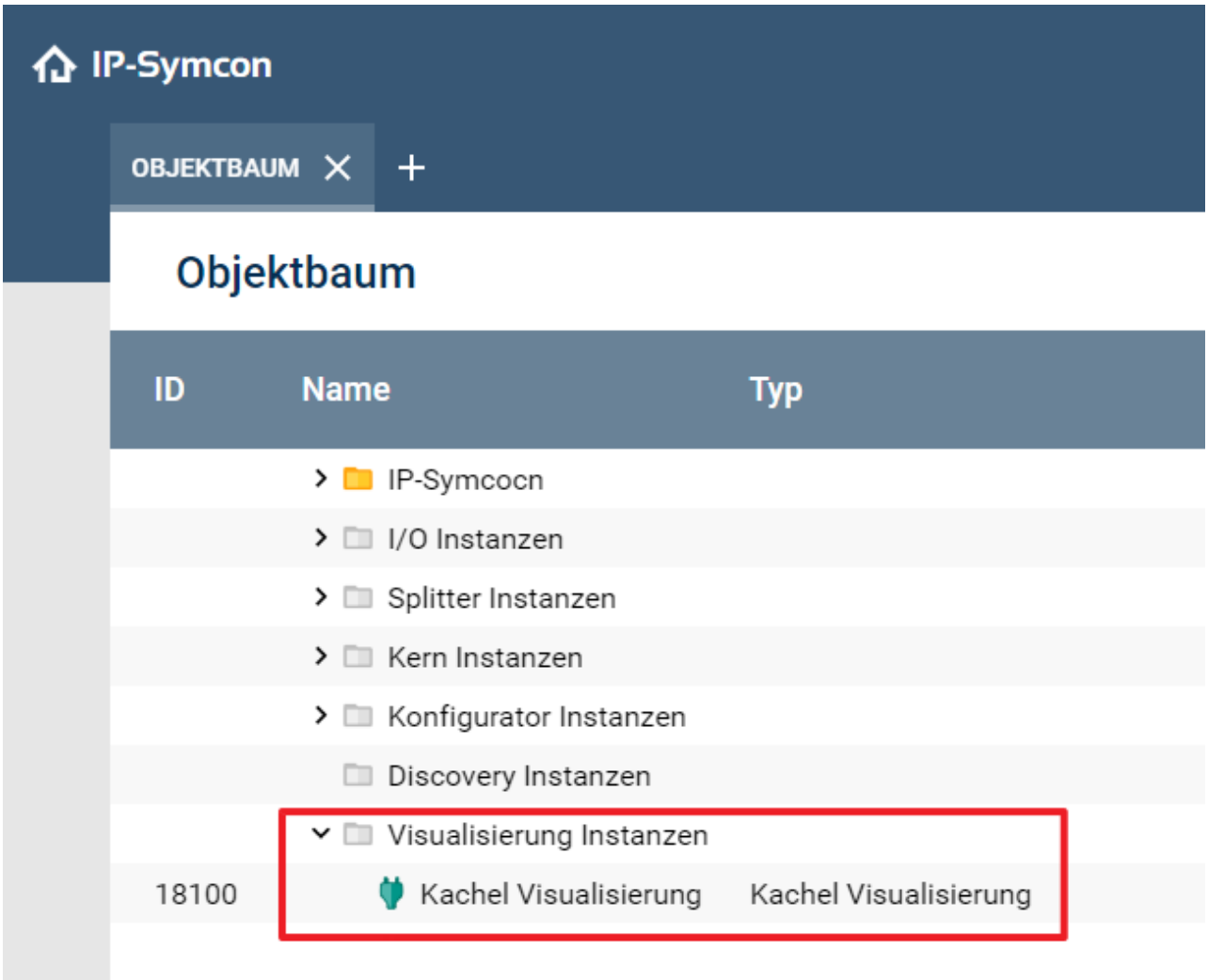
Die Seitenleiste kann über das Icon in der oberen linken Ecke geöffnet werden. Hier werden alle Kategorien in einer Baumansicht darstellt und können zur Navigation genutzt werden. Neben dem Namen der Visualisierung kann über eine Schaltfläche zwischen den verfügbaren Visualisierungen des Servers gewechselt werden.




## EINRICHTUNG

Um die Kachel Visualisierung einzurichten muss eine Kachel Visualisierung Instanz vorhanden sein. Standardmäßig wird mit der Installation eine Instanz erstellt und befindet sich im Objektbaum unter "Visualisierung Instanzen".




Sollen weitere Kachel Visualisierungen eingerichtet werden, so kann dieses im Objektbaum über das "+" -> "Instanz" -> "Kachel Visualisierung" hinzugefügt werden. Dies

erstellt eine neue Kachel Visualisierung, welche komplett unabhängig konfiguriert werden kann.

## KONFIGURATION

Grundsätzlich läuft die Kachel Visualisierung ohne weitere Konfiguration. Es können allerdings diverse Einstellungen gemacht werden. Informationen über die einzelnen Einstellungen und Konfigurationsmöglichkeiten können unter Instanzkonfiguration gefunden werden. Die Konfiguration kann durch einen Doppelklick im Objektbaum auf die Kachel Visualisierung geöffnet werden.

## MEHRERE VISUALISIERUNGEN

In IP-Symcon Basic ist eine Visualisierung verfügbar. In der IP-Symcon Professional Edition können insgesamt bis zu fünf Kachel Visualisierungen erstellt werden, die eigenständig über die Visualisierung Startseite aufgerufen und bei Bedarf mit einem Passwort abgesichert werden können. Eine Visualisierung kann als eigenständige Einheit - für z.B. ein Visualisierungsobjekt - erstellt werden, eine spezielle Ansicht für den Wartungstechniker darstellen oder einfach nur eine sicherheitsspezifische Trennung erzeugen. Pro Visualisierung sind dieselben Optionen verfügbar und können beliebig angepasst werden. In der Unlimited Edition sind beliebig viele Visualisierungen erstellbar.

# Individualisierung

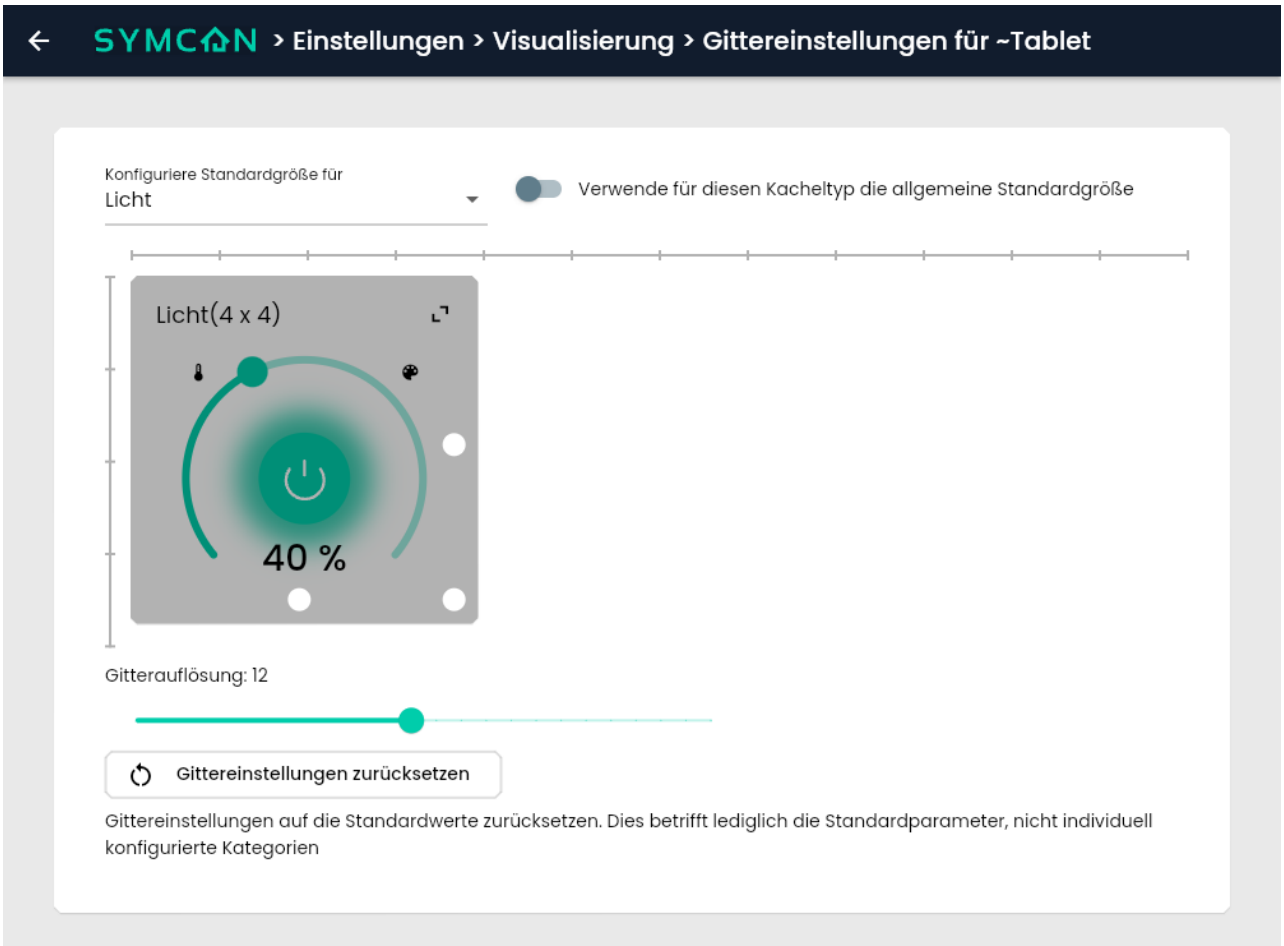
| | |
|---|---|
| Layout bearbeiten | Die Größe und Position von Kacheln kann individuell angepasst werden |
| Kacheln bearbeiten | Die genaue Darstellung einer Kachel kann angepasst werden |
| Designs | Über Designs kann das grundlegende Aussehen der Visualisierung angepasst werden |
| Darstellungsprofile | Über verschiedene Darstellungsprofile kann die gleiche Visualisierung für verschieden große Bildschirme optimiert werden |

# Layout bearbeiten

Benötigt: IP-Symcon >= 7.0

## STANDARDWERTE

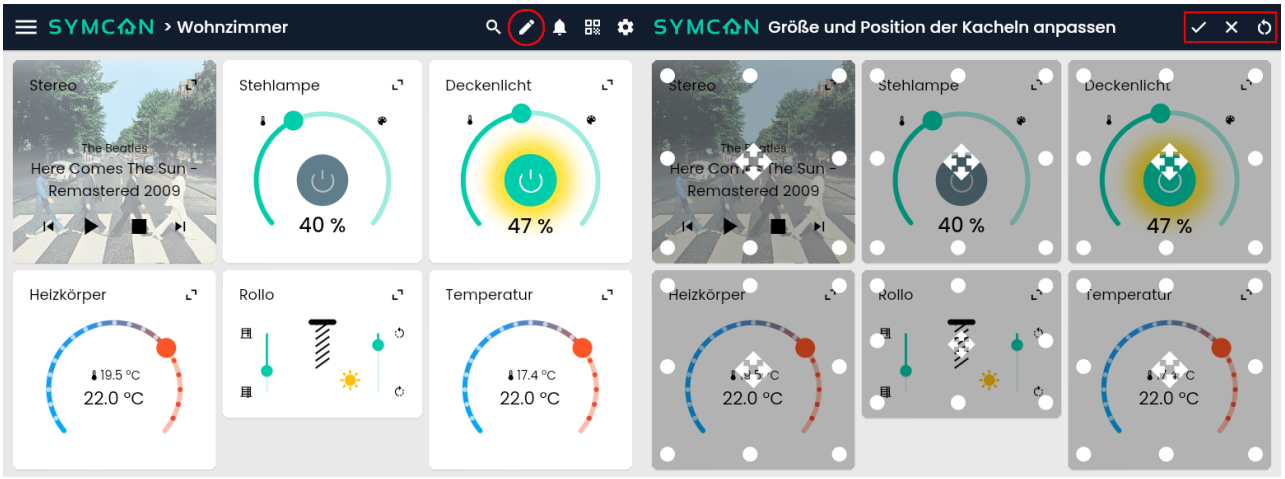
Die Position der Kacheln ergibt sich aus der Position der entsprechenden Objekte im Objektbaum. Die Positionierung verläuft von oben link nach rechts und von oben nach unten.
Alle Kacheln haben eine zugewiesene Standardgröße. Die initialen Standardgrößen ergeben sich aus dem ausgewählten Darstellungsprofil. Dieses wird basierend auf dem Gerätetyp der Bildschirmgröße und -orientierung gewählt, um die best möglichen Ausgangswerte für die Kacheln zu wählen. Diese Standardgrößen können für jede Darstellungsart einer Kachel in den Einstellungen unter Visualisierung>Gittereinstellungen angepasst werden. Hier kann auch die Auflösung des Gitters, auf dem die Kacheln angeordnet sind, bestimmt werden.
Die Breite des Gitters ist dabei fest. Da vertikal gescrollt werden kann ist das Gitter in der Höhe nicht begrenzt.




## BEARBEITUNGSMODUS

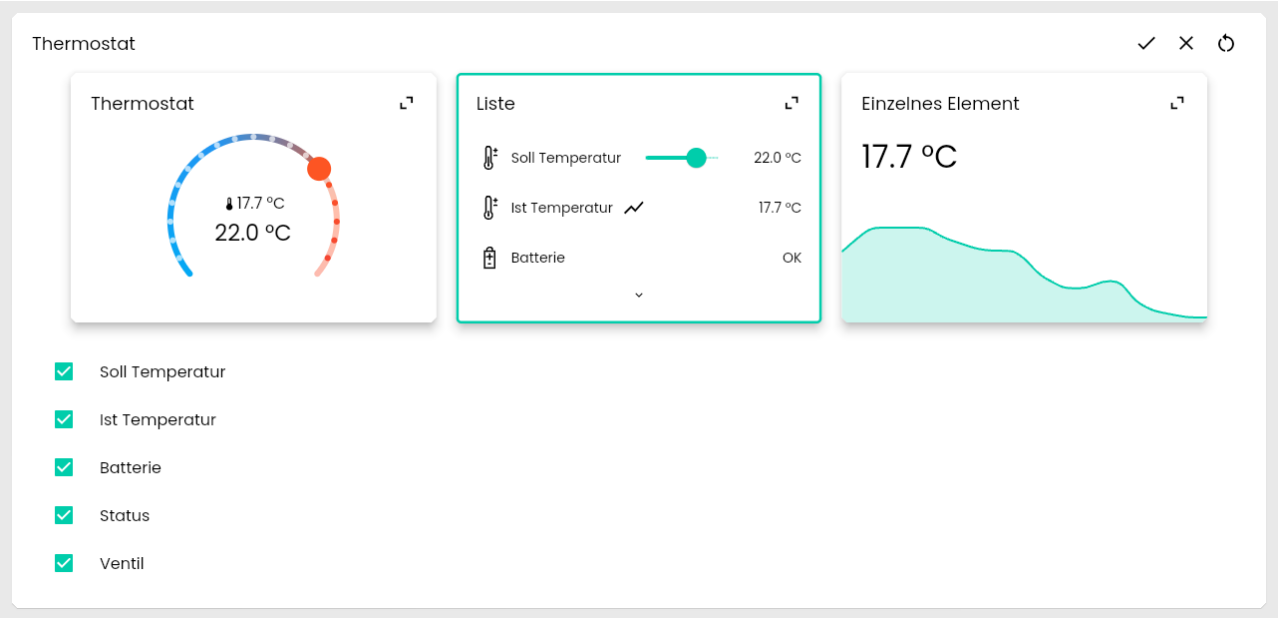
Auf jeder Seite der Visualisierung kann über das Bleistift-Symbol in der Appbar der Bearbeitungsmodus aktiviert werden. Die Position der Kacheln kann mittels Darg & Drop verändert werden. Die Größe kann durch ziehen der Punkte an den Rändern verändert werden. Wenn der Bearbeitungsmodus aktiviert ist, werden in der Appbar drei neue

Optionen angeboten. Die Aktuellen Änderungen können gespeichert oder verworfen werden. Außerdem kann das Layout auf die Standardgrößen zurückgesetzt werden. Das Zurücksetzen bezieht sich hierbei nur auf die aktuell bearbeitete Kategorie. Sobald ein Layout gespeichert wurde, kann es sich nicht mehr durch Positionsänderungen in der Konsole verändern. Ein Hinweis darauf erscheint beim erstmaligen anpassen des Layouts. Wenn zwischen den Kachel leerer Raum entsteht, kann dieser über den Papierkorb entfernt werden.



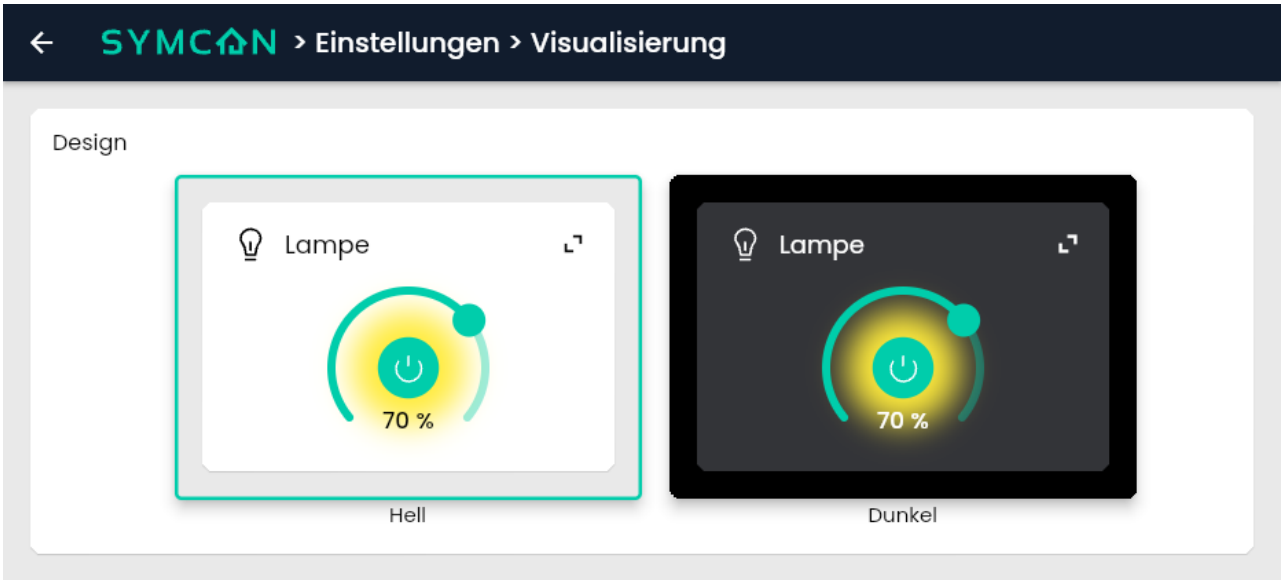
# Kacheln bearbeiten

Benötigt: IP-Symcon >= 7.0
Die Darstellung der einzelner Objekte ergibt sich grundsätzlich aus dem Objekttyp und bei Variablen etwa noch durch das Variablenprofil. Näheres hierzu ist im Abschnitt Objekt-Darstellung beschrieben. Wie Instanzen Dargestellt werden entscheidet sich durch die untergeordneten Variablen. IP-Symcon wählt automatisch das passende Element aus. Um die Darstellung einer Kachel genauer zu konfigurieren muss diese geöffnet und über den Bleistift in den Bearbeitungsmodus gewechselt werden.



# Designs

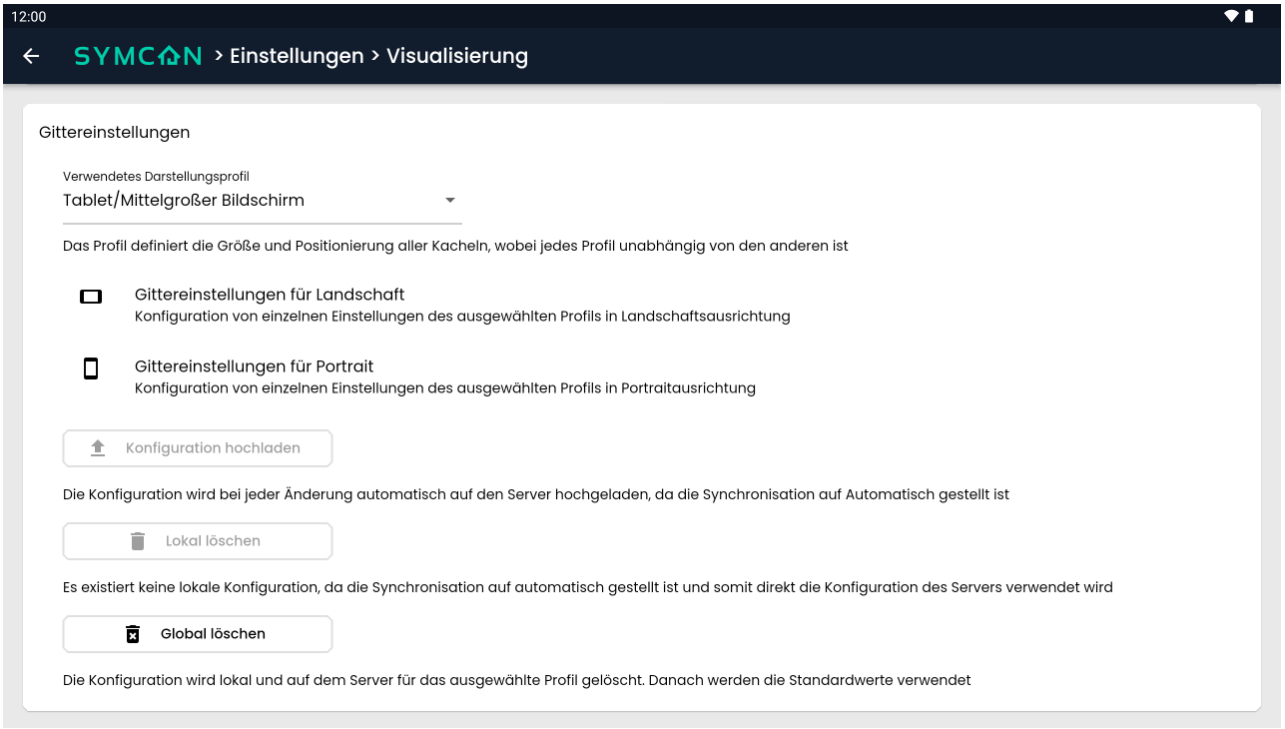
Benötigt: IP-Symcon >= 7.0
Über das Zahnrad rechts in der Appbar kann in den Einstellungen unter Visualisierung das Design verändert werden. Das Design wird pro Gerät gespeichert.




Hintergrund und Textfarbe sowie Schriftart der HTMLBox werden durch das Design beeinflusst.

# Darstellungsprofile

Benötigt: IP-Symcon >= 7.0
Darstellungsprofile ermöglichen verschiedene Darstellungskonfigurationen, basierend auf dem Endgerät. Ein Darstellungsprofil speichert Position und Größe der Kacheln sowie deren Konfiguration. Dies beinhaltet die Kachelauflösung und die Standardgrößen.




Es gibt drei Profile, deren Standardgrößen jeweils auf einen Gerätetyp optimiert sind: Rechner, Tablet und Mobiltelefon. Das passende Profil wird basierend auf dem Betriebssystem und der Bildschirmgröße gewählt. Bei Bedarf kann das Profil manuell ausgewählt werden.

## ORIENTIERUNG

Alle Darstellungsprofile speichern die Einstellungen in Kombination mit der Orientierung. Somit können für jedes Profil Konfigurationen für die Portrait und Landschaft Ausrichtung vorgenommen werden. In der App wird das passende Profil für die aktuelle Orientierung ausgewählt. Im Browser ist nur die Orientierung Landschaft verfügbar.

## GITTEREINSTELLUNGEN

In den Gittereinstellungen können die Standardeinstellungen für das aktuell gewählte Profil angepasst werden. In der App stehen sowohl die Einstellungen für Portrait als auch für Landschaft zur Verfügung, im Browser ist nur die Orientierung Landschaft anpassbar.




Im Dropdown oben kann ausgewählt werden, wofür die Standardgröße konfiguriert werden soll. Hier stehen alle Objekt-Darstellungen zur Auswahl sowie "Alle Kacheln". Bei der Auswahl von "Alle Kacheln" kann durch Anpassung der Größe der dargestellten Kachel, analog zu Layout bearbeiten, die allgemeine Standardgröße für alle Darstellungen konfiguriert werden. Wird eine Objekt-Darstellung gewählt, kann für diese eine individuelle Standardgröße konfiguriert werden. Der Schalter "Verwende für diesen Kacheltyp die allgemeine Standardgröße" deaktiviert die individuelle Standardgröße und verwendet für diese Objekt-Darstellung somit die allgemeine Standardgröße.
Somit ergibt sich für jede Kachel in der Visualisierung folgende Größe:

- Wurde via Layout bearbeiten eine individuelle Größe für die Kachel definiert, so wird diese verwendet
- Ist "Verwende für diesen Kacheltyp die allgemeine Standardgröße" deaktiviert, so wird die individuelle Standardgröße für diese Objekt-Darstellung verwendet
- Ist "Verwende für diesen Kacheltyp die allgemeine Standardgröße" aktiviert, so wird die allgemeine Standardgröße für diese Objekt-Darstellung verwendet, welche per "Alle Kacheln" konfiguriert wurde.

Mit dem Regler "Gitterauflösung" kann die Auflösung des Gitters definiert werden, auf welchem die Kacheln platziert werden. Eine höhere Auflösung ermöglicht somit eine detailliertere Anpassung von Breite und Höhe.
Durch Betätigen des Buttons "Gittereinstellungen zurücksetzen" werden die Standardgrößen und die Gitterauflösung wieder auf die Standardwerte passend zum aktuellen Profil und Orientierung zurückgesetzt.

## SYNCHRONISATION

Für die Darstellung eines Profils gibt es eine Konfiguration auf dem Server, welche für alle Geräte identisch ist, sowie eine lokale Konfiguration, die für jedes Gerät individuell ist. Für die einzelnen Parameter, wird primär der Wert aus der lokalen Konfiguration verwendet. Wurde hier nichts definiert, wird stattdessen der Wert der Serverkonfiguration verwendet. Sollte auch hier kein Wert gesetzt sein, wird der Standardwert passend zum aktuellen Profil und Ausrichtung verwendet.
Je nach Einstellung innerhalb der Visualisierungsinstanz wird die lokale Konfiguration automatisch oder manuell auf den Server hochgeladen oder ist das Hochladen komplett gesperrt, siehe auch Instanzkonfiguration.

## BUTTONS ZUR SYNCHRONISATION

| BUTTON | BESCHREIBUNG |
|---|---|
| Konfiguration hochladen | Lädt die lokale Konfiguration auf den Server hoch |
| Lokal löschen | Löscht die lokale Konfiguration |
| Global löschen | Löscht die lokale Konfiguration und die Serverkonfiguration |

# Instanzkonfiguration

Benötigt: IP-Symcon >= 7.0

## STARTKATEGORIE

Die Ausgewählte Kategorie legt fest von wo aus die Visualisierung aus aufgebaut wird.

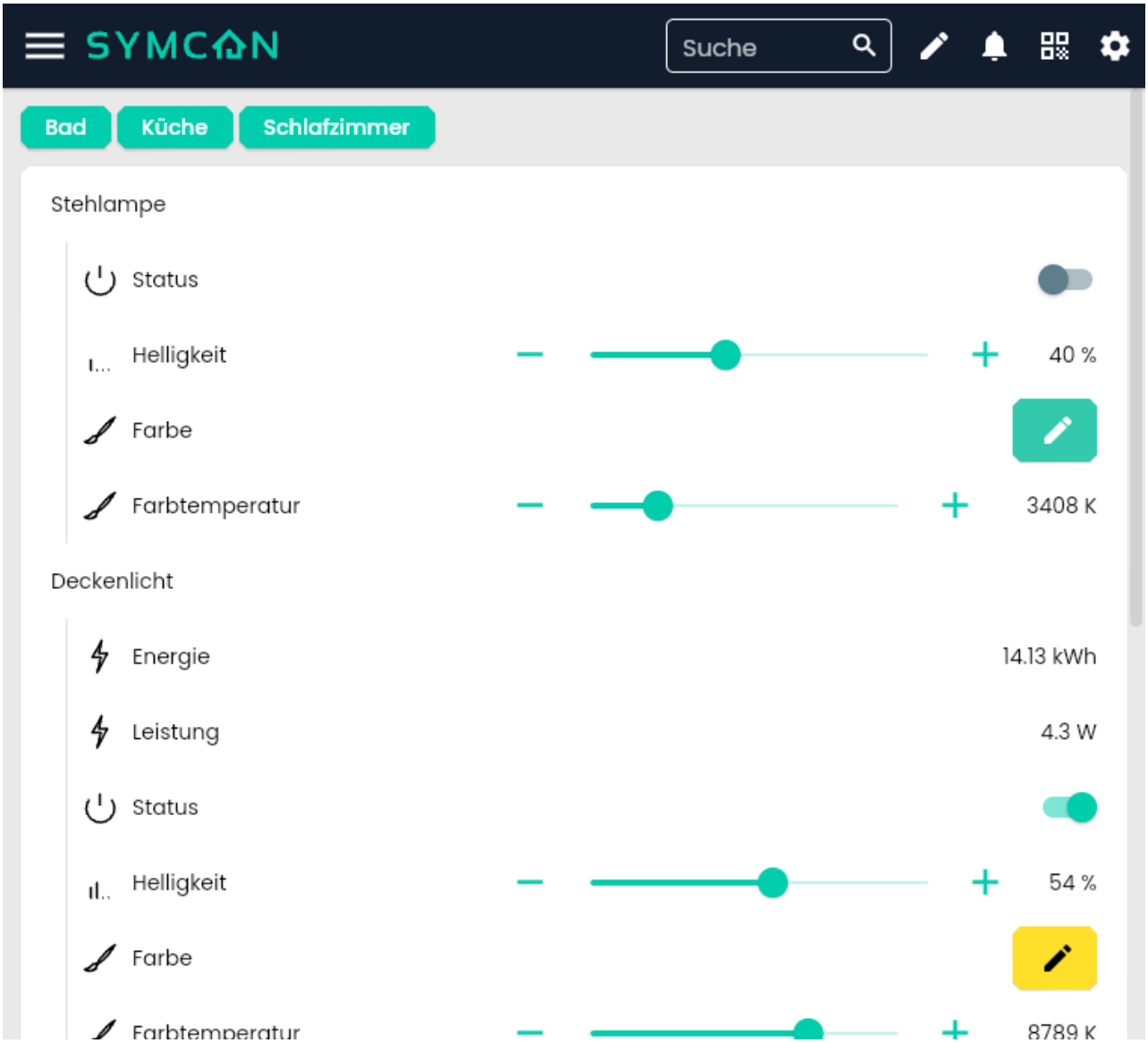
## BEGRÜßUNG ANZEIGEN

Die Begrüßung wird im Startbereich der Visualisierung angezeigt und kann über verschiedene Parameter konfiguriert werden.

| BEGRÜßUNG ANZEIGEN | BESCHREIBUNG |
|---|---|
| Automatisch | Der Text wird basierend auf der Tageszeit angepasst und kann optional mit einem Namen erweitert werden. |
| Keine | Die Begrüßung wird ausgeblendet. Wenn keine Favoriten oder Automationen vorhanden sind, wird der gesamte Startbereich ausgeblendet. |
| Variable | Der Begrüßungstext wird auf den Wert einer Variable gesetzt. Der Typ ist hierbei nicht relevant. |

## LEGACY ANZEIGEMODUS

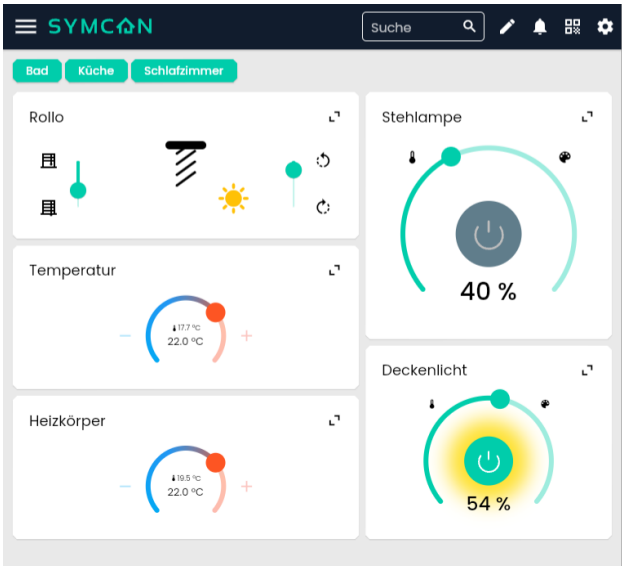
Ist der Legacy Anzeigemodus aktiviert werden alle Objekte in einer Liste dargestellt.

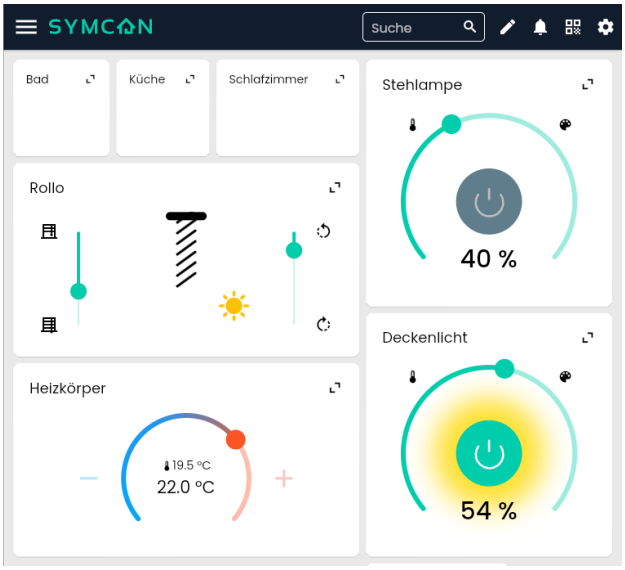



## KATEGORIE DARSTELLUNG

Diese Einstellung dient als Grundlage und kann für jede Kategorie individuell überschrieben werden.

| DARSTELLUNG | BESCHREIBUNG |
|---|---|
| Kategorieleiste (Einzeilig) | Alle Unterkategorien der aktuellen Kategorie werden am oberen Bildschirmrand in einer Zeile angezeigt. Geht die Anzahl der Kategorien über die breite des Bildschirms hinaus, kann diese horizontal gescrollt werden. |
| Kategorieleiste (Mehrzeilig) | Alle Unterkategorien der aktuellen Kategorie werden am oberen Bildschirmrand in einer Zeile angezeigt. Geht die Anzahl der Kategorien über die breite des Bildschirms hinaus, werden die Kategorien in mehreren Zeilen dargestellt. |
| Kacheln | Kategorien werden wie alle anderen Objekte als Kacheln angezeigt. Somit können Größe und Position frei verändert werden |
| Nur in Seitenleiste | Die Kategorien werden nur in der Seitenleiste angezeigt |



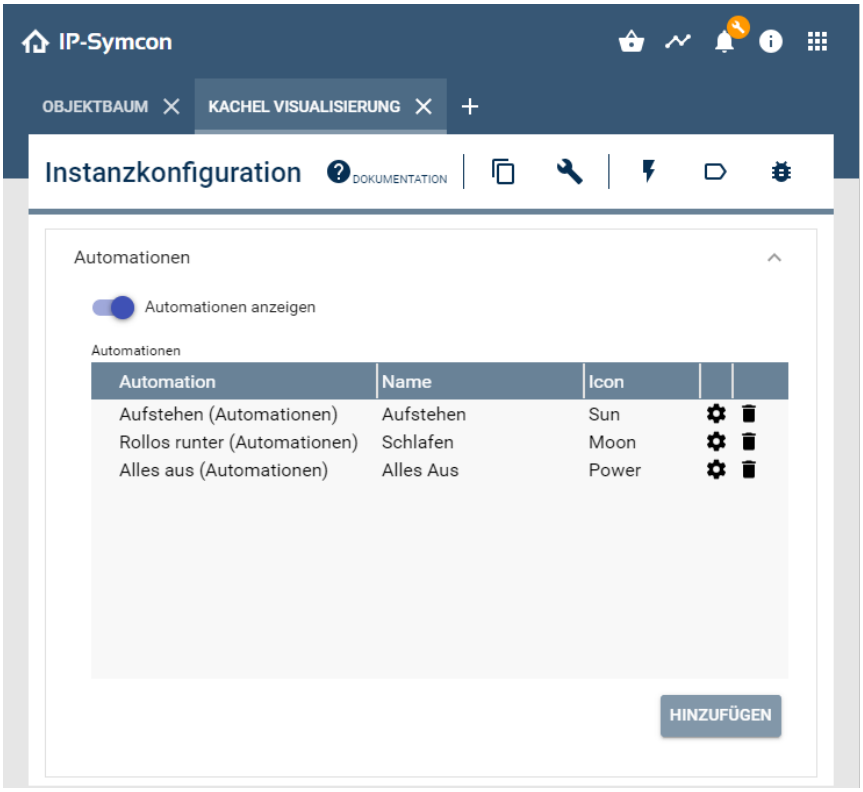

## KONFIGURATION SYNCHRONISIERUNG

Die Funktionsweise der Synchronisierung wird im Bereich Darstellungsprofile erläutert.

| MODUS | BESCHREIBUNG |
|---|---|
| Automatisch | Jede Änderung am Layout wird sofort mit dem Server und dadurch mit allen Geräten synchronisiert |
| Manuell | Lokale Änderungen müssen in den Einstellungen hochgeladen werden, damit sie auch mit anderen Geräten synchronisiert werden können |
| Nur-Lesen | Die Konfiguration kann an Endgeräten nur lokal verändert und nicht auf den Server hochgeladen werden |

# AUTOMATIONEN

Hier kann eine Liste an Automationen zusammengestellt werde, welche dann in der Startleiste zur Verfügung stehen.
Jede Automation kann mit einem eigenen Namen und Icon versehen werden.

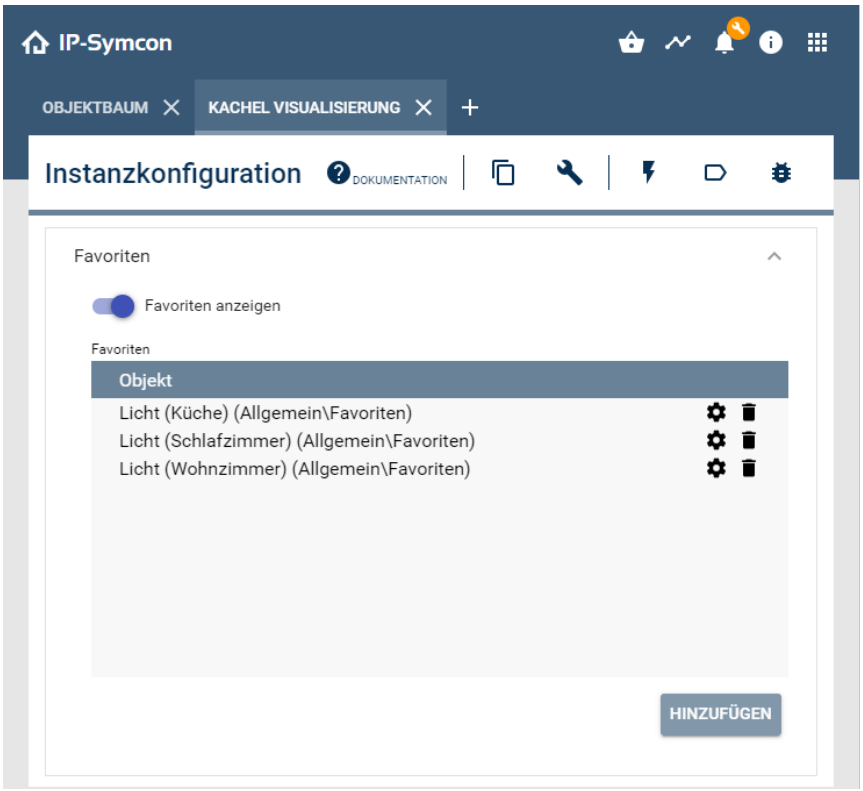


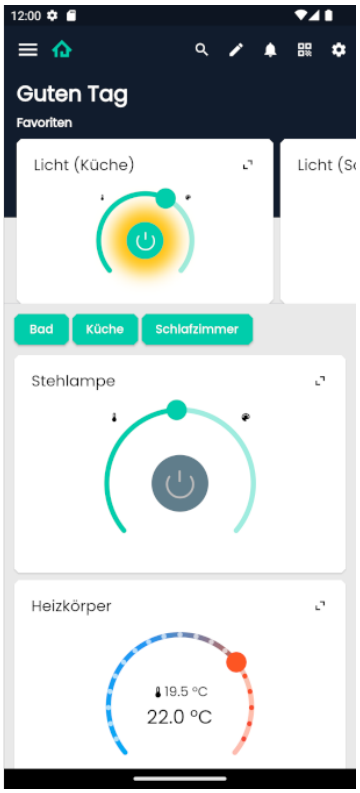




# FAVORITEN

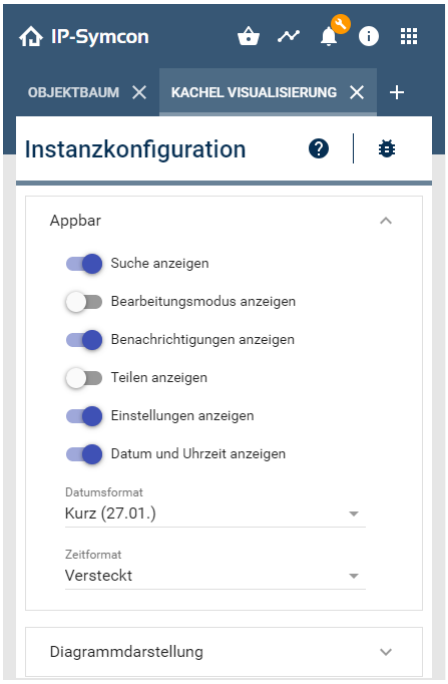
Als Favoriten können jegliche Objekte hinzugefügt werden. Diese werden im oberen Bereich des Bildschirms als normale Kacheln angezeigt. In der Variante für große Bildschirme kann auf diese via Herz-Button zugegriffen werden.

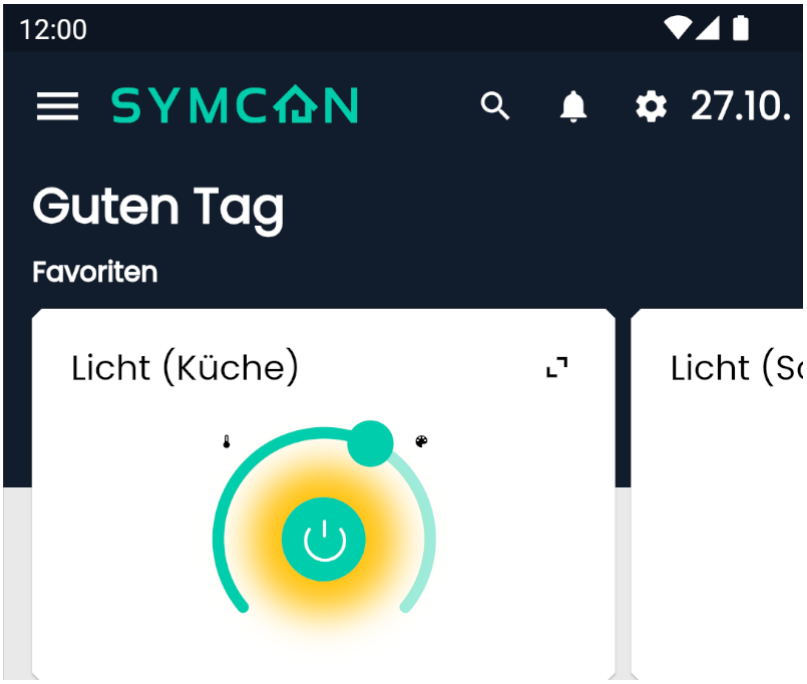


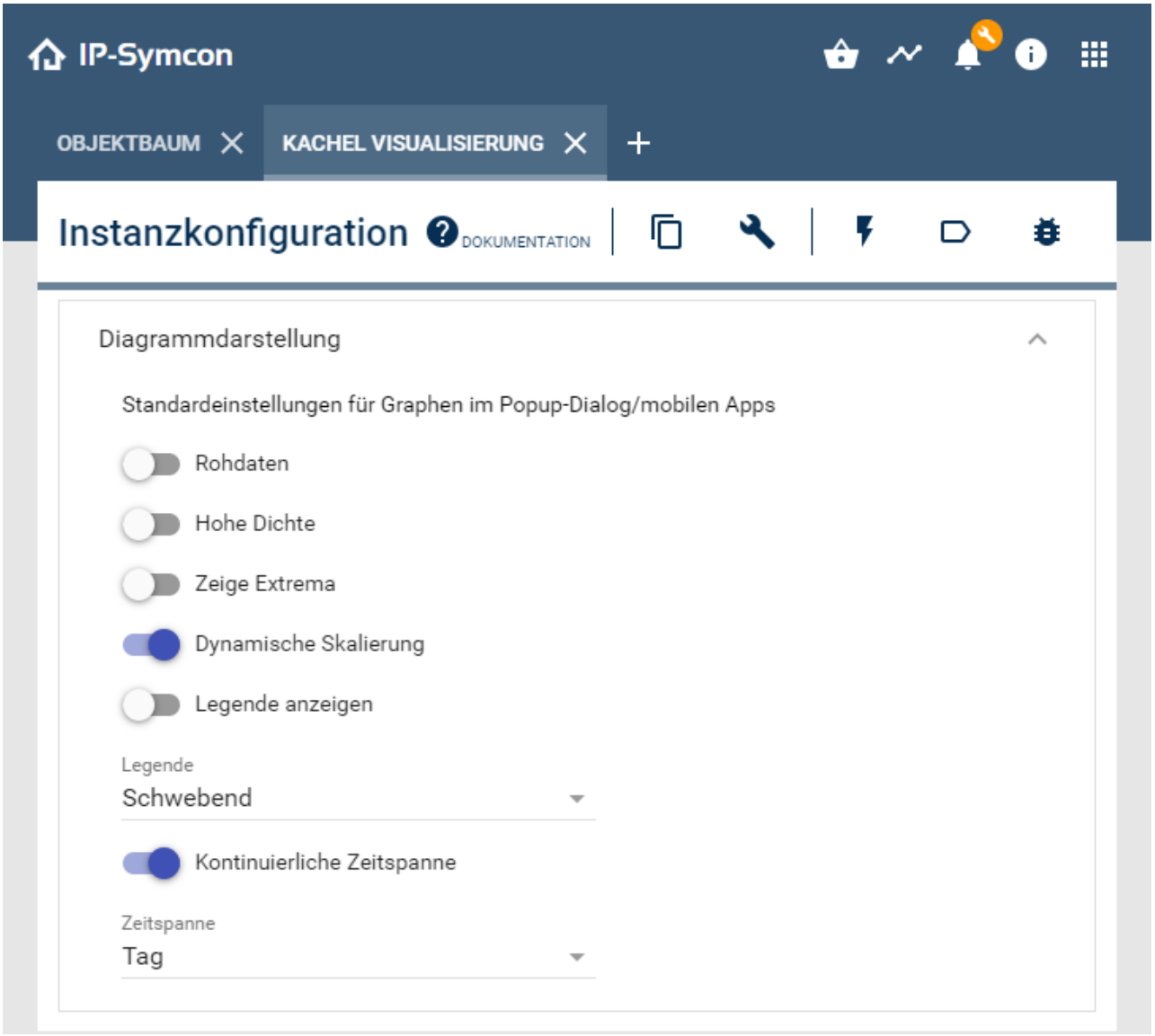

# APPBAR

Falls nicht genug Platz für die konfigurierten Elemente vorhanden ist, werden überzählige Elemente in einem Menü zusammengefasst.




# DIAGRAMMDARSTELLUNG



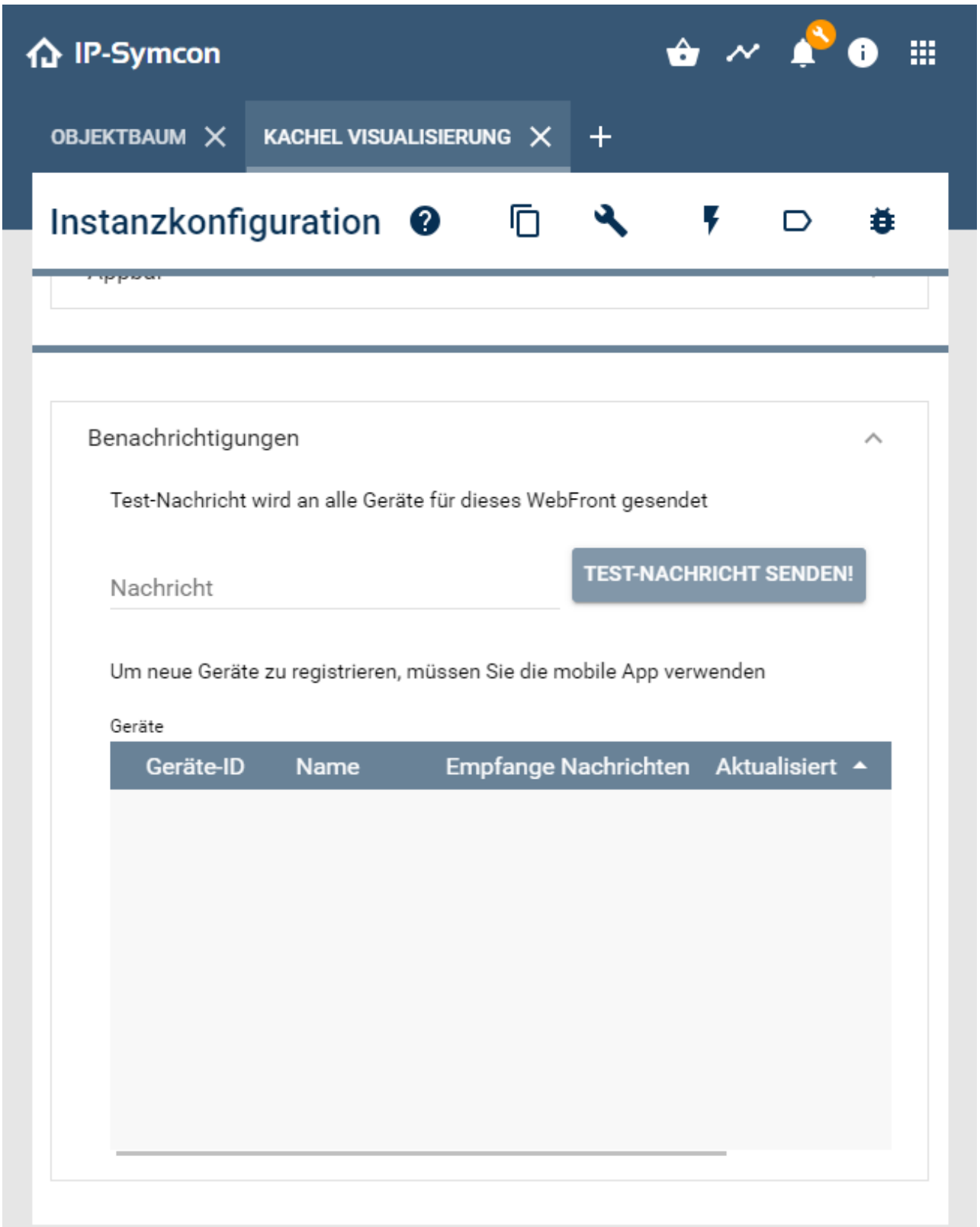

Hier können die Standardeinstellung für Diagramme festgelegt werden.
Was welcher Begriff genau bedeutet kann unter Darstellungsarten von Charts nachgelesen werden.

# WEITERE KONFIGURATION

| SEITE | BESCHREIBUNG |
|---|---|
| Benachrichtigungen | Erlaubt das Versenden von Push-Nachrichten und Verwalten konfigurierter Geräte. |
| Sicherheit | Ermöglicht die Einstellung eines Passwortes zur Zugriffskontrolle und einer speziellen IP-AutoStart Funktion. |
| Virtuelle Hosts | Erlaub das Aufteilen von Visualisierungen auf verschiedene virtuelle Hosts |

# Benachrichtigungen

Der Reiter bietet eine Übersicht über die registrierten Geräte. Es kann ausgewählt werden, ob das jeweilige Gerät Meldungen empfängt.
Auch kann eine Testnachricht verschickt werden.




## REGISTRIERUNG ZUM PUSH-NACHRICHTEN DIENST

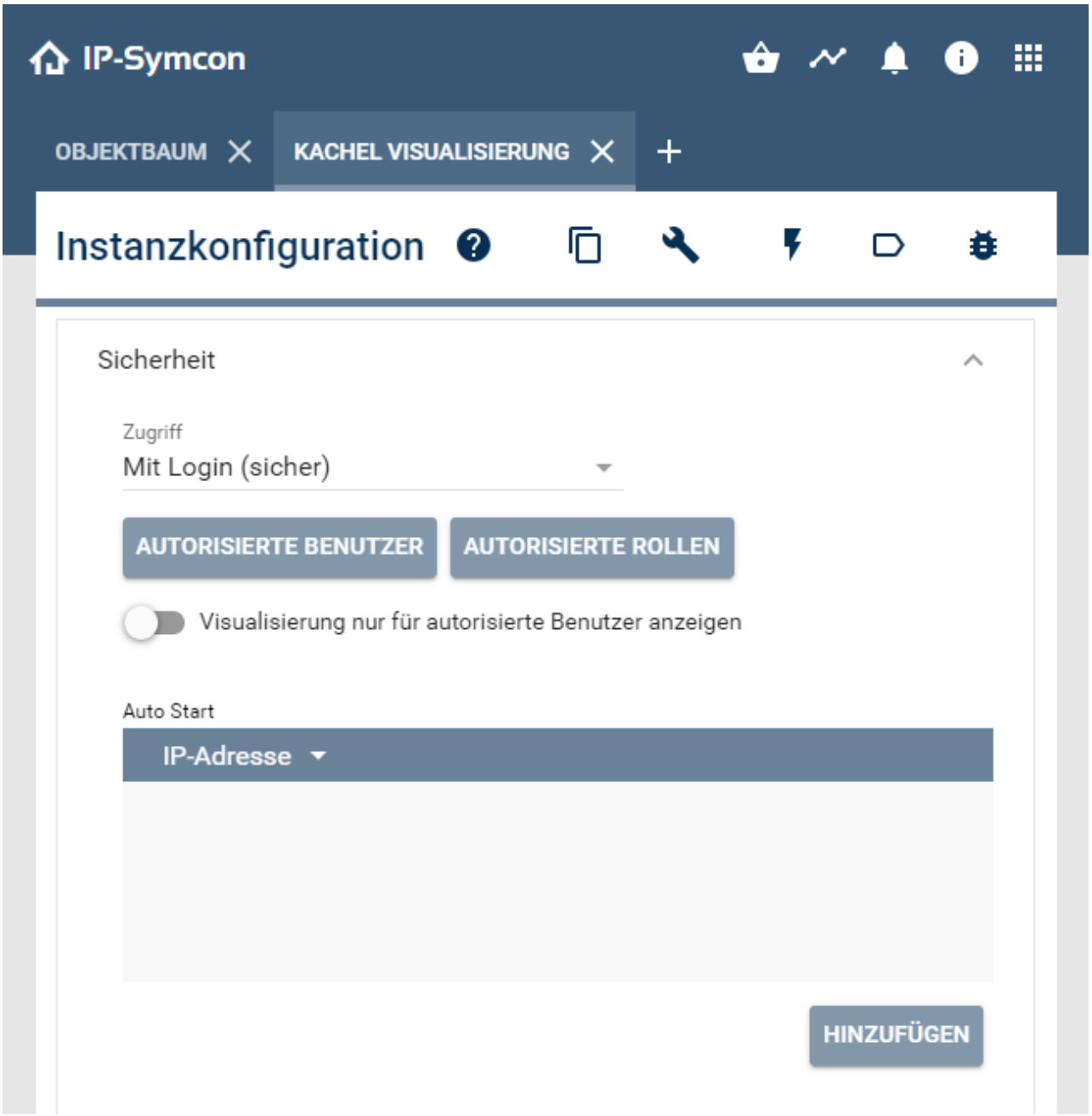
Sobald ein Mobilgerät sich in einer Visualisierung das erste Mal erfolgreich einloggt, wird dieses Gerät automatisch zum Push-Nachrichten Dienst für diese Instanz hinzugefügt. Wenn keine Push-Nachrichten an dieses Gerät versendet werden sollen, so muss dieses Gerät explizit für den Empfang in der Liste deaktiviert werden. Zu beachten ist, dass die Registrierung zum Push-Nachrichten Dienst unberührt bleibt, falls das Kennwort für die Visualisierung geändert wird. Wird ein Gerät komplett aus der Liste entfernt, so wird es beim nächsten Login wie ein unbekanntes Gerät behandelt und erneut zum Push-Nachrichten Dienst hinzugefügt.

## NOTIFICATION CONTROL

Die Nachrichten werden via Notification Control versendet.
Weitere Informationen befinden sich unter Notification Control .

# Sicherheit

Der Zugriff auf eine Visualisierung kann auf verschiedene Arten gesichert werden. Weitere Sicherheitsinformationen können unter Sicherheit eingesehen werden.




## ZUGRIFF

Jede der verfügbaren Zugriffsarten bietet verschiedene Einstellungsmöglichkeiten.

### MIT PASSWORT (SICHER)

Für jede Visualisierungsinstanz kann ein Passwort als Zugriffschutz vergeben werden. Dieses Passwort wird vor dem Zugriff auf die jeweilige Visualisierung über Browser und die mobilen Apps gefordert. Zusätzlich kann die Spezialoption aktiviert werden, dass das Passwort nicht innerhalb des lokalen Netzwerk gefordert wird. Dieses Passwort gilt nur

für den Zugriff auf das System und dessen Komponenten. Die Abfrage der verfügbaren Visualisierungen ist jederzeit ohne Passwort möglich.

### OHNE PASSWORT (UNSICHER)

Ohne Passwort wird der Zugriff von extern standardmäßig blockiert. Soll die Visualisierung allerdings bewusst öffentlich zugänglich sein, kann dies mit der Option "Externen Zugriff erlauben, auch wenn kein Kennwort gesetzt ist" deaktiviert werden.

### MIT LOGIN (SICHER)



Der Zugriff mit Login ist eine kostenpflichtige Erweiterung, welche zu jeder bestehenden IP-Symcon Lizenz erworben werden kann. Für eine passende Demo-Version wenden Sie sich bitte an unseren Support. Die Erweiterung kann direkt im Shop erworben werden.

Der Zugriff mit Login erlaubt das Öffnen der Visualisierung nur autorisierten Benutzern oder Benutzern mit einer autorisierten Rolle. Die benötigten Berechtigung für Benutzer und Rollen können über die jeweiligen Schaltflächenvergeben werden. Um eine Visualisierung nur für autorisierten Benutzern anzuzeigen, kann die entsprechende Option aktiviert werden. Nutzer und Rollen können über die Berechtigungsteuerung verwaltet werden.

## AUTOSTART FÜR IP-ADRESSE

Eine besondere Funktion ist der Auto-Start, welcher einen Visualisierung direkt ohne Passwortabfrage öffnet, wenn die IP-Adresse mit einer der eingetragenen IP-Adressen in dieser Liste übereinstimmt. Über diese Funktion kann z.B. auf einem Touch-Panel eine Visualisierung direkt geöffnet werden, ohne den normalen Auswahldialog angezeigt zu bekommen und ohne die Passwortabfrage beantworten zu müssen. Es ist zu beachten diese Funktion nur innerhalb eines sicheren Netzwerks zu benutzen.



Es wird auch die CIDR Notation unterstützt, sodass z.B. 192.168.1.0/24 alle Adresse von 192.168.1.1 bis 192.168.1.255 enthält.



Diese Funktion steht nicht zur Verfügung, sofern Anfragen an IP-Symcon über einen Reverse-Proxy oder über den Connect-Dienst gehen, da der Ursprung dann nicht sicher und zuverlässig verifiziert werden kann. Eine Ausnahme ist der CIDR Wert 0.0.0.0/0, welcher alle Anfragen unabhängig der Herkunft umleitet und somit grundsätzlich unsicher ist. Dieser wird auch für die o.g. Ausnahmen ausgewertet und leitet somit alle Anfragen unabhängig des Ursprungs weiter.

## AUFRUF ÜBER URL

Es gibt Situationen (z.B. in Verbindung mit NAT) bei denen die Autostart via IP nicht funktioniert. In diesen Fällen kann eine URL direkt aufgerufen werden.

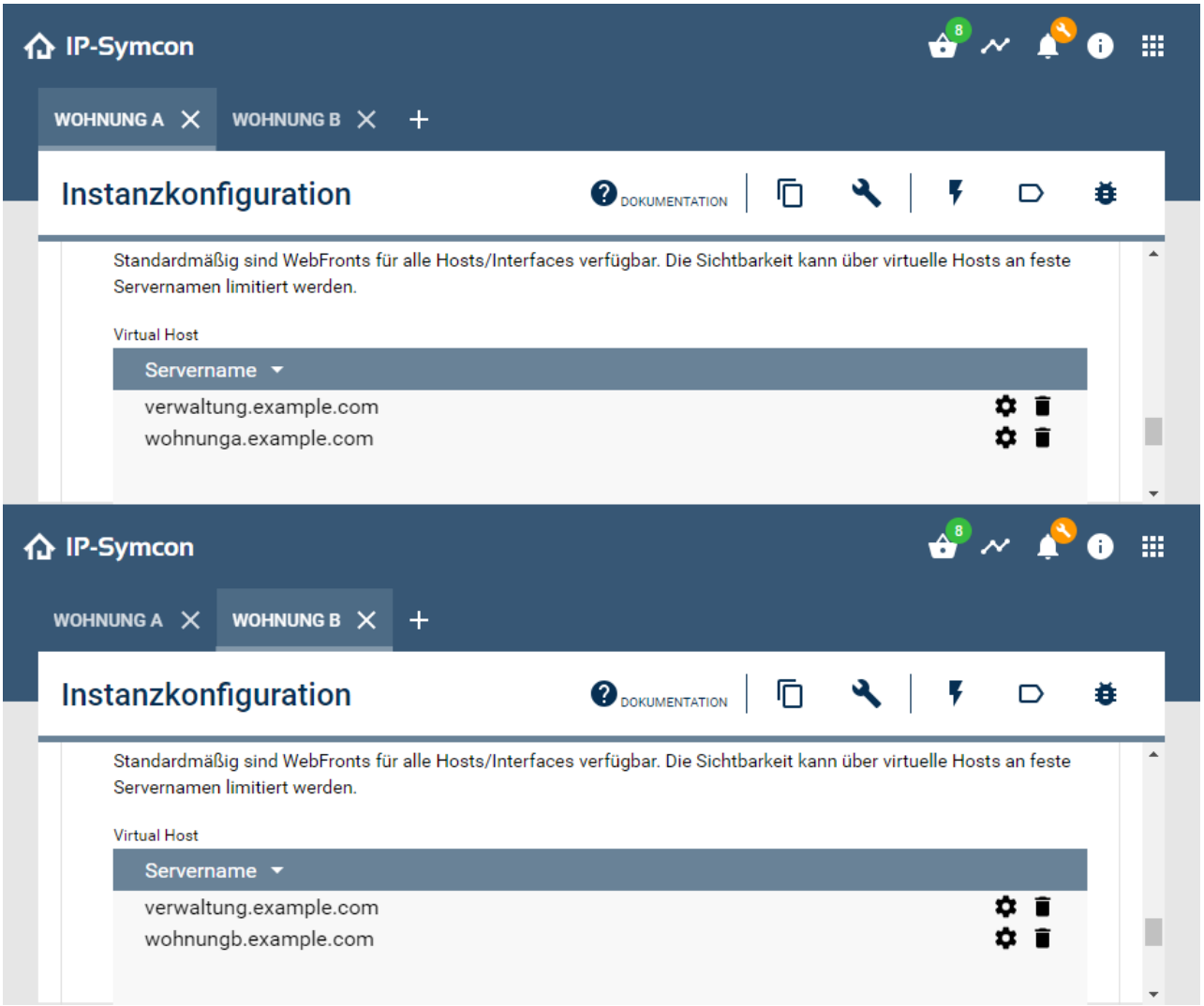
```
// Grundgerüst für den URL Aufruf
http://<SymconIP>:3777/#<VisualisierungID>

// Beispiel als Start mit dem Chromium Browser
// VisualisierungInstanzID: = 12345
chromium-browser --start-fullscreen 192.168.1.123:3777/#12345
```

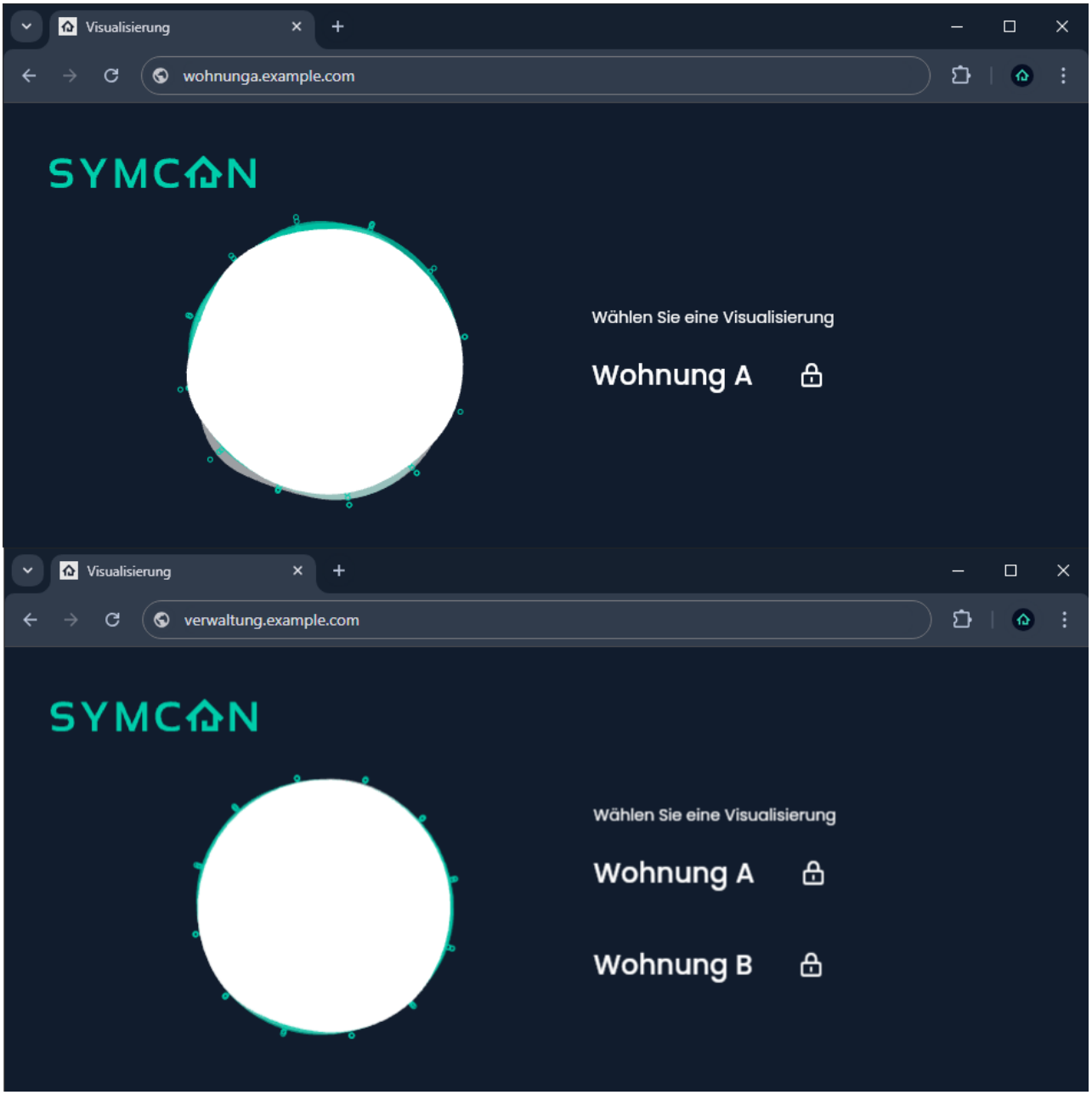
# Virtuelle Hosts

Virtuelle Hosts ist eine kostenpflichtige Erweiterung, welche zu jeder bestehenden Symcon Lizenz erworben werden kann. Für eine passende Demo-Version wenden Sie sich bitte an unseren Support. Die Erweiterung kann direkt im Shop erworben werden.

Über Virtuelle Hosts kann eine Liste von Hosts definiert werden, welche auf die Visualisierung zugreifen können. Wird die Visualisierung von einem Host aufgerufen, der nicht in der Liste vertreten ist wird die Visualisierung nicht angezeigt. Ist die Liste leer ist die Visualisierung für alle Hosts sichtbar.




Mit der obigen Konfiguration von zwei Visualisierungen werden die Startseiten wie folgt angezeigt.


Visualisierung
wohnunga.example.com
SYMCON
Wählen Sie eine Visualisierung
Wohnung A
Visualisierung
verwaltung.example.com
SYMCON
Wählen Sie eine Visualisierung
Wohnung A
Wohnung B

# Apps

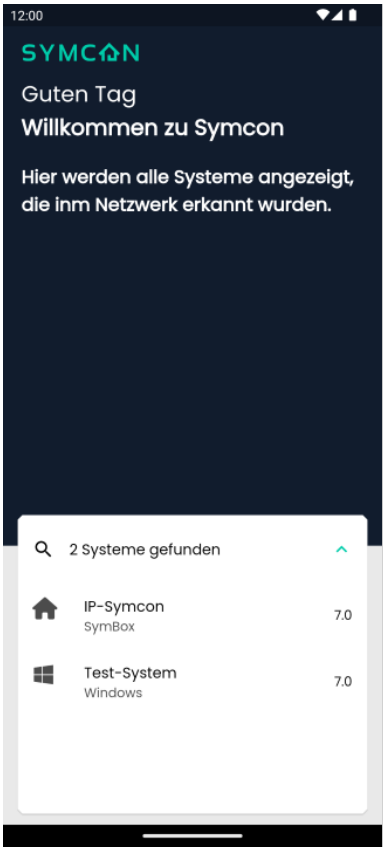
Benötigt: IP-Symcon >= 7.0
Die Apps können aus dem jeweiligen Appstore heruntergeladen werden (iOS, Android).
Die Mobilen Apps unterscheiden sich im Funktionsumfang nicht von der Visualisierung im Browser. Lediglich der Einrichtungsprozess ist ein anderer.

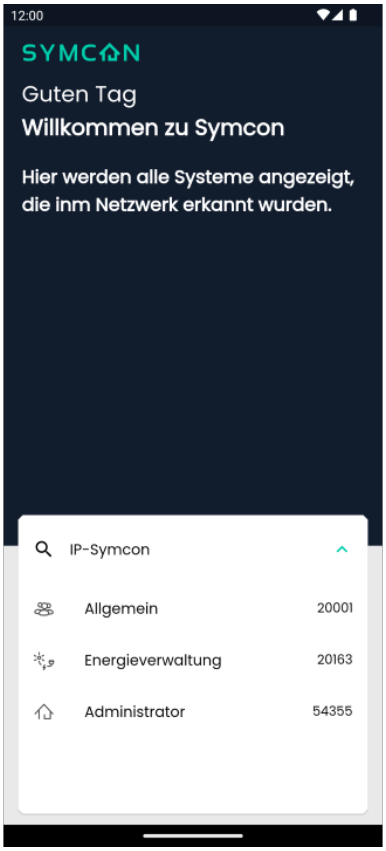
## VERBINDEN MIT DER KACHEL-VISUALISIERUNG UBER DIE APPS

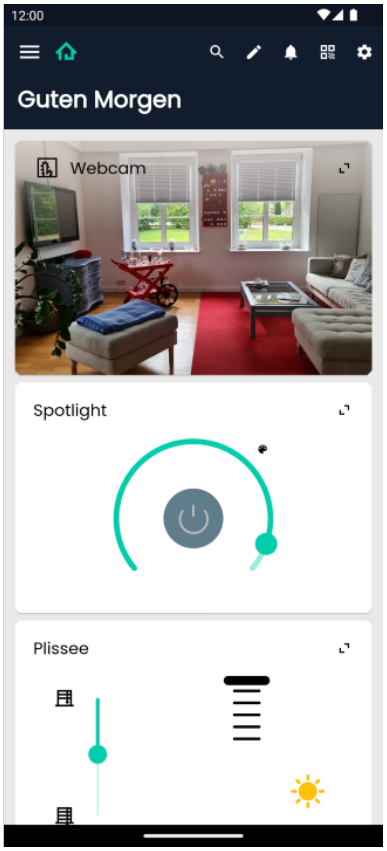
Um sich mit einer Visualisierung zu verbinden gibt es drei Möglichkeiten. Diese werden im Folgenden erläutert. Außerdem kann die App direkt in einem Demo-Visualisierung geöffnet werden.

### AUTOMATISCHE SUCHE

Wenn das Mobilgerät mit dem selben Netzwerk wie der IP-Symcon Server verbunden ist, kann dieser mit der automatischen Suche gefunden werden. Die erste Kachel zeigt die gefundenen IP-Symcon Server an. Diese können angetippt und anschließend die dazugehörige Visualisierung ausgewählt werden. Wenn für die Visualisierung ein Password gesetzt ist, muss dieses nun eingegeben werden. Sollte es im Netzwerk nur einen IP-Symcon Server mit einer Visualisierung geben wird diese direkt geöffnet.



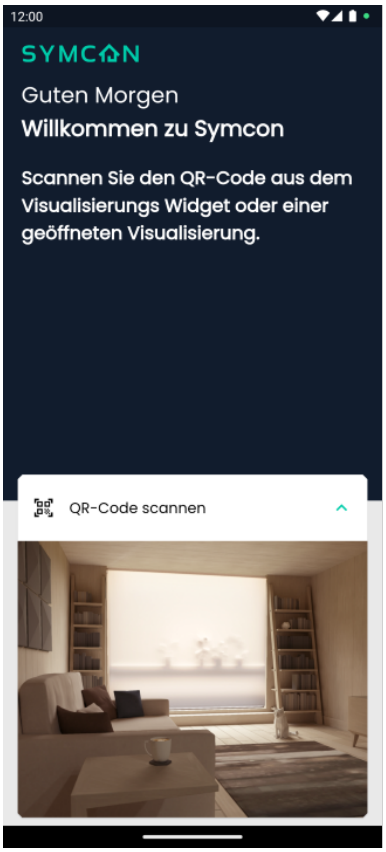



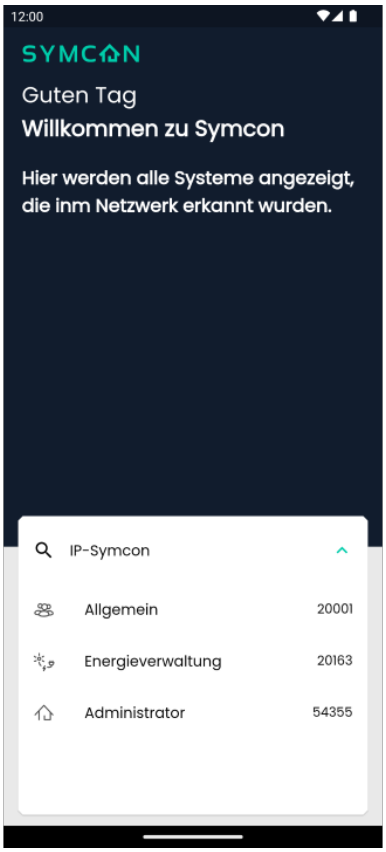



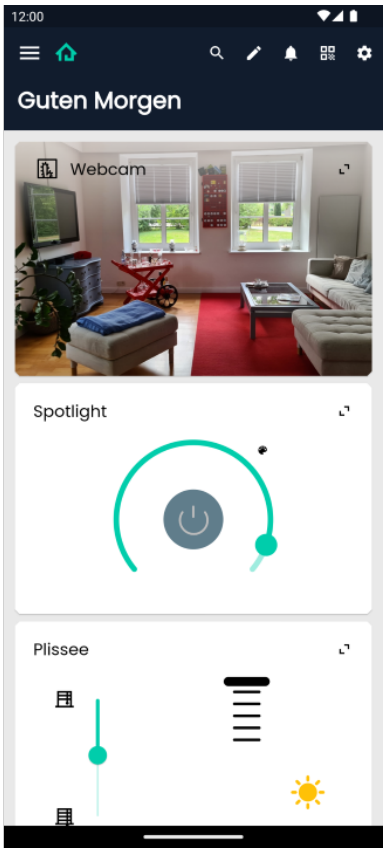

### QR-CODE

Der zum Scannen benötigte QR-Code kann aus dem Visualisierungs-Widget in der

Verwaltungskonsole oder aus einer bereits geöffneten Visualisierung erhalten werden. Um den Code zu scannen müssen die erforderlichen Berechtigungen in der App gewährt werden. Nachdem der Code gescannt wurde, kann die gewünschte Visualisierung ausgewählt und das Password eingegeben werden.



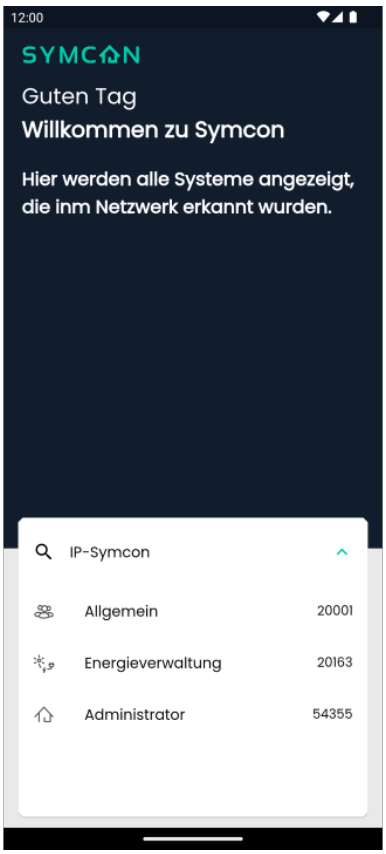



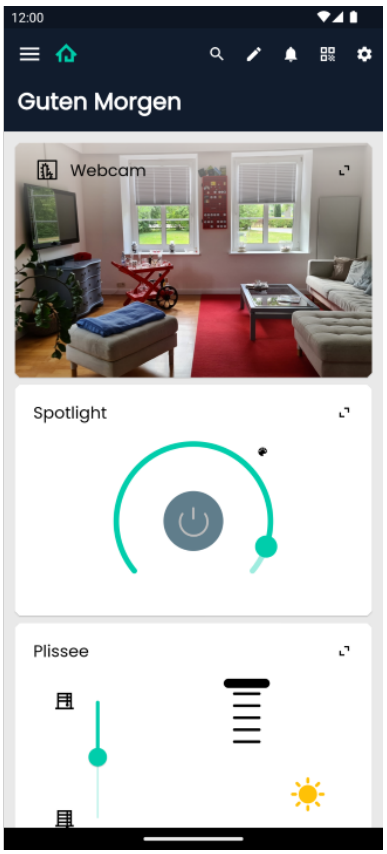




## MANUELL HINZUFÜGEN

Um einen IP-Symcon Server manuell hinzuzufügen muss lediglich die IP-Adresse und Port des Servers in das Eingabefeld eingetragen werden. Alternativ kann auch die Connect- oder eine eigen DynDNS-Adresse verwendet werden.

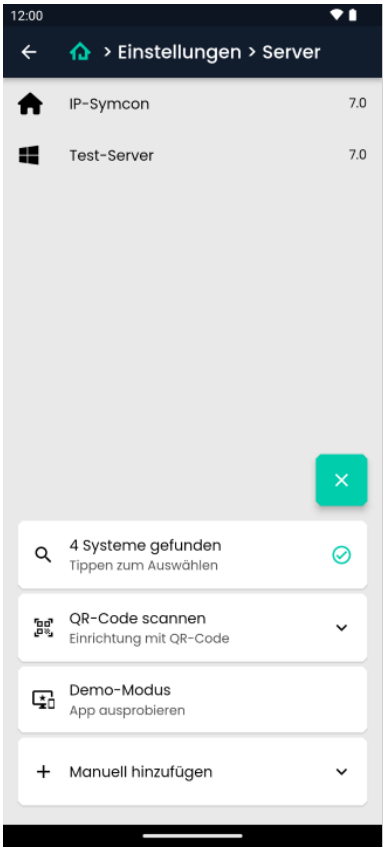


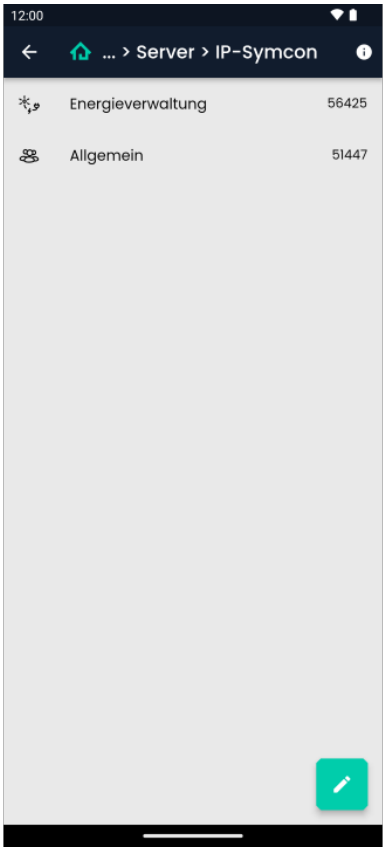



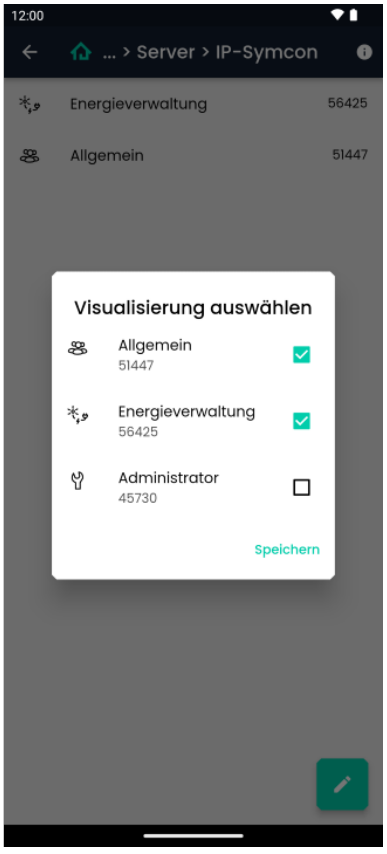




# EINSTELLUNGEN

Alle eingerichteten Server und Visualisierungen werden auf dem Gerät gespeichert. Über das Zahnrad in der Appbar können unter dem Punkt 'Server' neue Server eingerichtet werden. Wird ein Server ausgewählt, können über den Stift in der unteren rechten Ecke die gespeicherten Visualisierungen bearbeitet werden.



# Browser

Benötigt: IP-Symcon >= 7.0

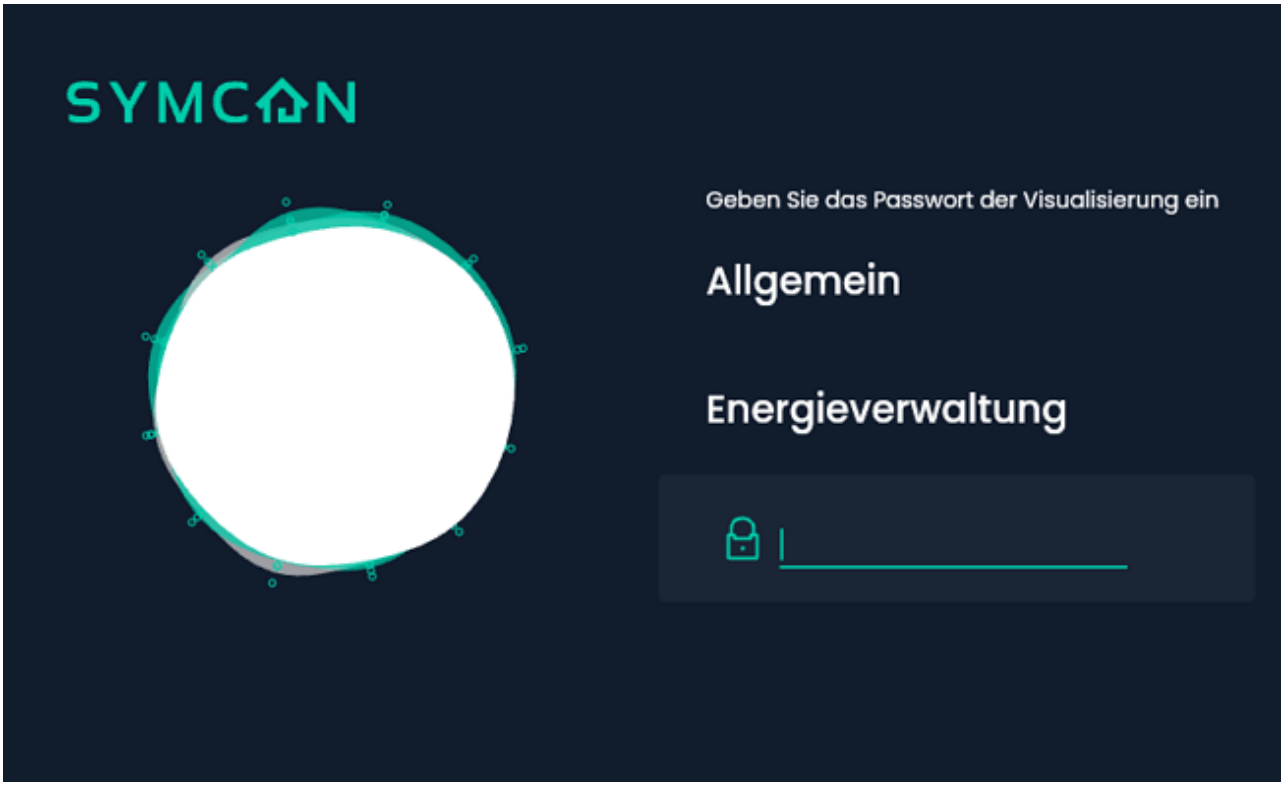
## VERBINDEN MIT DER KACHEL-VISUALISIERUNG IM BROWSER

Die Kachel-Visualisierung kann via "[ServerIP]:3777/" in einem Browser aufgerufen werden. Sollte noch die WebFront-Visualisierung als Standard eingestellt sein, so kann die Kachel-Visualisierung über "[ServerIP]:3777/tile/" erreicht werden.

## VISUALISIERUNG AUSWÄHLEN




Im Browser erscheint dann eine Liste der auf diesem Server verfügbaren Visualisierungen. Ist eine Visualisierung passwortgeschützt, wird dies mit einem Schloss-Icon symbolisiert. Mit einem Klick kann die zu öffnende Visualisierung ausgewählt werden. Ist die Visualisierung nicht passwortgeschützt, wird sie direkt geöffnet.




Ansonsten erscheint ein Textfeld zur Eingabe des Passwortes. Dort muss das Passwort eingegeben und abschließend mit Enter bestätigt werden. Ist das Passwort korrekt, öffnet sich dann die Visualisierung.

# WebFront Visualisierung

Das WebFront wird weiterhin unterstützt. Allerdings gibt es seit der 7.0 die Kachel Visualisierung als modernere Visualisierung. Auch wenn das WebFront immer noch nutzbar ist und bezüglich Fehlern und Kompatibilität weiter gepflegt wird, erhält es keine neuen Features mehr. Es wird empfohlen, die neue Kachel Visualisierung zu verwenden.

Das WebFront muss nicht konfiguriert werden und ist ohne jegliche erweiterte Einstellungen vollständig benutzbar. Alle hier vorgestellten Möglichkeiten sind optional und können von erfahrenen Benutzern verwendet werden, um die Visualisierung noch besser an Ihre Bedürfnisse anzupassen.

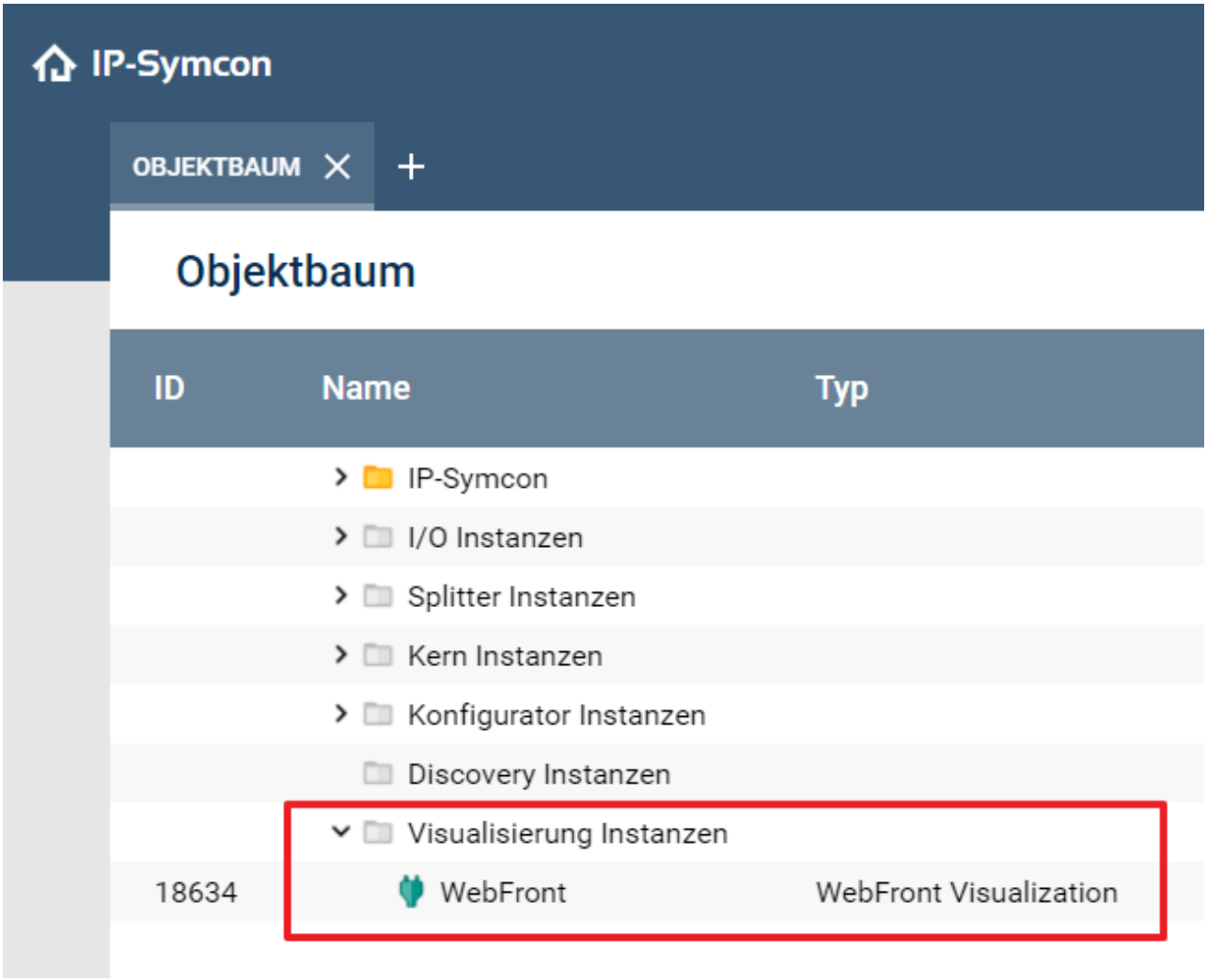
Das WebFront bietet eine einfache und ansprechende Möglichkeit, Geräte anzuzeigen und zu schalten, Bilder anzuzeigen und Skripte auszuführen.

## WEBFRONT VERSIONEN

| VERSION | BESCHREIBUNG |
|---|---|
| WebFront | Die HTML5 Version vom WebFront lässt sich mit den aktuellen Versionen der gängigsten Webbrowser aufrufen und bedienen. Dazu gehören:<br>*Google Chrome (64+) [offiziell empfohlen]*<br>Mozilla Firefox (58+)<br>*Apple Safari (OS X 11.1+)*<br>Opera (50+)<br>*Microsoft Edge (42+)*<br>Microsoft Internet Explorer (11+)<br>*Apple iOS (10.0+)*<br>Google Android (6.0+) |
| Mobile (iOS) | IP-Symcon Mobile ist eine native App, die direkt über den AppStore heruntergeladen werden kann. Es werden alle iOS-Geräte unterstützt, die iOS 10.x oder neuer installiert haben.<br>IP-Symcon Mobile für iPhone, iPod touch und iPad im iTunes App Store |
| Mobile (Android) | Das Android Gegenstück zu IP-Symcon Mobile ist auch eine native App, die direkt über Google Play heruntergeladen werden kann.<br>IP-Symcon Mobile für Android Geräte bei Google Play |

## EINRICHTUNG

Um ein WebFront einzurichten muss ein WebFront Konfigurator vorhanden sein. Standardmässig wird mit der Installation ein WebFront erstellt und befindet sich im Objektbaum unter "Visualisierung Instanzen".



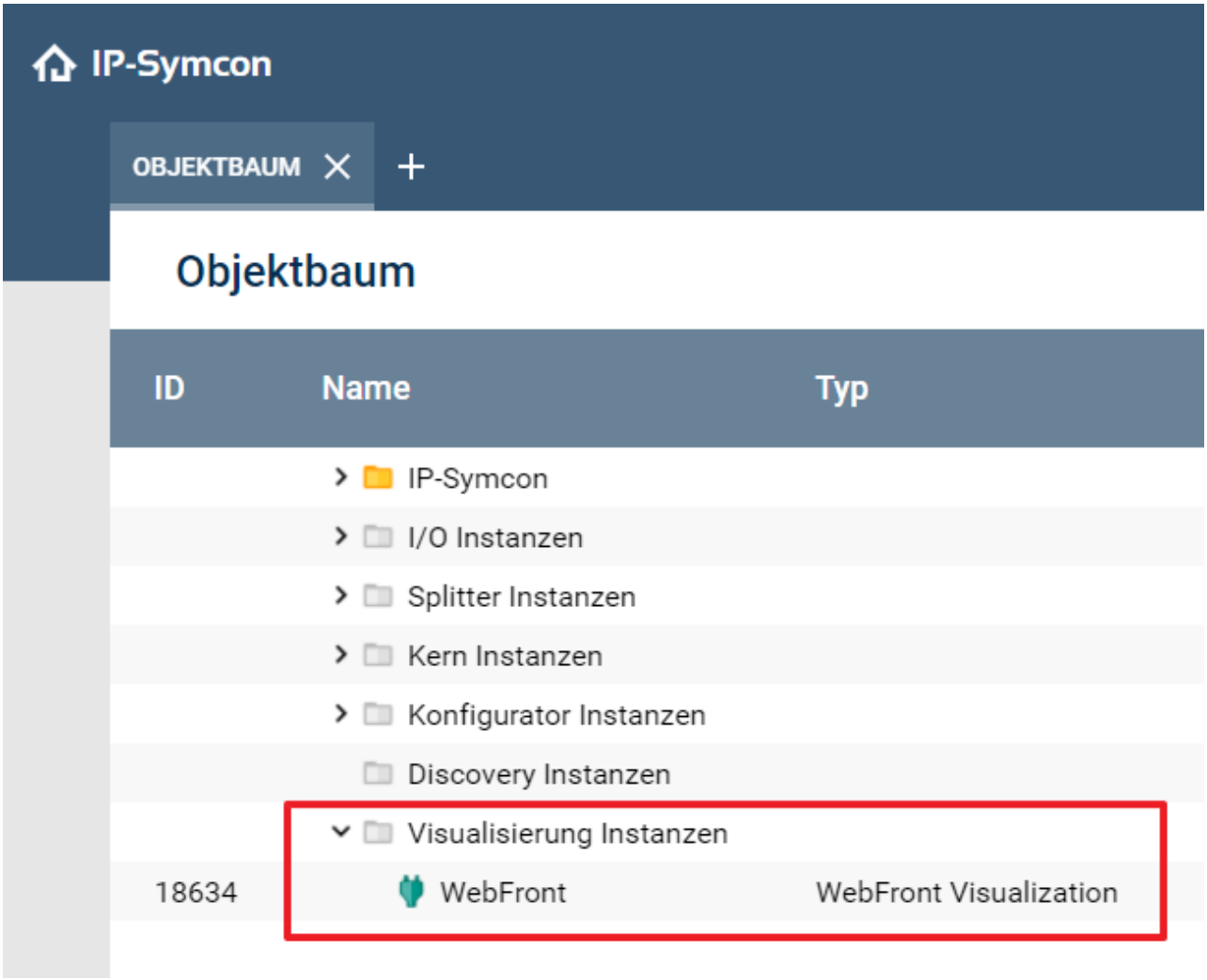

Sollen weitere WebFront eingerichtet werden, so kann dieses im Objektbaum über das "+" -> "Instanz" -> "WebFront Visualization" hinzugefügt werden. Dies erstellt eine neue WebFront Visualisierung, welche komplett individuell konfiguriert werden kann.

## KONFIGURATION

Grundsätzlich läuft das WebFront ohne weitere Konfiguration. Es können allerdings diverse Einstellungen gemacht werden. Die entsprechende Dokumentation kann den Unterseiten von diesem Dokument entnommen werden. Die generelle, versionsübergreifende Konfiguration wird weiter unten beschrieben.
Eine Auswahl von Icons für Objekte im WebFront ist unter WebFront Icons zu finden.

## KONFIGURATION ÖFFNEN

Die Konfiguration kann durch einen Doppelklick im Objektbaum auf die WebFront Visualisierung geöffnet werden.




Informationen über die einzelnen Einstellungen und Konfigurationsmöglichkeiten kann unter WebFront gefunden werden.

## VISUALISIERUNGEN

In IP-Symcon Basic ist eine Visualisierung verfügbar, die jedoch nur sehr eingeschränkt verändert werden kann. In der IP-Symcon Professional Edition können insgesamt bis zu fünf WebFront Visualisierungen erstellt werden, die eigenständig über die WebFront-Startseite aufgerufen und bei Bedarf mit einem Passwort abgesichert werden können. Eine Visualisierung kann als eigenständige Einheit - für z.b. ein Visualisierungsobjekt - erstellt werden, eine spezielle Ansicht für den Wartungstechniker darstellen oder einfach nur eine sicherheitsspezifische Trennung erzeugen. Pro Visualisierung sind dieselben Optionen verfügbar und können beliebig angepasst werden. In der Unlimited Edition sind beliebig viele Visualisierungen erstellbar.

# Web

Benötigt: IP-Symcon >= 3.0



Die Möglichkeit zur Konfiguration des WebFronts benötigt eine **IP-Symcon Professional** oder **IP-Symcon Unlimited** Edition



Im WebFront können nur Objekte angezeigt und hinzugefügt werden, welche zuvor in der Verwaltungskonsole im Objektbaum erstellt wurden. Der WebFront Editor hat keine Möglichkeit neue Kategorien oder Objekte zu erstellen. Er dient nur dazu vorhandene Objekte nach eigenen Vorstellungen zu arrangieren.



Skripte, die durch das WebFront ausgeführt werden, beinhalten diese Systemvariablen



Es wird empfohlen im Objektbaum die Struktur des WebFronts mit Links aufzubauen. Diese können dann via Kategorie im WebFront-Editor ausgewählt werden.

## BEDIENUNG

Am oberen Bildschirmrand können im WebFront eingerichtete Elemente ausgewählt werden. Dies sind standardmäßig IP-Symcon zur Gebäudesteuerung und Wetter zur Anzeige der aktuellen Wetterdaten und Voraussagen des Deutschen Wetterdienstes. Ein Klick auf einen Modulnamen zeigt dieses im Mittelteil an. Der aktuelle Modulname wird heller hinterlegt. Im Mittelteil werden die Hauptelemente eines Moduls angezeigt. Die weiteren Leisten am oberen Bildschirmrand bieten Steuerungsmöglichkeiten zur aktuell ausgewählten Kategorie.

## WEBFRONT-EDITOR

## KONFIGURIEREN

Die Konfiguration ist in mehrere Teile aufgeteilt. In jedem Reiter können verschiedene Einstellungen angepasst werden. In diesem Artikel werden primär die Einstellungsmöglichkeiten für das normale WebFront vorgestellt.
In den einzelnen Reitern können folgende Einstellungen angepasst werden:

| REITER | BESCHREIBUNG |
| --- | --- |
| Editor (ab 5.0) | Startet den WebFront Editor |
| Darstellung | Konfiguration von einem Titel und Icon im Login-Birdschirm |
| Sicherheit | Ermöglich die Einstellung eines Passwortes zur Zugriffskontrolle und einer speziellen IP-AutoStart Funktion |
| Benachrichtigungen | Erlaubt das Versenden von Test-Nachrichten an konfigurierte Geräte und das Löschen. |
| Struktur (bis 4.4) | Konfiguration der einzelnen Elemente aus denen ein WebFront besteht |

## WEITERES WEBFRONT EINRICHTEN

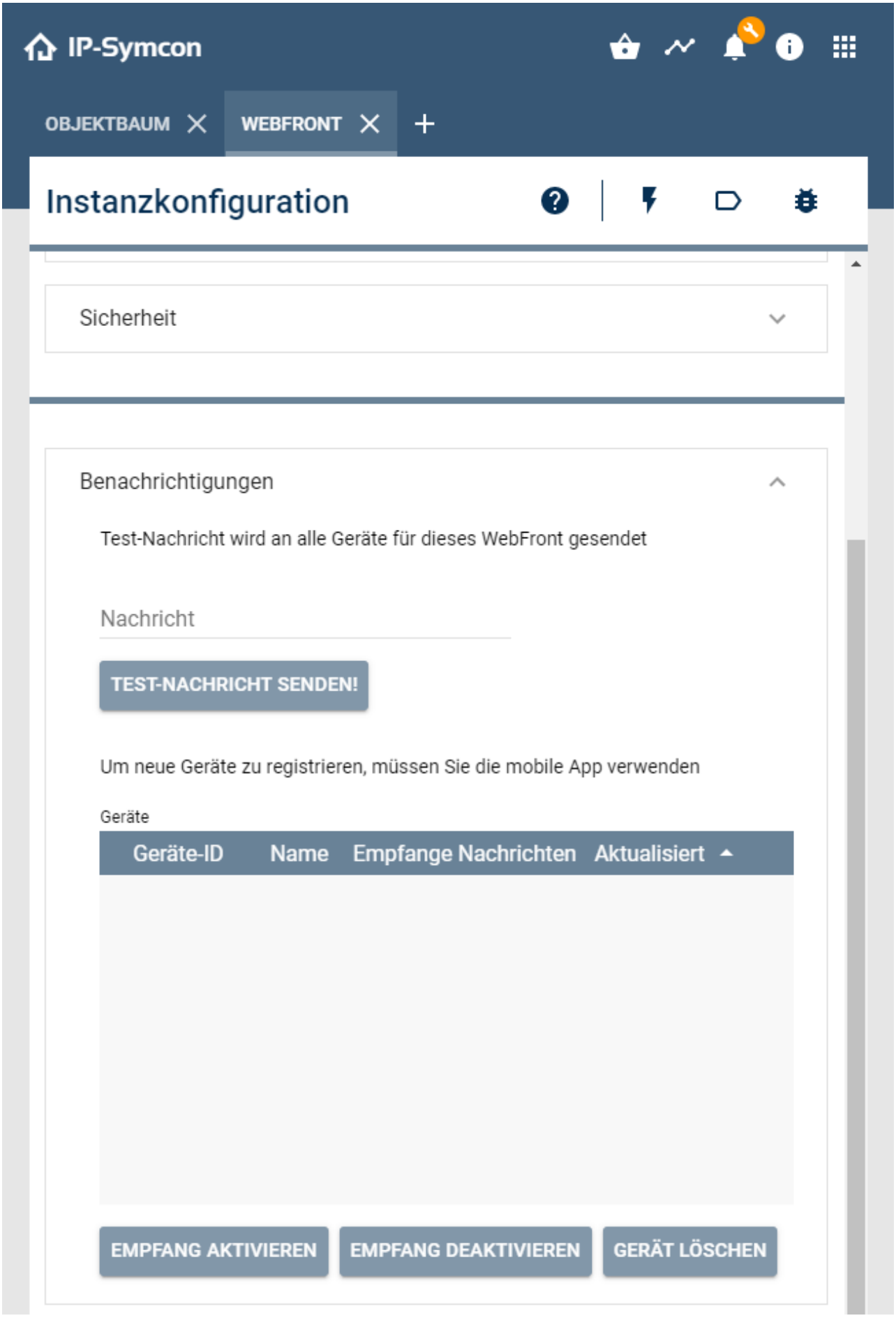


Um mehr als ein WebFront zu nutzen wird eine **IP-Symcon Professional** oder **IP-Symcon Unlimited** Edition benötigt.

Im Widget-Bereich "Visualisierung" in der Verwaltungskonsole kann die jeweilige WebFront Visualisierung geöffnet werden.
Um eine neue Visualisierung und somit ein neues WebFront zu erstellen, kann man im Objektbaum auf "+" -> "Instanz" klicken. Der Name der Visualisierungs-Instanz ist auch der Anzeigename des WebFronts im Browser.

# Benachrichtigungen

Der Reiter bietet eine Übersicht über die registrierten Geräte. Es kann ausgewählt werden, ob das jeweilige Gerät Meldungen empfängt.
Auch kann eine Testnachricht verschickt werden.




## REGISTRIERUNG ZUM PUSH-NACHRICHTEN DIENST

Sobald ein Mobilgerät sich in einem WebFront das erste Mal erfolgreich einloggt, wird dieses Gerät automatisch zum Push-Nachrichten Dienst für diesen Konfigurator hinzugefügt. Wenn keine Push-Nachrichten an dieses Gerät versendet werden sollen, so muss dieses Gerät explizit für den Empfang im Konfigurator deaktiviert werden. Zu beachten ist, dass die Registrierung zum Push-Nachrichten Dienst unberührt bleibt, falls das Kennwort für den Konfigurator geändert wird oder der Mobilzugriff komplett deaktiviert wird. Wird ein Gerät komplett aus der Liste entfernt, so wird es beim nächsten Login wie ein unbekanntes Gerät behandelt und erneut zum Push-Nachrichten Dienst hinzugefügt.

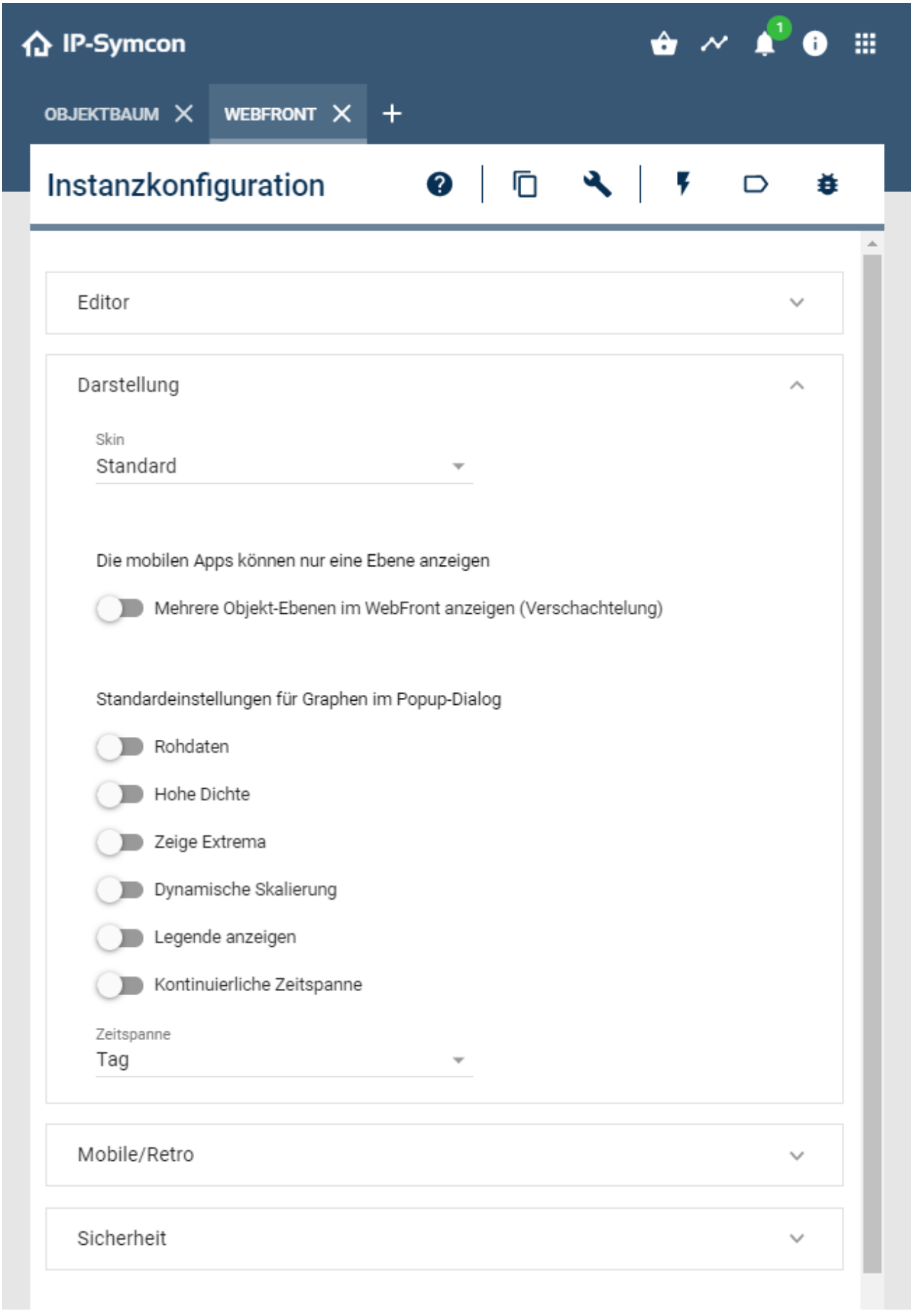
## NOTIFICATION CONTROL

Die Nachrichten werden via Notification Control versendet.
Weitere Informationen befinden sich unter Notification Control .

# Darstellung

## DARSTELLUNG

Die Darstellung kümmert sich um das Aussehen und mögliche Standardeinstellungen des WebFronts.




## NAME, ICON, SICHTBARKEIT

Über "Konfigurieren" können verschiedene Standardeinstellung ausgewählt werden.

## SKIN

Ab IP-Symcon 3.0 ist es möglich das WebFront über Skins zu gestalten. Weitere Informationen sind im Skin Control.

## VERSCHACHTELUNG

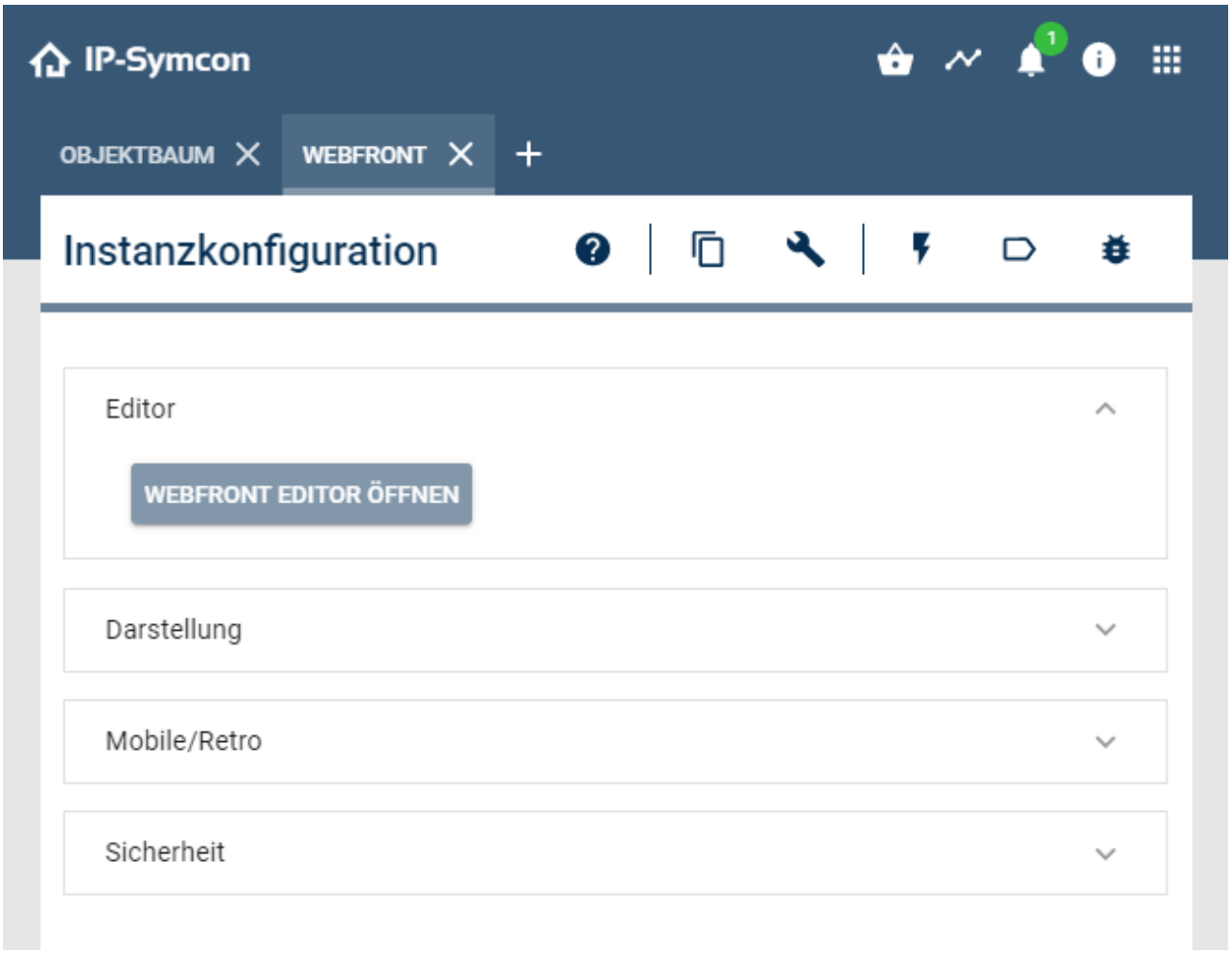
Diese Option ist seit IP-Symcon 5.5 standardmäßig aktiviert und ermöglicht die Darstellung von Verschachtelung (z.B. Instanzen unterhalb von Instanzen). Die mobilen Apps in der Version 5.5 und neuer unterstützen diese Option ebenfalls. Diese Option kann nach dem Aktivieren nicht mehr deaktiviert werden, da diese langfristig entfernt werden soll.

## STANDARDEINSTELLUNG FÜR POPUP-GRAPHEN

Hier kann eine Standardauswahl für Graphen im Popupfenster festgelegt werden. Was welcher Begriff genau bedeutet kann unter Darstellungsarten von Charts nachgelesen werden.

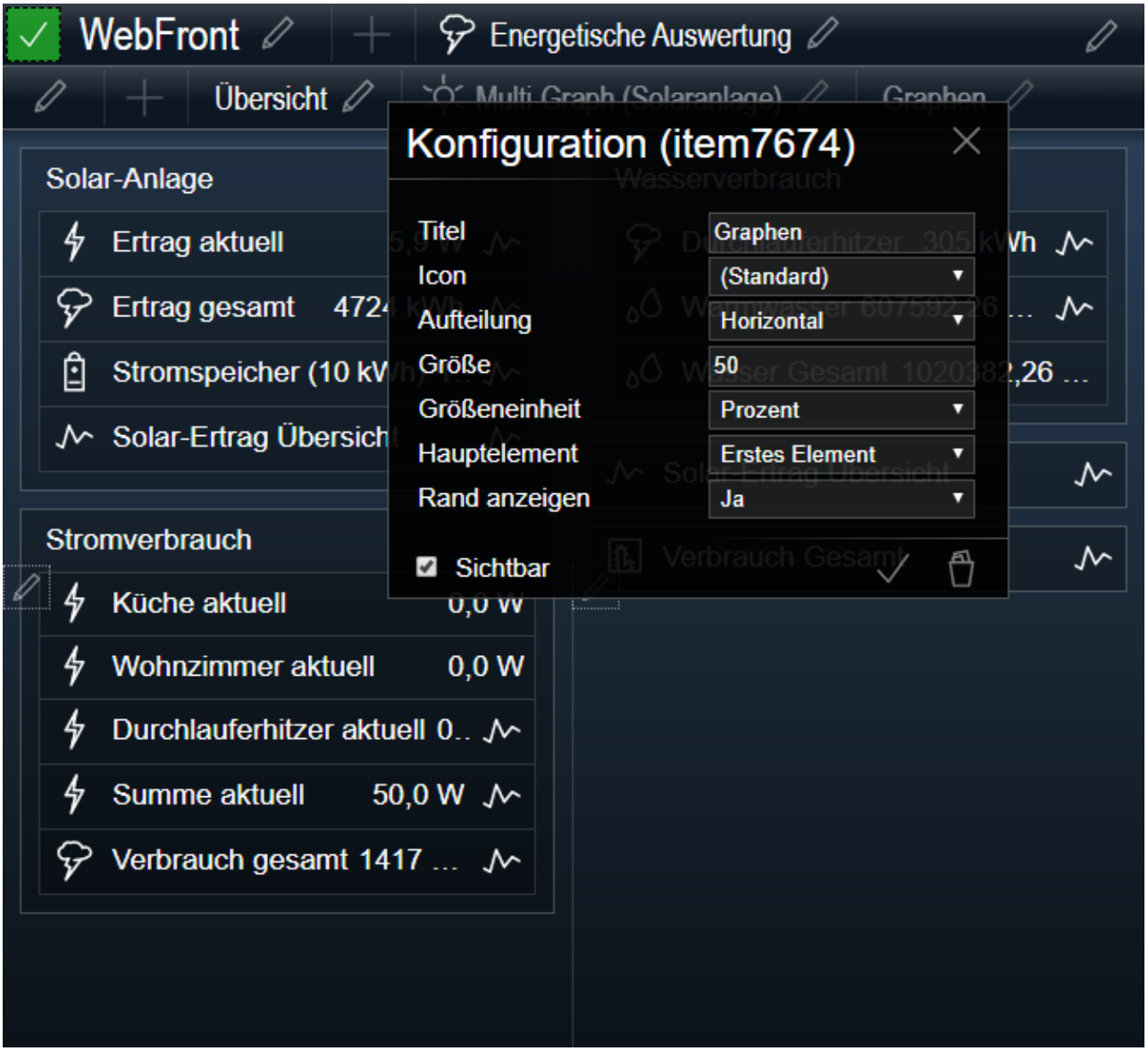
# Editor

Über den Button "WebFront Editor öffnen" unter "Editor" im WebFront Konfigurator kann der Editor gestartet werden.




Um den WebFront Editor vollständig nutzen zu können muss mindestens eine Professional Lizenz vorhanden sein. Ansonsten ist der Umfang extrem reduziert.

## VERWENDUNG DES EDITORS



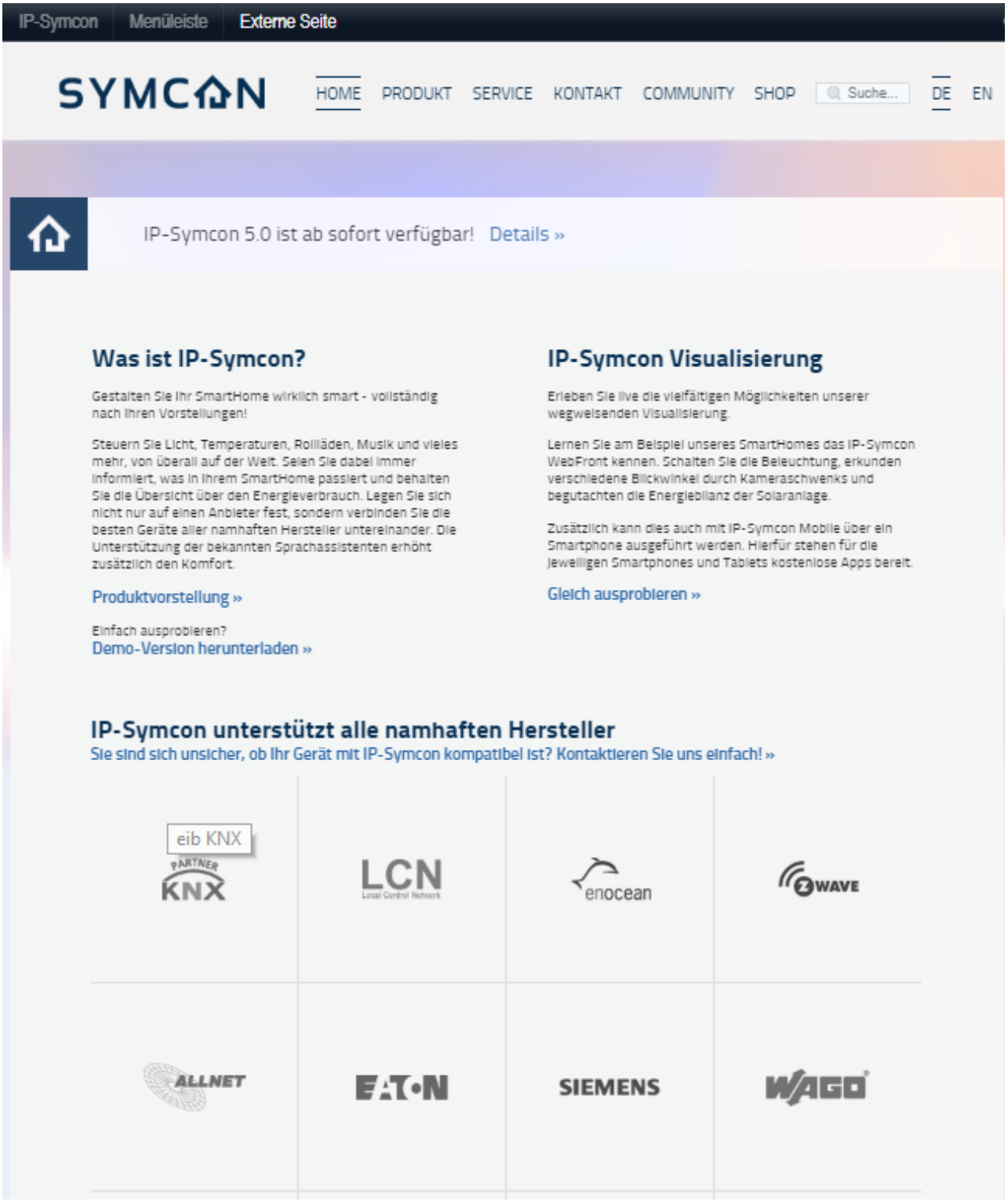

Im Editor wird das WebFront wie gewohnt angezeigt und ist um Konfigurationselemente erweitert. So kann durch einen Klick auf das Plus ein weiteres Element hinzugefügt werden. Hinzugefügte Elemente können über einen Klick auf den Stift konfiguriert werden. Es öffnet sich ein Dialog, in dem die Parameter des Elements eingestellt werden können. Die einzelnen Parameter der verschiedenen Elemente sind auf den einzelnen Dokumentationsseiten erklärt. Nach einer Anpassung der Parameter werden diese durch einen Klick auf den Haken im Dialog übernommen. Alternativ kann ein Element im Konfigurationsdialog auch durch einen Klick auf den Mülleimer gelöscht werden. Abschließend müssen die Änderungen durch einen Klick auf den grünen Haken oben links bestätigt werden, ansonsten werden sie nicht gespeichert.

## ÜBERSICHT DER VERFÜGBAREN ELEMENTE

| ELEMENT | BESCHREIBUNG |
| --- | --- |
| Kategorie | Über dieses Element lässt sich eine Kategorie und all seine Unterobjekte im WebFront anzeigen. Die ID der anzuzeigenden Kategorie kann über den Startkategorie Parameter verändert werden. |
| Graph | Über dieses Element werden die im Archiv Modul eingestellten Graphen direkt angezeigt. Normalerweise können Graphen über das Icon links vom Wert der Variable aufgerufen werden. Falls Graphen direkt in z.B. Menüleisten oder Seitenteiler eingebettet werden sollen, kann dies über das Graphen Element geschehen. |
| Externe Seite | Dieses Element fügt eine externe Seite in das WebFront über ein IFrame ein. Die Parameter-URL und Intervall sind verfügbar. |
| Inhaltswechsler | Über den Inhaltswechsler können mehrere Bilder, Streams, HTML-Boxen, Text-Boxen oder Mail-Instanzen(IMAP, POP3) innerhalb einer Seite umgeschaltet werden. |
| Seitenteiler | Über Seitenteiler kann die WebFront-Ansicht in zwei Teile aufgespaltet werden und in jeden Teil ein beliebig anderes Element eingefügt werden. Wenn Seitenteiler in Seitenteiler eingefügt werden, kann somit z.B. ein QuadView nachgebildet werden. |
| Menüleiste | Über Menüleisten können im oberen Bereich benutzerdefinierte Menüleisten erstellt werden, die auf weitere Elemente verweisen. |
| Wetter (DWD) | Zeigt das Wetter des DWD mit der Möglichkeit an, einzelne Bereiche Deutschands genauer auszuwählen und zwischen einem normalen Bild oder einem Radarbild zu wechseln. |
| Uhrzeit Widget | Zeigt die aktuelle Uhrzeit an. |
| Info Widget | Kann entweder dazu verwendet werden ein statisches Icon anzuzeigen oder verknüpft mit einer Variable deren Inhalt/Icon, welcher/welches je nach Variablenwert unterschiedlich sein kann. |
| Logout Widget | Dieses Element fügt ein Widget hinzu, über das man zum Auswahlbildschirm der WebFronts zurückgeleitet wird. |
| Idle Widget | Dieses Element fügt ein unsichtbares Widget hinzu, welches nach einer Zeit Untätigkeit auf eine bestimmte Seite wechselt. |

# Externe Seite

Dieses Element ermöglicht die Anzeige externer Webseiten. Es sind sowohl lokal gespeicherte Seiten, als auch externe Internetseiten darstellbar. Auch ist es möglich, die Seite automatisch in einem bestimmten Intervall aktualisieren zu lassen.




## EXTERNE INTERNETSEITEN

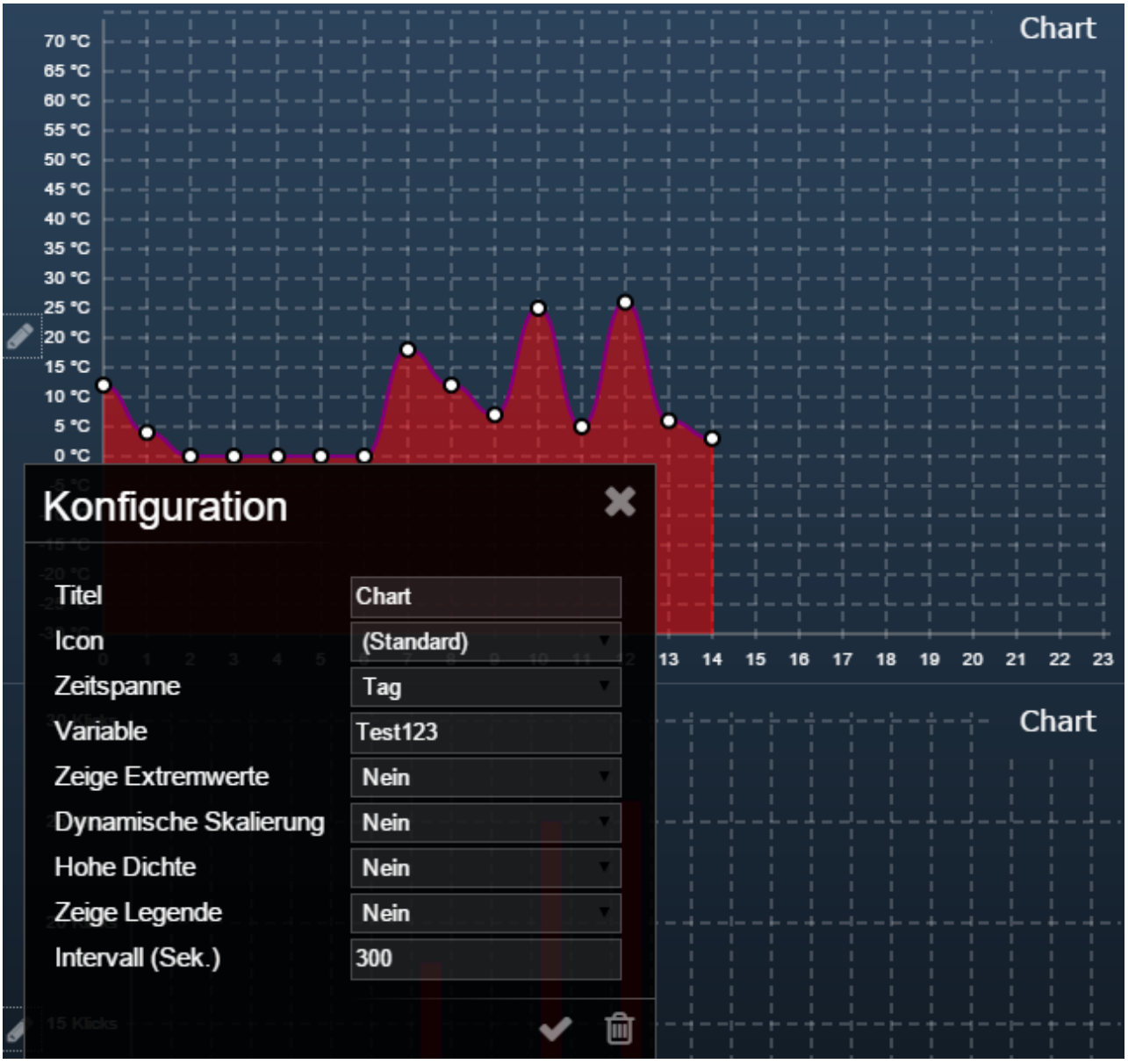
Externe Internetseiten sind im normalen URL-Format anzugeben, also z.B. https://www.symcon.de.

## EIGENSCHAFTEN

**Titel:** Titel für das Element in der Menüleisten-Navigation. Standardmäßig wird “Externe Seite” angezeigt.
**Icon:** Icon, das in der Menüleisten-Navigation neben dem Titel angezeigt wird. Standardmäßig wird kein Icon angezeigt.
**URL:** Hier ist die Adresse der Seite anzugeben, deren Inhalt angezeigt werden soll.
**Intervall:** Optional kann hier das automatische Aktualisierungsintervall für die Seite in Sekunden angegeben werden.
**Sicherheit:** Optional kann das Sandboxing deaktiviert werden. Dies kann in einigen Fällen dazu führen, dass die externe Seite euer WebFront deaktiviert und sich selbst in voller Größe darstellt.

# Graph

Der Vollbild-Graph wird direkt im WebFront eingerichtet.
Er ermöglicht die Anzeige einer einzelnen geloggten Variable oder eines Medienobjekts vom Typ Chart .




## EIGENSCHAFTEN

**Titel:** Titel für das Element in der TabPane-Navigation. Wird kein eigener Wert angegeben, so wird automatisch der Objektname der gewählten Variablen angezeigt.
**Icon:** Icon, das in der TabPane-Navigation neben dem Titel angezeigt wird. Standardmäßig wird kein Icon angezeigt.
**Zeitspanne:** Der Anzeigebereich des Graphen kann hier auf Stunde, Tag, Woche, Monat oder Jahr gesetzt werden.
**Variable:** Hier ist eine geloggte Variable oder Chart-Mediendatei für die Anzeige auszuwählen.
**Zeige Extremwerte:** Gibt an, ob der Graph mit oder ohne Extremwertkurven, falls verfügbar, angezeigt wird.

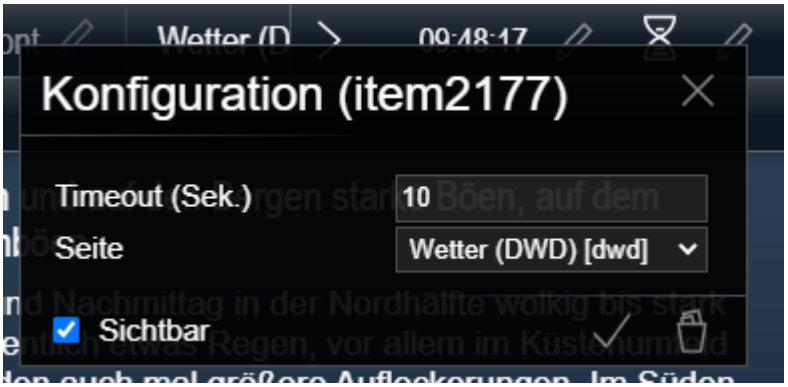
Dynamische Skalierung: Wird die dynamische Skalierung aktiviert, so wird der Graph dynamisch entsprechend des minimalen und des maximalen Werts im Zeitbereich skaliert. Andernfalls erfolgt die Skalierung entsprechend des im Variablenprofil der gewählten Variable angegebenen Minimums und Maximums.
Hohe Dichte: Erhöht die Datenpunktdichte, wodurch der Graph genauer wird. Funktioniert nur mit Line-Charts oder geloggten Variablen auf "Standard"-Logging.
Zeige Legende: Schaltet die Legende Ein/Aus.
Intervall (Sek.): Der dargestellte Graph wird automatisch entsprechend dem hier eingestellten Intervall in Sekunden aktualisiert. Standardmäßig geschieht dies alle 300 Sekunden.

# Idle Widget

Das Idle-Widget bietet die Möglichkeit im Widget-Bereich einer Menüleiste einen unsichtbaren Timer zu erstellen. Dieser ist nur im Editormodus sichtbar und wechselt nach einer einstellbaren Zeit Untätigkeit die angezeigte Seite.

> Seitenteiler und die Menüleisten selbst können nicht direkt angesprungen werden. Dies sind rein optische Elemente. Elemente innerhalb des Seitenteiler oder z.B. hinzugefügte Kategorien in der Menüleisten können ausgewählt werden.

## EIGENSCHAFTEN




**Timeout (sec):** Nach welcher Zeit Untätigkeit auf die eingestellte Seite gewechselt werden soll.
**Page:** Die Seite auf welche gewechselt werden soll.

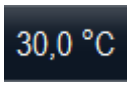
# Info Widget

Das Info-Widget bietet die Möglichkeit im Widget-Bereich einer Menüleiste, ein festes Icon, ein dynamisches Icon zu einer Variable oder einen textuellen Variablenwert anzuzeigen. Zusätzlich kann ein Skipt ausgewählt werden, das beim Anklicken des Widgets ausgeführt wird.

## ANZEIGEVARIANTEN

ICON-DARSTELLUNG TEXT-DARSTELLUNG






Ein festes Icon kann mit der Eigenschaft Icon zur Anzeige ausgewählt werden. Hierbei ist zu beachten, dass weder Typ noch Variable angegeben werden dürfen.
Ein dynamisches Icon wird angezeigt, wenn eine Variable angegeben und Typ auf Icon gesetzt ist. Welches Icon angezeigt wird und welche Icons standardmäßig verfügbar sind, ist unter Icons beschrieben. Soll das Icon “Transparent” angezeigt werden, so wird das Widget ausgeblendet. Auf diese Weise lassen sich Informations-Icons dynamisch einblenden, z.B. wenn ein Gerät einen Batteriewechsel benötigt.
Der aktuelle Variablenwert wird textuell anhand des der Variable zugeordneten Variablenprofils formatiert, wenn eine Variable angegeben ist und der Typ auf Text eingestellt ist.

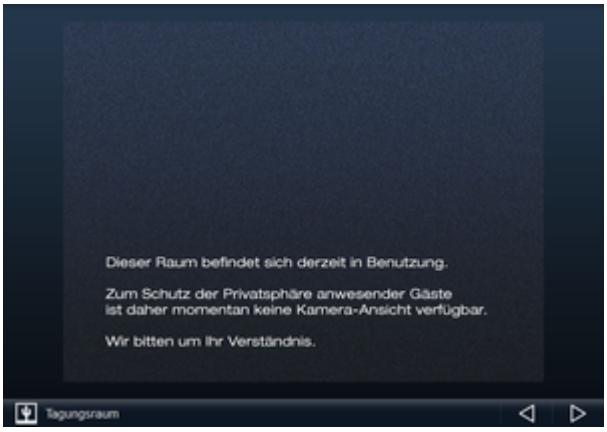
## EIGENSCHAFTEN

**Icon:** Hier kann ein festes Icon zur Darstellung ausgewählt werden.
**Typ:** Der Typ bestimmt, ob eine Variable als Text oder als Icon dargestellt werden soll.
**Variable:** Hier kann eine Variable ausgewählt werden, die entsprechend des gewählten Typs dargestellt wird.
**Bei Klick:** Ein Skript, das bei Klick auf das Widget ausgeführt wird, lässt sich hier einstellen.

# Inhaltswechsler

Der Inhaltswechsler dient zur Anzeige folgender Objekte:
Email-Instanzen (IMAP, POP3)
String-Variablen mit Variablenprofil ~TextBox oder ~HTMLBox
Medien-Objekte vom Typ Bild oder Stream

## DARSTELLUNG




Wird ein darstellbares Objekt direkt im Konfigurator als Quellobjekt zur Anzeige ausgewählt, so wird auch nur dieses Objekt angezeigt. Andernfalls werden alle darstellbaren direkten Unterobjekte des ausgewählten Quellobjekts zur Anzeige ausgewählt. Wird dabei mehr als ein darstellbares Unterobjekt gefunden, so wird eine Auswahlleiste zum Umschalten zwischen den Unterobjekten eingeblendet. Bei editierbaren String-Variablen vom Typ ~TextBox erscheint dabei auch ein Button, um das On-Screen-Keyboard zum Bearbeiten des Textes aufzurufen. Das Umschalten zwischen den darzustellenden Objekten kann auch automatisch erfolgen, wenn ein Änderungsintervall angegeben wird.

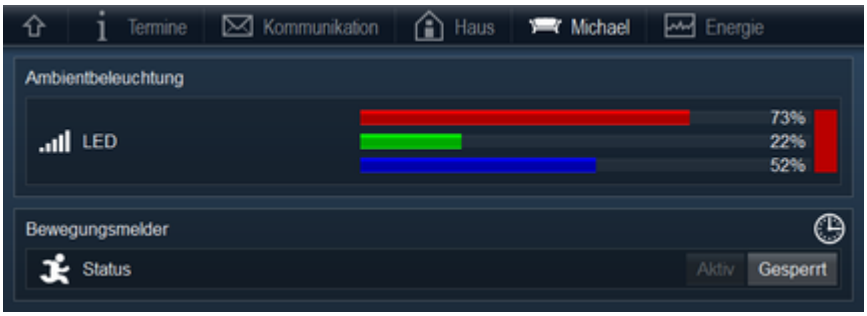
## EIGENSCHAFTEN

**Titel:** Titel für das Element in der Menüleisten-Navigation. Wird kein eigener Wert angegeben, so wird automatisch der Objektname des gewählten Quellobjekts angezeigt.
**Icon:** Icon, das in der Menüleisten-Navigation neben dem Titel angezeigt wird. Standardmäßig wird kein Icon angezeigt.
**Quellobjekt:** Das Objekt, das direkt angezeigt werden soll oder dessen Unterobjekte angezeigt werden sollen.
**Änderungsintervall:** Intervall in Sekunden, in dem automatisch zwischen den darzustellenden Objekten gewechselt wird. Der Standardwert von "0" deaktiviert die Funktion.
**Hochskalieren:** Gibt an, ob anzuzeigende Bilder automatisch hochskaliert werden, wenn Platz verfügbar ist. Standardmäßig wird nicht hochskaliert.
**Runterskalieren:** Ist diese Option aktiviert, werden Bilder, wenn nicht genügend Platz verfügbar ist, automatisch runterskaliert. Die Option ist standardmäßig aktiviert.
**Seitenverhältnis beibehalten:** Sofern Hochskalieren oder Runterskalieren aktiviert ist, ist mit dieser Option einstellbar, ob das Seitenverhältnis der zu skalierenden Bilder beim Skalieren beibehalten wird, oder nicht. Standardmäßig wird das Seitenverhältnis

beibehalten.

# Kategorie

Das Kategorie-Element ermöglicht die Anzeige der Objekte, die sich in einer Kategorie befinden, sowie die Navigation in Unterkategorien.




## OBJEKTBEREICH

Im Objektbereich werden die Unterobjekte der zur Ansicht ausgewählten Kategorie dargestellt. Nähere Informationen und mögliche Darstellungsarten, sind im Bereich Objekt-Darstellung zu finden.

## NAVIGATIONSLEISTE

Sofern die angezeigte Kategorie sichtbare Unterkategorien hat, wird automatisch eine Kategorie-Navigationsleiste angezeigt. Wird über die Navigation eine Unterkategorie ausgewählt, so wird im Objektbereich deren Inhalt dargestellt. Besitzt eine so angezeigte Kategorie ebenfalls sichtbare Unterkategorien, so wird die Navigation automatisch erweitert. Selbstverständlich werden auch per Link-Objekt verknüpfte Kategorien in der Navigationsleiste angezeigt. Über die Eigenschaft “Zeige Navigation” lässt sich die Kategorieleiste bei Bedarf ausblenden.

## EIGENSCHAFTEN

**Titel:** Titel für das Element in der Menüleisten-Navigation. Wird kein eigener Wert angegeben, so wird automatisch der Objektname der gewählten Startkategorie gesetzt. Wird der Wert gesetzt, aber der Text leer gelassen, so wird der Titel nicht angezeigt.
**Icon:** Icon, das in der Menüleisten-Navigation neben dem Titel angezeigt wird. Standardmäßig wird kein Icon angezeigt.
**Startkategorie:** Kategorie, deren Inhalt vom Element dargestellt wird. Standardmäßig wird die Hauptkategorie mit "ID 0" angezeigt.
**Zeige Navigation:** Mit dieser Option lässt sich steuern, ob die Navigationsleiste angezeigt wird, sofern sich unterhalb der Startkategorie Unterkategorien befinden. Standardmäßig wird die Navigationsleiste angezeigt.

## BESONDERHEITEN

Wird der Titel auf einen leeren Text gesetzt und kein Icon angegeben, so wird das Kategoriemodul in der Navigation nicht angezeigt, bleibt aber über WFC_SwitchPage erreichbar.

Soll eine Kategorie im WebFront zwar mit Icon aber ohne Titel angezeigt werden, muss ein Leerzeichen im Titel (" ") angegeben werden.

# Logout Widget

Das Logout-Widget bietet die Möglichkeit im Widget-Bereich einer Menüleiste, einen Button anzuzeigen. Bei Benutzung wird man auf den WebFront-Login zurückgebracht.

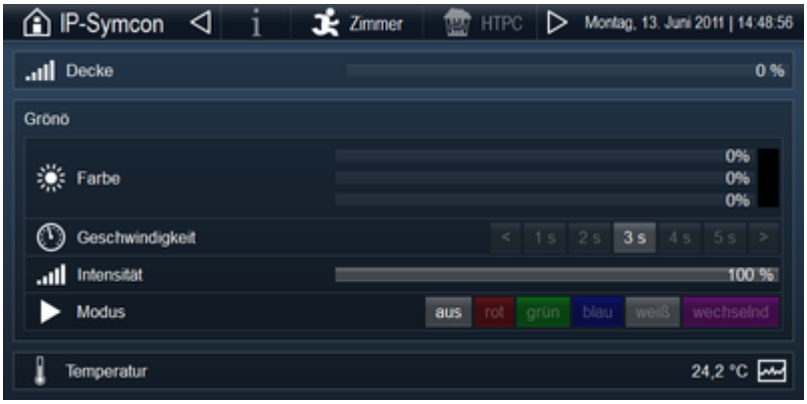
## EIGENSCHAFTEN

**Timeout (Sek.):** Timeout, welches bei Inaktivität automatisch einen Logout ausführt. Bei Wert = 0 ist dieser ausgeschaltet.

# Menüleiste

Das Menüleiste ist ein Element, das die Navigation zwischen allen anderen Typen von Elementen ermöglicht, sowie Widgets darstellen kann.

## DARSTELLUNG




Am oberen Rand der Menüleiste befindet sich eine Navigationsleiste, darunter der Inhaltsbereich. In der Navigationsleiste werden auf der linken Seite zunächst, sofern gewählt, das *"Untericon"* und der *"Untertitel"* angezeigt. Rechts daneben folgt der Navigationsbereich, der für alle den Menüleisten untergeordneten Elemente anklickbare Felder mit deren Icon und Titel, sofern eingestellt, anzeigt. Durch Anklicken eines Feldes wird im Inhaltsbereich der Menüleiste der Inhalt des zugeordneten Elements angezeigt. Am linken Rand der Navigationsleiste befindet sich der Widget-Bereich, in dem alle der Menüleiste untergeordneten Widgets angezeigt werden.
Die Menüleiste wird aufgrund der gebotenen Navigationsfähigkeiten in der Regel als das oberste Element einer WebFront-Konfiguration verwendet. Selbstverständlich kann die Menüleiste dabei auch weitere Menüleisten aufnehmen.
Reicht die Bildbreite nicht aus, um alle Navigationsfelder anzuzeigen, so werden automatisch anklickbare Pfeile zum Scrollen des Navigationsbereichs eingeblendet. Die Breite des Navigationsbereichs verkleinert sich dabei bis auf 100 Pixel. Danach wird der Widget-Bereich verkürzt.

## EIGENSCHAFTEN

**Titel:** Titel der Menüleiste in der Navigation einer gegebenenfalls übergeordneten Menüleiste. Standardmäßig wird "Menüleiste" angezeigt.
**Icon:** Icon, das in der Navigation einer gegebenenfalls übergeordneten Menüleiste neben dem Titel angezeigt wird. Standardmäßig wird kein Icon angezeigt.
**Untertitel:** Text, der am linken Rand der Navigationsleiste angezeigt wird. Standardmäßig wird kein Text angezeigt.
**Untericon:** Icon, das am linken Rand der Navigationsleiste angezeigt wird. Standardmäßig ist kein Icon ausgewählt.

# Seitenteiler

Der Seitenteiler ist ein Layout-Element, das zwei Unterelemente horizontal oder vertikal aufgeteilt darstellen kann. Auf diese Weise lassen sich z.B. ein Inhalt-Wechsler für die Kamera-Ansicht neben einer Kategorie anzeigen. Da unterhalb eines Seitenteiler auch weitere Layout-Elemente wie andere seitenteiler und Menüleisten eingefügt werden können, ist eine individuelle Layout-Anpassung des WebFronts möglich.




## AUFTEILUNG DER UNTERELEMENTE

Mit der Eigenschaft "Aufteilung" lässt sich einstellen, ob die beiden Unterelemente des Seitenteilers übereinander angezeigt werden sollen (horizontale Aufteilung), oder nebeneinander (vertikale Aufteilung). Wie viel Platz die Unterelemente bekommen, wird nun über die Eigenschaften *"Größe"*, *"Größeneinheit"* und *"Verhältnis Ziel"* bestimmt. Mit *"Verhältnis Ziel"* lässt sich bestimmen, ob man sich bei dem über *"Größe"* und *"Größeneinheit"* einstellbaren Wert auf das erste oder zweite Unterelement beziehen möchte. Das angezielte Element bekommt dann den Platz zugewiesen, der über die Eigenschaft *"Größe"* in der eingestellten *"Größeneinheit"* angegeben ist.

## BEISPIEL

Es soll ein Bild mit einer Bildbreite von 640 Pixeln rechts neben einer Kategorie angezeigt werden. Hierzu gibt man dem Seitenteiler für die Eigenschaft "Aufteilung" den Wert Vertikal, setzt das *"Verhältnis Ziel"* auf zweites Element, *"Größe"* auf 640 und *"Größeneinheit"* auf Pixel. Nun erstellt man im WebFront-Konfigurator unterhalb des Seitenteilers ein Unterelement vom Typ Kategorie und ein Unterelement vom Typ Inhalt-Wechsler. Bei den Unterelementen ist die Positions so einzustellen, dass die Kategorie Position 0 und der Inhalt-Wechsler Position 1 erhält. Nun können die Unterelemente so konfiguriert werden, dass die gewünschte Kategorie und das gewünschte Bild angezeigt werden.

## EIGENSCHAFTEN

| EIGENSCHAFT | BESCHREIBUNG |
| --- | --- |
| Titel | Der Titel für das Element in der Menüleisten-Navigation. Standardmäßig wird "Seitenteiler" angezeigt. |
| Icon | Das Icon, welches in der Menüleisten-Navigation neben dem Titel angezeigt wird. Standardmäßig wird kein Icon angezeigt. |
| Aufteilung | Hier lässt sich auswählen, ob die Trennung der beiden Unterelemente horizontal oder vertikal erfolgen soll. Standardmäßig wird horizontal getrennt. |
| Größe | Die Größe des für die Aufteilung gewählten Unterelements entsprechend der Größeneinheit. Der Standardwert ist 50. |
| Größeneinheit | Hier kann ausgewählt werden, ob sich die Größe auf Prozent oder auf Pixel bezieht. Standardmäßig werden Prozent genommen. |
| Hautpelement | Über diese Eigenschaft lässt sich auswählen, ob das erste oder das zweite Unterelement des Seitenteilers entsprechend der angegebenen Größenangabe angezeigt wird. Standardmäßig ist das erste Element ausgewählt. |
| Rand anzeigen | Mit dieser Eigenschaft lässt sich der standardmäßig zur Trennung der Unterelemente angezeigte Rand ein- oder ausblenden. |

# Uhrzeit Widget

Das Uhrzeit-Widget ermöglicht die Anzeige des aktuellen Datums und der aktuellen Uhrzeit im Widget-Bereich einer Menüleiste. Sofern kein benutzerdefiniertes Anzeigeformat eingestellt ist, richtet sich die Darstellung nach der im Webbrowser eingestellten Landessprache. Es kann dabei für das Datumsformat und das Zeitformat jeweils zwischen den Standardformatierungen "Versteckt", "Kurz", "Mittel", "Lang" und "Komplett" gewählt werden. Das benutzerdefinierte Format ist bei Bedarf, entsprechend der datePattern-Tabelle unter dojotoolkit.org/reference-guide/dojo/date/locale/format.html einzugeben.




## EIGENSCHAFTEN

**Datumsformat:** Hier kann ein vordefiniertes Muster für die Formatierung des Datums ausgewählt werden.
**Zeitformat:** Wie auch beim Datumsformat lässt sich hier ein vordefiniertes Muster für die Formatierung der Uhrzeit auswählen.
**Benutzerdefiniertes Format:** Optional kann hier ein benutzerdefiniertes Format angegeben werden.

# Wetter (DWD)

Das Wetter-Element ermöglicht die Anzeige der Wettervorhersage des Deutschen Wetterdienstes für Deutschland und die einzelnen Bundesländer in Text und Bild.




## NAVIGATIONSLEISTE

Im oberen Bereich befindet sich die Navigationsleiste. Hier lässt sich, sofern über die Eigenschaft “Region” eingestellt, zwischen dem Wetterbericht für Deutschland und dem gewählten Bundesland wechseln. Zusätzlich lässt sich der Vorhersagezeitraum zwischen Heute, Morgen, 3. Tag und 4. Tag wechseln.

## VORHERSAGEBEREICH

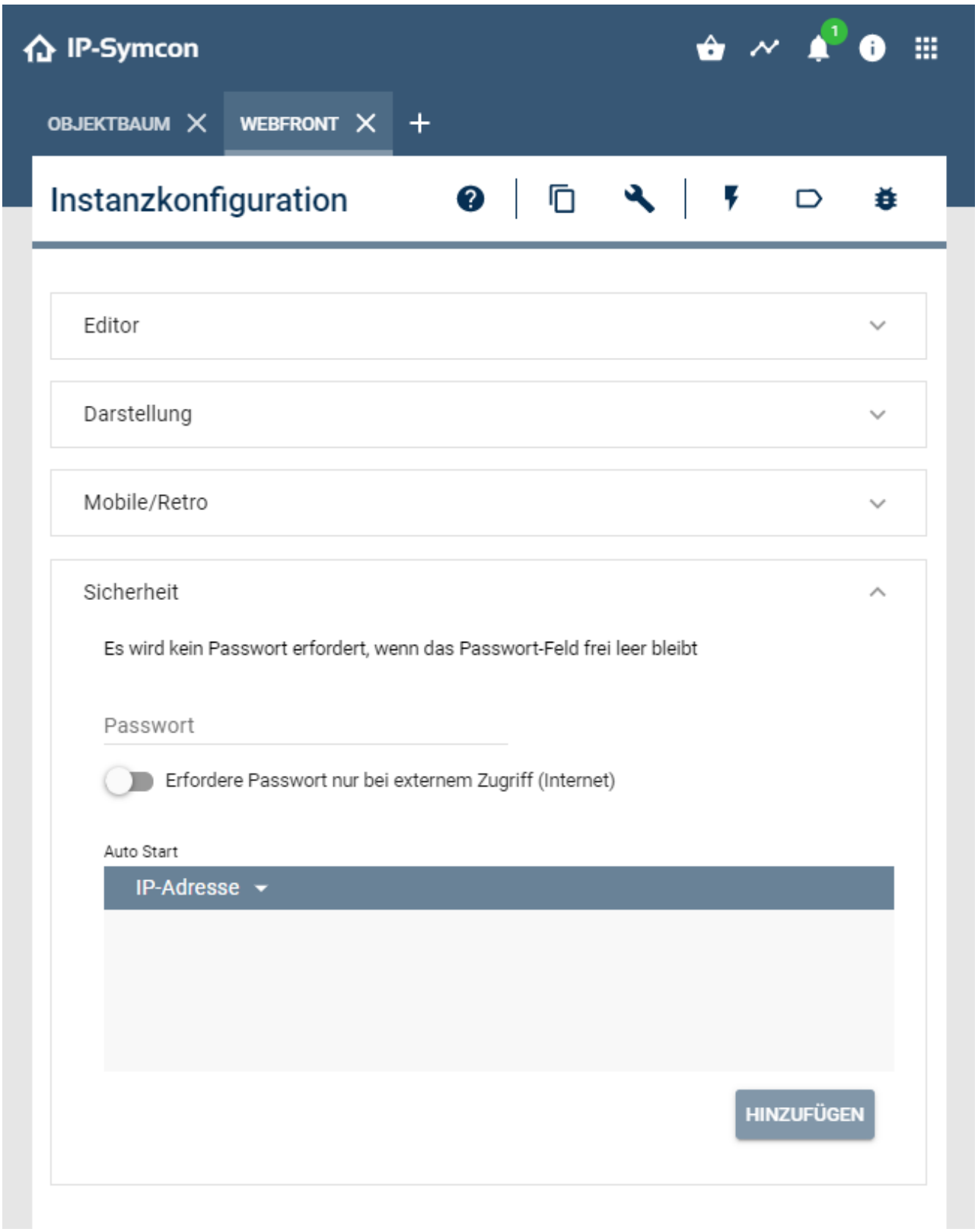
Im Vorhersagebereich ist rechts der Vorhersagetext zu lesen und links daneben eine passende Wetterkarte.

## EIGENSCHAFTEN

**Titel:** Titel für das Element in der Menüleisten-Navigation. Standardmäßig wird “Wetter” angezeigt.
**Icon:** Icon, das in der Menüleisten-Navigation neben dem Titel angezeigt wird. Standardmäßig wird kein Icon angezeigt.
**Region:** Optional kann hier ein Bundesland ausgewählt werden, dessen Wettervorhersage sich anstelle der deutschlandweiten Vorhersage in der Navigationsleiste anzeigen lässt.

# Sicherheit

In diesem Reiter kann eine Passwortabfrage für das WebFront und dessen Editor eingerichtet werden.
Weitere Sicherheitsinformationen können unter Sicherheit eingesehen werden.




## ERFORDERE PASSWORT

Für jeden Konfigurator kann ein Passwort als Zugriffschutz vergeben werden. Dieses Passwort wird vor dem Zugriff des jeweiligen Konfigurators gefordert, oder bei der Konfiguration innerhalb der mobilen Apps. Zusätzlich kann die Spezialoption aktiviert werden, dass das Passwort nicht innerhalb des lokalen Netzwerk gefordert wird. Dieses Passwort gilt nur für den Zugriff auf das System und dessen Komponenten. Die Abfrage der verfügbaren Konfiguratoren (Mobile) und der WebFront-Startseite (Übersicht) ist jederzeit ohne Passwort möglich.

## AUTOSTART FÜR IP-ADRESSE

Eine besondere Funktion ist der Auto-Start, welcher einen Konfigurator direkt ohne Passwortabfrage öffnet, wenn die IP-Adresse mit einer der eingetragenen IP-Adressen in dieser Liste übereinstimmt. Über diese Funktion kann z.B. auf einem Touch-Panel ein WebFront direkt geöffnet werden, ohne den normalen Auswahldialog angezeigt zu bekommen und ohne die Passwortabfrage beantworten zu müssen. Es ist zu beachten diese Funktion nur innerhalb eines sicheren Netzwerks zu benutzen.



Ab IP-Symcon 5.5 wird auch die CIDR Notation unterstützt, sodass z.B. 192.168.1.0/24 alle Adresse von 192.168.1.1 bis 192.168.1.255 enthält.



Diese Funktion steht nicht zur Verfügung, sofern Anfragen an IP-Symcon über einen Reverse-Proxy oder über den Connect-Dienst gehen, da der Ursprung dann nicht sicher und zuverlässig verifiziert werden kann. Eine Ausnahme ist der CIDR Wert 0.0.0.0/0, welcher alle Anfragen unabhängig der Herkunft umleitet und somit grundsätzlich unsicher ist. Dieser wird auch für die o.g. Ausnahmen ausgewertet und leitet somit alle Anfragen unabhängig des Ursprungs weiter.

### AUFRUF ÜBER URL

Es gibt Situationen (z.B. in Verbindung mit NAT) bei denen die Autostart via IP nicht funktioniert. In diesen Fällen kann eine URL mit dem WebFront-Passwort gestartet werden.

```
// Grundgerüst für den URL Aufruf
http://<DeineIP>:3777/?password=<Passwort>#<WebFrontID>

// Beispiel als Start mit dem Chromium Browser
// Passwort: "geheim", WebFrontInstanzID: = 12345
chromium-browser --start-fullscreen 192.168.1.123:3777/?
password=geheim#12345
```

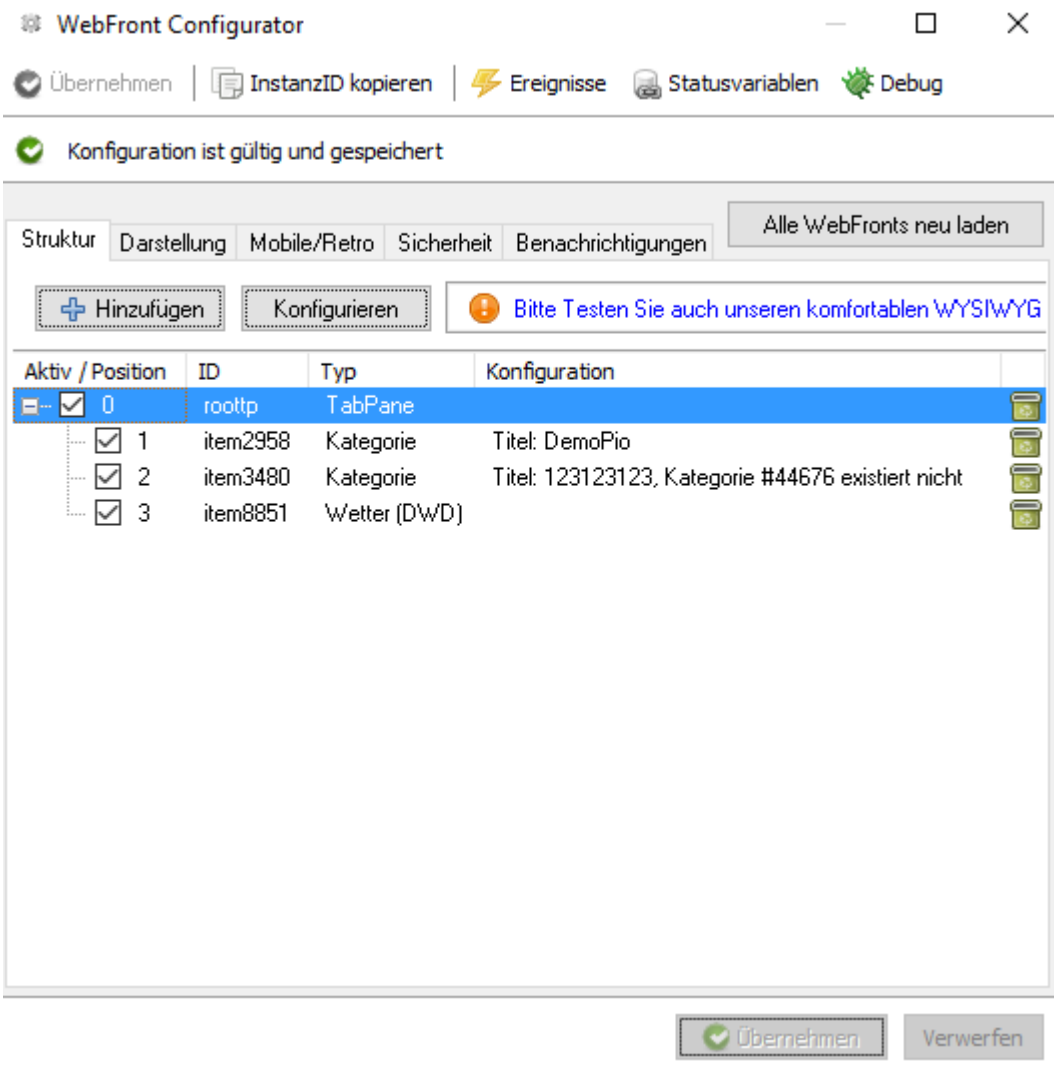


Achtung dies birgt ein Sicherheitsrisiko, da die URL das WebFront Passwort in Klartext überträgt und in der URL sichtbar ist.

# Struktur (veraltet)

Diese Funktion ist nur bis Version 4.4 vorhanden und wurde komplett durch den Editor abgelöst.

Der Konfigurator kann **Elemente** enthalten, die in der Visualisierung als Reiter ausgehend vom oberen linken Rand anfangen und zusätzlich **Widgets** enthalten, die in einem extra Bereich am oberen rechten Rand beginnen. Dieser Bereich ist dem SysTray von Windows ähnlich.
Ein Element oder Widget kann über den **Hinzufügen** Button eingefügt und entsprechend konfiguriert werden. Ein Element besteht aus einer eindeutigen ID, die automatisch vergeben wird, einer Position und einer dem Element entsprechenden Konfiguration.
Die ID kann vom Benutzer beim Erstellen angegeben werden, um das Element bei Aktionen besser wiedererkennen zu könne. Die ID hat keinen Zusammenhang mit den ObjektIDs der Instanzen oder Variablen.
Die Position gibt jeweils die Position im Reiter oder der Widget-Leiste an und muss einen Wert größer gleich Null erhalten. Wenn zwei Elemente die gleiche Positionsnummer haben, wird die Reihenfolge durch den Titel des Elements definiert.




## ÜBERSICHT DER VERFÜGBAREN ELEMENTE

| ELEMENT | BESCHREIBUNG |
| --- | --- |
| Kategorie | Über dieses Element lässt sich eine Kategorie und all seine Unterobjekte im WebFront anzeigen. Die ID der anzuzeigenden Kategorie kann über den Startkategorie Parameter verändert werden. |
| Graph | Über dieses Element werden die im Archiv Modul eingestellten Graphen direkt angezeigt. Normalerweise können Graphen über das Icon links vom Wert der Variable aufgerufen werden. Falls Graphen direkt in z.B. Menüleisten oder Seitenteiler eingebettet werden sollen, kann dies über das Graphen Element geschehen. |
| Externe Seite | Dieses Element fügt eine externe Seite in das WebFront über ein IFrame ein. Die Parameter-URL und Intervall sind verfügbar. |
| Inhaltswechsler | Über den Inhaltswechsler können mehrere Mediendateien, Strings, HTMLBoxen, StringBoxen, Mail-Instanzen oder externe Seiten innerhalb einer Seite umgeschaltet werden. |
| Seitenteiler | Über Seitenteiler (ehemals Container) kann die WebFront-Ansicht in zwei Teile aufgespaltet werden und in jeden Teil ein beliebig anderes Element eingefügt werden. Wenn Seitenteiler in Seitenteiler eingefügt werden, kann somit z.B. ein QuadView nachgebildet werden. |
| Menüleiste | Über TabPanes können im oberen Bereich benutzerdefinierte Menüleisten erstellt werden, die auf weitere Elemente verweisen. |
| Wetter (DWD) | Zeigt das Wetter des DWD mit der Möglichkeit an, einzelne Bereiche Deutschands genauer auszuwählen und zwischen einem normalen Bild oder einem Radarbild zu wechseln. |
| Uhrzeit Widget | Zeigt die aktuelle Uhrzeit an. |
| Info Widget | Kann entweder dazu verwendet werden ein statisches Icon anzuzeigen oder verknüpft mit einer Variable deren Inhalt/Icon, welcher/welches je nach Variablenwert unterschiedlich sein kann. |
| Logout Widget | Dieses Element fügt ein Widget hinzu, über das man zum Auswahlbildschirm der WebFronts zurückgeleitet wird. |
| Idle Widget | Dieses Element fügt ein unsichtbares Widget hinzu, welches nach einer Zeit Untätigkeit auf eine bestimmte Seite wechselt. |

# Mobile

Die Mobile Visualisierung für Mobile Geräte kann über den jeweiligen AppStore heruntergeladen werden und enthält dort auch die aktuellste Beschreibung der Funktionen der App. ( **Android:** IP-Symcon Mobile bei Google Play / **iOS:** Symcon Visualization für iPhone, iPod touch und iPad im iTunes App Store )
Diese Seite soll als Beschreibung und Referenz für häufig gestellte Fragen und Funktionen dienen, welche mögliche Unterschiede der Apps zueinander aufführen. Desweiteren stellt keine App Anspruch auf volle Kompatibilität zum WebFront. Diese dienen als Erweiterung zu IP-Symcon, um schnell und unkompliziert auf die Grundfunktionen des Systems zugreifen zu können. Für eine vollständige Visualisierung kann und muss auf das WebFront zurückgegriffen werden. Hier ist auch keine Änderung vorgesehen, da die Mobile Applikationen mit immer mehr werdenden Funktionen langsamer werden würde. Das würde aber mit dem Hauptmerkmal “Geschwindigkeit” kollidieren.
Die Dokumentation der Push-Nachrichten befindet sich im Abschnitt Notification Control



Es ist zu beachten, dass der Zugriff auf das System über die mobilen Apps standardmäßig deaktiviert ist. Es wird durch die Fehlermeldung "Kein gültiger Konfigurator angegeben!" darauf hingewiesen. Es muss im gewünschten WebFront Konfigurator die mobile Nutzung aktiviert werden.

## FUNKTIONEN, DIE ABSICHTLICH NICHT VERFÜGBAR SIND

Benutzerdefinierte Icons werden nicht geladen, da dies einen beträchtlichen Geschwindkeitsverlust bedeuten würde und außerdem mehrere Icongrößen z.B. für das Retina Display erfordern würde, die die wenigsten der User zur Hand haben. Mehrfachverschachtelungen sind nicht verfügbar. So soll auf mobilen Geräten die Displayfläche optimal genutzt werden.

## FUNKTIONEN, DIE SICH ANDERS ALS DAS WEBFRONT VERHALTEN

Boolean-Variablen werden standardmäßig als An/Aus (Switch, CheckBox) dargestellt. Dieses Verhalten kann über die WebFront Visualisierung Instanz verändert werden, sodass nur ~Switch Profile diese Darstellung haben.

## WEBFRONT FREIGABELINK

Über erstellte WebFront Freigabelinks lässt sich die App direkt starten und mit einem WebFront verknüpfen.
Sollte dieses WebFront in der App noch nicht vorhanden sein, wird automatisch ein Konfigurator eingerichtet.
Es gibt 2 Arten von Links:
Für Server ohne SSL: **symcon://www.webfront.info/#98765**
Für Server mit SSL: **symcons://mein.ssl-server.com/#12345**
Sollte der Server/Konfigurator nicht in der App Liste stehen, so wird dieser automatisch hinzugefügt und geöffnet.

Somit kann zum Beispiel, ohne dass manuelle Einstellungen notwendig werden, dem Besuch Zuhause oder gar dem Tagungsteilnehmer ein schneller Zugriff auf das lokale WebFront ermöglicht werden.

## BEKANNTE FEHLER

Bei aktiviertem SSL können extern geladene Inhalte (z.B. Bilder) nicht geladen werden, wenn das Zertifikat nicht vom Browser akzeptiert wurde.
Bestimmte Profile und Status-/Variablen (z.B. bei HomeMatic Fensterkontakten oder LCN Relais) können in der Mobile App geschaltet werden. Diese Möglichkeit der Aktion wird im WebFront farbig (grün/rot) dargestellt und bedeutet, dass entweder ein Aktionsskript definiert oder dass eine Standardaktion verfügbar ist. Soll das Verhalten geändert werden, gibt dieses Thema im Forum entsprechende Antwort: Veränderbarkeit von unveränderlichen Variablen

## TIPPS & TRICKS

Falls beim Ladevorgang ein JSON-Fehler auftritt, muss überprüft werden, ob mindestens IP-Symcon 3.1 #3367 installiert wurde.

# Legacy-Icons

Das WebFront verwendet die Legacy-Icons, nicht die üblichen Icons der Kachelvisualisierung.

## ICONTYP

Es wird unterschieden zwischen "normalen" Icons und adaptiven Icons. Normale Icons ändern ihr Aussehen nicht, wobei adaptive Icons sich der gesetzten Variable je nach Wert im Aussehen anpassen.
Adaptive Icons orientieren sich dabei anhand der im Profil eingestellten Min/Max Werte. Diese werden als 0%-100% gehandhabt. Soll es als 100%-0% gehandhabt werden, muss ein ".Reversed" im Profilnamen angegeben werden.
Die einzelnen Adaptiven Icon (z.B. Intensity-25) können auch direkt angegeben werden und werden dann als "normales" Icon eingebunden.

## BENUTZERDEFINIERTE ICONS EINBINDEN

Ab IP-Symcon 3.0 werden benutzerdefinierte Icons über Skins eingebunden.
Weitere Informationen befinden sich im Entwicklerbereich als SDK: Skins SDK

## AUSWAHL IN DER VERWALTUNGSKONSOLE

Die Icon-Auswahl in der Verwaltungskonsole wurde auf die aktuellen Icons hin optimiert. Dennoch können die Legacy-Icons eingestellt werden, indem der Iconname manuell in das entsprechende Feld eingetragen wird.

## ÜBERSICHT DER ICONS

Folgende Icons sind im WebFront enthalten und können an den jeweiligen Stellen in IP-Symcon ausgewählt werden. Ein Sonderfall ist das Icon **Transparent**. Wird dieses Icon ausgewählt, so kann ein anderes Icon, welches niedriger in der Hierarchie ist (s.o.), überschrieben werden, so dass kein Icon angezeigt wird. Dies kann als Stilelement verwendet werden, um z.B. bei verlinken Objektkategorien kein Icon anzuzeigen, obwohl die Kategorie ein Icon zugewiesen hatte.

## ICONS

**Aircraft**

**Alert**

**ArrowRight**

**Backspace**

**Basement**

**Bath**

**Battery**

**Bed**

**Bike**

**Book**

**Bulb**

**Calendar**

**Camera**

**Car**

**Caret**

**Cat**

**Climate**

**Clock**

**Close**

**Closeall**

**Cloud**

**Cloudy**

**Cocktail**

**Cross**

**Database**

**Dining**

**Distance**

**DoctorBag**

**Dog**

**Doll**

**Dollar**

**Door**

**Download**

**Drops**

**Duck**

**Edit**

**Electricity**

**EnergyProduction**

**EnergySolar**

**EnergyStorage**

**ErlenmeyerFlask**

**Euro**

**Execute**

**Eyes**

**Factory**

**Favorite**

**Female**

**Fitness**

**Flag**

**Flame**

**FloorLamp**

**Flower**

**Fog**

**Garage**

**Gas**

**Gauge**

**Gear**

**Graph**

**GroundFloor**

**Handicap**

**Heart**

**Help**

**HollowArrowDown**

**HollowArrowLeft**

**HollowArrowRight**

**HollowArrowUp**

**HollowDoubleArrowDown**

**HollowDoubleArrowLeft**

**HollowDoubleArrowRight**

**HollowDoubleArrowUp**

**HollowLargeArrowDown**

**HollowLargeArrowLeft**

**HollowLargeArrowRight**

**HollowLargeArrowUp**

**Hourglass**

**HouseRemote**

**Image**

**Information**

**Intensity**

**Internet**

**Ips**

**Jalousie**

**Key**

**Keyboard**

**Kitchen**

**Leaf**

**Light**

**Lightning**

**Link**

**Lock**

**LockClosed**

**LockOpen**

**Macro**

**Mail**

**Male**

**Melody**

**Menu**

**Minus**

**Mobile**

**Moon**

**Motion**

**Move**

**Music**

**Network**

**Notebook**

**Ok**

**Pacifier**

**Paintbrush**

**Pants**

**Party**

**People**

**Plug**

**Plus**

**Popcorn**

**Power**

**Presence**

**Radiator**

**Raffstore**

**Rainfall**

**Recycling**

**Remote**

**Repeat**

**Return**

**Robot**

**Rocket**

**Script**

**Shift**

**Shower**

**Shuffle**

**Shutter**

**Sink**

**Sleep**

**Sleet**

**Snow**

**Snowflake**

**Sofa**

**Speaker**

**Speedo**

**Stars**

**Sun**

**Sunny**

**TV**

**Talk**

**Tap**

**Teddy**

**Tee**

**Telephone**

**Temperature**

**Thunder**

**Title**

**TopFloor**

**Tree**

**TurnLeft**

**TurnRight**

**Umbrella**

**Unicorn**

**Ventilation**

**WC**

**Warning**

**Wave**

**Wellness**

**WindDirection**

**WindSpeed**

**Window**

**XBMC**

## ADAPTIVE ICONS

| | | | | | | |
|---|---|---|---|---|---|---|
| **Battery** | **0%** | **50%** | **100%** | | | |
| **Door** | **0%** | **100%** | | | | |
| **EnergyStorage** | **0%** | **25%** | **50%** | **75%** | **100%** | |
| **Eyes** | **0%** | **100%** | | | | |
| **FloorLamp** | **0%** | **100%** | | | | |
| **Garage** | **0%** | **25%** | **100%** | | | |
| **Hourglass** | **0%** | **30%** | **60%** | **100%** | | |
| **Intensity** | **0%** | **25%** | **50%** | **75%** | **100%** | |
| **Jalousie** | **0%** | **50%** | **100%** | | | |
| **Light** | **0%** | **1%** | **25%** | **50%** | **75%** | **100%** |
| **Lock** | **0%** | **100%** | | | | |
| **Presence** | **0%** | **100%** | | | | |
| **Raffstore** | **0%** | **50%** | **100%** | | | |
| **Speaker** | **0%** | **1%** | **25%** | **50%** | **100%** | |
| **Speedo** | **0%** | **25%** | **50%** | **75%** | **100%** | |
| **Temperature** | **0%** | **25%** | **50%** | **75%** | **100%** | |

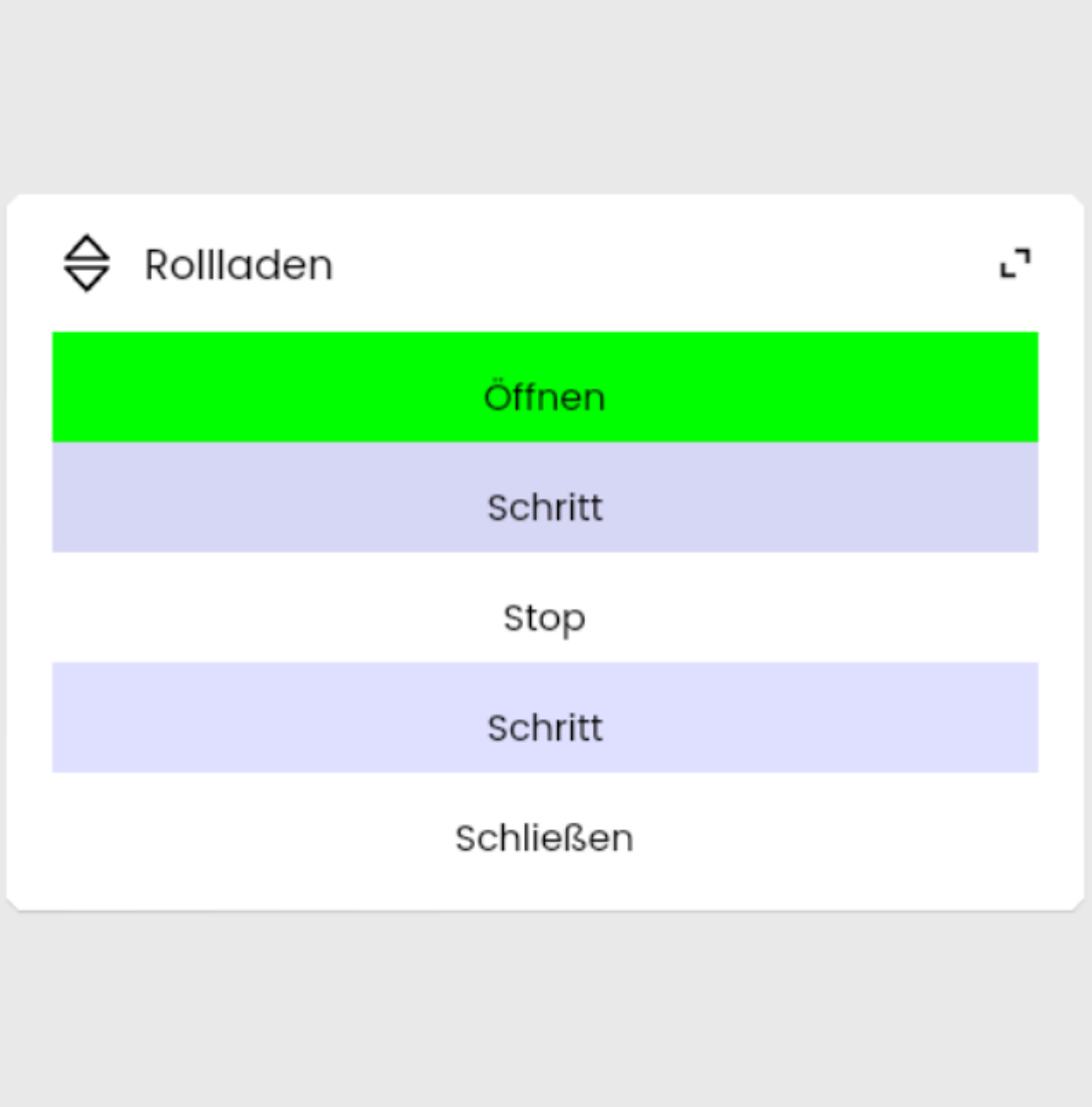
| | | | | |
|---|---|---|---|---|
| **WindSpeed** | **0%** | **30%** | **60%** | **100%** |
| **Window** | **0%** | **100%** | | |

Folgende Icons sind erst ab IP-Symcon 5.0/Symcon Mobile 5.0 verfügbar: Bath, Bike, Book, Cloudy, Doll, Door, Download, EnergyProduction, EnergySolar, EnergyStorage, Favorite, Fitness, FloorLamp, Gas, Handicap, Heart, Help, Link, Menu, Pants, Party, People, Presence, Raffstore, Remote, Sink, Sleet, Sunny, Teddy, Tee, Thunder, Umbrella

# Objekt-Darstellung

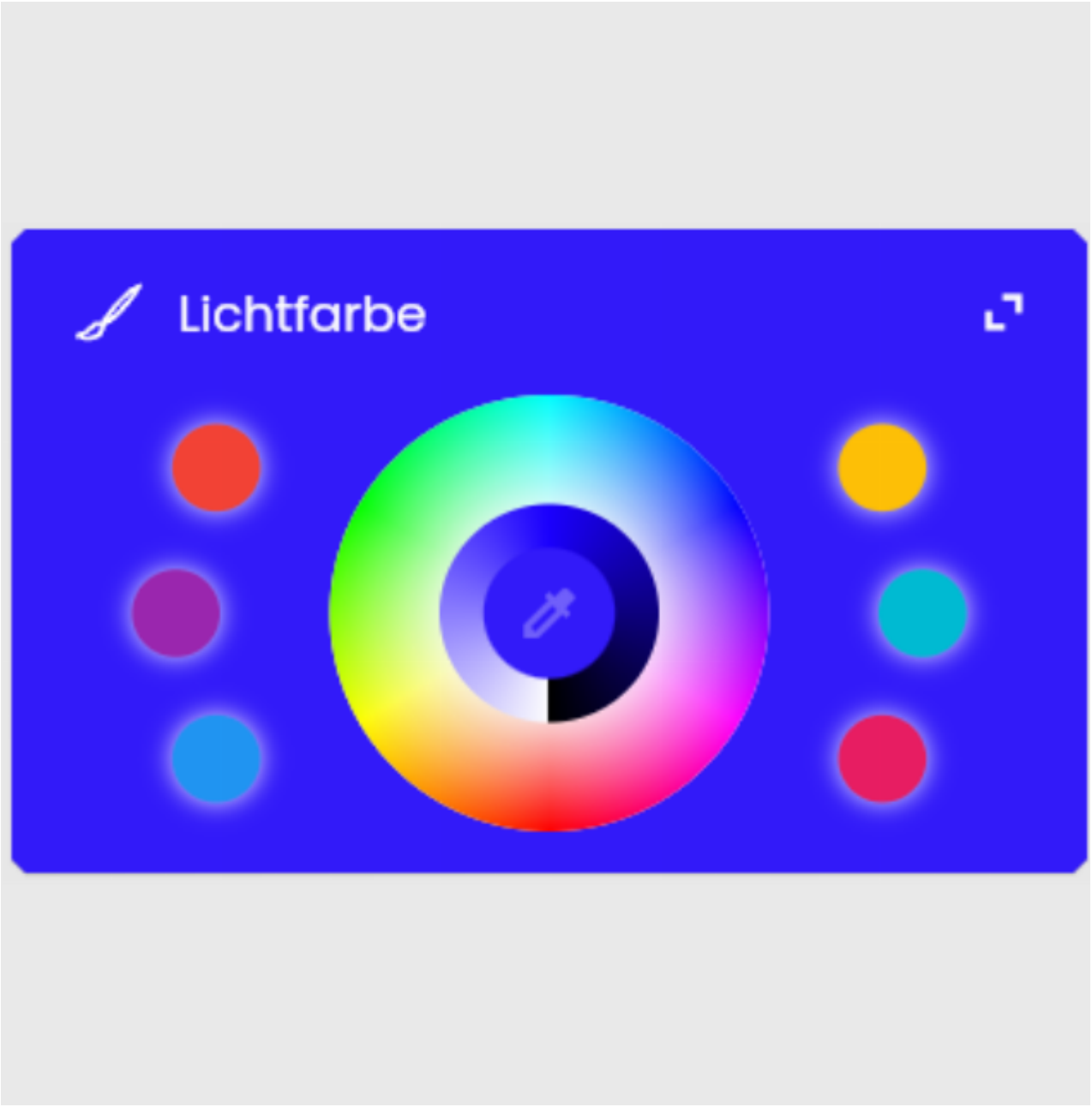
## AKTIVE ELEMENTE

Einzelne schaltbare Elemente können verschieden dargestellt werden.



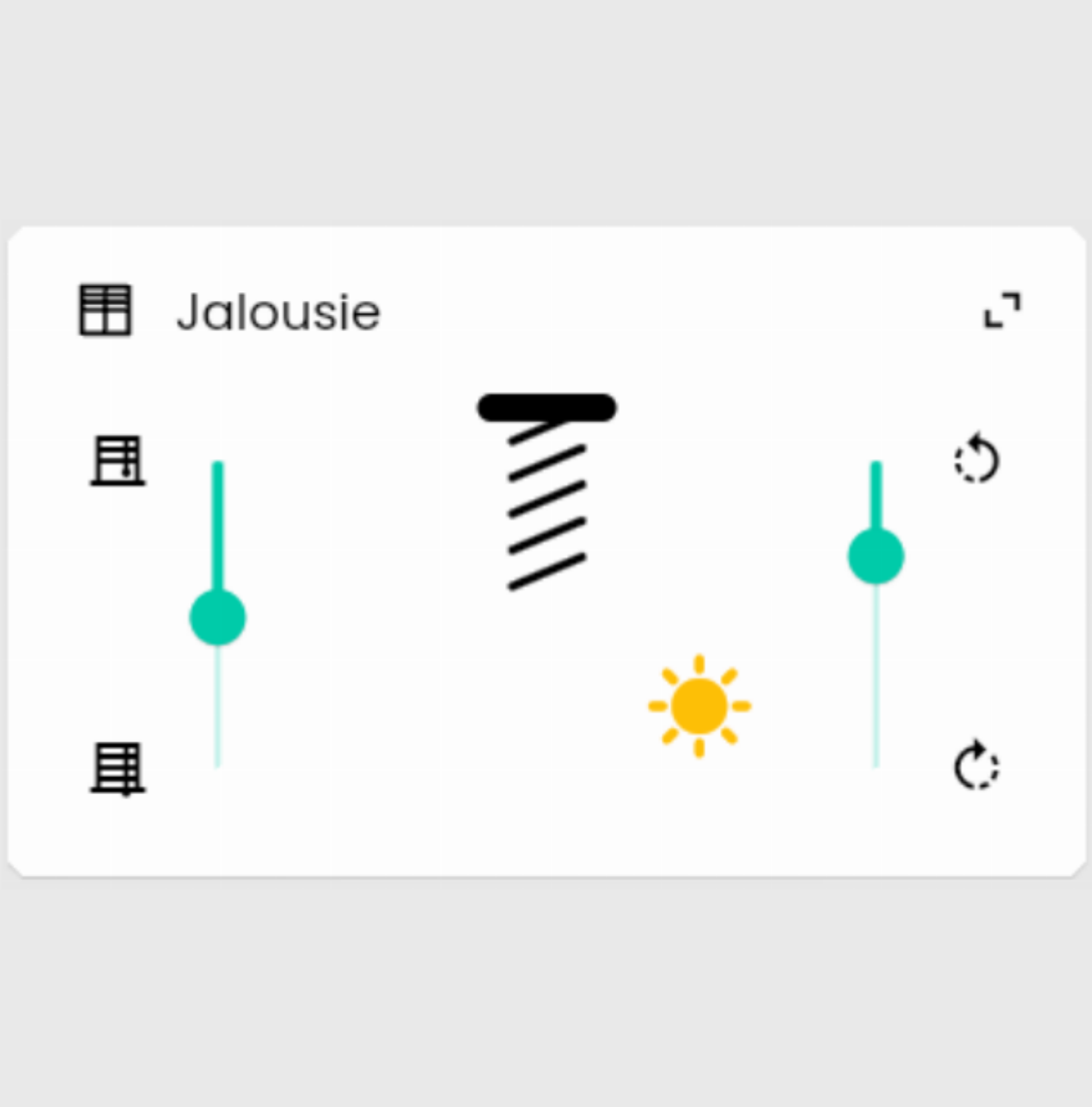

## AUFZÄHLUNG

Eine Aufzählung ist eine Liste von Werten, welche aktiviert werden können.



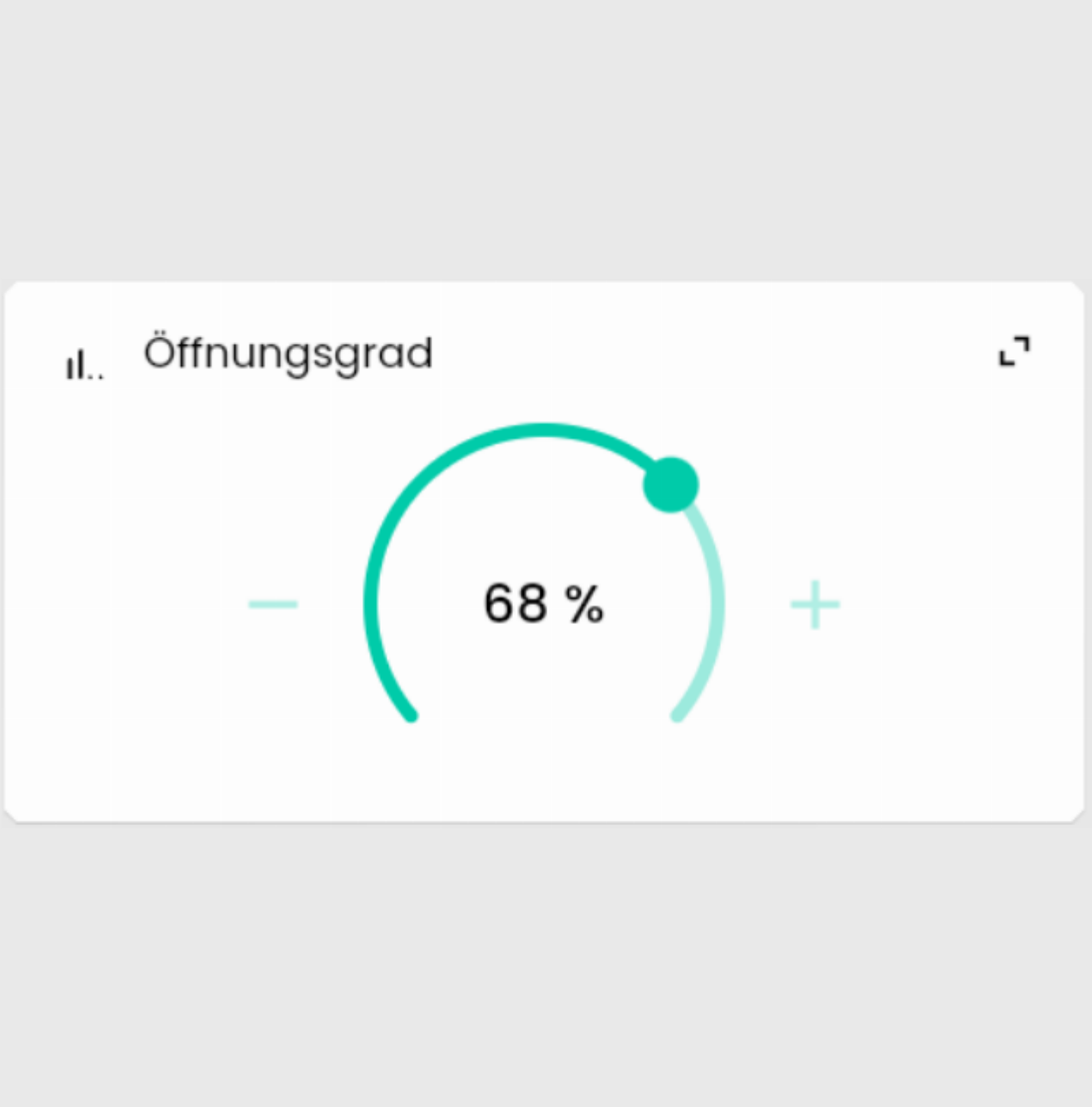

## FARBAUSWAHL

Mit einer Farbauswahl kann eine Farbe dargestellt und konfiguriert werden.



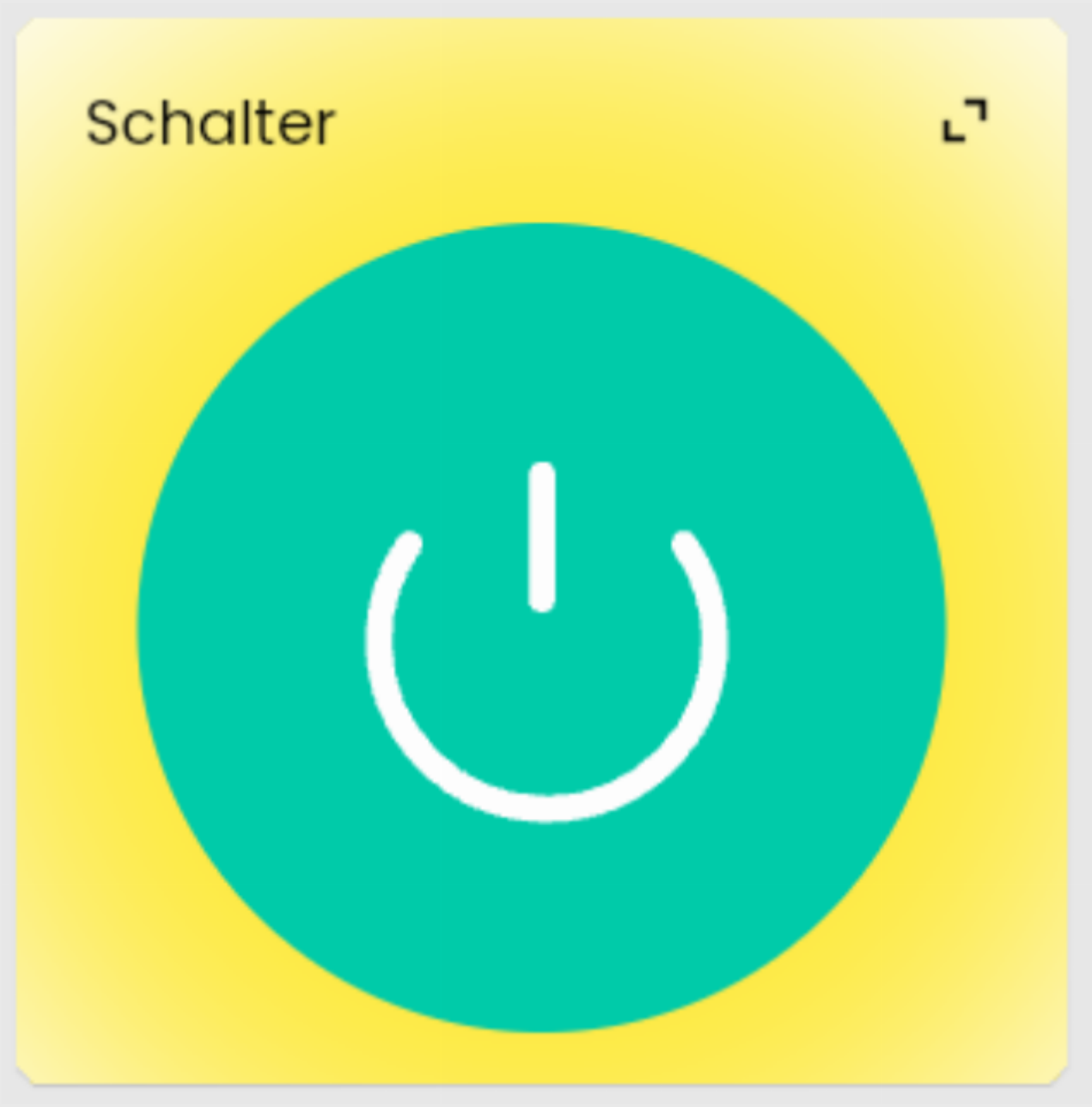

## ROLLLADEN

Für einen Rollladen kann der Öffnungs- oder Neigungsgrad der Lamellen eingestellt werden.




## SCHIEBEREGLER

Bei einem Schieberegler kann ein Wert eingestellt werden.



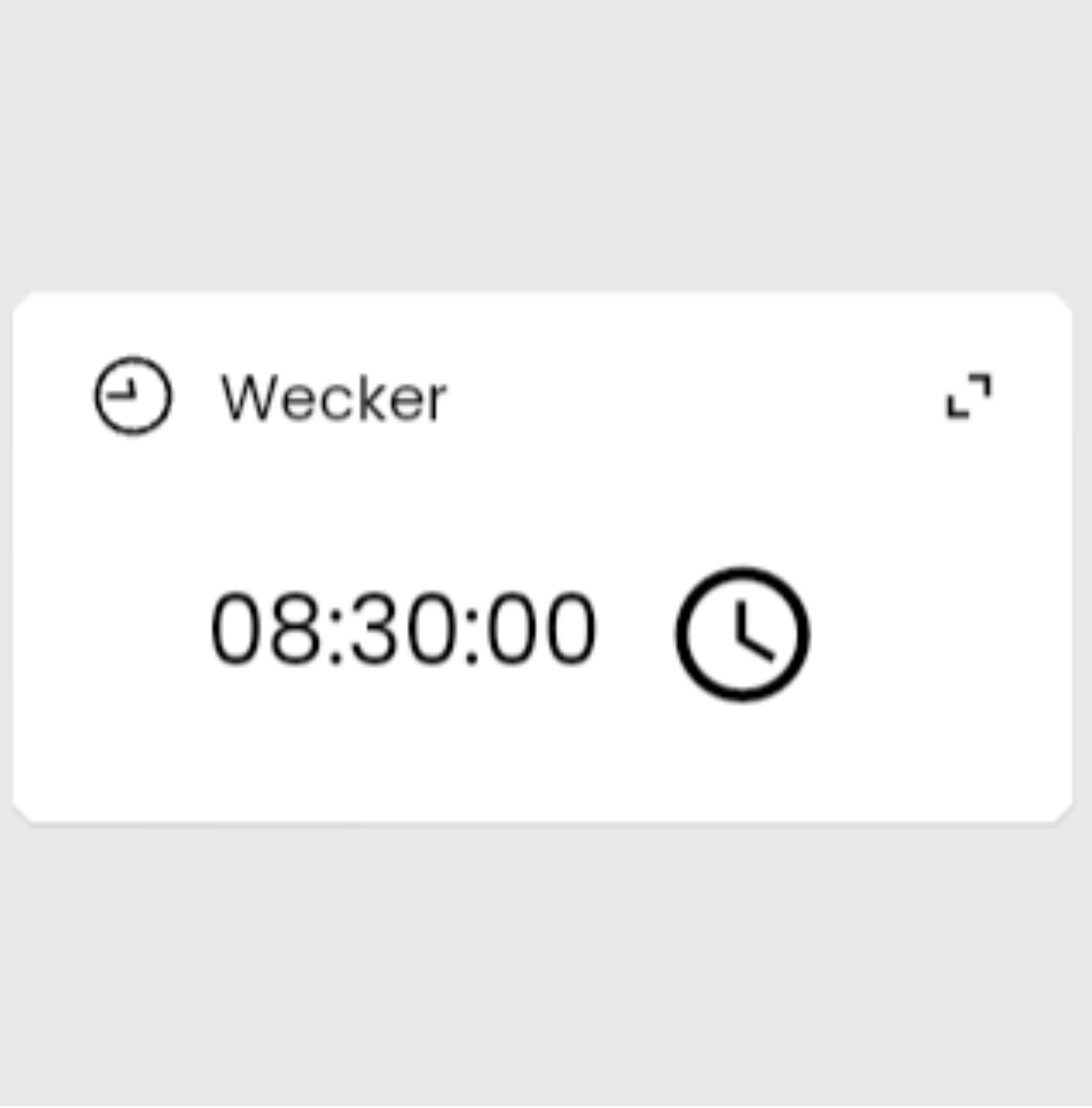

## SCHALTER

Ein Schalter kann An (true) oder Aus (false) geschaltet werden.



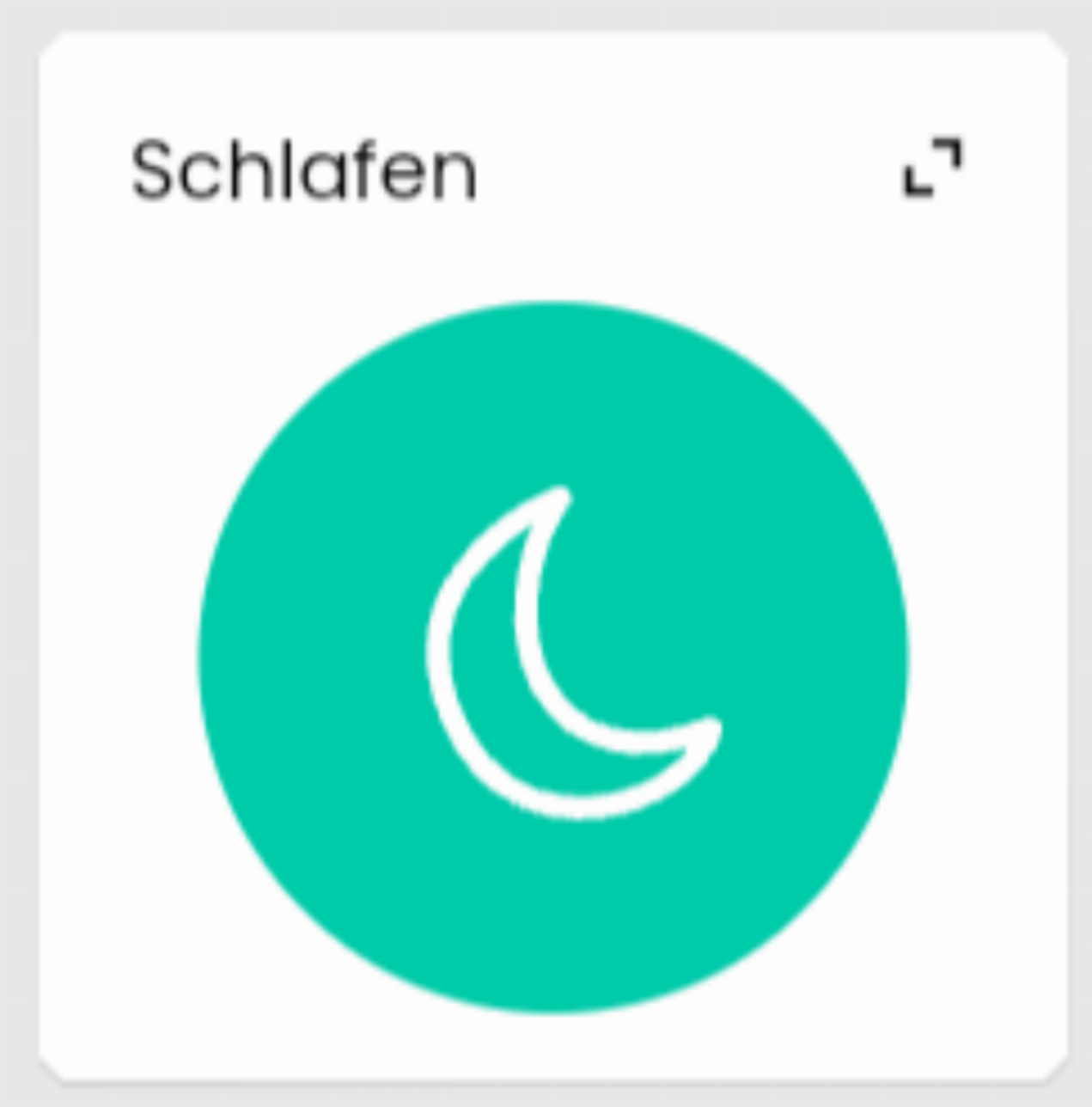

## WERTEINGABE

Eine Werteingabe ermöglicht das verändern von Werten als Freitext.




## DATUM/UHRZEIT

Ermöglicht das Darstellen und Anpassen eines spezifischen Zeitpunktes.

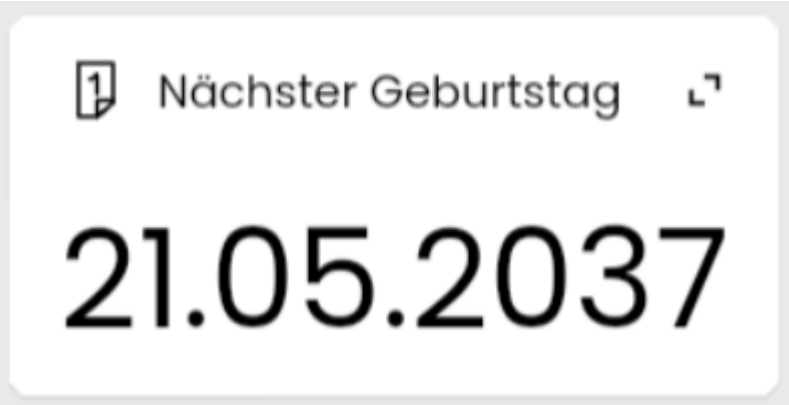



## AUTOMATION

Automationen können über einen Button ausgeführt werden.

## PASSIVE ELEMENTE

Einzelne Elemente, die nicht geschaltet werden können, haben verschiedene Darstellungen.



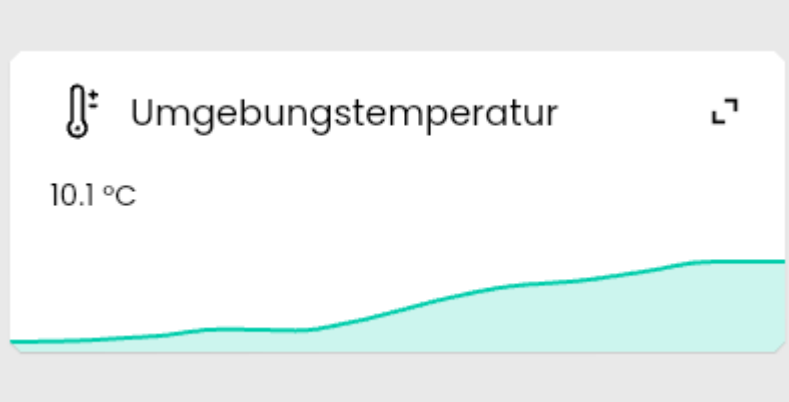

## DATUM/UHRZEIT

Ermöglicht es einen Zeitpunkt im Format Datum und/oder Uhrzeit darzustellen



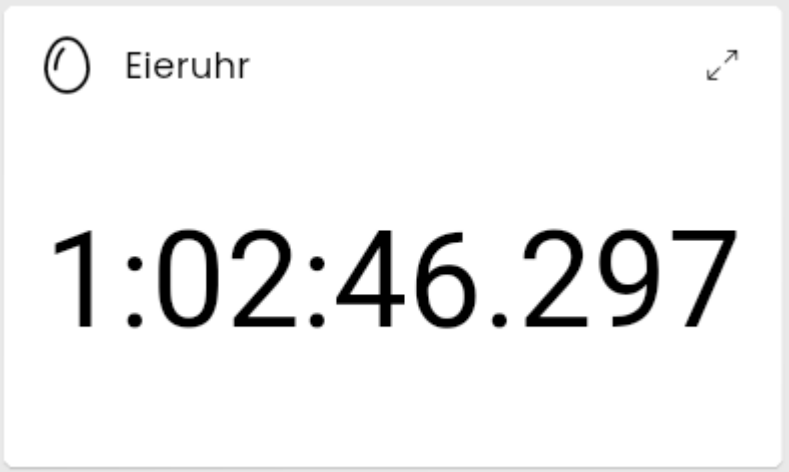

## FARBANZEIGE

Eine Farbanzeige stellt eine Farbe dar.




## WERTANZEIGE

Zeigt einen Wert optional mit Hintergrundfarbe oder Werteverlauf an.

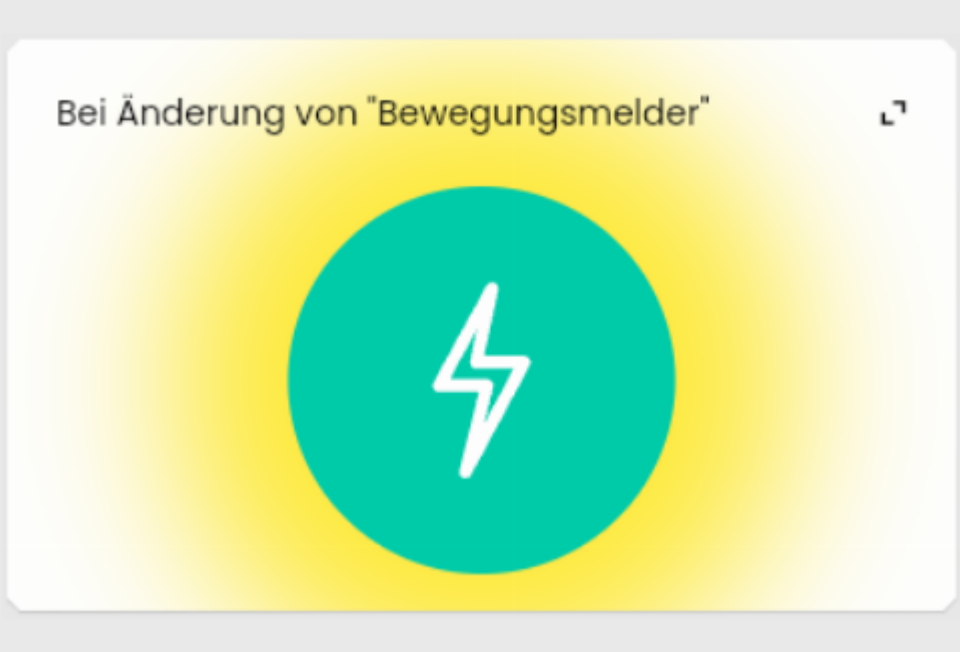



## DAUER

Zeigt die verbleibende Zeit bis zu einem Zeitpunkt an.

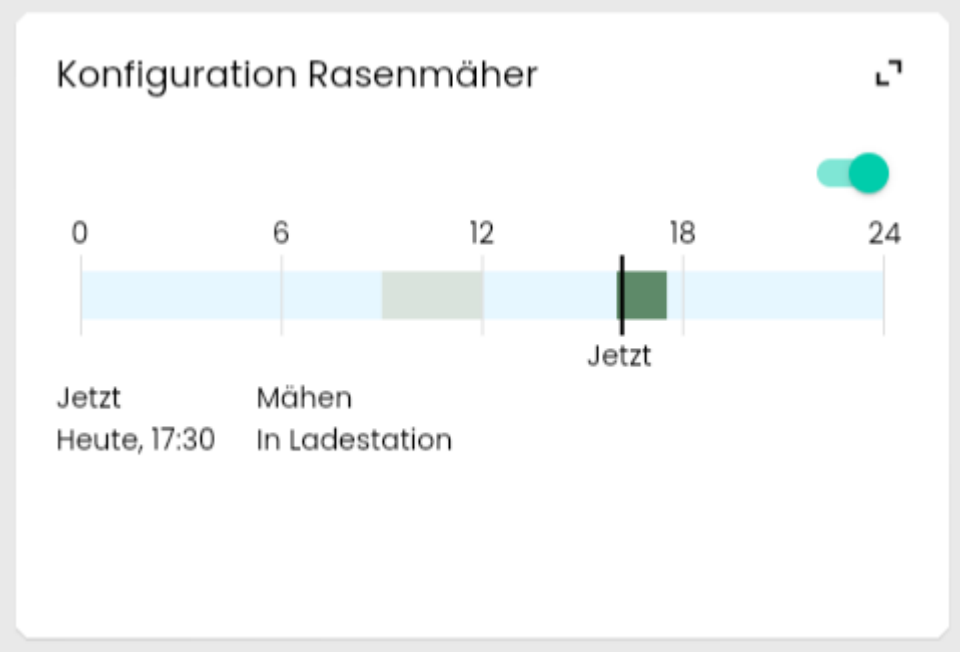
## EREIGNISSE

Erlaubt das Ausführen von Logik basierend auf Uhrzeit und Datum oder Veränderungen anderer Objekte.



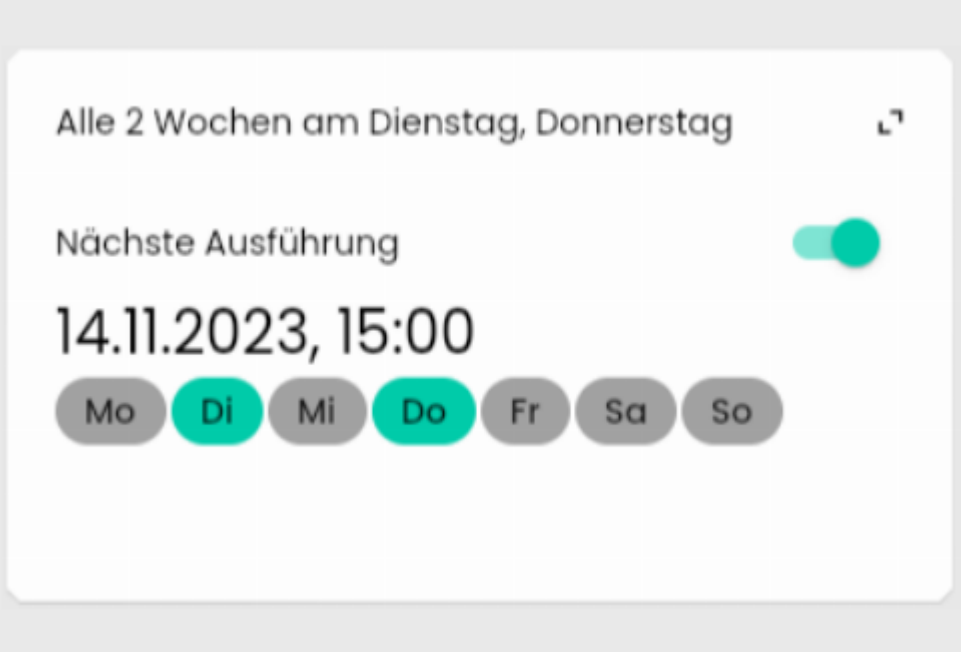

## AUSGELÖSTES EREIGNIS

Ein ausgelöstes Ereignis kann in der Visualisierung aktiviert bzw. deaktiviert werden.




## WOCHENPLAN

Ein Wochenplan kann in der Visualisierung betrachtet und konfiguriert werden.




## ZYKLISCHES EREIGNIS

Ein zyklisches Ereignis kann in der Visualisierung betrachtet und konfiguriert werden.

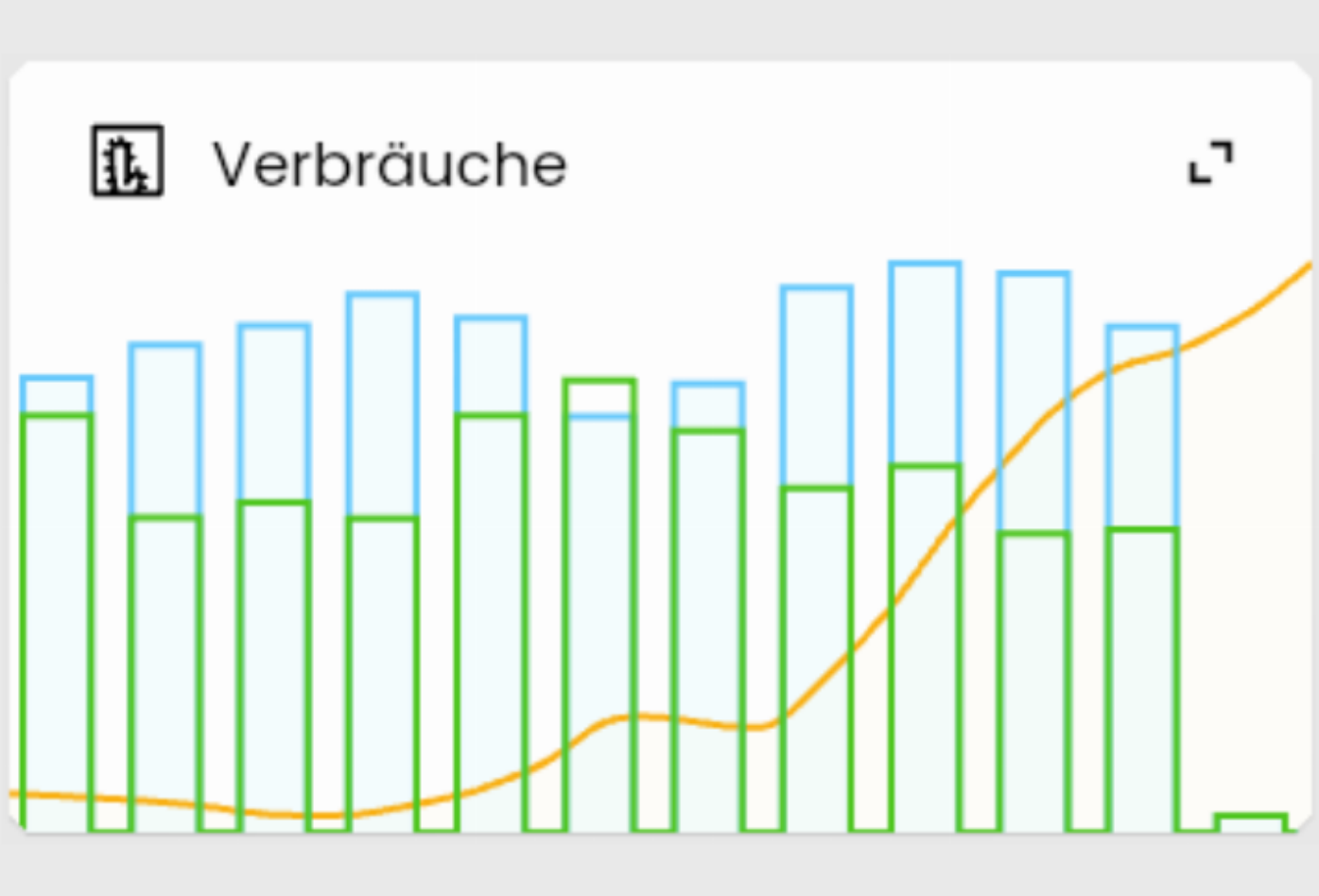
## EXTERNE ELEMENTE

Externe Dateien wie etwa **Medien** können in der Visualisierung angezeigt werden.




## BILD

In der Visualisierung können beliebige Bilder eingebunden werden.




## DIAGRAMM

In einem Diagramm können die Verläufe mehrere Variablen dargestellt werden.




## DOKUMENT

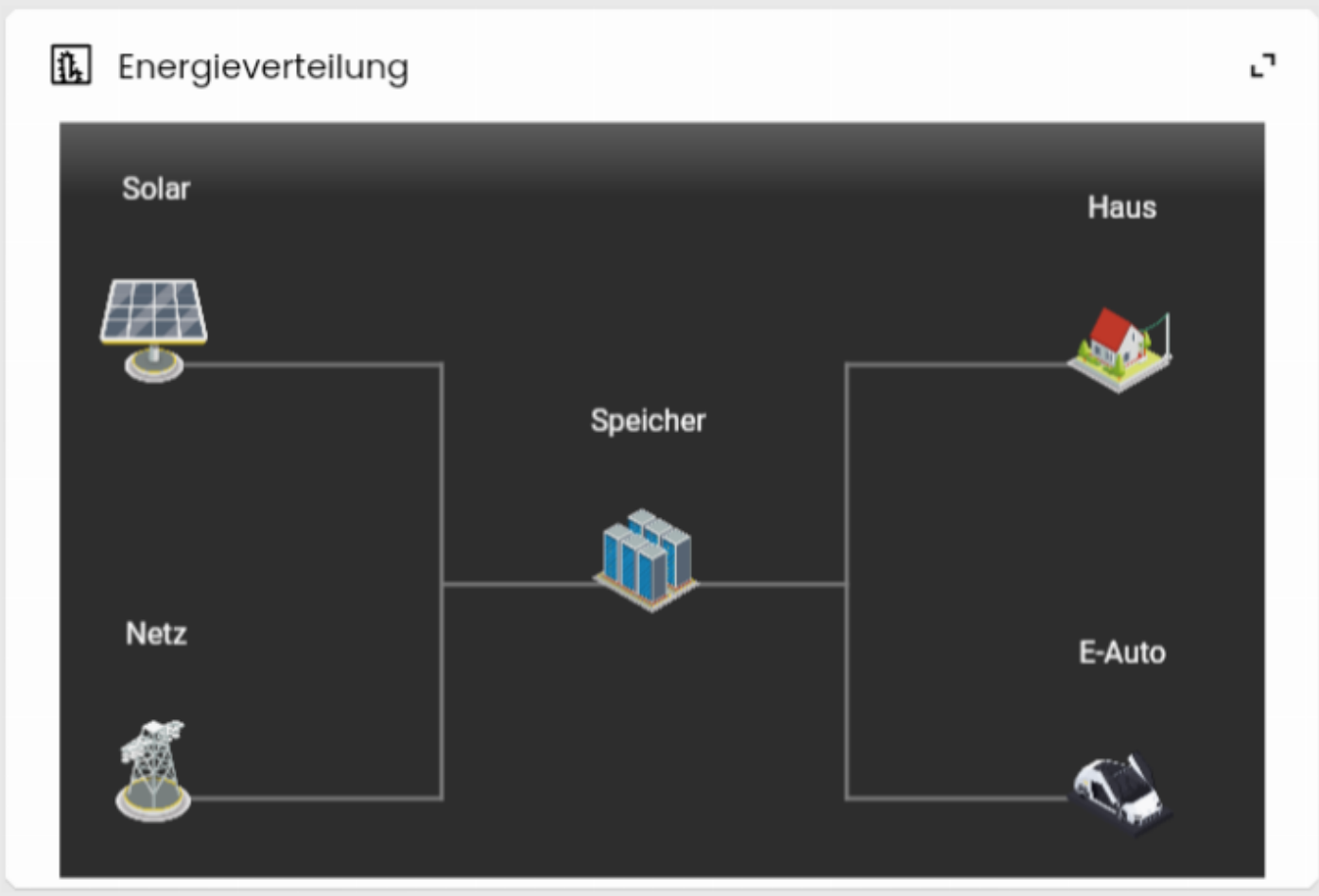
Ein Dokument kann per Auswahl des Buttons heruntergeladen werden.

Grundlagen-Tutorial




## WEBINHALT

Der Webinhalt bietet Experten die Möglichkeit beliebige HTML-Inhalte darzustellen und Webseiten einzubinden.



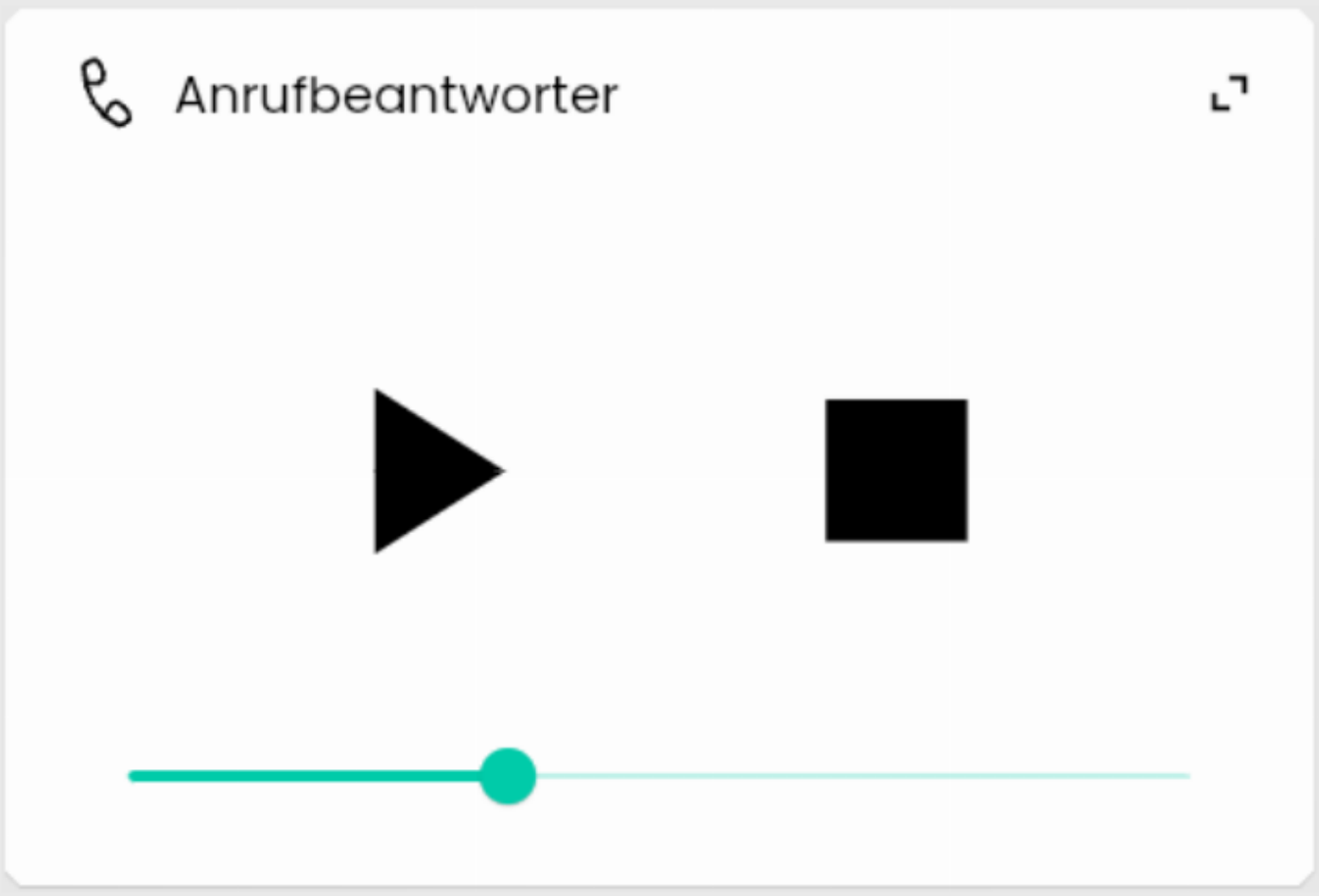

## IPSVIEW

IPSViews lassen sich nahtlos in die Visualisierung einbinden.




## STREAM

Streams stellen die Videoausgabe, etwa einer Überwachungskamera, dar.




## TON

In der Visualisierung können Audiodateien eingebunden und wiedergegeben werden.

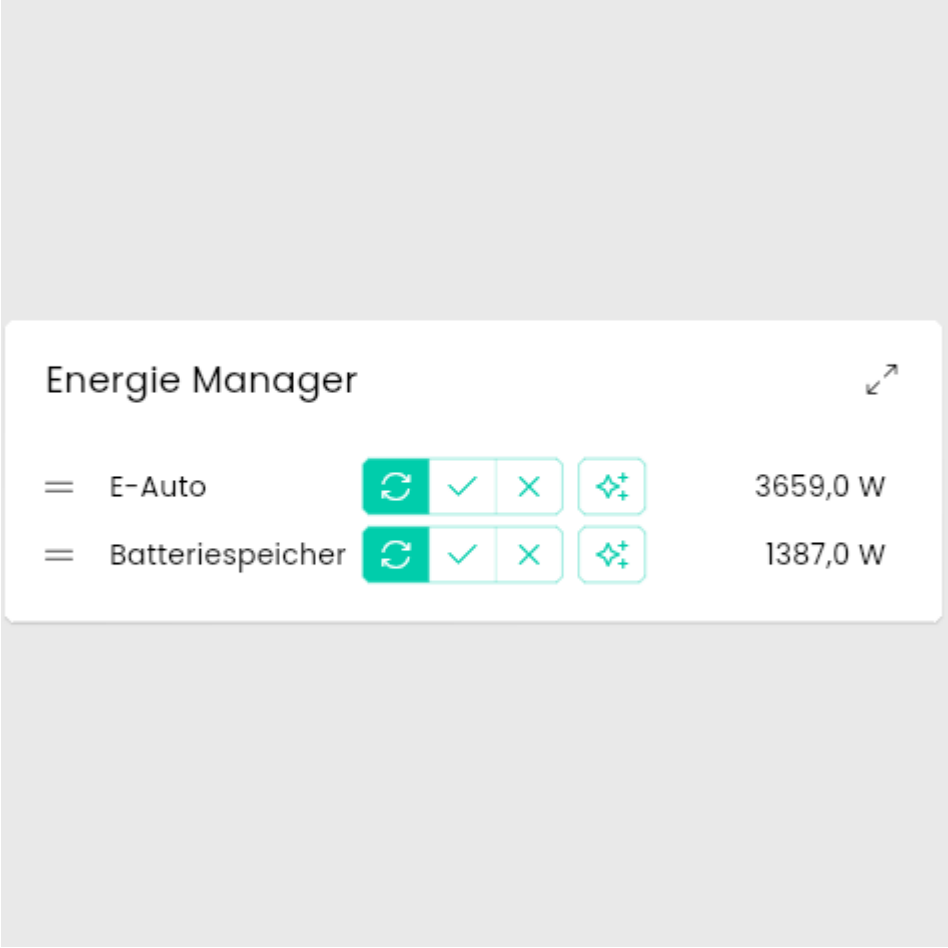
## ZUSAMMENGEFASSTE DARSTELLUNGEN

Mehrere Elemente können intelligent in einer Darstellung zusammengefasst werden.



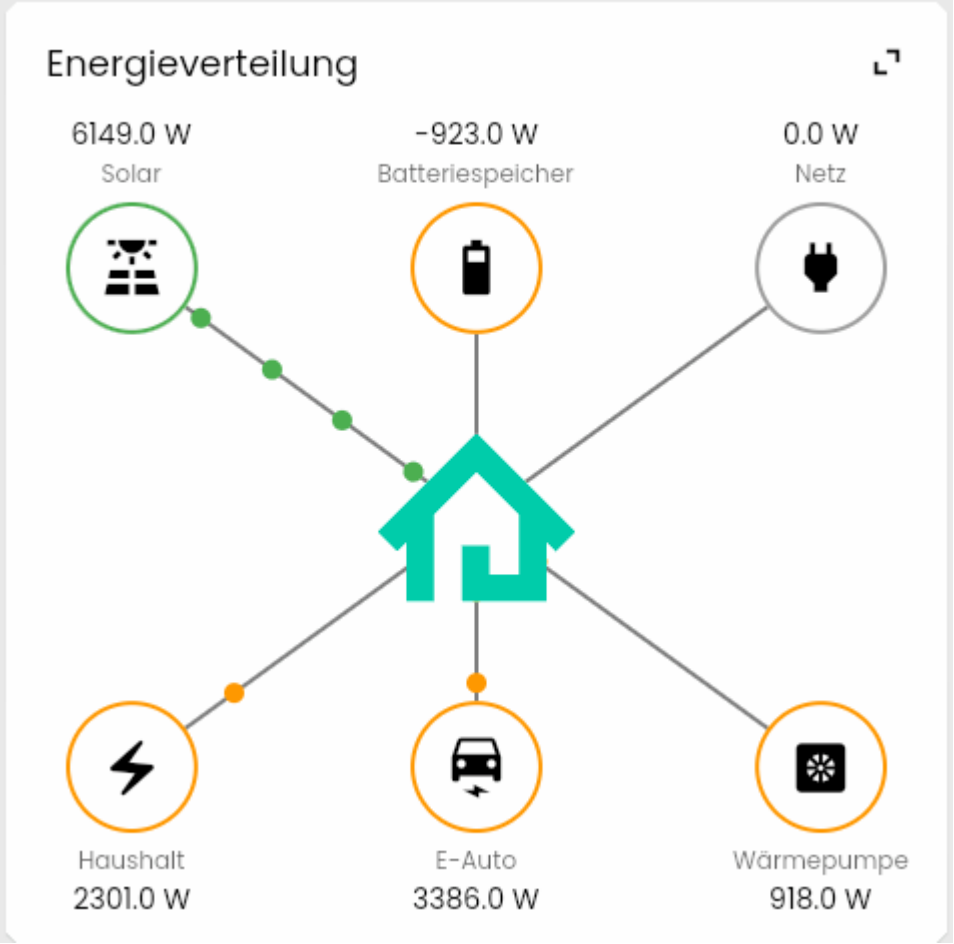

## EINZELNES ELEMENT

Ermöglicht die Darstellung eines Elements aus einer Liste als einzelne Kachel.



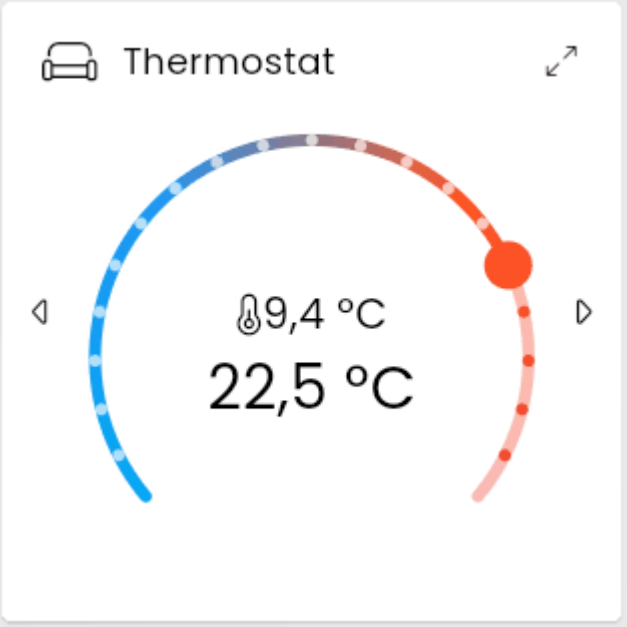

## ENERGIE MANAGER

Der Energie Manager schaltet intelligent Geräte, wenn genug Energie zur Verfügung steht.



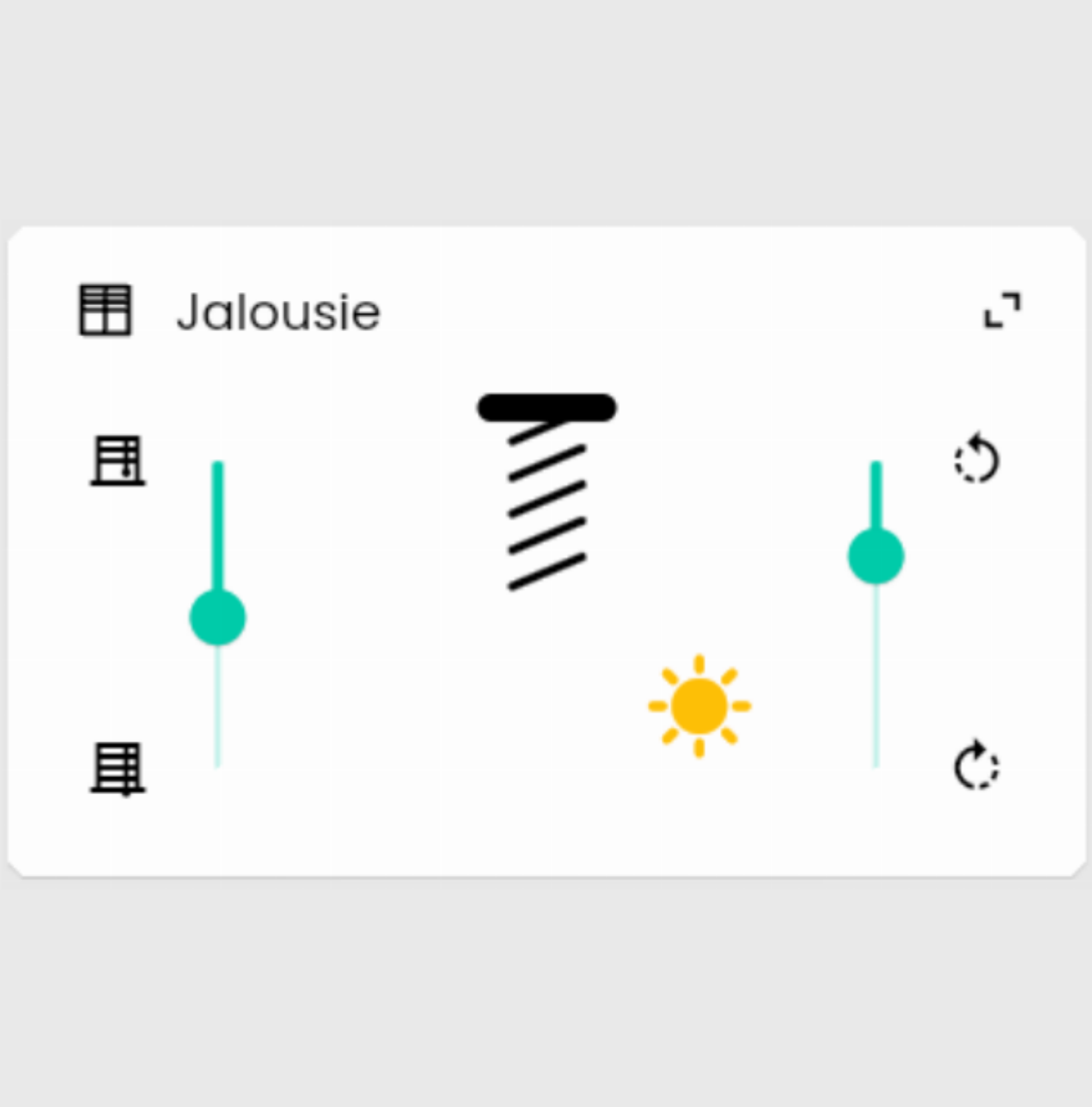

## ENERGIEVERTEILUNG

Die Energieverteilung stellt den Fluss von erzeugter und verbrauchter Energie dar.



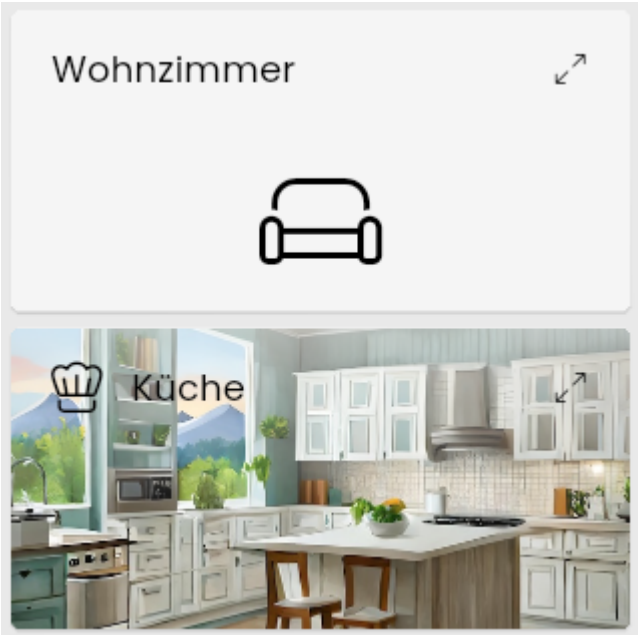

## INHALTSWECHSLER

Es kann zwischen mehreren Elementen in einer Kachel gewechselt werden.




## JALOUSIE

Für eine Jalousie kann der Öffnungsgrad sowie der Neigungsgrad der Lamellen eingestellt werden.



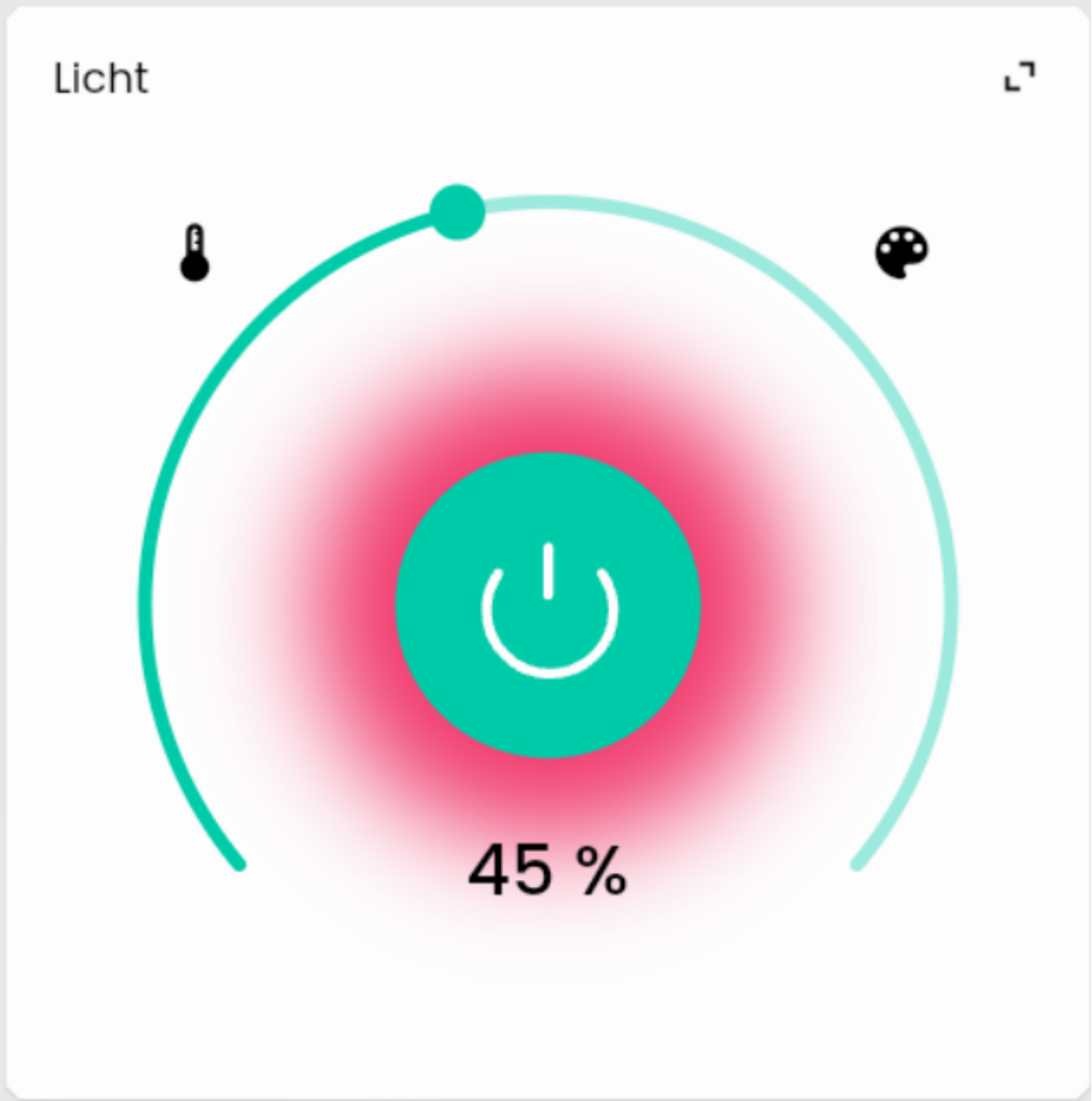

## KATEGORIE

Kategorien können geöffnet und die darin enthaltenen Objekte betrachtet werden.



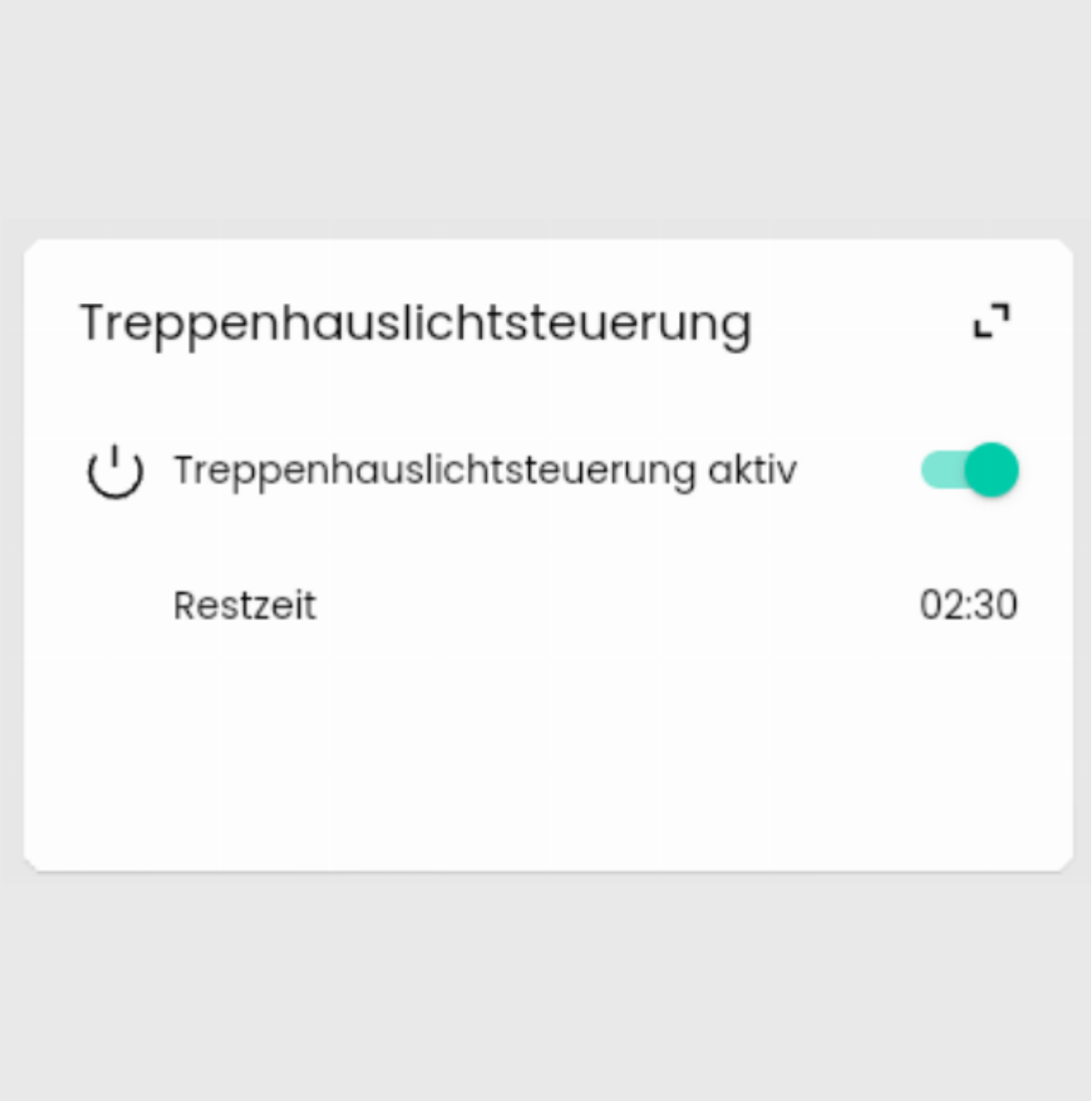

## KATEGORIELEISTE

Stellt Kategorien in einer Leiste am oberen Bildschirmrand dar.




## LICHT

In der Darstellung Licht werden verschiedene Werte einer Lampe gemeinsam visualisiert.



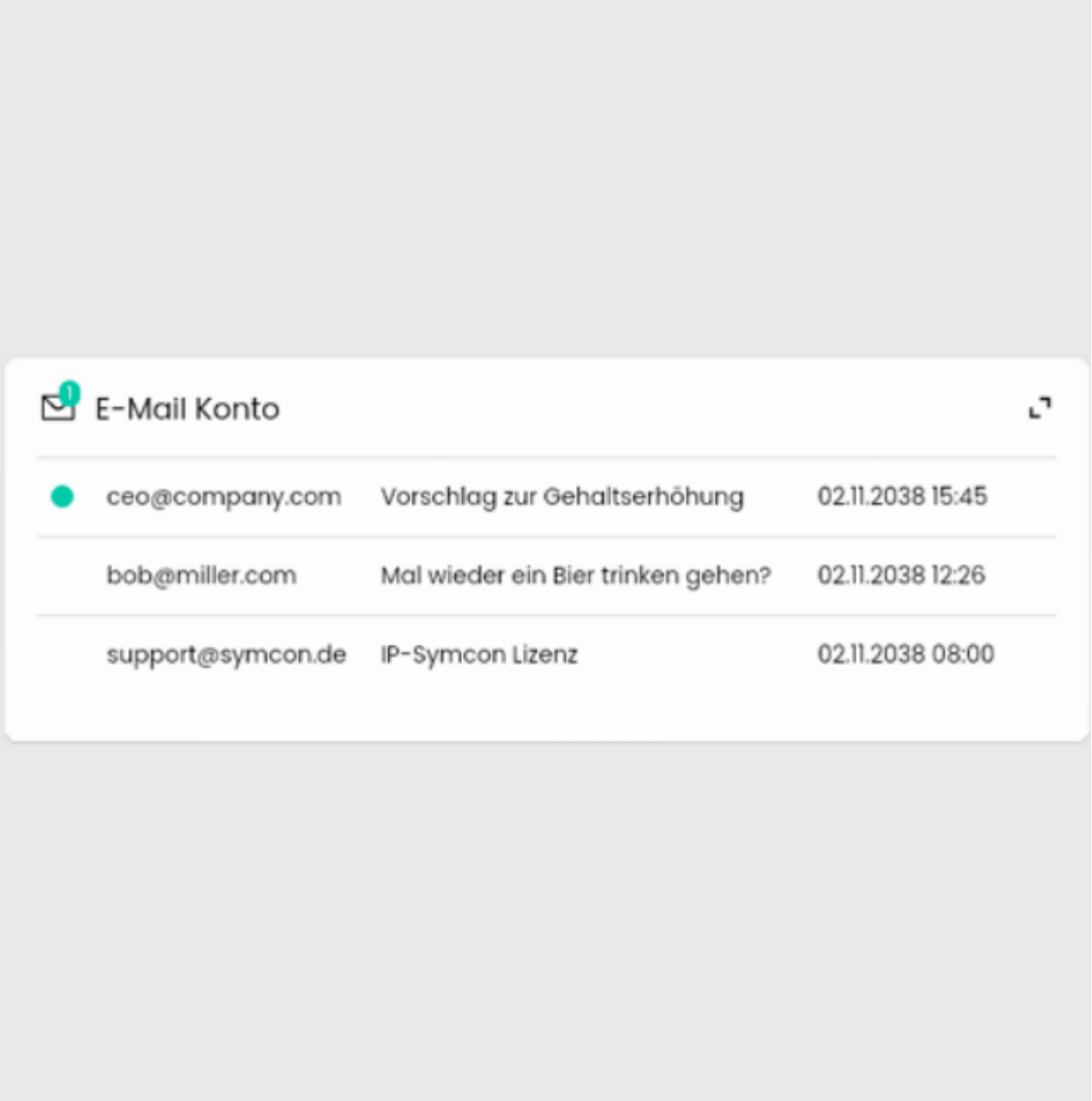

## LISTE

Eine Listendarstellung kann beliebige Objekte untereinander darstellen.



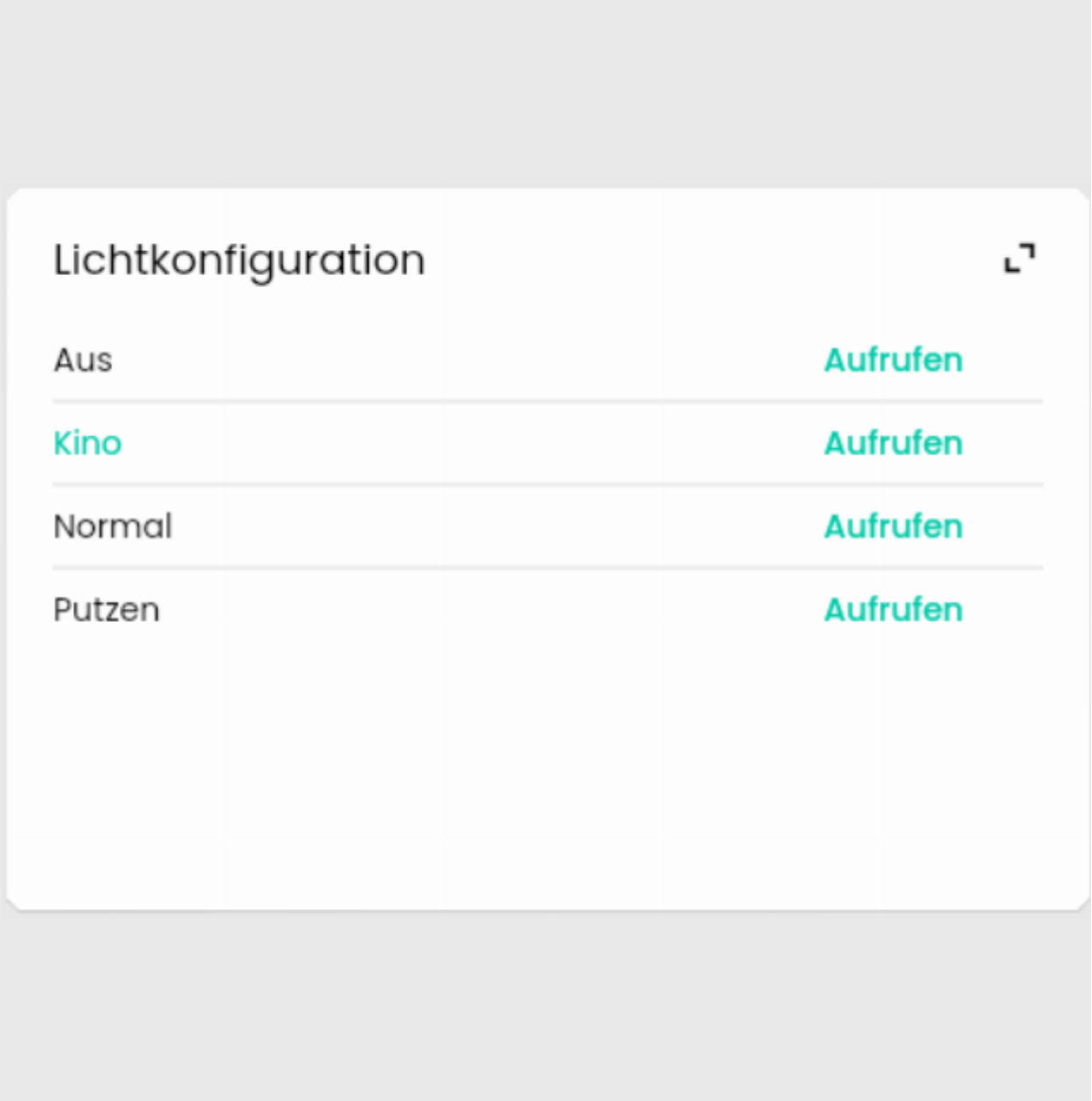

## MEDIA PLAYER

Ein Media Player stellt Parameter zur Wiedergabe von Medien dar und ermöglicht das Schalten.



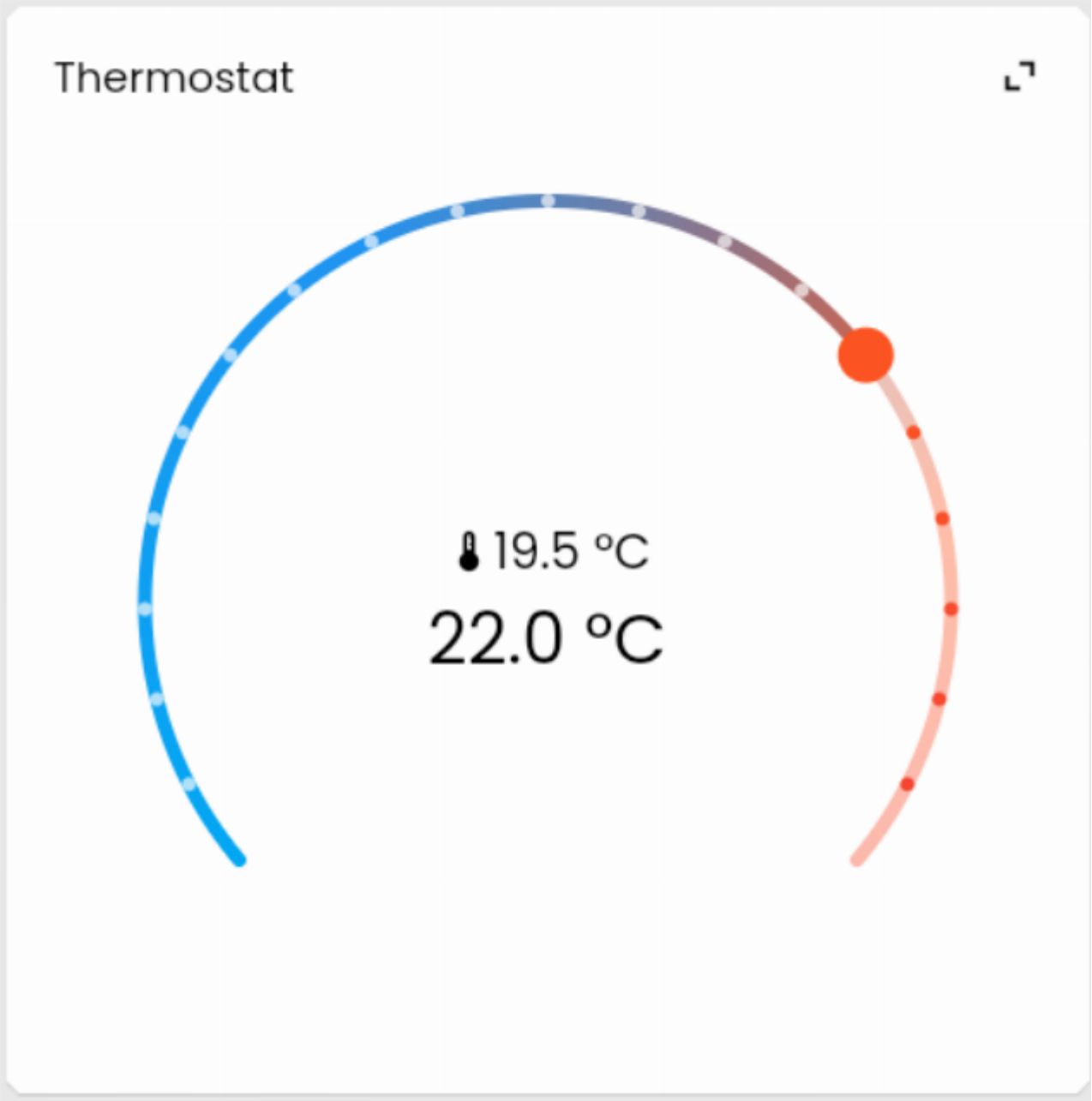

## MITTEILUNGEN

Die Mitteilung Darstellung visualisiert aktuelle und ungelesene Nachrichten.




## SZENEN

Geräteeinstellungen können als Szene gespeichert und aufgerufen werden.

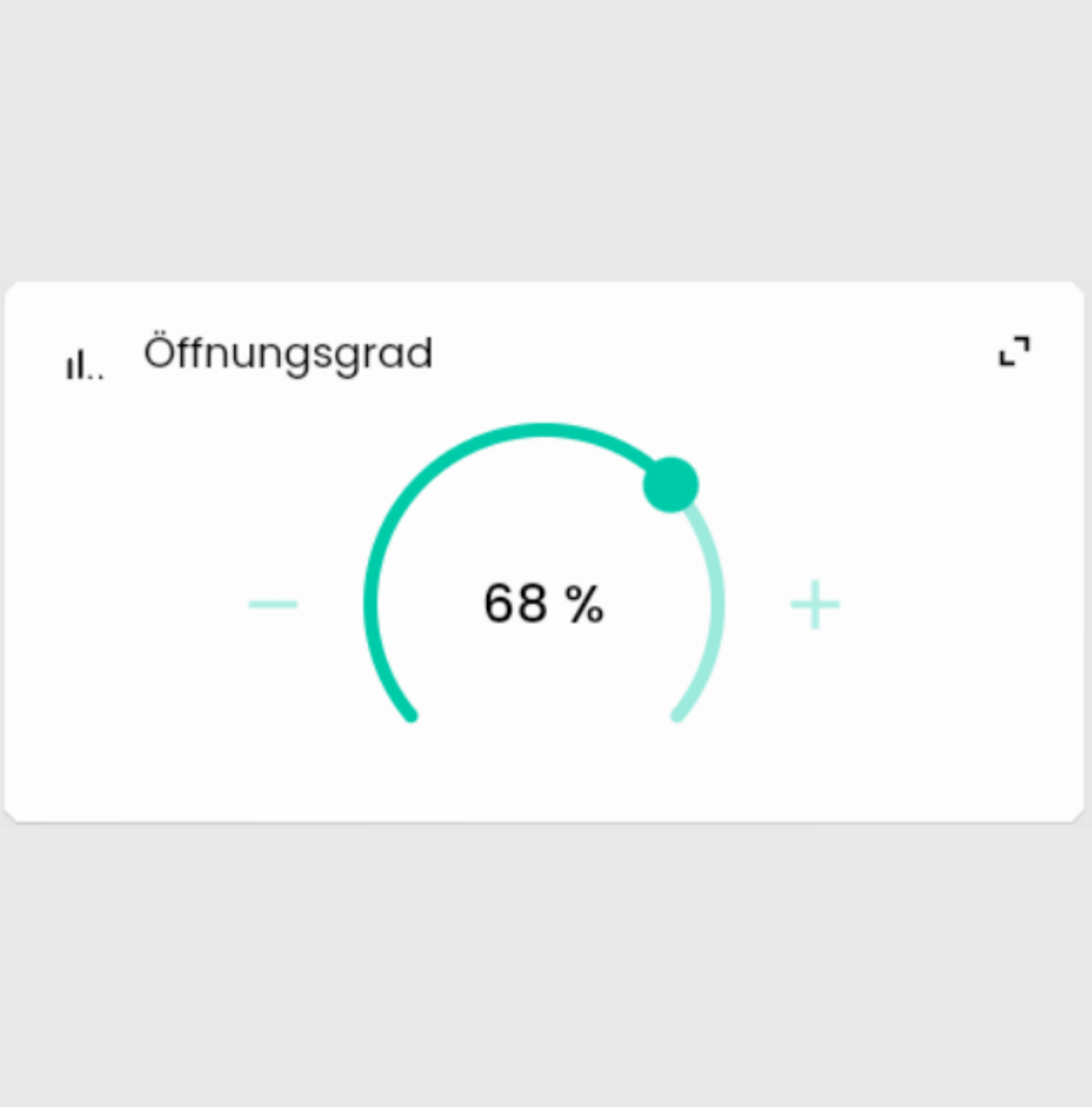



## THERMOSTAT

Ein Thermostat emöglicht das Einstellen der Soll-Temperatur und zeigt die Ist-Temperatur an.

# LEGACY-DARSTELLUNGEN

Dies sind veraltete Darstellungen, welche auf ihre tatsächlichen Darstellungen weiterleiten.




## LEGACY-PROFIL

Variablenprofile können zur Darstellung verwendet werden.

# Aufzählung

Eine Aufzählung ist eine Liste von Werten, welche ausgewählt werden können.

## VORAUSSETZUNGEN

Eine Aufzählung ist eine Variable mit der Darstellung Aufzählung. Um diese Darstellung zu verwenden muss die Variable folgende Bedingungen erfüllen:
Eingerichtete Variablenaktion

### PARAMETER

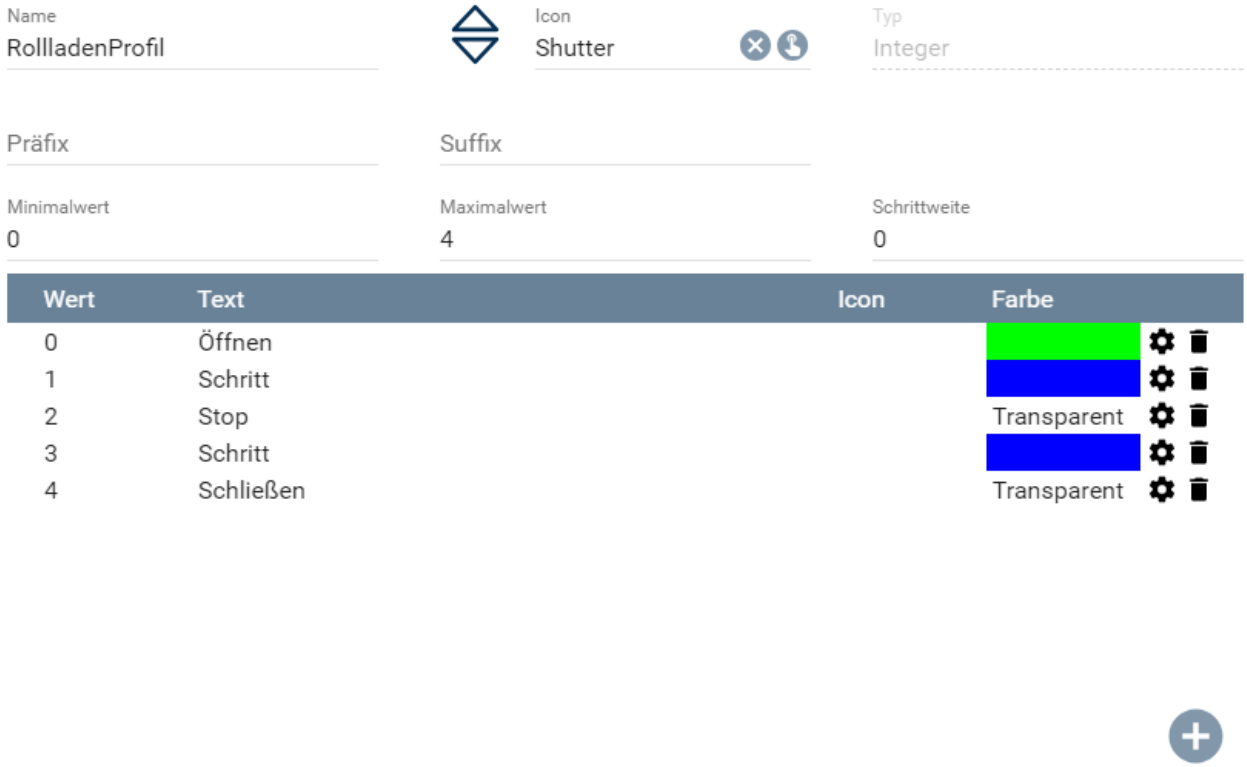
| PARAMETER | BESCHREIBUNG |
|---|---|
| Standard Icon | Das Icon, welches für die Variable verwendet wird |
| Optionen | Die einzelnen Optionen der Aufzählung |
| Anordnung | Die Anordnung der Optionen in einer Kachel. Mögliche Anordnungen sind Spalte, Reihe und Gitter. Gitter ist nicht für Variablen vom Typ Boolean verfügbar. Dieser Parameter wird nur bei der Darstellung als eigene Kachel ausgewertet, nicht bei der Anzeige in einer Liste. |

Optionen

| PARAMETER | BESCHREIBUNG |
|---|---|
| Wert | Variablenwert für den diese Option gilt |
| Beschriftung | Text, der für diese Option angezeigt wird |
| Icon überschreiben | Falls gesetzt, wird das Standardicon dieser Darstellung überschrieben, falls die Variable den Wert dieser Option hat |
| Icon | Icon für die Variable, falls die Variable den Wert dieser Option hat (nur verfügbar wenn "Icon überschreiben" gesetzt) |
| Farbe | Farbe mit welcher die Option in der Aufzählung dargestellt wird |

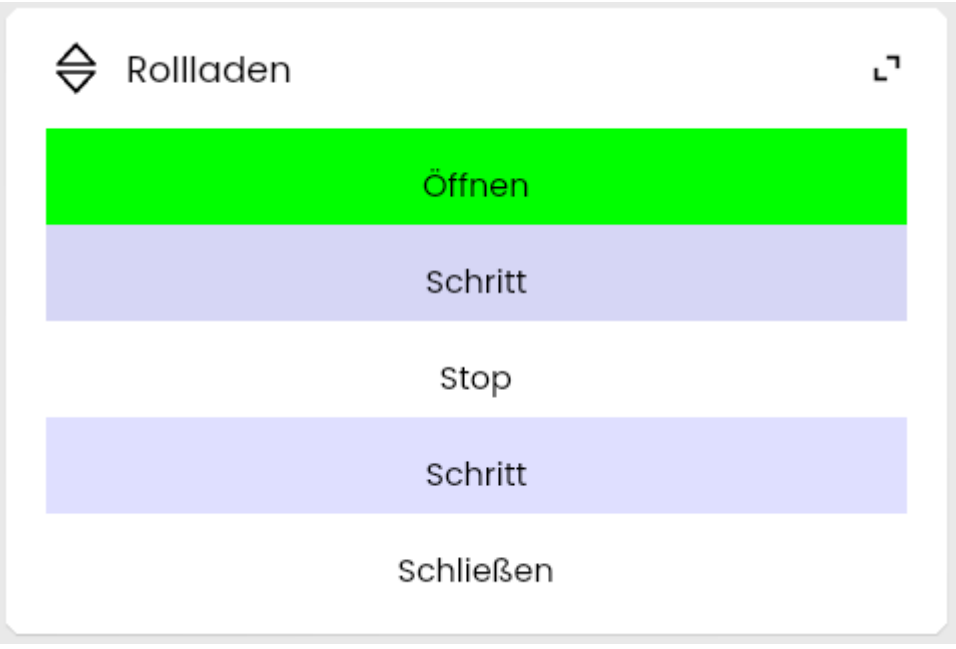
## DARSTELLUNG ALS LEGACY-PROFIL

Verwendet eine Variable die Darstellung Legacy Profil, so wird diese als Aufzählung darstellt, falls sie zusätzlich folgende Bedingungen erfüllt:
Variablenprofil mit folgenden Parametern:
Schrittweite auf 0 (nur bei Integer/Float, sonst keine Option für Schrittweite)
mind. 1 gesetzte Assoziation
Minimalwert und Maximalwert irrelevant




## AUSSEHEN IN KACHEL VISUALISIERUNG

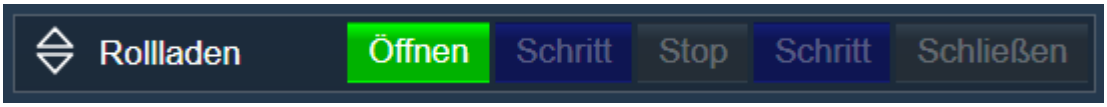
### ALS EIGENE KACHEL




### INNERHALB EINER LISTE



## AUSSEHEN IM WEBFRONT



# Ausgelöstes Ereignis

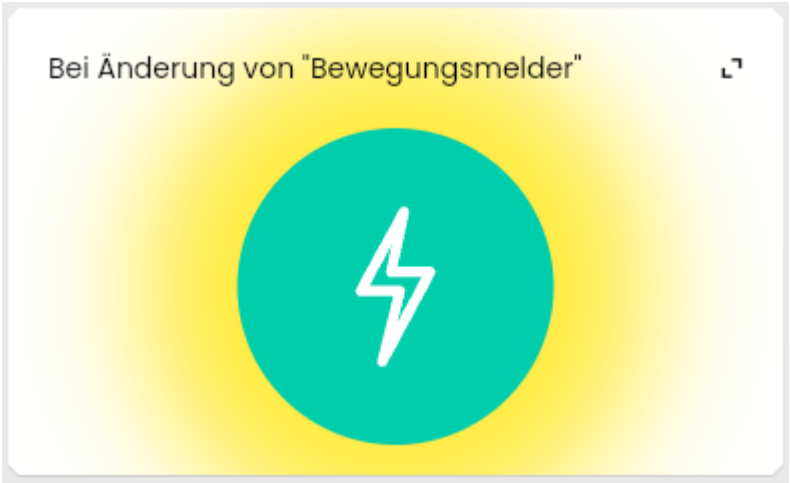
Benötigt: IP-Symcon >= 7.0
Ein ausgelöstes Ereignis kann in der Visualisierung aktiviert bzw. deaktiviert werden.

### VORAUSSETZUNGEN

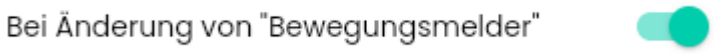
Ein ausgelöstes Ereignis ist ein Ereignis vom Typ Ausgelöst.

## AUSSEHEN IN KACHEL VISUALISIERUNG

### ALS EIGENE KACHEL




### INNERHALB EINER LISTE




## AUSSEHEN IM WEBFRONT

Ausgelöste Ereignisse werden im WebFront nicht dargestellt.

# Automation

Automationen können über einen Button ausgeführt werden.

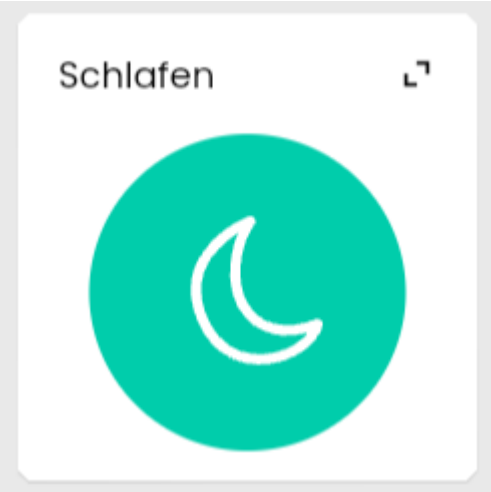
### VORAUSSETZUNGEN

Eine eingerichtete Automation kann in der Visualisierung ausgeführt werden

> In der Darstellung als eigene Kachel richtet sich das Icon des Buttons nach dem Icon der Automation

## AUSSEHEN IN KACHEL VISUALISIERUNG

### ALS EIGENE KACHEL




### INNERHALB EINER LISTE




## AUSSEHEN IM WEBFRONT



# Bild

In der Visualisierung können beliebige Bilder eingebunden werden.

### VORAUSSETZUNGEN

Ein Bild ist ein Medium vom Typ Bild.

## AUSSEHEN IN KACHEL VISUALISIERUNG

### ALS EIGENE KACHEL




### INNERHALB EINER LISTE




## AUSSEHEN IM WEBFRONT

Garten



# Datum/Uhrzeit

Benötigt: IP-Symcon >= 4.1
Mit Datum/Uhrzeit kann Datum und/oder Uhrzeit von einem spezifischen Zeitpunkt dargestellt und ggfs. angepasst werden.

### VORAUSSETZUNGEN

Datum/Uhrzeit ist eine Variable mit der Darstellung Datum/Uhrzeit. Um diese Darstellung zu verwenden muss die Variable folgende Bedingungen erfüllen:
Typ Integer

### PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| Datumsanzeige | Bestimmt wie das Datum formatiert wird. Möglich sind: "Keine", "Jahr, Monat und Tag", "Monat und Tag", und "Jahr und Monat" |
| Formatiere Monat als | Wenn im Datum der Monat angezeigt werden soll, kann dessen Formatierung angepasst werden. Möglich sind: "Text" und "Zahl" |
| Zeige Wochentag | Legt fest ob der Wochentag im Format mit angezeigt wird |
| Zeitanzeige | Bestimmt wie die Zeit formatiert werden soll. Möglich sind: "Keine", "Stunden und Minuten" und "Stunden, Minuten und Sekunden" |

## DARSTELLUNG ALS LEGACY-PROFIL

Verwendet eine Variable die Darstellung Legacy Profil, so wird diese als Datum/Uhrzeit darstellt, falls sie eines der folgenden Variablenprofile ausgewählt ist:
~UnixTimestamp
~UnixTimestampDate
~UnixTimestampTime
Die unterschiedlichen Profile entsprechen festen Parameter Kombinationen.

### ~UNIXTIMESTAMP

Datumsanzeige: Jahr, Monat und Tag
Formatiere Monat als: Text
Zeige Wochentag: Nein
Zeitanzeige: Stunden und Minuten

### ~UNIXTIMESTAMPDATE

Datumsanzeige: Jahr, Monat und Tag
Formatiere Monat als: Text
Zeige Wochentag: Nein
Zeitanzeige: Keine

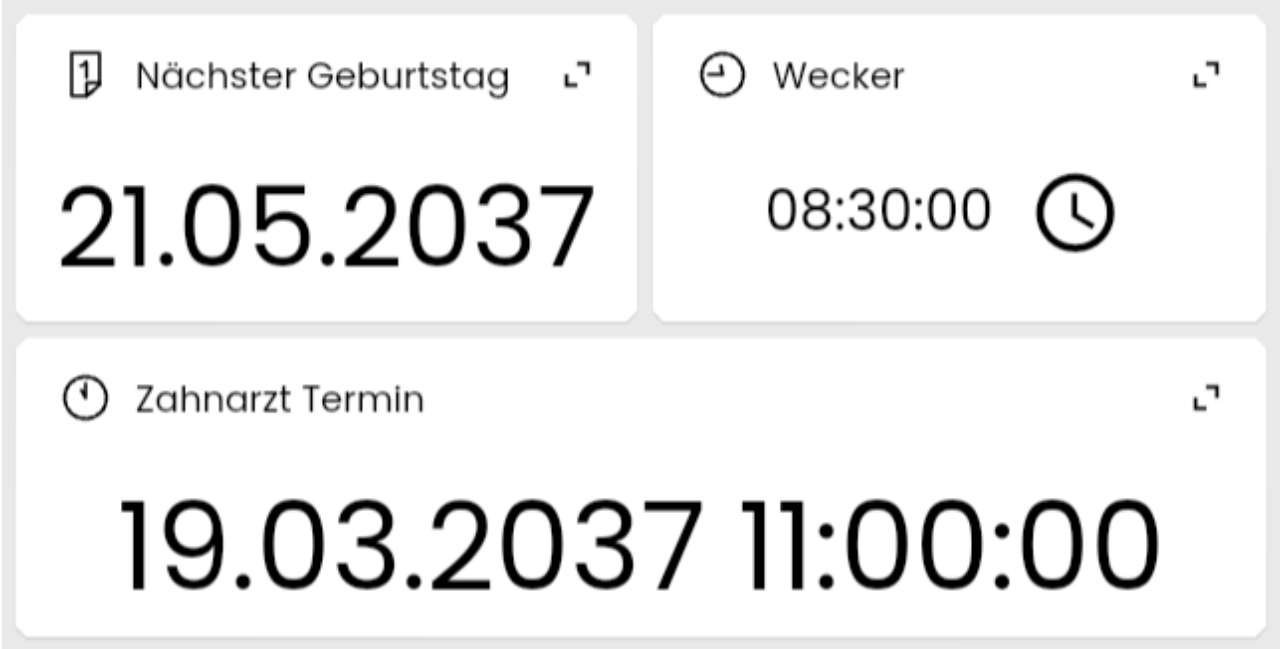
### ~UNIXTIMESTAMPDATE

Datumsanzeige: Keine
Zeitanzeige: Stunde und Minuten
Falls die Variable eine Variablenaktion hat, kann der Zeitpunkt in der Visualisierung angepasst werden

## AUSSEHEN IN KACHEL VISUALISIERUNG

In den Gittereinstellungen können die verschiedenen Darstellungen separat als "Datum", "Uhrzeit" und "Datum und Uhrzeit" konfiguriert werden.

### ALS EIGENE KACHEL




### INNERHALB EINER LISTE




## AUSSEHEN IM WEBFRONT


Nächster Geburtstag
21. Mai 2037
Wecker
08:30
Zahnarzt Termin
19. März 2037 11:00

# Dauer

Benötigt: IP-Symcon >= 8.0
Mit Dauer kann die Zeit bis/seit einem Zeitpunkt oder ein einfacher Wert als Dauer formatiert werden.

### VORAUSSETZUNGEN

Dauer ist eine Variable mit der Darstellung Dauer. Um diese Darstellung zu verwenden muss die Variable folgende Bedingungen erfüllen:
- Typ Integer oder Float
- keine eingerichtete Variablenaktion

### PARAMETER

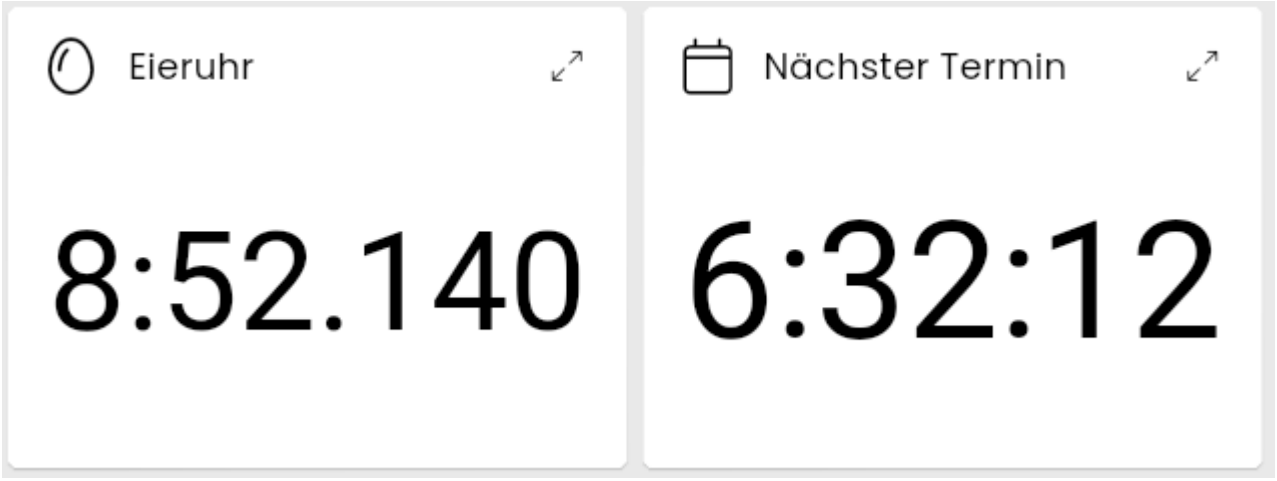
| PARAMETER | BESCHREIBUNG |
|---|---|
| Darstellungstyp | Legt fest wie der angezeigt Wert zustande kommt.<br>**Wert in Variable:** Die Dauer in Sekunden welche formatiert werden sollen<br>**Dauer bis zum Wert in der Variable:** Die Dauer bis zum als Unix Zeitstempel kodierten Zeitpunkt<br>**Dauer seit Wert in der Variable:** Die Dauer seit dem als Unix Zeitstempel kodierten Zeitpunkt |
| Format | Das Format für in dem die Dauer formatiert wird. Möglich sind: "Nur Sekunden", "Minuten und Sekunden", und "Stunden, Minuten und Sekunden" |
| Zeige Millisekunden | Bestimmt ob Millisekunden zusätzlich angezeigt werden sollen |



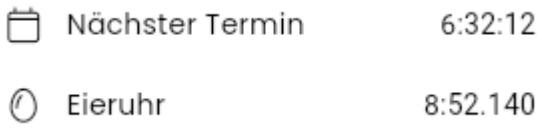
Wenn der Darstellungstyp einen Zeitpunkt beinhaltet aktualisiert sich der dargestellt Wert automatisch.

## AUSSEHEN IN KACHEL VISUALISIERUNG

### ALS EIGENE KACHEL




### INNERHALB EINER LISTE




## AUSSEHEN IM WEBFRONT

Diese Darstellung wird nicht vom WebFront unterstützt. Stattdessen wir der unformatierte Wert der Variable dargestellt.

# Diagramm

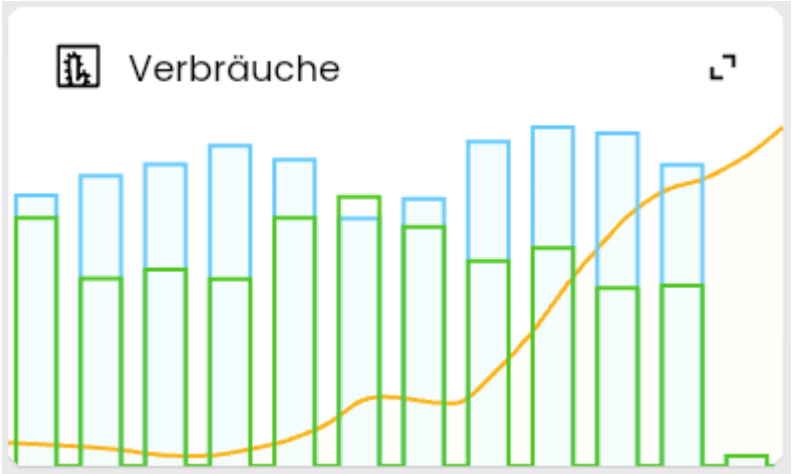
In einem Diagramm können die Verläufe mehrere Variablen dargestellt werden.

### VORAUSSETZUNGEN

Ein Diagramm ist ein Medium vom Typ Diagramm.

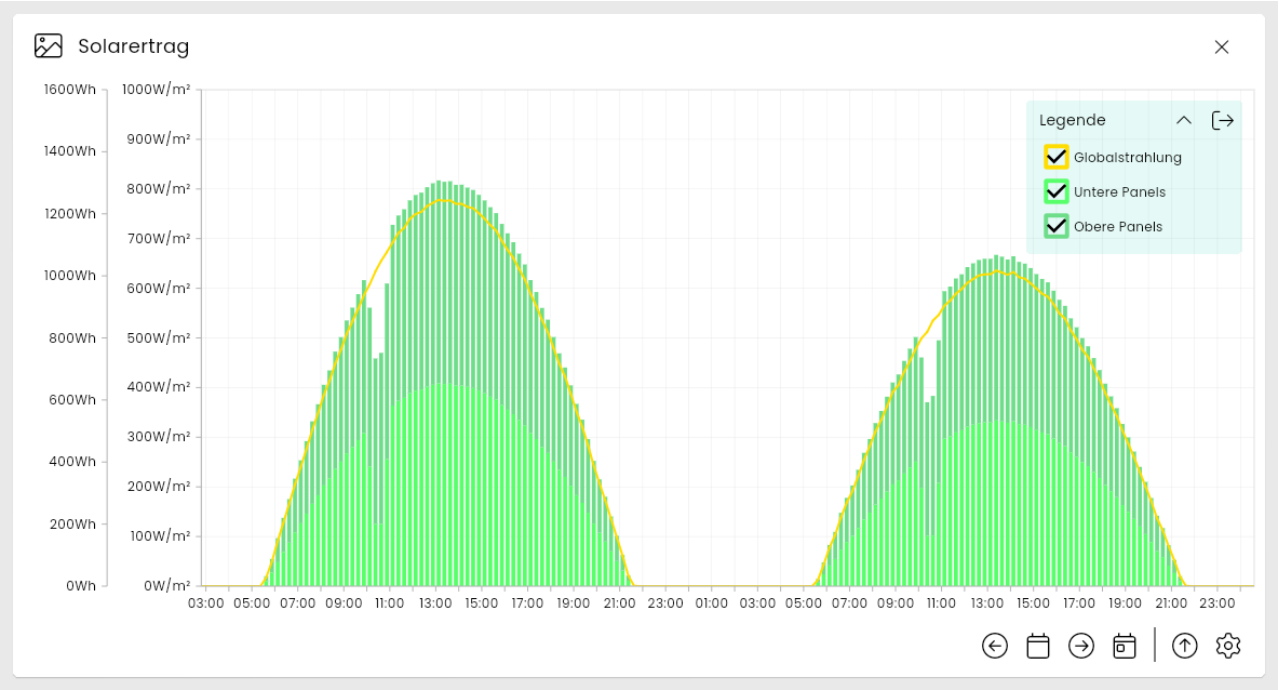
## AUSSEHEN IN KACHEL VISUALISIERUNG

### ALS EIGENE KACHEL




Ist eine Kachel ausreichend groß, wird sie stattdessen analog zur Vollbilddarstellung angezeigt.

### IM VOLLBILD

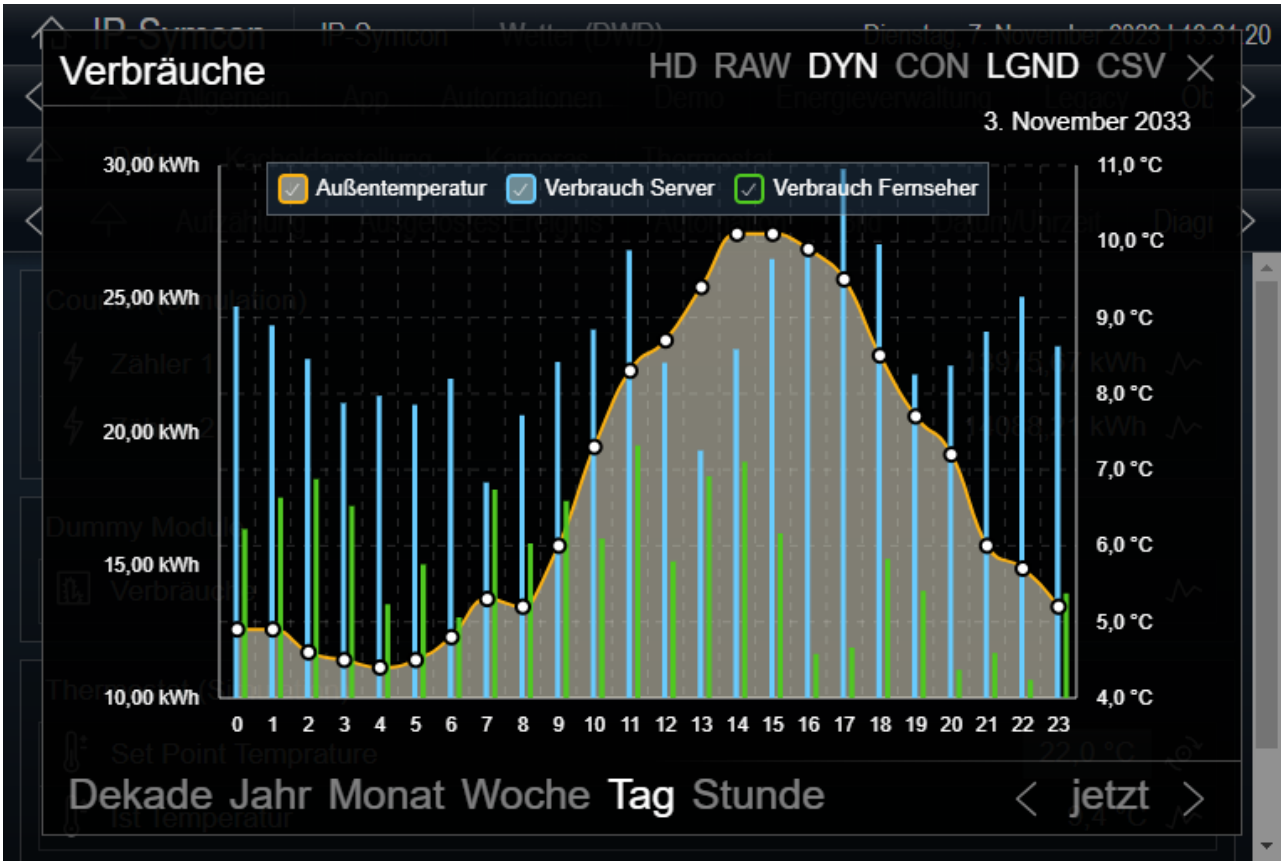



## INNERHALB EINER LISTE




## AUSSEHEN IM WEBFRONT




Verbräuche
HD RAW DYN CON LGND CSV
3. November 2033
30,00 kWh
25,00 kWh
20,00 kWh
15,00 kWh
10,00 kWh
11,0 °C
10,0 °C
9,0 °C
8,0 °C
7,0 °C
6,0 °C
5,0 °C
4,0 °C
Außentemperatur
Verbrauch Server
Verbrauch Fernseher
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
Dekade Jahr Monat Woche Tag Stunde
jetzt

# Dokument

Ein Dokument kann per Auswahl des Buttons heruntergeladen werden.

### VORAUSSETZUNGEN

Ein Dokument ist ein Medium vom Typ Dokument.

## AUSSEHEN IN KACHEL VISUALISIERUNG

### ALS EIGENE KACHEL




### INNERHALB EINER LISTE




## AUSSEHEN IM WEBFRONT



# Einzelnes Element

Benötigt: IP-Symcon >= 7.0
Ermöglicht die Darstellung eines Elements aus einer Liste als einzelne Kachel.

### VORAUSSETZUNGEN

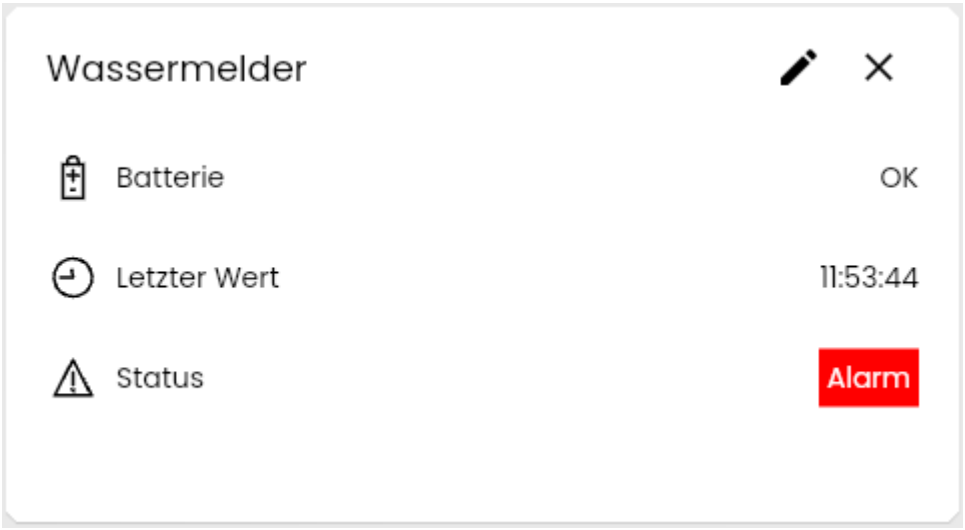
Jede Instanz kann als Einzelnes Element dargestellt werden. Die Darstellung wird automatisch gewählt, wenn eine Instanz nur ein Unterobjekt hat. Alternativ kann sie bei jeder Instanz mit mehreren Objekten über "Kachel bearbeiten" ausgewählt werden.

## AUSSEHEN IN KACHEL VISUALISIERUNG

### ALS EIGENE KACHEL




Wird die Kachel geöffnet, werden alle Objekte der Instanz als Liste dargestellt.



## AUSSEHEN IM WEBFRONT

Diese Darstellung wird nicht vom WebFront unterstützt. Stattdessen wir die Instanz als Liste dargestellt.

# Energie Manager

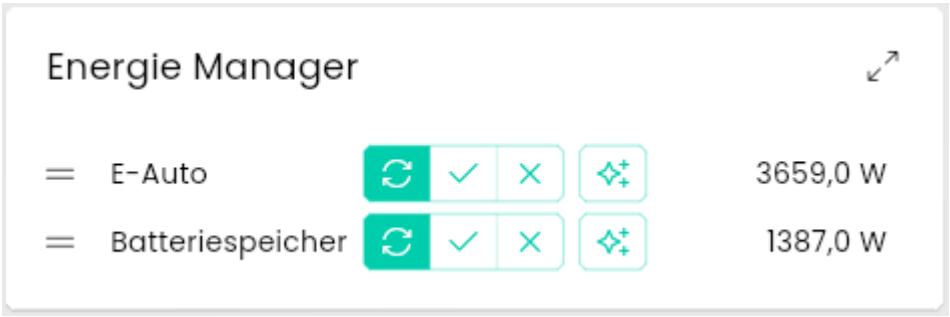
Benötigt: IP-Symcon >= 7.0
Der Energie Manager schaltet intelligent Geräte, wenn genug Energie zur Verfügung steht. Diese Darstellung ermöglicht es die Priorität einzelner Geräte Drag & Drop zu verändern oder etwa aus der Berechnung auszuschließen. Darüber hinaus kann die intelligente Optimierung wie Nachtladen oder Günstiges Laden über einen Dialog konfiguriert werden.

### VORAUSSETZUNGEN

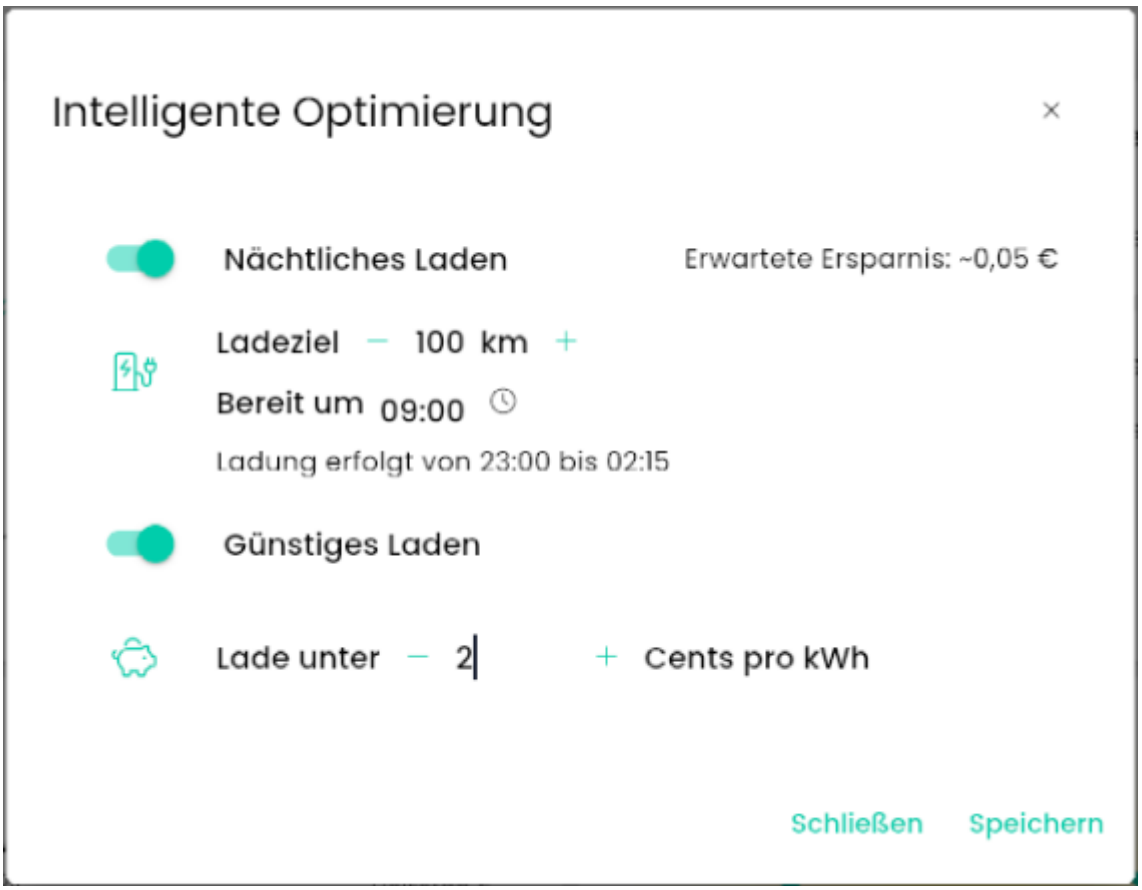
Der Energie Manager erfordert eine Instanz des Moduls Energie Manager.

## AUSSEHEN IN KACHEL VISUALISIERUNG

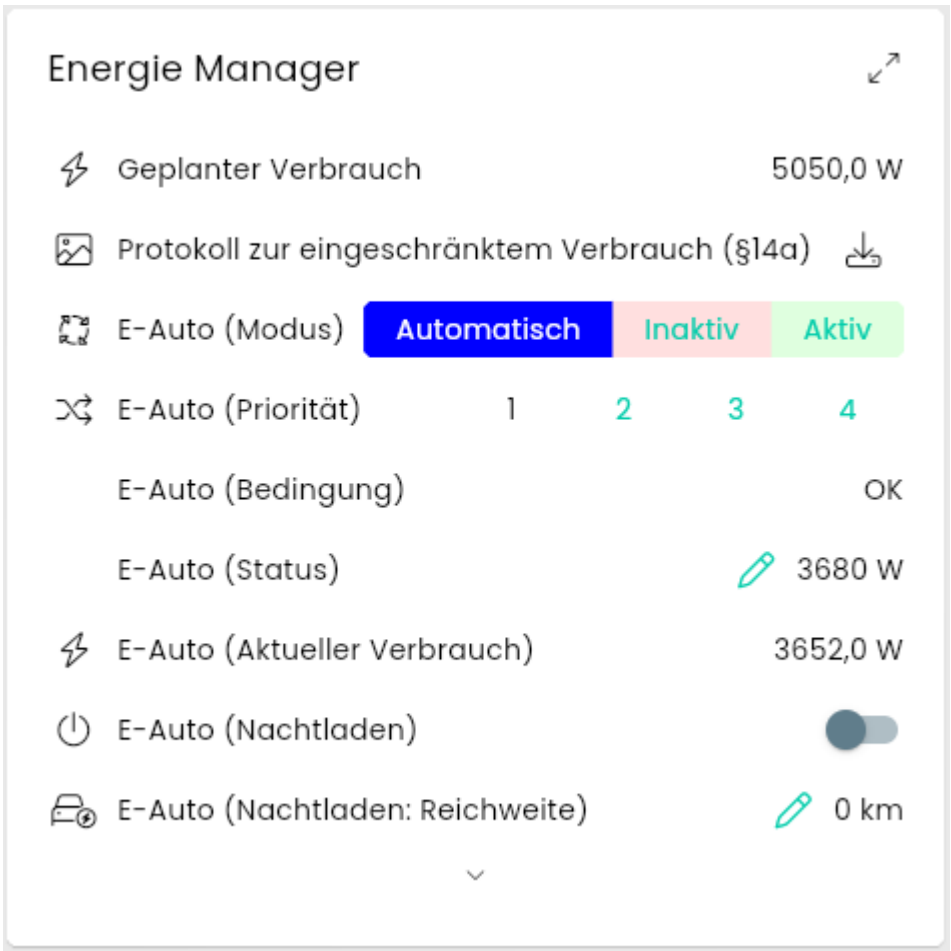
### ALS EIGENE KACHEL




### DIALOG FÜR INTELLIGENTE OPTIMIERUNG




## INNERHALB EINER LISTE



## AUSSEHEN IM WEBFRONT

Diese Darstellung wird nicht vom WebFront unterstützt. Stattdessen wir die Instanz als Liste dargestellt.

# Energieverteilung

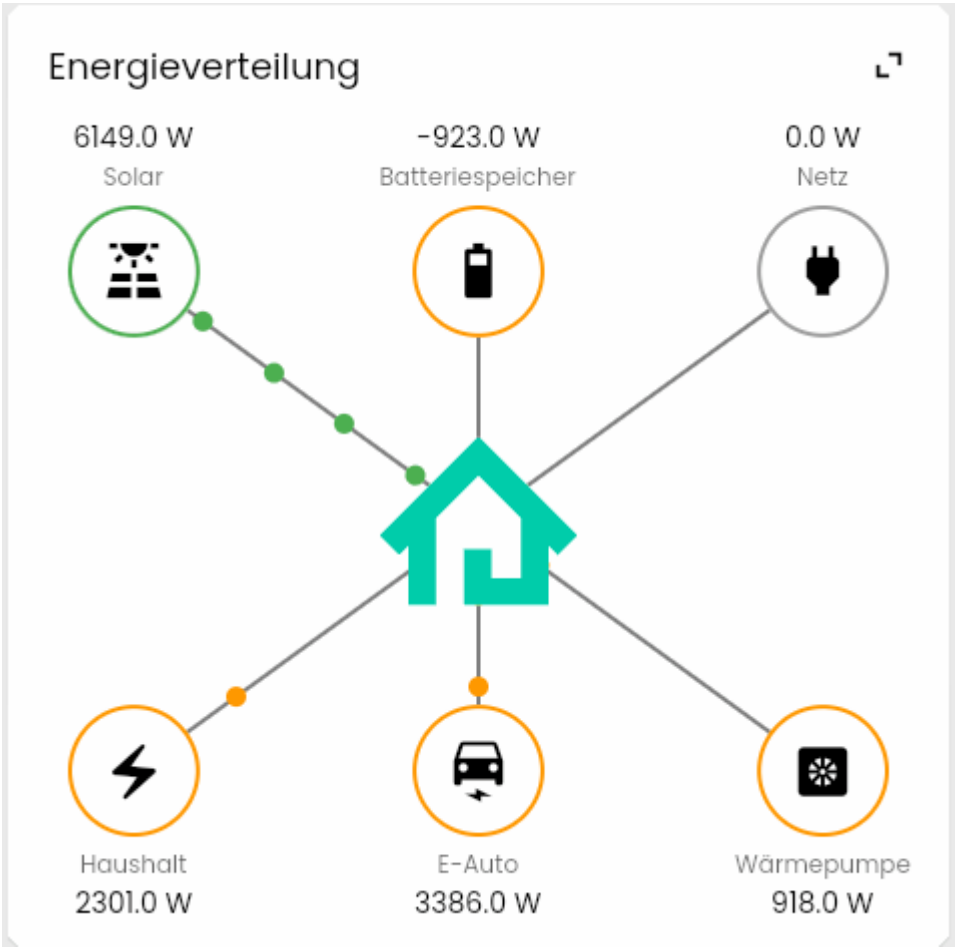
Benötigt: IP-Symcon >= 7.0
Die Darstellung Energieverteilung stellt den Fluss von erzeugter und verbrauchter Energie dar.

### VORAUSSETZUNGEN

Die Darstellung Energieverteilung erfordert eine Instanz des Moduls Energieverteilung.

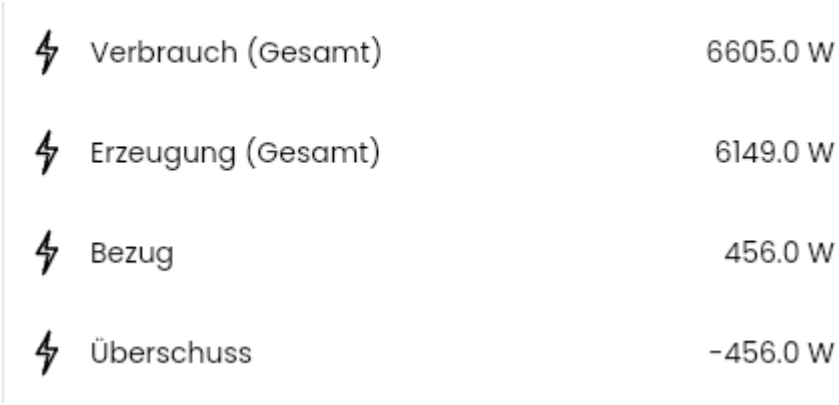
## AUSSEHEN IN KACHEL VISUALISIERUNG

### ALS EIGENE KACHEL




### INNERHALB EINER LISTE

Energieverteilung




## AUSSEHEN IM WEBFRONT

Diese Darstellung wird nicht vom WebFront unterstützt. Stattdessen wir die Instanz als Liste dargestellt.

# Farbe

Mit einer Farbe kann eine Farbe dargestellt und ggfs. konfiguriert werden.

### VORAUSSETZUNGEN

Eine Farbe ist eine Variable mit der Darstellung Farbe. Um diese Darstellung zu verwenden muss die Variable folgende Bedingungen erfüllen:
Typ Integer oder String

### PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| Kodierung | Verwendete Kodierung der Farbe, nur anpassbar für Variablen vom Typ String |
| Standardwerte | Liste mit Standardwerten, welche in einer Auswahl angeboten werden, nur verfügbar für Variablen mit Variablenaktion. Nicht verfügbar bei Kodierung 'xy' |
| Farbraum | Der Farbraum aus welchem eine Farbe ausgewählt werden kann. Dieser bestimmt ein Dreieck |
| Benutzerdefinierter Farbraum | Liste in welcher der Rot-, Grün-, Blau-Wert und Weißpunkt eines eigenen Farbraumes definiert werden kann. |
| Farbkurve | Wenn eine Farbkurve ausgewählt ist, wird die Auswahl der Farbe auf diese beschränkt |
| Benutzerdefinierte Farbkurve | Eine List von Koordinaten, die den |

## DARSTELLUNG ALS LEGACY-PROFIL

Verwendet eine Variable die Darstellung Legacy Profil, so wird diese als Farbe darstellt, falls sie zusätzlich folgende Bedingungen erfüllt:
ausgewähltes Variablenprofil "~HexColor"

## AUSSEHEN IN KACHEL VISUALISIERUNG

### ALS EIGENE KACHEL MIT VARIABLENAKTION

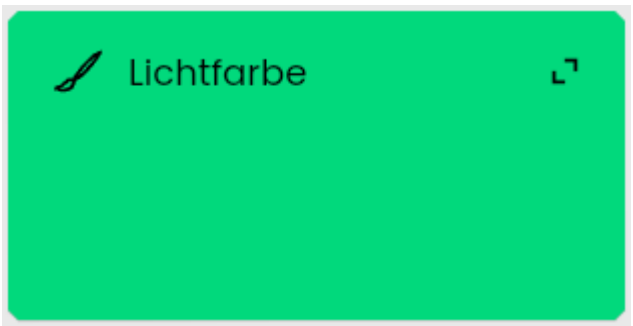






Bei Auswahl der Pipette in der Mitte des Kreises öffnet sich ein Dialog um die Farbe alternativ im Detail einzustellen

### ALS EIGENE KACHEL OHNE VARIABLENAKTION



### INNERHALB EINER LISTE MIT VARIABLENAKTION




Bei Auswahl des Stifts öffnet sich ein Dialog, analog zur Darstellung als eigene Kachel.

### INNERHALB EINER LISTE OHNE VARIABLENAKTION




## AUSSEHEN IM WEBFRONT

### MIT VARIABLENAKTION




Bei Auswahl des Pinsels rechts öffnet sich ein Dialog zur Farbauswahl:




### OHNE VARIABLENAKTION



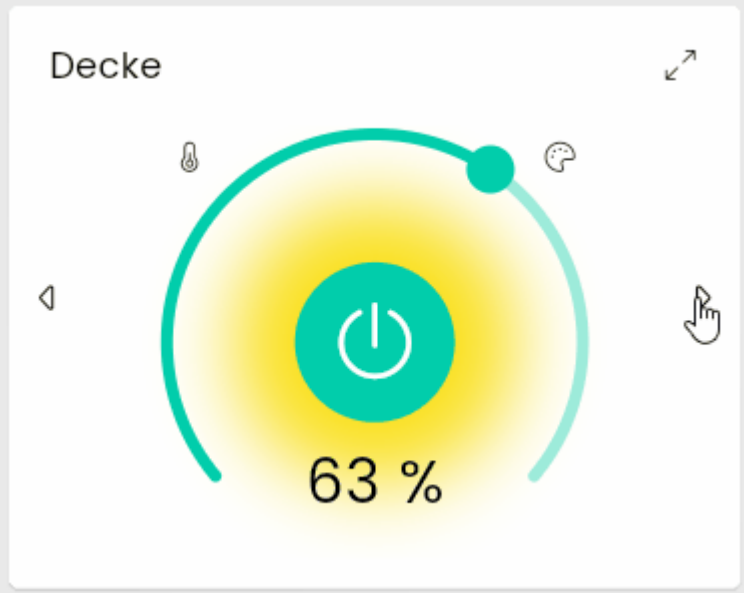
# Inhaltswechsler

Ein Inhaltswechsler erlaubt es untergeordnete Objekte innerhalb einer einzelnen Kachel darzustellen. Die einzelnen Objekte können durch ein Klicken oder nach einer gewissen Zeit automatisch durchgewechselt werden.

### VORAUSSETZUNGEN

Ein Objekt des Typs Kategorie bei welcher in der Visualisierung im Editiermodus die Darstellung Inhaltswechsler ausgewählt wurde.

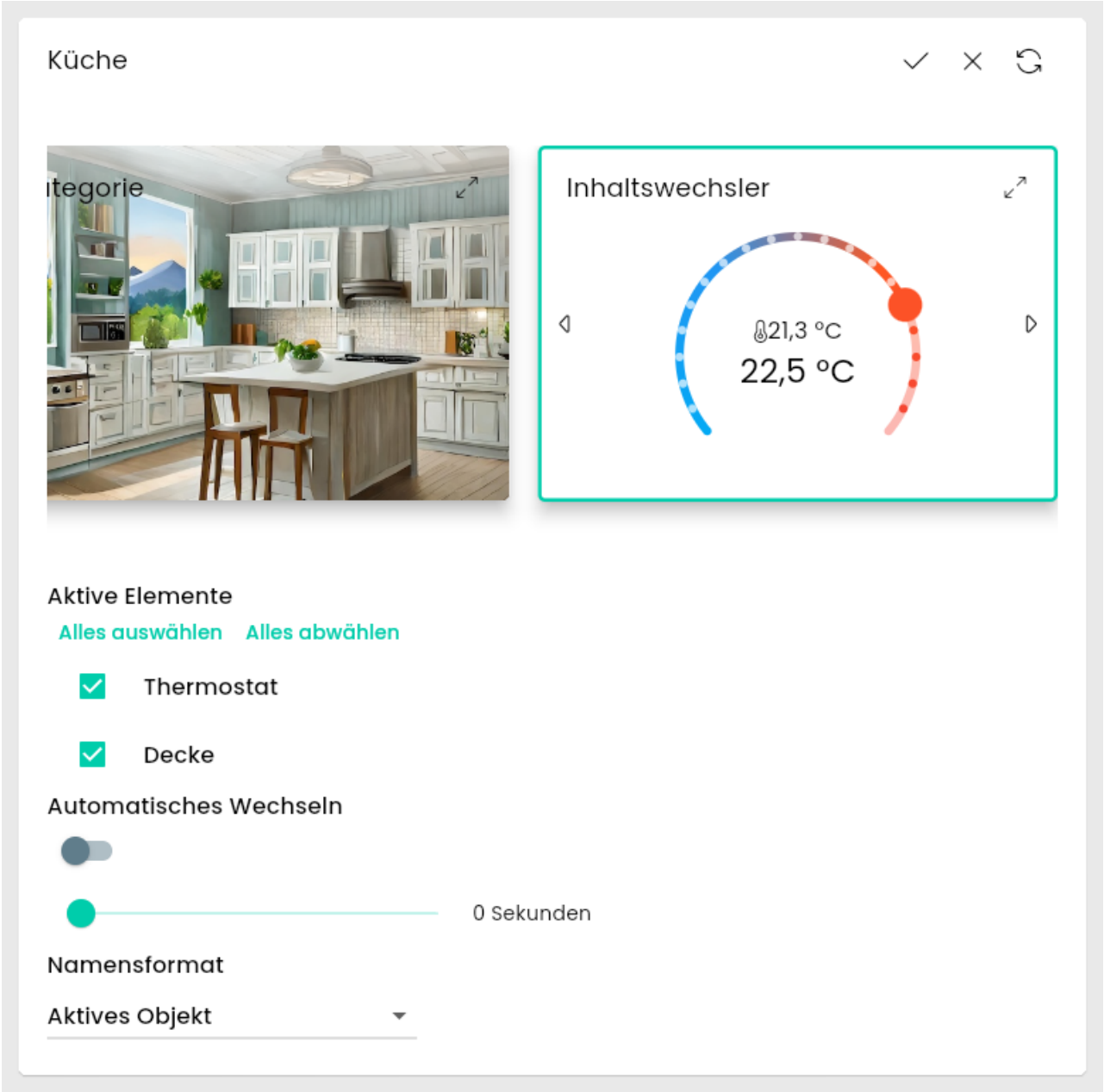
## AUSSEHEN IN KACHEL VISUALISIERUNG




## KONFIGURATION

Wenn ein Inhaltswechsler in der Visualisierung editiert wird, stehen folgende Parameter zur Verfügung.

| NAME | BESCHREIBUNG |
|---|---|
| Aktive Elemente | Hier kann aus allen sichtbaren Elementen der Kategorie ausgewählt werden, welche angezeigt werden sollen. |
| Automatisches Wechseln | Wenn aktiv, springt der Inhaltswechsler nach einer festgelegten Dauer zum nächsten Element. Die Maximale Dauer liegt bei 100 Sekunden |
| Namensformat | Der Name der als Titel der Kachel angezeigt werden soll. (Aktives Objekt, Kategorie, Aktives Objekt und Kategorie) |




## AUSSEHEN IM WEBFRONT

Diese Darstellung wird wie eine normale Kategorie angezeigt.




Etwas Vergleichbares kann über den Inhaltswechsler im WebFront Editor erreicht werden.

# IPSView

IPSViews lassen sich nahtlos in die Visualisierung einbinden.

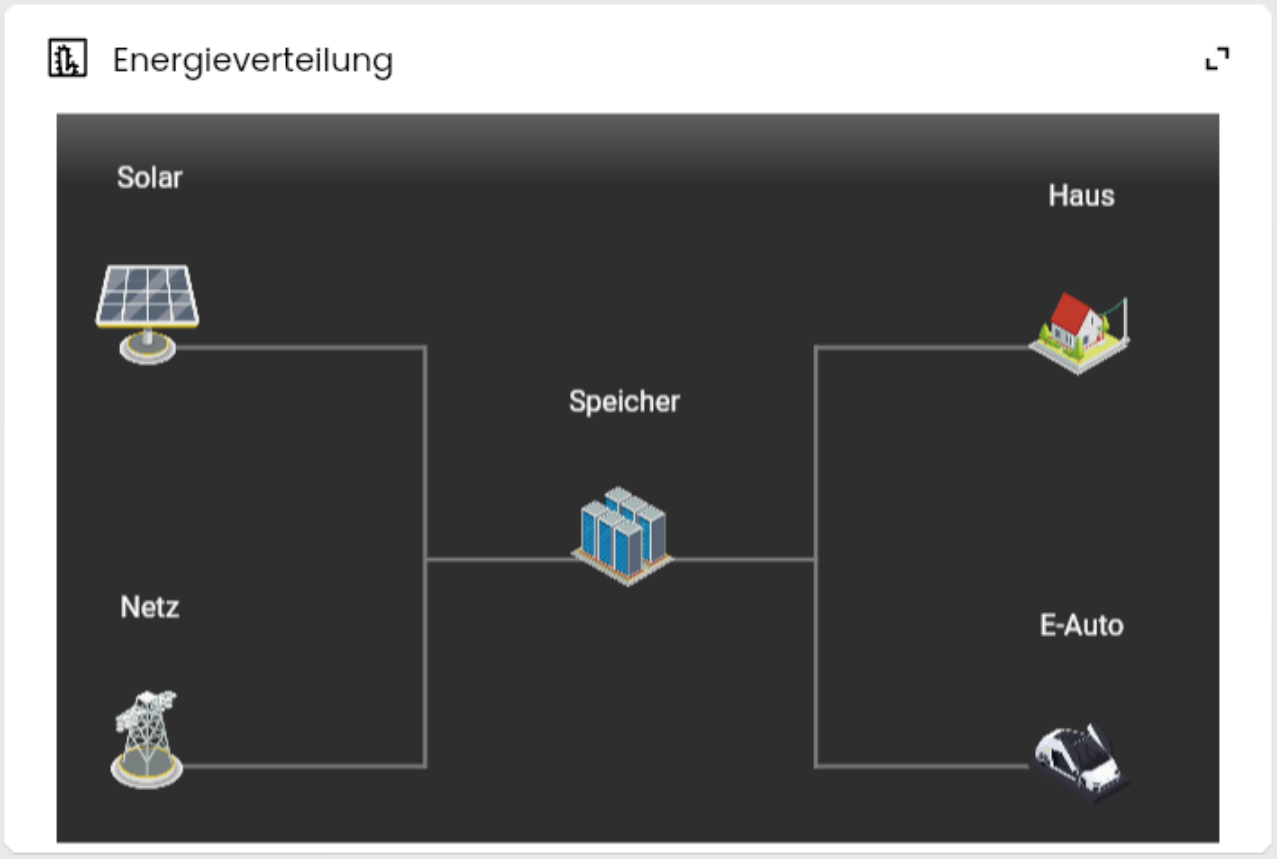
> Die Erweiterung IPSView muss zur Verwendung separat erworben werden.

### VORAUSSETZUNGEN

Eine IPSView ist ein Medium vom Typ IPSView.

## AUSSEHEN IN KACHEL VISUALISIERUNG

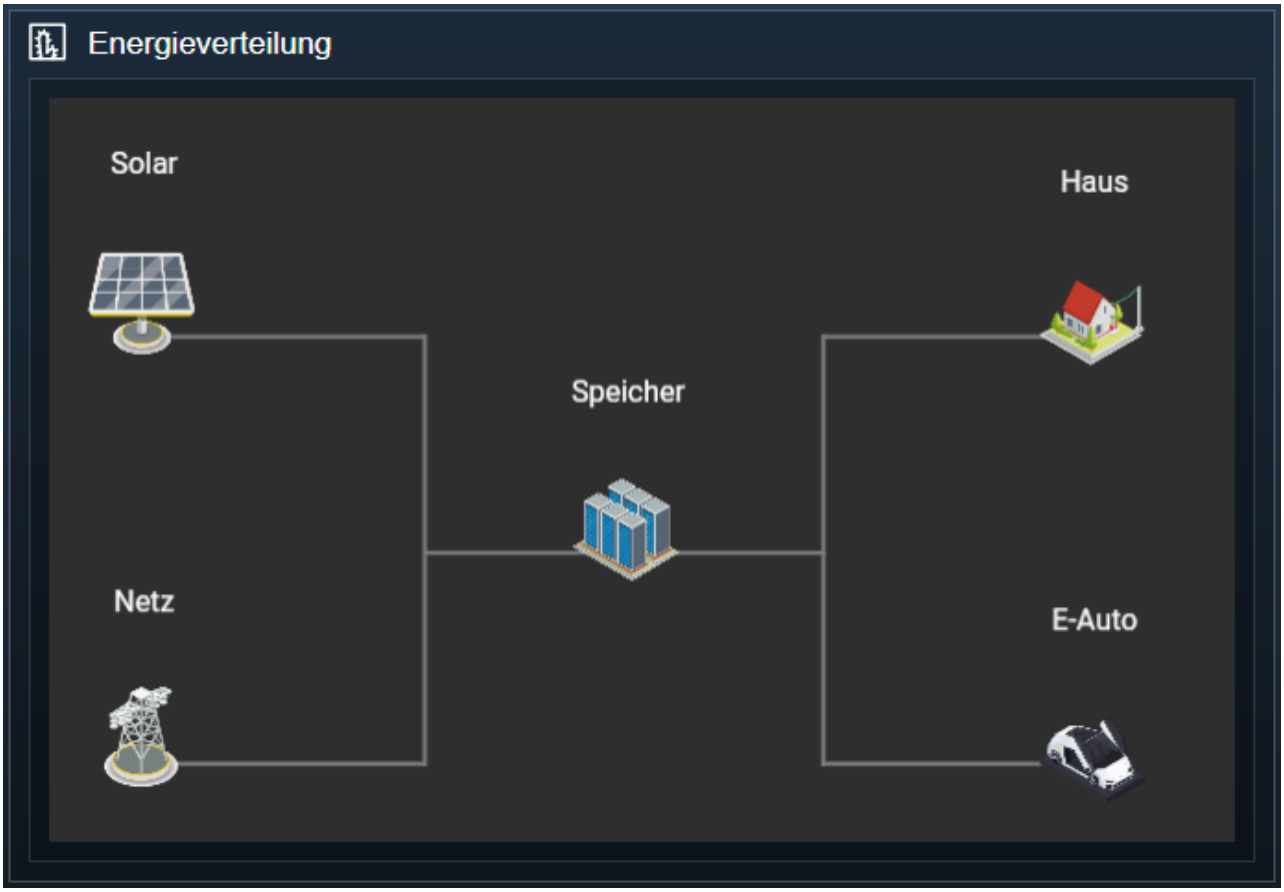
### ALS EIGENE KACHEL




### INNERHALB EINER LISTE



## AUSSEHEN IM WEBFRONT



# Rollladen

Für einen Rollladen kann der Öffnungsgrad sowie optional der Neigungsgrad der Lamellen eingestellt werden.

## VORAUSSETZUNGEN

Ein Rollladen kann entweder als Variable oder als Instanz dargestellt werden.

### ALS VARIABLE

Eine Variable mit der Darstellung Rollladen, muss folgende Voraussetzungen erfüllen:
Typ Integer
Eingerichtete Variablenaktion

### PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| Verwendungstyp | Bestimmt ob mit dem Schieberegler die Rotation von Lamellen oder der Öffnungsgrad des Rollladen gesteuert wird. Die Der Parameter Verwendungstyp bestimmt welche weitere Parameter zur Verfügung stehen. Die Auswahl bestimmt welche weiteren Parameter zur Verfügung stehen |
| Offen bei | Der Wert bei dem der Rollladen als geöffnet gilt (nur bei Verwendungstyp Öffnung) |
| Geschlossen bei | Der Wert bei dem der Rollladen als geschlossen gilt (nur bei Verwendungstyp Öffnung) |
| Innen bei | Der Wert bei dem die Lamellen als nach innen gedreht gelten (nur bei Verwendungstyp Rotation) |
| Außen bei | Der Wert bei dem die Lamellen als nach außen gedreht gelten (nur bei Verwendungstyp Rotation) |
| Rotation | Die feste rotation der dargestellten Lamellen (nur bei Verwendungstyp Öffnung) |
| Maximale Rotation nach innen | Der maximal Rotationswert der Lamellen nach innen in Grad (nur bei Verwendungstyp Rotation) |
| Maximale Rotation nach außen | Der maximale Rotationswert der Lamellen nach außen in Grad (nur bei Verwendungstyp Rotation) |
| Sonnenposition | Bestimmt ob und wo eine Sonne neben dem Rolladen angezeigt werden soll. Die Höhe der Sonne ist von der Tageszeit abhängig. In der Nacht wird ein Mond angezeigt |

### ALS INSTANZ

Eine Instanz kann als Rollladen dargestellt werden, wenn folgende untergeordnete Objekte vorhanden sind:
Position (optional): Variable mit dem Verwendungstyp Öffnung

Drehgrad (optional): Variable mit dem Verwendungstyp Rotation
Mindestens eine optionale Variable muss existieren.



Die Standardgröße von Rollläden kann in den Gittereinstellungen für Variablen als "Rollladenvariable" und Instanzen als "Rollladen" separat konfiguriert werden

## DARSTELLUNG ALS LEGACY-PROFIL

Verwendet eine Variable die Darstellung Legacy Profil, so wird diese als Rollladen darstellt, falls sie eines der folgenden Variablenprofile ausgewählt ist:
~Shutter
~Lamella
Die unterschiedlichen Profile entsprechen festen Parameter Kombinationen.

### ~SHUTTER

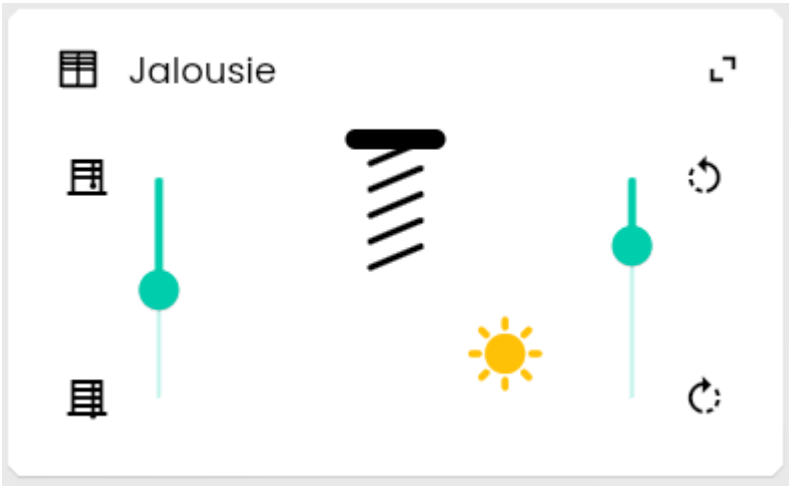
Verwendungstyp: Öffnung
Offen bei: 0
Geschlossen bei: 100
Rotation: -55°
Sonnenposition: Rechts

### ~LAMELLA

Verwendungstyp: Rotation
Innen bei: 0
Außen bei: 100
Maximale Rotation nach innen: -55°
Maximale Rotation nach außen: 55°
Sonnenposition: Rechts

## AUSSEHEN IN KACHEL VISUALISIERUNG

### ALS EIGENE KACHEL




### INNERHALB EINER LISTE

In der Listenansicht fällt die Visualisierung auf die Darstellung Liste zurück.




## AUSSEHEN IM WEBFRONT

Diese Darstellung wird nicht vom WebFront unterstützt. Stattdessen werden Variablen als Schieberegler dargestellt und Instanzen als Liste.

# Kategorie

Eine Kategorie kann geöffnet werden um die darin enthaltenen Objekte zu betrachten.

### VORAUSSETZUNGEN

Ein Objekt des Typs Kategorie.

## AUSSEHEN IN KACHEL VISUALISIERUNG

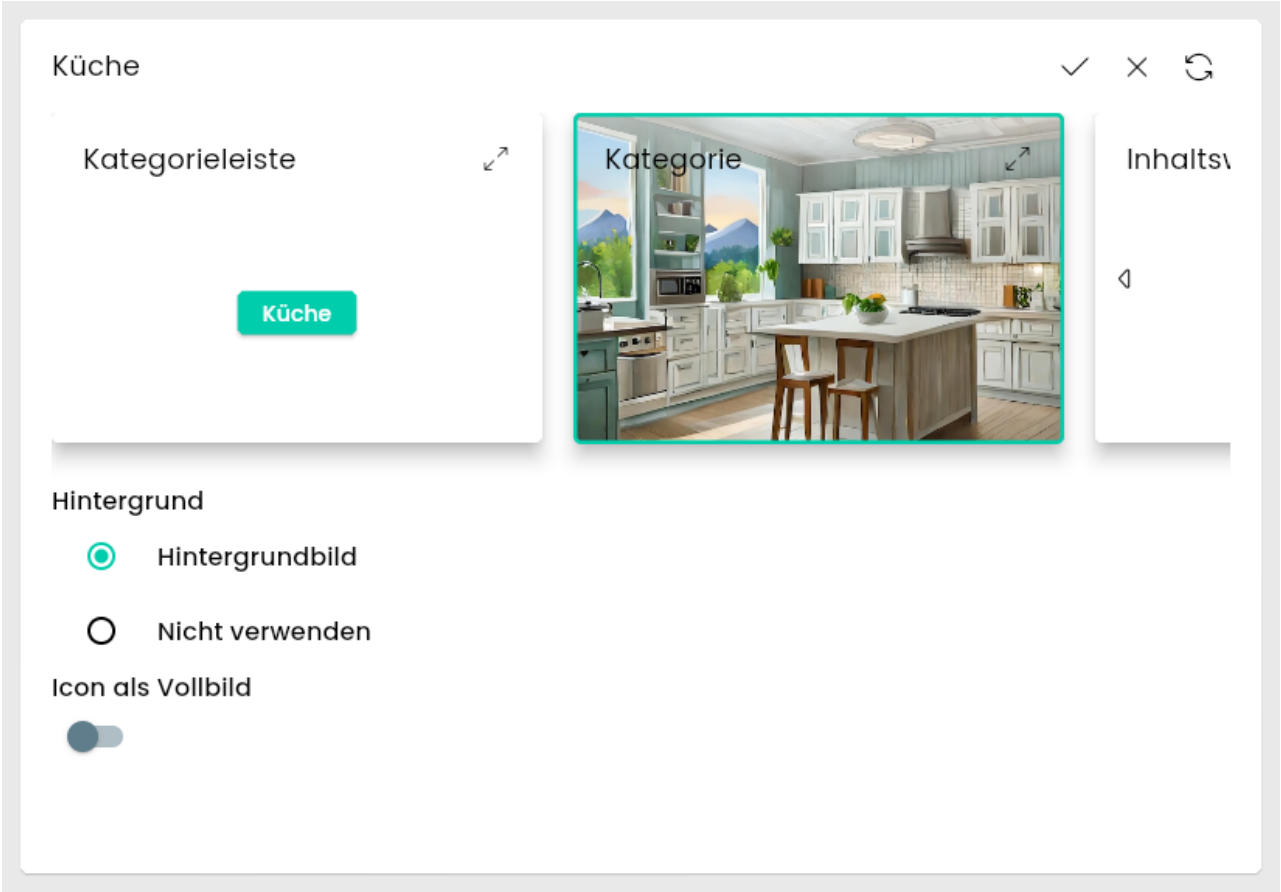
In der Instanzkonfiguration Kann bei 'Standard Kategorie Darstellung' 'Kacheln' ausgewählt werden. Diese Standardeinstellung kann im Editiermodus in der Visualisierung je Objekt überschrieben werden.




## KONFIGURATION

Wenn eine Kategorie in der Visualisierung editiert wird, stehen folgende Parameter zur Verfügung.

| NAME | BESCHREIBUNG |
|---|---|
| Hintergrund | Ist in der Kategorie ein Bild als Medienobjekt vorhanden, kann dieses hier als Hintergrund ausgewählt werden. Das Medienobjekt darf auch über die 'Visuellen Einstellungen' in der Konsole unsichtbar geschaltet sein. |
| Icon als Vollbild | Wenn aktiv wird das Icon des Kategorie-Objekts mittig auf der Kachel dargestellt |




## AUSSEHEN IM WEBFRONT



# Kategorieleiste

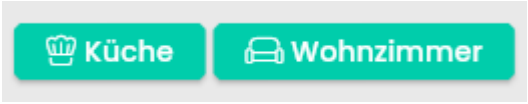
Eine Kategorie kann geöffnet werden um die darin enthaltenen Objekte zu betrachten.

### VORAUSSETZUNGEN

Ein Objekt des Typs Kategorie.

## AUSSEHEN IN KACHEL VISUALISIERUNG

In der Instanzkonfiguration Kann bei 'Standard Kategorie Darstellung' als 'Standard Kategorie Darstellung' 'Kategorieleiste (Einzeilig)' oder 'Kategorieleiste (Mehrzeilig)' ausgewählt werden. Diese Standardeinstellung kann im Editiermodus in der Visualisierung je Objekt überschrieben werden. Neben dem Objektnamen wird auch das Icon der Kategorie angezeigt.




## AUSSEHEN IM WEBFRONT



# Legacy Profil

Diese Variablendarstellung wertet ein Variablenprofil aus und übersetzt es auf die tatsächliche Darstellung.

### VORAUSSETZUNGEN

Eine Variable verwendet die Variablendarstellung "Legacy Profil". Die Darstellung hat keine weiteren Voraussetzungen.

### PARAMETER

| PARAMETER | BESCHREIBUNG |
| --- | --- |
| Profil | Das zugrunde liegende Variablenprofil |

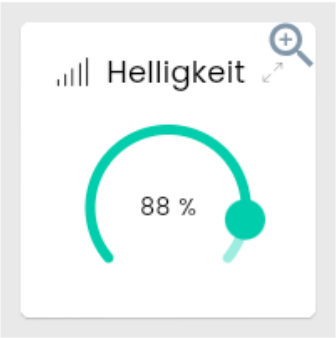
Über Legacy Profile sind folgende Variablendarstellungen realisierbar. Die genauen Anforderungen können den einzelnen Seiten für die entsprechende Darstellung entnommen werden:

- Aufzählung
- Datum/Uhrzeit
- Farbe
- HTML Box
- Jalousie
- Schalter
- Schieberegler
- Wertanzeige
- Werteingabe

## KONVERTIEREN

In dem "Variable bearbeiten" Dialog kann unter dem Punk Variablendarstellung über die entsprechende Schaltfläche eine Legacy-Darstellung in ihre tatsächliche Darstellung konvertierte werden.

Variablendarstellung




AUF STANDARD ZURÜCKSETZEN

Darstellung
Legacy Profil

Vorlage
(Benutzerdefiniert)

## Darstellungsparameter

Profil
~Intensity.100

KONVERTIERE LEGACY PROFIL ZU NEUER DARSTELLUNG

# Licht

Benötigt: IP-Symcon >= 7.0
In der Darstellung Licht können Status, Helligkeit, Farbtemperatur und Farbe in einer einzelnen Kachel visualisiert werden.

### VORAUSSETZUNGEN

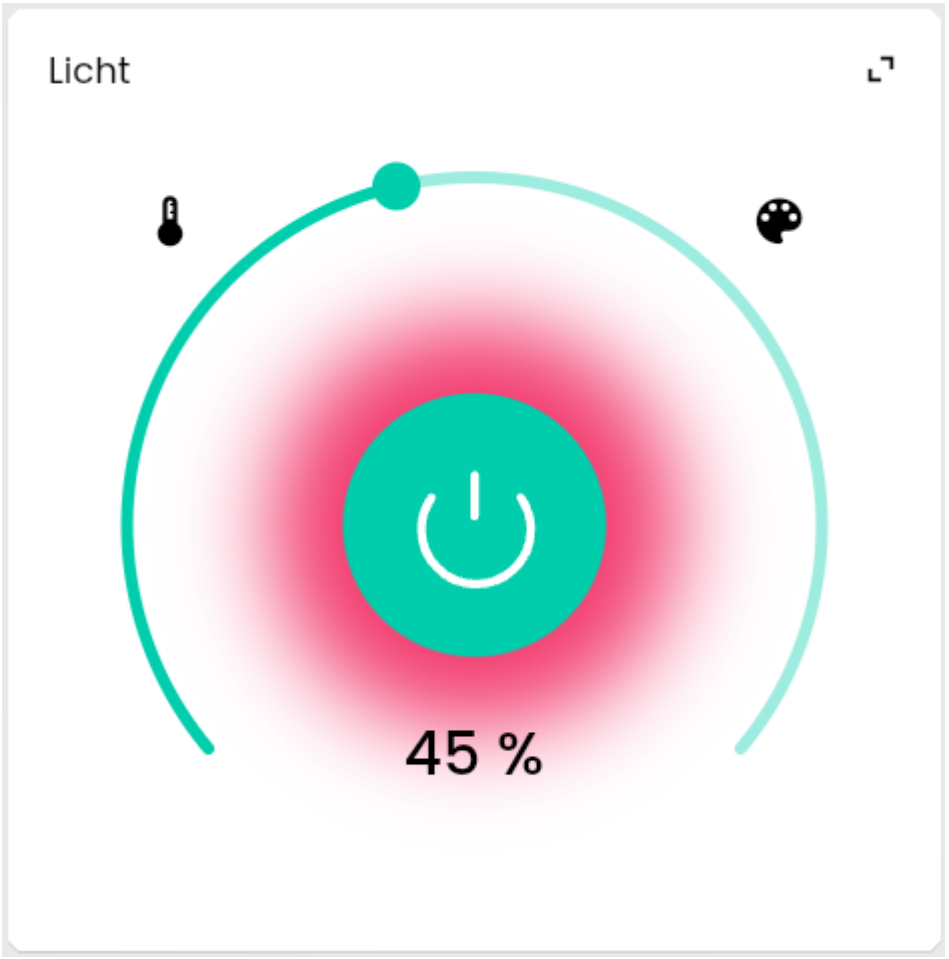
Eine Instanz kann als Licht dargestellt werden, wenn folgende untergeordnete Objekte vorhanden sind:

- Status:
  - Verwendet die Variablendarstellung Schalter
  - Verwendung der Variablen ist "An/Aus"
- Helligkeit (optional):
  - Verwendet die Variablendarstellung Schieberegler
  - Verwendung der Variablen ist "Intensität"
- Farbtemperatur (optional):
  - Verwendet die Variablendarstellung Schieberegler
  - Verwendung der Variablen ist "Farbtemperatur"
- Farbe (optional):
  - Verwendet die Variablendarstellung Farbe
  - Eingerichtete Variablenaktion

Mindestens eine optionale Variable muss existieren.

## AUSSEHEN IN KACHEL VISUALISIERUNG

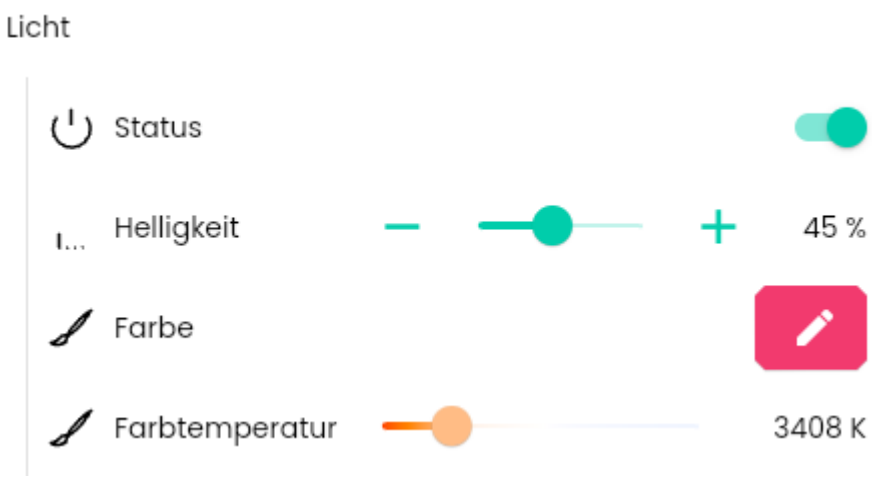
### ALS EIGENE KACHEL


Licht
45 %


Licht

### INNERHALB EINER LISTE

In der Listenansicht fällt die Visualisierung auf die Darstellung Liste zurück.




## AUSSEHEN IM WEBFRONT

Diese Darstellung wird nicht vom WebFront unterstützt. Stattdessen wir die Instanz als Liste dargestellt.

# Liste

Eine Listendarstellung kann beliebige Objekte untereinander darstellen.

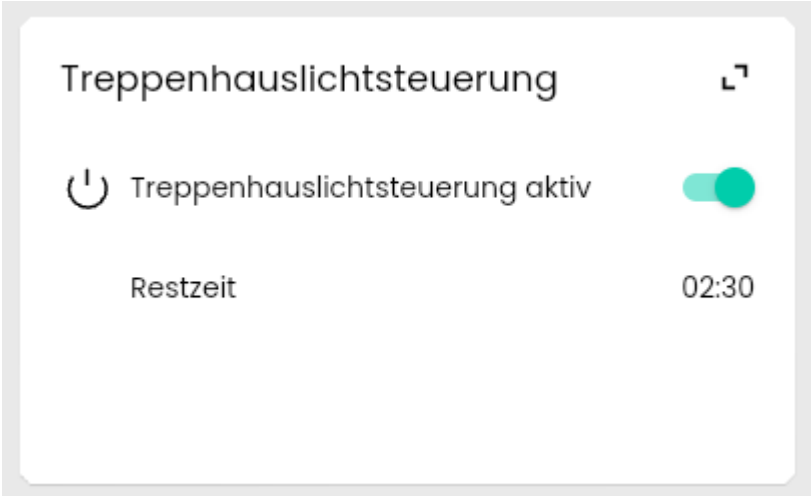
### VORAUSSETZUNGEN

Jede Instanz kann als Liste dargestellt werden.

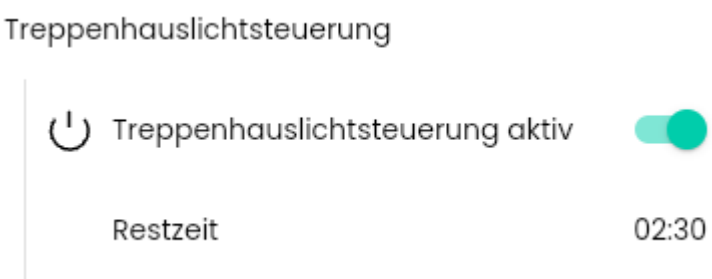
> Sollen einfach Werte zusammengefasst werden ohne dass es eine passende übergeordnete Instanz gibt, kann ein Dummy Modul zur Strukturierung verwendet werden

## AUSSEHEN IN KACHEL VISUALISIERUNG

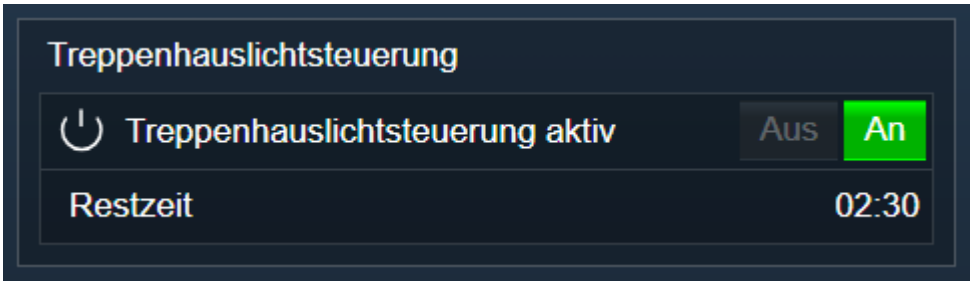
### ALS EIGENE KACHEL




### INNERHALB EINER LISTE




## AUSSEHEN IM WEBFRONT


Treppenhauslichtsteuerung
Treppenhauslichtsteuerung aktiv
Aus
An
Restzeit
02:30

# Media Player

Benötigt: IP-Symcon >= 7.0
Ein Media Player kann diverse Parameter zur Medienwiedergabe darstellen und ggfs. schalten lassen:

- Medienwiedergabe abspielen und pausieren
- Medienwiedergabe stoppen (optional)
- Sprung zum nächsten oder vorherigen Element (optional)
- Anzeige und Anpassung des Fortschritts im aktuellen Element (optional)
- Lautstärke einstellen (optional)
- Stumm schalten (optional)
- Wiederholung einstellen (optional)
- Zufallswiedergabe aktivieren (optional)
- Cover anzeigen (optional)
- Interpret anzeigen (optional)
- Titel anzeigen (optional)
- Aktuelle Wiedergabeliste inklusive aktueller Position darstellen und anpassen (optional)

## VORAUSSETZUNGEN

Ein Media Player kann entweder als Variable oder als Instanz dargestellt werden.

### ALS VARIABLE

Eine Variable, die als Media Player dargestellt werden soll, muss folgende Voraussetzungen erfüllen:

- Typ Integer
- Eingerichtete Variablenaktion
- ausgewähltes Variablenprofil:
  - ~PlaybackNoStop: Unterstützt nur abspielen und pausieren
  - ~Playback: Unterstützt abspielen, pausieren und stoppen
  - ~PlaybackPreviousNextNoStop: Unterstützt abspielen, pausieren und Sprung zum nächsten oder vorherigen Element
  - ~PlaybackPreviousNext: Unterstützt abspielen, pausieren, stoppen und Sprung zum nächsten oder vorherigen Element
  - Alle weiteren Funktionen des Media Players werden bei Darstellung als Variable nicht unterstützt.

### ALS INSTANZ

Eine Instanz kann als Media Player dargestellt werden, wenn folgende untergeordnete Objekte vorhanden sind:

- Wiedergabe:
  - Variable wie oben beschrieben
- Fortschritt (optional):
  - Verwendet die Variablendarstellung Schieberegler
  - Verwendung der Variablen ist "Fortschritt"
  - Alternativ kann eine Variable des Typs Integer mit der Variablendarstellung Legacy Profil und dem Profil "~Progress" verwendet werden

Falls die Variable eine Variablenaktion hat, ist die Position anpassbar, ansonsten wird sie lediglich dargestellt

Lautstärke (optional):
Verwendet die Variablendarstellung Schieberegler
Verwendung der Variablen ist "Lautstärke"

Stumm (optional):
Verwendet die Variablendarstellung Schalter
Verwendung der Variablen ist "Stumm schalten"

Wiederholung (optional):
Aufzählung
Variablentyp Integer
Variablenprofil "~Repeat"

Zufallswiedergabe (optional):
Aufzählung
Variablentyp Boolean
Variablenprofil "~Shuffle"

Cover (optional):
Bild

Interpret (optional):
Variable vom Typ String
Variablenprofil "~Artist"

Titel (optional):
Variable vom Typ String
Variablenprofil "~Song"

Wiedergabeliste (optional):
Variable vom Typ String
Variablenprofil "~Playlist"
Falls die Variable eine Variablenaktion hat, ist die Position anpassbar, ansonsten wird sie lediglich dargestellt
Der Wert für die Wiedergabeliste ist ein JSON-codiertes Objekt mit folgenden Parametern:

| PARAMETER | TYP | BESCHREIBUNG |
|---|---|---|
| entries | Array | Einträge der Wiedergabeliste |
| current (optional) | Integer | Der Index des aktuell aktiven Eintrags |

Darstellung der Einträge

| PARAMETER | TYP | BESCHREIBUNG |
|---|---|---|
| song (optional) | String | Titel des Eintrags |
| artist (optional) | String | Interpret des Eintrags |
| duration (optional) | Integer | Dauer des Eintrags in Sekunden |
| Weitere Einträge (optional) | beliebig | Für die Funktionalität können beliebige weitere Parameter hinzugefügt werden. Diese werden allerdings nicht dargestellt |

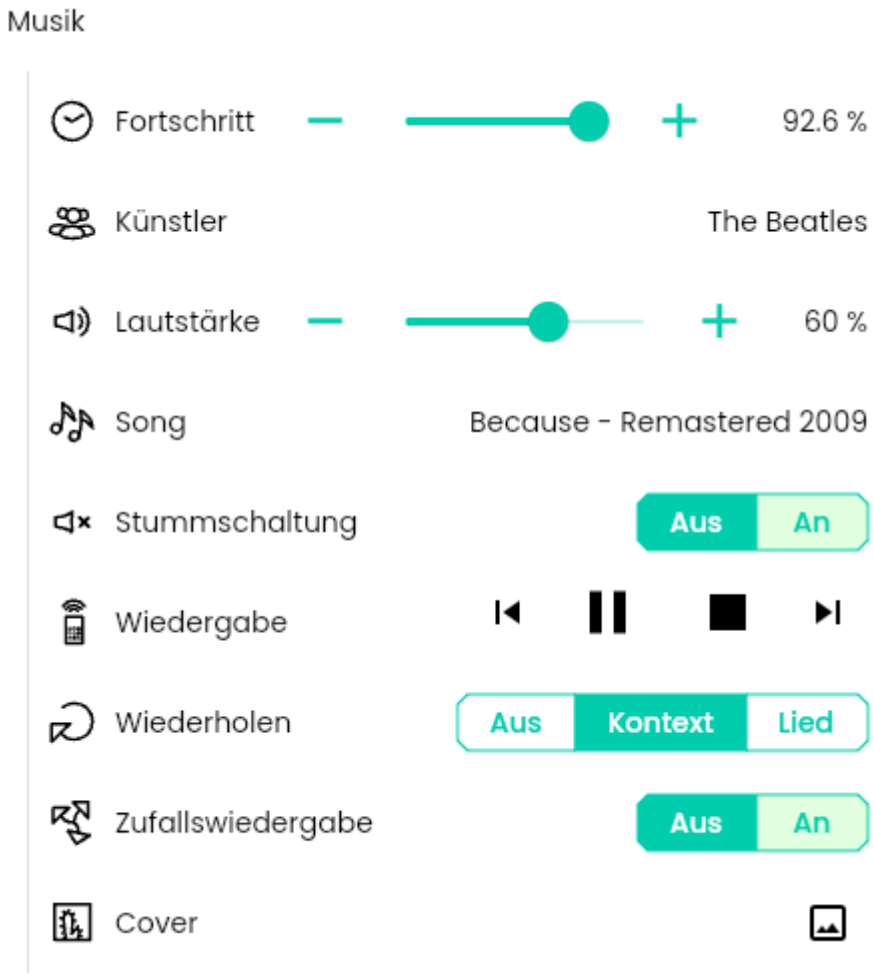
## AUSSEHEN IN KACHEL VISUALISIERUNG

## ALS EIGENE KACHEL




## INNERHALB EINER LISTE

In der Listenansicht fällt die Visualisierung auf die Darstellung Liste zurück.




## AUSSEHEN IM WEBFRONT

Diese Darstellung wird nicht vom WebFront unterstützt. Stattdessen werden Variablen als Aufzählung dargestellt und Instanzen als Liste.

# Mitteilungen

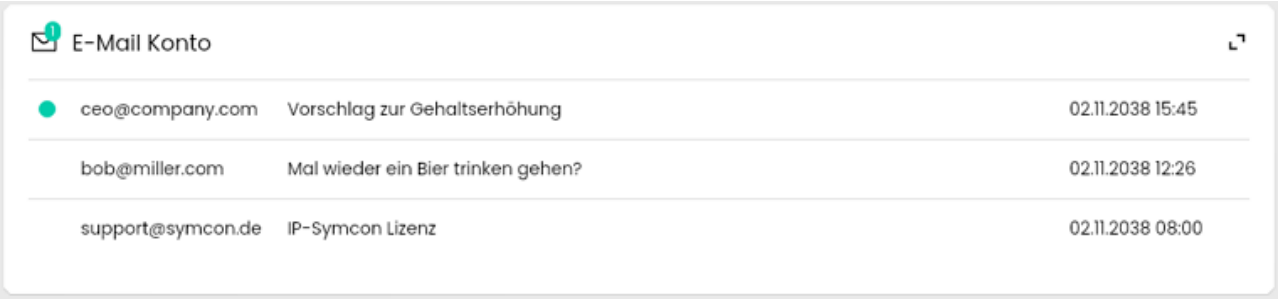
Mitteilungen visualisiert die aktuellen Mitteilungen sowie die Anzahl der ungelesenen.

### VORAUSSETZUNGEN

Mitteilungen erfordert eine Instanz des Moduls IMAP oder POP3.

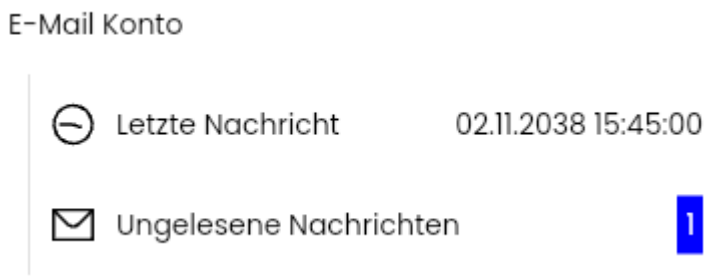
## AUSSEHEN IN KACHEL VISUALISIERUNG

### ALS EIGENE KACHEL

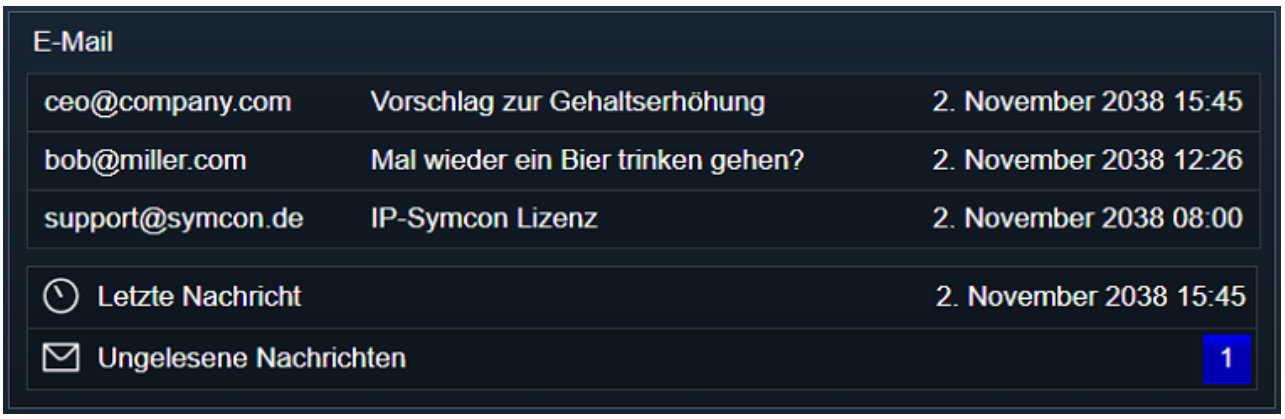



### INNERHALB EINER LISTE

In der Listenansicht fällt die Visualisierung auf die Darstellung Liste zurück.




## AUSSEHEN IM WEBFRONT


E-Mail
ceo@company.com
Vorschlag zur Gehaltserhöhung
2. November 2038 15:45
bob@miller.com
Mal wieder ein Bier trinken gehen?
2. November 2038 12:26
support@symcon.de
IP-Symcon Lizenz
2. November 2038 08:00
Letzte Nachricht
2. November 2038 15:45
Ungelesene Nachrichten
1

# Popup

Mehrere Objekte können gemeinsam in einem Popup dargestellt werden.

### VORAUSSETZUNGEN

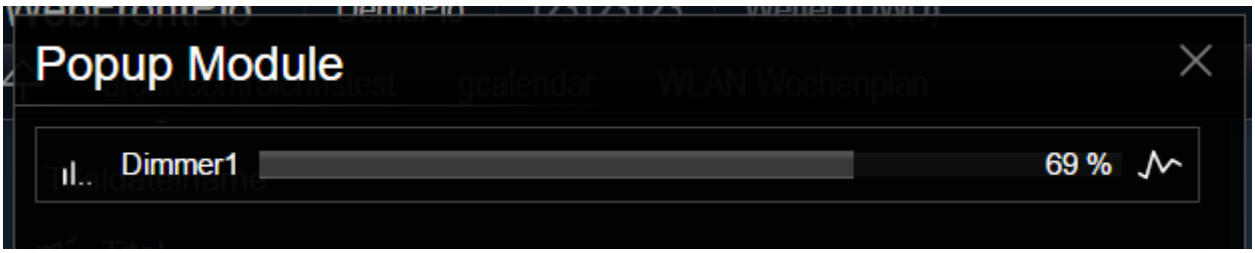
Ein Popup wird als Popup-Modul-Instanz eingerichtet. Dargestellte Objekte werden unterhalb des Popup Moduls platziert bzw. verlinkt.

## AUSSEHEN IN KACHEL VISUALISIERUNG

Eine Popup-Darstellung in der Kachel Visualisierung folgt in den kommenden Versionen, ist aktuell aber nicht vorhanden.

## AUSSEHEN IM WEBFRONT



# Schalter

Benötigt: IP-Symcon >= 5.0
Ein Schalter kann An (true) oder Aus (false) geschaltet werden.

### VORAUSSETZUNGEN

Ein Schalter ist eine Variable mit der Darstellung Schalter. Um diese Darstellung zu verwenden muss die Variable folgende Bedingungen erfüllen:
- Typ Boolean
- Eingerichtete Variablenaktion

### PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| Individuelle Icons passend zum Wert | Falls gesetzt, kann für true und false je individuelle Icons gewählt werden. Ansonsten wird ein einzelnes Icon gewählt, welches immer dargestellt wird |
| Icon | Icon welches dauerhaft auf dem Schalter angezeigt wird (nur verfügbar wenn "Individuelle Icons passend zum Wert" nicht gesetzt ist) |
| Icon für true | Icon welches auf dem Schalter angezeigt wird wenn die Variable den Wert true hat (nur verfügbar wenn "Individuelle Icons passend zum Wert" gesetzt ist) |
| Icon für false | Icon welches auf dem Schalter angezeigt wird wenn die Variable den Wert false hat (nur verfügbar wenn "Individuelle Icons passend zum Wert" gesetzt ist) |
| Farbe des Leuchtens wenn aktiv | Konfiguriert die Farbe des Leuchtens um den Schalter, wenn dieser aktiv ist, die Variable also den Wert true hat |
| Intensität des Leuchtens wenn aktiv | Konfiguriert die Intensität des Leuchtens um den Schalter, wenn dieser aktiv ist, die Variable also den Wert true hat |
| Verwendung der Variable | Verwendung der Variable in zusammengefassten Darstellungen. Der Wert wirkt sich nicht direkt auf die Darstellung der Variable auf, sondern nur auf deren Verwendung in zusammengefassten Darstellungen. |

## DARSTELLUNG ALS LEGACY-PROFIL

Verwendet eine Variable die Darstellung Legacy Profil, so wird diese als Schieberegler darstellt, falls sie zusätzlich folgende Bedingungen erfüllt:
- Kein Variablenprofil gesetzt oder Variablenprofil "~Switch" oder "~Mute"

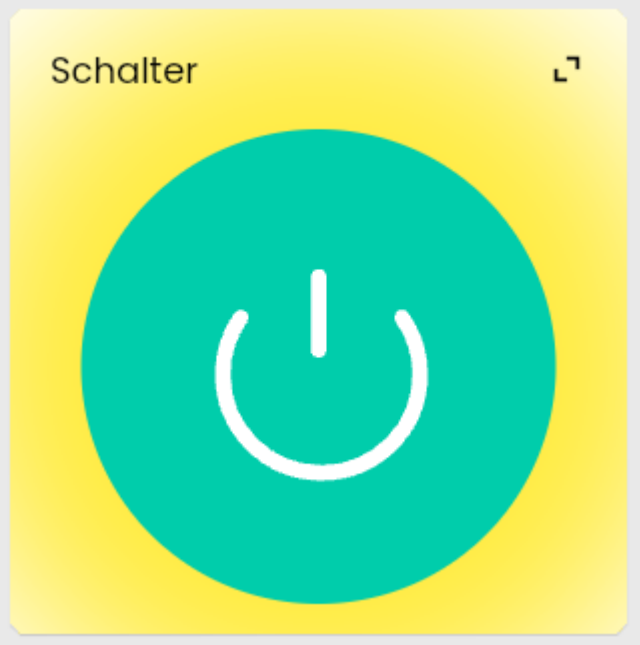
Bei der Verwendung eines Variablenprofils werden einige Parameter des Schiebereglers abgeleitet

### VERWENDUNG DER VARIABLEN

Die Verwendung hängt vom Profilnamen ab.
An/Aus für Profilnamen
~Switch
Kein Profil
Stumm schalten für Profilnamen "~Mute"

## AUSSEHEN IN KACHEL VISUALISIERUNG

### ALS EIGENE KACHEL



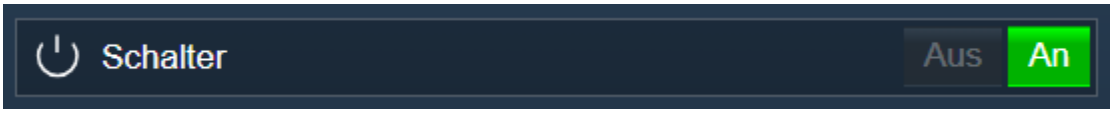

### INNERHALB EINER LISTE




## AUSSEHEN IM WEBFRONT

Falls die Darstellung über ein Legacy-Profil erfolgt, unterscheidet das WebFront ob kein Variablenprofil gesetzt ist oder das Profil "~Switch" und hat für jeden Fall eine eigene Darstellung.

### MIT DARSTELLUNG "SCHALTER" ODER PROFIL ~SWITCH



**OHNE PROFIL**



# Schieberegler

Bei einem Schieberegler kann ein Wert eingestellt werden.

## VORAUSSETZUNGEN

Ein Schieberegler ist eine Variable mit der Darstellung Schieberegler. Um diese Darstellung zu verwenden muss die Variable folgende Bedingungen erfüllen:

- Typ Integer oder Float
- Eingerichtete Variablenaktion

## PARAMETER

| PARAMETER | BESCHREIBUNG |
| --- | --- |
| Minimaler Wert | Minimaler Wert des Schiebereglers |
| Maximaler Wert | Maximaler Wert des Schiebereglers |
| Schrittweite | Größe eines einzelnen Schrittes im Schieberegler. Bei 0 hat der Schieberegler keine feste Schrittweite. |
| Gradient | Definiert die Optik des Gradienten |
| Benutzerdefinierter Gradient | In der Liste können für spezifische Werte Farben ausgewählt werden. Der Gradient des Schiebereglers besteht dann aus einem Verlauf über die definierten Farben. (nur sichtbar und verwendbar falls Gradient = "Benutzerdefiniert") |
| Verwendung der Variable | Verwendung der Variable in zusammengefassten Darstellungen. Der Wert wirkt sich nicht direkt auf die Darstellung der Variable auf, sondern nur auf deren Verwendung in zusammengefassten Darstellungen. |
| Präfix | Wird bei Formatierung vor den Variablenwert geschrieben |
| Suffix | Wird bei Formatierung hinter den Variablenwert geschrieben |
| Anzeigetyp | Bei "Prozentual" werden die Werte prozentual dargestellt. Somit wird der Minimalwert als 0 und der Maximalwert als 100 dargestellt, unabhängig von deren absoluten Werten. Andere Werte werden ebenfalls dementsprechend skaliert. Bei "Absolut" erfolgt keine Umrechnung. |
| Tausendertrennzeichen | Definiert das verwendete Tausendertrennzeichen bei Formatierung des Wertes |
| Nachkommastellen | Die Anzahl der Nachkommastellen, welche bei Formatierung des Wertes dargestellt werden (nur verfügbar für Variablen des Typs Float) |
| Dezimaltrennzeichen | Definiert das verwendete Dezimaltrennzeichen bei Formatierung des Wertes (nur verfügbar für Variablen des Typs Float) |
| Icon | Das Icon, welches für die Variable verwendet wird |
| Verwende aktualisierte Parameter für spezifische Intervalle | Aktiviert eine Liste von Intervallen, welche die Formatierung in spezifischen Wertbereichen überschreiben können (nicht verwendbar falls Anzeigetyp = "Prozentual") |
| Intervalle | Eine Liste von Intervallen zum Überschreiben der Formatierung in spezifischen Wertbereichen |

Intervalle

| PARAMETER | BESCHREIBUNG |
|---|---|
| Anfang des Intervalls | Ab diesem Wert wird das Intervall angewandt |
| Ende des Intervalls | Bis zu diesem Wert wird das Intervall angewandt. Sollte es ein Intervall mit Anfang auf diesem Wert geben, so wird bei dem exakten Wert für Ende das andere Intervall verwendet und das definierte Intervall nur für Werte genutzt, welche kleiner als das Ende sind. |
| Anzeige | Bei "Formatierter Wert" wird der Wert mit angepassten Parametern formatiert, bei "Konstante" wird die Darstellung durch einen konstanten Wert ersetzt |
| Konstante | Definiert den Text, welcher ohne weitere Formatierung für Werte innerhalb dieses Intervalls dargestellt werden soll. (nur verfügbar falls Anzeige = "Konstante") |
| Umrechnungsfaktor | Vor der Formatierung wird der Wert durch den Umrechnungsfaktor geteilt (nur verfügbar falls Anzeige = "Formatierter Wert") |
| Überschreibe Präfix | Ist der Schalter aktiviert wird nicht der Präfix der Hauptkonfiguration verwendet, sondern der im Intervall definierte Präfix (nur verfügbar falls Anzeige = "Formatierter Wert") |
| Präfix | Verwendeter Präfix innerhalb des Intervalls (nur verfügbar falls Schalter "Überschreibe Präfix" aktiv) |
| Überschreibe Suffix | Ist der Schalter aktiviert wird nicht der Suffix der Hauptkonfiguration verwendet, sondern der im Intervall definierte Suffix (nur verfügbar falls Anzeige = "Formatierter Wert") |
| Suffix | Verwendeter Suffix innerhalb des Intervalls (nur verfügbar falls Schalter "Überschreibe Suffix" aktiv) |
| Überschreibe Nachkommastellen | Ist der Schalter aktiviert werden nicht die Nachkommastellen der Hauptkonfiguration verwendet, sondern die im Intervall definierten Nachkommastellen (nur verfügbar falls Anzeige = "Formatierter Wert") |
| Nachkommastellen | Verwendete Nachkommastellen innerhalb des Intervalls (nur verfügbar falls Schalter "Überschreibe Nachkommastellen" aktiv) |
| Überschreibe Icon | Ist der Schalter aktiviert wird nicht das Icon der Hauptkonfiguration verwendet, sondern das im Intervall definierte Icon |
| Icon | Verwendetes Icon innerhalb des Intervalls (nur verfügbar falls Schalter "Überschreibe Icon" aktiv) |

## DARSTELLUNG ALS LEGACY-PROFIL

Verwendet eine Variable die Darstellung Legacy Profil, so wird diese als Schieberegler darstellt, falls sie zusätzlich folgende Bedingungen erfüllt:

Erfüllt nicht die Bedingungen von einer der folgenden Darstellungen:

- Jalousie
- Aufzählung

Variablenprofil mit folgenden Parametern:

- Max Wert > Min Wert
- Schrittweite beliebig, hat aber Einfluss auf die Schrittweite des Sliders

Bei der Verwendung eines Variablenprofils werden einige Parameter des Schiebereglers abgeleitet

### GRADIENT

Der Gradient hängt vom Profilnamen ab.
  Temparatur für Profilnamen
    ~Temperature
    ~Temperature.Difference
    ~Temperature.Fahrenheit
    ~Temperature.Room
  Farbtemperatur für Profilnamen "~TWColor"
  Standard sonst

### VERWENDUNG DER VARIABLEN

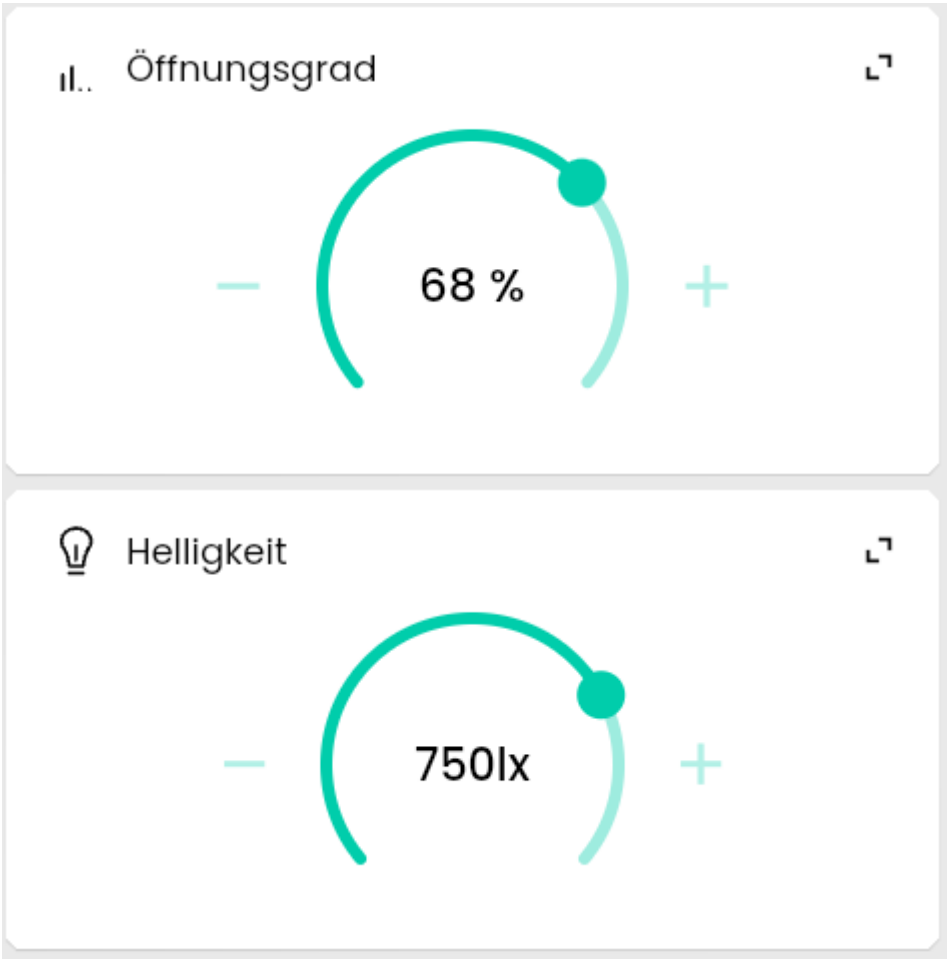
Die Verwendung hängt vom Profilnamen ab.
  Temparatur für Profilnamen
    ~Temperature
    ~Temperature.Difference
    ~Temperature.Fahrenheit
    ~Temperature.Room
  Farbtemperatur für Profilnamen "~TWColor"
  Intensität falls Profilname mit "~Intensity." beginnt
  Lautstärke für Profilnamen "~Volume"
  Fortschritt für Profilnamen "~Progress"
  Keine hiervon sonst

### ANZEIGETYP

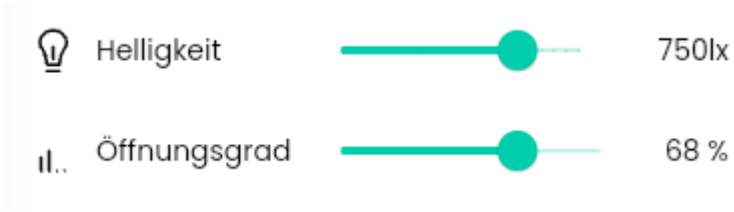
"Prozentual" falls Suffix "%", sonst "Absolut"

## AUSSEHEN IN KACHEL VISUALISIERUNG

### ALS EIGENE KACHEL



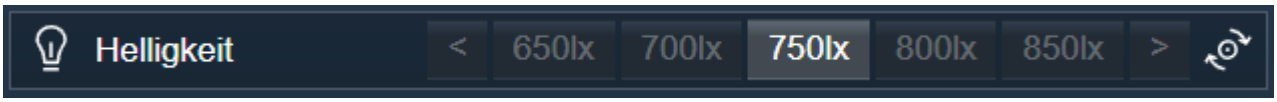

### INNERHALB EINER LISTE




## AUSSEHEN IM WEBFRONT

Das WebFront unterscheidet je nach Anzeigetyp und hat für jeden Fall eine eigene Darstellung:

### ANZEIGETYP "ABSOLUT"




### ANZEIGETYP "PROZENTUAL"


Öffnungsgrad
68 %

# Stream

Streams stellen die Videoausgabe, beispielsweise von einer Überwachungskamera, dar.

### VORAUSSETZUNGEN

Ein Stream ist ein Medium vom Typ Stream.

## AUSSEHEN IN KACHEL VISUALISIERUNG

### ALS EIGENE KACHEL




### INNERHALB EINER LISTE




## AUSSEHEN IM WEBFRONT

Kamera



# Szenen

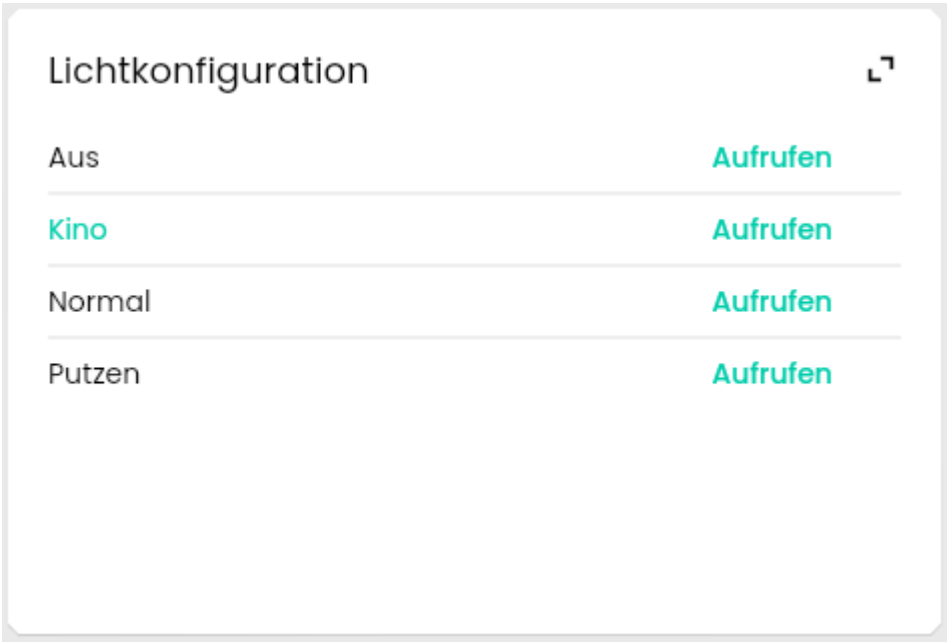
Benötigt: IP-Symcon >= 7.0
Verschiedene Geräteeinstellungen können gemeinsam als Szene abgespeichert und wieder aufgerufen werden.

### VORAUSSETZUNGEN

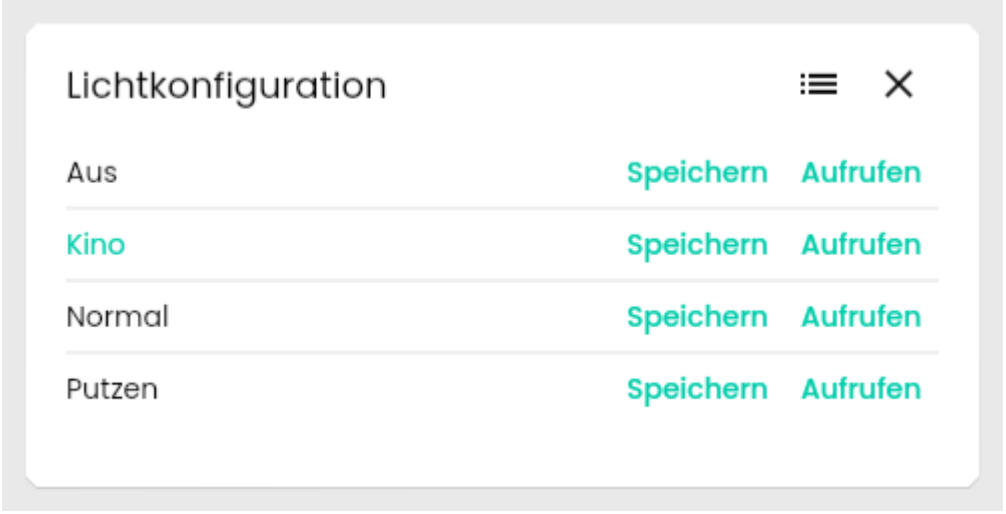
Szenen erfordern eine Instanz des Moduls SzenenSteuerung.

## AUSSEHEN IN KACHEL VISUALISIERUNG

### ALS EIGENE KACHEL

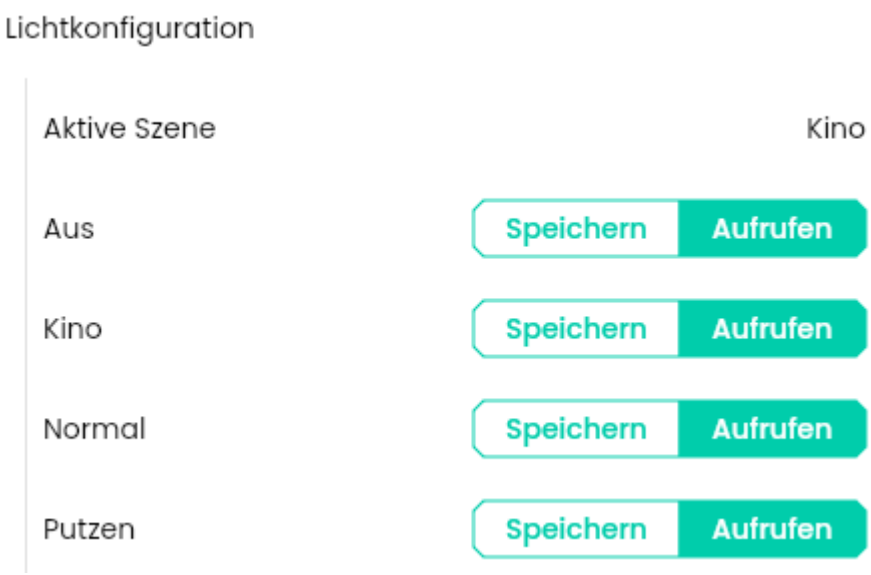



### IM VOLLBILD



### INNERHALB EINER LISTE

In der Listenansicht fällt die Visualisierung auf die Darstellung Liste zurück.




## AUSSEHEN IM WEBFRONT

Diese Darstellung wird nicht vom WebFront unterstützt. Stattdessen wir die Instanz als Liste dargestellt.

# Thermostat

Ein Thermostat ermöglicht die Einstellung einer gewünschten Temperatur kombiniert mit einer Anzeige der aktuellen Temperatur.

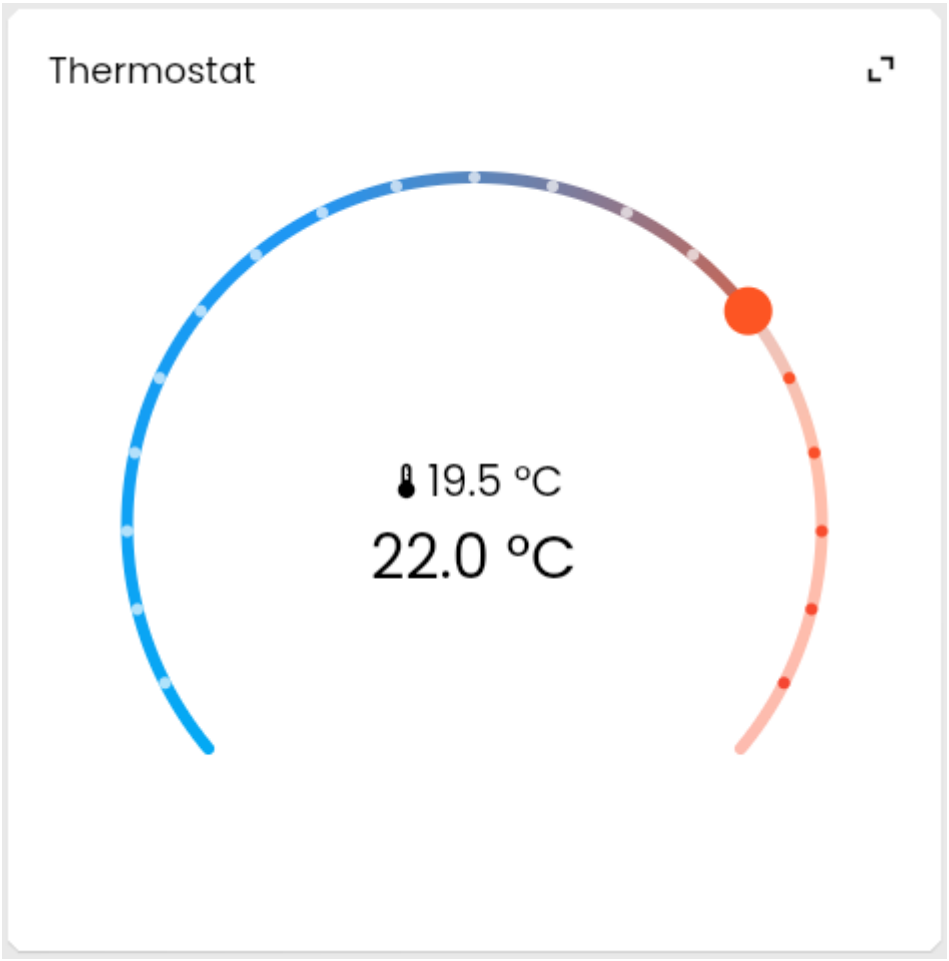
### VORAUSSETZUNGEN

Eine Instanz kann als Thermostat dargestellt werden, wenn folgende untergeordnete Objekte vorhanden sind:

- Soll Temperatur:
  - Verwendet die Variablendarstellung Schieberegler
  - Verwendung der Variablen ist "Temperatur"
- Ist Temperatur:
  - Variable vom Typ Float
  - keine eingerichtete Variablenaktion
  - Verwendet dieVariablendarstellung Legacy Profil:
    - ~Temperature
    - ~Temperature.Difference
    - ~Temperature.Fahrenheit
    - ~Temperature.Room

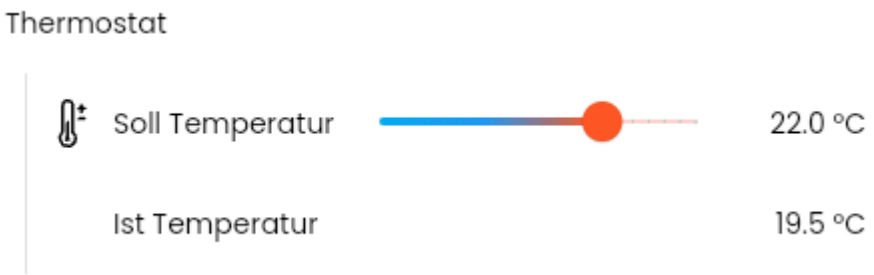
## AUSSEHEN IN KACHEL VISUALISIERUNG

### ALS EIGENE KACHEL




### INNERHALB EINER LISTE

In der Listenansicht fällt die Visualisierung auf die Darstellung Liste zurück.




## AUSSEHEN IM WEBFRONT

Diese Darstellung wird nicht vom WebFront unterstützt. Stattdessen wir die Instanz als Liste dargestellt.

# Ton

In der Visualisierung können Audiodateien eingebunden und wiedergegeben werden.

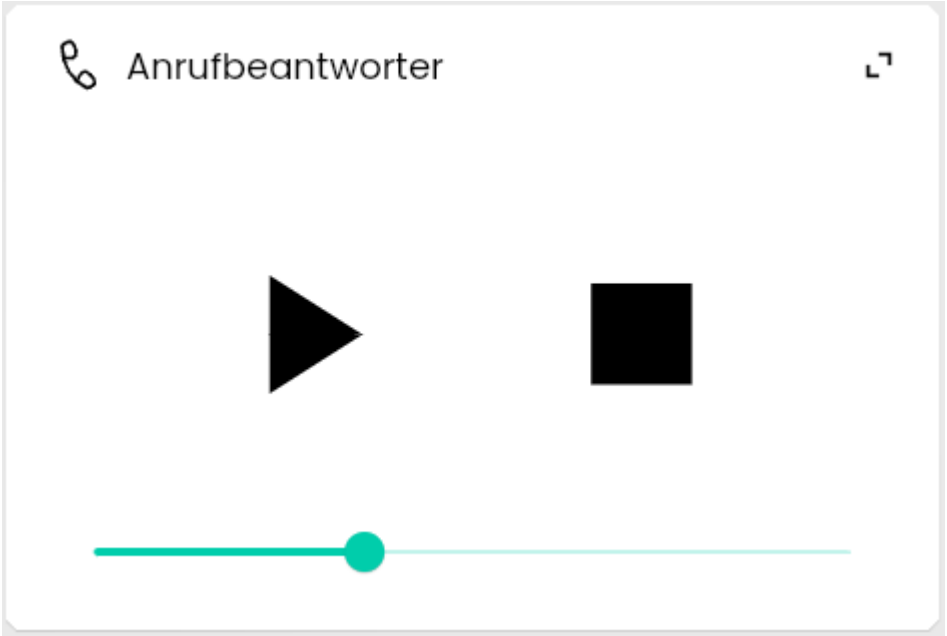
### VORAUSSETZUNGEN

Ein Ton ist ein Medium vom Typ Ton.

## AUSSEHEN IN KACHEL VISUALISIERUNG

### ALS EIGENE KACHEL

Die Fortschrittleiste und der Stop-Button können je nach Größe der Kachel wegfallen.




### INNERHALB EINER LISTE




## AUSSEHEN IM WEBFRONT

Die Darstellung kann je nach Browser variieren.



# Webinhalt

Ein Webinhalt kann beliebige HTML-Inhalte darstellen und bietet somit Experten die Möglichkeit individuelle Elemente darzustellen. Durch CSS wird Text innerhalb der Darstellung analog zum Rest der Visualisierung dargestellt. Außerdem kann die Adresse einer beliebigen Internetseite angegeben werden im sie ein der Visualisierung darzustellen.

### VORAUSSETZUNGEN

Ein Webinhalt ist eine Variable mit der Darstellung Webinhalt. Um diese Darstellung zu verwenden muss die Variable folgende Bedingungen erfüllen:
keine eingerichtete Variablenaktion
Typ String

## PARAMETER

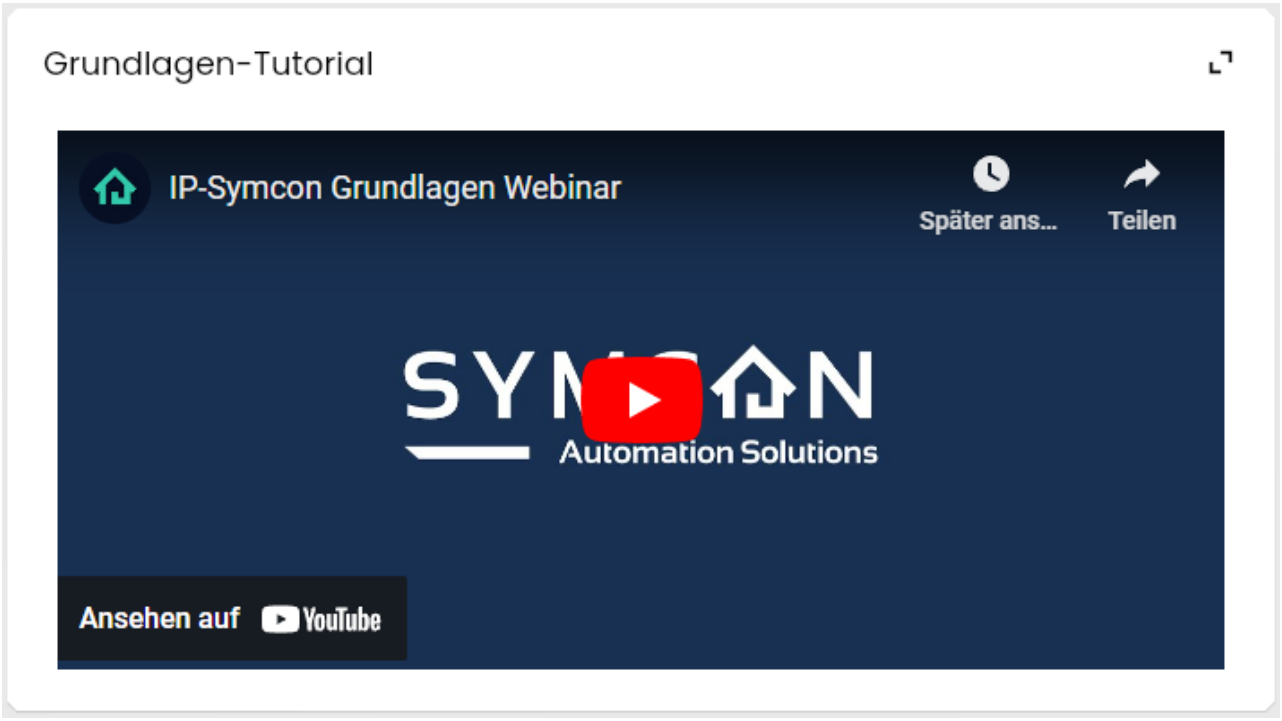
| PARAMETER | BESCHREIBUNG |
|---|---|
| Darstellungstyp | Bei "HTML Inhalt" wird der Variablenwert als HTML interpretiert und dargestellt. Bei "Webseite" kann die Adresse einer Website angegeben werden, solange diese das Einbetten nicht blockiert. |
| Ränder entfernen | Wenn aktiviert, werden die Ränder der Kachel entfernt. |

## DARSTELLUNG ALS LEGACY-PROFIL

Verwendet eine Variable die Darstellung Legacy Profil, so wird diese als Webinhalt darstellt, falls sie zusätzlich folgende Bedingungen erfüllt:
Ausgewähltes Variablenprofil ~HTMLBox

## AUSSEHEN IN KACHEL VISUALISIERUNG

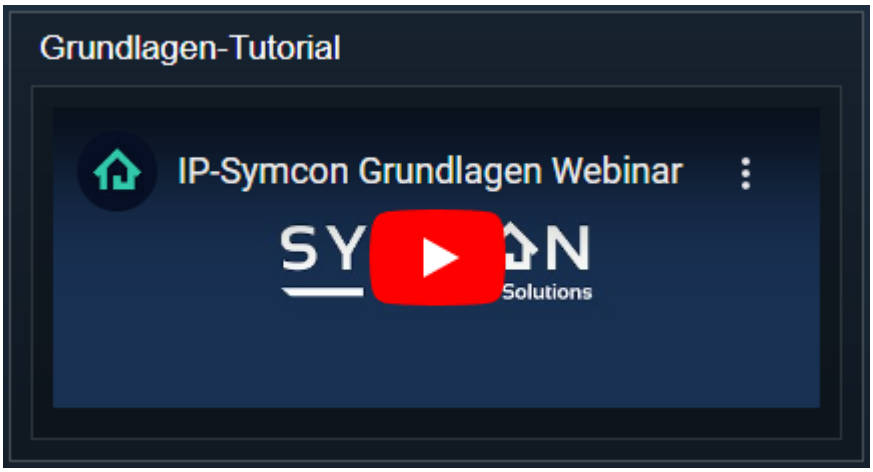
### ALS EIGENE KACHEL




### INNERHALB EINER LISTE




## AUSSEHEN IM WEBFRONT



# Wertanzeige

Zeigt einen Wert optional mit Hintergrundfarbe oder Werteverlauf an.

## VORAUSSETZUNGEN

Eine Wertanzeige ist eine Variable mit der Darstellung Wertanzeige. Um diese Darstellung zu verwenden muss die Variable folgende Bedingungen erfüllen:
Keine eingerichtete Variablenaktion

## PARAMETER

Je nach Variablentyp (Boolean, Integer, Float, String) stehen verschiedene Parameter zur Verfügung. Die folgenden gelten für alle.

| PARAMETER | BESCHREIBUNG |
|---|---|
| Standardicon | Das Icon, welches angezeigt wird. Es kann durch weitere Parameter überschrieben werden |
| Standardfarbe | Die Farbe, mit welcher der Wert hinterlegt wird |
| Präfix | Wird bei Formatierung vor den Variablenwert geschrieben (nicht verfügbar für Boolean) |
| Suffix | Wird bei Formatierung hinter den Variablenwert geschrieben (nicht verfügbar für Boolean) |

### FLOAT UND INTEGER

| PARAMETER | BESCHREIBUNG |
| --- | --- |
| Verwendung der Variable | Verwendung der Variable in zusammengefassten Darstellungen. Der Wert wirkt sich nicht direkt auf die Darstellung der Variable auf, sondern nur auf deren Verwendung in zusammengefassten Darstellungen. |
| Anzeigetyp | Bei "Prozentual" werden die Werte prozentual dargestellt. Somit wird der Minimalwert als 0 und der Maximalwert als 100 dargestellt, unabhängig von deren absoluten Werten. Andere Werte werden ebenfalls dementsprechend skaliert. Bei "Absolut" erfolgt keine Umrechnung. |
| Tausendertrennzeichen | Definiert das verwendete Tausendertrennzeichen bei Formatierung des Wertes |
| Minimaler Wert | Minimaler Wert der Variable (nur verwendbar falls Anzeigetyp = "Prozentual") |
| Maximaler Wert | Maximaler Wert der Variable (nur verwendbar falls Anzeigetyp = "Prozentual") |
| Nachkommastellen | Die Anzahl der Nachkommastellen, welche bei Formatierung des Wertes dargestellt werden (nur verfügbar für Variablen des Typs Float) |
| Dezimaltrennzeichen | Definiert das verwendete Dezimaltrennzeichen bei Formatierung des Wertes (nur verfügbar für Variablen des Typs Float) |
| Icon | Das Icon, welches für die Variable verwendet wird |
| Verwende aktualisierte Parameter für spezifische Intervalle | Aktiviert eine Liste von Intervallen, welche die Formatierung in spezifischen Wertbereichen überschreiben können (nicht verwendbar falls Anzeigetyp = "Prozentual") |
| Intervalle | Eine Liste von Intervallen zum Überschreiben der Formatierung in spezifischen Wertbereichen |

Intervalle

| PARAMETER | BESCHREIBUNG |
|---|---|
| Anfang des Intervalls | Ab diesem Wert wird das Intervall angewandt |
| Ende des Intervalls | Bis zu diesem Wert wird das Intervall angewandt. Sollte es ein Intervall mit Anfang auf diesem Wert geben, so wird bei dem exakten Wert für Ende das andere Intervall verwendet und das definierte Intervall nur für Werte genutzt, welche kleiner als das Ende sind. |
| Anzeige | Bei "Formatierter Wert" wird der Wert mit angepassten Parametern formatiert, bei "Konstante" wird die Darstellung durch einen konstanten Wert ersetzt |
| Konstante | Definiert den Text, welcher ohne weitere Formatierung für Werte innerhalb dieses Intervalls dargestellt werden soll. (nur verfügbar falls Anzeige = "Konstante") |
| Umrechnungsfaktor | Vor der Formatierung wird der Wert durch den Umrechnungsfaktor geteilt (nur verfügbar falls Anzeige = "Formatierter Wert") |
| Überschreibe Präfix | Ist der Schalter aktiviert wird nicht der Präfix der Hauptkonfiguration verwendet, sondern der im Intervall definierte Präfix (nur verfügbar falls Anzeige = "Formatierter Wert") |
| Präfix | Verwendeter Präfix innerhalb des Intervalls (nur verfügbar falls Schalter "Überschreibe Präfix" aktiv) |
| Überschreibe Suffix | Ist der Schalter aktiviert wird nicht der Suffix der Hauptkonfiguration verwendet, sondern der im Intervall definierte Suffix (nur verfügbar falls Anzeige = "Formatierter Wert") |
| Suffix | Verwendeter Suffix innerhalb des Intervalls (nur verfügbar falls Schalter "Überschreibe Suffix" aktiv) |
| Überschreibe Nachkommastellen | Ist der Schalter aktiviert werden nicht die Nachkommastellen der Hauptkonfiguration verwendet, sondern die im Intervall definierten Nachkommastellen (nur verfügbar falls Anzeige = "Formatierter Wert" und Variable des Typs Float) |
| Nachkommastellen | Verwendete Nachkommastellen innerhalb des Intervalls (nur verfügbar falls Schalter "Überschreibe Nachkommastellen" aktiv) |
| Überschreibe Icon | Ist der Schalter aktiviert wird nicht das Icon der Hauptkonfiguration verwendet, sondern das im Intervall definierte Icon |
| Icon | Verwendetes Icon innerhalb des Intervalls (nur verfügbar falls Schalter "Überschreibe Icon" aktiv) |
| Überschreibe Farbe | Ist der Schalter aktiviert wird nicht die Farbe der Hauptkonfiguration verwendet, sondern die im Intervall definierte Farbe |
| Farbe | Verwendete Farbe innerhalb des Intervalls (nur verfügbar falls Schalter "Überschreibe Farbe" aktiv) |

## BOOLEAN UND STRING

| PARAMETER | BESCHREIBUNG |
|---|---|
| Mehrere Zeilen | (Nur String) Stellt den Variablenwert in mehreren Zeilen dar |
| Optionen | Die einzelnen Optionen der Darstellung, welche es erlauben für verschiedene Variablenwerte bestimmte Texte, Farben und Icons anzugeigen |

Optionen

| PARAMETER | BESCHREIBUNG |
|---|---|
| Wert | Variablenwert für den diese Option gilt |
| Beschriftung | Text, der für diese Option angezeigt wird |
| Überschreibe Icon | Falls gesetzt, wird das Standardicon dieser Darstellung überschrieben, falls die Variable den Wert dieser Option hat |
| Icon | Icon für die Variable, falls die Variable den Wert dieser Option hat (nur verfügbar wenn "Icon überschreiben" gesetzt) |
| Überschreibe Farbe | Falls gesetzt, wird die Standardfarbe dieser Darstellung überschrieben, falls die Variable den Wert dieser Option hat |
| Farbe | Farbe mit welcher die Option in der Aufzählung dargestellt wird |

### WERTEVERLAUF

Ist das Logging für die Variable aktiviert, wird eine Vorschau des Werteverlaufs angezeigt.

## DARSTELLUNG ALS LEGACY-PROFIL

Verwendet eine Variable die Darstellung Legacy Profil, so wird diese als Wertanzeige darstellt, falls sie zusätzlich folgende Bedingungen erfüllt:
Erfüllt nicht die Bedingungen von einer der folgenden Darstellungen:
- Datum/Uhrzeit
- Farbanzeige
- Mehrzeilige Textbox
- HTML Box

Bei der Verwendung eines Variablenprofils werden einige Parameter des Wertanzeige abgeleitet

### VERWENDUNG DER VARIABLEN

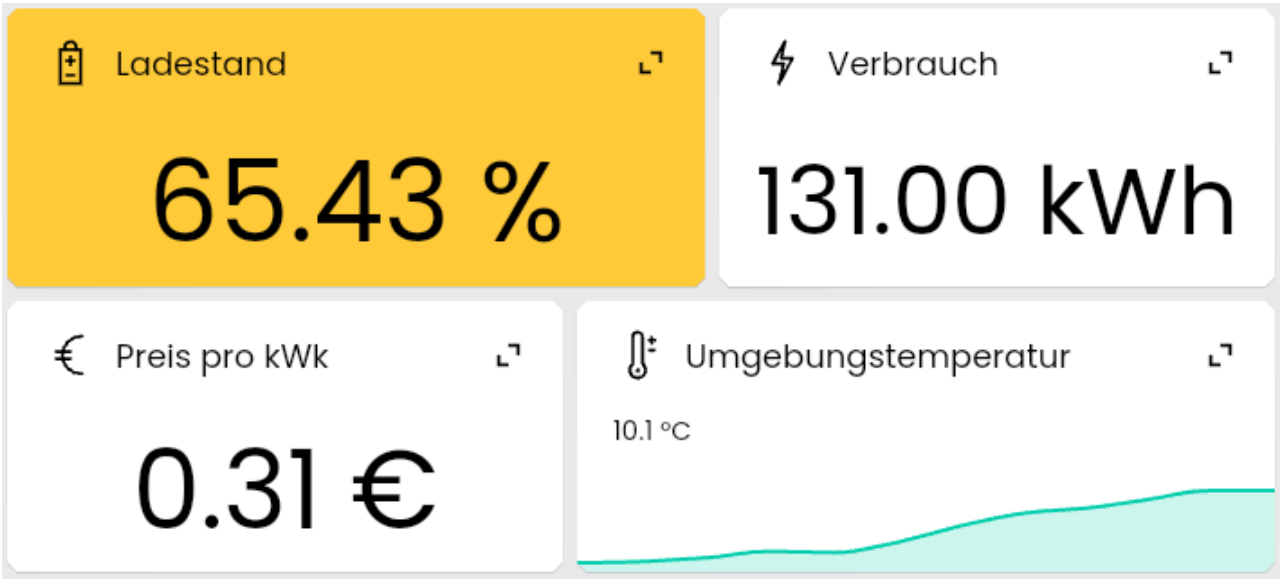
Die Verwendung hängt vom Profilnamen ab.
Temparatur für Profilnamen
- ~Temperature
- ~Temperature.Difference
- ~Temperature.Fahrenheit
- ~Temperature.Room

### ANZEIGETYP

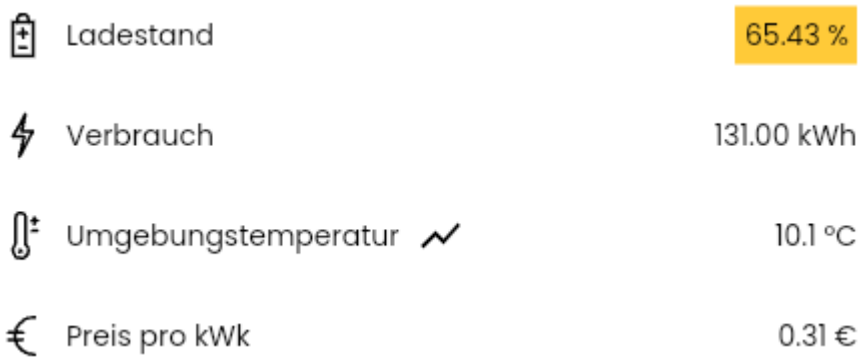
"Prozentual" falls Suffix "%", sonst "Absolut"

## AUSSEHEN IN KACHEL VISUALISIERUNG

### ALS EIGENE KACHEL

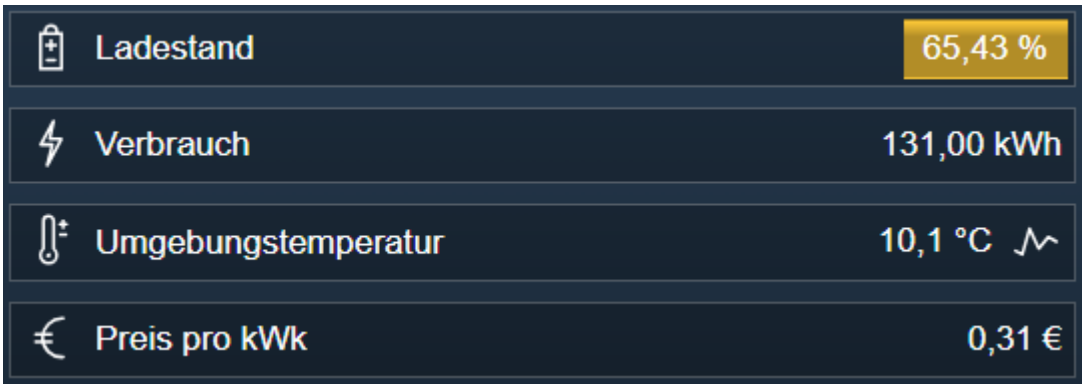



### INNERHALB EINER LISTE




## AUSSEHEN IM WEBFRONT

Im WebFront wird keine Vorschau des Werteverlaufs angezeigt.



# Werteingabe

Benötigt: IP-Symcon >= 5.0
Eine Werteingabe ermöglicht das verändern von Werten als Freitext.

## VORAUSSETZUNGEN

Eine Werteingabe ist eine Variable mit der Darstellung Werteingabe. Um diese Darstellung zu verwenden muss die Variable folgende Bedingungen erfüllen:
- Eingerichtete Variablenaktion
- Variablentyp Integer, Float oder String

### PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| Präfix | Wird bei Formatierung vor den Variablenwert geschrieben |
| Suffix | Wird bei Formatierung hinter den Variablenwert geschrieben |
| Mehrzeilige Eingabe | Falls aktiv, ist eine Eingabe über mehrere Zeilen möglich, ansonsten ist die Eingabe auf eine einzelne Zeile eingeschränkt (nur verfügbar für Variablen des Typs String) |



Ist die Variable numerisch, werden Eingaben, die keine Zahlen sind, herausgefiltert.

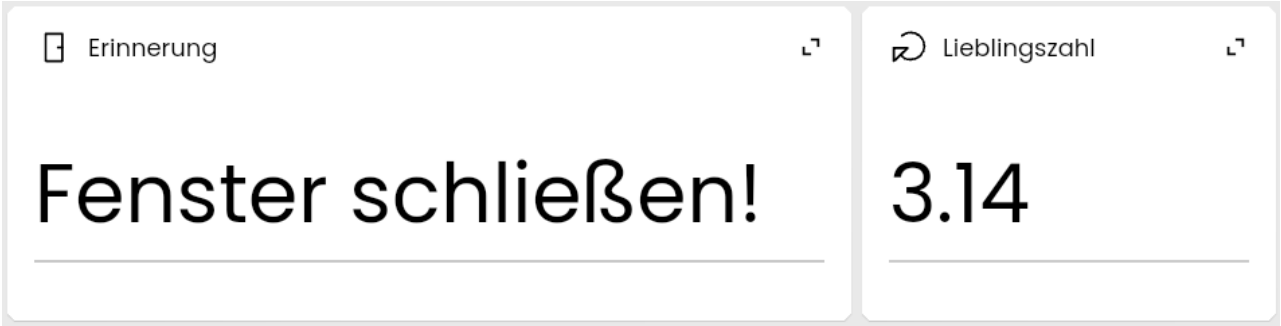
## DARSTELLUNG ALS LEGACY-PROFIL

Verwendet eine Variable die Darstellung Legacy Profil, so wird diese als Schieberegler darstellt, falls sie zusätzlich folgende Bedingungen erfüllt:
- kein Variablenprofil oder Variablenprofil "~TextBox"

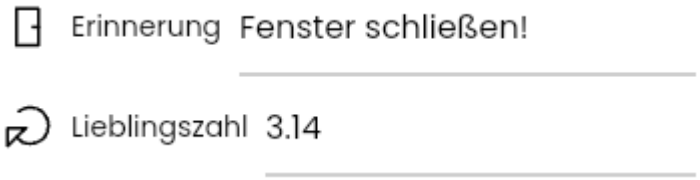
Bei der Verwendung eines Variablenprofils ist der Parameter "Mehrzeilige Eingabe" für das Variablenprofil "~TextBox" aktiviert, ansonsten deaktiviert.

## AUSSEHEN IN KACHEL VISUALISIERUNG

### ALS EIGENE KACHEL




### INNERHALB EINER LISTE




## AUSSEHEN IM WEBFRONT



# Wochenplan

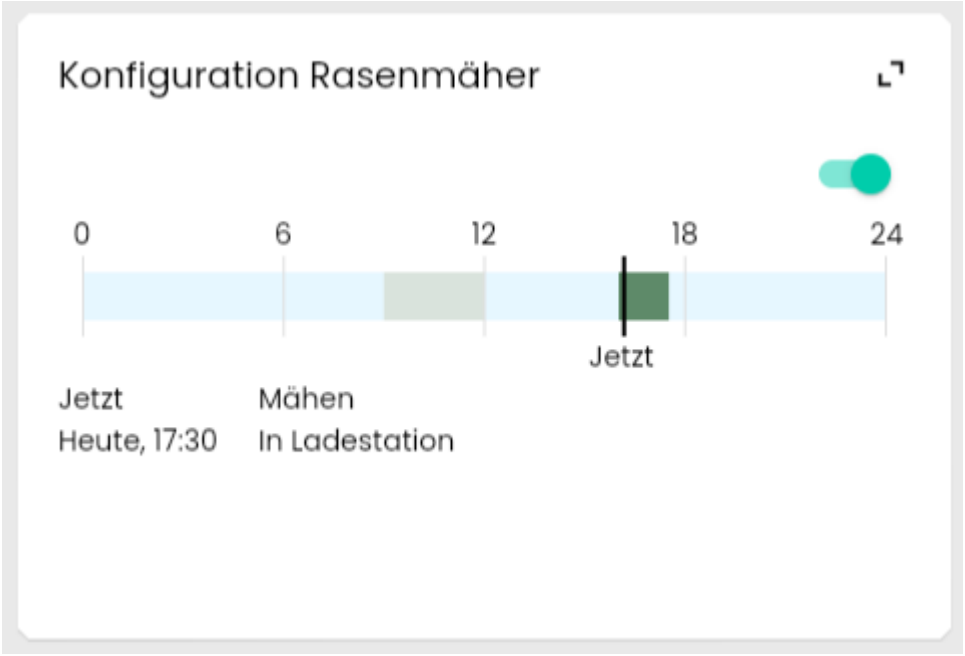
Ein Wochenplan kann in der Visualisierung betrachtet und konfiguriert werden.

### VORAUSSETZUNGEN

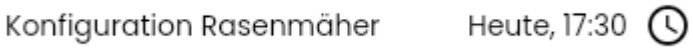
Ein Wochenplan ist ein Ereignis vom Typ Wochenplan.

## AUSSEHEN IN KACHEL VISUALISIERUNG

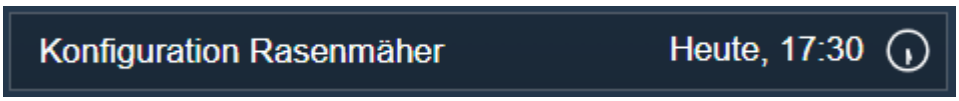
### ALS EIGENE KACHEL




### INNERHALB EINER LISTE




## AUSSEHEN IM WEBFRONT



# Zyklisches Ereignis

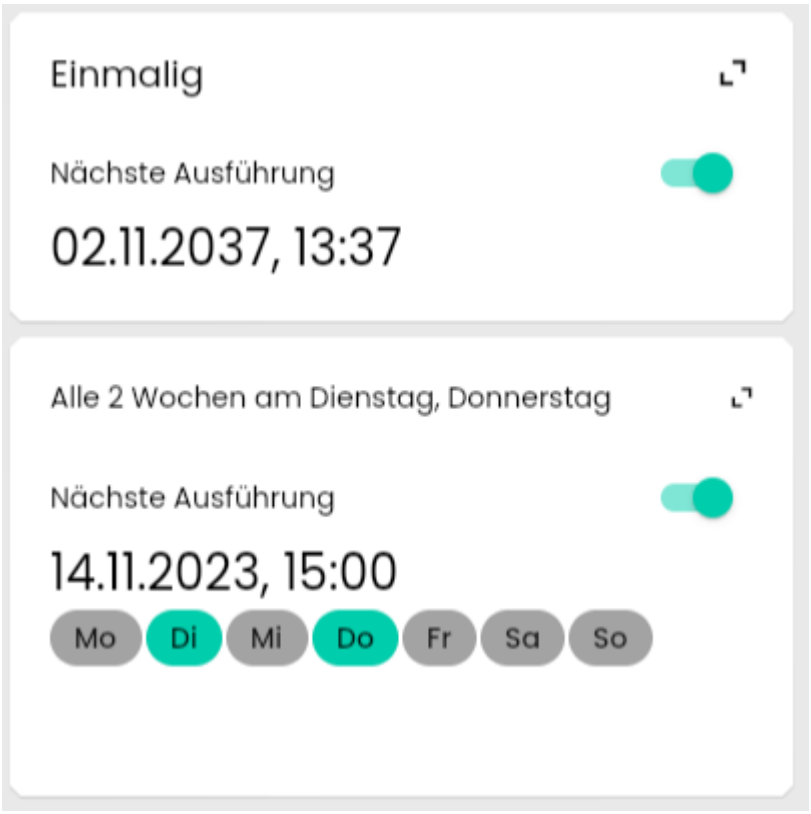
Ein zyklisches Ereignis kann in der Visualisierung betrachtet und konfiguriert werden.

### VORAUSSETZUNGEN

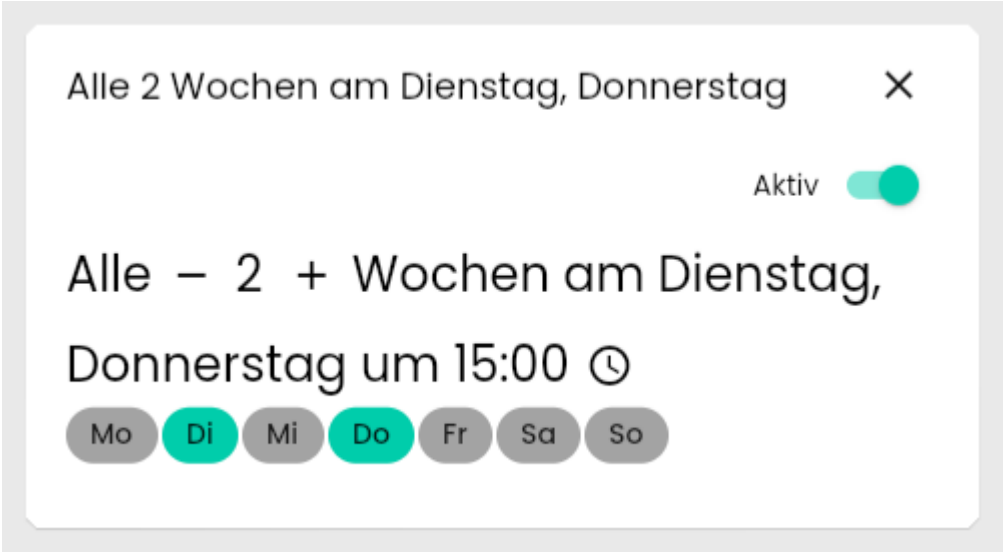
Ein zyklisches Ereignis ist ein Ereignis vom Typ Zyklisch.

## AUSSEHEN IN KACHEL VISUALISIERUNG

### ALS EIGENE KACHEL




### IM VOLLBILD




### INNERHALB EINER LISTE




## AUSSEHEN IM WEBFRONT



# Icons

Jedes Objekt in Symcon kann ein Icon besitzen. Zusätzlich können Icons innerhalb von Variablendarstellungen vorhanden sein oder durch verlinkte Strukturen (siehe Links) vererbt werden.
Icons werden nach folgender Reihenfolge priorisiert gesetzt:

Das dem Link-Objekt zugewiesene Icon (bei Verlinkung)
Das dem (verlinkten) Objekt zugewiesene Icon
Objekttypabhängig:
- Bei Automationen: das Icon *caret-right*
- Bei Medien: das Icon *image*
- Bei Variablen: Icon basierend auf der Darstellung

Icons, die anhand der Variablendarstellung gesetzt werden, werden automatisch bei Variablenwertänderung aktualisiert.
Wenn ein vererbtes Icon explizit entfernt werden soll, so kann das spezielle Icon *Transparent* ausgewählt werden.

## ÜBERSICHT DER ICONS

In der Kachelvisualisierung können die mehr als 3000 klassischen Icons von Font Awesome verwendet werden: Verwendbare Icons. Diese Icons werden im Light-Stil dargestellt. Außerdem sind alle Brand-Icons verwendbar. Schließlich stehen noch eine Reihe von Icons, die speziell für die Verwendung in Symcon erstellt wurden, zur Auswahl:

**heatpump**

**marquee-closed**

**marquee-half**

**marquee-open**

**terrace-door-closed**

**terrace-door-open**

**terrace-door-tilted**

**volant-closed**

**volant-half**

**volant-open**

**window-left-closed-right-closed**

**window-left-closed-right-open**

**window-left-closed-right-tilted**

**window-left-closed**

**window-left-open-right-closed**

**window-left-open-right-open**

**window-left-open-right-tilted**

**window-left-open**

**window-left-tilted-right-closed**

**window-left-tilted-right-open**

**window-left-tilted-right-tilted**

**window-left-tilted**

**window-right-closed**

**window-right-open**

**window-right-tilted**

**window-roof-closed**

**window-roof-open**

Im WebFront können die Legacy-Icons verwendet werden.

## DAS ICON *TRANSPARENT*

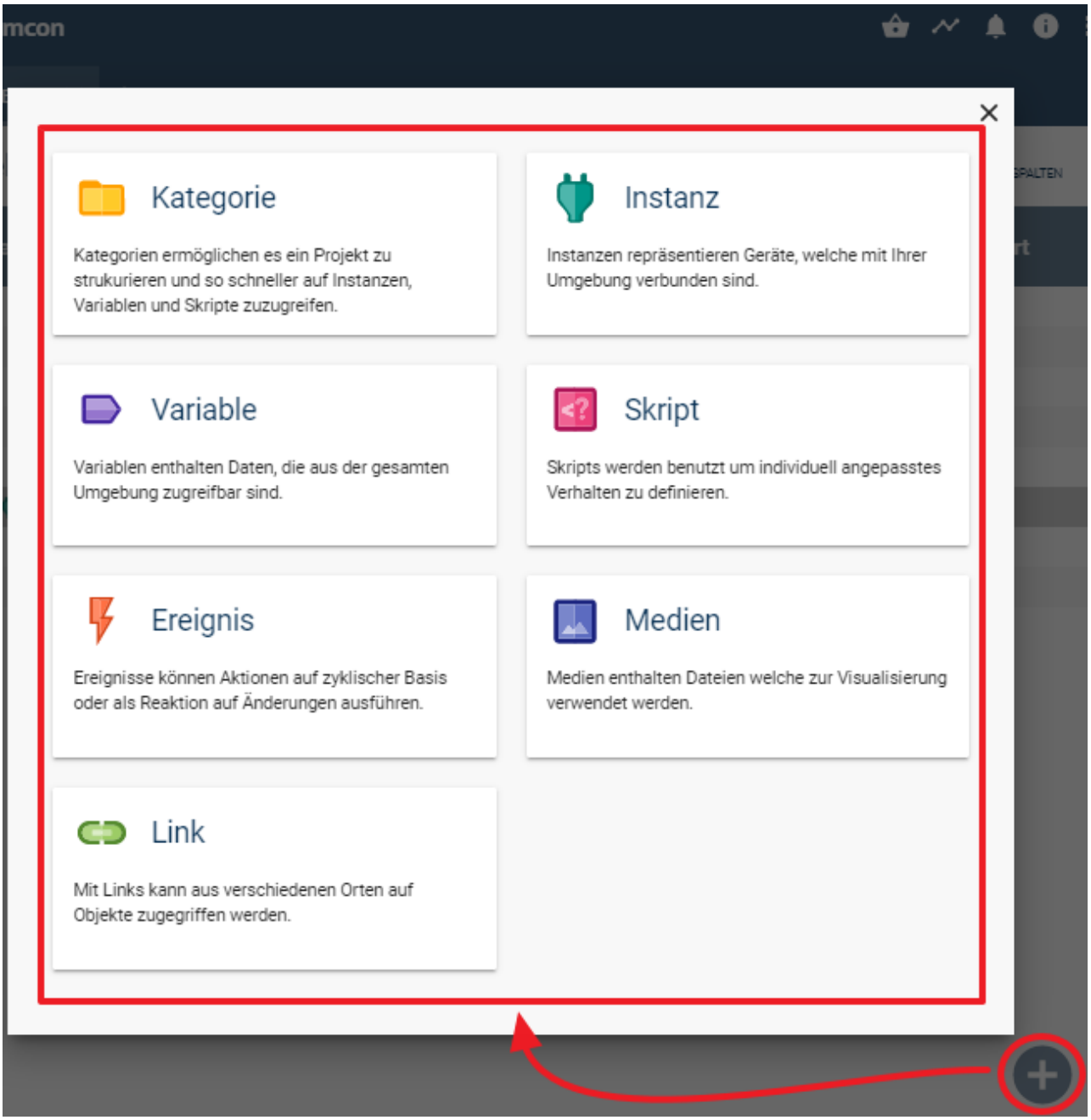
Das Icon *Transparent* ist ein Sonderfall. Wird dieses Icon ausgewählt, so kann ein anderes Icon, welches niedriger in der Hierarchie ist (s.o.), überschrieben werden, so dass kein Icon angezeigt wird. Dies kann als Stilelement verwendet werden, um z.B. bei verlinken Objektkategorien kein Icon anzuzeigen, obwohl die Kategorie ein Icon zugewiesen hatte.

# Vorgehensweisen

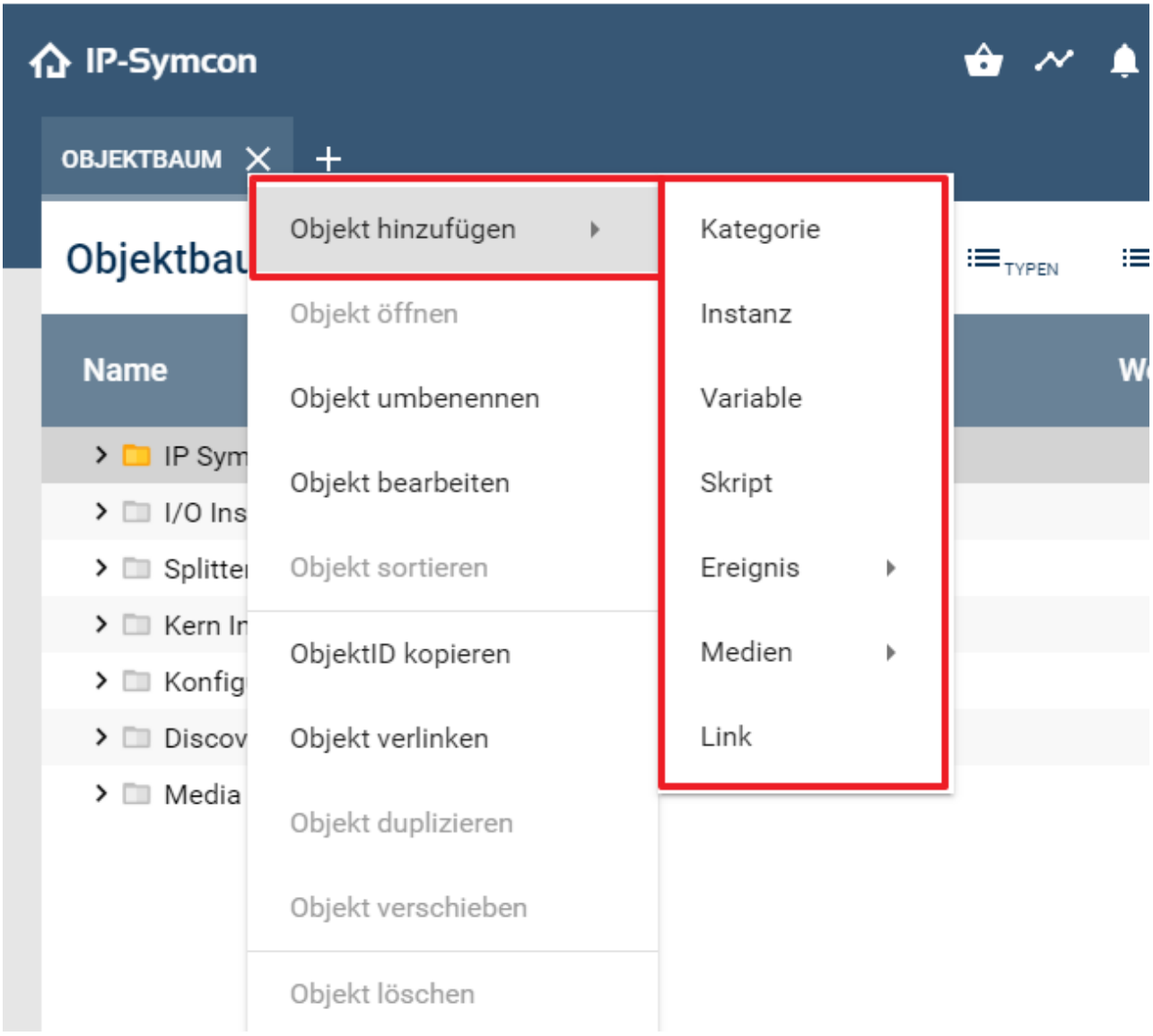
Im folgenden Abschnitt wird anhand von Bespielen erklärt wie man grundlegende oder häufig verwendete Funktionen nutzt.
Ein genereller Einstieg kann unter Schnelleinstieg gefunden werden.

# Objekt hinzufügen

Um ein neues "Objekt hinzufügen", muss auf das "+" unten rechts im Objektbaum geklickt werden. Es erscheint folgender Dialog und Auswahl von verschiedenen Objekten:




Alternativ kann innerhalb des Objektbaums auch die "Rechte Maustaste" mit einem Klick auf die gewünschte Stelle im Baum verwendet werden. Es öffnet sich folgendes Kontextmenü:


IP-Symcon
OBJEKTBAUM
Objektbau
TYPEN
Name
IP Sym
I/O Ins
Splitte
Kern In
Konfig
Discov
Media
Objekt hinzufügen
Objekt öffnen
Objekt umbenennen
Objekt bearbeiten
Objekt sortieren
ObjektID kopieren
Objekt verlinken
Objekt duplizieren
Objekt verschieben
Objekt löschen
Kategorie
Instanz
Variable
Skript
Ereignis
Medien
Link

# Geräte austauschen

Sollte ein Gerät defekt sein und physikalisch ausgetauscht werden müssen, kann dieses ohne das Anlegen einer neuen Instanz in IP-Symcon verwirklicht werden. Dies hat den Vorteil das sich die ObjektIDs der Instanz und den dazugehörigen Variablen nicht ändern. Dadurch müssen auch ggf. verknüpfte Skripte nicht auf die neuen IDs angepasst werden.

## GERÄT INNERHALB DER INSTANZ AUSTAUSCHEN

Geräte werden in IP-Symcon über sogenannte Instanzen angesprochen und eingebunden. Auf der Konfigurationsseite der jeweiligen Instanz sind oftmals systemspezifische Adressierungen vorhanden.
Diese kann geändert werden ohne die Objekt-ID zu verändern.

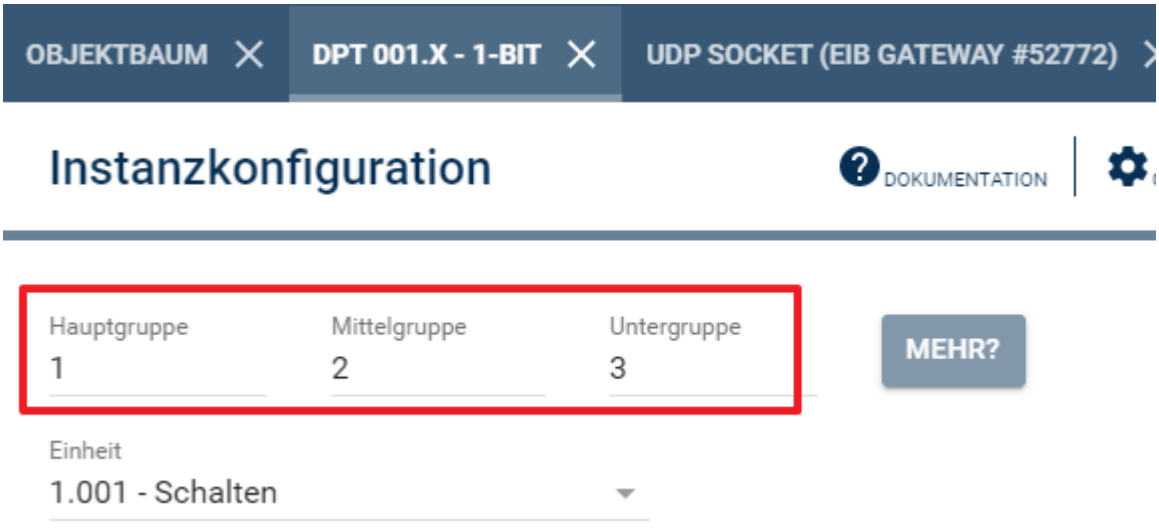
> Sofern das genutzte System einen Konfigurator besitzt kann dieser aktualisiert werden und übernimmt ebenfalls die Änderungen

## BEISPIELE

### KNX

Zum Austausch muss in IP-Symcon die Konfiguration des defekten Geräts geöffnet und die dreistufige Adresse des alten Geräts gegen die des Neuen ausgetauscht werden. Die Adresse kann im Konfigurator nachgesehen werden.
- KNX Konfigurator öffnen
- "Suchen" aber KEINE Instanz "Erstellen"
- Neue Adresse merken und die Konfiguration des alten Geräts öffnen
- Neue Adresse eintragen (siehe Screenshot) und mit "Übernehmen" speichern
- Zum Konfigurator zurückkehren und via "Suchen" aktualisieren

Die neue Adresse sollte nun eine InstanzID haben.

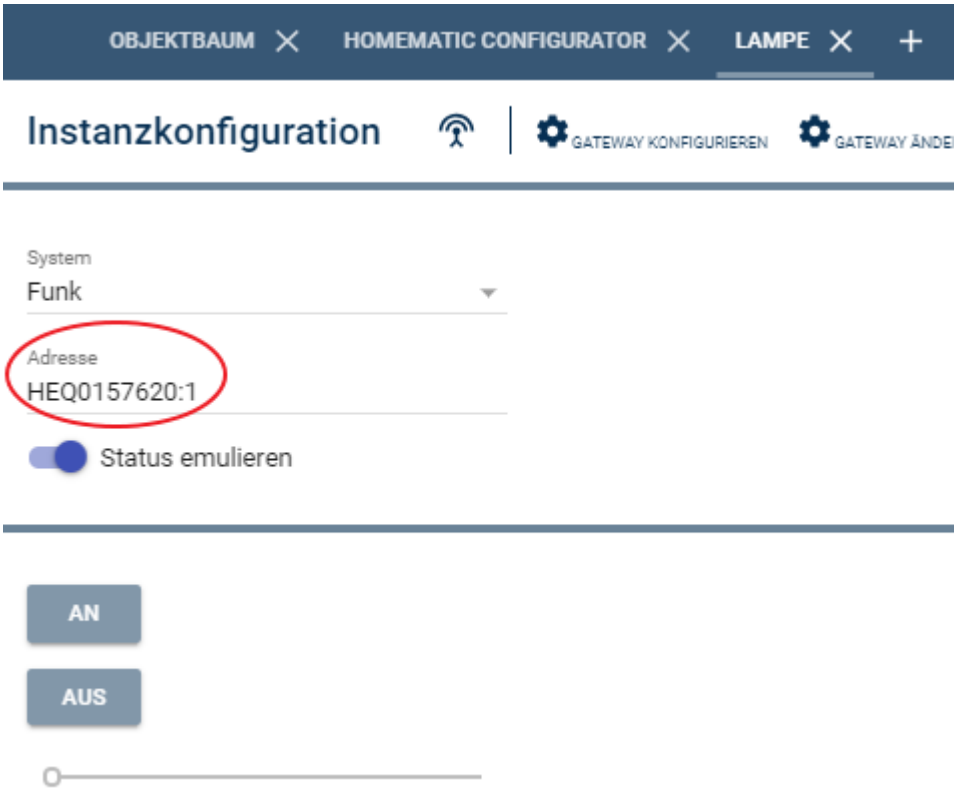



Die dreistufige Adresse im markierten Bereich muss ausgetauscht und mit "Übernehmen"

gespeichert werden.

## HOMEMATIC

Zum Austausch muss in IP-Symcon die Konfiguration des defekten Geräts geöffnet und die Adresse des alten Geräts gegen die des Neuen ausgetauscht werden. Die Adresse steht entweder auf dem Gerät selbst oder kann via "Suchen" ausgewählt werden.



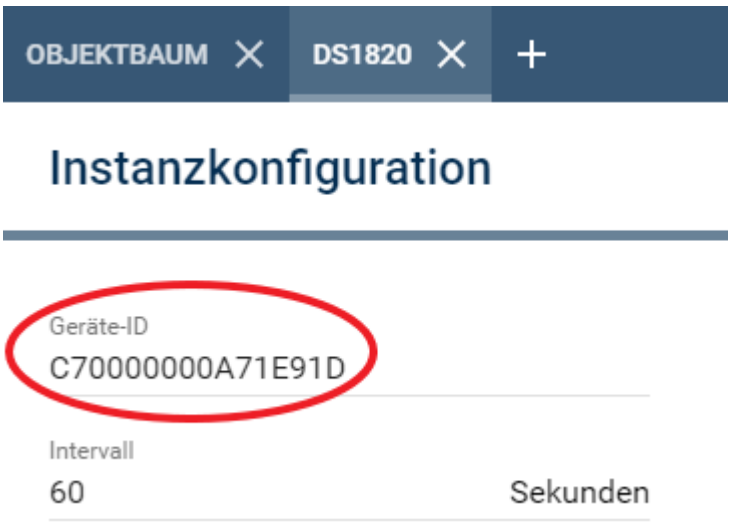

Die Adresse im markierten Bereich muss ausgetauscht und mit "Übernehmen" gespeichert werden.

## ONEWIRE

Zum Austausch muss in IP-Symcon die Konfiguration des defekten Geräts geöffnet und die 16-stellige Adresse des alten Geräts gegen die des Neuen ausgetauscht werden. Die Adresse kann im Konfigurator nachgesehen werden.

OneWire Konfigurator öffnen
"Suchen" aber KEINE Instanz "Erstellen"
Neue Adresse merken und die Konfiguration des alten Geräts öffnen
Neue Adresse eintragen (siehe Screenshot) und mit "Übernehmen" speichern
Zum Konfigurator zurückkehren und via "Suchen" aktualisieren

Die neue Adresse sollte nun eine InstanzID haben.



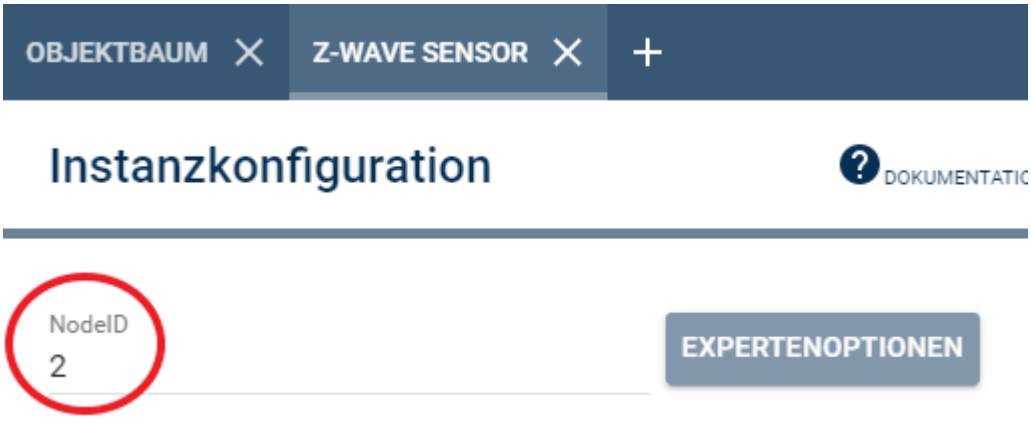

Die 16-stellige Adresse im markierten Bereich muss ausgetauscht und mit "Übernehmen" gespeichert werden.

### Z-WAVE

Zum Austausch muss in IP-Symcon die Konfiguration des defekten Geräts geöffnet und die NodeID des alten Geräts gegen die des Neuen ausgetauscht werden. Die NodeID kann im Konfigurator nachgesehen werden.

Z-Wave Konfigurator öffnen
Gerät anlernen aber KEINE Instanz "Erstellen"
NodeID merken und die Konfiguration des alten Geräts öffnen
Neue NodeID eintragen (siehe Screenshot) und mit "Übernehmen" speichern
Zum Konfigurator zurückkehren und "Aktualisieren"

Die neue NodeID sollte nun eine InstanzID haben.




Die NodeID im markierten Bereich muss ausgetauscht und mit "Übernehmen" gespeichert werden.

# Geräte einbinden

Geräte werden in IP-Symcon über sogenannte Instanzen angesprochen.



Das Prinzip vom Anlegen der Geräte hat sich von Version 1 an grundlegend verändert. I/O Instanzen, Splitter oder Ähnliches müssen nicht mehr erstellt/verbunden werden. IP-Symcon erledigt dies automatisch. IP-Symcon legt auch die Status-Variablen automatisch an.

## INSTANZ ERSTELLEN

Bei folgenden Systemen stehen Konfiguratoren für eine komfortablere Konfiguration von Geräten zur Verfügung. Dafür kann innerhalb der webbasierten Verwaltungskonsole über den Knopf "+" den Objektbaum der passende Konfigurator hinzugefügt werden.

- digitalStrom
- Eaton xComfort
- KNX
- HomeMatic
- LCN
- MQTT
- Siemens OZW
- Z-Wave
- 1-Wire

Bei allen anderen Systemen muss das jeweilige Gerät selbst hinzufügt werden indem eine passende Instanz erstellt wird.
Der Erstellungsdialog kann im Objektbaum über den Button "+" -> "Instanz" erreicht werden.
IP-Symcon kümmert sich darum, dass benötigte Gateways, Splitter und I/O's zu erstellen.
Im Reiter "Meldungen" kann überprüft werden ob eine Instanz einen Fehler hat.
Interessant ist eine gelb oder rot unterlegte Meldung. Dies bedeutet es fehlt noch eine Einstellung damit eine Instanz funktioniert. Ein "Doppelklick" auf die jeweilige Meldung führt direkt zu dem Abschnitt von IP-Symcon, bei dem besagte Einstellung noch benötigt wird.
Ein Zugriff auf die zu jedem Gerät gehörigen Gateway und I/O Schnittstellen ist über das Zahnrad "Gateway Konfigurieren" oder "Schnittstelle Konfigurieren" im Geräte-"Konfiguration"-Reiter möglich. Dort können benötigte Einstellungen überprüft und ggf. nachgebessert werden.
Im Bereich "Modulreferenz " befinden sich, beschrieben oder per Video-Tutorial, die Einrichtungsanleitungen der jeweiligen Geräte/Instanzen.

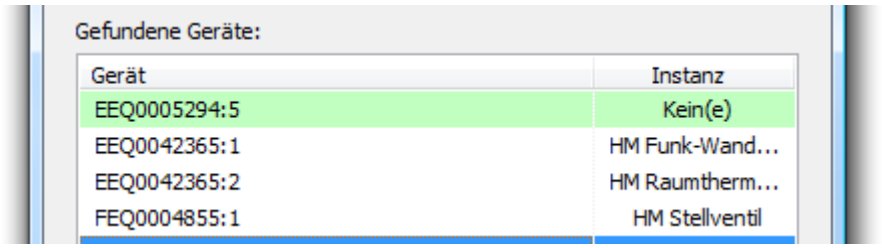
# Geräte suchen

IP-Symcon verfügt über eine sehr komfortable Funktion, die es dem Anwender ermöglicht, schnell und einfach neue Geräte dem System hinzuzufügen. Voraussetzung ist, dass die neuen, anzulernenden Geräte von sich aus Meldungen verschicken (z.B. Funk-Temperatur-Fühler). Alternativ kann - oder muss - der Anwender eine "Anlern-Taste" oder ganz einfach einen Knopf auf der Fernbedienung drücken. Bei einigen Systemen, wie z.B. das 1-Wire-System, kann IP-Symcon gezielt nach angeschlossenen, unbekannten Geräten suchen. Nach dem Motto: "Hallo wer ist denn da?". Und die Geräte antworen mit: "Ich bin es, der Helligkeissensor".

Geräte können über die Konfiguratoren der jeweiligen Systeme gesucht werden.

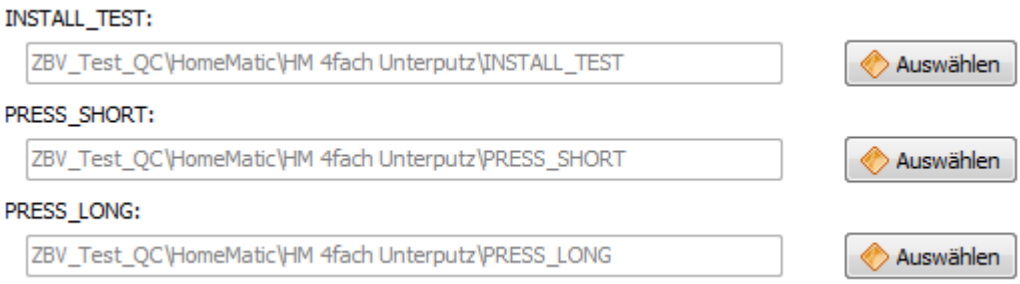
## BEISPIELE

Das folgende Beispiel zeigt die empfangenen Geräte des HomeMatic Systems:



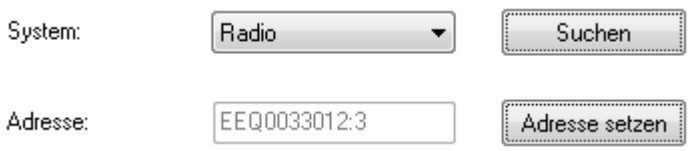

Es kann mehrere Minuten dauern, bis alle Geräte in der Liste erscheinen!

Ein grüner Hintergrund signalisiert ein neues Modul. Durch einen Doppelklick kann es dann ausgewählt werden.
Je nach Eigenschaften des Gerätes legt IP-Symcon automatisch Variablen an, die Funktionalität wiederspiegeln. Die Namen sowie den Ort können Sie jederzeit verändern. Das folgende Beispiel zeigt die drei Variablen eines Funkschalters bzw. einer Fernbedienung des HomeMatic System:




Beim HomeMatic System muss zusätzlich für die Suche angegeben weden, ob funk- oder kabelgebundene Geräte gesucht werden sollen.

Wenn die BidCoS-Seriennummer bekannt ist, kann diese auch direkt eingegeben werden:



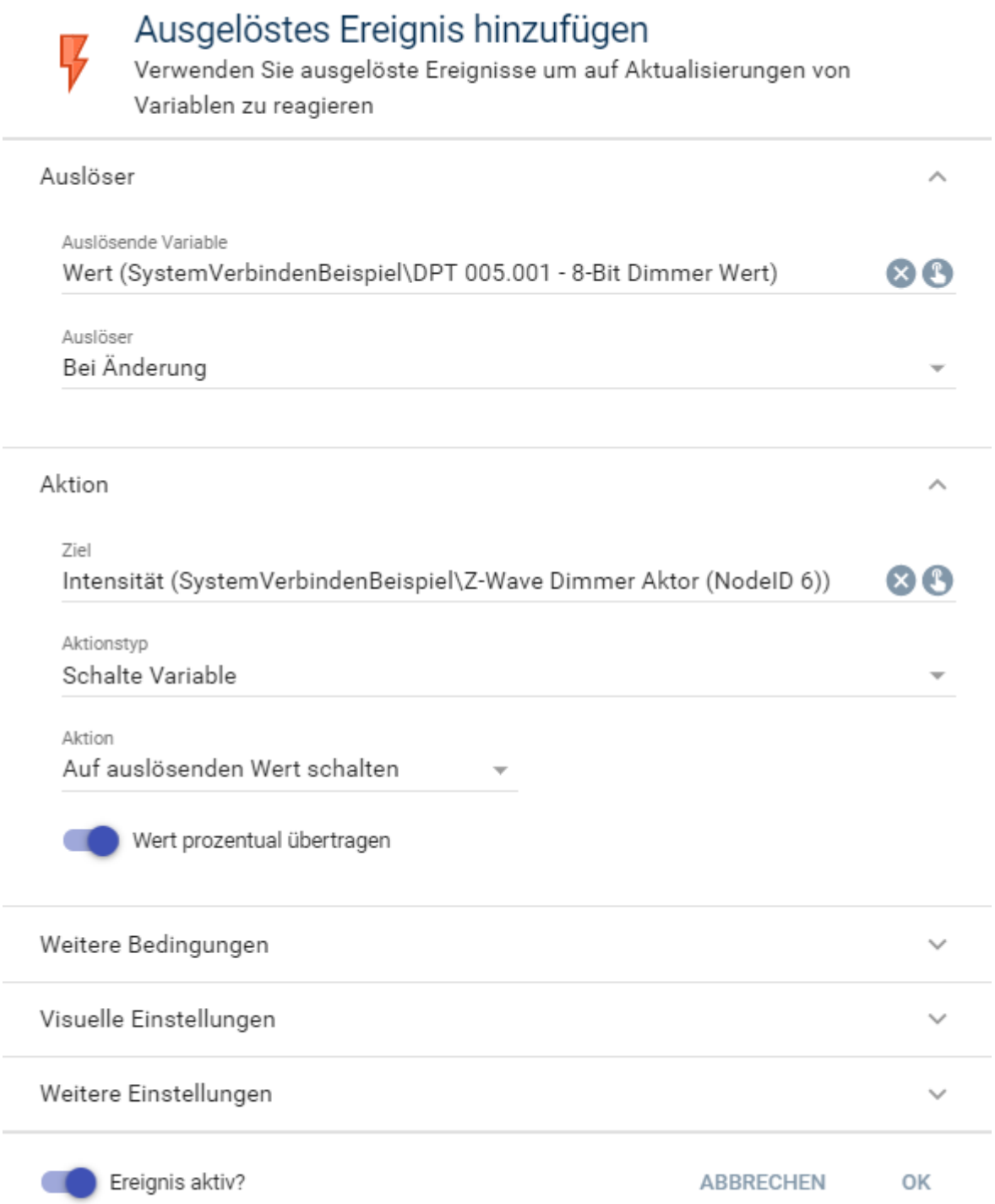

Beachten Sie bitte, dass erst beim zweiten Empfang eines Datensatzes des jeweiligen Gerätes seine Variablen mit Werten gefüllt werden.

# Systeme verbinden über Ereignisse

IP-Symcon verfügt über eine sehr komfortable Möglichkeit Werte von einem System direkt auf ein anderes zu übertragen. Dazu kann ein Ereignis hinzugefügt werden, welches bei Änderung des auslösenden Wertes einen anderen Wert auf den auslösenden Wert setzen kann. Dabei ist es unerheblich, ob die beiden Werte zu verschieden Systemen gehören. Im unten aufgeführten Beispiel wird ein KNX Dimmerwert auf ein Z-Wave Dimmer übertragen.

## BEISPIEL

Ein KNX Dimmerwert kann an eine Z-Wave Lampe mit Dimmerwert übertragen werden. Dafür wird auf eine Änderung des KNX Wertes reagiert und der neue Wert auf den Dimmwert der Z-Wave Lampe übertragen und diese geschaltet.
Im Objektbaum sind die beiden Geräteinstanzen bereits angelegt und funktionieren für sich einwandfrei. Nun muss ein Ereignis erstellt werden. Dazu kann über das "+" unten Rechts im Objektbaum ein Ereignis hinzugefügt werden.
Das Ereigniss muss wiefolgt konfiguriert werden.
Als auslösende Variable wird der Dimmwert ausgewählt, dessen Wert bei Änderung übertragen werden soll. In diesem Beispiel ist es der Wert der DPT 005.001.
Unter Aktion wird als "Ziel" die Intensität der Z-Wave Lampe mit Dimmer Aktor ausgewählt. Als Aktionstyp muss "Schalte Variable" und als Aktion "Auf auslösenden Wert schalten" ausgewählt werden.
Mit "OK" wird das Ereignis erstellt und die Konfiguration gespeichert.




Ab sofort wird bei jeder Wertänderung des KNX Dimmerwerts die Intensität der Z-Wave Instanz auf den gleichen Wert gedimmt.
Der Objektbaum sollte dann so aussehen.



# Skripte wiederverwenden



Dies ist ein Beispiel, welches sich auch mit Ereignissen abbilden lässt. Dieser Artikel möchte eher die Systemvariablen und Wiederverwendbarkeit von Skriptabschnitten näherbringen.

In diesem Beispiel geht es um eine Flurlicht-Steuerung, die auf allen Etagen gleich ist. Diese soll möglichst nicht dreimal programmiert werden.
Ein Möglichkeit wäre es 3 Skripte anzulegen und den Inhalt von einem Skript mit Copy und Paste in die einzelnen Skripte kopieren. Dies ist für eine schnelle Lösung sicherlich Möglich. Dies führt aber auch zu einer gewissen Redundanz und wenn an einer Stelle im Skript etwas verbessert werden soll, muss dies nicht einmal sondern dreimal gemacht werden.
Als Beispiel für Code-Reusing kann zum Beispiel ein Skript mit dem Namen "Lichtsteuerung (xComfort, Flur, Bewegungsmelder, Helligkeitsregelung)" er stellt werden. Der Name sollte sprechend sein und beinhalten welche Systeme, Geräte und Status es verwendet.
Folgender Code könnte der Inhalt sein:

```
//Wenn keine Bewegung - siehe Dokumentation "Systemvariablen" für
die Bedeutung von $_IPS['VALUE']
if(!$_IPS['VALUE']) {
MXC_SwitchMode($lampenID, false);
}
//Wenn Bewegung festgestellt wurde
else {
//Nur nach Sonnenuntergang
if(!GetValueBoolean($istTag)) {
//Wenn es zwischen 22:30 Abends und 06:00 Morgends ist
| if((time() > strtotime("22:30")) || (time() < strtotime("06:00")))
{ |
MXC_DimSet($lampenID, 15);
}
else {
MXC_DimSet($lampenID, 50);
}
}
}
```

Dieses Skript muss von einem Auslösenden Ereignis aufgerufen werden - hier normalerweise von einem/mehreren Bewegungsmeldern. Das Licht wird dann abhängig von der Tageszeit auf eine unterschiedliche Dimmstufe gestellt. Sobald der Bewegungsmelder keine Bewegung mehr erkennt und den FALSE Impuls sendet, schaltet das Gerät ab.
Die Logik ist somit ausgelagert und es müssen nur noch Skripte erstellt werden, welche die fehlenden Variablen setzen und von einem Ereignis aufgerufen werden. Nachfolgend ein Beispiel:

```
//Auslösung des Ereignis durch Bewegungsmelder Variable!
```

```
//Eindeutige Geräte-ID
$lampenID = 54321 /*[Erdgeschoss\Flur\Deckenlampe]*/;
//Tag/Nacht Variable aus dem Location Control
$istTag = 56789 /*[IstTag]*/;

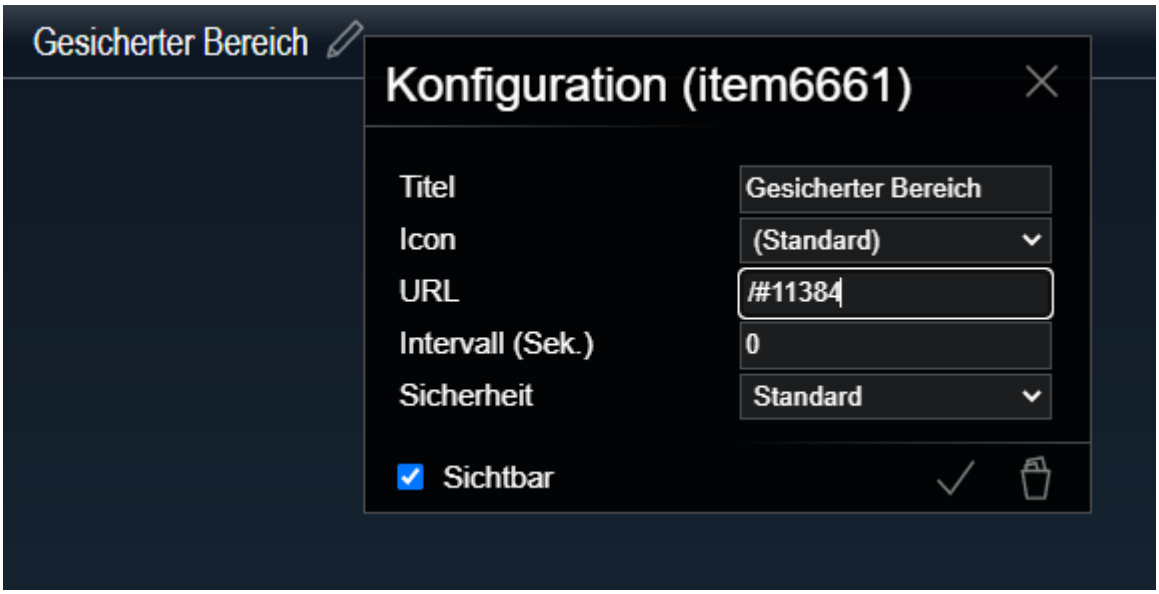
includeScript(12345 /*[Szenarien\Lichtsteuerung]*/);

//Funktion zum Einfügen des Skriptinhalts
function includeScript($scriptID) {
$s = IPS_GetScript($scriptID);
include($s['ScriptFile']);
}
```

So können "Funktionsvorlagen" erstellt und diese dann mit den nötigen IDs gefüttert werden. In diesen Skripten sind dann die IDs der Geräte/ Variablen, zu denen IP-Symcon vollständige und aussagekräftige Namen erstellt (sofern die Objektbaumstruktur vernünftig erstellt wurde).
Bei Bedarf kann das Skript auch für andere Systeme angepasst und dynamisch anhand der übergebenen InstanzID die richtigen Funktion aufgerufen oder die systemspezifischen Änderungen vorgenommen werden.

# Passwort geschützte Kategorie im WebFront

Um bestimmte Kategorien aus dem Objektbaum im WebFront mit Passwort zu schützen, müssen folgende Schritte beachtet werden.
Die Kategorie, welche gesichert werden soll, muss im Objektbaum aus dem restlichen WebFront verschoben werden.
Anschließend muss ein neues WebFront eingerichtet werden. Ist dies geschehen, kann ein beliebiges Passwort in der Sicherheit des WebFronts eingerichtet werden.
Im zweiten WebFront muss über den Editor alles entfernt und die Kategorie, welche den passwortgeschützen Inhalt anzeigen soll, hinzugefügt werden.
Im Anschluss muss im ersten WebFront das zweite WebFront als Externe Seite mit einem Pfad hinzugefügt werden.

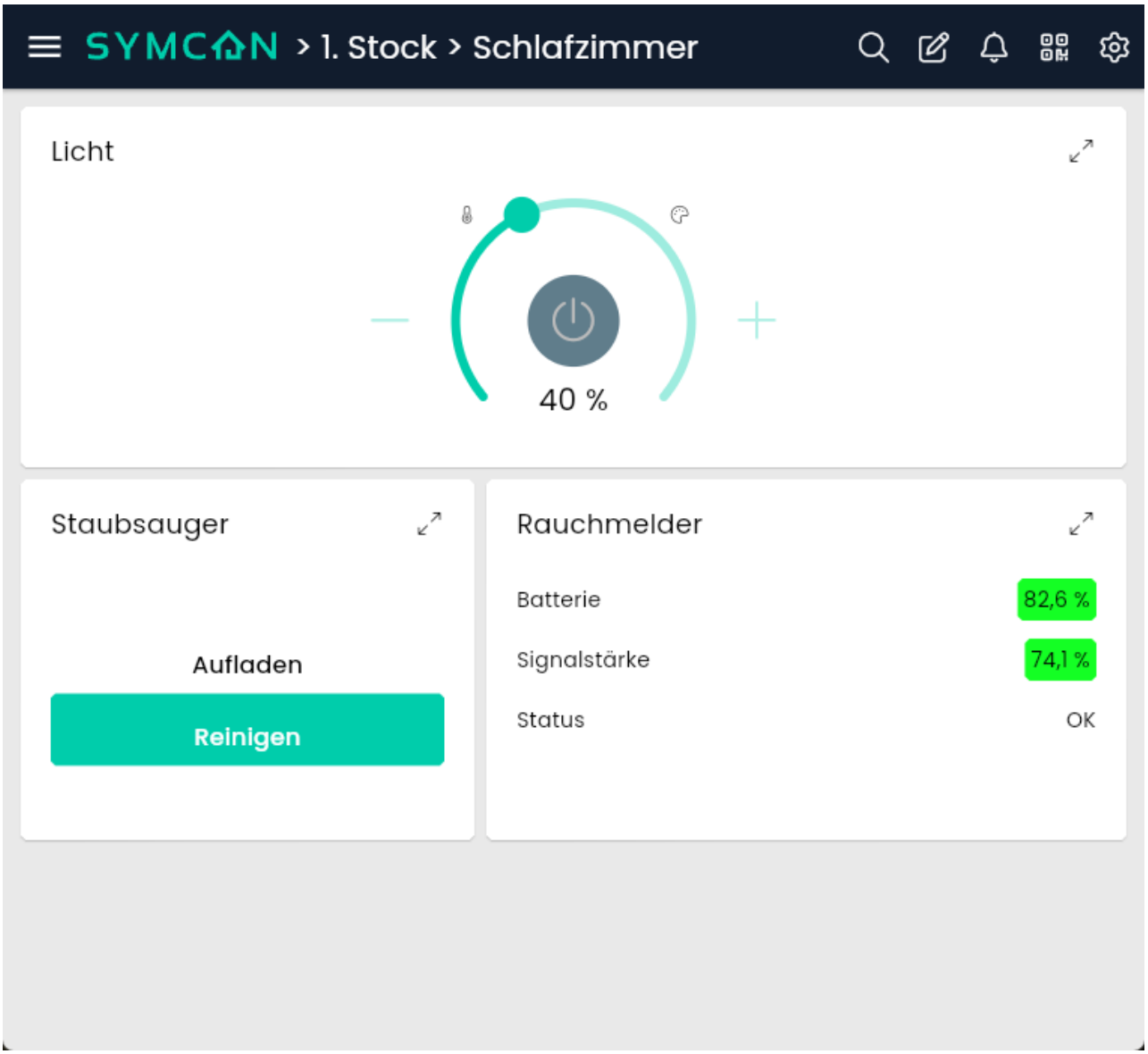



Der Aufruf der Externen Seite sieht dann wiefolgt aus.

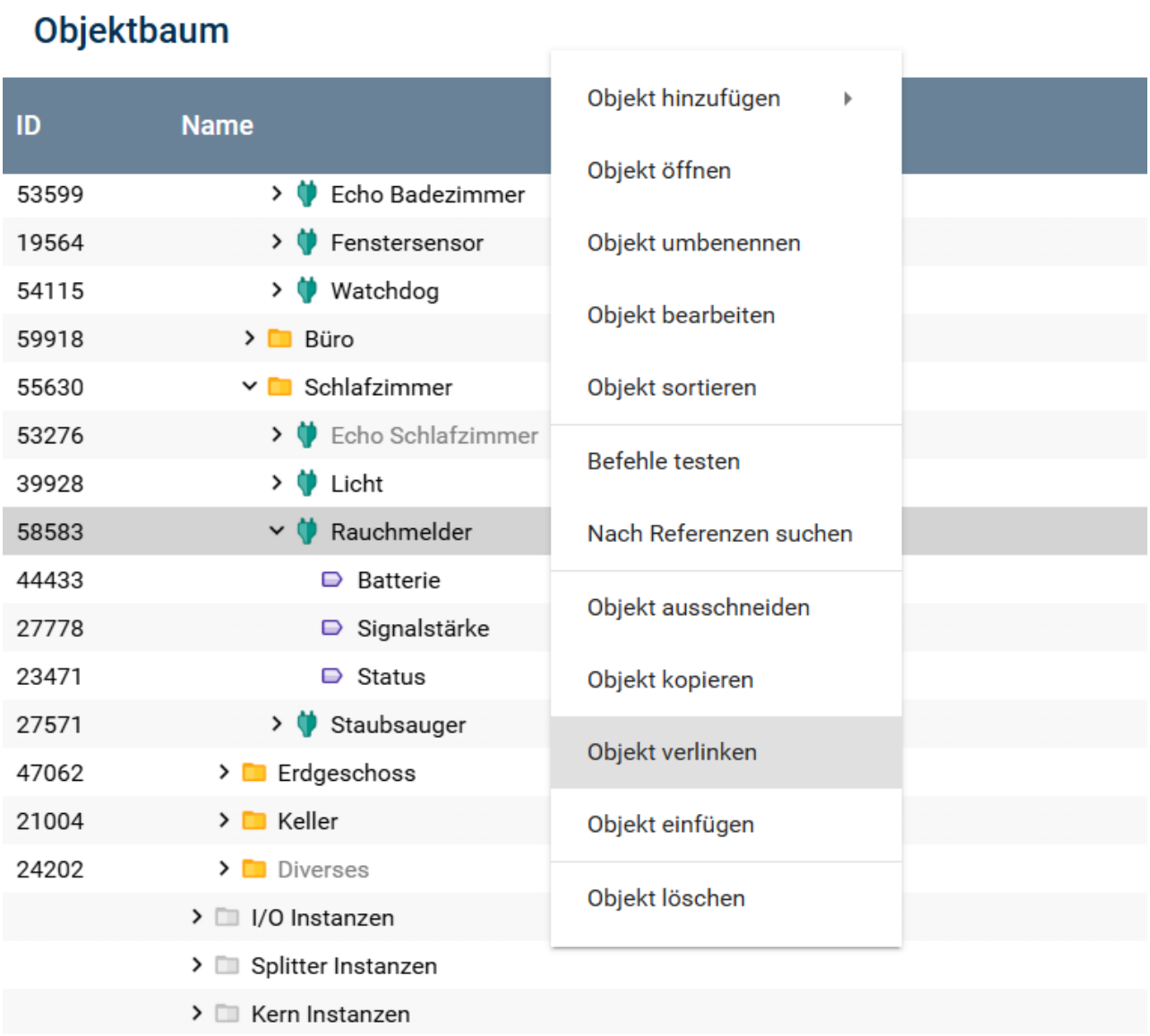

IP-Symcon
WebFront
Wetter (DWD)
Gesicherter Bereich
14:48:18
SYMCON
WebFront
Benötige Kennwort

# Links verwenden

Im folgenden Beispiel wird dem Prinzip der Links nachgegangen. In einem Haus sind zum Beispiel etliche Rauchmelder verbaut und angeschlossen.
Zur Sicherheit sollen nun alle drei Monate die Rauchmelder kontrolliert werden. Eine erste Kontrolle war recht aufwändig. Bei der Strukturierung der Visualisierung wurde der Aufbau sinnvollerweise nach Etage -> Zimmer -> Gerät gestaltet. Bei der Kontrolle muss nun allerdings jedes Zimmer einzeln durchlaufen werden um jeden Rauchmelder zu kontrollieren. Das erscheint recht zu aufwendig.




Nun wird neben den Stockwerken eine Kategorie namens “Rauchmelder” erstellt und alle Rauchmelder werden per “Drag and Drop” in diese Kategorie verschoben.
Dadurch, dass alle Rauchmelder in Symcon einer neuen Kategorie zugeordnet wurden, sind diese in der Visualisierung nun nicht mehr in den einzelnen Räumen sichtbar. Damit fehlen die Rauchmelder aber nun leider in ihren den einzelnen Räumen, in denen sie installiert sind.
Symcon bietet Ihnen die Möglichkeit, Verknüpfungen zu einzelnen Objekten zu erstellen. Um diese Verknüpfungen zu erstellen, muss mit der rechten Maustaste im Objektbaum auf ein ausgewähltes Objekt - in unserem Fall also auf einen Rauchmelder - geklickt werden.




Sobald auf “Objekt verlinken” geklickt wurde ist das Objekt zum Verlinken in der Zwischenablage. An beliebiger Stelle im Objektbaum kann nun per Rechtsklick und “Objekt einfügen” ein Link auf das ursprünglich ausgewählte Objekt erstellt werden. Dies kann mit allen Rauchmeldern wiederholt werden und schon können die Rauchmelder problemlos in der Visualisierung kontrolliert werden, ohne ständig hin und her zu klicken. Dennoch sind sie weiterhin in ihren jeweiligen Zimmern zu finden. Obendrein können Verknüpfungen per Rechtsklick auf “Objekt bearbeiten” einen eigenen Namen und ein eigenes Icon erhalten.

# Push Notification an verschiedene Benutzergruppen senden

Im Folgenden wird das Vorgehen zum Versenden von Push-Notifications an verschiedene Benutzergruppen erklärt.
Eine Benutzergruppe besteht in diesem Beispiel aus mehreren an ein WebFront angemeldeten Endgeräten (Smartphones).

## EINRICHTUNG

Benutzergruppen für Push-Notifications werden in IP-Symcon über seperate WebFronts bzw. WebFront Visualisierungen realisiert.



Um mehr als ein WebFront einrichten zu können wird eine **IP-Symcon Professional** oder **IP-Symcon Unlimited** Version benötigt.

Wie ein weiteres WebFront eingerichtet wird kann unter "Weiteres WebFront einrichten " nachgelesen werden.
Diese WebFronts können anschließend nach persönlichen Bedürfnissen seperat eingerichtet werden.



Um gleiche Instanzen in verschiedenen WebFronts nutzen zu können werden Links benötigt. Ein Beispiel ist unter "Links verwenden " zu finden.

## ANMELDUNG IN DEN BENUTZERGRUPPEN

Wie oben beschrieben werden Benutzergruppen über WebFronts verwirklicht. Dabei reicht es aus, sich mit der App auf dem Smartphone, welches angemeldet werden soll, einmalig in dem jeweiligen WebFront anzumelden. Dadurch wird das Gerät automatisch der jeweiligen Benutzergruppe hinzugefügt. Ab diesem Zeitpunkt bekommt die App des Smartphones automatisch jede Push-Notification. Auf diese Weise kann das Smartphone bei beliebig vielen Benutzergruppen angemeldet werden.



Ein einloggen in das jeweilige WebFront über den Handy-Browser hat keine Eintragung in die Benutzergruppe zur Folge.

## ABMELDUNG AUS EINER BENUTZERGRUPPE

Abmeldung aus einzelner Benutzergruppe:
In der WebFront Visualisierung können im Reiter "Benachrichtigungen" die angemeldeten Geräte verwaltet werden. Somit kann ein einzelnes Smartphone von einer Benutzergruppe abgemeldet werden.

**Abmeldung aus allen Benutzergruppen:**
Soll ein Smartphone aus sämtlichen Benutzergruppen abgemeldet werden, ist dies über die Kern Instanz "Notification Control" möglich.
Im Konfigurations-Reiter sind alle Mobilgeräte ersichtlich, welche in einem WebFront angemeldet sind. Löscht man hier das Gerät, ist das Smartphone aus allen Benutzergruppen abgemeldet.

## VERSENDEN VON PUSH-NOTIFICATIONS

Eine Push-Notification kann durch WFC_PushNotification erstellt werden. Diese braucht die ObjektID des jeweilgen WebFronts an das die Push-Notification gesendet werden soll. Alle an diesem WebFront angemeldeten Smartphones bekommen dann über die App die Notification.

## TIPPS & TRICKS

Wenn über den Reiter "Sicherheit" ein Passwort für das jeweilige WebFront eingerichtet wird, ist es leicht zu verhindern, dass man sich für eine Benutzergruppe "versehentlich" anmeldet.

## ANWENDUNGSBEISPIELE

Ein Haushalt mit mehreren WebFronts.

**Ein WebFront für Administratoren:** Nur Administratoren sind in dieser Benutzergruppe und werden über vermeintliche Systemprobleme benachrichtigt.
**Ein WebFront für einzelnes Familienmitglied:** Beinhaltet persönliche Interessen und sendet Nachrichten die nur die jeweilige Person interessieren.
**Ein WebFront für alle Familienmitglieder:** Der Kühlschrank meldet "Die Milch ist leer, jemand muss einkaufen", Das Wettermodul meldet "Es regnet gleich" oder Einbrecheralarm

# Wie kann ich...?



Viele dieser Skripte verwenden spezielle IP-Symcon Funktionen. Bitte nutzen Sie die Befehlsreferenz/Modulreferenz, wenn Sie die genaue Funktionsweise verstehen wollen.


... ein Gerät einschalten und nach 60 Sekunden wieder aussschalten
... eine Liste der Modulnamen samt GUID bekommen
... eine Instanz aus PHP konfigurieren
... die Anzahl der Sekunden herausfinden, die ein ScriptTimer noch abwartet
... ein Skript per ID direkt includen
... UpdateTime in einer eigenen String Variable ausgeben (1 Skript - n Variablen)
... aus PHP einen Timer & Variable anlegen
... eine Datei aus dem Internet herunterladen
... einen Ordner rekursiv in die MediaPlayer Playlist laden
... ein Variablenprofil exportieren


## ... EIN GERÄT EINSCHALTEN UND NACH 60 SEKUNDEN WIEDER AUSSSCHALTEN

```
if($_IPS['SENDER'] == "TimerEvent")
{
//Aus Befehl
...

//Timer ausschalten
IPS_SetScriptTimer($_IPS['SELF'], 0);
} else {
//An Befehl
...

//Timer anschalten
IPS_SetScriptTimer($_IPS['SELF'], 60);
}
```

## ... EINE LISTE DER MODULNAMEN SAMT GUID BEKOMMEN

```
foreach(IPS_GetModuleList() as $mid)
{
$m = IPS_GetModule($mid);
echo $mid."=".$m['ModuleName']."\n";
}
```

## ... EINE INSTANZ AUS PHP KONFIGURIEREN

```
//Eigenschaft verändern
WWWReader_SetPage($id,"http://www.google.de");

//Änderung abspeichern
IPS_ApplyChanges($id);

//Neue URL abrufen
WWWReader_UpdatePage($id);
```

## ... DIE ANZAHL DER SEKUNDEN HERAUSFINDEN, DIE EIN SCRIPTTIMER NOCH ABWARTET

```
echo GetTimeRemaining($_IPS['SELF']); //Von sich selbst herausfinden
function GetTimeRemaining($id)
{
$eid=@IPS_GetEventIDByName("ScriptTimer", $id);
if($eid === false) {
return -1;
} else {
$e=IPS_GetEvent($eid);
if($e['NextRun'] == 0)
{
return -1;
} else {
return $e['NextRun'] - microtime(true);
}
}
}
```

## ... EIN SKRIPT PER ID DIREKT INCLUDEN

```
//Skript mit ID 14871 einbinden
include(IPS_GetScriptFile(14871));
```

## ... UPDATETIME IN EINER EIGENEN STRING VARIABLE AUSGEBEN (1 SKRIPT - N VARIABLEN)

```
//Event auswerten
if($_IPS['SENDER'] != "Variable")
return;

SetValue(CreateVariableIDByName($_IPS['VARIABLE'], 'Updated', 3),
date("d.m.y H:i:s"));

function CreateVariableIDByName($id, $name, $type)
```

```
{
$vid = @IPS_GetVariableIDByName($name, $id);
if($vid===false) {
$vid = IPS_CreateVariable($type);
IPS_SetParent($vid, $id);
IPS_SetName($vid, $name);
IPS_SetInfo($vid, "This Variable was created by Script #".
$_IPS['SELF']);
}
return $vid;
}
```

## ... AUS PHP EINEN TIMER & VARIABLE ANLEGEN



Wenn das Skript ausgeführt wird, legt es einen Timer an der alle sechs Stunden startet und dann eine Variable mit der jeweiligen Tageszeit als Wert zwischen 0-3 in die Variable ablegt.

```
//HINWEIS:
//~~~~~~~~
//Dieses Script richtet sicht automatisch ein, wenn es ausgeführt
wird
//
//- Es wird abhängig von der Tageszeit eine Variable gesetzt (0-3)
// 0 = 0-6
// 1 = 6-12
// 2 = 12-18
// 3 = 19-24

//------------------------------------------------------------------------
//Ab diesem Punkt muss nichts verändert werden
//------------------------------------------------------------------------
if($_IPS['SENDER'] == "Execute")
{
$eventid = @IPS_GetEventIDByName("Timer", $_IPS['SELF']);
if($eventid === false)
{
$eventid = IPS_CreateEvent(1); //Cyclic
IPS_SetEventActive($eventid, true);
IPS_SetName($eventid, "Timer");
IPS_SetEventScript($eventid, $_IPS['SELF']);
IPS_SetEventCyclic($eventid, 0, 0, 0, 0, 3, 6);
}
$variableid = @IPS_GetVariableIDByName("Daytime", $_IPS['SELF']);
if($variableid === false)
{
$variableid = IPS_CreateVariable(1);
IPS_SetName($variableid, "Daytime");
IPS_SetParent($variableid, $_IPS['SELF']);
}
}
SetValue(IPS_GetVariableIDByName("Daytime", $_IPS['SELF']),
```

```
floor(date("H") / 6));
```

## ... EINE DATEI AUS DEM INTERNET HERUNTERLADEN

```
$remoteImage = "https://www.google.com/images/srpr/logo3w.png";
$localImage = IPS_GetKernelDir()."\\media\\bild.jpg";

//Downloaden
$content = @file_get_contents($remoteImage);
if((strpos($http_response_header[0], "200") === false))
{
return;
}

//Speichern
file_put_contents( $localImage, $content );
```

# Tastenkombinationen

Bezeichnung der Sondertasten auf verschiedensprachigen Tastaturen:

| DEUTSCH | ENGLISCH |
|---|---|
| Eingabe | Enter |
| Entf | Del |
| Leertaste | Space |
| Strg | Ctrl |
| Umschalt | Shift |

## TASTENKOMBINATIONEN GENERELL

| SHORTCUT | BESCHREIBUNG |
|---|---|
| Esc | Schließt den aktuellen Reiter |

## TASTENKOMBINATIONEN IM SKRIPTEDITOR

| SHORTCUT | BESCHREIBUNG |
|---|---|
| F3 | Suche nächsten |
| Umschalt + F3 | Suche vorherigen |
| Strg + A | Alles auswählen |
| Strg + C | Kopiert Text mit Strg + V |
| Strg + E | Ausführen |
| Strg + F | Suchen |
| Strg + H | Ersetzen |
| Strg + O | Objekt auswählen |
| Strg + R | Skript umbenennen |
| Strg + S | Speichern |
| Strg + D | Ereignisse anzeigen |
| Strg + X | Verschiebt Text mit Strg + V |
| Strg + Umschalt + K | Zeile löschen |
| Strg + V | Fügt Text aus Zwischenablage ein |
| Strg + Z | Rückgängig machen |
| Strg + Umschalt + Z, Strg + Y | Wiederholen |
| Strg + Leertaste | Funktionsliste anzeigen |
| Strg + Umschalt + F | In allen Skripten suchen |
| Strg + Umschalt + H | In allen Skripten ersetzen |

## TASTENKOMBINATIONEN IM OBJEKTBAUM

| SHORTCUT | BESCHREIBUNG |
|---|---|
| Eingabe | Öffnet Objekt oder bearbeitet Objekt, falls es nicht geöffnet werden kann |
| Entf | Löscht Objekt |
| Alt + 0 | Kategorie hinzufügen |
| Alt + 1 | Instanz hinzufügen |
| Alt + 2 | Variable hinzufügen |
| Alt + 3 | Skript hinzufügen |
| Alt + 4 | Ereignis hinzufügen |
| Alt + 5 | Medien hinzufügen |
| Alt + 6 | Link hinzufügen |
| Strg + C | Kopiert Objekt mit Strg + V, kopiere ObjektID in Zwischenablage |
| Strg + L | Erstellt Link auf Objekt mit Strg + V |
| Strg + X | Verschiebt Objekt mit Strg + V |
| Strg + V | Kopiert, verlinkt, oder bewegt Objekt |
| Strg + E | Führt Objekt aus |
| Strg + F | Suche ID |
| Strg + R, F2 | Objekt umbenennen |
| Strg + Eingabe | Objekt bearbeiten |

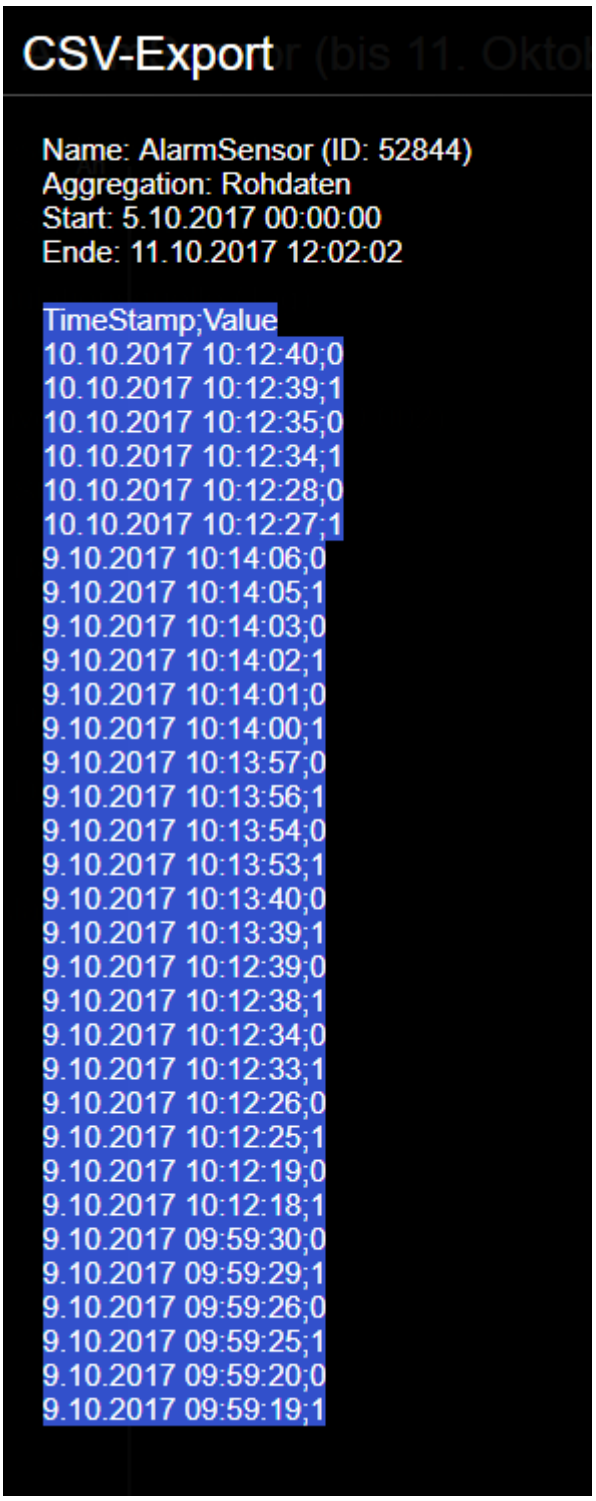
# CSV Daten aus WebFront in Excel speichern

Graphen im WebFront werden über CSV Datensätze dargestellt. Um diese für eine weitere Verwendung in Excel, in ordentlicher und leserlicher Form, zur Verfügung zu stellen gibt es die Funktionalität "Text in Spalten".
Dies funktioniert für jegliche CSV Datensätze. Die so aufgebrochenen CSV Daten können zum Beispiel für Excelgrafiken o. ä. genutzt werden.

## BEISPIEL

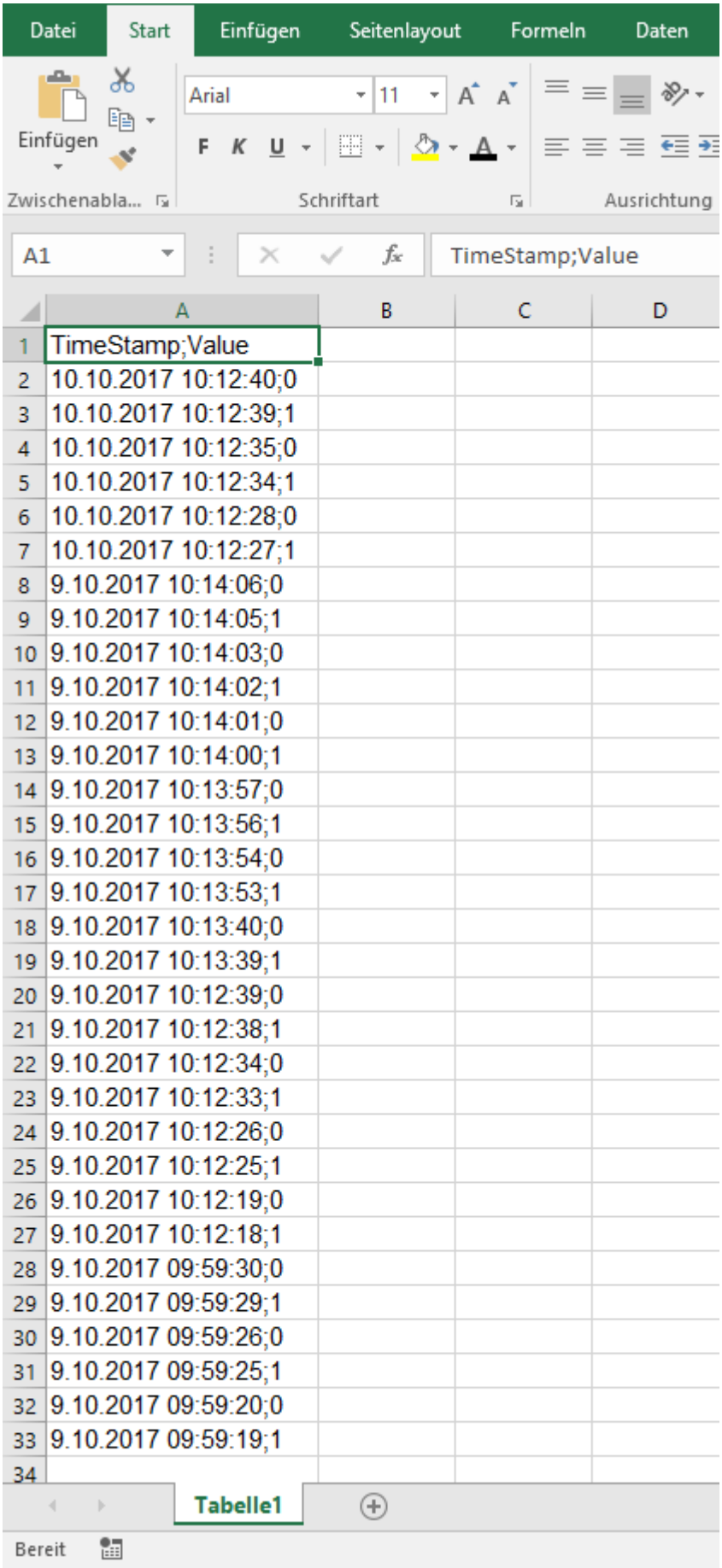
### SCHRITT 1 - CSV EXPORT DATENSATZ KOPIEREN

Innerhalb des WebFronts muss auf das Graphensymbol und anschließend auf den Menüpunkt "CSV" geklickt werden. In dem Popup (siehe Bild) müssen nur die Datensätze markiert und mit "Strg + C" kopiert werden.




### SCHRITT 2 - EINFÜGEN

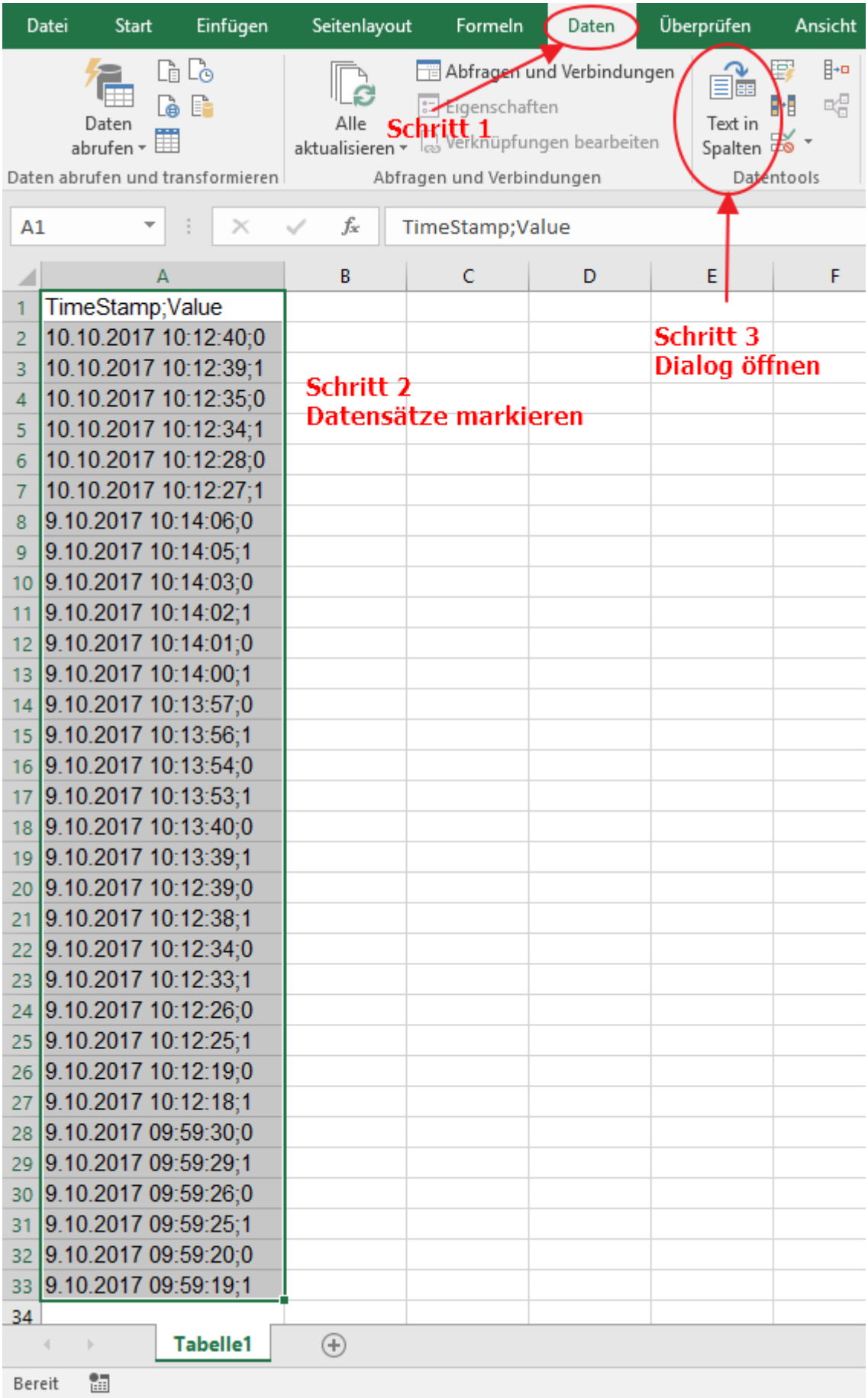
Die Daten in eine leere Tabelle mit "Strg + V" einfügen. Gegebenenfalls die Schriftfarbe auf schwarz ändern, da die weiße Schriftfarbe möglicherweise übernommen wurde.




## SCHRITT 3 - TEXT IN SPALTEN

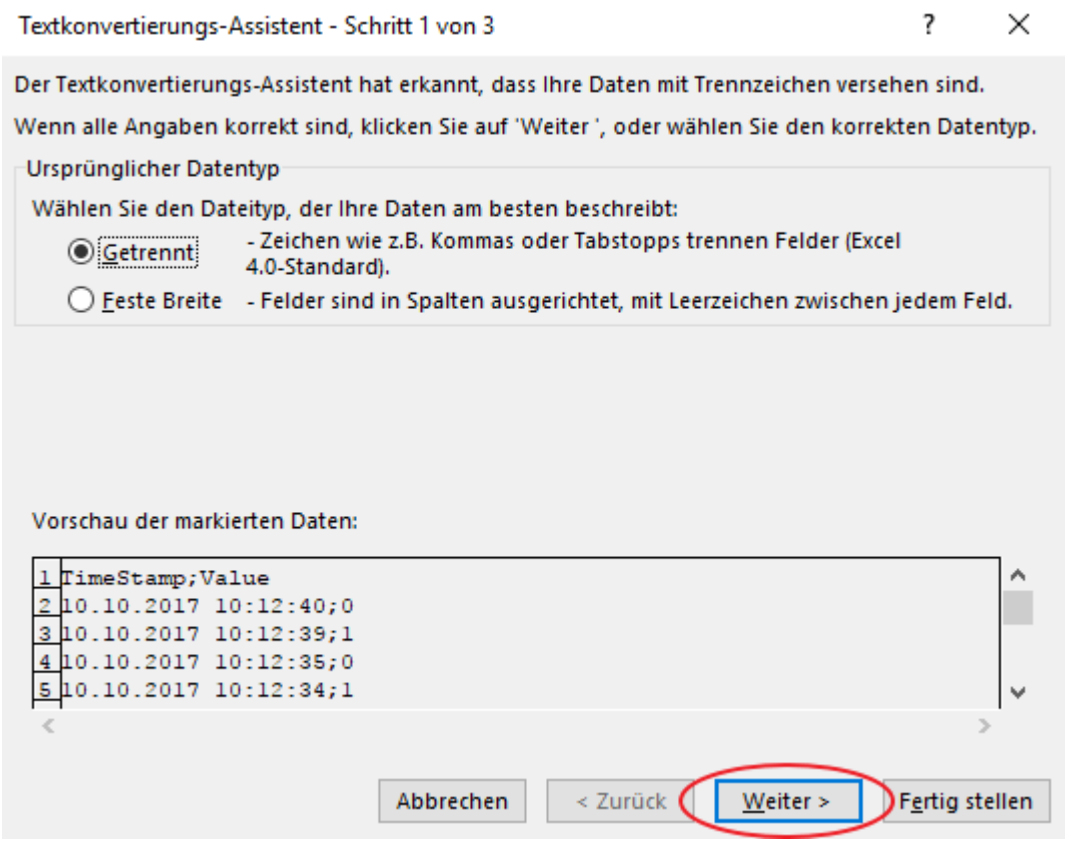
Um die Datensätze aufzuteilen wird die Funktion "Text in Spalten" genutzt.

Dazu muss der Reiter "Daten" ausgewählt und alle Datensätze markiert werden.
Anschließen auf den Button "Text in Spalten" klicken und es öffnet sich ein Dialog.

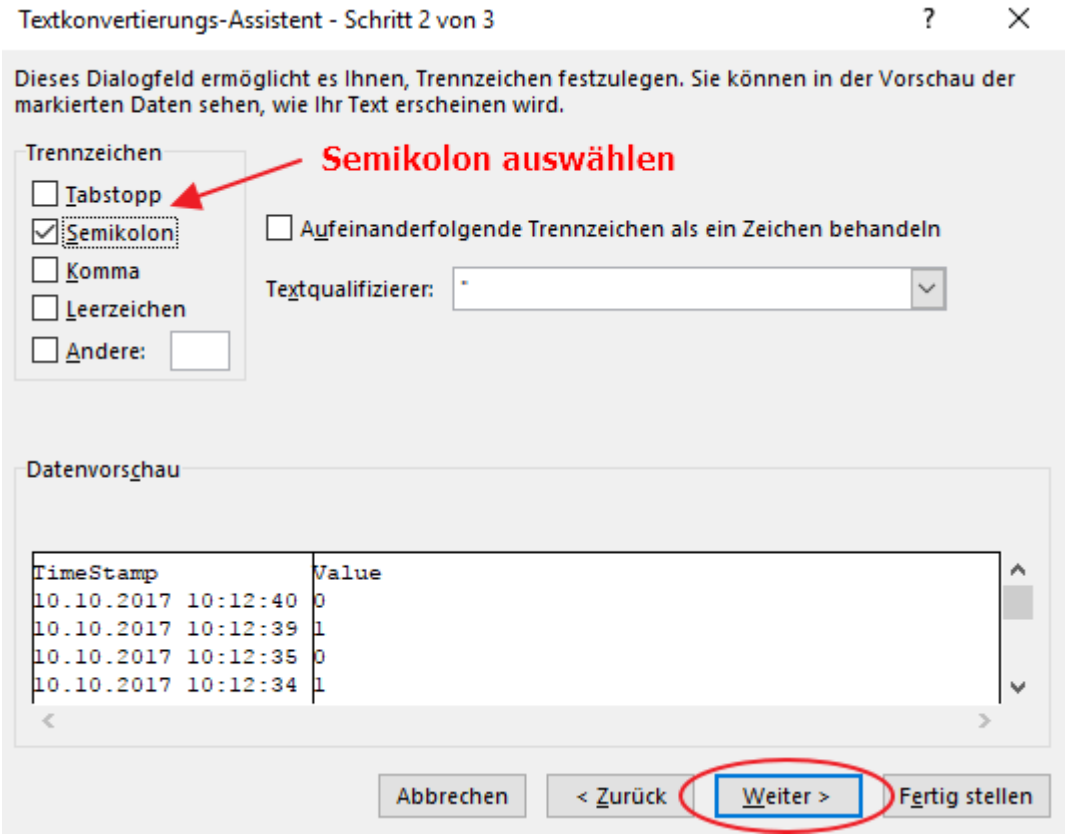


## SCHRITT 4 - DIALOG

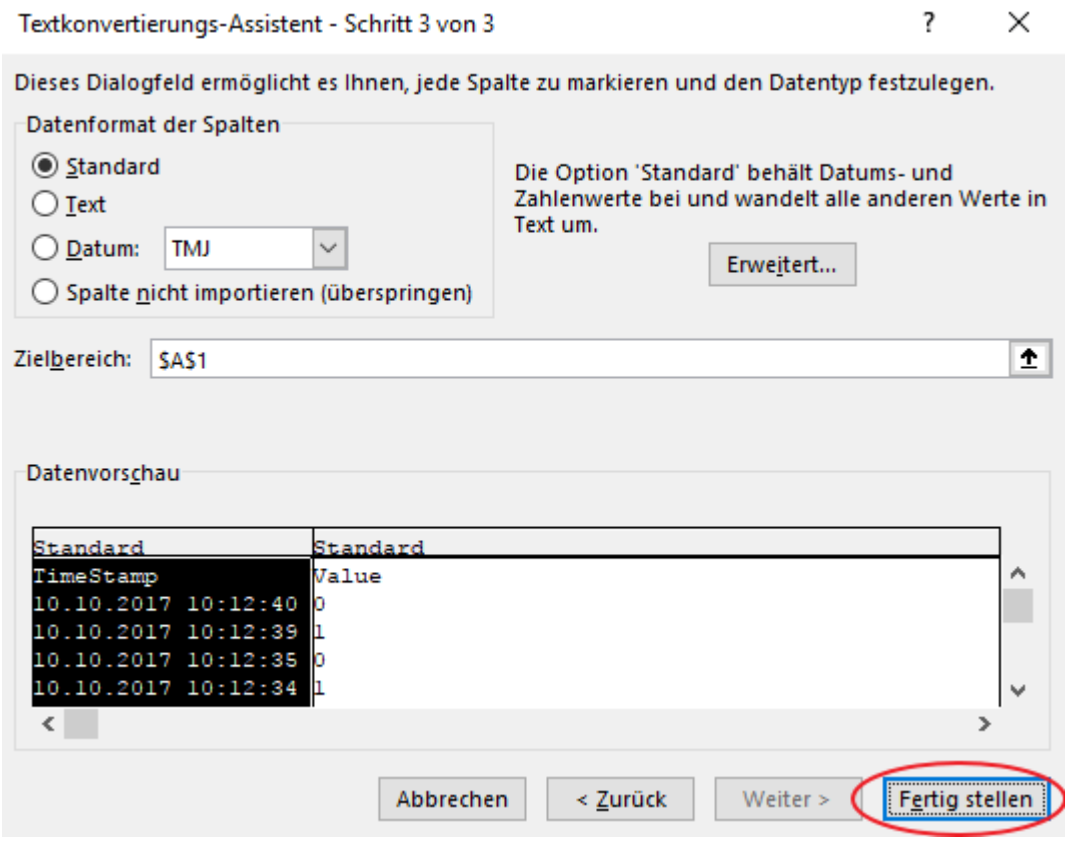
Der Dialog verläuft in 3 Schritten.




"Weiter" klicken.



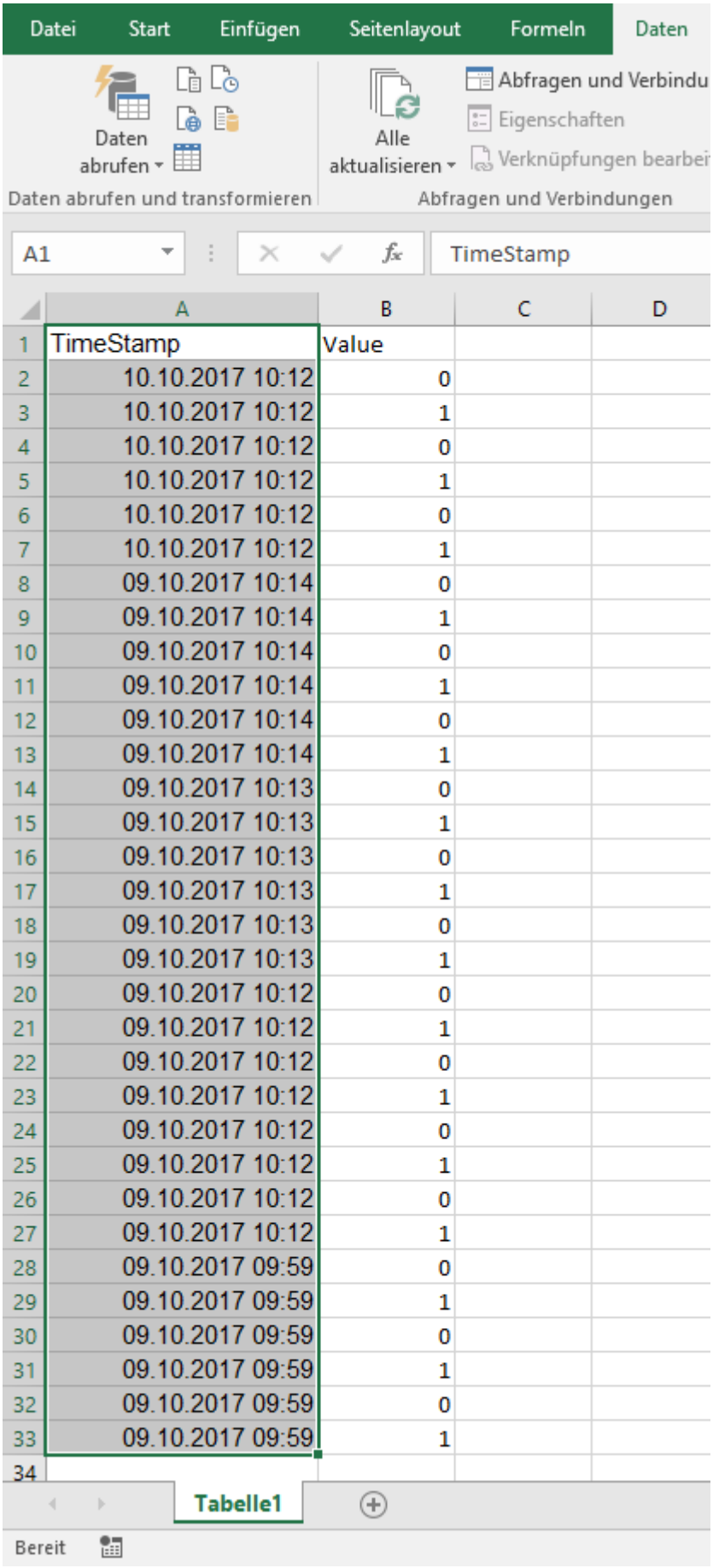
Semikolon auswählen, da die Datensätze durch dieses getrennt werden. "Weiter" klicken.




"Fertig stellen" klicken.

## ERGEBNIS

So sollte die Tabelle dann aussehen.



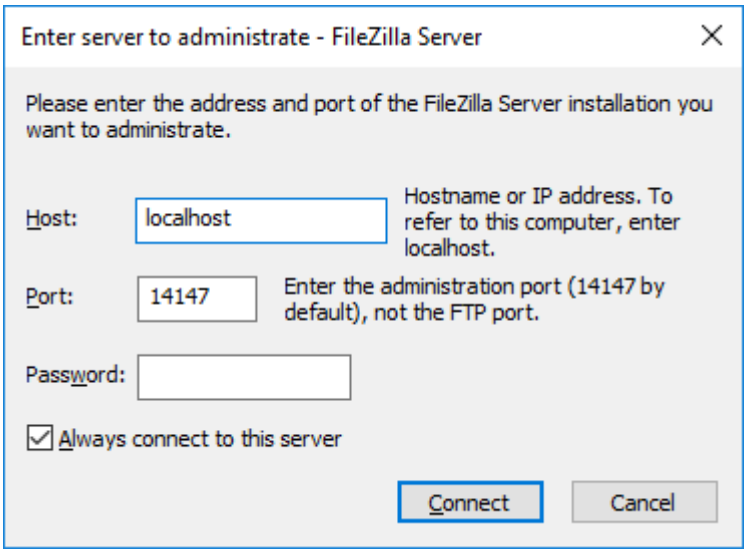

| | A | B | C | D |
|---|---|---|---|---|
| 1 | TimeStamp | Value | | |
| 2 | 10.10.2017 10:12 | 0 | | |
| 3 | 10.10.2017 10:12 | 1 | | |
| 4 | 10.10.2017 10:12 | 0 | | |
| 5 | 10.10.2017 10:12 | 1 | | |
| 6 | 10.10.2017 10:12 | 0 | | |
| 7 | 10.10.2017 10:12 | 1 | | |
| 8 | 09.10.2017 10:14 | 0 | | |
| 9 | 09.10.2017 10:14 | 1 | | |
| 10 | 09.10.2017 10:14 | 0 | | |
| 11 | 09.10.2017 10:14 | 1 | | |
| 12 | 09.10.2017 10:14 | 0 | | |
| 13 | 09.10.2017 10:14 | 1 | | |
| 14 | 09.10.2017 10:13 | 0 | | |
| 15 | 09.10.2017 10:13 | 1 | | |
| 16 | 09.10.2017 10:13 | 0 | | |
| 17 | 09.10.2017 10:13 | 1 | | |
| 18 | 09.10.2017 10:13 | 0 | | |
| 19 | 09.10.2017 10:13 | 1 | | |
| 20 | 09.10.2017 10:12 | 0 | | |
| 21 | 09.10.2017 10:12 | 1 | | |
| 22 | 09.10.2017 10:12 | 0 | | |
| 23 | 09.10.2017 10:12 | 1 | | |
| 24 | 09.10.2017 10:12 | 0 | | |
| 25 | 09.10.2017 10:12 | 1 | | |
| 26 | 09.10.2017 10:12 | 0 | | |
| 27 | 09.10.2017 10:12 | 1 | | |
| 28 | 09.10.2017 09:59 | 0 | | |
| 29 | 09.10.2017 09:59 | 1 | | |
| 30 | 09.10.2017 09:59 | 0 | | |
| 31 | 09.10.2017 09:59 | 1 | | |
| 32 | 09.10.2017 09:59 | 0 | | |
| 33 | 09.10.2017 09:59 | 1 | | |
| 34 | | | | |

# FTP benutzen

Dateien auf einem FTP Server können via Skript geöffnet werden.

## FTP SERVER VORBEREITEN

Falls noch nicht geschehen muss ein FTP Server aufgesetzt und konfiguriert werden. Die Beschreibung basiert auf FileZilla Server . Der Vorgang ist bei anderen FTP Tools ähnlich. Bereits während der Installation kann der FileZilla Server einen FTP Server auf dem localhost initialisieren.

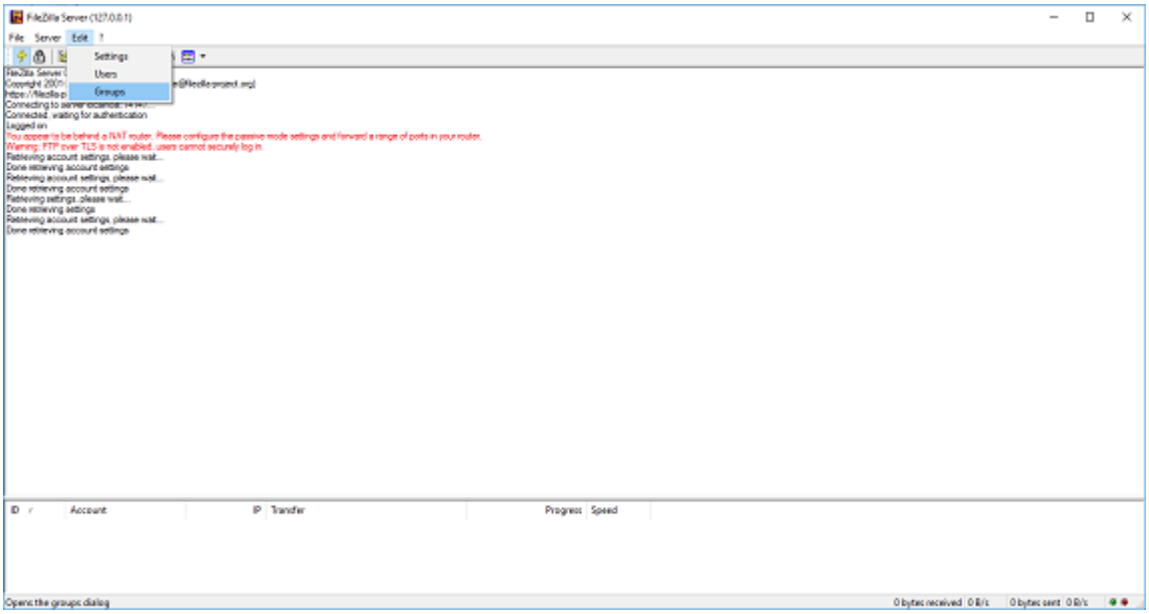


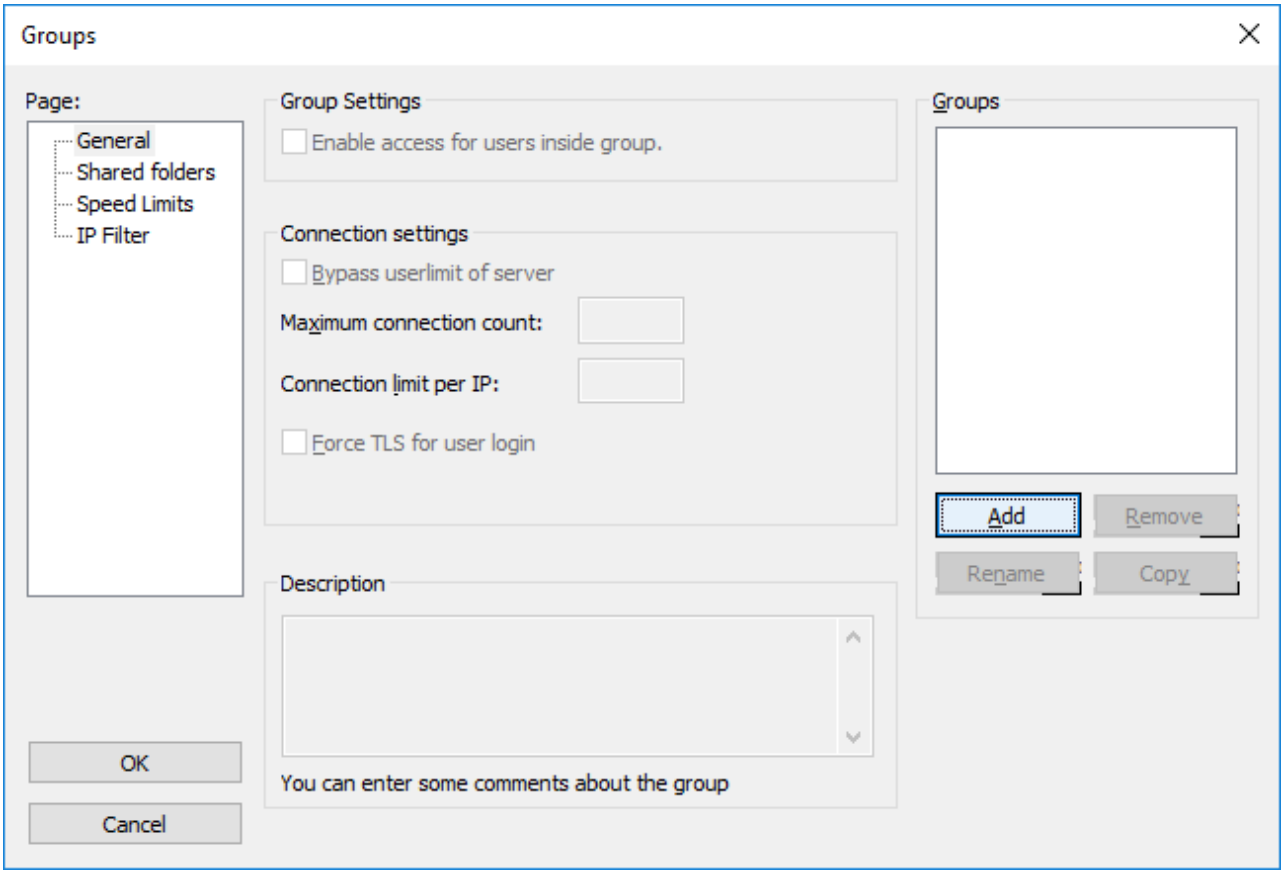

Nach dem Öffnen des Tools erscheint folgender Dialog. Wenn die Standardeinstellungen bei der Installation genutzt wurden, können folgende Daten genutzt werden: "Host": "localhost", "Port": 14147, und kein Passwort.

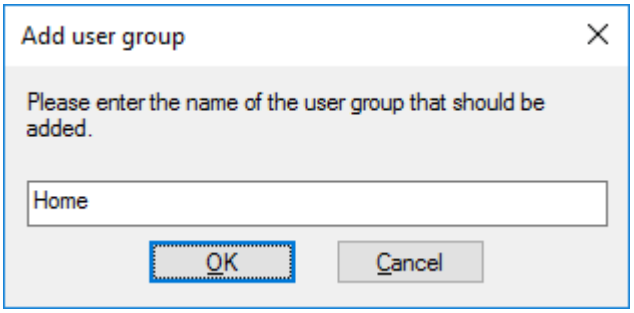
In den meisten Fällen muss noch die Firewall so konfiguriert werden, sodass diese den Zugriff auf den FTP erlaubt. Hier wird die Konfiguration beschrieben

### EINE GRUPPE ERSTELLEN

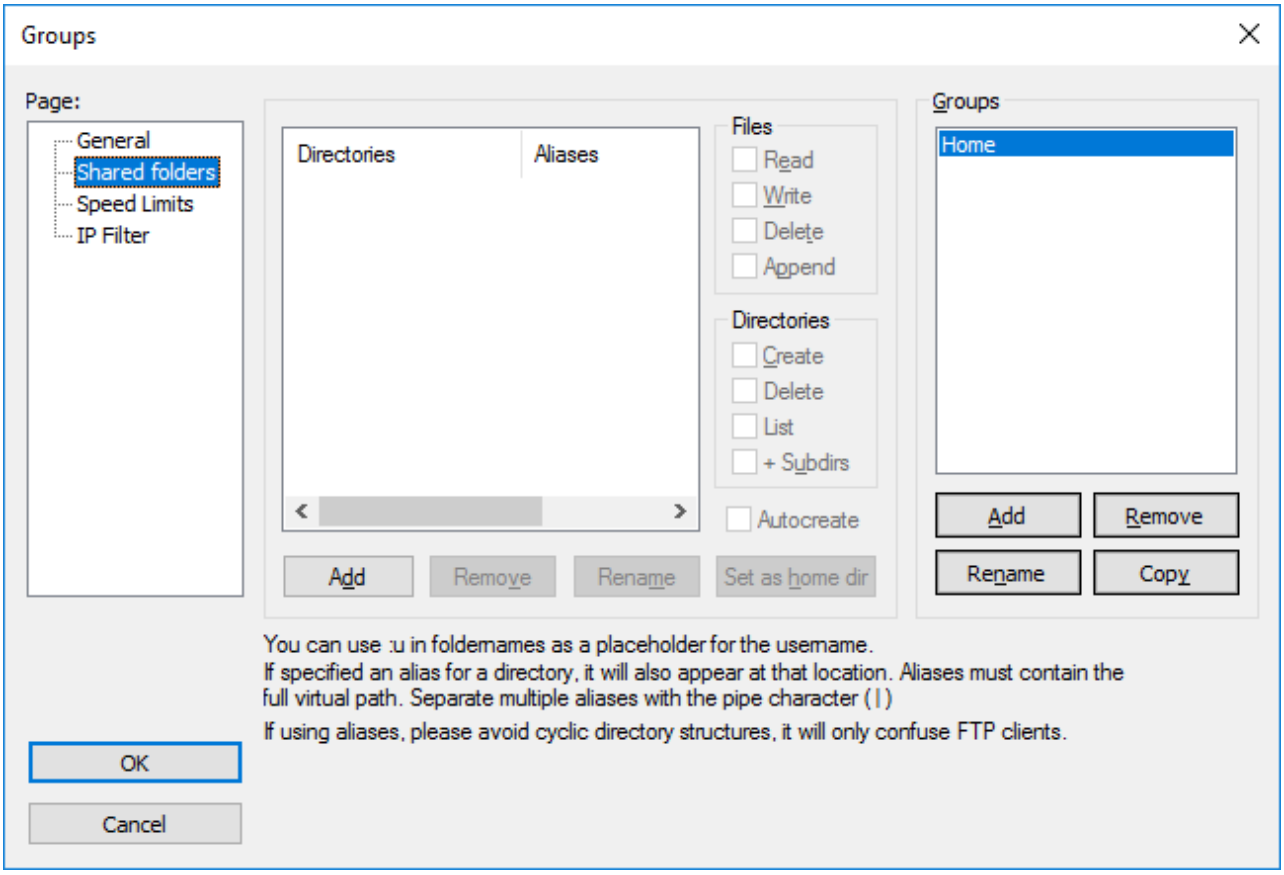
Der nächste Schritt ist das Erstellen einer Benutzergruppe. Benutzergruppen können genutzt werden um verschiedene Rechte und Zugriffsrechte zu vergeben. Es ist auch möglich eine allgemeingültige Benutzergruppe zu erstellen.

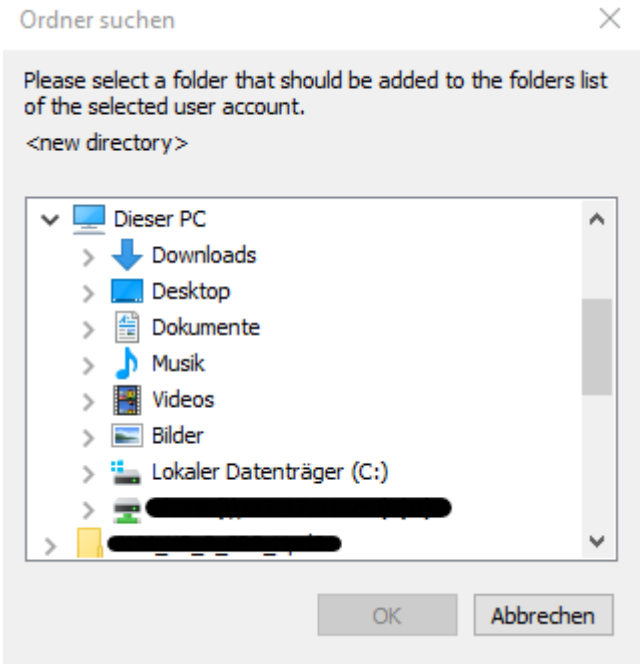


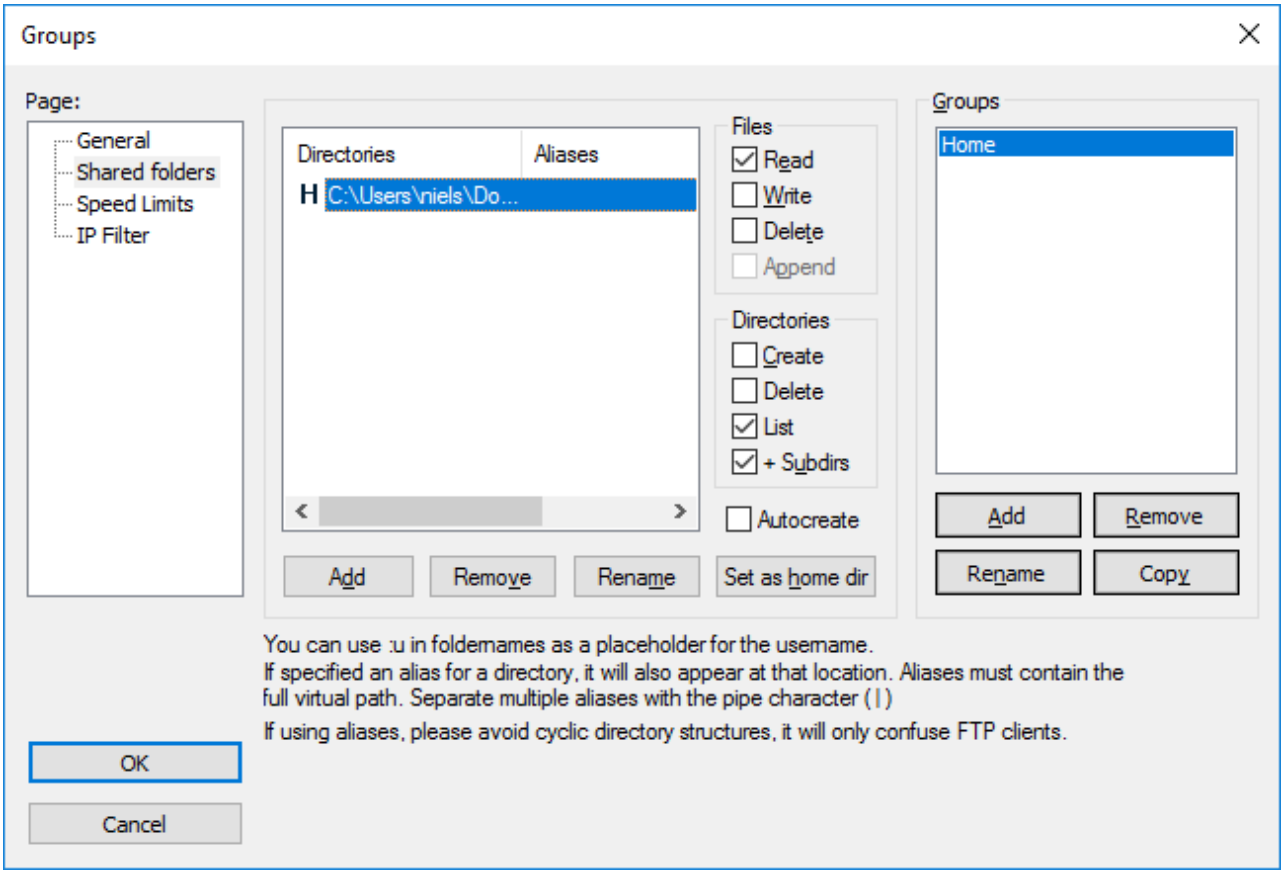




Die Benutzergruppeneinstellung kann via "Edit->Groups" verändert werden. Eine neue Benutzergruppe wird via "Add" hinzugefügt. Nach Eingabe eines Namens und Bestätigung ist die neue Benutzergruppe erstellt.



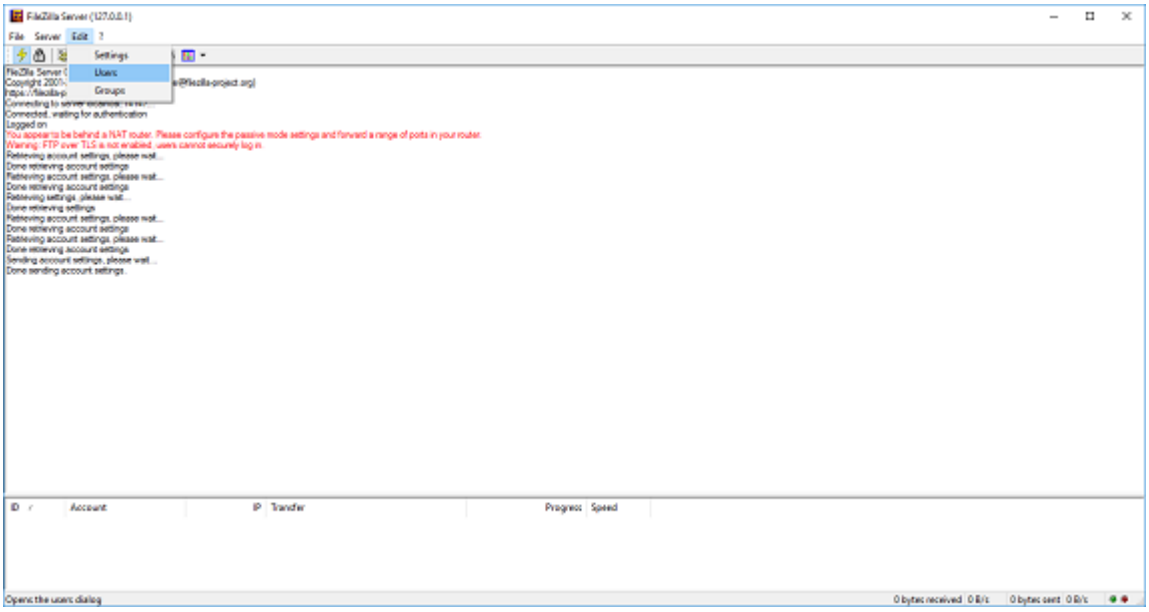



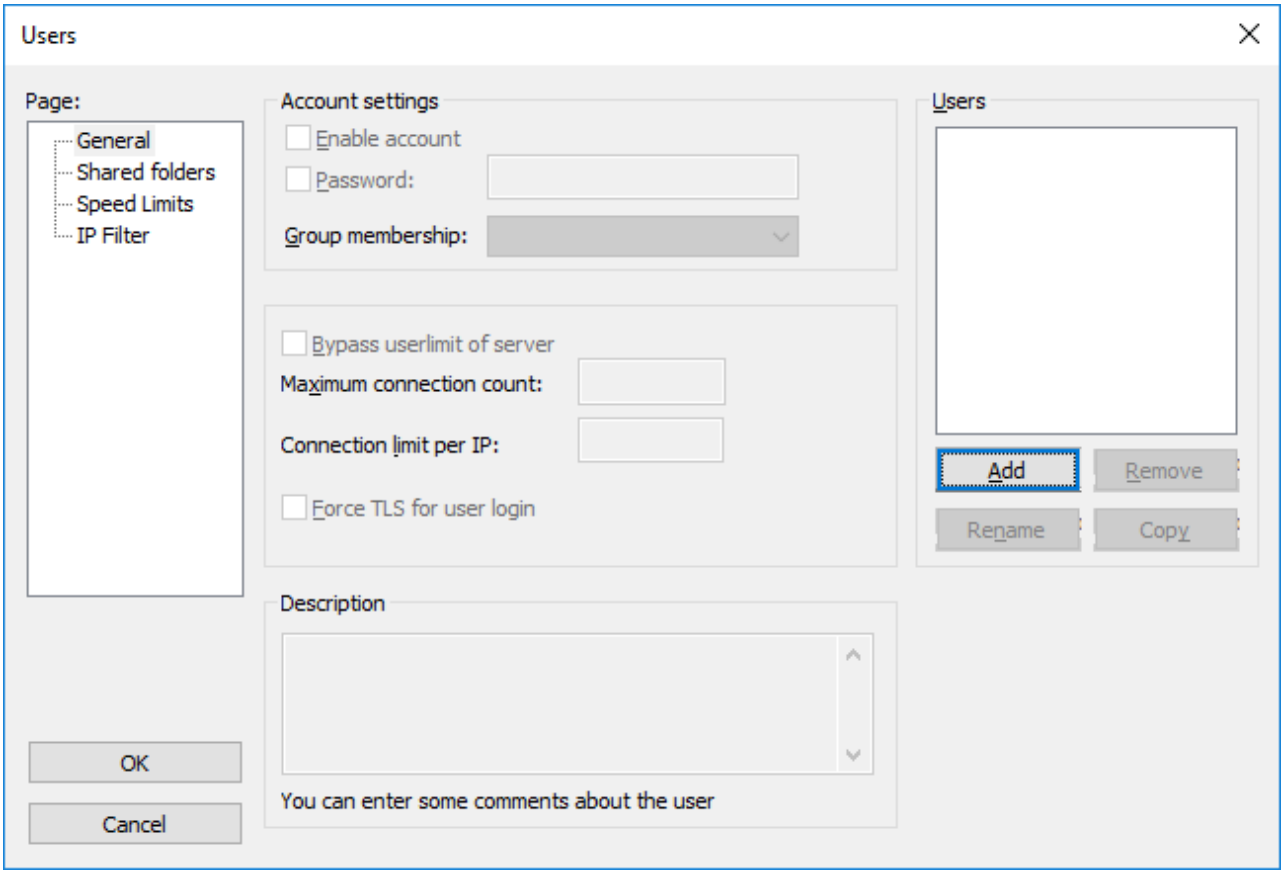

In der Kategorie "Shared folders" kann der Ordner ausgewählt werden auf welchen via FTP zugegriffen werden soll. Mit "Add" und dem darauffolgenden Dialog kann dieser hinzugefügt werden.

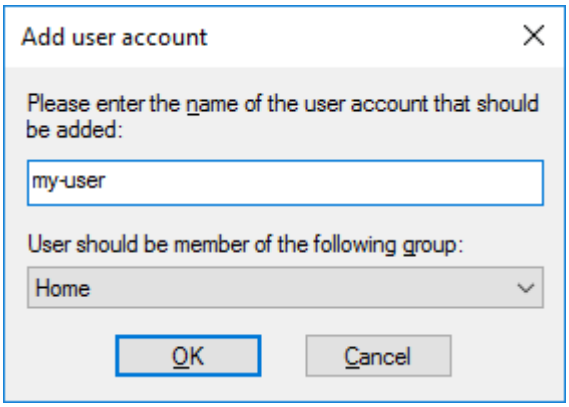



Nachdem ein gemeinsamer Ordner ausgewählt wurde können die Zugriffsrechte eingestellt werden. Standardmäßig ist nur Lesen und nicht Schreiben erlaubt.

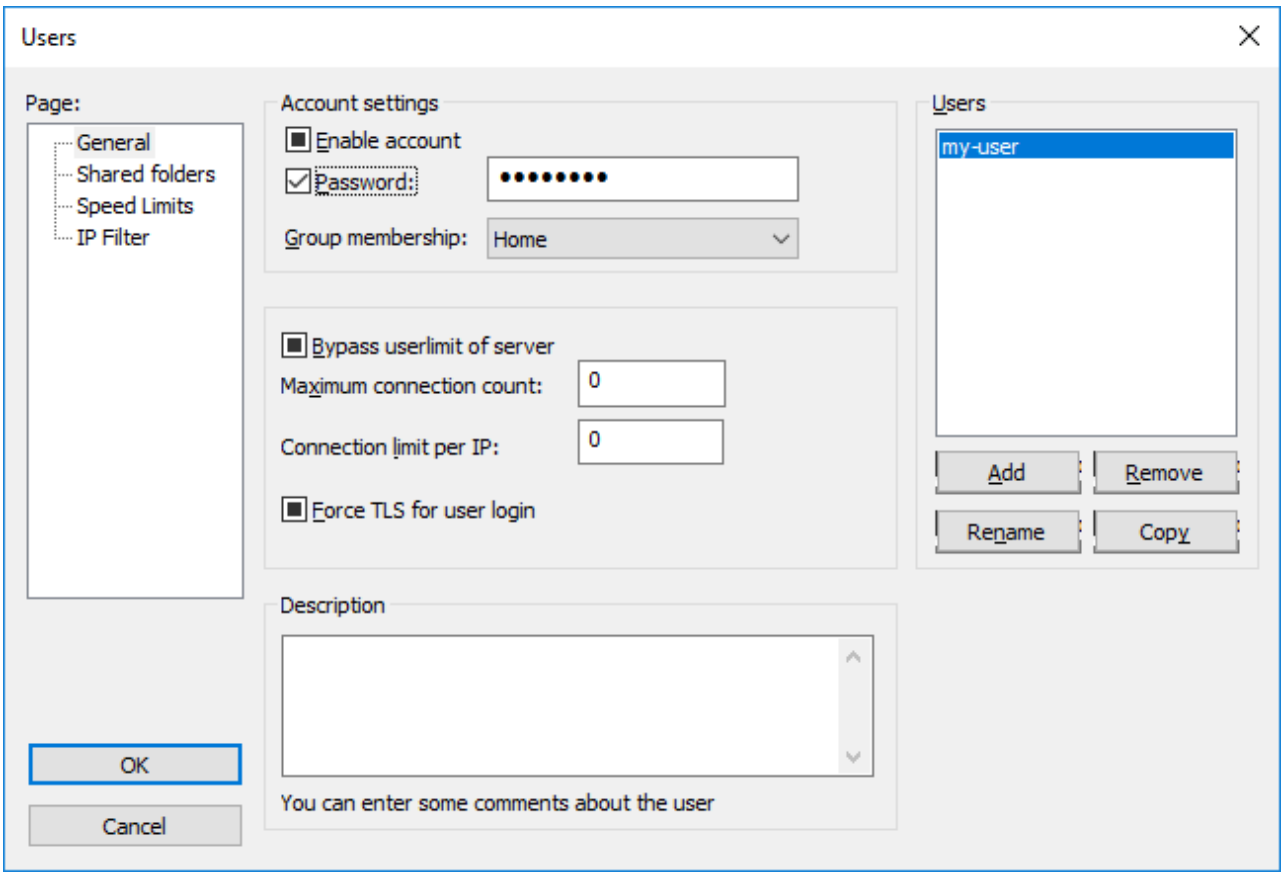
## BENUTZER ANLEGEN






Die Benutzer können über die Benutzerkonfiguration verwaltet werden. Diese kann via "Edit->Users" aufgerufen werden. Ein neuer Benutzer kann mit "Add" hinzugefügt werden. Im Dialog können dann Benutzername und Benutzergruppe ausgewählt werden. Nach Bestätigung wird der Benutzer erstellt.




Ein Benutzerpasswort kann bei Aktivierung der "Password" Checkbox eingetragen werden.



Wenn nur ein Benutzer existiert kann das Erstellen einer Benutzergruppe übersprungen werden. Stattdessen wird der Benutzer dem gemeinsamen Ordner zugewiesen.

## DATEIZUGRIFF AUF FTP VIA SKRIPT

```
// Inhalt auslesen
$adresse = "ftp://my-user:password@localhost/test.txt";
$inhalt = file_get_contents($adresse);
```

```
echo $inhalt;
// Inhalt schreiben
$zielAdresse = "ftp://my-user:password@localhost/test.txt";
$neuerInhalt = "Hallo Welt! Dies ist der neue Inhalt.";
file_put_contents($zielAdresse, $neuerInhalt);
```

Eine Datei auf dem FTP Server kann wie eine lokale Datei geöffnet werden. Nur die Adressierung ändert sich. Das Beispiel liest Datei "text.txt" ein und gibt diese aus. Die Adresse ist folgendermaßen strukturiert: "ftp://<Benutzername>(:<Passwort>)@<Servername>/<Dateipfad>".



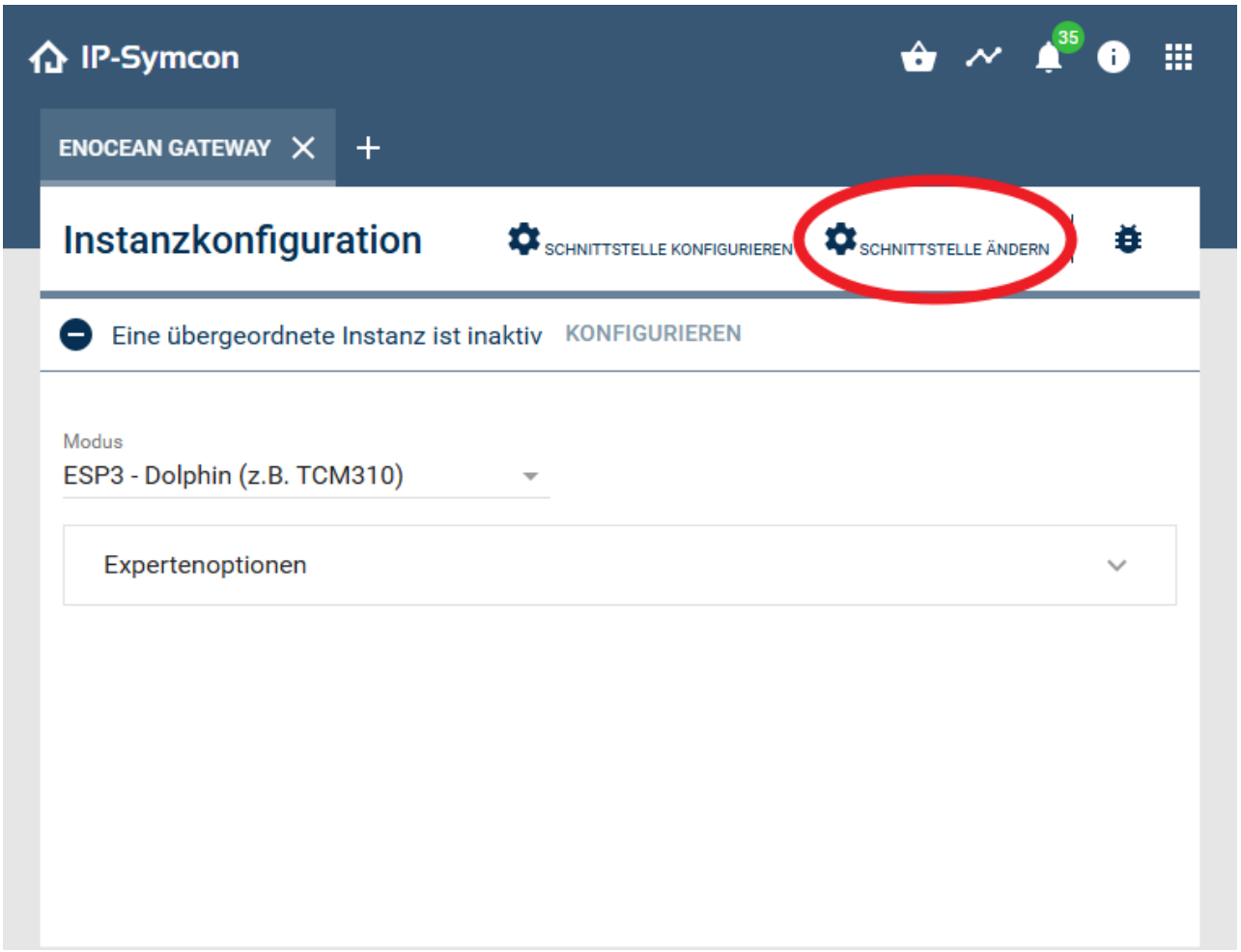
Es ist auch möglich spezielle PHP Funktionen zu nutzen, welche hier beschrieben sind.

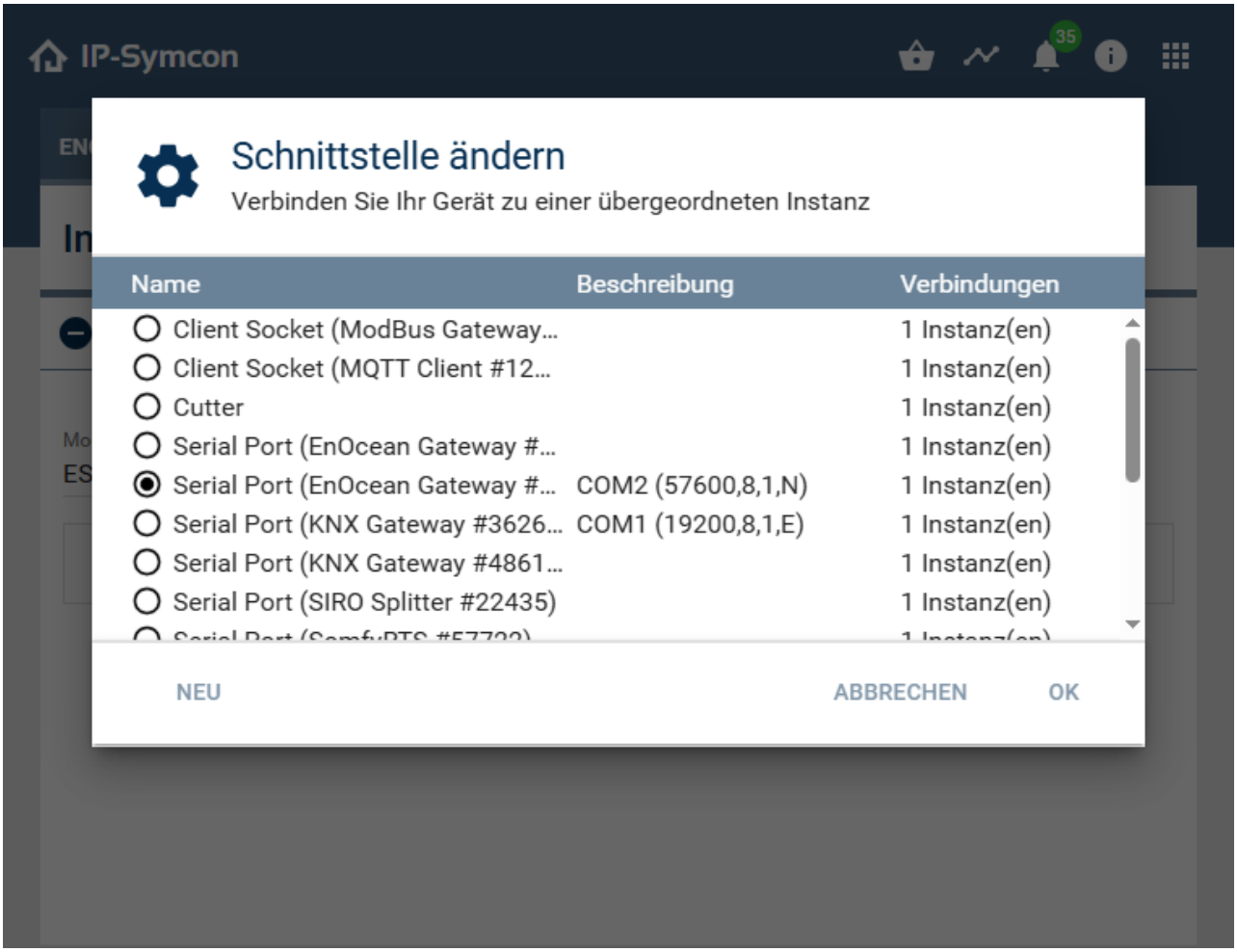
# I/O eines Gateways austauschen

Manchmal ist es nötig die I/O Instanz eines Gateways auszutauschen.
Dies wäre dann der Fall, wenn z.B. statt einer seriellen, eine TCP basierende Kommunikation verwendet werden soll. Oder ein anderes/neues Gateway zum Einsatz kommt.

> Eine Liste der verschiedenen I/O's befindet sich unter Instanzen -> Verbindungskomponenten .

Das Gateway zur jeweiligen Geräte-Instanz erreicht man am einfachsten über "Gateway konfigurieren" im oberen Bereich der Instanzkonfiguration.
In der Instanzkonfiguration des Gateways kann nun die übergeordnete Instanz, in diesem Fall die I/O Instanz, über "Schnittstelle ändern" im oberen Bereich angepasst werden.
In dem erscheinenden Dialog kann ein existierender kompatibler I/O ausgewählt oder alternativ per "Neu" ein neuer I/O erstellt werden.



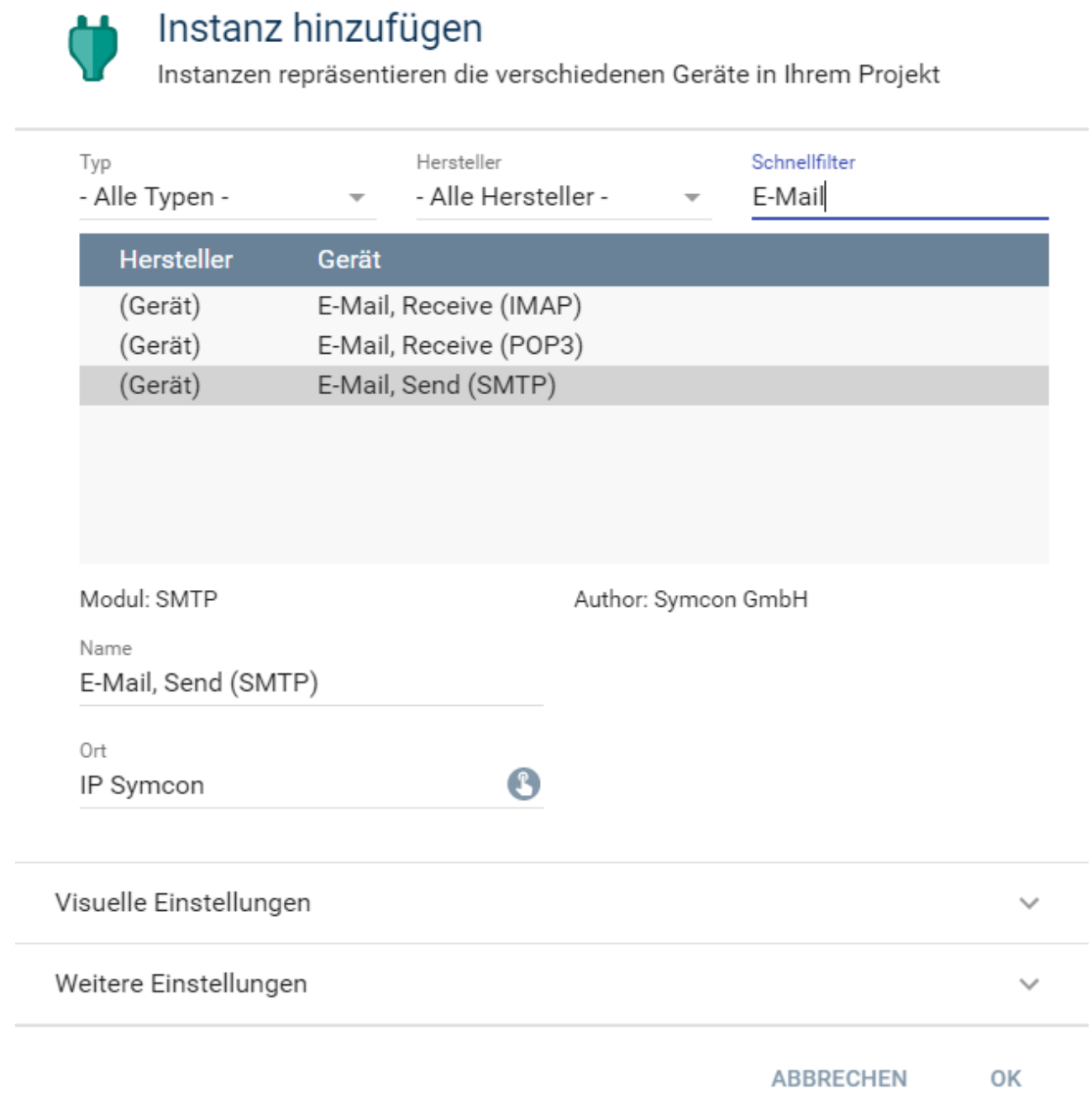

IP-Symcon
35
Schnittstelle ändern
Verbinden Sie Ihr Gerät zu einer übergeordneten Instanz
Name
Beschreibung
Verbindungen
Client Socket (ModBus Gateway...
1 Instanz(en)
Client Socket (MQTT Client #12...
1 Instanz(en)
Cutter
1 Instanz(en)
Serial Port (EnOcean Gateway #...
1 Instanz(en)
Serial Port (EnOcean Gateway #...
COM2 (57600,8,1,N)
1 Instanz(en)
Serial Port (KNX Gateway #3626...
COM1 (19200,8,1,E)
1 Instanz(en)
Serial Port (KNX Gateway #4861...
1 Instanz(en)
Serial Port (SIRO Splitter #22435)
1 Instanz(en)
NEU
ABBRECHEN
OK

# Email Benachrichtigung

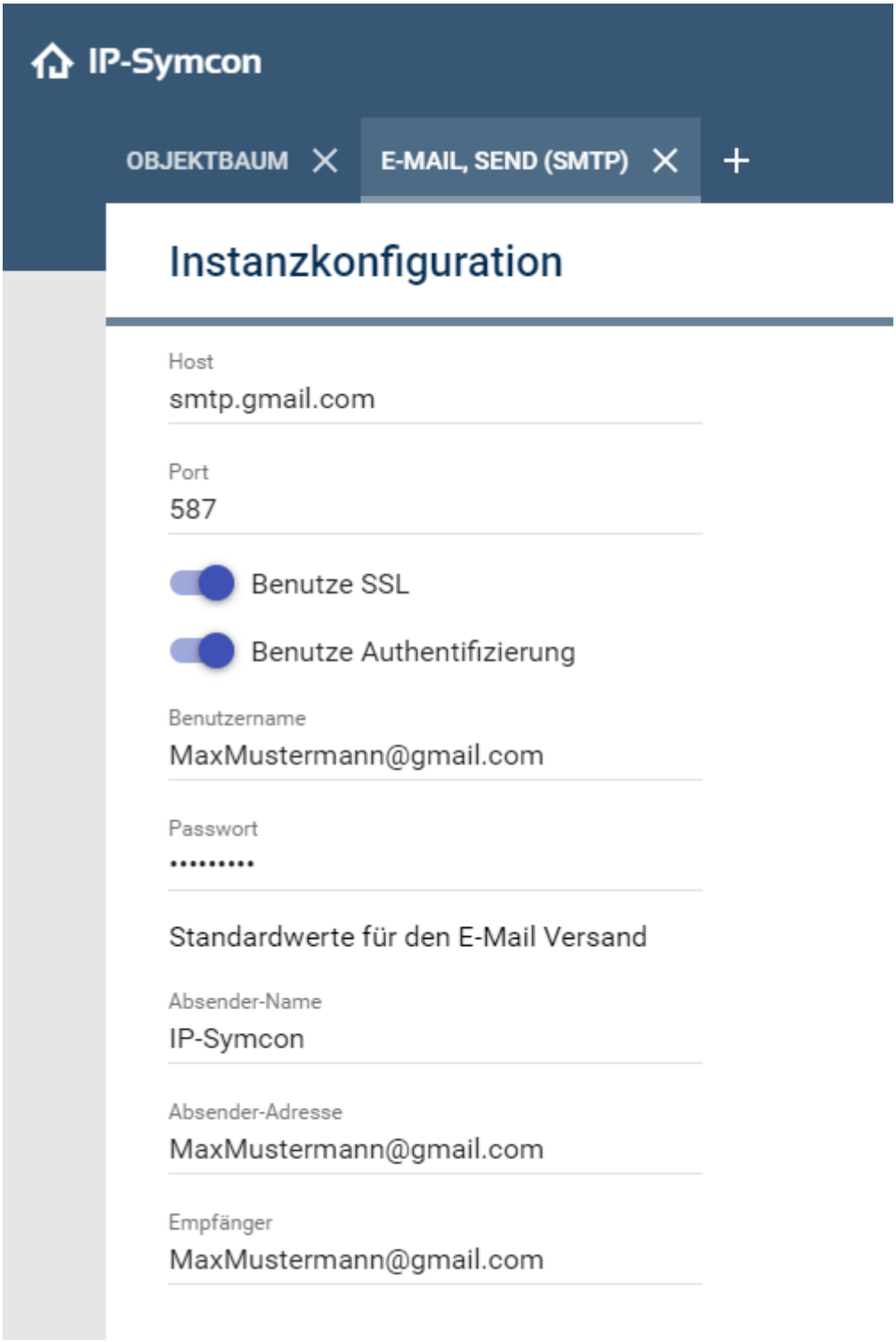
IP-Symcon beinhaltet die Möglichkeit E-Mails via SMTP-Instanz zu versenden. Dies bietet beispielsweise die Möglichkeit zu informieren, ob im SmartHome die Haustür oder ein Fenster (Sensoren vorausgesetzt) geöffnet wurde, ob eine bestimmte Person anwesend ist (Anwesenheits-Kontroll-Modul vorausgesetzt) oder der allgemeine Status von verschiedensten Daten und Verbräuchen. Dies kann einerseits für eine erhöhte Sicherheit (z.B., wenn sich ein Einbrecher an einem Fenster oder der Tür zu schaffen macht) oder aber andererseits auch für eine Steigerung des Komforts (z.B. allgemeine SmartHome Informationen) sorgen. Auch Benachrichtigungen zu einem bestimmten Zeitpunkt (z.B. als Erinnerung, monatlicher Bericht) kann verwirklicht werden.

## INSTANZ ERSTELLEN

Um eine E-Mail zu einem bestimmten Zeitpunkt zu senden, muss zuerst eine Instanz erstellt werden. Instanzen repräsentieren z.B. Geräte, die an IP-Symcon angeschlossen sind. Diese können sowohl konfiguriert werden, als auch Befehle erhalten.
Eine neue SMTP-Instanz kann erstellt werden, indem im Objektbaum der Verwaltungskonsole auf "+" geklickt oder per Kontextmenü der Punkt "Objekt hinzufügen" -> "Instanz" ausgewählt wird. Die zweite Möglichkeit hat den Vorteil, dass die Instanz direkt unterhalb des von ausgewählten Objekts im Objektbaum erstellt wird und nicht hinterher einsortiert werden muss.
Im Menü "Instanz hinzufügen" kann über den Schnellfilter nach "E-Mail" gesucht werden und "E-Mail, Send (SMTP)" hinzugefügt werden.




Innerhalb der Konfigurationsseite "E-Mail, Send (SMTP)" müssen die persönlichen Zugangsdaten eintragen werden.




"Host", "Port" und "Benutze SSL" (Verschlüsselung) müssen konfiguriert werden. Beim jeweiligen Provider kann nach den korrekten Einstellungen (Stichwort: SMTP) gesucht werden.
Unter "Benutze Authentifizierung" müssen Benutzername und Passwort des E-Mail-Postfachs eingetragen werden. Der "Absender-Name" kann frei gewählt werden. Dieser wird später bei Empfängerpostfach als Absender angezeigt.
Bei "Absender-Adresse" und bei "Empfänger" müssen E-Mail-Adressen eingetragen werden.



Bei der "Absender-Adresse" handelt es sich um die Adresse, die später bei E-Mail-Eingang als Absender-Adresse angezeigt wird, während "Empfänger" die Adresse darstellt, an die IP-Symcon eine Nachricht verschickt.

Nachdem alles korrekt ausgefüllt wurde, muss die Konfiguration mit "Übernehmen" gespeichert werden. Nun kann innerhalb der Testumgebung, im unteren Bereich der Konfigurationsseite, eine Testnachricht an die "Empfänger" E-Mail-Adresse geschickt werden.

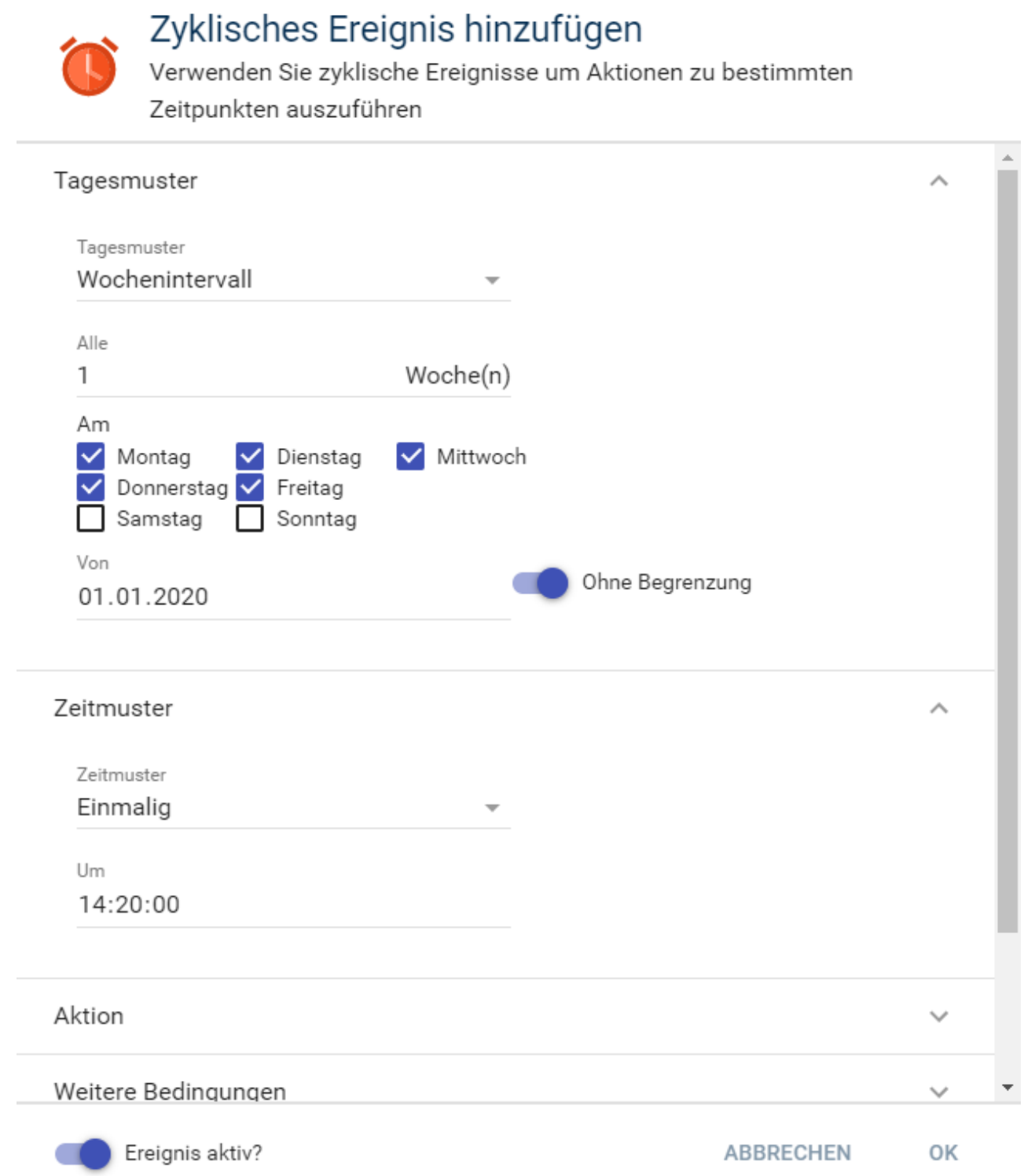
# BEISPIELE

## 1. E-MAIL ZU EINER BESTIMMTEN ZEIT

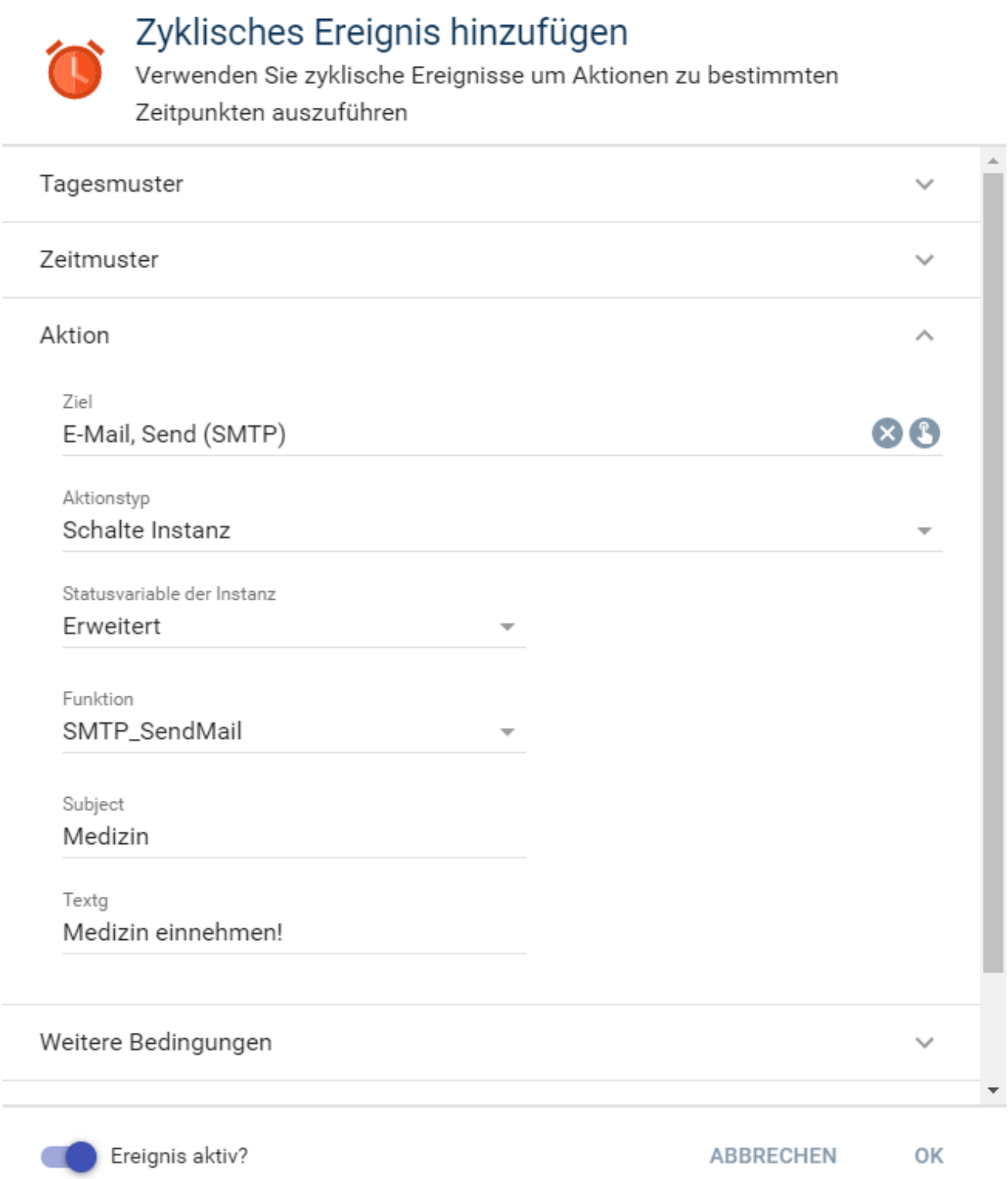
In diesem Beispiel soll zu einem bestimmten Zeitpunkt eine E-Mail zur Erinnerung versandt werden.

### EREIGNIS ERSTELLEN

Es muss ein zyklisches Ereignis erstellt werden. Ereignisse können bestimmte Vorgänge bei bestimmten Bedingungen oder Zeitpunkten starten.
Ein neues Ereignis kann erstellt werden, indem im Objektbaum der Verwaltungskonsole auf "+" geklickt oder per Kontextmenü den Punkt "Objekt hinzufügen" -> "Ereignis" -> "Zyklisch" ausgewählt wird. Die zweite Möglichkeit hat den Vorteil, dass die Instanz direkt unterhalb des ausgewählten Objekts im Objektbaum erstellt wird und nicht hinterher einsortiert werden muss.




Nun muss genau eingetragen werden, wann die Erinnerung per E-Mail erfolgen soll. Dies ist sinnvoll, wenn beispielsweise während der Arbeit im Büro (Montag bis Freitag) per E-Mail daran erinnert werden soll, dass nach der Mittagspause (um 14:20 Uhr) ein bestimmtes Medikament eingenommen werden muss.




Es muss bei "Aktion" "Schalte Instanz" und bei "Ziel" die zuvor erstellte SMTP-Instanz ausgewählt werden. Bei "Funktion" muss "SMTP_SendMail" ausgewählt werden, da ansonsten keine E-Mail gesendet wird. Bei Parameter kann nun die Nachricht eingegeben werden, die zum ausgewählten Zeitpunkt per E-Mail verschickt werden soll. Es wird nun von Montag bis Freitag täglich eine E-Mail versendet, die daran erinnert Medikamente einzunehmen.

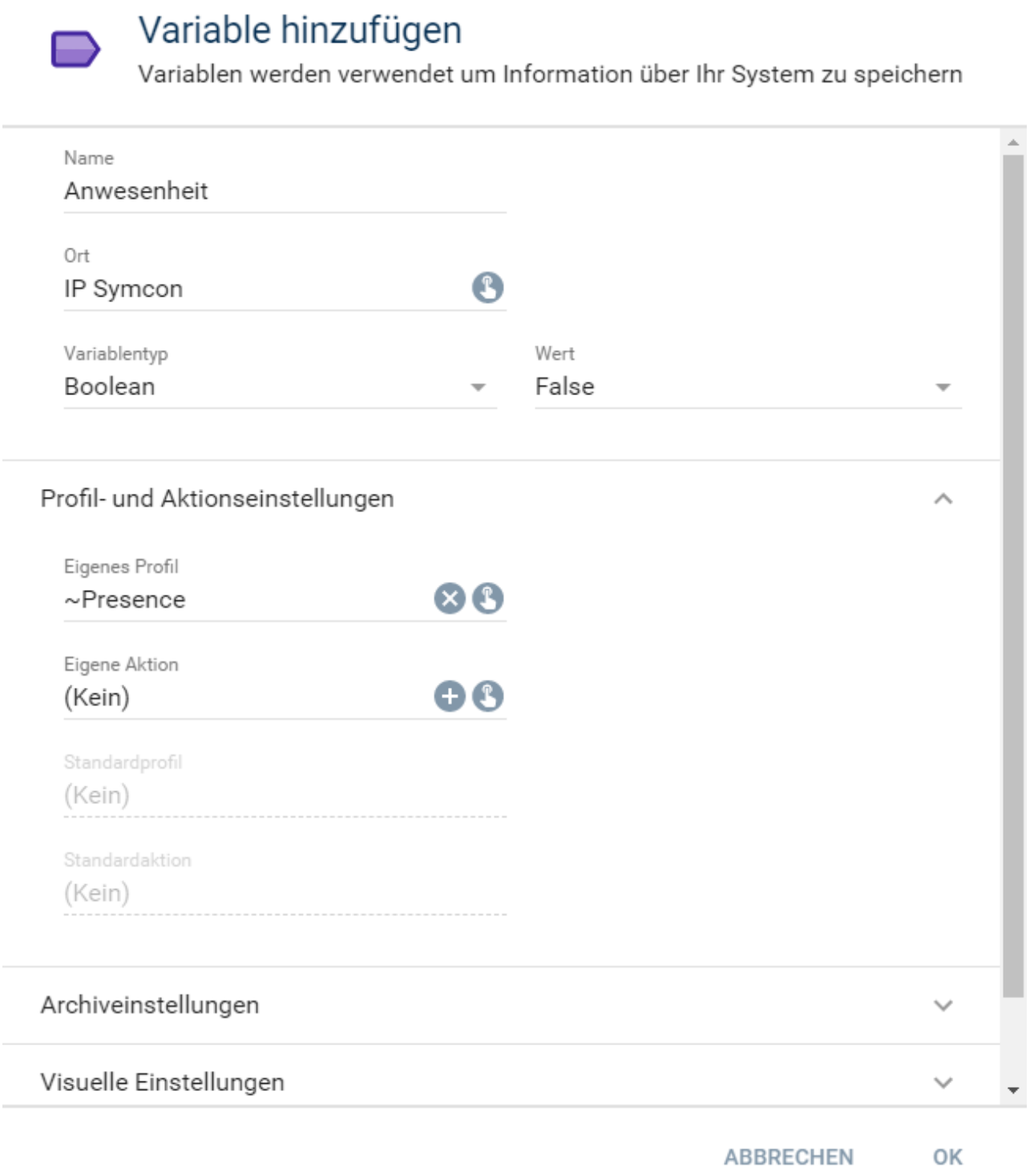
## 2. E-MAIL BEI BESTIMMTEM EREIGNIS

In diesem Beispiel soll eine E-Mail versendet werden, welche an ein bestimmtes Ereignis geknüpft ist.

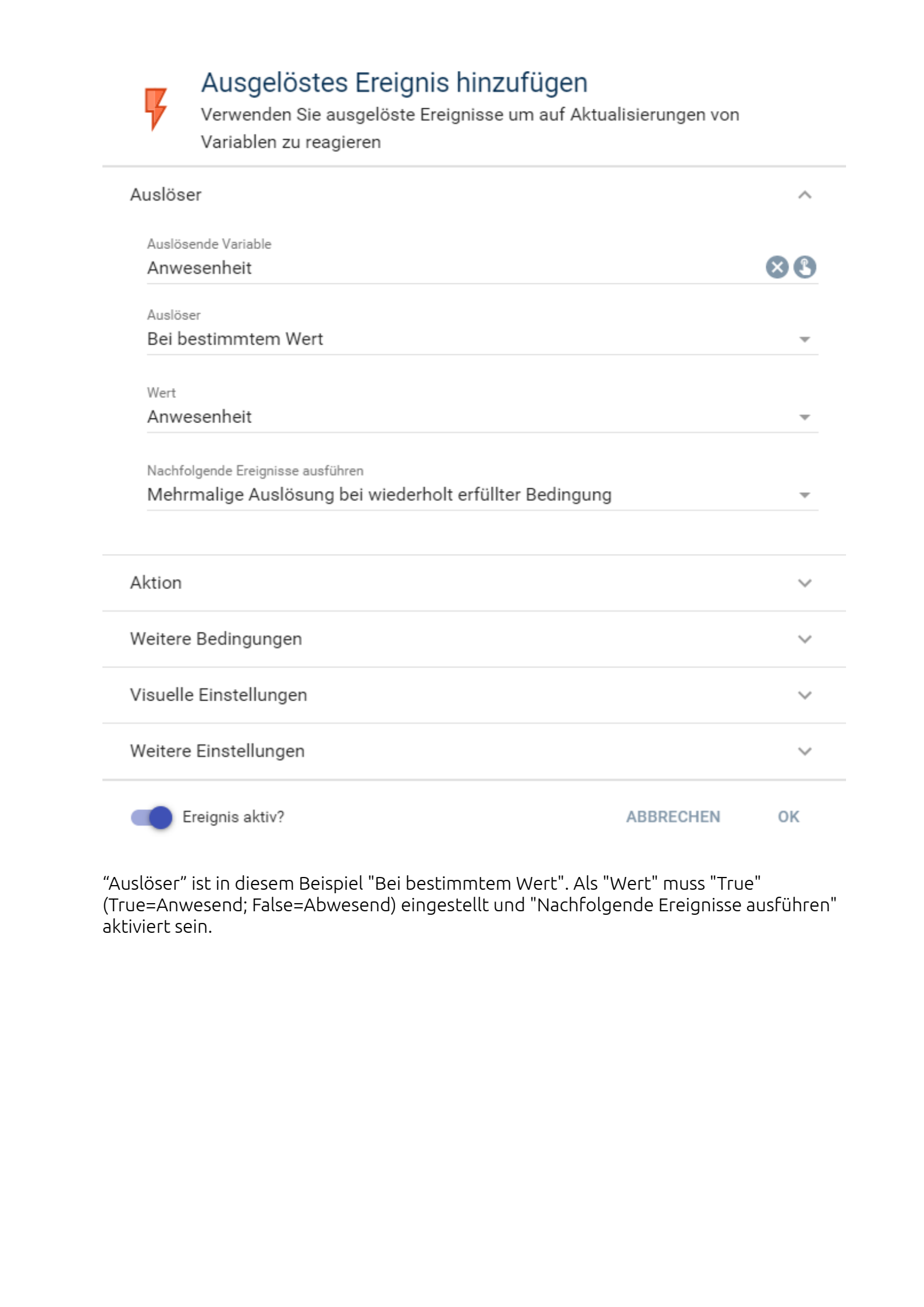
Es soll z.B. unterwegs eine E-Mail gesendet werden, wenn unerlaubterweise ein Fenster oder Tür geöffnet wurde und möglicherweise ein Einbrecher am Werk ist.

## VARIABLE UND EREIGNIS ERSTELLEN

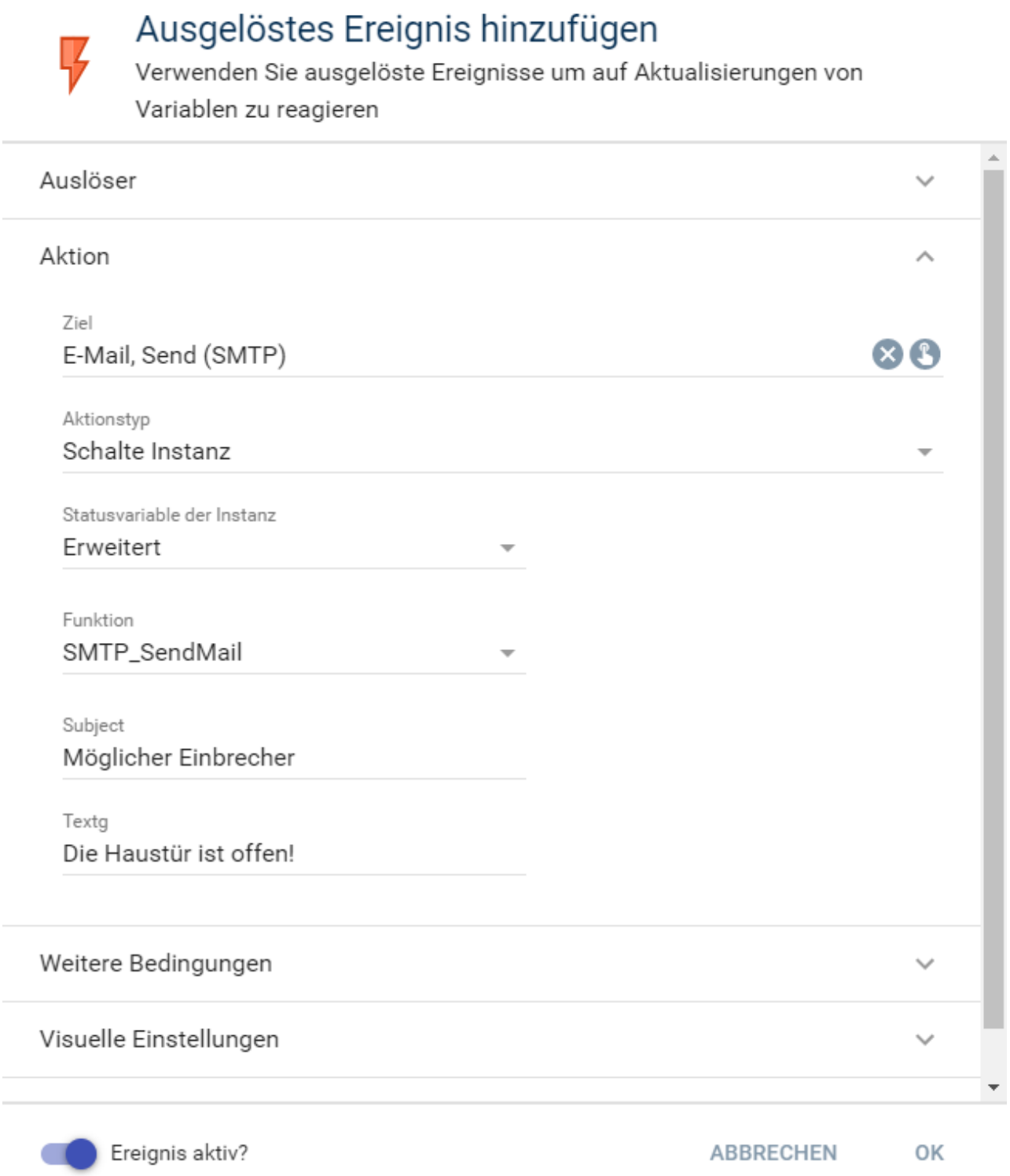
Zur Verwirklichung des Beispiels kann ein Ereignis an eine bestehende Statusvariable oder eine selbsterstellte Variable geknüpft werden. Variablen sind Datenhalter, die es erst ermöglichen, dass ein An- bzw. Ausschalten (z.B. des Türsensors) funktioniert und im WebFront angezeigt wird.
In diesem Beispiel stellt die selbsterstellte Variable einen imaginären Türsensor dar.
Eine neue Variablen kann angelegt werden, indem im Objektbaum der Verwaltungskonsole auf "+" geklickt oder per Kontextmenü den Punkt "Objekt hinzufügen" -> "Variable" -> “Variable hinzufügen” ausgewählt wird. Die zweite Möglichkeit hat den Vorteil, dass die Variable direkt unterhalb des von ausgewählten Objekts im Objektbaum erstellt wird und nicht hinterher einsortiert werden muss.
Im Menü "Variable hinzufügen" muss bei "Typ" nun "boolean" ausgewählt werden, denn es soll eine E-Mail verschickt werden, wenn der Türsensor aktiviert wird.




Danach kann als "Eigenes Profil" "~Presence" ausgewählt werden, da die Anwesenheit einer Person bzw eine geöffnete Tür im Haus angezeigt werden soll.
Dann muss ein Name für Variable gewählt werden. Es wird empfohlen sprechende Namen zu Wählen (z.B.: "Anwesenheit").
Nun muss anstatt ein "Zyklisches" ein "Ausgelöstes Ereignis" erstellt werden. In der Konfiguration des Ereignisses muss bei "Variable" die erstellte Anwesenheitsvariable ausgewählt werden.




“Auslöser” ist in diesem Beispiel "Bei bestimmtem Wert". Als "Wert" muss "True" (True=Anwesend; False=Abwesend) eingestellt und "Nachfolgende Ereignisse ausführen" aktiviert sein.




Es muss bei "Aktion" "Instanz schalten" und bei "Ziel" die zuvor erstellte SMTP-Instanz ausgewählt werden. Bei "Funktion" muss "SMTP_SendMail" ausgewählt werden, da ansonsten keine E-Mail gesendet wird. Im Parameter muss nun die Nachricht eingegeben werden, die zum ausgewählten Zeitpunkt per E-Mail verschickt werden soll. Nun wird eine Mail geschickt, wenn die Variable auf True gesetzt (imaginäre Tür geöffnet) wird.

# Modulreferenz

Was IP-Symcon ist, wird auf der Produktseite beschrieben.
Eine Tabelle zu allen unterstützten Systemen wird unter Schnittstellen aufgelistet.

In der Modulreferenz befinden sich die nativ unterstützten Module. In den jeweiligen Modulreferenzen befinden sich Anleitungen, Beispiele und Tipps & Tricks zur Einrichtung des jeweiligen Systems.
Desweiteren werden hier zur Übersicht die modulspezifischen Befehle aufgelistet.

Eine Auflistung der IP-Symcon spezifischen Befehle kann unter Befehlsreferenz gefunden werden.

## GERÄTE

Hardwarespezifische Module für Systeme, Aktoren und Sensoren

| | |
|---|---|
| 1-Wire | Funktionen zur Ansteuerung von 1-Wire Komponenten |
| DS2405 | 1-fach Switch |
| DS2406 | 2-fach Switch |
| DS2408 | 8-fach Switch |
| DS2413 | 2-fach Switch |
| DS2438 | Smart Battery Monitor |
| DS2450 | A/D-Wandler 16bit |
| DS2890 | Digital-Potentiometer 8 bit |
| Geräteliste | Geräteliste 1-Wire |
| OW_RequestStatus | sende eine Leseanfragen an das OneWire Gerät |
| ABL | Einrichtung von ABL Geräten |
| Geräteliste | |
| Alfen | Einrichtung von Alfen Geräten |
| Geräteliste | |
| ALLNET | Funktionen zur Ansteuerung von ALLNET Komponenten |
| ALL_SetAnalog | setzt den Analogwert eines Kanals |
| ALL_SwitchActor | schaltet ein ALLNET Aktor ein/aus |
| ALL_SwitchMode | schaltet ein ALLNET Gerät ein/aus |
| ALL_UpdateValues | sende eine Aktualisierungsanfrage an das ALLnet Gerät |
| Geräteliste | Geräteliste ALLNET |
| BACnet | Funktionen zur Ansteuerung von BACnet Komponenten |
| BAC_RelinquishPresetValue | Rücksetzung des PresetValues |
| digitalSTROM | Funktionen zur Ansteuerung von digitalSTROM Komponenten |
| DS_CallScene | ruft eine Szene auf |
| DS_DimSet | dimmt eine digitalSTROM Klemme auf einen bestimmten Wert |
| DS_MakeRequest | sendet einen digitalSTROM Befehl direkt an den dSS |
| DS_RequestStatus | ruft den Status ab |
| DS_ShutterMove | bewegt den Rollladen bis zu einer gewünschten Position/ Stop |
| DS_ShutterMoveDown | bewegt den Rollladen nach unten bis zur Endposition/Stop |
| DS_ShutterMoveUp | bewegt den Rollladen nach oben bis zur Endposition/Stop |
| DS_ShutterStop | stoppt einen Bewegungsvorgang |
| DS_SwitchMode | schaltet eine digitalSTROM Klemme an/aus |
| Geräteliste | Geräteliste digitalSTROM |
| DMX / ArtNet | Funktionen zur Ansteuerung von DMX Komponenten |
| DMX_FadeChannel | dimmt einen Kanal auf einen bestimmten Wert in einer bestimmten Zeit |
| DMX_FadeChannelDelayed | dimmt einen Kanal nach einer Verzögerung auf einen |

## LOGIKEN

Zusammenfassende Funktionsinstanzen für größeren Komfort

| | |
|---|---|
| Aktivliste | Übersicht aller aktiven Variablen in der Visualisierung |
| AL_SwitchOff | Schaltet alle in der Liste vorhandenen aktiven Variablen inaktiv. |
| Anwesenheits-Simulation | Anwesenheit über geloggte Werte simulieren |
| AS_SetSimulation | De-/Aktiviert die Anwesenheits-Simulation |
| Bild-Archiv | Das Modul kopiert bei Auslösung durch eine gewählte Variable ein Bild. |
| BA_AddImage | Kopiert das aktuelle Quellbild in die Kategorie "Bilder" des Moduls "BildArchiv" mit der InstanzID. |
| Countdown | |
| CSV ZIP Export | Exportiert aggregierte Werte einer Variable als CSV-Datei in einem ZIP-Archiv |
| CSV_DeleteZip | Entfernt die generierte Datei. |
| CSV_Export | Erzeugt ein Zip-Archiv basierend auf den gegebenen Parametern. |
| CSV_SendMail | Senden durch eine SMTP Instanz eine Mail mit einer erzeugten Zip-Datei |
| Dummy Modul | Platzhalter Modul zur Darstellung als Instanz in der Visualisierung |
| Eieruhr | Eine Eieruhr, die eine gewisse Zeit läuft und jederzeit wieder aufgezogen werden kann. Ideal für die Verwendung in Ereignissen und Skripten. |
| Gruppensteuerung | Mithilfe der Gruppensteuerungen können Variablen in Gruppen zusammen geschaltet werden. |
| JSON Decoder | Dekodiert JSON und stellt den Inhalt in Variablen zur Verfügung |
| JSON Exporter | Exportiert beliebige Variablen in einer benutzerdefinierten Struktur über einen HTTP Endpunkt. |
| Logik Gatter | Verknüpft Variablen durch logische Operationen |
| Rechenmodule | Module, welche verschiedene Berechnungen innerhalb von IP-Symcon ermöglicht |
| Rechen-Modul | Berechnung verschiedener Hilfswerte auf Basis einer Gruppe von Variablen, z.B. Summer oder Durchschnitt. |
| Umrechnen | Rechnet eine Variable über eine selbstdefinierte Formel um. |
| UmrechnenMultiGrenzen | Rechnet eine Variable in eingerichteten Grenzen über eine selbstdefinierte Formel um. |
| WertebereichSkalieren | Skalierung eines Wertes innerhalb eines Wertebereiches |
| RGBMultiplexer | Verbindet einzelne R, G, B Kanäle zu einer Variable, welche mit dem Farbrad angesteuert werden kann. |
| RGBM_RequestStatus | Ermittelt den Wert der einzelnen R, G, B Kanäle und setzt die Color Variable. |
| RGBM_SetRGB | Sendet die R, G, B Werte an die einzelnen Kanäle und aktualisiert die Color Variable |
| Schleppzeiger | Eine Variable, welche auf dem höchsten oder niedrigsten Wert bis zum Reset stehen bleibt. |

# ENERGIE

Module rund um das Thema Energie

| | |
|---|---|
| Arbeitszahl | Berechnung des Wirkungsgrades |
| ARZ_Calculation | Berechnet den Wirkungsgrad |
| Betriebsstundenzähler | Ermittelt und zeigt die Betriebszeit eines Gerätes an |
| BSZ_Calculate | Die Betriebsstunden-Variable wird auf den errechneten Wert gesetzt |
| Energie Dashboard | Eine Übersicht mit mehrern Diagrammen |
| Energie Manager | Module, welches die Verbraucher an und ausschaltet, wenn Energie übrig ist. |
| Energieverteilung | |
| Energiezähler | Berechnet den Verbrauch |
| Energiezähler Impuls | Rechnet Zählerwerte in einen Momentanverbrauch um. |
| Energiezähler Strom | Rechnet aus Strom/Leistung den momentanen und kumlativen Stromverbrauch. |
| Stromabrechnungsmodul | Das Modul liefert eine Kosten-Aufstellung ähnlich der Jahresabrechnung vom Energieversorger. |
| SAM_UpdateCalculations | Berechnet alle angezeigten Werte |
| Strompreis | Liest die aktuellen/vorhergesagten Strompreise von aWATTar, Tibber oder Epex Spot DE aus. |
| SPX_Update | Aktualisiert die Daten |
| Verbrauch in Kategorie | Zeigt den Verbrauch in Prozent in Kategorie. |
| VIK_CalculationConsumption | Berechnet die Kategorien neu |
| Verbrauch in Zeitspanne | Berechnet den Verbrauch zwischen einem Start- und End-Datum |
| VIZ_Calculate | Berechnet den Verbrauch zwischen Start- und End-Datum neu |
| Verbrauchsverhalten | Das Verbrauchsverhalten-Modul zeigt den erwarteten Verbrauch an. |
| VBV_UpdateCalculation | Die Funktion aktualisiert die prognostizierten Werte |
| VirtuelleMessstelle | Rechnet positive Variablenänderungen nach einstellbaren Regeln mit Veränderungen einer Hauptvariable zusammen und addiert dieses mit dem Wert eine Variable. |
| VM_Update | Berechnet die Steigung der sekundären Messstellen und addiert dieses mit dem Ergebnis |
| Virtueller Zähler | Eingabemaske für manuelle abgelesene Zählerstände. |
| VZ_WriteNewCounterValue | Setzt nach Überprüfung den aktuellen Zählerstand |
| Zählerstand (Tag) | Für ein ausgewähltes Datum kann der Zählerstand angezeigt werden |
| ZST_Calculate | Ermittelt den Wert für das gewählte Datum neu |
| Zählerüberlauf | Zähler, welche überlaufen können als Zähler dargestellt werden. |

## VISUALISIERUNGEN

Visualisierungen für Symcon.

| | |
|---|---|
| Kachel Visualisierung | Funktionen zum Ansteuern der Kachel Visualisierung auf den Client-Geräten |
| VISU_PostNotification | sendet eine Push-Benachrichtigung auf die Kachel Visualisierung |
| VISU_PostNotificationEx | sendet eine Push-Benachrichtigung auf die Kachel Visualisierung |
| WebFront Visualisierung | Funktionen zum Ansteuern des WebFronts auf den Client-Rechnern |
| WFC_AudioNotification | sendet eine Audio-Meldung auf das WebFront |
| WFC_OpenCategory | Öffnet im WebFront eine Kategorie als Popup |
| WFC_PushNotification | sendet eine Push-Nachricht an die mobilen Apps |
| WFC_Reload | lädt das WebFront auf dem Client-Rechner neu |
| WFC_SendNotification | sendet eine kleine Meldung auf das WebFront |
| WFC_SendPopup | sendet eine große Meldung an das WebFront |
| WFC_SwitchPage | wechselt den Reiter im WebFront |

## SPRACHASSISTENTEN

Sprachassistenten zum Einbinden

| | |
|---|---|
| Amazon Alexa | Modul zur Verknüpfung mit Amazon Alexa |
| Expertenoptionen | Expertenoptionen von Amazon Alexa |
| Fernsehgerät | Fernsehgerät in Amazon Alexa |
| Generischer Schalter | Generische Schalter in Amazon Alexa |
| Generischer Schieberegler | Generische Slider in Amazon Alexa |
| Lautsprecher | Lautsprecher in Amazon Alexa |
| Lautsprecher mit Stummschaltung | Lautsprecher mit Stummschaltung in Amazon Alexa |
| Licht (Dimmer) | Licht in Amazon Alexa, welches gedimmt werden kann |
| Licht (Experte) | Expertenlicht in Amazon Alexa |
| Licht (Farbe) | Farbiges Licht in Amazon Alexa |
| Licht (Schalter) | Licht in Amazon Alexa, welches an- und ausgeschaltet werden kann |
| Medienabspieler | Medienabspieler in Amazon Alexa |
| Rollladen | Rollläden in Amazon Alexa |
| Schloss | Schlösser in Amazon Alexa |
| Szenen | Szenen in Amazon Alexa |
| Szenen (deaktivierbar) | Szenen in Amazon Alexa |
| Temperatursensor | Temperatursensoren in Amazon Alexa |
| Thermostat | Thermostate in Amazon Alexa |
| Google Assistant | Modul zur Verknüpfung mit Google Assistant |
| Expertenoptionen | Expertenoptionen von Google Assistant |
| Generischer Schalter | Generische Schalter in Google Assistant |
| Licht (Dimmer) | Licht in Google Assistant, welches gedimmt werden kann |
| Licht (Experte) | Expertenlicht in Google Assistant |
| Licht (Farbe) | Farbiges Licht in Google Assistant |
| Licht (Schalter) | Licht in Google Assistant, welches an- und ausgeschaltet werden kann |
| Rollladen | Rollläden in Google Assistant |
| Szenen | Szenen in Google Assistant |
| Szenen (deaktivierbar) | Szenen in Google Assistant |
| Thermostat | Thermostate in Google Assistant |

## BENACHRICHTIGUNGEN

Module, damit sich Symcon beim Nutzer melden kann

| | |
|---|---|
| Alarmierung | Löst einen Alarm aus, wenn eine der Sensorenvariablen aktiv wird. |
| ARM_GetLastAlertID | Gibt die ID der Variable zurück, die als letztes einen Alarm ausgelöst hat. |
| ARM_SetActive | Schaltet das Alarmierungsmodul mit der InstanzID __InstanzID__ auf den Wert __Value__ (true = An; false = Aus). |
| ARM_SetAlert | Schaltet den Alarm mit der InstanzID __InstanzID__ auf den Wert __Value__ (true = An; false = Aus). |
| Benachrichtigung | Ermöglicht eine mehrstufige Benachrichtigung, wobei die Stufe sich nach einer definierten Zeit erhöht und bei einer Quittierung zurückgesetzt wird. |
| BN_IncreaseLevel | Erhöht die Stufe und führt die Aktion aus |
| BN_Reset | Setzt die Benachrichtigungsstufe zurück und deaktiviert die Kette |
| BN_SetNotifyLevel | Setzt die Benachichtigung auf eine vorgegebene Stufe |
| Durchsage | Das Durchsage Modul bietet die Möglichkeit von AWS Polly erzeugte Audiodaten über Sonos oder den Media Player wiederzugeben. |
| DS_Play | Spielt den Text als Durchsage ab. |
| Dynamische E-Mail | Versendet eine E-Mail mit einem dynamischen Text. |
| DM_SendMail | Nutzt den Betreff und Text um eine E-Mail zu versenden. |
| FertigMelder | Meldet nach einer Totzeit, nachdem ein Gerät einen gewissen Verbrauch unterschritten hat. |
| IMAP | Funktionen zum IMAP-Modul zur Abfrage von E-Mails |
| IMAP_DeleteMail | löscht eine Mail mit einer bestimmten E-Mail(UID) |
| IMAP_GetCachedMails | gibt ein Array zu den gecachten E-Mails zurück |
| IMAP_GetMailEx | liefert ein Array mit Informationen zu einer bestimmten E-Mail(UID) wieder |
| MediaPlayer | Funktionen zur Ansteuerung von Mediaplayer-Instanzen |
| WAC_AddFile | fügt Musikdatei zur Playlist hinzu |
| WAC_ClearPlaylist | löscht die Playlist |
| WAC_GetPlaylistLength | löscht die Playlist |
| WAC_GetPlaylistPosition | löscht die Playlist |
| WAC_Next | spielt den nächsten Titel der Playlist ab |
| WAC_Pause | pausiert die Wiedergabe |
| WAC_Play | spielt die Playlist ab |
| WAC_PlayFile | spielt eine Musikdatei direkt ab |
| WAC_Prev | spielt den vorherigen Titel der Playlist ab |
| WAC_SetPlaylistPosition | löscht die Playlist |
| WAC_SetPosition | setzt den Positionsmarker auf eine bestimmte Sekunde |
| WAC_SetRepeat | schaltet die Wiederholung der Playlist ein/aus |
| WAC_SetShuffle | schaltet die Zufallswiedergabe ein/aus |
| WAC_SetVolume | setzt die Lautstärke |

## KERN INSTANZEN

Module, welche im Kern von Symcon verankert sind

| | |
|---|---|
| Archive Control | Funktionen zur Verwaltung des Datenloggings |
| AC_AddLoggedValues | fügt einer geloggten Variable weitere Datensätze hinzu |
| AC_ChangeVariableID | migriert Daten einer Variable in eine bisher nicht geloggte Variable |
| AC_DeleteVariableData | löscht in einem bestimmten Zeitraum alle Datensätze einer geloggten Variable |
| AC_GetAggregatedValues | holt aggregierte Daten aus dem Archiv |
| AC_GetAggregationType | gibt den Aggregationstyp einer Variable wieder |
| AC_GetAggregationVariables | gibt ein Array aller geloggten Variablen zurück |
| AC_GetCounterIgnoreZeros | holt den Status ob Nullen und negative Werte ignoriert werden |
| AC_GetGraphStatus | fragt ab ob eine Variable visualisiert wird |
| AC_GetLoggedValues | holt Rohdaten aus dem Archiv Control |
| AC_GetLoggingStatus | fragt ab ob eine Variable geloggt wird |
| AC_ReAggregateVariable | startet die Reaggregation einer Variable |
| AC_SetAggregationType | setzt den Aggregationstyp einer Variable |
| AC_SetCompaction | Konfiguriert die Verdichtung der Variable |
| AC_SetCounterIgnoreZeros | setzt den Status ob Nullen und negative Werte ignoriert werden |
| AC_SetGraphStatus | setzt die Eigenschaft zur Visualisierung einer Variable |
| AC_SetLoggingStatus | setzt die Eigenschaft zum Loggen einer Variable |
| Datenformat | |
| Berechtigungssteuerung | Benutzerverwalung und Synchronisierung mit LDAP Server |
| Calendar Control | Kalender Modul |
| Connect Control | Verbindungsmodul um eine gesicherte Verbindung von außerhalb zu ermöglichen |
| Cutter | |
| Cutter_ClearBuffer | löscht den Buffer der Cutter Instanz |
| DNS-SD Control | verwaltet eingerichtete Bonjour Services |
| ZC_QueryService | fragt Informationen eines einzelnen Services ab |
| ZC_QueryServiceType | fragt Informationen von Services eines bestimmten Types ab |
| ZC_QueryServiceTypes | Fragt alle eingerichteten Services ab |
| Event Control | Modul um Skripte bei festgesetzten Ereignissen zu starten |
| Location Control | berechnet Dämmerungszeitpunkte anhand von Längen- und Breitengrad |
| Module Control | verwaltet externe und hinzugeladene PHP-Module |
| Notification Control | ein Modul zum verschicken von Push-Notifications |
| Presence Control | Funktionen zur Detektion von Anwesenheit in Bereichen |
| PC_Enter | setzt die Anwesenheit auf true |
| PC_Leave | setzt die Anwesenheit auf false |

## I/O INSTANZEN

Funktionen zur Verwaltung der I/O Instanzen

| | |
|---|---|
| Client Socket | TCP Schnittstelle |
| CSCK_SendText | sendet einen String auf den I/O |
| HID | HID Schnittstelle |
| HID_SendEvent | sendet einen String auf den I/O |
| HTTP Client | Schnittstelle zum auslesen von Webseiten |
| WWW_UpdatePage | lässt die konfigurierte URL abfragen |
| Multicast Socket | Multicast Schnittstelle |
| MSCK_SendText | sendet einen String auf den I/O |
| Serial Port | Serielle Schnittstelle |
| SPRT_SendText | sendet einen String auf den I/O |
| SPRT_SetBreak | schaltet den Break einer Serial Port Instanz |
| SPRT_SetDTR | schaltet den DTR einer Serial Port Instanz |
| SPRT_SetRTS | schaltet den RTS einer Serial Port Instanz |
| Server Sent Event Client | TCP Schnittstelle |
| Server Socket | öffnet einen Port fürs Netzwerk |
| SSCK_SendPacket | sendet einen String auf den I/O |
| SSCK_SendText | sendet einen String auf den I/O |
| UDP Socket | UDP Schnittstelle |
| USCK_SendPacket | sendet einen String auf den I/O |
| USCK_SendText | sendet einen String auf den I/O |
| Virtual I/O | Virtuelle I/O Schnittstelle |
| VIO_Connect | sendet ein Connect Datenpaket |
| VIO_Disconnect | sendet ein Disconnect Datenpaket |
| VIO_GetPacketList | liefert ein Array von erweiterten Datenpaketen |
| VIO_GetTextList | liefert ein Array von simplen Datenpaketen |
| VIO_PushPacket | sendet ein erweitertes Datenpaket |
| VIO_PushPacketHEX | sendet ein erweitertes Datenpaket |
| VIO_PushText | sendet ein simples Datenpaket |
| VIO_PushTextHEX | sendet ein simples Datenpaket |
| VIO_SendPacket | sendet ein erweitertes Datenpaket an sich selbst |
| VIO_SendText | sendet ein simples Datenpaket an sich selbst |
| WebSocket Client | TCP Schnittstelle |
| WSC_SendMessage | sendet eine Nachricht über den WebSocket |

## DATENSICHERUNG

Sicherungen durch Backup Module

| | |
|---|---|
| Backup | Erstellt ein Backup über SFTP, FTP oder FTPS |
| SB_CreateBackup | Erstellt oder updatet ein Backup |

# LEGACY

## Nicht mehr unterstützte Module

| | |
|---|---|
| RRDTool | Direkteinbindung des RRDTool |
| RRD_Execute | gibt einen Befehl an die RRDTool Library weiter |
| Shutter Control (legacy) | Funktionen zur Ansteuerung von Shutter(Rollladen) Komponenten (legacy) |
| SC_Move | bewegt den Rollladen an eine bestimmte Position |
| SC_MoveDown | bewegt den Rollladen bis zu Endposition runter |
| SC_MoveUp | bewegt den Rollladen bis zu Endposition hoch |
| SC_Stop | stoppt einen Bewegungsvorgang |
| USBMapper | Setzt automatisch den korrekten USB-Port zu den eingetragenen Serial-Ports. |
| USBM_FixPorts | Kontroliert ob die USB-Ports noch richtig konfiguriert sind |
| Web Graph | Beliebige Graphen können über den Web Hook eigenständig angezeigt werden |
| Wunderground Weather | Holt Wetterdaten (Aktuell/stündlich/12-stündlich/Unwetterwarnung) via API von www.wunderground.com |
| WGW_UpdateStormWarningData | Aktualisiert die Unwetterwarnungen des Weathergroundmoduls |
| WGW_UpdateWeatherData | Aktualisiert die Wetterdaten (aktuell/stündlich/ täglich) des Weathergroundmoduls |

# Geräte

Geräte sind Hardwarekomponenten, welche bestimmte Systeme, Aktoren oder Sensorik einbinden.

## 1-WIRE

Funktionen zur Ansteuerung von 1-Wire Komponenten

| | |
|---|---|
| DS2405 | 1-fach Switch |
| OW_SwitchMode | setzt den Pin vom DS2405 auf an/aus |
| OW_ToggleMode | toggelt den Pin vom DS2405 |
| DS2406 | 2-fach Switch |
| OW_SetPin | setzt einen Pin vom DS2406 auf an/aus |
| DS2408 | 8-fach Switch |
| OW_SetPin | setzt einen Pin vom DS2408 auf an/aus |
| OW_SetPort | setzt den kompletten Port vom DS2408 auf Wert |
| OW_SetStrobe | schaltet den Strobe vom DS2408 an/aus |
| OW_WriteBytes | sendet mehrere Bytes an den DS2408 |
| OW_WriteBytesMasked | sendet mehrere Bytes an den DS2408 unter Berücksichtigung einer Bitmaske |
| DS2413 | 2-fach Switch |
| OW_SetPin | setzt einen Pin vom DS2413 auf an/aus |
| DS2438 | Smart Battery Monitor |
| DS2450 | A/D-Wandler 16bit |
| OW_SetPin | setzt einen Pin vom DS2450 auf an/aus |
| DS2890 | Digital-Potentiometer 8 bit |
| OW_SetPosition | setzt die Position vom DS2490 auf an/aus |
| Geräteliste | Geräteliste 1-Wire |
| OW_RequestStatus | sende eine Leseanfragen an das OneWire Gerät |

## ABL

Einrichtung von ABL Geräten

| | |
|---|---|
| Geräteliste | |

## ALFEN

Einrichtung von Alfen Geräten

| | |
|---|---|
| Geräteliste | |

## ALLNET

Funktionen zur Ansteuerung von ALLNET Komponenten

| | |
|---|---|
| ALL_SetAnalog | setzt den Analogwert eines Kanals |
| ALL_SwitchActor | schaltet ein ALLNET Aktor ein/aus |
| ALL_SwitchMode | schaltet ein ALLNET Gerät ein/aus |
| ALL_UpdateValues | sende eine Aktualisierungsanfrage an das ALLnet Gerät |
| Geräteliste | Geräteliste ALLNET |

## BACNET

Funktionen zur Ansteuerung von BACnet Komponenten

| | |
|---|---|
| BAC_RelinquishPresetValue | Rücksetzung des PresetValues |

## DIGITALSTROM

Funktionen zur Ansteuerung von digitalSTROM Komponenten

| | |
|---|---|
| DS_CallScene | ruft eine Szene auf |
| DS_DimSet | dimmt eine digitalSTROM Klemme auf einen bestimmten Wert |
| DS_MakeRequest | sendet einen digitalSTROM Befehl direkt an den dSS |
| DS_RequestStatus | ruft den Status ab |
| DS_ShutterMove | bewegt den Rollladen bis zu einer gewünschten Position/Stop |
| DS_ShutterMoveDown | bewegt den Rollladen nach unten bis zur Endposition/Stop |
| DS_ShutterMoveUp | bewegt den Rollladen nach oben bis zur Endposition/Stop |
| DS_ShutterStop | stoppt einen Bewegungsvorgang |
| DS_SwitchMode | schaltet eine digitalSTROM Klemme an/aus |
| Geräteliste | Geräteliste digitalSTROM |

## DMX / ARTNET

Funktionen zur Ansteuerung von DMX Komponenten

| | |
|---|---|
| DMX_FadeChannel | dimmt einen Kanal auf einen bestimmten Wert in einer bestimmten Zeit |
| DMX_FadeChannelDelayed | dimmt einen Kanal nach einer Verzögerung auf einen bestimmten Wert |
| DMX_FadeRGB | dimmt den RGB-Kanal mit einer Überblendzeit |
| DMX_FadeRGBDelayed | dimmt verzögert den RGB-Kanal mit einer Überblendzeit |
| DMX_RequestInfo | Fragt die Informationen einer DMX Instanz ab |
| DMX_ResetInterface | setzt ein DMX Interface zurück |
| DMX_SetBlackOut | setzt alle 512 Kanäle auf 0 oder den alten Wert |
| DMX_SetChannel | setzt einen Kanal auf einen bestimmten Wert |
| DMX_SetRGB | setzt den RGB Kanal auf einen bestimmten Wert |
| Geräteliste | Geräteliste DMX |

## EGIGEOZONE

Empfängt Daten von der EgiGeoZone App für Google Android.

## EKEY

Funktionen zur Ansteuerung von ekey Komponenten

| | |
|---|---|
| Geräteliste | Geräteliste eKey |

## EKEY BIONYX

## ENOCEAN

Funktionen zur Ansteuerung von EnOcean Komponenten

| | |
|---|---|
| ENO_DimSet | dimmt ein EnOcean Gerät auf eine bestimmte Helligkeit |
| ENO_SetActiveMessage | setzt eine bestimmte Message aktiv |
| ENO_SetFanStage | setzt die FanStage auf einen bestimmten Wert |
| ENO_SetIntensity | dimmt den Opus-Dimmer auf eine bestimmte Helligkeit |
| ENO_SetLockFanStage | locked die FanStage |
| ENO_SetLockRoomOccupancy | locked die Occupancy |
| ENO_SetMode | setzt den Stellantrieb in einen bestimmten Modus |
| ENO_SetPosition | setzt die Position des Stellantriebes |
| ENO_SetRoomOccupancy | setzt die Occupancy auf True/False |
| ENO_SetTemperature | setzt den Stellantrieb auf eine bestimmte Soll-Temperatur |
| ENO_SetTemperature1 | setzt das Gerät auf eine bestimmte Temperatur |
| ENO_ShutterMoveDown | bewegt den Rollladen nach unten bis zur Endposition/Stop |
| ENO_ShutterMoveDownEx | bewegt den Rollladen für eine bestimmte Zeit |
| ENO_ShutterMoveUp | bewegt den Rollladen nach oben bis zur Endposition/Stop |
| ENO_ShutterMoveUpEx | bewegt den Rollladen für eine bestimmte Zeit |
| ENO_ShutterStop | stoppt einen Bewegungsvorgang |
| ENO_SwitchMode | schaltet ein EnOcean Gerät ein/aus |
| ENO_SwitchModeEx | sendet einen erweiterten Schaltbefehl |
| Geräteliste | Geräteliste EnOcean/EEP |

## FHZ1X00PC

Funktionen zur Ansteuerung von FS20, HMS, FHT Komponenten

| | |
|---|---|
| FHT | Funk-Heizungssystem |
| FHT_SetMode | sendet einen neuen Soll-Modus an den FHT |
| FHT_SetTemperature | sendet eine neue Soll-Temperatur an den FHT |
| FS20 | FunkSystem für digitale Geräte |
| FS20_DimDown | dimmt ein FS20 Gerät eine Stufe runter |
| FS20_DimUp | dimmt ein FS20 Gerät ein Stufe höher |
| FS20_SetIntensity | dimmt ein FS20 Gerät in einer bestimmten Zeitspanne |
| FS20_SwitchDuration | schaltet ein FS20 Gerät für eine bestimmte Zeit ein/aus |
| FS20_SwitchMode | schaltet ein FS20 Gerät ein/aus |
| Geräteliste | Geräteliste FHZ |
| HMS | HausMeldeSystem für analoge Geräte |
| HMS_ReleaseFI | löst den FI Schalter aus |
| KS300 | Kombisensor Wetterstation |

## FS10 WETTER

Wetterstation mit Funk auf 433MHz

## GARDENA SMART SYSTEM

Modul zum Hinzufügen von GARDENA smart system Geräten

| | |
|---|---|
| smart Irrigation Control | Eine GARDENA Bewässerungssteuerung mit 6 Ventilen |
| smart SILENO Mower | Ein smart SILENO Mower von GARDENA |
| smart Power Socket | Ein Zwischenstecker von GARDENA |
| smart Pressure Pump | Eine GARDENA smart Pump mit einem Ventil |
| smart Sensor | Ein GARDENA smart Sensor mit verschiedenen Messwerten |
| smart Water Control | Eine GARDENA Bewässerungssteuerung mit einem Ventil |

## GEOFENCY

Empfängt Daten von der Geofency App für Apple iOS.

## HEATING CONTROL

Funktionen zur Ansteuerung vom Heizungsmodul zur Regelung von Raumtemperaturen

| | |
|---|---|
| HC_TargetValue | sendet einen Soll Wert an das Heating-Modul |

## HOME CONNECT

Modul zum Hinzufügen von Home Connect Geräten

| | |
|---|---|
| Geräteliste | Geräteliste Home Connect |
| Home Connect Gerät | Die durch den Home Connect Konfigurator erstellte Geräte Instanz |

## HOMEMATIC

| | |
|---|---|
| Geräteliste | Geräteliste HomeMatic |
| HomeMatic CCU | Funktionen zur Ansteuerung von HomeMatic Komponenten |

| | |
|---|---|
| BidCos Service | Einrichtung BidCos Service |
| HM_ReadServiceMessages | liest die Service-Meldungen aus |
| HM_RequestStatus | fragt den Ist-Wert vom Gerät ab |
| HM_WriteValueBoolean | sendet einen neuen Soll-Wert vom Typ Boolean an die CCU |
| HM_WriteValueFloat | sendet einen neuen Soll-Wert vom Typ Float an die CCU |
| HM_WriteValueInteger | sendet einen neuen Soll-Wert vom Typ Integer an die CCU |
| HM_WriteValueString | sendet einen neuen Soll-Wert vom Typ String an die CCU |

HomeMatic HCU (Beta)

# IMAGE GRABBER

Modul zum Abfragen von Webcam Bildern

- IG_UpdateImage — aktualisiert das Bild der Instanz

# IPS-868

Funktionen zur Ansteuerung von IPS868 Komponenten

- AKM-868 — Anwesenheitskontrollmodul mitsamt Tracker
- ANA-868 — Analog-Digital Wandler
  - PJ_RequestStatus — Frage den Status von Variablen ab und setzt diese
  - PJ_SetVoltage — Setzt den Ausgang auf eine bestimmte Spannung
- EKM-868 — Ein Energiekontrollmodul mit 4-Kanal-Zählermodul mit je 32 Bit
  - PJ_RequestStatus — Frage den Status von Variablen ab und setzt diese
- FD-868 — Ein Funkdisplay mit einem 2*16-stelligen LC-Display
  - PJ_Backlight — schaltet die Displaybeleuchtung An oder Aus
  - PJ_Beep — gibt ein Tonsignal aus
  - PJ_LCDText — sendet einen Text an das Display
  - PJ_SwitchLED — schaltet einen LED vom Display
  - PJ_SwitchMode — schaltet ein FS20 Gerät ein/aus
- Geräteliste — Geräteliste IPS 868
- JKM-868 LevelJET — Ein Ausleser für LevelJET Füllstandsanzeige
  - PJ_RequestStatus — Frage den Status von Variablen ab und setzt diese
- JKM-868 ThermoJET — Ein Ausleser für Einzelwerte des ThermoJET Temperaturreglers
  - PJ_RequestStatus — Frage den Status von Variablen ab und setzt diese
- LGS-868 — Luftgütesensor zur Analyse der Raumluft
  - PJ_RequestStatus — Frage den Status von Variablen ab und setzt diese
  - PJ_SetLEDs — schaltet LEDs am LuftGüteSensor
- RGBW-868 — Steuergerät für LED-RGBW-Stripes
  - PJ_DimRGBW — dimmt einen LED Stripe
  - PJ_RunProgram — startet ein Lichtprogramm
  - PJ_SetRGBW — setzt einen Farbwert
- SERVO-868 — Ein 4-fach Servosteuermodul
  - PJ_DimServo — setzt Schrittweise einen Servo-Wert
  - PJ_SetServo — setzt einen Servo Kanal
- WDT-868 — Ein Watch-Dog-Timer zur Überwachung des Systems
  - PJ_SwitchDuration — schaltet ein Relais
  - PJ_SwitchMode — sendet einen Schaltbefehl

## IR TRANS

Funktionen zur Ansteuerung von Infrarot Komponenten

| | |
|---|---|
| Geräteliste | Geräteliste IR-Trans |
| IRT_SendOnce | sendet einen IR Befehl |

## KEBA

Geräteliste

## KNX

Funktionen zur Ansteuerung von KNX Komponenten

| | |
|---|---|
| EIB_Char | Sendet einen einzelnen Buchstaben |
| EIB_Counter8Bit | sendet einen 8bit Zählerwert |
| EIB_Counter16Bit | sendet einen 16bit Zählerwert |
| EIB_Counter32Bit | sendet einen 32bit Zählerwert |
| EIB_Date | sendet den Datumswert |
| EIB_DimControl | sendet eine Dimmwertverstellung |
| EIB_DimValue | sendet einen Dimmwert als Zahl oder Prozent |
| EIB_DriveBladeValue | sendet einen Positionswert als Zahl oder Prozent |
| EIB_DriveMove | bewegt ein Gerät Hoch oder Runter |
| EIB_DriveShutterValue | sendet einen Positionswert als Zahl oder Prozent |
| EIB_DriveStep | bewegt ein Gerät schrittweise |
| EIB_FloatValue | sendet einen Floatwert |
| EIB_Move | bewegt einen Rollladen |
| EIB_Position | bewegt einen Rollladen auf eine Position |
| EIB_PriorityControl | setzt einen Gerätezustand |
| EIB_PriorityPosition | setzt ein Gerät in eine bestimmte Richtung |
| EIB_RequestStatus | schickt eine Leseanfrage für EIB-Instanzen auf den Bus |
| EIB_Scale | sendet einen Zahl-, Prozent- oder Gradwert. |
| EIB_SetRGB | setzt einen RGB Stripe auf einen bestimmten Farbton |
| EIB_SetRGBW | setzt einen RGBW Stripe auf einen bestimmten Farbton |
| EIB_Str | sendet einen Text |
| EIB_Switch | schaltet ein Gerät Ein/Aus |
| EIB_Time | sendet einen Zeitwert |
| EIB_Value | sendet einen Floatwert |
| Geräteliste | Geräteliste EIB/KNX |
| KNX_RequestStatus | schickt eine Leseanfrage für DPT-Instanzen auf den Bus |

# LCN

Funktionen zur Ansteuerung von LCN Komponenten

| | |
|---|---|
| Geräteliste | Geräteliste LCN |
| LCN_AddGroup | fügt ein Gerät zu einer Gruppe hinzu |
| LCN_AddIntensity | erhöht die Leuchtintensität ein Gerät |
| LCN_AddThresholdCurrent | addiert Schwellwert zu aktuellen Wert |
| LCN_AddThresholdDefined | addiert Schwellwert zu vordefinierten Wert |
| LCN_Beep | lässt ein Gerät piepen |
| LCN_DeductIntensity | veringert die Leuchtintensität ein Gerät |
| LCN_DeductThresholdCurrent | subtrahiert Schwellwert zu aktuellen Wert |
| LCN_DeductThresholdDefined | subtrahiert Schwellwert zu vordefinierten Wert |
| LCN_FadeOut | setzt ein Gerät auf eine feste Intensität und dimmt dann auf 0% |
| LCN_FlipRelay | schaltet ein Relais um |
| LCN_LimitOutput | begrenzt einen Ausgang für einen bestimmten Zeitraum |
| LCN_LoadScene | ruft eine Szene für einen Ausgang auf |
| LCN_LockTargetValue | sperrt einen Regler des Geräts |
| LCN_RampStop | stoppt einen Dimmvorgang |
| LCN_ReleaseTargetValue | entsperrt einen Regler des Geräts |
| LCN_RemoveGroup | entfernt ein Gerät aus einer Gruppe |
| LCN_RequestLights | fragt die LEDs des Tableaus ab |
| LCN_RequestRead | fragt die Werte ab |
| LCN_RequestStatus | fragt alle Status des Moduls ab |
| LCN_RequestThresholds | fragt Schwellenwerte und Hysterese ab |
| LCN_SaveScene | speichert eine Szene für einen Ausgang |
| LCN_SelectSceneRegister | wählt ein Register aus |
| LCN_SendCommand | sendet einen benutzerdefinierten Befehl |
| LCN_SetDisplayText | Zeigt einen Text auf einer Zeile des Displays an |
| LCN_SetDisplayTime | setzt die Anzeigedauer einer Zeile im Display |
| LCN_SetIntensity | dimmt ein Gerät in einer bestimmten Zeit |
| LCN_SetLamp | schalten einer LED |
| LCN_SetRelay | setzt alle Relais in einem Modul (8Bit) |
| LCN_SetRGBW | setzt eine Instanz auf einen RGBW Farbwert |
| LCN_SetTargetValue | setzt einen Regler auf einen bestimmten Wert |
| LCN_ShiftTargetValue | Shiftet den aktuellen Sollwert |
| LCN_ShutterMove | startet eine Rollladenbewegung auf prozentuale Position |
| LCN_ShutterMoveDown | startet eine Rollladenbewegung nach unten |
| LCN_ShutterMoveUp | startet eine Rollladenbewegung nach oben |
| LCN_ShutterStop | stoppt eine Rollladenbewegung |
| LCN_StartFlicker | ein Gerät flackern lassen |
| LCN_StopFlicker | stoppt das Flackern eines Geräts |
| LCN_SwitchDurationMin | nach wievielen Minuten abgedimmt werden soll |
| LCN_SwitchDurationSec | nach wie vielen Sekunden abgedimmt werden soll |

## LJQUICK

Das KNX quick (Lingg&Janke) Modul unterstütz bei der Einrichtung von KNX Instanzen.

| | |
|---|---|
| LJ_SendDateTime | Sendet das aktuelle Datum/Uhrzeit auf den KNX-Bus |

## M-BUS

Funktionen zur Ansteuerung von M-Bus Komponenten

| | |
|---|---|
| Geräteliste | Geräteliste M-Bus |
| MBUS_UpdateValues | aktualisiert alle gerätespezifischen Werte |

## MENNEKES

Einrichtung von Mennekes Geräten

| | |
|---|---|
| Geräteliste | |

## MODBUS RTU/TCP

Funktionen zur Ansteuerung von ModBus RTU/TCP Komponenten

| | |
|---|---|
| Geräteliste | Geräteliste ModBus RTU/TCP |
| ModBus_RequestRead | führt einen Lesevorgang auf einem Gerät aus |
| ModBus_WriteCoil | setzt eine Adresse auf An/Aus |
| ModBus_WriteRegister | schreibt einen Wert in die Schreiben-Adresse |
| ModBus_WriteRegisterByte | setzt eine Adresse auf einen bestimmten Byte-Wert |
| ModBus_WriteRegisterChar | setzt eine Adresse auf einen bestimmten Char-Wert |
| ModBus_WriteRegisterDWord | setzt eine Adresse auf einen bestimmten DWord-Wert |
| ModBus_WriteRegisterInt64 | setzt eine Adresse auf einen bestimmten Int64-Wert |
| ModBus_WriteRegisterInteger | setzt eine Adresse auf einen bestimmten Integer-Wert |
| ModBus_WriteRegisterReal | setzt eine Adresse auf einen bestimmten Float-Wert |
| ModBus_WriteRegisterReal64 | setzt eine Adresse auf einen bestimmten Float-Wert |
| ModBus_WriteRegisterShort | setzt eine Adresse auf einen bestimmten Short-Wert |
| ModBus_WriteRegisterString | setzt eine Adresse auf einen bestimmten String |
| ModBus_WriteRegisterWord | setzt eine Adresse auf einen bestimmten Word-Wert |

## MÖHLENHOFF ALPHA 2

Empfangen und Schalten von Möhlenhoff Alpha2 Daten

| | |
|---|---|
| MA2_RequestStatus | Ruft die in der Alpha2 mit der InstanzID gespeicherten Werte auf und setzt die dazugehörigen Variablen. |
| MA2_WriteValue | Schreibt den Wert in die Variable mit dem Ident in der Möhlenhoff Alpha 2 mit der InstanzID. |

## MQTT

Funktionen zur Ansteuerung von MQTT Komponenten

| | |
|---|---|
| Geräteliste | Geräteliste MQTT |
| MQTT Client | Funktionen zur Ansteuerung von MQTT Client |
| MQTT-Server | Funktionen zur Ansteuerung von MQTT Server |

## NEA SMART

Empfangen und Schalten von NEA Smart Daten

| | |
|---|---|
| NEAS_RequestStatus | Ruft die in der NEA Smart mit der InstanzID gespeicherten Werte auf und setzt die dazugehörigen Variablen. |
| NEAS_WriteValue | Schreibt den Wert in die Variable mit dem Ident in der NEA Smart mit der InstanzID. |

## OCPP

Über das OCPP Protokoll können Wallboxen bequem über Symcon überwacht und geschaltet werden.

| | |
|---|---|
| OCPP_RemoteStartTransaction | Wenn der Status auf 'Preparing' oder 'Finishing' steht, kann durch diesen Befehl das Laden freigeschaltet werden. |
| OCPP_RemoteStopCurrentTransaction | Wenn der Zustand auf 'Charging' steht, kann durch diesen Befehl das Laden beendet werden. |
| OCPP_RemoteStopTransaction | Wenn der Zustand auf 'Charging' steht, kann durch diesen Befehl das Laden beendet werden. |

## OPC UA

Funktionen zur Ansteuerung von OPC UA Komponenten

## SAGEGLASS (BACNET)

Integriert die SageGlass SIM II (BACnet) in IP-Symcon

## SHUTTER CONTROL

Funktionen zur Ansteuerung von Shutter(Rollladen) Komponenten

| | |
|---|---|
| SC_Move | bewegt den Rollladen an eine bestimmte Position |
| SC_MoveDown | bewegt den Rollladen bis zu Endposition runter |
| SC_MoveUp | bewegt den Rollladen bis zu Endposition hoch |
| SC_Stop | stoppt einen Bewegungsvorgang |

## SIEMENS OZW

Funktionen zur Ansteuerung von Siemens OZW Komponenten

| | |
|---|---|
| Geräteliste | Geräteliste Siemens OZW |
| OZW_RequestStatus | fragt den Status eines Geräts ab |
| OZW_WriteDataPoint | schreibt einen bestimmten Wert in einen Datenpunkt |

## SNMP

Zeige Werte für beliebige OIDs eines Walks an. Optional können diese als Variable erstellt und beschreiben werden.

| | |
|---|---|
| SNMP_UpdateValues | Updated die Variablenwerte, welche unterhalb der Instanz liegen. |

## SNOM

Integration von Snom IP Telefone

## SPS: SIEMENS, VIPA, LOGO

Funktionen zur Ansteuerung von Siemens/Vipa/Logo SPS Komponenten

| | |
|---|---|
| Geräteliste | Geräteliste SPS Siemens/Vipa |
| Logo-VM-Adressen | VM Adressen der Logo7/8 |
| S7_RequestRead | führt einen Lesevorgang auf einem Gerät aus |
| S7_Write | schreibt einen Wert in die konfigurierte Adresse |
| S7_WriteBit | setzt eine Adresse auf An/Aus |
| S7_WriteByte | setzt eine Adresse auf einen bestimmten Bytewert |
| S7_WriteChar | setzt eine Adresse auf einen bestimmten Char |
| S7_WriteDWord | setzt eine Adresse auf einen bestimmten unsigned Integerwert |
| S7_WriteInteger | setzt eine Adresse auf einen bestimmten Integerwert |
| S7_WriteReal | setzt eine Adresse auf einen bestimmten Floatwert |
| S7_WriteShort | setzt eine Adresse auf einen bestimmten Shortwert |
| S7_WriteWord | setzt eine Adresse auf einen bestimmten unsigned Smallintegerwert |

## SPS: WAGO, BECKHOFF, ABB

Funktionen zur Ansteuerung von Wago/Beckhoff SPS Komponenten

## SYNC REMOTE

Synchronisation zwischen zwei Symcon Systemen

## TECHNISCHE ALTERNATIVE

Funktionen zur Ansteuerung von Technische Alternative Komponenten

| | |
|---|---|
| Geräteliste | Geräteliste Technische Alternative |

## VOICE OVER IP

Funktionen zur Ansteuerung VoIP

| | |
|---|---|
| Geräteliste | Geräteliste Voice over IP |
| VoIP_AcceptCall | akzeptiert eine Verbindung |
| VoIP_Connect | baut eine Verbindung auf |
| VoIP_Disconnect | beendet eine Verbindung |
| VoIP_GetConnection | gibt Informationen zu einer Verbindung wieder |
| VoIP_GetData | gibt Daten einer Verbindung zurück |
| VoIP_PlayWave | spielt eine Wavedatei ab |
| VoIP_RejectCall | lehnt eine Verbindung ab |
| VoIP_SendDTMF | sendet eine Abfolge von Zeichen als DTMF |
| VoIP_SetData | setzt die Daten einer Verbindung |

## WINLIRC

Funktionen zum Ansteuern von WinLIRC Komponenten

| | |
|---|---|
| WinLIRC_SendOnce | sendet einmalig einen Fernbedienungsbefehl |

## WIRELESS M-BUS

Funktionen zur Ansteuerung von Wireless M-Bus Komponenten

| | |
|---|---|
| Geräteliste | Geräteliste Wireless M-Bus |

## WMRS200

Funktionen zum Ansteuern von WMRS200 Komponenten

| | |
|---|---|
| Geräteliste | Geräteliste WMRS200 |

## W&T

Funktionen zum Ansteuern von W&T Komponenten

| | |
|---|---|
| WUT_SwitchMode | schaltet einen Kanal An/Aus |

## XBEE

Funktionen zur Ansteuerung von XBee

| | |
|---|---|
| XBee_SendBuffer | sendet einen Datenstring an ein bestimmtes Gerät |
| XBee_SendCommand | sendet einen Befehl an einen bestimmten XBee-Splitter |

## XCOMFORT

Funktionen zur Ansteuerung von xComfort Komponenten

| | |
|---|---|
| Geräteliste | Geräteliste Eaton xComfort |
| MXC_DimBrighter | fängt einen Dimmvorgang auf eine hellere Stufe an |
| MXC_DimDarker | fängt einen Dimmvorgang auf eine dunklere Stufe an |
| MXC_DimSet | dimmt ein xComfort Gerät auf eine bestimmte Stufe |
| MXC_DimStop | hält einen Dimmvorgang an |
| MXC_RequestStatus | sende eine Statusmeldungsanfrage an ein Gerät |
| MXC_SendBoolean | sendet einen Boolean-Wert an den Room-Mananger |
| MXC_SendFloat | sendet einen Float-Wert an den Room-Mananger |
| MXC_SendInteger | sendet einen Integer-Wert an den Room-Mananger |
| MXC_SetTemperature | sendet einen Sollwert an den Stellantrieb |
| MXC_ShutterMoveDown | bewegt den Rollladen nach unten bis zur Endposition/Stop |
| MXC_ShutterMoveUp | bewegt den Rollladen nach oben bis zur Endposition/Stop |
| MXC_ShutterStepDown | bewegt den Rollladen ein Stück nach unten |
| MXC_ShutterStepUp | bewegt den Rollladen ein Stück nach oben |
| MXC_ShutterStop | stoppt einen Bewegungsvorgang |
| MXC_SwitchMode | schaltet ein xComfort Gerät an/aus |

## Z-WAVE

Funktionen zur Ansteuerung von Z-Wave Komponenten

| | |
|---|---|
| Geräteliste | Geräteliste Z-Wave |
| ZW_Basic | bient ein Gerät über die Basisfunktion |
| ZW_ColorCW | setzt den KaltWeiß Kanal eines RGBWW Z-Wave Geräts |
| ZW_ColorRGB | setzt die Farbe eines Z-Wave Geräts |
| ZW_ColorRGBWW | setzt die Farben und Intensität eines RGBWW Z-Wave Geräts |
| ZW_ColorWW | setzt den WarmWeiß Kanal eines Z-Wave Geräts |
| ZW_DimDown | beginnt ein Runterdimmen |
| ZW_DimDownEx | beginnt mit einer Laufzeit Runterzudimmen |
| ZW_DimSet | dimmt ein Z-Wave Gerät auf eine bestimmte Stufe |
| ZW_DimSetEx | dimmt ein Z-Wave Gerät in einer Laufzeit auf eine bestimmte Stufe |
| ZW_DimStop | stoppt die Dimmung eines Z-Wave Geräts |
| ZW_DimUp | beginnt ein Hochdimmen |
| ZW_DimUpEx | beginnt mit einer Laufzeit Hochzudimmen |
| ZW_DoorLockOperation | Setzt den Door Lock Operation Modus eines Z-Wave Geräts |
| ZW_LockMode | setzt den LockMode |
| ZW_MeterReset | resetet das Meter |
| ZW_Optimize | Startet den Optimierugnsprozess des Funknetzwerks für ein Z-Wave Gerät |
| ZW_ProtectionSet | setzt die Protection und dessen Mode |
| ZW_RequestStatus | sende eine Statusmeldungsanfrage an ein Gerät |
| ZW_ShutterMoveDown | bewegt den Rollladen nach unten bis zur Endposition/Stop |
| ZW_ShutterMoveUp | bewegt den Rollladen nach oben bis zur Endposition/Stop |
| ZW_ShutterStop | stoppt einen Bewegungsvorgang |
| ZW_SwitchAllMode | Setzt den Switch All Modus eines Z-Wave Geräts |
| ZW_SwitchMode | schaltet ein Z-Wave Gerät an/aus |
| ZW_Test | Testet ob eine Verbindung zu einem Z-Wave Gerät funktioniert |
| ZW_ThermostatFanModeSet | Setzt den FanMode für Thermostat |
| ZW_ThermostatModeSet | Setzt den Thermostat Mode |
| ZW_ThermostatSetPointSet | Setzt den PointSet Value |

## ZEVVY

### Sendet Daten an ein Zevvy Konto

| | |
|---|---|
| ZY_SendMeasurements | Sendet die Daten der eingerichtet Variablen an ein Zevvy Konto |

# 1-Wire

Das 1-Wire Bussystem ähnelt dem I2C. Es lassen sich auf einfache Weise verschiedene Sensoren an ein normales 3-adriges Kabel (Masse, Daten, und +5V) anschließen. Die Verbindung mit IP-Symcon erfolgt durch ein USB-, Seriell-, LAN Gateway oder die SymBox integrierte Schnittstelle .



Folgende Geräte werden von IP-Symcon unterstützt:
Unterstützte Geräte



Ab Version 4.0 wird, außer für DS2490/DS9490R, **kein** TMEX Treiber mehr für den Betrieb benötigt.

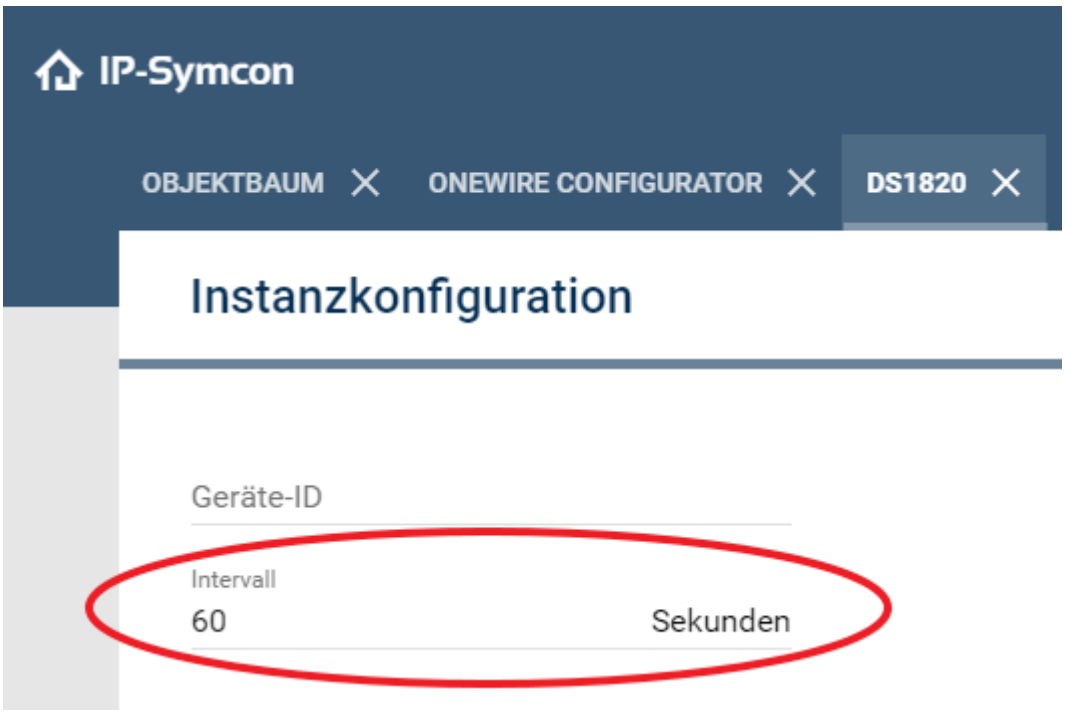


IP-Symcon unterstützt nicht den **parasitären** Modus.
Eine **+5V** Spannungsversorgung ist zwingend notwendig!

## VIDEO-TUTORIAL ZUR EINRICHTUNG

## EINBINDUNG IN IP-SYMCON

Bei Verwendung des LAN-Gateways kann dies über die Gerätesuche eingebunden werden. Hierfür muss als System "OneWire Discovery" gewählt werden. Die Discovery-Instanz bietet dann die Erstellung eines OneWire Konfigurators an. Nachdem der Konfigurator erstellt wurde, können über diesen die einzelnen Geräte mit "Erstellen" in einer Instanz erstellt werden.
Die am 1-Wire-Bus angeschlossenen Bausteine versenden nicht automatisch ihre Werte – diese müssen angefragt werden. Wie oft dies geschehen soll, kann mit dem **Intervall** in der jeweiligen Instanzkonfiguration angegeben werden.




Jede Abfrage kostet Zeit und belegt den Bus. Darum sollte der "Intervall" sparsam genutzt werden. In der Praxis ist z.B. ein Abfragezyklus von 60 Sekunden, um eine Zimmertemperatur auszulesen, mehr als ausreichend. Kleinere Werte unter einer Sekunde können aus den o.g. Gründen nicht eingestellt werden.

Desweiteren sollte beachtet werden, dass je mehr Geräte an einem Adapter angeschlossen werden, die Busauslastung steigt. In unserem Musterhaus laufen 30 Geräte an einem Adapter ohne Probleme. Wir empfehlen bei größeren Installationen eine eigene Spannungsversorgung sowie mehrere Adapter einzusetzen.

## TIPPS & TRICKS

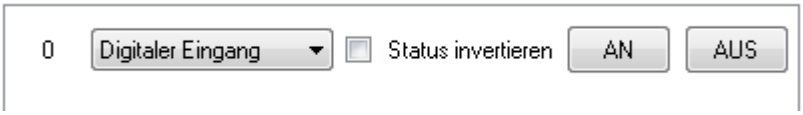
Der serielle Adapter hat eine andere Pinbelegung als der USB-Adapter und liefert keine +5V Busspannung.
Zur Spannungsversorgung wird ein stabilisiertes 5V-Netzteil mit Strombegrenzung empfohlen.
Für kleinere Systeme wird empfohlen ein Modular-Kabel mit RJ12 oder RJ45 Stecker zum Selbercrimpen und zur Verteilung eine 3er-Modular-Buchse zu nutzen. Das System wird dann 1:1 verkabelt (sternförmige Verkabelung).

**Weitere Links:**
Wikipedia: de.wikipedia.org/wiki/1-Wire
Hersteller: [https://www.maximintegrated.com/en/pl_list.cfm/filter/21]
**COM Port eines USB Geräts auslesen unter Windows**

| | |
|---|---|
| DS2405 | 1-fach Switch |
| OW_SwitchMode | setzt den Pin vom DS2405 auf an/aus |
| OW_ToggleMode | toggelt den Pin vom DS2405 |
| DS2406 | 2-fach Switch |
| OW_SetPin | setzt einen Pin vom DS2406 auf an/aus |
| DS2408 | 8-fach Switch |
| OW_SetPin | setzt einen Pin vom DS2408 auf an/aus |
| OW_SetPort | setzt den kompletten Port vom DS2408 auf Wert |
| OW_SetStrobe | schaltet den Strobe vom DS2408 an/aus |
| OW_WriteBytes | sendet mehrere Bytes an den DS2408 |
| OW_WriteBytesMasked | sendet mehrere Bytes an den DS2408 unter Berücksichtigung einer Bitmaske |
| DS2413 | 2-fach Switch |
| OW_SetPin | setzt einen Pin vom DS2413 auf an/aus |
| DS2438 | Smart Battery Monitor |
| DS2450 | A/D-Wandler 16bit |
| OW_SetPin | setzt einen Pin vom DS2450 auf an/aus |
| DS2890 | Digital-Potentiometer 8 bit |
| OW_SetPosition | setzt die Position vom DS2490 auf an/aus |
| Geräteliste | Geräteliste 1-Wire |
| OW_RequestStatus | sende eine Leseanfragen an das OneWire Gerät |

# DS2405

Der DS2405 ist ein 1-fach Switch (4mA) & Input (PIO) Baustein.
Dieser wird nicht mehr hergestellt. Als Ersatztyp wird der DS2413 empfohlen.
Der Pin kan wahlweise als “Digitaler Ausgang” oder “Digitaler Eingang” eingestellt werden.
Zusätzlich kann der Status über ”Status invertieren” invertiert werden(siehe DS2413).
Zu Testzwecken kann der Ausgang “AN" und “AUS” geschaltet werden.



# OW_SwitchMode

```
boolean OW_SwitchMode (int $InstanzID, boolean $Status)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Status | TRUE für An, FALSE für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Setzt den Zustand des Geräts mit der ID **InstanzID** auf Wert **Status**



Da der Baustein kein definiertes Setzen unterstützt, funktioniert der Befehl auf dem Prinzip, den Eingang zu lesen und für den Fall, dass Status <> Eingangsstatus ist, den Ausgang des Bausteins zu toggeln. Beim Design externer Schaltungen sollte dies unbedingt beachtet werden.

## BEISPIEL

```
OW_SwitchMode(12345, true); //Gerät einschalten
```

# OW_ToggleMode

```
boolean OW_ToggleMode (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Wechselt den Zustand des Geräts mit der ID **InstanzID**. Der aktuelle Wert kann nach dem Vorgang in der Status Variable abgefragt werden.

## BEISPIEL

```
OW_ToggleMode(12345); //Gerät togglen
```

# DS2406

Der DS2406 ist ein 2-fach Switch (50/8mA), Input (PIO) Baustein & 1Kb Memory Baustein.

# OW_SetPin

```
boolean OW_SetPin (int $InstanzID, int $Pin, boolean $Status)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Pin | Pin des Bausteins (0..1) |
| Status | **TRUE** für An, **FALSE** für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den Zustand vom Pin **Pin** des Geräts mit der ID **InstanzID** auf Wert **Status**

## BEISPIEL

```
OW_SetPin(12345, 0, true); //Gerät einschalten
```

# DS2408

Der DS2408 und DS2408+ ist ein 8-fach Switch & Input (PIO) Baustein.
Jeder der acht Kanäle kann als “Digitaler Ausgang” oder “Digitaler Eingang” eingestellt werden.
Zusätzlich kann der Status über ”Status invertieren” invertiert werden (siehe DS2413).
Zu Testzwecken kann der Ausgang “AN" und “AUS” geschaltet werden.



# OW_SetPin

```
boolean OW_SetPin (int $InstanzID, int $Pin, boolean $Status)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Pin | 0-7 |
| Status | **TRUE** für An, **FALSE** für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den Zustand vom Pin **Pin** des Geräts mit der ID **InstanzID** auf Wert **Status**

## BEISPIEL

```
OW_SetPin(12345, 2, true); //Pin 2 vom Gerät einschalten
```

# OW_SetPort

```
boolean OW_SetPort (int $InstanzID, int $Bitmaske)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Bitmaske | 0-255 (Bit0 = Pin0 … Bit7 = Pin7) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den Zustand von allen acht Pins auf den Wert von **Bitmaske** des Geräts mit der ID **InstanzID**.

## BEISPIEL

```
OW_SetPort(12345, 0); //Alles ausschalten
```

# OW_SetStrobe

```
boolean OW_SetStrobe (int $InstanzID, boolean $Status)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Status | TRUE für An, FALSE für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Ermöglicht es, den RSTZ-Pin des DS2408 des Geräts mit der ID InstanzID auf Wert Status zu setzten

## BEISPIEL

```
OW_SetStrobe(12345, true); //Strobe einschalten
```

# OW_WriteBytes

```
boolean OW_WriteBytes (int $InstanzID, string $Daten)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Daten | Folge von Bytes, die versendet werden sollen |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt eine Abfolge von Bytes (**Daten**) über alle Pins des Geräts mit der ID **InstanzID** ab.



Nur als Output definierte Pins werden beschrieben.

## BEISPIEL

```
OW_WriteBytes(12345, Chr(0).Chr(1)); //Abfolge senden
```

# OW_WriteBytesMasked

```
boolean OW_WriteBytesMasked (int $InstanzID, string $Daten, int
$Maske)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Daten | Folge von Bytes, die versendet werden sollen |
| Maske | Bitmaske der Bit die gesetzt werden sollen |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt eine Abfolge von Bytes (**Daten**)unter Berücksichtigung einer Bitmaske **Maske** über alle Pins des Geräts mit der ID **InstanzID** ab.



Nur als Output definierte Pins werden beschrieben!

## BEISPIEL

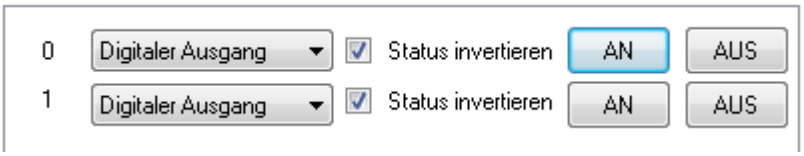
```
OW_WriteBytesMasked(12345, Chr(255), Chr(1)); //Nur Bit 0 setzen
```

# DS2413

Der DS2413 ist ein 2-fach Switch & Input (PIO) Baustein.
Seine Ausgänge können 20mA liefern. Weitere Einzelheiten können dem Datenblatt entnommen werden.

Definition des "Status Pins":
Ein logischer Low-Pegel am Ausgang wird als “Aus” / “False” gewertet (hierbei ist der interne Transistor aktiv).

Wird z.B. eine LED mit gemeinsamer Anode angeschlossen, leuchtet diese bei “Status Pin == Aus”.
In diesem Fall sollte “Status invertieren” aktiviert werden.




Wird ein Pin als “Digitaler Eingang” verwendet, so ist darauf zu achten, dass der DS205 kein Latch besitzt. Es werden also keine Schaltimpulse gespeichert.

Um z.B. einen PIRI (Bewegungsmelder) auszuwerten, sollte der Timer auf zehn Sekunden eingestellt werden. Zusätzlich ist am Eingang, ein Widerstand 1M sowie ein Kondensator 10µF vorzusehen. Beide werden gemeinsam mit 0V verbunden. Ein Schaltimpuls zieht den Eingang kuzfristig auf +5V. Der Kondensator puffert solange die Spannung, bis der nächste Abfragezyklus den Pin als HI erkennt.

# OW_SetPin

```
boolean OW_SetPin (int $InstanzID, int $Pin, boolean $Status)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Pin | 0-1 |
| Status | TRUE für An, FALSE für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Setzt den Zustand vom Pin Pin des Geräts mit der ID InstanzID auf Wert Status

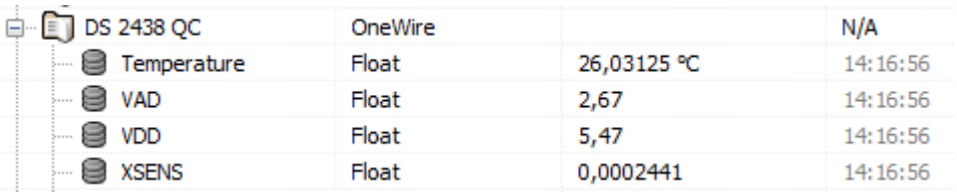
## BEISPIEL

```
OW_SetPin(12345, 0, true); //Pin 0 einschalten
```

# DS2438

Der DS2438 ist ursprünglich ein “Smart Battery Monitor”.
Für den Einsatz in der Hausautomation besitzt er vier funktionale Eigenschaften:

Ein Spannungsmesser, dieser liefert die Busspannung (VDD)
Ein Thermometer – misst die Temperatur (Temperatur)
Ein AD-Wandler (VAD) – kann u.a. zur Messung der Feuchtigkeit oder Helligkeit verwendet werden
Einen Stromeingang (XSENS) - zur freien Verfügung



| | | | |
|---|---|---|---|
| DS 2438 QC | OneWire | | N/A |
| Temperature | Float | 26,03125 °C | 14:16:56 |
| VAD | Float | 2,67 | 14:16:56 |
| VDD | Float | 5,47 | 14:16:56 |
| XSENS | Float | 0,0002441 | 14:16:56 |

Praxisbeispiel: Feuchtigkeitsmessung
Der Feuchtigkeitsfühler **HIH-4000** von HONEYWELL besitzt eine Genauigkeit von 3,5% im Bereich von 0 to 100% RH.
Wegen seines linearen Ausgangs kann er direkt mit dem DS2438 ausgelesen werden. Da dieser einen internen Temperatur-Sensor hat, kann auch eine Temperatur-Kompensation berücksichtigt werden.
Beispielskript:

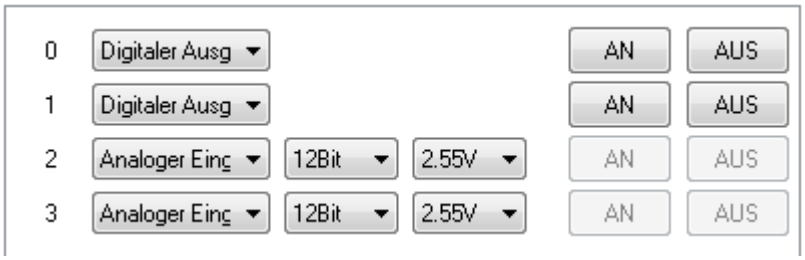
```
$Vad = GetValue(44045);
$Vdd = GetValue(14570);
$temp = GetValue(18691);
$Srh = ($Vad - 0.958062) * 30.680;
$Srh = $Srh / ((1.0305 + (0.000044 * $temp) - (0.0000011 *
pow($temp,2))));
echo "Feuchte: $Srh %rh Temp.Comp.\n";
SetValue(49935 /*[Feuchte %rh]*/, $Srh);
```

# DS2450

Der DS2450 ist ein A/D-Wandler-Baustein mit einer Auflösung von bis zu 16-Bit.
Zusätzlich können die vier Kanäle auch als Ausgang (4mA) verwendet werden.
Weitere Einzelheiten können dem Datenblatt entnommen werden.

## KONFIGURATION

Jeder der vier Kanäle kann als “Digitaler Ausgang” oder “Analoger Eingang” eingestellt werden.
Zusätzlich sind zwei Spannungsbereiche wählbar: 2,55V sowie 5,1V.
Desweiteren ist die Auflösung zwischen 1-Bit und 16-Bit frei einstellbar:



# OW_SetPin

```
boolean OW_SetPin (int $InstanzID, int $Pin, boolean $Status)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Pin | 0-3 (A-D) |
| Status | TRUE für An, FALSE für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Setzt den Zustand vom Pin Pin des Geräts mit der ID InstanzID auf Wert Status

## BEISPIEL

```
OW_SetPin(12345, 2, true); //Pin C vom Gerät einschalten
```

# DS2890

Der DS2890 ist ein Digital-Potentiometer mit 8-Bit-Auflösung.
Dieser wird nicht mehr hergestellt. Einen Ersatztyp gibt es nicht.




Um auf dem 1-Wire-Center 0 bis 10V auszugeben, sollte “CHARGE PUMP CONSIDERATIONS” verwendet werden. Hierzu muss “Benutze CPC” aktiviert werden.
Für einen Test kann der Schieberegler verändert werden und “SETZEN” gedrückt werden.

# OW_SetPosition

```
boolean OW_SetPosition (int $InstanzID, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | 0-255 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt das Gerät mit der ID **InstanzID** auf Wert **Wert**



Bitte beachten Sie den CPC Parameter in der Konfiguration.

## BEISPIEL

```
OW_SetPosition(12345, 123); //Gerät auf den Wert 123 setzen
```

# Geräteliste

## UNTERSTÜTZTE GATEWAYS

| GATEWAY | BESCHREIBUNG |
|---|---|
| Symcon 1-Wire LAN Gateway<br>Produktinformation \| Jetzt bestellen | IP Gateway |
| DS2480B | serielles Interface |
| DS2490 (Nur Windows) | USB-Adapter |
| DS9097U | serielles Interface |
| DS9490R (Nur Windows) | USB-Adapter |
| Link45 | serielles Interface |
| LinkUSB | USB-Adapter |

## UNTERSTÜTZTE KOMPONENTEN

| PRODUKT | BESCHREIBUNG |
|---|---|
| DS1820 | Thermometer -55°C bis +125°C, 0.5 genau bei -10°C bis +85°C |
| DS18B20 | Thermometer -55°C bis +125°C, 0.5 genau bei -10°C bis +85°C |
| DS18S20 | Thermometer -55°C bis +125°C, 0.5 genau bei -10°C bis +85°C |
| DS1920 | Thermometer -55°C bis +100°C, 0.5 |
| DS2405 | 1-fach: Switch & Input (PIO) |
| DS2406 | 2-fach: Switch & Input (PIO) |
| DS2407 | 2-fach: Switch & Input (PIO) |
| DS2408 | 8-fach: Switch & Input (PIO) |
| DS2408+ | 8-fach: Switch & Input (PIO) |
| DS2413 | 2-fach: Switch & Input (PIO) |
| DS2423P | 4KB Zähler |
| DS2438 | Strom- und Spannungsmessung |
| DS2450 | A/D-Wandler |
| DS2890 | Digital Potentiometer |

# OW_RequestStatus

```
boolean OW_RequestStatus (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Führt einen erneuten Lesevorgang für das Gerät mit der ID **InstanzID** aus.
Sobald die Funktion ausgeführt wurde, sind die neuen Werte in den Status-Variablen abrufbar.

## BEISPIEL

```
OW_RequestStatus(12345); //Gerät auslesen
```

# ABL

Benötigt: IP-Symcon >= 7.0
ABL bietet verschiedene Wallboxen/Ladesäulen an. Über ModBus TCP können diese ausgelesen werden. Eine Verbindung mit IP-Symcon ist über ein LAN-IP Gateway möglich.

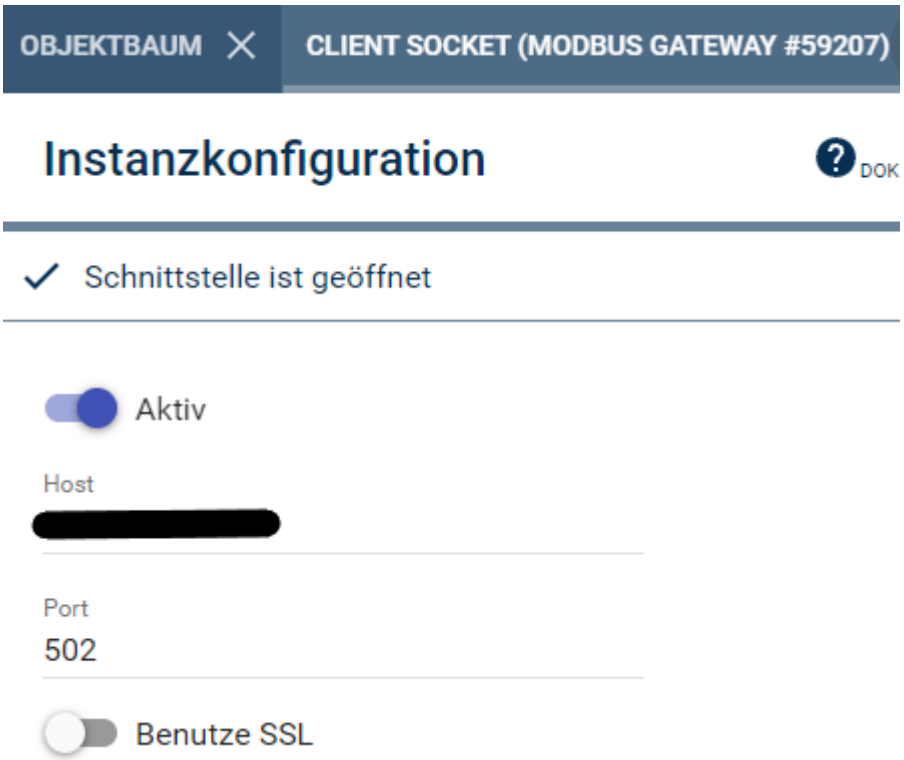
Folgende Geräte werden von IP-Symcon unterstützt:
Unterstützte Komponenten

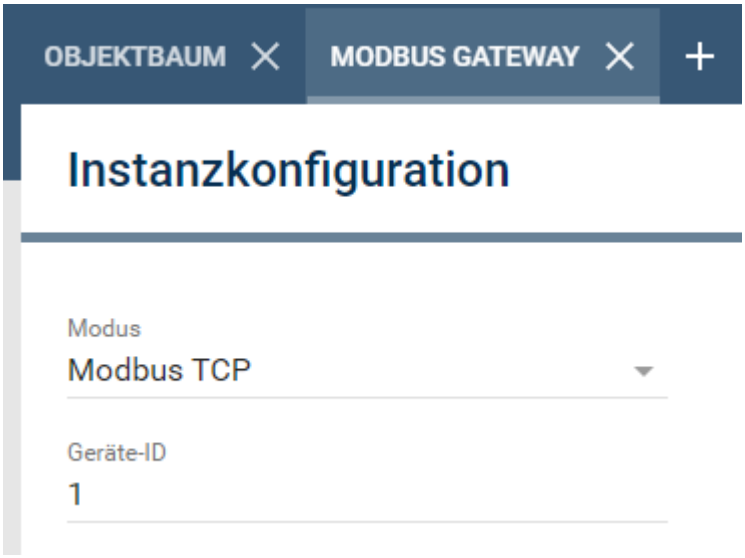
## INSTALLATION

Um ABL Wallboxen/Ladesäulen in IP-Symcon nutzen zu können, muss eine Verbindung via Ethernet vorhanden sein. Innerhalb der Wallbox/Ladesäulen muss lediglich der Ethernet RJ45 Stecker nach Anleitung eingesteckt werden.

## EINBINDUNG IP-SYMCON

Zuerst muss eine "ModBus Device" Instanz innerhalb des Objektbaums von IP-Symcon hinzugefügt werden. Im folgenden Dialog muss die IP-Adresse der Wallbox eingetragen werden. Der Port ist Standardmässig 502. Beide Informationen können auf der Weboberfläche der WallBox/Ladesäule eingesehen werden.

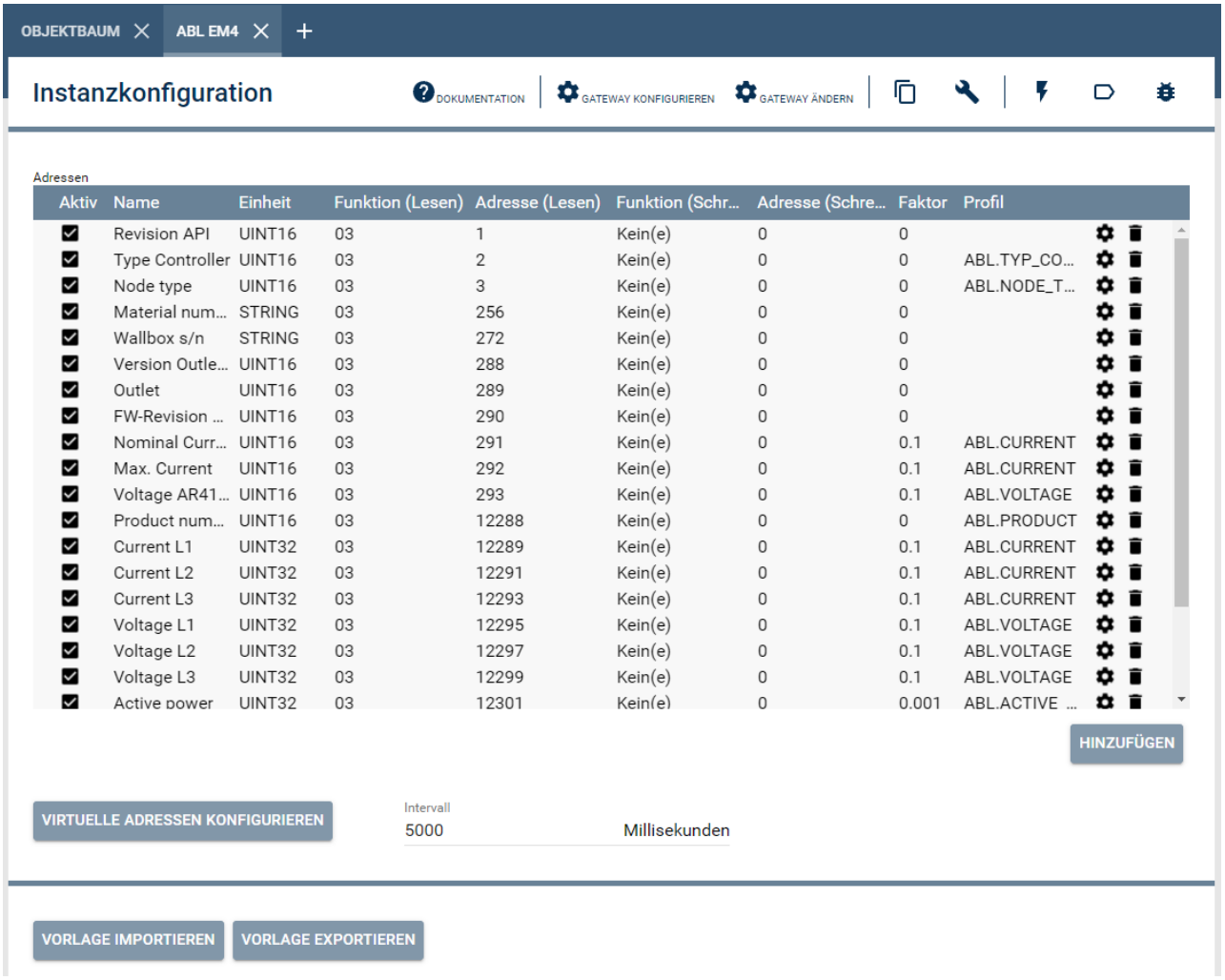





Im Anschluss kann die ModBus Vorlage für ABL Wallboxen/Ladesäulen heruntergeladen werden. Diese beinhaltet die ganze Konfiguration des ModBus Geräts. Nach dem Download, kann die "ABLmodbusX_vx.json" über "Importieren" geladen werden. Anschließend sind alle ModBus Adressen der gängigen ABL Wallboxen/Ladesäulen eingerichtet.

ModBus Vorlage für ABL Wallboxen/Ladesäulen:
Download eMH1
Download eM4
Diese beinhaltet den Outlet/Zähler 1.
Sollten weitere benötigt werden kann der Support kontaktiert werden.

eMH2 und eMH3 können über OCPP genutzt werden.




## HINZUFÜGEN VON ADRESSEN

Sollen weitere Adressen hinzugefügt werden, kann dies über "Hinzufügen" realisiert werden.
Je nach Ausführung der Wallbox kann es sein, dass einzelne Adressen de-/aktiviert werden müssen. Dies kann über die Aktiv Spalte gesteuert werden.

# Geräteliste

Benötigt: IP-Symcon >= 7.0

## UNTERSTÜTZTE KOMPONENTEN

| SYSTEM | BESCHREIBUNG |
| --- | --- |
| Wallbox eMH1 | In allen Versionen unterstützt |
| Wallbox eMH2 | In allen Versionen unterstützt |
| Wallbox eMH3 | In allen Versionen unterstützt |
| Wallbox eM4 | In allen Versionen unterstützt |
| Wallbox eMC | In allen Versionen unterstützt |

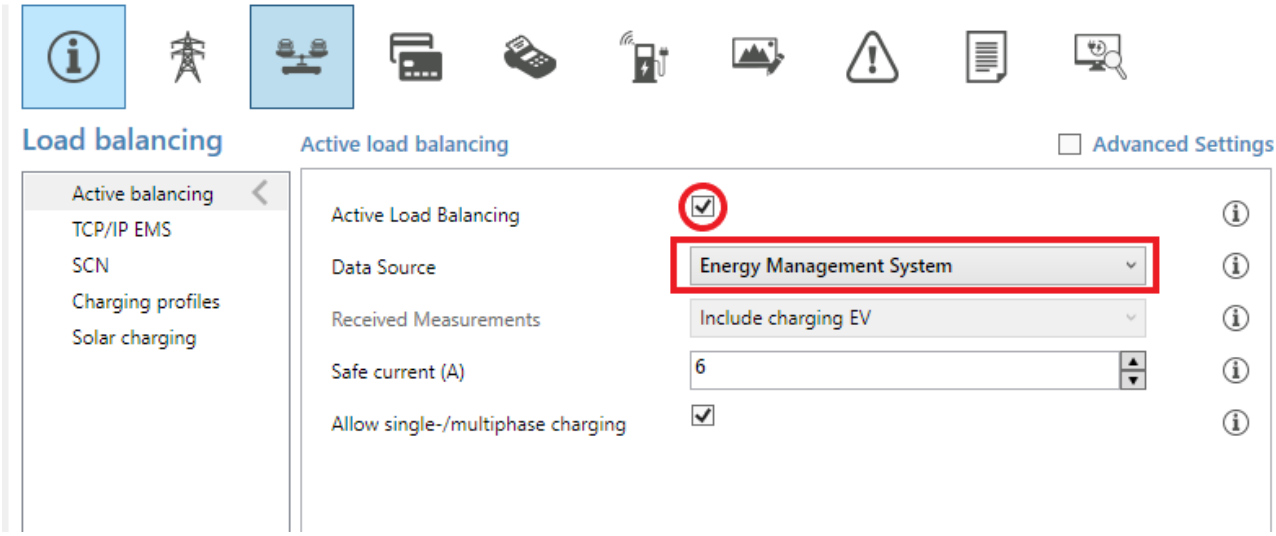
# Alfen

Benötigt: IP-Symcon >= 7.0
Alfen bietet verschiedene Wallboxen/Ladesäulen an. Über ModBus TCP können diese ausgelesen werden. Eine Verbindung mit IP-Symcon ist über ein LAN-IP Gateway möglich.

Folgende Geräte werden von IP-Symcon unterstützt:
Unterstützte Komponenten

## INSTALLATION

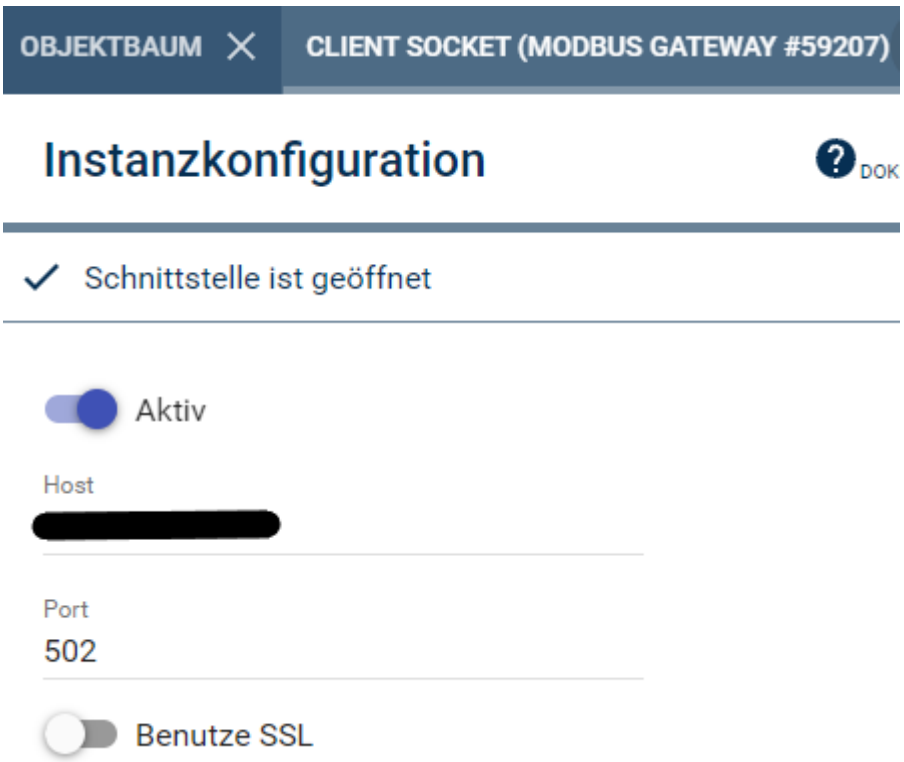
Um Alfen Wallboxen/Ladesäulen in IP-Symcon nutzen zu können, muss eine Verbindung via Ethernet zur Wallbox vorhanden sein. Innerhalb der Wallbox/Ladesäulen ist ein LAN Anschluss integriert.

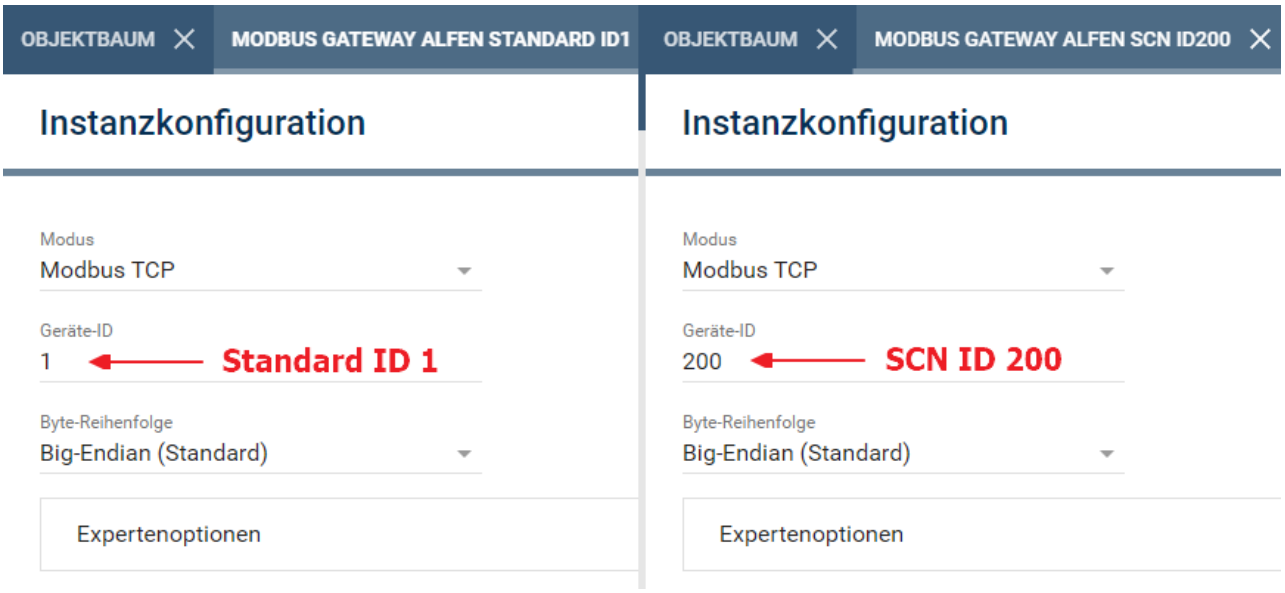



Innerhalb der ACE Service Software muss, wie im Bild einsehbar, das Active Load Balancing aktiviert und "Energy Managment System" ausgewählt werden. Dadurch wird auch das ModBus TCP aktiviert.

## EINBINDUNG IP-SYMCON

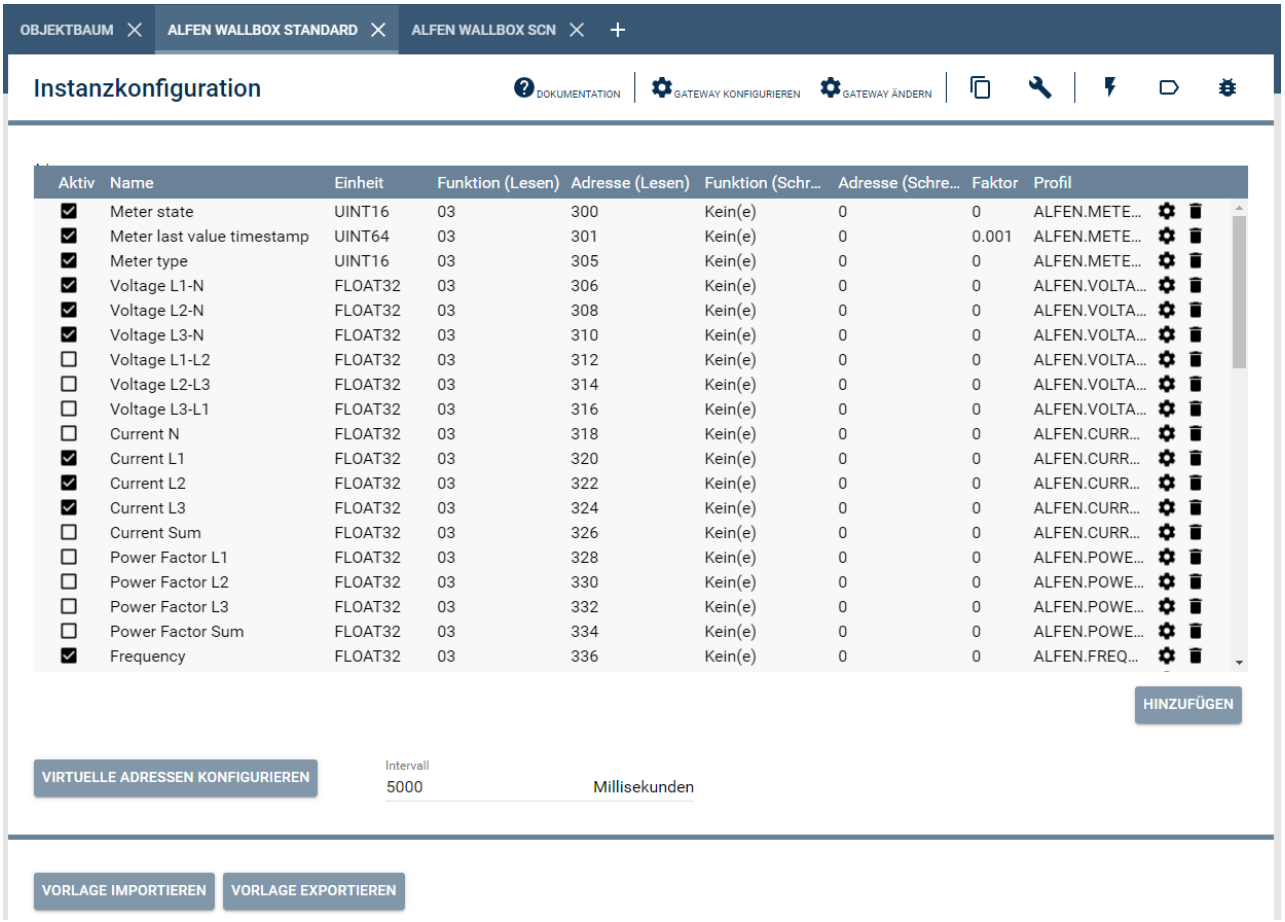
Zuerst muss eine "ModBus Device" Instanz innerhalb des Objektbaums von IP-Symcon hinzugefügt werden. Im folgenden Dialog muss die IP-Adresse der Wallbox eingetragen werden. Der Port ist Standardmässig 502. Beide Informationen können in der App oder ACE Service Software der WallBox/Ladesäule eingesehen werden.

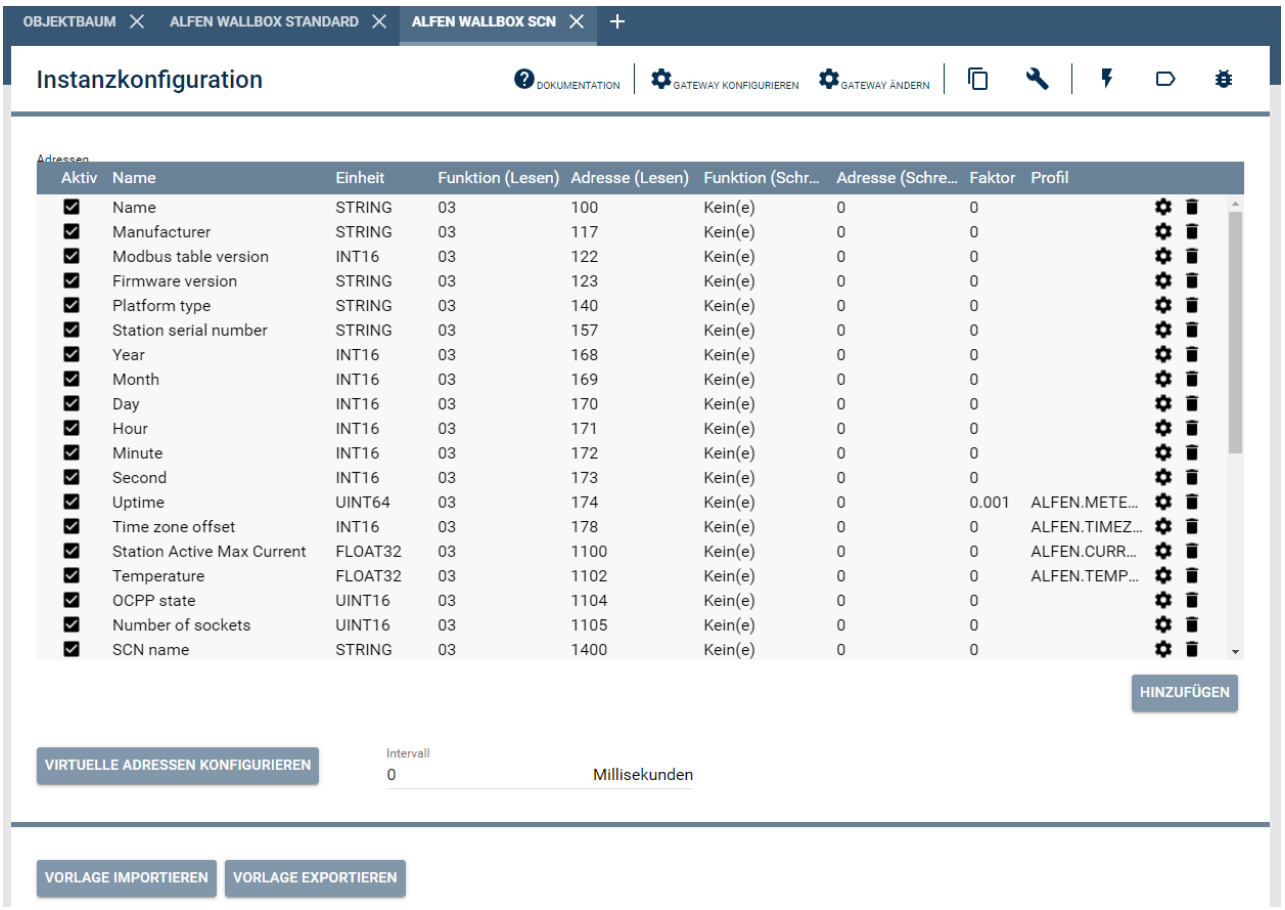



Im Anschluss kann die ModBus Vorlage ("AlfenModBusX_vx.json") für Alfen Wallboxen/ Ladesäulen heruntergeladen werden. In diesem Download befinden sich entweder die Standard- oder SCN-Konfiguration des ModBus-Geräts. Über "Importieren" kann die gewünschte Vorlage geladen werden. Anschließend sind alle ModBus Adressen der gängigen Alfen Wallboxen/Ladesäulen eingerichtet.

ModBus Vorlage für Alfen Wallboxen/Ladesäulen:
Download Standard
Download SCN



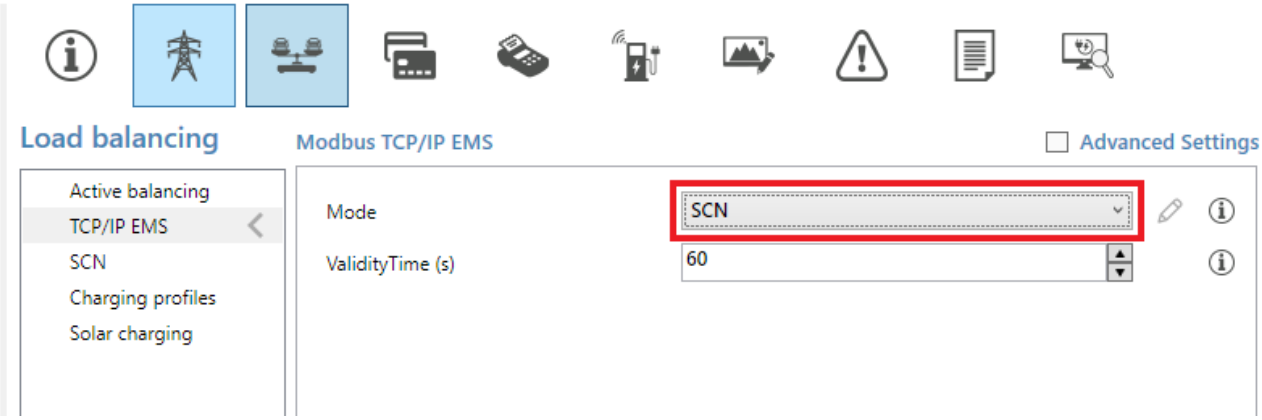




# HINZUFÜGEN VON ADRESSEN

Sollen weitere Adressen hinzugefügt werden, kann dies über "Hinzufügen" realisiert werden.
Je nach Ausführung der Wallbox kann es sein, dass einzelne Adressen de-/aktiviert werden müssen. Dies kann über die Aktiv Spalte gesteuert werden.

## STANDARD ODER SCN NUTZEN

In der ACE Service Software kann man, wie im Bild erkennbar, zwischen einem Standard Lastmanagment (Socket) oder SCN Betrieb wählen. SCN bildet mehrere Wallboxen ab, wobei Socket nur eine Wallbox unterstützt.




## PROTOKOLLBESCHREIBUNG

Protokollbeschreibung für Alfen Wallboxen/Ladesäulen: Download

# Geräteliste

Benötigt: IP-Symcon >= 7.0

## UNTERSTÜTZTE KOMPONENTEN

| SYSTEM | BESCHREIBUNG |
| --- | --- |
| Eve Single S-line | In allen Versionen unterstützt |
| Eve Single Pro-line | In allen Versionen unterstützt |
| Eve Single Pro-line DE | In allen Versionen unterstützt |
| Eve Double Pro-line | In allen Versionen unterstützt |
| Eve Double Pro-line DE | In allen Versionen unterstützt |
| Eve Double PG-line DE | In allen Versionen unterstützt |
| Eve Twin 4/5XL | In allen Versionen unterstützt |

# ALLNET

Die Firma ALLNET stellt Geräte zum Steuern, Messen und Regeln her. Diese kommunizieren IP basierend via LAN. Um diese mit IP-Symcon zu verbinden reicht es, IP-Symcon ins LAN zu integrieren.

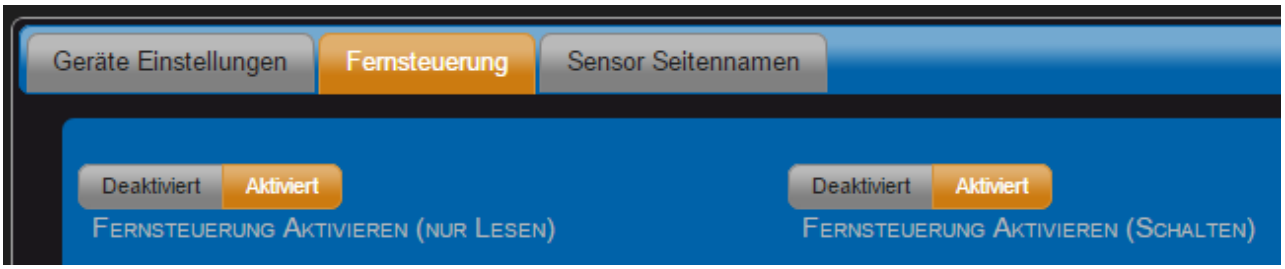
Folgende Geräte werden von IP-Symcon unterstützt:
Unterstützte Geräte

## INSTALLATION

Um IP-Symcon den Zugriff auf ein ALLNET-Gerät zu ermöglichen muss folgende Geräteeinstellung vorgenommen werden.
Über die Geräte-IP in einem beliebigen Browser gelangt man zur geräteinternen Konfigurationsseite.
Den Reiter "Fernsteuerung" findet man unter dem Menüpunkt Konfiguration -> Geräte Einstellungen.




Hier müssen beide Fernsteuerungsoptionen aktiviert werden.
Ab sofort hat IP-Symcon vollen Zugriff auf das Gerät.

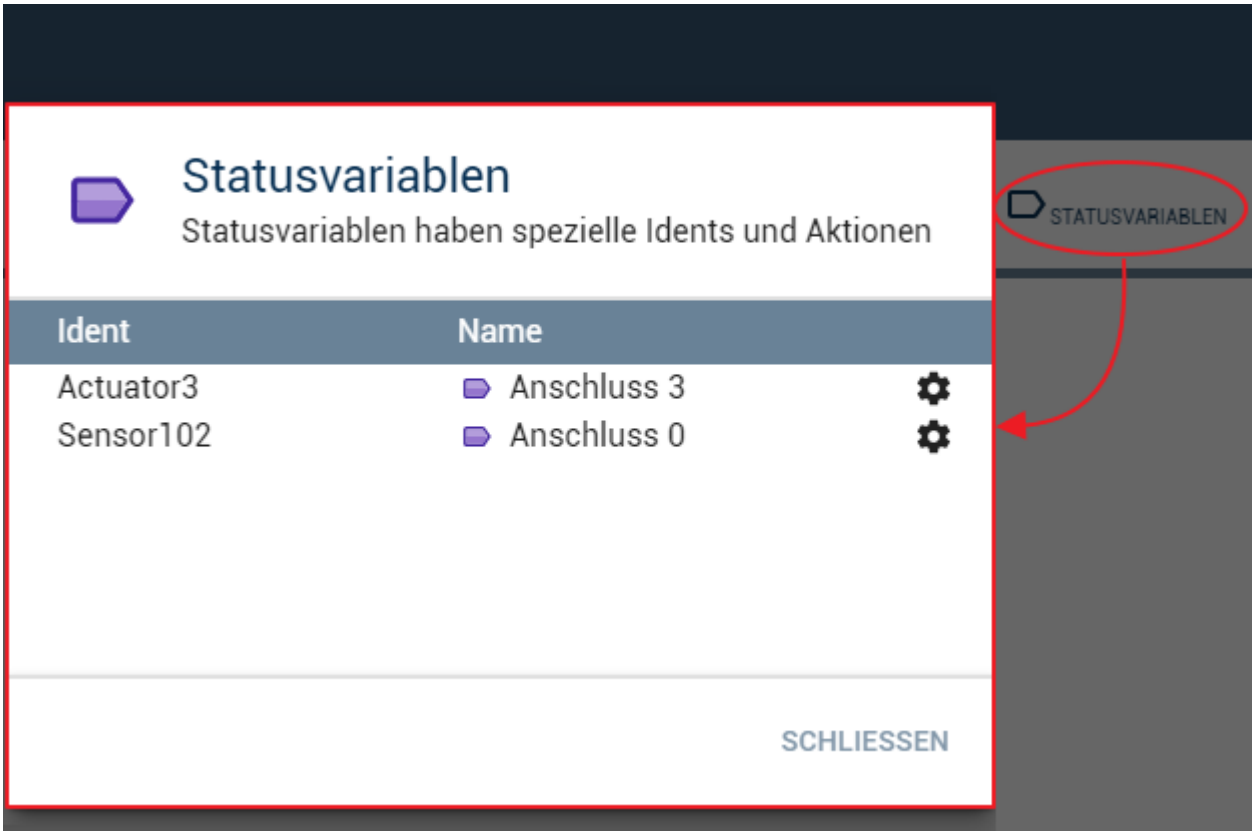



Die Fernsteuerung muss nach einem Firmwareupgrade erneut aktiviert werden!

## KANALID HERAUSFINDEN

Über die Statusvariablen können die KanalIDs herausgefunden werden. Dazu müssen innerhalb der Konfigurationsseite die Statusvariablen geöffnet werden. Die Zahl am Ende

der Idents spiegelt auch die KanalID wieder.
*In diesem Beispiel Actuator3 = KanalID 3*



# ALL_SetAnalog

Benötigt: IP-Symcon >= 5.0

```
boolean ALL_SetAnalog (int $InstanzID, int $KanalID, float $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| KanalID | ID des zu schaltenden Kanals |
| Wert | Wert auf den der Kanal gesetzt werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den Wert des Kanals mit der KanalID **Kanal** des Geräts mit der ID **InstanzID** auf den Wert **Wert**



Die KanalID ist über die Statusvariablen der Instanz ersichtlich.
Siehe KanalID herausfinden

## BEISPIEL

```
// Setzt den Kanal 3 des Geräts 12345 auf 20,4
ALL_SetAnalog(12345, 3, 20.4); //Gerät einschalten
```

# ALL_SwitchActor

```
boolean ALL_SwitchActor (int $InstanzID, int $KanalID, boolean
$Status)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| KanalID | ID des zu schaltenden Kanals |
| Status | **TRUE** für An, **FALSE** für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schaltet den Aktor mit der Geräte ID **InstanzID** auf dem Kanal **KanalID** auf den Wert **Status**



Die KanalID ist über die Statusvariablen der Instanz ersichtlich.
Siehe KanalID herausfinden

## BEISPIEL

```
ALL_SwitchActor(12345, 1, true); //Aktor des Geräts 12345 auf dem
Kanal 1 einschalten
```

# ALL_SwitchMode

```
boolean ALL_SwitchMode (int $InstanzID, boolean $Status)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Status | TRUE für An, FALSE für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Schaltet das Gerät mit der ID InstanzID auf Wert Status

## BEISPIEL

```
ALL_SwitchMode(12345, true); //Gerät einschalten
```

# ALL_UpdateValues

```
boolean ALL_UpdateValues (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Führt eine Aktualisierungsanfrage für das Gerät mit der ID InstanzID aus.
Sobald die Funktion ausgeführt wurde, sind die neuen Werte in den Status-Variablen abrufbar.

## BEISPIEL

```
//Das Gerät mit der ID "12345" auslesen
ALL_UpdateValues(12345);
```

# Geräteliste

## UNTERSTÜTZTE KOMPONENTEN

Ein Auszug unterstützter Komponenten.
Auf Anfrage überprüfen wir gerne weitere, nicht aufgelistete, Geräte auf ihre Kompatibilität.

### ALLNET GERÄTE MIT FIRMWARE 3.X

| PRODUKTBEZEICHNUNG | BESCHREIBUNG |
|---|---|
| ALL3040 | WiFi Basic Sensor Zentrale |
| ALL3072 | Netzwerksteckdose zum Schalten über LAN/W-LAN |
| ALL3073WLAN | Netzwerksteckdose zum Schalten über LAN/W-LAN |
| ALL3075 V2 | Netzwerksteckdose zum Schalten über LAN (inkl. Verbrauchsmessung) |
| ALL3075 V3 | Netzwerksteckdose zum Schalten über LAN (inkl. Verbrauchsmessung) |
| ALL3418 V2 | IP Triometer |
| ALL3419 | IP Thermometer/Sensorzentrale |
| ALL3500 | IP Homeautomation Appliance |
| ALL3505 | IP Homeautomation Appliance |
| ALL3653 | Remote Control |
| ALL3690 | PM1 PowerMeter (3 Phasen) |
| ALL3691 | PM2 PowerMeter (2x3 Phasen, 2xS0, 2xD0) |
| ALL3692 | PM3 PowerMeter (4xD0, 2xI2C) |
| ALL3696 | IP PowerMeter |
| ALL3697-32A | PM4 PowerMeter (HUT Powermeter, 3 Phasen) |
| ALL4075 | 4-fach Netzwerkrelais (LAN/WLAN) |
| ALL4076 | 6-fach IP Steckdosenleiste |
| ALL4175 | 4-fach Netzwerkrelais |
| ALL4176 | 6-fach IP Steckdosenleiste |
| ALL4500 | Sensoric Appliance |
| ALL4550 | POE Informationsdisplay |
| ALL5000 | IP Facility Control Server |

### ALLNET GERÄTE MIT FIRMWARE 2.X

| PRODUKTBEZEICHNUNG | BESCHREIBUNG |
|---|---|
| ALL3000 | Internet Thermometer |
| ALL3000RF MK2 | Steuerungseinheit für Intertechno Funksteckdosen |
| ALL3075 | Netzwerksteckdose zum Schalten über LAN |
| ALL3076 | Netzwerksteckdose zum Dimmen über LAN |
| ALL3090 / ALL3096 | 8 Port Reset Switch |
| ALL3100 | Ethernet Power Switch |
| ALL3270 | Narrowband Powerline Master Ethernet Bridge |
| ALL3275 | Narrowband Powerline Slave Steckdose zum Schalten über Powerline |
| ALL3418 | IP Triometer |
| ALL3421 | IP Triometer |
| ALL4000 | Ethernet Sensormeter |
| ALL4001 | HUT Ethernet Sensormeter für Hutschiene |
| ALL4100 | 8 Port Power Switch 19” mit 8 Kaltgerätesteckdosen |

# BACnet

Benötigt: IP-Symcon >= 6.3



Das BACnet Modul ist eine kostenpflichtige Erweiterung, welche zu jeder bestehenden Symcon Lizenz im Shop erworben werden kann. Bei jeder Symcon Lizenz können bis zu 3 BACnet Objekte kostenlos erstellt werden.

Die BACnet (Client) Anbindung erlaubt das komfortable Einrichten von BACnet Geräten innerhalb von Symcon. Dabei werden über eine Discovery Instanz alle im lokalen Netzwerk verfügbaren BACnet Geräte gesucht und entsprechend auch angelegt. Im nächsten Schritt wird jedes angelegte Gerät abgefragt und die auf dem Gerät verfügbaren Datenpunkte dargestellt. Es können beliebige Datenpunkte erstellt werden bzw. im einfachsten Fall alle Datenpunkte in Symcon importiert werden. Die Datenpunkte zeigen dabei den "Present Value" an, welcher sich automatisch über COV aktualisiert, sofern das Gerät dies unterstützt. Alternativ kann der Wert auch zyklisch abgefragt werden. Der Wert kann zusätzlich per Visualisierung, Skript und Ablaufplan verändert werden.
Der BACnet (Server) erlaubt es über einen Konfigurator beliebige Variablen aus Symcon heraus nach BACnet zu bringen. Dabei können die verfügbaren Datenpunkte automatisch vom Client ausgelesen werden und auch COV wird vom Server entsprechend unterstützt. Ebenfalls kann Symcon als BACnet Geräte automatisch im Netzwerk erkannt werden.

## VIDEO-TUTORIAL ZUR EINRICHTUNG

### UNTERSTÜTZTES PROTOKOLL

BACnet/IP

### UNTERSTÜTZTE BIBBS (CLIENT)

DM-DDB-A (Dynamic Device Binding)
DS-RP-A (Read Property)
DS-RPM-A (Read Property Multiple)
DS-WP-A (Write Property)
DS-COV-A (Change of Value Notification)
DS-COVP-A (Change Of Value Property)

### UNTERSTÜTZTE BIBBS (SERVER)

DM-DDB-B (Dynamic Device Binding)
DS-RP-B (Read Property)
DS-RPM-B (Read Property Multiple)
DS-WP-B (Write Property)
DS-COV-B (Change of Value Notification)
DS-COVP-B (Change Of Value Property)

## UNTERSTÜTZTE BIBBS (SONSTIGE)

NM-FDR-A (Foreign Device Registration)

## UNTERSTÜTZTE STANDARD OBJECT TYPES

Analog Input
Analog Output
Analog Value
Binary Input
Binary Output
Binary Value
MultiState Input
MultiState Output
MultiState Value
Accumulator

## UNTERSTÜTZTE SEGMENTIERUNG

Empfangen: Ja (>64 Segmente, Max 1476 Bytes pro Segment)
Senden: Nein (bisher nicht erforderlich)

## BEISPIEL EINES WRAPPER MODUL FÜR NICHT STANDARDKONFORME GERÄTE

Integriert die SageGlass SIM II (BACnet) in Symcon

# BAC_RelinquishPresetValue

```
boolean BAC_RelinquishPresetValue (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Zurücksetzen des Wertes, damit auch Werte niedrigerer Priorität gesetzt werden können.

## BEISPIEL

```
BAC_RelinquishPresetValue(12345);
```

# digitalSTROM

DigitalSTROM ist ein System, welches die Stromnetz-Verdrahtung nutzt. Eine Verbindung mit IP-Symcon wird via LAN(IP) mit dem digitalSTROM Server hergestellt.
Desweiteren wird in dem zu bedienenden Gerät eine digitalSTROM-Klemme eingebaut, bzw. die herkömmliche Lüsterklemme ersetzt.
Alle elektrischen Geräte, die via digitalSTROM-Klemme mit dem digitalSTROM-Server kompatibel sind, sind auch mit IP-Symcon kompatibel und können eingebunden werden.

## VIDEO-TUTORIAL ZUR EINRICHTUNG

## ANSCHLUSS

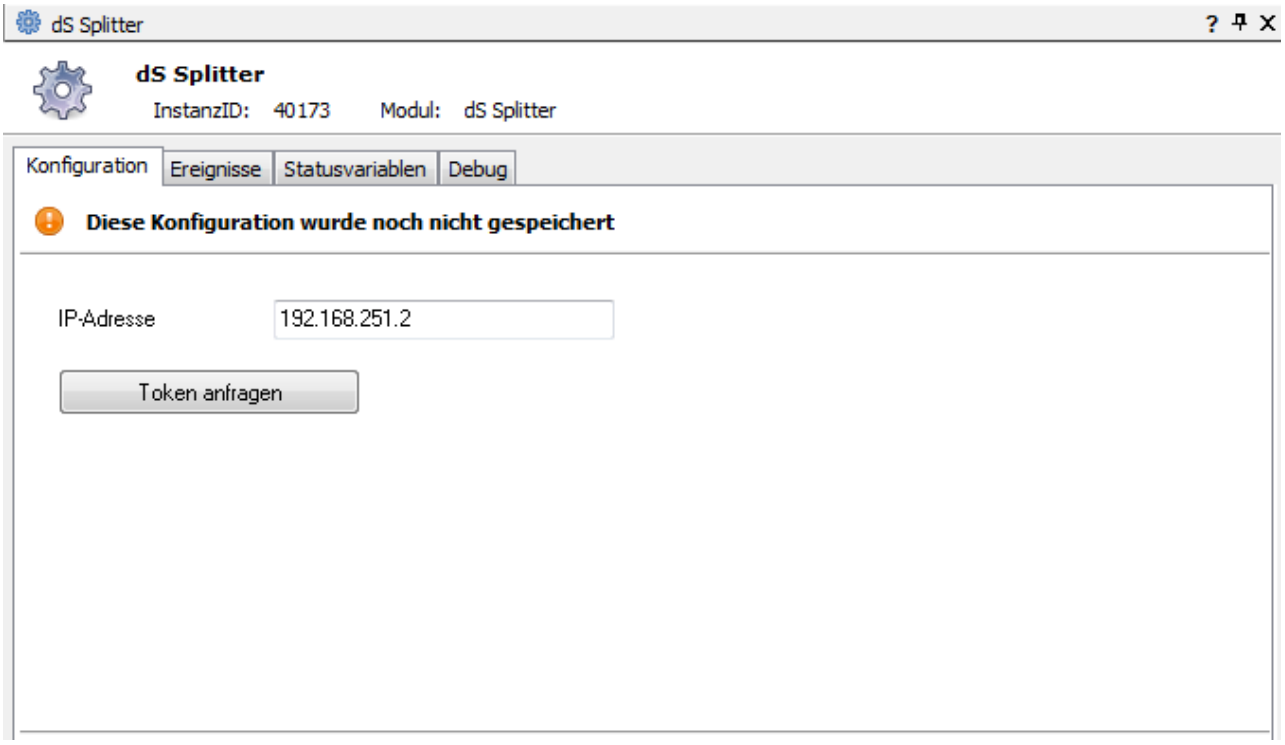
Bevor der digitalSTROM-Server mit dem PC verbunden wird, sollten alle zu steuernde elektrische Geräte, die mit dem PC bedient werden sollen, korrekt via digitalSTROM-Klemme mit diesem verbunden sein. Anschließend muss der digitalSTROM-Server durch ein Netzwerkkabel mit dem Netzwerk verbunden werden.

## INSTALLATION

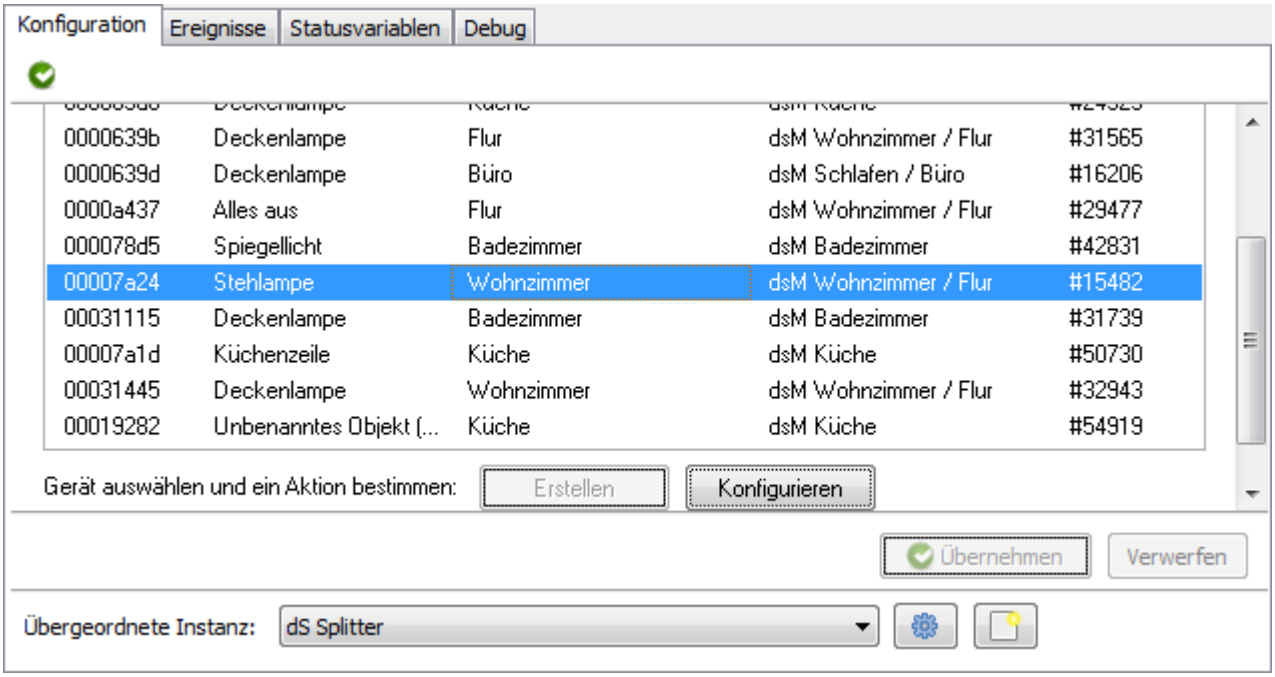
Wenn der digitalSTROM-Server korrekt angeschlossen ist und ein DHCP-Server verwendet wird, ist der digitalSTROM-Server nun mit dem lokalen Netzwerk verbunden.
(Der DHCP-Server übernimmt automatisch die Adressvergabe für die Netzwerkteilnehmer.)
Der digitalSTROM-Server erscheint in der Netzwerkumgebung unter „Andere Geräte“ als „dSS“.
Durch "Doppelklicken" auf „dSS“ erscheint ein Fenster in dessen unteren Bereich die IP-Adresse des digitalSTROM-Servers befindet. Nachdem diese kopiert und in einen beliebigen Browser kopiert wurde, öffnet sich die digitalSTROM Konfigurations-Webpage.
Es erscheint möglicherweise eine Sicherheitswarnung, welche mit "Ich kenne das Risiko" bestätigt werden muss.
Beim ersten Login gilt: "Name" = "dssadmin" und "Passwort" = "dssadmin".
Das Passwort sollte nach dem ersten Login geändert werden. Groß- und Kleinschreibung ist zu beachten.
Auf der Konfigurations-Webpage können einige Voreinstellungen vorgenommen werden.
Es ist auch möglich, die IP-Adresse manuell zu vergeben.

## EINBINDUNG IN IP-SYMCON

Innerhalb der Verwaltungskonsole kann über "Willkommenseite -> Konfiguratoren verwalten" ein "dS Konfigurator" erstellt werden. Über das Zahnrad im unteren Fensterbereich kann der "dS Splitter" konfiguriert werden.

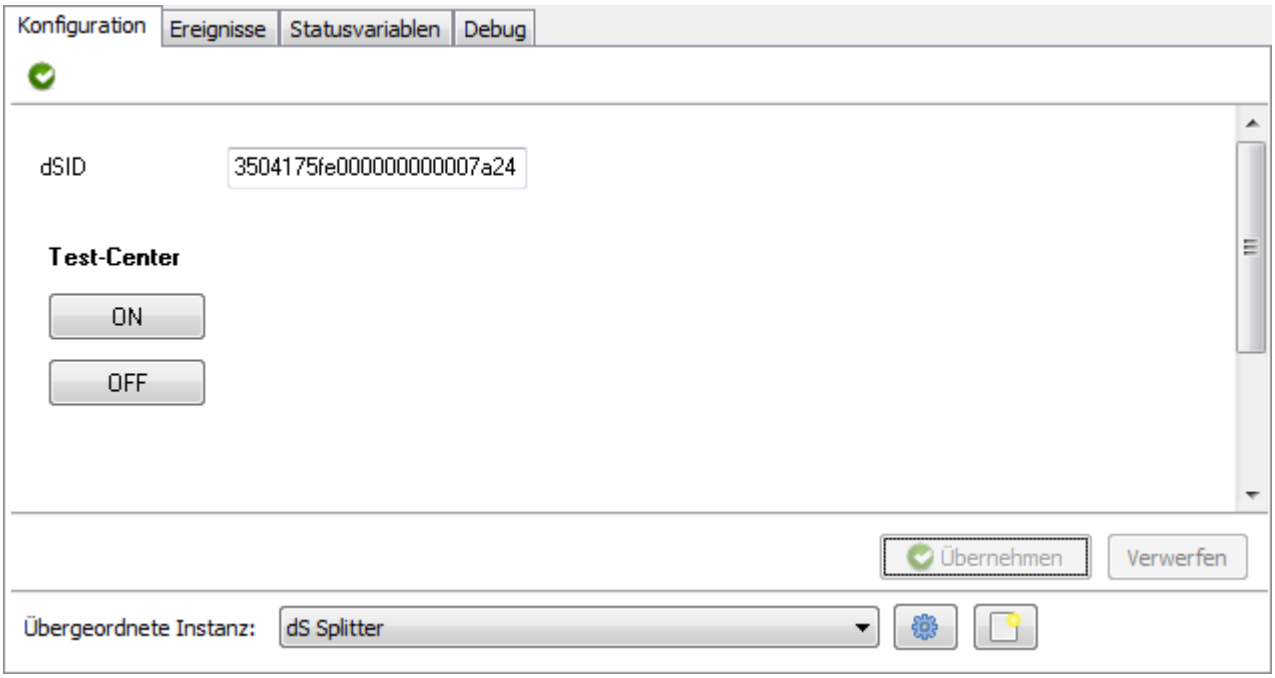



Es öffnet sich die Konfiguration des „dS Splitters“. Bei "IP-Adresse" muss die IP-Adresse des digitalSTROM-Servers eingetragen werden. Mit "Übernehmen" wird die Einstellung gespeichert. Nach einem Klick auf “Request Token” muss Name (Username) und Passwort (Password) der Konfigurations-Webpage eingetragen werden. Bei korrekter Eingabe erscheinen neben Token nun viele Zahlen und kleine Buchstaben. Die "dS Splitter" Konfiguration kann nun geschlossen werden. Im digitalSTROM Konfigurator kann jetzt die Suche via “Suchen” gestartet werden. Anschließend erscheinen alle an den digitalSTROM-Server per digitalSTROM-Klemmen angeschlossenen Geräte.




Bei der ID auf der linken Seite handelt es sich um die Identifikationsnummer, die auf der Rückseite der jeweiligen digitalSTROM-Klemme wiedergefunden werden kann. Im Falle einer großen Anzahl von installierten Geräten ist damit eine problemlose Zuordnung und Konfiguration gewährleistet. Bei der ID auf der rechten Seite handelt es sich um die von IP-Symcon zugeordnete InstanzID. Diese ist einmalig und unveränderbar.
Jedes Gerät kann einzeln als Instanz erstellt werden. Dazu muss dieses ausgewählt

werden und auf den Button "Erstellen" geklickt werden. Die erstellten Geräte sind nun bedienbar.
Nach dem Ausgewählen eines digitalSTROM-Geräts und einem Klick auf “Konfigurieren”, kann die Funktion jedes einzelnen Gerätes getestet werden.




Die Konfigurationsseite kann auch über ein "Doppelklick" innerhalb des Objektbaums aufgerufen werden. (Diese Geräte sind im Objektbaum unter Splitter-Instanzen wiederzufinden.)

# DS_CallScene

```
boolean DS_CallScene (int $InstanzID, int $Szene)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Szene | Szenen von 0 bis 127 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schaltet das Gerät mit der ID **InstanzID** auf Wert **Szene**.
Weitere Informationen zu den Serverszenen

## BEISPIEL

```
DS_CallScene(12345, 42);
```

# DS_DimSet

```
boolean DS_DimSet (int $InstanzID, int $Intensität)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Intensität | Wert von 0-255 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Dimmt das Gerät mit der ID **InstanzID** auf Wert **Intensität**

## BEISPIEL

```
DS_DimSet(12345, 128); //Auf 50% dimmen
```

# DS_MakeRequest

```
string DS_MakeRequest (int $InstanzID, string $Request, string
$Data)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Request | Anfrage welche aufgerufen werden soll |
| Data | Parameter für Anfrage |

## RÜCKGABEWERT

JSON kodierter String

## BESCHREIBUNG

Sendet an das Gerät mit der ID **InstanzID** die Anfrage **Request** mit dem Parameter **Data**. Weitere Informationen zu den möglichen Anfragen und den jeweilig benötigten Parametern, kann unter dSS Json gefunden werden.

## BEISPIEL

```
// Anfrage nach Systemversion
// Sendet den Request "system/version" an die Instanz mit der ID
12345
DS_MakeRequest(12345, "system/version", "");

// Ruft Szene 0 auf
// Sendet den Request "apartment/callScene" mit den Daten
"sceneNumber=0" an die Instanz mit der ID 12345
DS_MakeRequest(12345, "apartment/callScene", "sceneNumber=0");
```

# DS_RequestStatus

Benötigt: IP-Symcon >= 4.0

```
boolean DS_RequestStatus (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Fragt den Status für das Gerät mit der ID **InstanzID** ab.



Dieser Befehl fragt die Klemme über den dS-Bus ab, was einiges an Zeit in Anspruch nehmen kann. In dieser Zeit ist der dS-Bus belegt und kann keine weiteren Aktionen durchführen. Sollte der Befehl für viele Klemmen und oft aufgerufen werden, so kann dies zu spürbaren Verzögerungen im gesamten digitalStrom System führen.

## BEISPIEL

```
DS_RequestStatus(12345);
```

# DS_ShutterMove

```
boolean DS_ShutterMove (int $InstanzID, int $Position)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Position | Position von 0-255 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Bewegt den Rollladen/Markise/Jalousie mit der ID **InstanzID** auf die Position **Position**, bis DS_ShutterStop ausgeführt wird oder die Zielposition erreicht wird.

## BEISPIEL

```
DS_ShutterMove(12345, 128); //Auf 50% fahren
```

# DS_ShutterMoveDown

```
boolean DS_ShutterMoveDown (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Bewegt den Rollladen/Markise/Jalousie mit der ID **InstanzID** nach unten, bis DS_ShutterStop ausgeführt wird oder die Endposition erreicht wird.

## BEISPIEL

```
DS_ShutterMoveDown(12345); //Nach unten bewegen
```

# DS_ShutterMoveUp

```
boolean DS_ShutterMoveUp (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Bewegt den Rollladen/Markise/Jalousie mit der ID **InstanzID** nach oben, bis DS_ShutterStop ausgeführt wird oder die Endposition erreicht wird.

## BEISPIEL

```
DS_ShutterMoveUp(12345); //Nach oben bewegen
```

# DS_ShutterStop

```
boolean DS_ShutterStop (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Stoppt eine aktuelle Bewegung des/der Rollladens/Markise/Jalousie.

## BEISPIEL

```
DS_ShutterStop(12345); //Anhalten
```

# DS_SwitchMode

```
boolean DS_SwitchMode (int $InstanzID, boolean $Status)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Status | TRUE für An, FALSE für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Schaltet das Gerät mit der ID InstanzID auf Wert Status.

## BEISPIEL

```
DS_SwitchMode(12345, true); //Gerät einschalten
```

# Geräteliste

## UNTERSTÜTZTE GATEWAYS

| GATEWAY | BESCHREIBUNG |
|---|---|
| dSS11 | digitalSTROM Server |
| dSS20 | digitalSTROM Server |

## UNTERSTÜTZTE KOMPONENTEN

### LICHT (GELB)

| HERSTELLERBEZEICHNUNG | ANSCHLUSSART | BESCHREIBUNG |
|---|---|---|
| GE-KL200 | Relaisklemme | Schalten 1400 W |
| GE-KM200 | Lüsterklemme | Schalten/Dimmen 150 W |
| GE-KM220 | Lüsterklemme | Sparen 150 W |
| GE-SDM200 | Schnurdimmer | Dimmen 150 W |
| GE-SDS200 | Schnurdimmer | Dimmen 150 W |
| GE-SDS210 | Schnurdimmer | Sparen 150 W |
| GE-TKM210 | Tasterklemme | mit Ausgang Bedienen + Dimmen 150 W |
| GE-TKM220 | Tasterklemme | Bedienen 1-fach |
| GE-TKM230 | Tasterklemme | Bedienen 2-fach |
| GE-TUP200 | Taster | Bedienen 1-fach |
| GE-TUP210 | Taster | Bedienen 2-fach |
| GE-TUP220 | Taster+ | Bedienen 1-fach Plus |
| GE-UMV200 | Schraubklemmen | 1-10V Universal Modul |
| GE-ZWS210 | Zwischenstecker | Schalten 2300 W |
| GE-ZWS200 | Zwischenstecker | Schalten 2300 W |

### SCHATTEN (GRAU)

| HERSTELLERBEZEICHNUNG | ANSCHLUSSART | BESCHREIBUNG |
|---|---|---|
| GR-HKL230 | Relaisklemme | Jalousie Schaltaktor |
| GR-KL200 | Relaisklemme | Rollladen 700 VA |
| GR-KL210 | Relaisklemme | Markise 700 VA |
| GR-KL220 | Relaisklemme | Jalousie 700 VA |
| GR-TKM200 | Tasterklemme | Bedienen 1-fach |
| GR-TKM210 | Tasterklemme | Bedienen 2-fach |
| GR-TUP200 | Taster | Bedienen 1-fach |
| GR-TUP210 | Taster | Bedienen 2-fach |
| GR-TUP220 | Taster | Bedienen 1-fach Plus |

## HEIZUNG/KLIMA (BLAU)

| HERSTELLERBEZEICHNUNG | ANSCHLUSSART | BESCHREIBUNG |
|---|---|---|
| BL-KL200 | Relaisklemme | Schalten 700 W |
| BL-KM200 | Lüsterklemme | Schalten 150 W |
| BL-SDS2001 | Sensor mit Ausgang | Schalten 150 W + Sensor |
| BL-TUP2001 | Sensor | Klima-Sensor |

## JOKER (SCHWARZ)

| HERSTELLERBEZEICHNUNG | ANSCHLUSSART | BESCHREIBUNG |
|---|---|---|
| dS-iSens200 | Lüsterklemme | Raumfühler für Temperatur und Luftfeuchte |
| dS-Alarm400 | Tasterklemme | 4 Sensoren anschließbar |
| SW-AKM200 | Tasterklemme | 4 Sensoren anschließbar |
| SW-AKM210 | Tasterklemme | 2 Sensoren anschließbar |
| SW-SSL200 | Schnurdimmer | 1-fach Schalter |
| SW-TKM200 | Tasterklemme | Bedienen 4-fach universell |
| SW-TKM210 | Tasterklemme | Bedienen 2-fach universell |
| SW-TUP200 | Taster | Bedienen 1-fach universell |
| SW-UMR200 | Schraubklemmen | Relais Universal Modul |
| SW-ZWS200 | Zwischenstecker | Schalten 2300 W |
| SW-ZWS210 | Zwischenstecker | Schalten 2300 W |

## ZUGANG (GRÜN)

Keine gesonderte Steuerung nötig.

### SICHERHEIT (ROT)

Keine gesonderte Steuerung nötig.

### AUDIO (HELLBLAU)

Keine gesonderte Steuerung nötig.

# DMX / ArtNet

DMX bzw. DMX512 ist ein Protokoll zur Steuerung von Lichttechnikkomponenten. Die Anbindung an IP-Symcon erfolgt beispielsweise über ein DMX USB-Interface(seriell) oder DMX LAN-Interface. Über das LAN-Interface wird auch das IP Protokoll ArtNet unterstützt.

Folgende Geräte werden von IP-Symcon unterstützt:
Unterstützte Komponenten

## BESCHREIBUNG

DMX (Lichttechnik) oder auch DMX512 genannt ist ein digitales Steuerprotokoll, das in der Bühnen- und Veranstaltungstechnik Verwendung findet. Es dient der Steuerung von Dimmern, Scheinwerfern und Effektgeräten auf Bühnen oder in Diskotheken. Die Abkürzung DMX steht für Digital Multiplex, es ist ein Standard, der durch die DIN 56930-2 genormt wurde. Die Anbindung an IP-Symcon erfolgt u.a. über ein DMX USB-Interface/ DMX LAN-Interface. Seit IP-Symcon 3.0 wird zusätzlich ArtNet unterstützt. Es ist ein IP-basiertes Protokoll zur Übertragung großer Mengen von DMX-Informationen über LAN. Die Spezifikation wurde von der Firma "Artistic Licence" erstellt und ist frei verfügbar. IP-Symcon unterstützt alle ArtNet-kompatiblen DMX Interfaces.

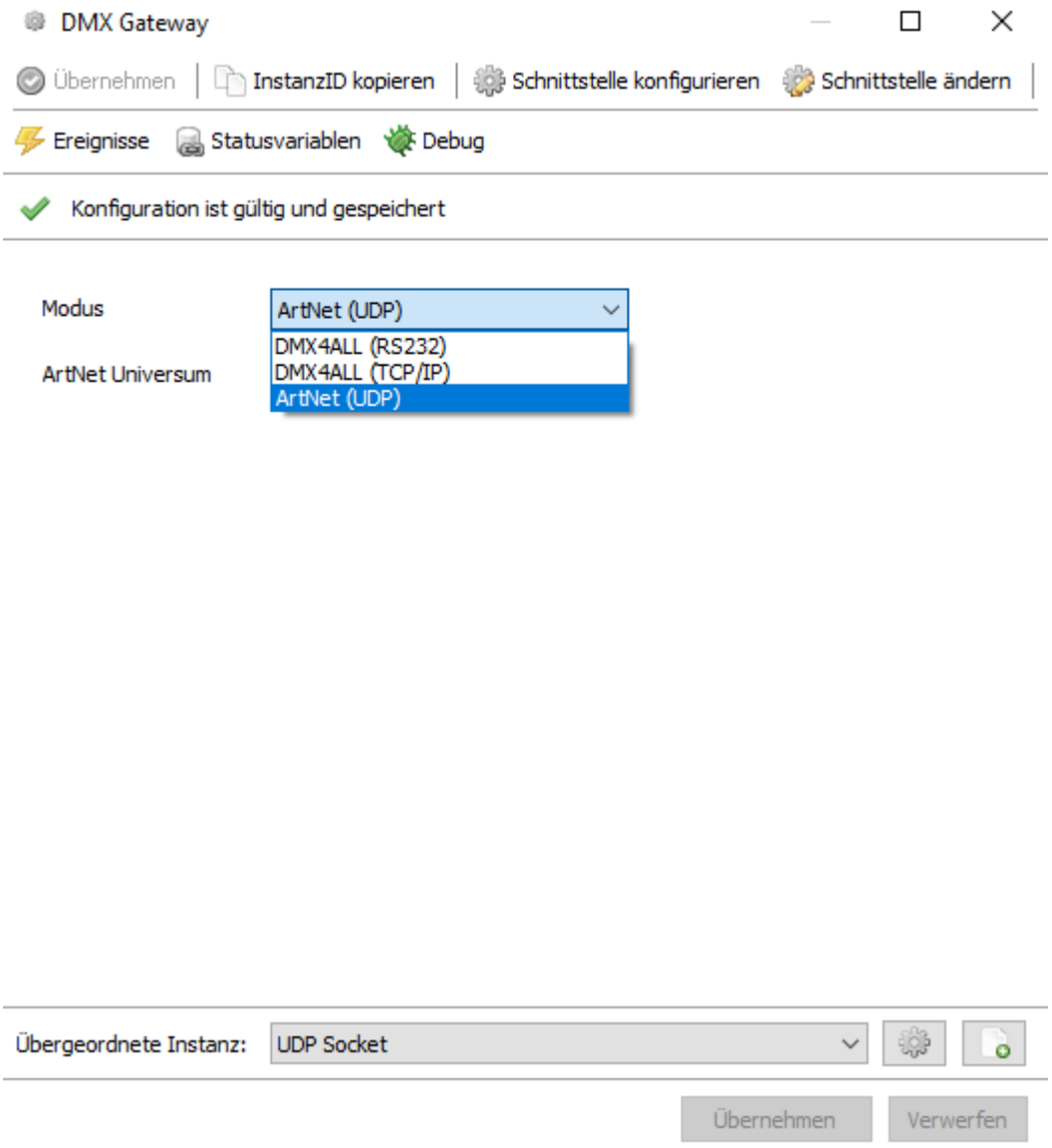
## VIDEO-TUTORIAL ZUR EINRICHTUNG

0:00 / 0:00

## EINBINDUNG IN IP-SYMCON

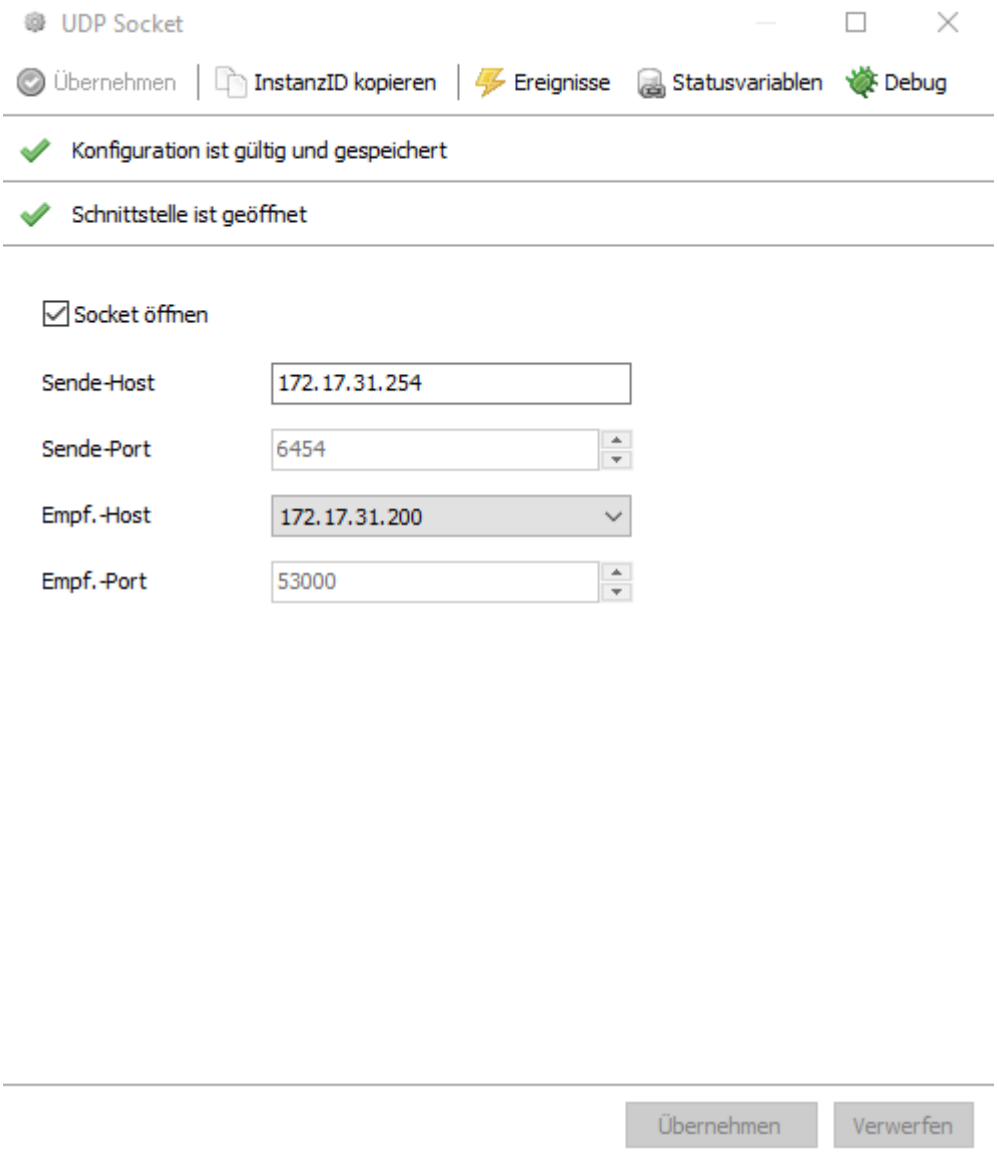
Voraussetzung für den Betrieb in Verbindung mit IP-Symcon ist eine ordnungsgemäß installierte Hardware sowie Kenntnis der verwendeten COM-Port-Schnittstelle bzw. der

IP-Adresse.
Es muss in IP-Symcon eine DMX-Instanz hinzugefügt und in der übergeordneten Instanz der Modus DMX oder ArtNet für die angeschlossene Hardware ausgewählt werden. Bei ArtNet ist das "Universe" in der Regel 0 und bei DMX ist der Eintrag irrelevant.

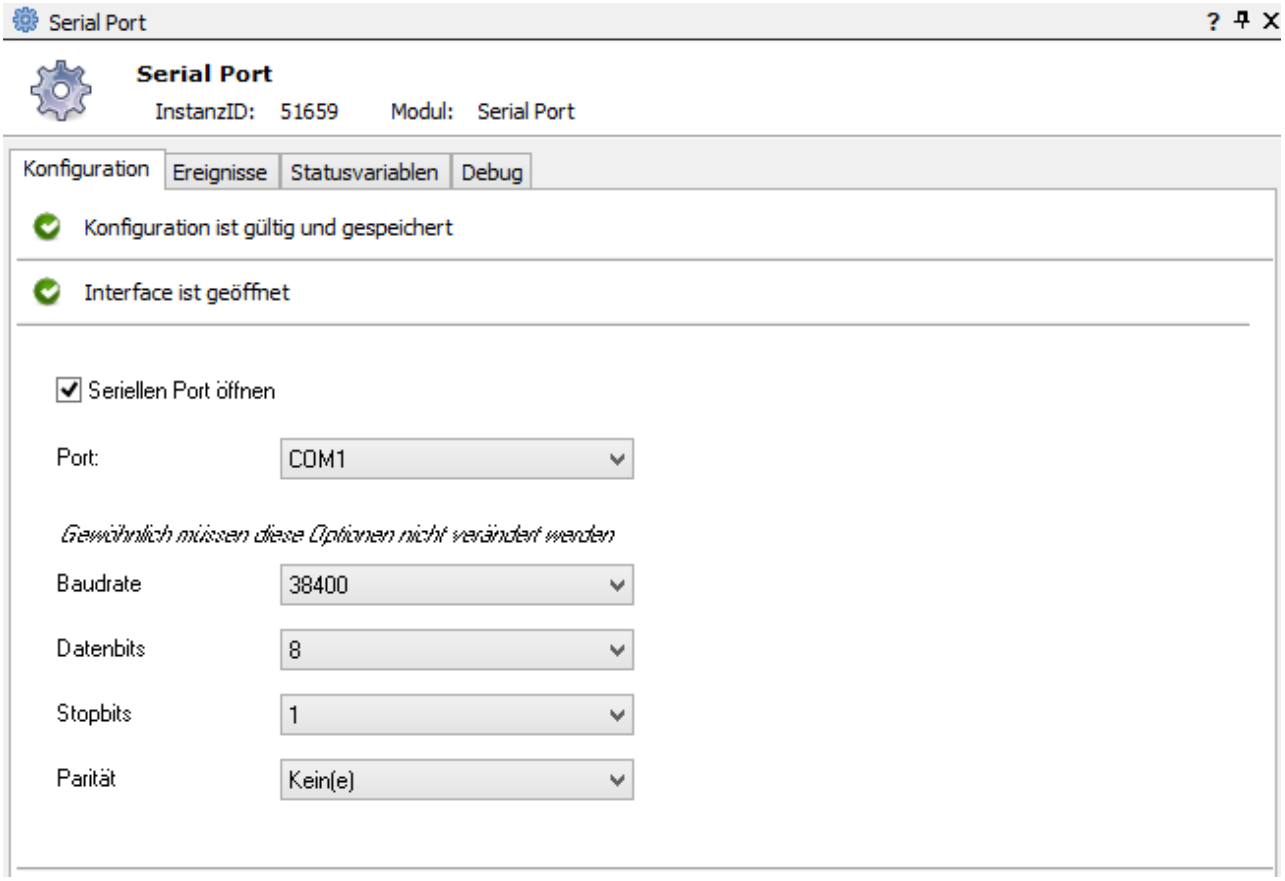



Bei der Nutzung von ArtNet sind in der übergeordneten Instanz folgende Einstellungen vorzunehmen:
'Sende-Host' ist die IP-Adresse des ArtNet-Interface und der 'Sende-Port': 6454 – der Standard-Port des ArtNet-Protokolls.
'Empf.-Host' ist die IP-Adresse der Netzwerkkarte, an die das Interface angeschlossen ist. In der Regel ist es die Adresse vom PC, auf dem der IP-Symcon Server läuft.

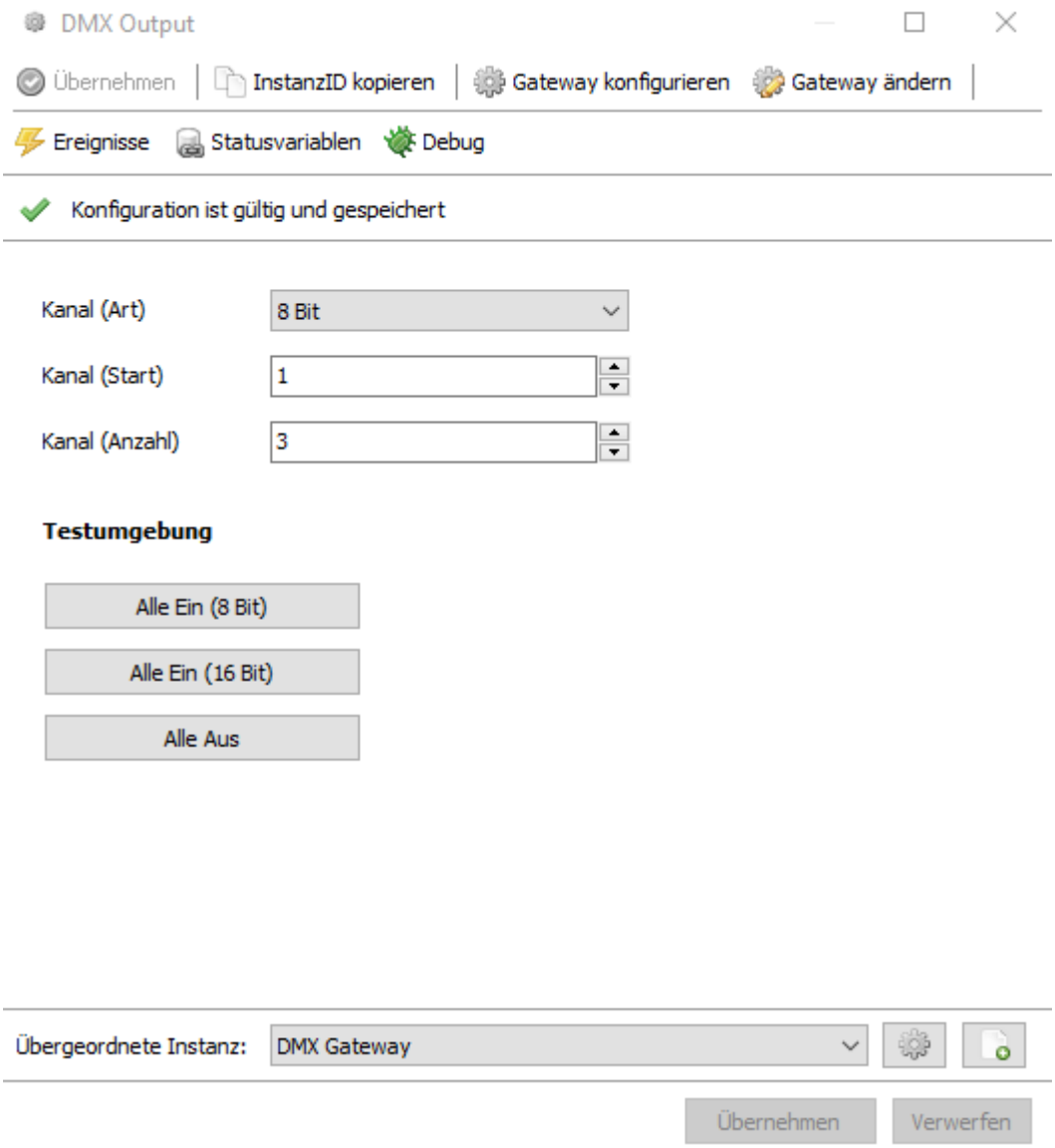
Wenn mehrere ArtNet Interfaces erstellt werden, muss für jede UDP Socket Instanz der Empfangsport (Default: 53000) um 1 erhöht werden. (D.h. das zweite Interface hat z.B. Port 53001)




Bei einem DMX-Interface, das über USB angeschlossen ist, wird im Windows-Gerätemanager ein virtueller COM-Port installiert. Dieser ist hier einzutragen. Die weiteren Einstellungen der Baudrate: 38400,8,1,N dürfen nicht verändert werden. Sind alle Einstellungen gemacht, muss noch der Port bzw. Socket geöffnet und "Übernehmen" gedrückt werden.




In der Instanz 'DMX-Output“ kann die Funktionalität in der Testumgebung überprüft werden.




Mit einer Instanz können mehrere Kanäle des DMX-Geräts angesprochen werden. Das ist nützlich um z.B. komplette DMX-RGB-Scheinwerfer als zusammenhängendes Gerät in der Visualisierung anzeigen zu können. Über die Buttons "Alle Ein" und "Alle Aus" können alle der Instanz zugewiesenen Kanäle ein bzw. aus geschaltet werden.

Bei einer DMX RGB Instanz mit Kanal(Art) = 16bit ist zu beachten, dass nicht nur der eingestellte Kanal genutzt wird, sondern auch der nächst höhere. DMX Kanäle sind 8 bit. Um 16 bit anzusteuern, werden 2 Kanäle genutzt. Beispiel: Kanal(Rot) = 40 bedeutet, dass Kanal 40 und 41 genutzt werden.

# DMX_FadeChannel

```
boolean DMX_FadeChannel (int $InstanzID, int $Kanal, int $Wert,
float $Überblendzeit)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Kanal | 0 = alle Instanz-Kanäle, 1 – 512 je nach Konfiguration |
| Wert | 0-255 / 0-65535 (8/16 bit) |
| Überblendzeit | Zeit in Sekunden |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Dimmt den **Kanal** des Geräts mit der ID **InstanzID** auf **Wert** mit einer bestimmten **Überblendzeit**

## BEISPIEL

```
//Alle Kanäle der Instanz innerhalb von 2,5sek auf 255 dimmen
DMX_FadeChannel(12345, 0, 255, 2.5);
```

# DMX_FadeChannelDelayed

```
boolean DMX_FadeChannelDelayed (int $InstanzID, int $Kanal, int
$Wert, float $Überblendzeit, float $Verzögerungszeit)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Kanal | 0 = alle Instanz-Kanäle, 1 – 512 je nach Konfiguration |
| Wert | 0-255 / 0-65535 (8/16 bit) |
| Überblendzeit | Zeit in Sekunden |
| Verzögerungszeit | Zeit in Sekunden |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Dimmt den Kanal des Geräts mit der ID InstanzID auf Wert mit einer bestimmten Überblendzeit. Es kann eine Verzögerungszeit angegeben werden, nach der der Dimmvorgang gestartet wird.



Wenn keine Verzögerungszeit angegeben wird, bricht dieser Befehl alle für die betroffenen Kanäle verzögertern über DMX_FadeChannelDelayed gesetzten Fadings ab. Damit kann eine neue Kette an verzögerten Fadings sauber gesetzt werden. Wenn keine verzögerten Fadings genutzt werden, ist dieser Hinweis nicht relevant

## BEISPIEL

```
//Dimmt ein Modul komplett hoch und wieder runter.
//Alle Kanäle der Instanz innerhalb von 5sek auf 255 dimmen
DMX_FadeChannel(12345, 0, 255, 5.0);
//Alle Kanäle der Instanz innerhalb von 2,5sek auf 0 dimmen, nach
einer Verzögerung von 5sek.
DMX_FadeChannelDelayed(12345, 0, 0, 2.5, 5);
```

# DMX_FadeRGB

```
boolean DMX_FadeRGB (int $InstanzID, int $R, int $G, int $B, float
$Überblendzeit)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| R | Helligkeitswert 0 (aus) bis 255/65535 (8/16 bit) (an, maximale Helligkeit) |
| G | Helligkeitswert 0 (aus) bis 255/65535 (8/16 bit) (an, maximale Helligkeit) |
| B | Helligkeitswert 0 (aus) bis 255/65535 (8/16 bit) (an, maximale Helligkeit) |
| Überblendzeit | Zeit in Sekunden |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Dimmt den Rot-, Grün- und Blaukanal der ID **InstanzID** mit einer bestimmten **Überblendzeit**.

## BEISPIEL

```
//Farbe auf Gelb in 5 Sekunden überblenden
//8bit Instanz
DMX_FadeRGB(12345, 255, 255, 0, 5.0);

//16bit Instanz
DMX_FadeRGB(12345, 65535, 65535, 0, 5.0);
```

# DMX_FadeRGBDelayed

```
boolean DMX_FadeRGBDelayed (int $InstanzID, int $R, int $G, int $B,
float $Überblendzeit, float $Verzögerungszeit)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| R | Helligkeitswert 0 (aus) bis 255/65535 (8/16 bit) (an, maximale Helligkeit) |
| G | Helligkeitswert 0 (aus) bis 255/65535 (8/16 bit) (an, maximale Helligkeit) |
| B | Helligkeitswert 0 (aus) bis 255/65535 (8/16 bit) (an, maximale Helligkeit) |
| Überblendzeit | Zeit in Sekunden |
| Verzögerungszeit | Zeit in Sekunden |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Dimmt den Rot-, Grün- und Blaukanal der ID **InstanzID** mit einer bestimmten **Überblendzeit**. Es kann eine Verzögerungszeit angegeben werden, nach der der Dimmvorgang gestartet wird.



Wenn keine Verzögerungszeit angegeben wird, bricht dieser Befehl alle für die betroffenen Kanäle verzögertern über DMX_FadeRGBDelayed gesetzten Fadings ab. Damit kann eine neue Kette an verzögerten Fadings sauber gesetzt werden. Wenn keine verzögerten Fadings genutzt werden, ist dieser Hinweis nicht relevant

## BEISPIEL

```
//Dimmt ein Modul komplett auf Gelb hoch und wieder runter.

//8bit
```

```
//Instanz innerhalb von 5sek auf Gelb dimmen
DMX_FadeRGB(12345, 255, 255, 0, 5.0);
//Instanz innerhalb von 2,5sek auf Gelb dimmen, nach einer
Verzögerung von 5sek
DMX_FadeRGBDelayed(12345, 255, 255, 0, 2.5, 5);

//16bit
//Instanz innerhalb von 5sek auf Gelb dimmen
DMX_FadeRGB(12345, 65535, 65535, 0, 5.0);
//Instanz innerhalb von 2,5sek auf Gelb dimmen, nach einer
Verzögerung von 5sek
DMX_FadeRGBDelayed(12345, 65535, 65535, 0, 2.5, 5);
```

# DMX_RequestInfo

```
boolean DMX_RequestInfo (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der DMX Splitter Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Fragt die Informationen der DMX Instanz mit der ID **InstanzID** ab.

## BEISPIEL

```
// Informationen der Instanz 12345 abfragen.
DMX_RequestInfo(12345);
```

# DMX_ResetInterface

```
boolean DMX_ResetInterface (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der DMX Splitter Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Setzt das DMX Splitter Interface mit der ID **InstanzID** zurück. Es empfiehlt sich, bei Problemen, ein Skript mit diesem Befehl zu erstellen und als Start-Skript zu deklarieren.

## BEISPIEL

```
//Splitter Instanz mit der ID 12345 zurücksetzen
DMX_ResetInterface(12345);
```

# DMX_SetBlackOut

```
boolean DMX_SetBlackOut (int $InstanzID, boolean $Status)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der DMX Splitter Instanz |
| Status | TRUE für alter Wert, FALSE für 0 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Setzt alle Kanäle des DMX Interfaces auf Wert **Status**. Die ID **InstanzID** ist dabei die InstanzID der Splitter-Instanz.



Dieser Befehl ist für ArtNet Geräte nicht verfügbar.

Dieser Befehl beeinflusst keine Status Variablen der Module.

## BEISPIEL

```
//Alle Kanäle deaktiveren
DMX_SetBlackOut(12345, true);
//Alle Kanäle reaktivieren
DMX_SetBlackOut(12345, false);
```

# DMX_SetChannel

```
boolean DMX_SetChannel (int $InstanzID, int $Kanal, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Kanal | 0 = alle Instanz-Kanäle, 1 – 512 je nach Konfiguration |
| Wert | 0-255 / 0-65535 (8/16 bit) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schaltet den Kanal vom Gerät mit der ID **InstanzID** auf Wert **Wert**



Dieser Befehl bricht alle für die betroffenen Kanäle über DMX_FadeChannelDelayed gesetzten Fadings ab.

## BEISPIEL

```
//Den ersten Kanal der Instanz auf 255 schalten
DMX_SetChannel(12345, 1, 255);
```

# DMX_SetRGB

```
boolean DMX_SetRGB (int $InstanzID, int $R, int $G, int $B)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| R | Helligkeitswert 0 (aus) bis 255/65535 (8/16 bit) (an, maximale Helligkeit) |
| G | Helligkeitswert 0 (aus) bis 255/65535 (8/16 bit) (an, maximale Helligkeit) |
| B | Helligkeitswert 0 (aus) bis 255/65535 (8/16 bit) (an, maximale Helligkeit) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet den Rot-, Grün- und Blaukanal der ID **InstanzID**.

## BEISPIEL

```
//Farbe auf Gelb setzen

//8bit Instanz
DMX_SetRGB(12345, 255, 255, 0);

//16bit Instanz
DMX_SetRGB(12345, 65535, 65535, 0);
```

# Geräteliste

## UNTERSTÜTZTE INTERFACES

DMX4ALL Mini-USB-DMX-Interface
DMX4ALL NanoDMX USB Interface
DMX4ALL USB-DMX STAGE-PROFI MK2
DMX4ALL USB-DMX STAGE-PROFI MK3
LAN-DMX STAGE-PROFI
AVR Art Net Node
Art Net Box (Quad Art Net Box)
Es werden alle ArtNet Interfaces unterstützt

## UNTERSTÜTZTE GERÄTE

### BIS VERSION 4.0

Alle Geräte, welche die oben aufgeführten Interfaces und maximal eine 8bit (256 Steps) Auflösung nutzen.

### AB VERSION 4.1

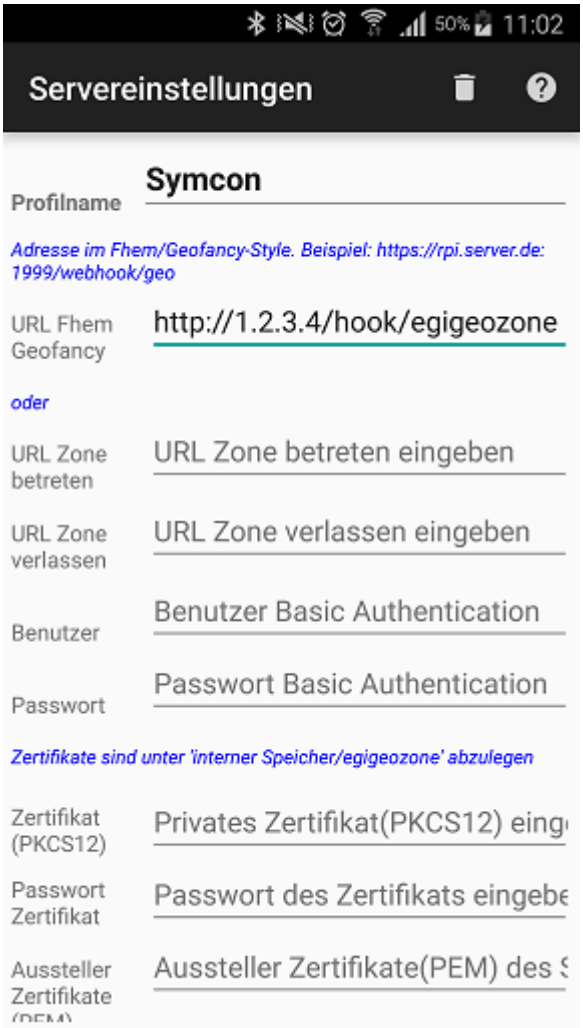
Alle Geräte, welche die oben aufgeführten Interfaces nutzen.
Es wird auch die 16bit (65535 Steps) Auflösung unterstützt.

# EgiGeoZone

Benötigt: IP-Symcon >= 4.2
Das Modul dient zum empfangen von EgiGeoZone Daten.

## FUNKTIONSUMFANG

Pro Gerät eine eigene Standortliste
Benutzername und Passwort Identifikation innerhalb von IP-Symcon.
Richtet automatisch den Webhook "/hook/egigeozone" ein.
Es wird empfohlen dies in Kombination mit dem Connect Modul zu nutzen.




## VORAUSSETZUNGEN

EgiGeoZone App für Google Android

## SOFTWARE-INSTALLATION

Über den Module Store das Modul EgiGeoZone installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'EgiGeoZone'-Modul mithilfe der Schnellsuche gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Benutzername | Benutzername, welcher in der EgiGeoZone App angegeben werden muss, um IP-Symcon Daten schicken zu können. |
| Passwort | Passwort, welches in der EgiGeoZone App angeben werden muss. |

*Werden diese Daten Leer gelassen kann jeder über den Hook Daten an IP-Symcon schicken.*

## STATUSVARIABLEN

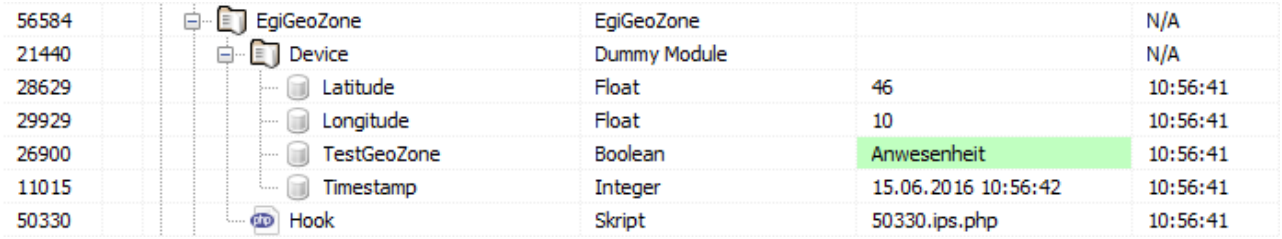
Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

Die Variablen werden anhand der Geräte ID und beim erstmaligen senden innerhalb des EgiGeoZone Moduls automatisch angelegt. Es können mehrere Geräte über einen Hook laufen. Jedes Gerät wird unter seiner eigenen "Kategorie" eingerichtet.

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Gerätename | Instanz (Dummy) | Dient als "Kategorie" in der sich alle überwachten Standorte, sowie der Zeitstempel und Längen-/ Breitengrad befinden. Wird pro Gerät erstellt. |
| Latitude | Float | Breitengrad der letzten Aktivität. |
| Longitude | Float | Längengrad der letzten Aktivität. |
| Timestamp | Integer | UnixTimestamp der letzten Aktivität. |
| Beispielstandort (GeoZoneTest) | Boolean | Present oder Absent. Information wird von Gefency geliefert. |

Beispiel:



## VISUALISIERUNG

Es gibt keine native Darstellung via Visualisierung oder in den mobilen Apps. Geräte und Variablen, welche angezeigt werden sollen, können via Link angezeigt werden.

# ekey

Die Firma ekey stellt verschiedene Fingerscanner her und diese können mithilfe eines Gateways in IP-Symcon integriert werden. IP-Symcon empfängt die Aktivitäten des Fingerscanners und stellt diese via Variable zur Verfügung.

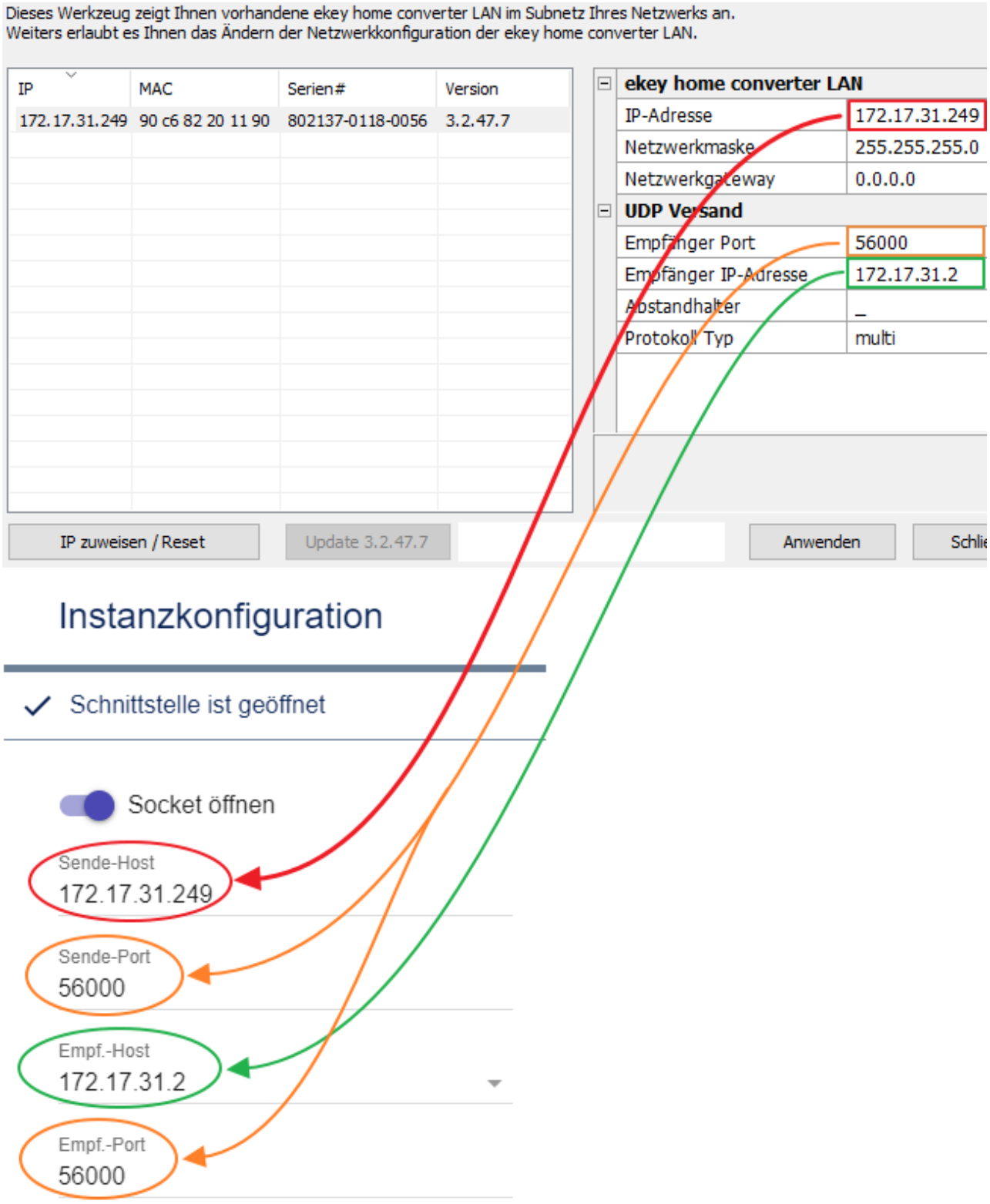
Folgende Geräte werden von IP-Symcon unterstützt:
Unterstützte Geräte
Für die Integration von ekey bionyx bitte auf folgende Seite wechseln ekey bionyx

## INSTALLATION

Um IP-Symcon den Zugriff auf das Home/Multi/Net System zu gewähren muss das passende CV LAN Modul (Gateway) ins Netzwerk integriert werden.
Die IP-Adresse für den UDP Socket kann über das eKey home CV LAN Tool gesetzt werden.

Das CV LAN Tool gibt es in verschiedenen Versionen für Home , Multi und Net.

Nach der Einrichtung des CV LAN müssen die angegebenen IP-Adressen noch in IP-Symcon konfiguriert werden.
In der eKey-Instanz muss das Protokoll, sowie Trennzeichen eingestellt werden.
Über "Gateway Konfigurieren" muss der Sende- und Empfangs-Host, sowie Port eingetragen werden.
Als Empfänger IP muss die IP vom IP-Symcon Server eingetragen werden.


Dieses Werkzeug zeigt Ihnen vorhandene ekey home converter LAN im Subnetz Ihres Netzwerks an.
Weiters erlaubt es Ihnen das Ändern der Netzwerkkonfiguration der ekey home converter LAN.
IP
MAC
Serien#
Version
172.17.31.249
90 c6 82 20 11 90
802137-0118-0056
3.2.47.7
ekey home converter LAN
IP-Adresse
172.17.31.249
Netzwerkmaske
255.255.255.0
Netzwerkgateway
0.0.0.0
UDP Versand
Empfänger Port
56000
Empfänger IP-Adresse
172.17.31.2
Abstandhalter
_
Protokoll Typ
multi
IP zuweisen / Reset
Update 3.2.47.7
Anwenden
Instanzkonfiguration
Schnittstelle ist geöffnet
Socket öffnen
Sende-Host
172.17.31.249
Sende-Port
56000
Empf.-Host
172.17.31.2
Empf.-Port
56000

# Geräteliste

## UNTERSTÜTZTE GATEWAYS

| GATEWAY | BESCHREIBUNG |
| --- | --- |
| ekey Home Converter LAN RS-485 | Home/Multi System |
| ekey Net CV LAN | Net System |
| ekey bionyx | Cloud System. |

## UNTERSTÜTZTE KOMPONENTEN

IP-Symcon unterstützt alle an ekey Home/Multi/Net anschließbaren Geräte.

# ekey bionyx

Benötigt: IP-Symcon >= 7.0
Die Firma ekey stellt verschiedene Fingerscanner her und diese können mithilfe eines Gateways in IP-Symcon integriert werden. IP-Symcon empfängt die Aktivitäten des Fingerscanners über Webhooks.

## FUNKTIONSUMFANG

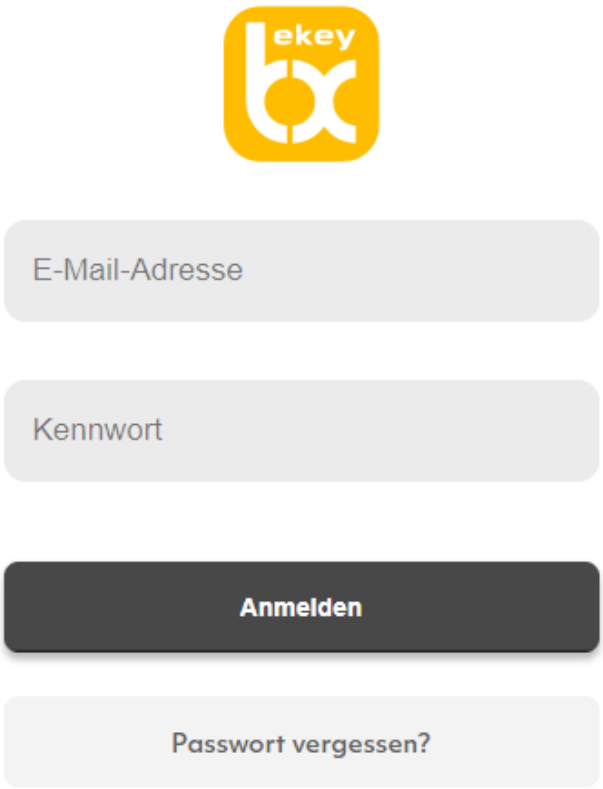
Einrichten eines oder mehrerer ekey bionyx Systeme
Hinterlegen von Aktionen zu Webhooks

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

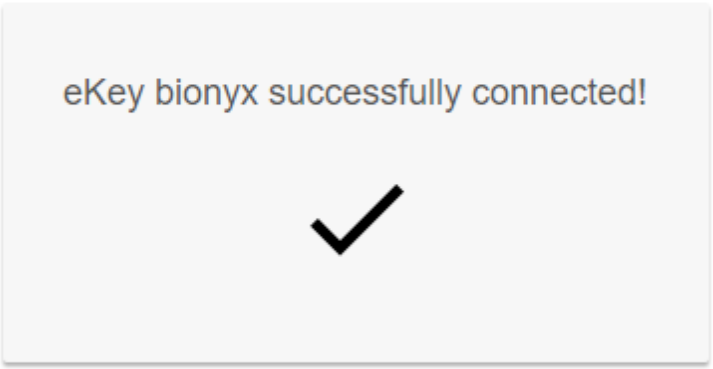
Unter 'Instanz hinzufügen' kann das 'ekey bionyx Konfigurator'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen
Beim Erstellen der Instanz kann die Schnittstelle konfiguriert werden.
Konfigurationsseite der Cloud:

| NAME | BESCHREIBUNG |
|---|---|
| Verbindung | **Lokal**: Ist der ekey Controller wie auch IP-Symcon im selben Netzwerk wird diese Option genutzt.<br>**Connect**: Sind ekey Controller und IP-Symcon nicht im selben Netzwerk, läuft die Kommunikation über den ConnectDienst |
| IP-Adresse | Die IP-Adresse des IP-Symcon Servers |

Danach wird die Instanz erstellt. Im Konfigurator erscheint der Hinweis das "Eine übergeordnete Instanz ist inaktiv". hier wird auf Konfigurieren geklickt. Die ekey Cloud Instanz wird aufgerufen. Damit die Einrichtung der ekey Cloud Instanz abgeschlossen wird, muss auf den Registrieren-Button geklickt werden. Im Browser öffnet sich nun ein neuer Tab mit der Anmeldungsmaske von ekey.




Nach Eingabe der Anmeldedaten wird bei Erfolg folgender Dialog angezeigt.




Nun kann zu IP-Symcon zurückgekehrt werden. In der ekey bionyx Cloud Instanz sollte jetzt ein Token vorhanden sein.
Im ekey bionyx Konfigurator kann nun auf aktualisieren geklickt werden, damit die vorhandenen Systeme, welche mit dem Account verknüpft sind, angezeigt werden. Diese können nun erstellt und konfiguriert werden.
Konfigurationsseite des Konfigurators:

| NAME | BESCHREIBUNG |
|---|---|
| ID | ID des ekey Controllers |
| Name | Name des ekey Account Admin |
| Informationen | Informationen wie viele Webhooks verfügbar und vorhanden sind für dieses System |

Pro Account sind 5 Webhooks erlaubt.
Konfigurationsseite des System:
Funktionen Webhooks:

| NAME | BESCHREIBUNG |
|---|---|
| ID | Id des Webhooks |
| Funktion | Funktionsname, welcher in der ekey bionyx App angezeigt wird |
| Ort | Ortsname, welcher in der ekey bionyx App angezeigt wird |
| Aktion | Aktion, welche ausgeführt wird, wenn der Webhook ausgelöst wird |

# EnOcean

EnOcean ist ein Funksystem, welches von mehreren Herstellern unterstützt wird. Für eine Verbindung mit IP-Symcon wird ein USB-Interface oder ein LAN-Interface benötigt.

Folgende Geräte werden von IP-Symcon unterstützt:
Unterstützte Geräte (inkl. Gateways) und EEP-Liste

## INHALT


Installation
Einbindung in IP-Symcon
Konfiguration
- Aktor
- Aktor (Eltako)
- Aktor (OPUS)
- Aktor (Stellantrieb)
- Sensor

Tipps & Tricks


## INSTALLATION

Das **USB Gateway** wird per USB mit dem Server verbunden und ist über einen Serial Port ansprechbar.
Das **Symcon LAN Gateway** wird per LAN mit dem Server verbunden. Zur Einrichtung wird das IP-Symcon "Network Configuration Tool" benötigt. Dieses steht zum Download bereit. Eine einfache Beschreibung zur Konfiguration des Gateways steht zur Verfügung. Im Anschluss ist das Gateway über die entsprechende IP-Adresse und Port (Standardmäßig: 5000) erreichbar.

Wenn kein IP-Symcon Haus auf der Vorderseite des LAN-Gateways abgedruckt ist, ist dies ein altes Modell des Gateways. Die Installationsanleitung für dieses Modell kann hier gefunden werden.

Das **Eltako FGW14** wird per RS232 mit der SymBox (mit integrierter RS232-Erweiterung) verbunden. Dieses wiederum ist mit dem **Eltako FAM14** über die "Eltako Bus14"-Steckbrücken verbunden. Hierzu ist folgende Anschlusskonfiguration zu beachten:
**FAM14:**
**Betriebsart (BA) Wählschalter:**
Die Betriebsart ist entsprechend der Betriebsanleitung einzustellen. Für bidirektionale Kommunikation muss die Betriebsart 2 gewählt werden.
Für Adressvergabe muss die Betriebsart 1 gewählt werden.
**Anschlussklemmen "N" und "L":**
230V Netspannung. Entweder Schuko-Stecker oder direkt im Schaltschrank mit entsprechenden Leitungen verbinden.
**Anschlussklemme "Hold":**

Anschlussklemme mit Anschlussklemme "Hold" des FGW14 verbinden.
**Anschlussklemme "GND":**
Mit dem Ground-Pin (⊖) der steckbaren Anschlussklemme der RS232-Schnittstelle der Symbox verbinden.

FGW14:

**Betriebsart (BA) Wählschalter:**
Betriebsart 6: "Bus14 ↔ RS232 58k Baud" auswählen.
**Anschlussklemme "Hold":**
Anschlussklemme mit Anschlussklemme "Hold" des FAM14 verbinden.
**Anschlussklemme "Tx":**
Mit Pin ↓ der RS232-Schnittstelle der SymBox verbinden.
**Anschlussklemme "Rx":**
Mit Pin ↑ der RS232-Schnittstelle der SymBox verbinden.

## EINBINDUNG IN IP-SYMCON

Bei Verwendung des LAN-Gateways kann dies über die Gerätesuche eingebunden werden. Hierfür muss als System "EnOcean Discovery" gewählt werden. Die Discovery-Instanz bietet dann die Erstellung eines EnOcean Konfigurators an. Nachdem der Konfigurator erstellt wurde, können dort die einzelnen Geräte wie unten beschrieben eingebunden werden.
Das USB-Gateway wird direkt als Konfigurator angelegt, da es über die Gerätesuche nicht gefunden werden kann. Hierfür wird im Objektbaum ein "EnOcean Configurator" erstellt. Bei der Konfiguration der Schnittstelle muss der passende ESP2 oder ESP3 Modus gewählt und anschließenden der pasende Port ausgewählt und geöffnet werden.

Wie die I/O Instanz (übergeordnete Instanz des Gateways) ausgewählt und ggf. ausgetauscht wird, ist hier nachzulesen: I/O Instanz austauschen

## KONFIGURATION

Abhängig vom Gerätetyp unterscheidet sich die Konfigurationsweise. Es wird zwischen 4 verschiedenen Gerätetypen unterschieden. Jede hat unterschiedliche Vorgehensweisen bei der Konfiguration. (siehe unten im jeweiligen Abschnitt)

Um ein EnOcean-Gerät nutzen zu können, muss dieses explizit in IP-Symcon angelernt werden.

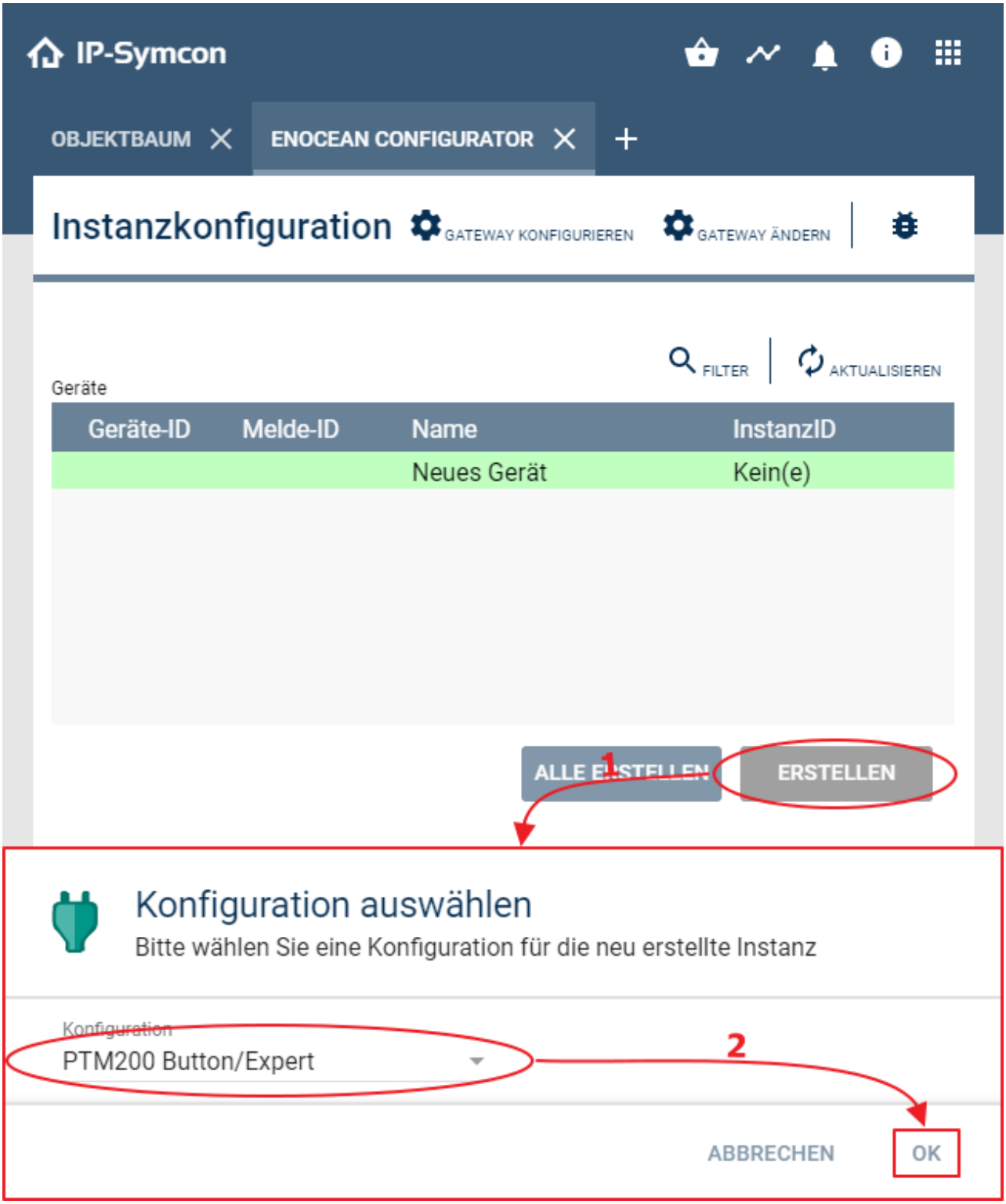
Jede Geräte-ID (0-127) darf nur ein einziges Mal vergeben werden.

Bevor eine Suche gestartet werden kann, muss die übergeordnete Instanz konfiguriert werden. Der korrekte serielle Port muss angeben und die Konfiguration übernommen werden. Die Einstellung von Baudrate etc. ist nicht notwendig, da diese vom System automatisch vorgenommen wird.

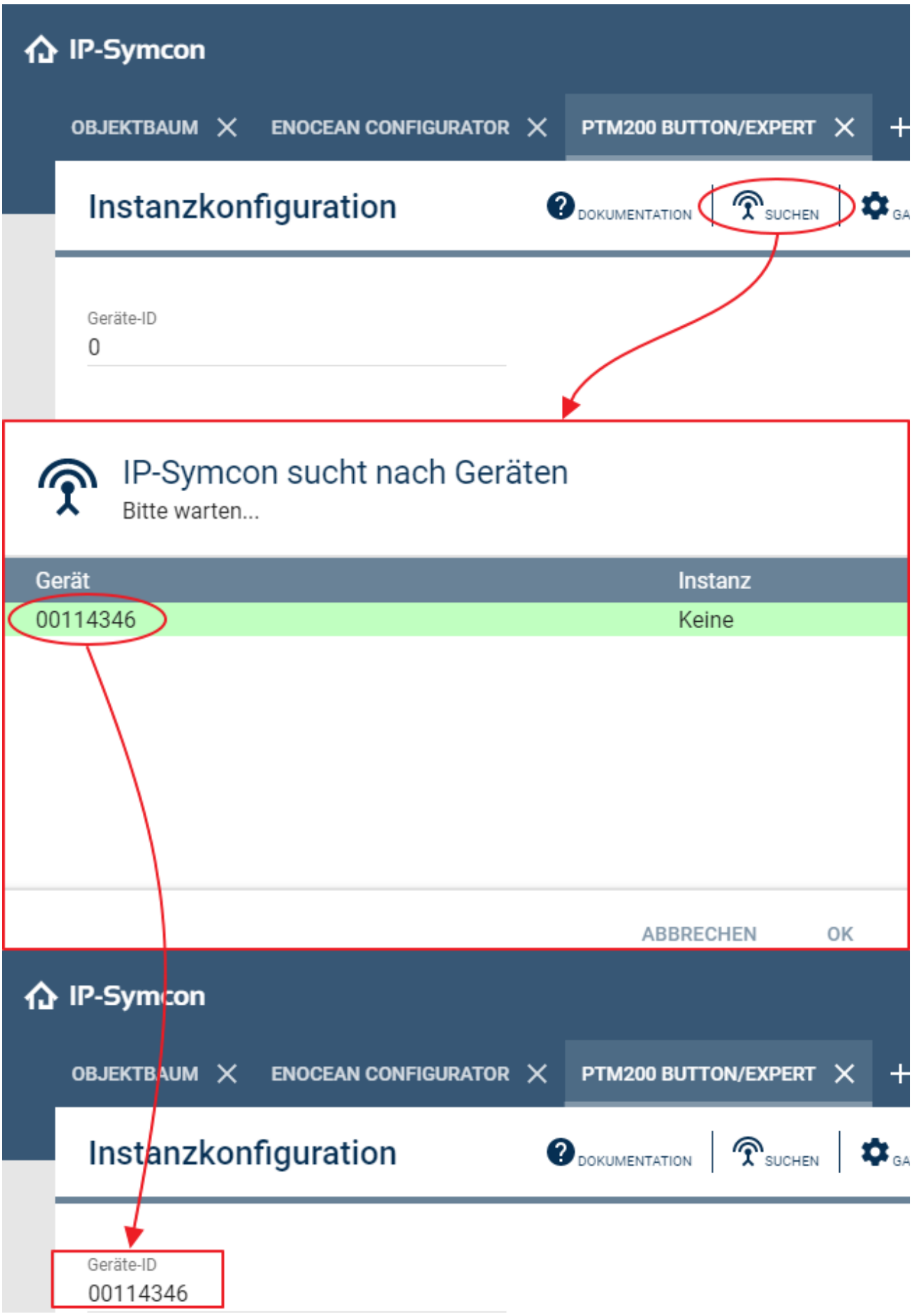
Eine allgemeine Anleitung zum Erstellen von Geräten/Instanzen in IP-Symcon kann hier gefunden werden: Geräte einbinden

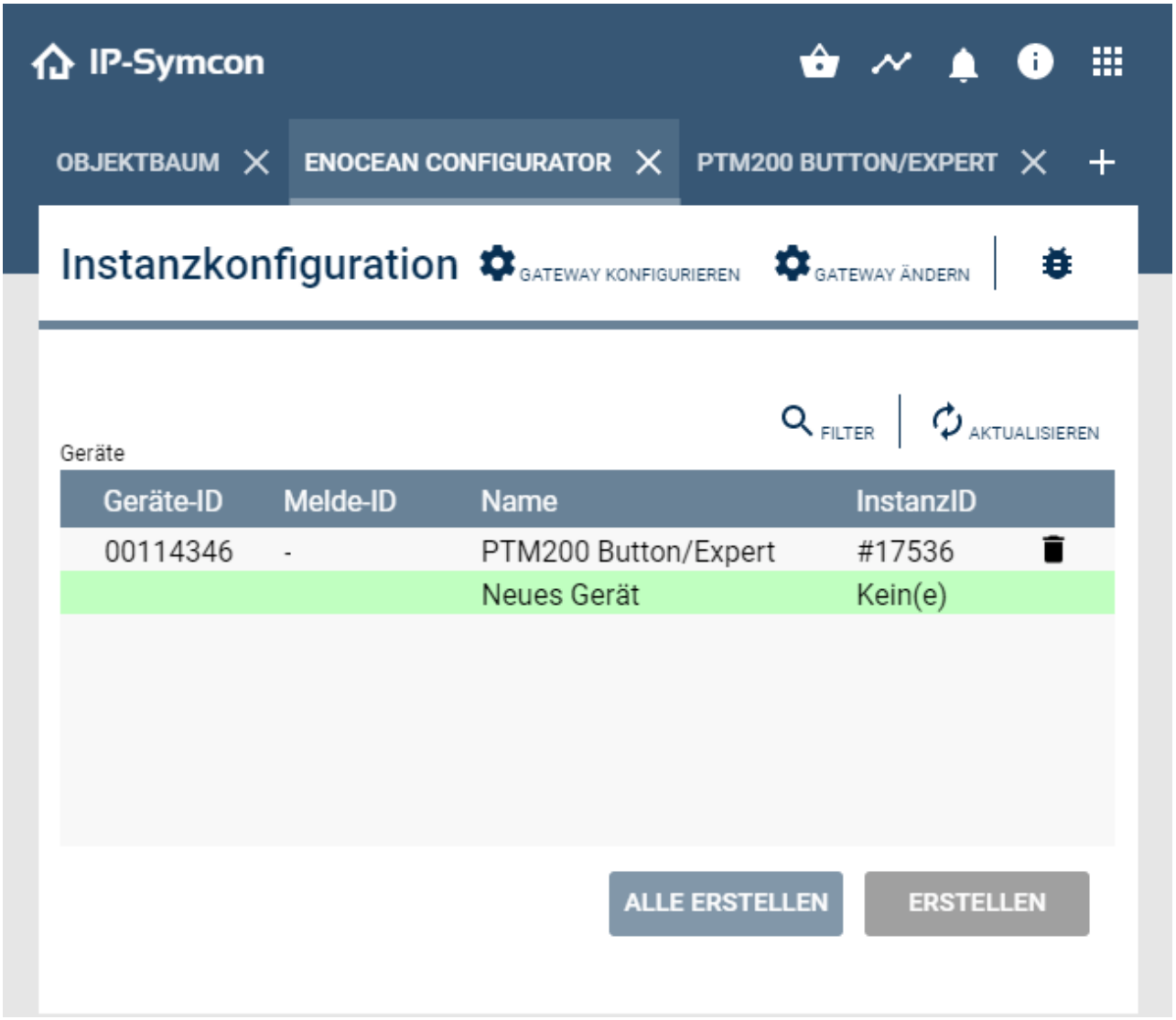
## KONFIGURATOR

Über den Konfigurator kann jedes EnOcean-Gerät als Instanz erstellt werden. Dazu muss "Neues Gerät" ausgewählt, auf "Erstellen" geklickt und anschließend das passende Gerät aus der Liste selektiert werden.




Sobald im Konfigurator eine Instanz erstellt und auf die Konfigurationstaste gedrückt wird, öffnet sich die Konfigurationsseite der Geräteinstanz. In dieser kann über den "Suchen"-Button und mehrfaches betätigen des EnOceangeräts die Geräte-ID gesucht und ausgewählt werden.


IP-Symcon
OBJEKTBAUM
ENOCEAN CONFIGURATOR
PTM200 BUTTON/EXPERT
Instanzkonfiguration
DOKUMENTATION
SUCHEN
Geräte-ID
0
IP-Symcon sucht nach Geräten
Bitte warten...
Gerät
Instanz
00114346
Keine
ABBRECHEN
OK
IP-Symcon
OBJEKTBAUM
ENOCEAN CONFIGURATOR
PTM200 BUTTON/EXPERT
Instanzkonfiguration
DOKUMENTATION
SUCHEN
Geräte-ID
00114346




Neue Instanzen werden im Objektbaum in der Hauptkategorie erstellt. Diese erstellten Instanzen können dann entsprechend umbenannt und an anderer Stelle einsortiert werden. Auch besteht die Möglichkeit via "Konfigurieren" im Konfigurator die jeweilige Instanzkonfiguration aufzurufen.

## AKTOR

Geräte-ID eintragen und mit "Übernehmen" speichern
Gerät in den Einlernmodus versetzen (siehe Herstelleranleitung)
"Einlernen" klicken
Funktion ggf. in der Testumgebung mit An/Aus testen

| ID | BESCHREIBUNG |
|---|---|
| Geräte-ID | Wird vom Benutzer vergeben. 0-127 ganzzahlig möglich |

## AKTOR (ELTAKO)

Den Geräten über das Anmelden am FAM14 eine ID zuweisen (siehe Herstelleranleitung)
Geräte-ID eintragen und mit "Übernehmen" speichern
Diese kann am einfachsten über die PCT14 eingesehen werden.
Gerät in den Einlernmodus versetzen (siehe Herstelleranleitung)

"Einlernen" klicken in der IP-Symocn Instanz
Funktion ggf. in der Testumgebung mit An/Aus testen
"Suchen" klicken. Das Gerät sollte 5mal An/Aus gehen
Gefundene ID auswählen und mit "OK" übernehmen

**Besonderheiten bei bestimmten Geräten**

**FUD14:** Im PCT14 muss im Konfigurationsbereich des FUD14 die Optionen "Bestätigungstelegramm mit Dimmwert" und "Bestätigungstelegramm mit Tastertelegramm" auf "ein" setzen. Es muss, wenn nicht durchs Anlernen eingetragen, innerhalb der PCT14 im Reiter "ID - Zuordnungsbereich" des Geräts in der Funktiongruppe 3 die Funktion 32 "Dimmwert von GFVS / Drehtaster" eingerichtet werden.
**FSR14-xR:** Muss als Richtungstaster oder Universal-Taster angelernt werden, je nachdem ob man mit der Taste Ein-/Ausschalten oder mit 2 Tasten schalten möchte.
**FSB14:** Es muss, wenn nicht durchs Anlernen eingetragen, innerhalb der PCT14 im Reiter "ID - Zuordnungsbereich" des Geräts in der Funktiongruppe die Funktion "Fahrzeit mit Zeitwertvorgabe FVS" eingerichtet werden.

Solange eine Verbindung über USB mit dem FAM14 über die PCT14 aktiv ist, ist ein Senden von Funktelegramme nicht Möglich.

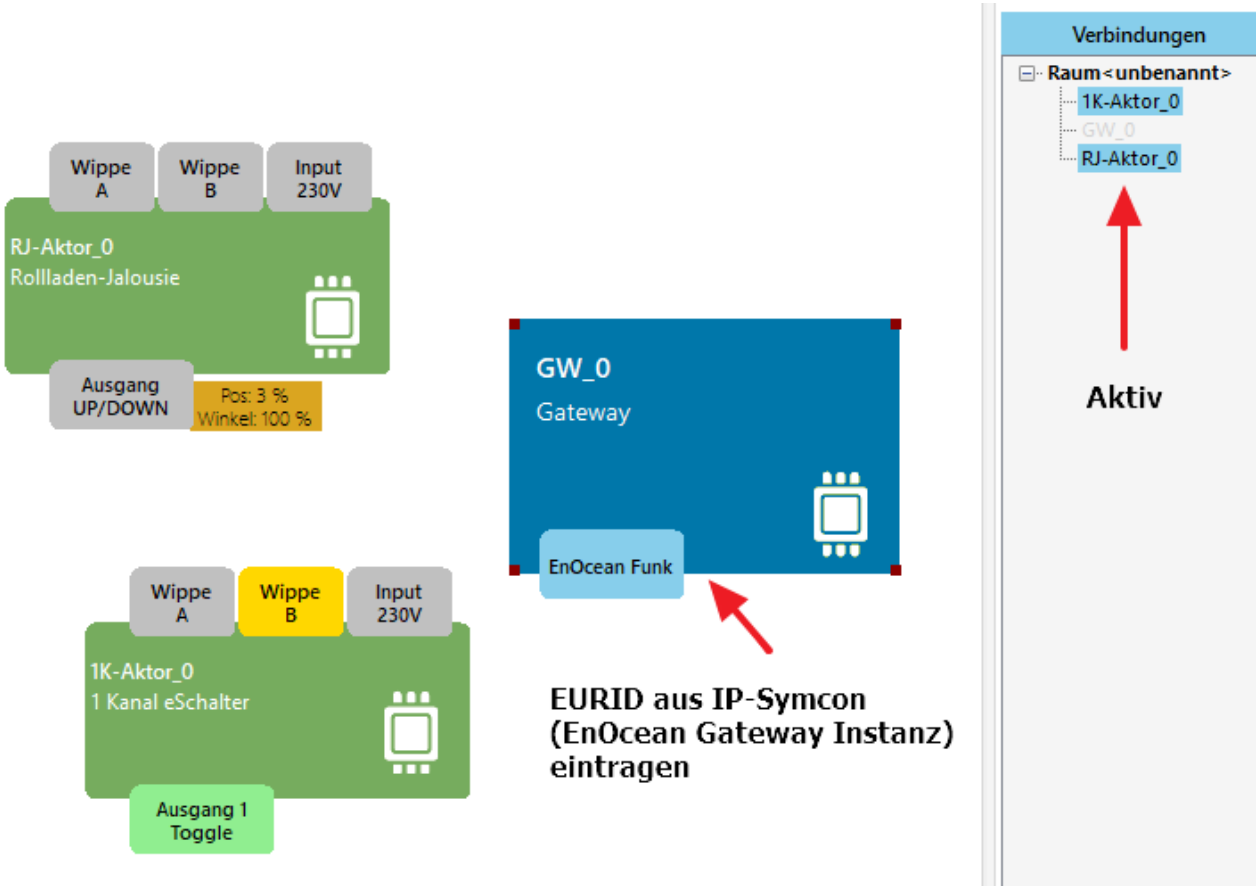
Bei Eltako Geräten muss der Schaltaktor im Richtungstastermodus eingelernt werden. Der Dimmaktor muss im Modus P,L,C (auch PCT) eingelernt werden. Dafür ist die speziell dafür vorgesehene Instanz und der Einlernen-Button zu nutzen. (Ab IP-Symcon 2.5)

Das Gerät Eltako FSB14 kann nur in RV 180/200 (Motor 1/2) eingelernt werden.

| ID | BESCHREIBUNG |
|---|---|
| Geräte-ID | Wird vom Benutzer vergeben. 0-127 ganzzahlig möglich |
| Melde-ID | Wird vom Gerät gesendet. Dient der Gerätezuordnung |

## AKTOR (OPUS)

Es wird das OPUS Config Tool in einer aktuellen Version benötigt, sodass ein generisches "Gateway" hinzugefügt werden kann.
Im Gateway muss die EURID vom verwendeten EnOcean Gateway eingetragen werden. (Diese ist unterhalb der EnOcean Gateway Splitter Instanz zu finden) [Bei älteren TCM310 ist es evtl. erforderlich dort die BaseID einzutragen]
Geräte im Gateway "verbinden" [Diese sind dann blau hinterlegt] und Geräte programmieren lassen
Die Geräte manuell über "Instanz hinzufügen" in IP-Symcon einbinden und die bekannte Geräte-ID in die Instanz eintragen
Schalten und empfangen sollte jetzt problemlos möglich sein




| ID | BESCHREIBUNG |
|---|---|
| Geräte-ID | 8-stellige EnOcean-ID. Diese ist auf dem Gerät zu finden |

## AKTOR (STELLANTRIEB)

Geräte-ID eintragen und mit "Übernehmen" speichern
"Suchen" klicken.
Gerät in den Einlernmodus versetzen (siehe Herstelleranleitung)
Gerät bestätigt das Einlernen (siehe Herstelleranleitung)
Gefundene ID auswählen, mit "OK" bestätigen und "Übernehmen"

| ID | BESCHREIBUNG |
|---|---|
| Geräte-ID | Wird vom Benutzer vergeben. 0-127 ganzzahlig möglich |
| Melde-ID | Wird vom Gerät gesendet. Dient der Gerätezuordnung |

## SENSOR

"Suchen" klicken.
Auf Datenempfang warten (ggf. Taster betätigen)
Gefundene ID auswählen und "Übernehmen"
Der Suchdialog zeigt ein Gerät erst an, wenn Daten empfangen wurden. Bei einem Taster können die Tasten gedrückt werden, um einen Sendevorgang aufzurufen. Bei Temperatursensoren muss das nächste Intervall abgewartet oder die ID manuell angegeben werden.

| ID | BESCHREIBUNG |
| --- | --- |
| Geräte-ID | Wird vom Gerät gesendet. Dient der Gerätezuordnung |

## TIPPS & TRICKS

Um Empfangsprobleme zu beseitigen oder um größere Distanzen zu überbrücken, können bis zu zwei Repeater vom Typ “TCM 120” eingesetzt werden.
Die Anleitung zur Änderung der BaseID (nicht zu verwechseln mit der Geräte ID) befindet sich hier .

Weitere Informationen über die verwendete Technologie und Bezugsquellen gibt es auf der Internetpräsenz von EnOcean.
Link: enocean.com/de

# ENO_DimSet

```
boolean ENO_DimSet (int $InstanzID, int $Intensität)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz des zu schaltenden Gerätes |
| Intensität | 1..100 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Dimmt das Gerät mit der ID **InstanzID** auf den Wert **Intensität**.



Diese Funktion kann nur mit Eltako Dimmer Instanzen verwendet werden.

## BEISPIEL

```
ENO_DimSet(12345, 10); //Gerät auf 10% dimmen
```

# ENO_SetActiveMessage

```
boolean ENO_SetActiveMessage (int $InstanzID, int $Message)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz des zu schaltenden Gerätes |
| Message | 0..8 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt eine Message mit dem Wert **Message** auf der Instanz mit der ID **InstanzID** aktiv.



Diese Funktion kann nur mit EnOcean Thanos Instanzen verwendet werden.

## BEISPIEL

```
ENO_SetActiveMessage(12345, 2);
```

# ENO_SetFanStage

```
boolean ENO_SetFanStage (int $InstanzID, int $FanStage)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz des zu schaltenden Gerätes |
| FanStage | 0..4 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt die FanStage des Geräts mit der ID **InstanzID** auf den Wert **Fanstage**.

| FANSTAGE WERT | BESCHREIBUNG |
|---|---|
| 0 | off |
| 1 | low |
| 2 | medium |
| 3 | high |
| 4 | automatic |



Diese Funktion kann nur mit EnOcean Thanos Instanzen verwendet werden.

## BEISPIEL

```
ENO_SetFanStage(12345, 4); //Gerät wird auf automatic gesetzt
```

# ENO_SetIntensity

```
boolean ENO_SetIntensity (int $InstanzID, boolean $Status, int
$Intensität)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz des zu schaltenden Gerätes |
| Status | TRUE für An, FALSE für Aus |
| Intensität | 1..20, 1 = 5%, 20 = 100% |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Dimmt das Gerät mit der ID **InstanzID** auf den Wert **Intensität**. Der Wert **Status** gibt an, ob das Gerät an/aus geschaltet werden soll.



Da das EnOcean System unidirektional ist, kann nicht festgestellt werden, ob der Befehl erfolgreich war. Fehlermeldungen werden nur gesendet, wenn ein systeminterner Fehler aufgetreten ist (z.B. wenn das Interface nicht angeschlossen ist).



Das EnOcean Protokoll sieht keine Dimmfunktion vor. Der Opus-Dimmer wird durch einen “Trick” über eine Zeitfunktion angesteuert, sodass das Resultat der Helligkeit je nach Vorgang ein wenig abweichen kann. Die Prozentwerte sind nur ungefähr und es kann keine lineare Helligkeitssteigerung angenommen werden.



Diese Funktion kann nur mit der Opus Dimmer Instanz verwendet werden.

## BEISPIEL

```
ENO_SetIntensity(12345, true, 10); //Gerät auf ca. 50% dimmen
```

# ENO_SetLockFanStage

```
boolean ENO_SetLockFanStage (int $InstanzID, boolean $Locked)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz des zu schaltenden Gerätes |
| Locked | TRUE für An, FALSE für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Locked die FanStage des Gerätes mit der ID InstanzID.



Diese Funktion kann nur mit EnOcean Thanos Instanzen verwendet werden.

## BEISPIEL

```
ENO_SetLockFanStage(12345, true);
```

# ENO_SetLockRoomOccupancy

```
boolean ENO_SetLockRoomOccupancy (int $InstanzID, boolean $Locked)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz des zu schaltenden Gerätes |
| Locked | TRUE für An, FALSE für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Locked die Occupancy des Gerätes mit der ID InstanzID.



Diese Funktion kann nur mit EnOcean Thanos Instanzen verwendet werden.

## BEISPIEL

```
ENO_SetLockRoomOccupancy(12345, true);
```

# ENO_SetMode

```
boolean ENO_SetMode (int $InstanzID, int $Value)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz des zu schaltenden Gerätes |
| Value | 0..5 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den Modus vom Stellantrieb mit der ID **InstanzID** auf den Wert **Value**.

| VALUE-WERT | BESCHREIBUNG |
|---|---|
| 0 | Default |
| 1 | Summer Mode |
| 2 | TestClose |
| 3 | TestOpen |
| 4 | Liftset |
| 5 | Init |

## BEISPIEL

```
ENO_SetMode(12345, 1); //Stellantrieb wird in den Summer Mode
gesetzt
```

# ENO_SetPosition

```
boolean ENO_SetPosition (int $InstanzID, int $Value)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz des zu schaltenden Gerätes |
| Value | 1..100 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt die Position des Stellantriebes mit der ID **InstanzID** auf den Wert **Value**



Der Modus des Stellantriebes muss auf "Valve position" stehen, damit diese Funktion aufgerufen werden kann.

## BEISPIEL

```
ENO_SetPosition(12345, 10);
```

# ENO_SetRoomOccupancy

```
boolean ENO_SetRoomOccupancy (int $InstanzID, boolean $Occupied)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz des zu schaltenden Gerätes |
| Occupied | TRUE für An, FALSE für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Setzt die Occupancy ("Raumbelegung") des Gerätes mit der ID **InstanzID** auf den Wert **Occupied**.



Diese Funktion kann nur mit EnOcean Thanos Instanzen verwendet werden.

## BEISPIEL

```
ENO_SetRoomOccupancy(12345, true);
```

# ENO_SetTemperature

```
boolean ENO_SetTemperature (int $InstanzID, float $Value)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz des zu schaltenden Gerätes |
| Value | 1...100 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den Soll-Temperaturwert des Stellantriebes mit der ID **InstanzID** auf den Wert **Value**.



Der Modus des Stellantriebes muss auf "Temperature Setpoint" stehen, damit diese Funktion aufgerufen werden.



Soll ein EnOcean Thanos Gerät auf eine Soll-Temperatur eingestellt werden ist der Befehl ENO_SetTemperature1 zu nutzen.

## BEISPIEL

```
ENO_SetTemperature(12345, 22.7);
```

# ENO_SetTemperature1

```
boolean ENO_SetTemperature1 (int $InstanzID, float $Temperature)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz des zu schaltenden Gerätes |
| Temperature | 1..100 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den Temperaturwert des Gerätes mit der ID **InstanzID** auf den Wert **Temperature**.



Diese Funktion kann nur mit EnOcean Thanos Instanzen verwendet werden. Soll ein Stellantrieb auf eine Soll-Temperatur eingestellt werden ist der Befehl ENO_SetTemperature zu nutzen.

## BEISPIEL

```
ENO_SetTemperature1(12345, 10);
```

# ENO_ShutterMoveDown

```
boolean ENO_ShutterMoveDown (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz des zu schaltenden Gerätes |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Bewegt den Rollladen mit der ID InstanzID nach unten, bis ENO_ShutterStop ausgeführt oder die Endposition erreicht wird.

## BEISPIEL

```
ENO_ShutterMoveDown(12345);
```

# ENO_ShutterMoveDownEx

Benötigt: IP-Symcon >= 4.2

```
boolean ENO_ShutterMoveDownEx (int $InstanzID, float $Laufzeit)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz des zu schaltenden Gerätes |
| Laufzeit | Sekunden wielange der Rollladen laufen soll. (0 - 6535.5) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Bewegt den Rollladen mit der ID InstanzID nach unten, bis ENO_ShutterStop ausgeführt oder die Laufzeit **Laufzeit** abgelaufen ist.



Diese Funktion wird nur von Eltako Shuttern unterstützt

## BEISPIEL

```
//Rollo für 10,5 Sekunden nach Unten fahren
ENO_ShutterMoveDownEx(12345, 10.5);
```

# ENO_ShutterMoveUp

```
boolean ENO_ShutterMoveUp (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz des zu schaltenden Gerätes |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Bewegt den Rollladen mit der ID InstanzID nach oben, bis ENO_ShutterStop ausgeführt oder die Endposition erreicht wird.

## BEISPIEL

```
ENO_ShutterMoveUp(12345);
```

# ENO_ShutterMoveUpEx

Benötigt: IP-Symcon >= 4.2

```
boolean ENO_ShutterMoveUpEx (int $InstanzID, float $Laufzeit)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz des zu schaltenden Gerätes |
| Laufzeit | Sekunden wielange das Rollo laufen soll. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Bewegt den Rollladen mit der ID InstanzID nach unten, bis ENO_ShutterStop ausgeführt oder die Laufzeit **Laufzeit** abgelaufen ist.



Diese Funktion wird nur von Eltako Shuttern unterstützt

## BEISPIEL

```
//Rollo für 10,5 Sekunden nach Oben fahren
ENO_ShutterMoveUpEx(12345, 10.5);
```

# ENO_ShutterStop

```
boolean ENO_ShutterStop (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz des zu schaltenden Gerätes |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Stoppt eine aktuelle Bewegung des Rollladens.

## BEISPIEL

```
ENO_ShutterStop(12345);
```

# ENO_SwitchMode

```
boolean ENO_SwitchMode (int $InstanzID, boolean $Status)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz des zu schaltenden Gerätes |
| Status | TRUE für An, FALSE für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Schaltet das Gerät mit der ID InstanzID auf den Wert Status



Da das EnOcean System unidirektonal ist, kann nicht festgestellt werden, ob der Befehl erfolgreich war. Fehlermeldungen werden nur gesendet, wenn ein systeminterner Fehler aufgetreten ist, z.B. wenn das Interface nicht angeschlossen ist.

## BEISPIEL

```
ENO_SwitchMode(12345, true); //Gerät einschalten
```

# ENO_SwitchModeEx

```
boolean ENO_SwitchModeEx (int $InstanzID, boolean $Status, int
$SendeModus)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz des zu schaltenden Gerätes |
| Status | **TRUE** für An, **FALSE** für Aus |
| SendeModus | 0 = NMessage, 1 = UMessage, 2 = Beide ( Wie ENO_SwitchMode ) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schaltet das Gerät mit der ID **InstanzID** auf den Wert **Status**



Da das EnOcean System unidirektonal ist, kann nicht festgestellt werden, ob der Befehl erfolgreich war. Fehlermeldungen werden nur gesendet, wenn ein systeminterner Fehler aufgetreten ist, z.B. wenn das Interface nicht angeschlossen ist.



Für die Erklärung der NMessage und UMessage sollte die jeweilige EnOcean Dokumentation gelesen werden.

## BEISPIEL

```
ENO_SwitchModeEx(12345, true, 0); //NMessage senden
IPS_Sleep(500); //Warten
ENO_SwitchModeEx(12345, true, 1); //UMessage senden
```

# Geräteliste

## UNTERSTÜTZTE GATEWAYS

| PRODUKT | BESCHREIBUNG |
|---|---|
| EnOcean LAN Gateway (TCM310/TCM515 LAN Gateway) Produktinformation | IP Gateway |
| BSC-BOR | USB Gateway |
| EnOcean USB300 | USB Gateway |
| EnOcean Pi | Raspberry Pi Aufsteckplatine |
| EnOcean FGW14 (mit FAM14) | USB/Seriell Gateway (Betriebsart 6) und Funkerweiterung FAM14 (Betriebsart 2) |
| FAM USB | USB Gateway (Muss als TCM 130 eingetragen werden) |
| TCM130 | Alle Geräte die das ESP2 Protokoll unterstützen |
| TCM310 | Alle Geräte die das ESP3 Protokoll (Dolphin) unterstützen |
| TCM515 | Alle Geräte die das ESP3 Protokoll (Dolphin) unterstützen |
| Thermokon STC-Ethernet | IP Gateway |

## UNTERSTÜTZTE GERÄTE

Folgende Geräte wurden auf Ihre Kompatibilität mit IP-Symcon untersucht.
Auf Anfrage überprüfen wir gerne weitere, nicht aufgelistete, Geräte.

### ALLGEMEIN

| PRODUKT | BESCHREIBUNG |
|---|---|
| PTM 200 | Button/Expert |
| PTM 200 | Schalter |
| RCM 1xx | Schaltaktor (Universell) |
| STM100 | Sensor |
| STM250 | Sensor |

### ALPHAEOS

| PRODUKT | BESCHREIBUNG |
|---|---|
| Drive B2 | Funkstellantrieb |
| SENSE TF-H | Raumklima-Sensor (Temperatur und Luftfeuchtigkeit) |

| PRODUKT | BESCHREIBUNG | EEP NUMMER (SIEHE EEP LISTE UNTEN) |
|---|---|---|
| DSZ14DRS | Funk-Wechselstromzähler-Sendemodul | EEP: A5-12-01 |
| DSZ14WDRS | Funk-Wechselstromzähler-Sendemodul | EEP: A5-12-01 |
| F2L14 | RS485-Bus-Aktor 2-Stufen-Lüftungsrelais | eigenständiges Modul |
| F3Z14D | RS485-Bus-Stromzähler-Sammler | EEP: A5-12-01 |
| F4HK14 | 4-Kanal-Heiz-Kühl-Relais | eigenständiges Modul |
| F4T65 | Funksensoren batterie-/leitungslos | EEP: F6-02-01 |
| FABH65S | Außen-Bewegungs-Helligkeitssensor | eigenständiges Modul |
| FADS60-230V | Funkaktor für Außen-Dämmerungssensor | eigenständiges Modul |
| FAE14LPR | Einzelraumregelung Heizen/Kühlen für 2 Zonen | eigenständiges Modul |
| FAE14SSR | Einzelraumregelung Heizen/Kühlen für 2 Zonen mit Solid-State-Relais | eigenständiges Modul |
| FAFT60 | Außen-Feuchte-Temperatursensor | eigenständiges Modul |
| FAH60 | Außen-Helligkeitssensor | eigenständiges Modul |
| FAH60B | Außen-Helligkeitssensor | eigenständiges Modul, EEP: A5-06-01 |
| FAH63 | Außen-Helligkeitssensor | eigenständiges Modul |
| FAH65S | Außen-Helligkeitssensor | eigenständiges Modul |
| FASM60 | Funk-Außen-Sendemodul | EEP: F6-02-01 |
| FBH65B | Bewegungs-Helligkeitssensor | eigenständiges Modul |
| FBH65S | Bewegungs-Helligkeitssensor | eigenständiges Modul |
| FBH65TFB | Bewegungs-Helligkeitssensor | eigenständiges Modul |
| FCO2TF65 | CO2-Sensor | EEP: A5-09-04 |
| FD62NP-230V | Funk-Universal-Dimmaktor ohne N-Anschluss | eigenständiges Modul |
| FD62NPN-230V | Funk-Universal-Dimmaktor | eigenständiges Modul |
| FFR14 | RS485-Bus-Aktor Feldfreischalter | eigenständiges Modul |
| FFR61-230V | Feldfreischalter | eigenständiges Modul |
| FHF | Funksensor Fenstergriff | EEP: F6-10-00 |
| FHK14 | Funkaktor für Heiz-Kühl-Relais | eigenständiges Modul |
| FHK61SSR-230V | Funkaktor für Heiz-Kühl-Relais | eigenständiges Modul |
| FHK61U-230V | Funkaktor für Heizrelais | eigenständiges Modul |
| FIFT65S | Innen-Feuchte- | eigenständiges Modul |

## HOPPE

| PRODUKT | BESCHREIBUNG | EEP NUMMER (SIEHE EEP LISTE UNTEN) |
|---|---|---|
| FHF | Fenstergriff | F6-10-00 |
| SecuSignal | eFenstergriff Amsterdam E0400/ FR-408 F69 | F6-10-00 |
| SecuSignal | eFenstergriff Atlanta 0530S/ FR-408 100 Nm F9 | F6-10-00 |
| SecuSignal | eFenstergriff Toulon E0400/ FR-408 F69 | F6-10-00 |
| ConnectHome | eFenstergriff Tokyo | F6-10-00 |
| ConnectHome | eFenstergriff New York | F6-10-00 |

## HORA

| PRODUKT | BESCHREIBUNG |
|---|---|
| SmartDrive MX | Stellantrieb |

## KIEBACK & PETER

| PRODUKT | BESCHREIBUNG |
|---|---|
| MD15 | Kleinstellantrieb |

## MACO

| PRODUKT | BESCHREIBUNG | EEP NUMMER (SIEHE EEP LISTE UNTEN) |
|---|---|---|
| Smart Ready Sensor eTRONIC | Tür-/ Fenstersensor | EEP: A5-14-01 |

## NODON

| PRODUKT | BESCHREIBUNG | EEP NUMMER (SIEHE EEP LISTE UNTEN) |
|---|---|---|
| ASP-2-1-00 | Smart Plug | EEP: D2-01-0A |
| ASP-2-1-01 | Smart Plug Metering | EEP: D2-01-0B |
| ASP-2-1-10 | Smart Plug | EEP: D2-01-0A |
| ASP-2-1-11 | Smart Plug Metering | EEP: D2-01-0B |
| SIN-2-1-01 | In-wall Multifunction Relay Switch 1-Channel | EEP: D2-01-0F |
| SIN-2-2-01 | In-wall ON/OFF Lighting Relay Switch 2-Channel | EEP: D2-01-0B |
| SIN-2-RS-01 | Roller Shutter Relay Switch | EEP: D2-05-00 |

## OPUS

| PRODUKT | BESCHREIBUNG | EEP NUMMER (SIEHE EEP LISTE UNTEN) |
|---|---|---|
| 514.441 | 55 Wandsender-Modul Standard | EEP: F6-02-01 |
| 561.150.9052 | Temperatursensor (2 Tasten) | EEP: A5-10-12 |
| 561.150.9069 | Temperatursensor (4 Tasten) | EEP: A5-10-22 |
| 563.010-C* | Bridge 1 Kanal | EEP: D2-01-01 |
| 563.014-C* | Bridge 1 Kanal, 16A | EEP: D2-01-01 |
| 563.020-C* | Bridge 2 Kanal | EEP: D2-01-11 |
| 563.031-C* | Bridge Rollladen / Jalousie | EEP: D2-05-02 |
| 563.035-C* | Bridge Dimmer | EEP: D2-01-03 |
| 563.051-C | Smart Motion Sensor (Licht) | EEP: F6-02-01 |
| 563.052-C | Smart Motion Sensor Multifunktion | EEP: F6-02-01 |
| 563.053 | Smart Motion Sensor Ambient Assisted Living - Bewegungsunfähigkeit | EEP: A5-07-03 |
| 563.055-C | Smart Motion Sensor (Anwesenheit) | EEP: A5-07-03 |
| 563.067-C | Wassermelder (Sofortmelder) | EEP: F6-05-01 |
| 563.068-C | Wassermelder (Quellscheiben) | EEP: A5-30-03 |
| 563.069-C | Rauchwarnmelder | EEP: F6-05-02 |

*Benötigt Einrichtung über Opus Config Tool (563.040)

### PEHA

| PRODUKT | BESCHREIBUNG |
|---|---|
| Diverse | Alle Geräte welche EEP unterstützen |

### THERMOKON

| PRODUKT | BESCHREIBUNG |
|---|---|
| SR04 | Temperatur/Feuchtigkeit |
| SAB02 | Stellantrieb |
| Thanos | Raumbediengerät |

## EEP

Alle Geräte die folgende EEPs verwenden werden von IP-Symcon unterstützt.

### AKTOREN

| NEU | AB VERSION | ALT | BESCHREIBUNG |
|---|---|---|---|
| A5-20-12 | 4.0 | 07-20-12 | Temperature Controller Input |

## BI-DIREKTIONAL

| NEU | AB VERSION | ALT | BESCHREIBUNG |
|---|---|---|---|
| A5-20-01 | 4.0 | 07-20-01 | Battery Powered Actuator |
| D2-01-00 | 4.3 | | Electronic switches and dimmer with Energy Measurement and Local Content |
| D2-01-01 | 4.3 | | Electronic switches and dimmer with Energy Measurement and Local Content |
| D2-01-02 | 4.3 | | Electronic switches and dimmer with Energy Measurement and Local Content |
| D2-01-03 | 4.3 | | Electronic switches and dimmer with Energy Measurement and Local Content |
| D2-01-04 | 4.3 | | Electronic switches and dimmer with Energy Measurement and Local Content |
| D2-01-05 | 4.3 | | Electronic switches and dimmer with Energy Measurement and Local Content |
| D2-01-06 | 4.3 | | Electronic switches and dimmer with Energy Measurement and Local Content |
| D2-01-07 | 4.3 | | Electronic switches and dimmer with Energy Measurement and Local Content |
| D2-01-08 | 4.3 | | Electronic switches and dimmer with Energy Measurement and Local Content |
| D2-01-09 | 4.3 | | Electronic switches and dimmer with Energy Measurement and Local Content |
| D2-01-0A | 4.3 | | Electronic switches and dimmer with Energy Measurement and Local Content |
| D2-01-0B | 4.3 | | Electronic switches and dimmer with Energy Measurement and Local Content |
| D2-01-0C | 4.3 | | Electronic switches and dimmer with Energy Measurement and Local Content |
| D2-01-0D | 4.3 | | Electronic switches and dimmer with Energy Measurement and Local Content |
| D2-01-0E | 4.3 | | Electronic switches and dimmer with Energy Measurement and Local Content |
| D2-01-0F | 4.3 | | Electronic switches and dimmer with Energy Measurement and Local Content |
| D2-01-10 | 4.3 | | Electronic switches and dimmer with Energy Measurement and Local Content |
| D2-01-11 | 4.3 | | Electronic switches and dimmer with Energy Measurement and Local Content |
| D2-01-12 | 4.3 | | Electronic switches and dimmer with Energy Measurement and Local Content |
| D2-01-13 | 4.3 | | Electronic switches and dimmer with Energy Measurement and Local Content |
| D2-01-14 | 4.3 | | Electronic switches and dimmer with Energy Measurement and Local Content |
| D2-03-00 | 5.5 | | Light, Switching + Blind Control Type 0x00 |
| D2-03-0A | 5.5 | | Push Button – Single Button |

## SENSOREN

| NEU | AB VERSION | ALT | BESCHREIBUNG |
|---|---|---|---|
| A5-02-01 | 4.0 | 07-02-01 | Temperature Sensor Range -40°C to 0°C |
| A5-02-02 | 4.0 | 07-02-02 | Temperature Sensor Range -30°C to +10°C |
| A5-02-03 | 4.0 | 07-02-03 | Temperature Sensor Range -20°C to +20°C |
| A5-02-04 | 4.0 | 07-02-04 | Temperature Sensor Range -10°C to +30°C |
| A5-02-05 | 4.0 | 07-02-05 | Temperature Sensor Range 0°C to +40°C |
| A5-02-06 | 4.0 | 07-02-06 | Temperature Sensor Range +10°C to +50°C |
| A5-02-07 | 4.0 | 07-02-07 | Temperature Sensor Range +20°C to +60°C |
| A5-02-08 | 4.0 | 07-02-08 | Temperature Sensor Range +30°C to +70°C |
| A5-02-09 | 4.0 | 07-02-09 | Temperature Sensor Range +40°C to +80°C |
| A5-02-0A | 4.0 | 07-02-0A | Temperature Sensor Range +50°C to +90°C |
| A5-02-0B | 4.0 | 07-02-0B | Temperature Sensor Range +60°C to +100°C |
| A5-02-10 | 4.0 | 07-02-10 | Temperature Sensor Range -60°C to +20°C |
| A5-02-11 | 4.0 | 07-02-11 | Temperature Sensor Range -50°C to +30°C |
| A5-02-12 | 4.0 | 07-02-12 | Temperature Sensor Range -40°C to +40°C |
| A5-02-13 | 4.0 | 07-02-13 | Temperature Sensor Range -30°C to +50°C |
| A5-02-14 | 4.0 | 07-02-14 | Temperature Sensor Range -20°C to +60°C |
| A5-02-15 | 4.0 | 07-02-15 | Temperature Sensor Range -10°C to +70°C |
| A5-02-16 | 4.0 | 07-02-16 | Temperature Sensor Range 0°C to +80°C |
| A5-02-17 | 4.0 | 07-02-17 | Temperature Sensor Range +10°C to +90°C |
| A5-02-18 | 4.0 | 07-02-18 | Temperature Sensor Range +20°C to +100°C |
| A5-02-19 | 4.0 | 07-02-19 | Temperature Sensor Range +30°C to +110°C |
| A5-02-1A | 4.0 | 07-02-1A | Temperature Sensor Range +40°C to +120°C |
| A5-02-1B | 4.0 | 07-02-1B | Temperature Sensor Range +50°C to +130°C |
| A5-02-20 | 4.0 | 07-02-20 | 10 Bit Temperature Sensor Range -10°C to +41.2°C |
| A5-02-30 | 4.0 | 07-02-30 | 10 Bit Temperature Sensor Range -40°C to +62.3°C |
| A5-04-01 | 4.0 | 07-04-01 | Range 0°C to +40°C and 0% to 100% |
| A5-04-02 | 4.1 | 07-04-02 | Range -20°C to +60°C and 0% to 100% |
| A5-04-03 | 4.1 | 07-04-03 | Range -20°C to +60°C 10bit-measurement and 0% to 100% |
| A5-04-03 | 5.3 | 07-04-03 | Temperature and Humidity Sensor |
| A5-06-01 | 4.0 | 07-06-01 | Range 300lx to 60.000lx |
| A5-06-02 | 4.0 | 07-06-02 | Range 0lx to 1.020lx |
| A5-06-03 | 4.0 | 07-06-03 | 10-bit measurement (1-Lux resolution) with range 0lx to 1000lx |
| A5-07-01 | 4.0 | 07-07-01 | Occupancy with Supply voltage monitor |
| A5-07-01 | 5.3 | 07-07-01 | gN-Deckenbewegungsmelder 360° |
| A5-07-01 | 5.3 | 07-07-01 | Desk Occupancy |
| A5-07-02 | 4.0 | 07-07-02 | Occupancy with Supply voltage monitor |
| A5-07-03 | 4.0 | 07-07-03 | Occupancy with Supply voltage monitor and 10-bit |

# FHZ1X00PC

Der FHZ1X00PC ist ein Adapter, welcher mit ELV-Geräten (FS20, HMS und FHT) via Funk kommuniziert und für IP-Symcon bereitstellt. Eine Verbindung mit IP-Symcon wird via USB hergestellt.
Unterstütze Komponenten

## BESCHREIBUNG

Die FHZ1X00PC (WLAN) ist eine von Contronics entwickelte und von ELV vertriebene Schnittstelle, die es dem Endanwender ermöglicht, mit FS20, HMS und FHT Komponenten über den PC zu kommunizieren. Ab der FHZ1300PC wird außerdem der Kombi-Wettersensor KS300 unterstützt. Die nur über Contronics erhältliche FHZ1350PC bietet außerdem die Möglichkeit, auf das Telefonwählgerät der ELV zuzugreifen. Diese Funktion wird von IP-Symcon nicht unterstützt.

## INSTALLATION

Nachdem das Gerät an den PC angeschlossen wurde, müssen die Treiber installiert werden. Diesen können gegebenenfalls im Servicebereich des Herstellers ELV heruntergeladen werden.

## TIPPS & TRICKS

Die FHZ ist mit allen gängigen USB->LAN Umwandlern kompatibel. (z.B. von Silex, Lantronix)
Information zum Umbau auf LAN oder zur Reichweitenerhöhung können im Forum gefunden werden
Die FHZ1300PC WLAN wird in IP-Symcon nicht offiziell unterstützt. Um diese in IP-Symcon verwenden zu können, muss der dort verwendete Verschlüsselungsalgorithmus deaktiviert werden.
Die nötigen Schritte können hier nachgelesen werden:
http://www.thinkwiki.org/wiki/User:Akw/FHZ

## PROBLEMLÖSUNG

In der Konfiguration steht: “Eine übergeordnete Konfiguration scheint fehlerhaft zu sein"
Durch Klicken auf die Meldung und aktivieren der entsprechenden Treiber.
Es werden keine Daten empfangen/gesendet
Überprüfen, ob die FHZ am System angeschlossen ist. (Hardware Manager)
Überprüfen, ob bei Senden/Empfangen die Rote LED an der FHZ blinkt.
Falls nicht, hat die FHZ keinen Strom oder wurde noch nicht im System eingerichtet.
Überprüfen, ob der FTDI Treiber richtig aktiviert wurde. (s.o.)
Geräte werden nur sporadisch geschaltet (FS20)
Ist das Gerät zu weit entfernt?
Die FHZ woanders platzieren oder das System um eine weitere FHZ erweitern.
Gibt es Störsender in der Nähe?

# FHT

Das Funk-Heizungssystem FHT dient dazu, um durch intelligente Regelung der Heizungsventile Energie zu sparen und flexibel zu verschiedenen Zeiten ein angenehmes Raumklima zu schaffen.

Nur FHT8b/FHT80b können mit der FHZ1000PC Professional, FHZ1300PC oder FHZ1300PC WLAN verbunden werden. Die wesentlich günstigeren FHT8R sind nur für den Einzel-Betrieb geeignet und können nicht mit der FHZ verbunden werden.

Wochen- und Tagesprogramme werden von IP-Symcon nicht an den FHT übertragen. Dieses Feature wurde schon des Öfteren gewünscht, wird es aber definitiv **nicht** geben.

## KONFIGURATION

### FHT AN FHZ ANMELDEN

Die Taste “PROG” halten, bis “Sond” im Display erscheint.
Das Stellrad drehen, bis “Code” erscheint. Es muss mit der Taste “Funktion" bestätigen werden.
Ein Zahlenpaar (00-99) als Code auswählen.
Durch Drücken der “PROG” Taste wird diese Auswahl bestätigt.
Es muss die “PROG” Taste erneut gedrückt werden.

### FHT IN IP-SYMCON EINRICHTEN:

FHT Instanz erstellen. Siehe Geräte einbinden
Den o.g. Code in der Eigenschaft-Seite angeben und mit “Übernehmen” bestätigen.

### FALLS DER RAUMREGLER EVENTUELL SCHON AN EINER ZENTRALE ANGEMELDET WAR, MUSS DIESE ANMELDUNG GELÖSCHT WERDEN.

“PROG” gedrückt halten bis “Sond” im Display erscheint.
Das Stellrad drehen, bis “Cent” erscheint und “PROG” drücken.
Das Stellrad drehen, bis “nA ” erscheint und “PROG” drücken.
Zur Sicherheit sollte ca. 15 Minuten gewartet werden bis der Abmeldevorgang abgeschlossen wurde.
Den oben beschriebenen Anmeldevorgang durchführen.

Durch die Option “Emuliere Status” wird der Wert der Soll-Variable sofort nach Absenden des Befehls aktualisiert, ohne das eigentliche Feedback vom FHT abzuwarten.

**Nach dem Anmeldevorgang muss mindestens ein Befehl (z.B. Soll-Temperatur setzen) an den FHT gesendet werden, um die Kommunikation zu starten.**

## VERFÜGBARE STATUSVARIABLEN

| STATUS | BESCHREIBUNG |
|---|---|
| LowBattery | Zeigt an, ob der FHT eine Batteriewarnung anzeigt. Falls der Wert TRUE wird, sollte ein Batteriewechsel vorgenommen und im Anschluß den Wert manuell auf FALSE zurückgesetzt werden. |
| Window Open | Zeigt an, ob der am FHT angeschlossene Fensterkontakt ein geöffnetes Fenster meldet. TRUE = Fenster offen, FALSE = Fenster geschlossen |
| Position | Wert der Stellventils von 0%-100% |
| Temperature | Die vom FHT empfangene Ist-Temperatur |
| Target Mode | Der vom FHT emfangene Modus |
| Beim Auslesen | Die Integer Variable wird mit folgenden Werten verwendet: 0: Automatik; 1: Manuell; 2: Urlaub; 3: Party |
| Target Mode (Pending) | Der an den FHT gesendete Modus. Sobald der Funkbefehl abgesetzt werden konnte, sind “Target Mode” und “Target Mode (Pending)” identisch. |
| Target Temperature | Die vom FHT emfangene Soll-Temperatur |
| Target Temperature (Pending) | Die an den FHT gesendete Soll-Temperatur. Sobald der Funkbefehl abgesetzt werden konnte, sind “Target Temperature” und “Target Temperature (Pending)” identisch. |

# FHT_SetMode

```
boolean FHT_SetMode (int $InstanzID, int $Modus)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Modus | **0** = Automatisch, **1** = Manuell |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den neuen Soll-Modus für den FHT mit der ID **InstanzID** auf Wert **Modus**.



Da das FS20 System unidirektonal ist, kann nicht festgestellt werden, ob der Befehl erfolgreich war. Fehlermeldungen werden nur gesendet, wenn ein systeminterner Fehler aufgetreten ist, z.B., wenn die FHZ nicht angeschlossen ist.



Die FHT Befehle werden von der FHZ zwischengepuffert und nur alle paar Minuten an den FHT gesendet. Bitte beachten Sie, dass der Sendepuffer beschränkt ist und mehrmaliges Senden von Werten die Übertragungszeit erhöht und ggf. den FHT überlastet.

## BEISPIEL

```
FHT_SetMode(12345, 1); //Auf manuellen Modus wechseln
```

# FHT_SetTemperature

```
boolean FHT_SetTemperature (int $InstanzID, float $Temperatur)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Temperatur | Temperaturwert in 0,5°C Schritten |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt die neue Soll-Temperatur für den FHT mit der ID **InstanzID** auf Wert **Temperatur**.



Da das FS20 System unidirektonal ist, kann nicht festgestellt werden, ob der Befehl erfolgreich war. Fehlermeldungen werden nur gesendet, wenn ein systeminterner Fehler aufgetreten ist, z.B., wenn die FHZ nicht angeschlossen ist.



Die FHT Befehle werden von der FHZ zwischengepuffert und nur alle paar Minuten an den FHT gesendet. Bitte beachten Sie, dass der Sendepuffer beschränkt ist und mehrmaliges Senden von Werten die Übertragungszeit erhöht und ggf. den FHT überlastet.

## BEISPIEL

```
FHT_SetTemperature(12345, 22.5); //Auf 22.5°C stellen
```

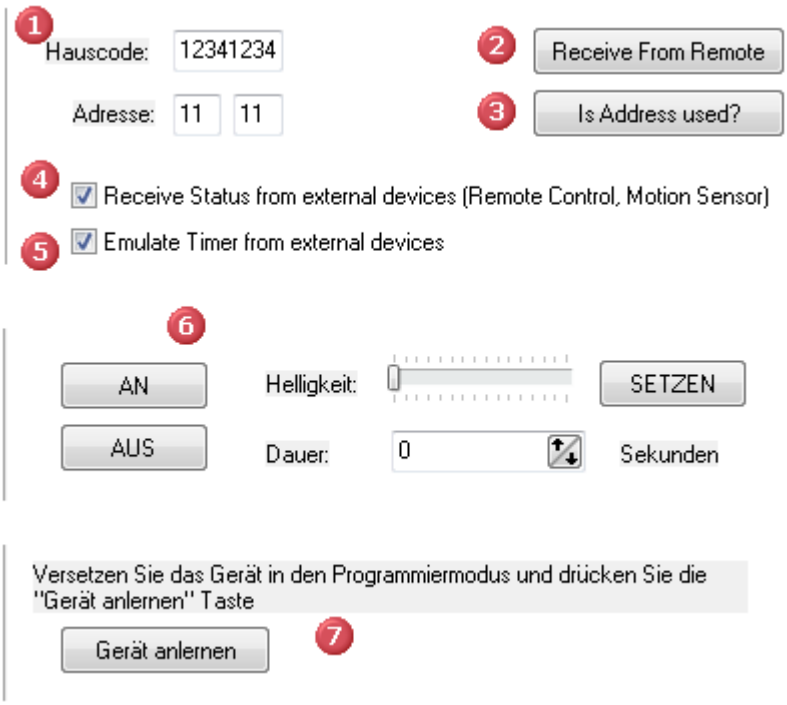
# FS20

Beim Erstellen der Instanz stehen für das FS20 System folgende Möglichkeiten zur Auswahl:

| INSTANZNAME | VERWENDUNGSZWECK |
|---|---|
| FS20 Gerät | Steckdosen, Dimmer, Markisensteuerung, Bewegungsmelder, … |
| FS20 Fernbedienung | Mehrkanalfernbedienungen, -taster, -sender |

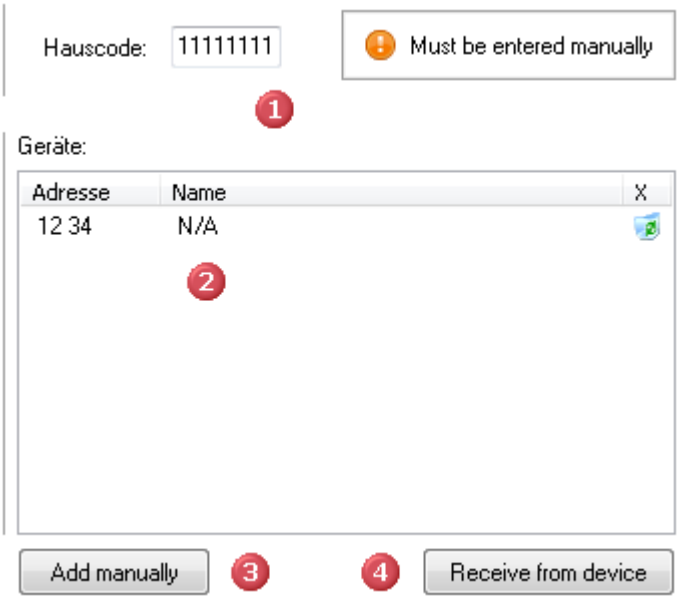
Falls das Gerät noch nicht konfiguriert ist, kann dies via "Geräte einbinden" eingerichtet werden.

## FS20 GERÄT KONFIGURIEREN




Der Hauscode kann per Hand eingegeben und übernommen werden. Mögliche Zeichen: 1-4
Der Hauscode kann von einem anderen Sender empfangen werden (z.B. von einer Fernbedienung).
Es kann überprüft werden, ob die angegebene Adresse bereits von einer anderen Instanz verwendet wird.
Wenn aktiviert, empfängt die Instanz Daten von FS20 Sendern und aktualisiert entsprechend die Statusvariable.
Wenn aktiviert, empfängt die Instanz Daten von FS20 Sendern, wertet die Timerdaten aus und emuliert den Zustand des Timers (z.B. AN für 30Sek).
Nachdem die neuen Einstellungen übernommen wurden, können diese getestet werden, indem An/Aus-Telegramme verschickt werden. Der Wert im Feld Dauer wird beim Übermitteln auf die nächste mögliche Zeit, die das FS20 System auswerten kann, gerundet.
Nachdem die neuen Einstellungen übernommen wurden, können FS20 Empfänger (z.B. Schaltsteckdose) angelernt werden. Dazu müssen diese in den “Anlernmodus” versetzt und innerhalb der nächsten 15 Sekunden durch drücken des Anlern-Buttons programmiert werden.

## FS20 FERNBEDIENUNG KONFIGURIEREN




Bevor die Konfiguration fortgesetzt werden kann, muss der Hauscode per Hand eingegeben werden. Mögliche Zeichen: 1-4
Die Liste zeigt bereits angelernte Taster der Fernbedienung.
Wenn die Adresse bekannt ist, kann diese auch manuell eingegeben werden. Mögliche Zeichen: 1-4
Wenn die Adresse unbekannt ist, kann diese von der Fernbedienung empfangen werden. Es ist zu beachten, dass der Hauscode eingegeben ist und dieser via “Übernehmen” gespeichert wurde.

## VERFÜGBARE STATUSVARIABLEN

| VARIABLE | FUNKTION |
|---|---|
| Status Variable | Zeigt den Status des Geräts an |
| Intensity Variable | Zeigt den Dimwert des Geräts in % an. Es ist zu beachten, dass intern die Werte 0-16 verwendet werden, falls die Variable in Skripten eingelesen werden. |
| Timer Variable | Die Zeit in Sekunden, die der Timer läuft. Pkt. 5 muss in der Konfiguration aktiviert sein, damit diese Variable von IP-Symcon auch ausgewertet wird. (Es wird die empfangene Gesamtzeit angezeigt – nicht die Restlaufzeit.) |
| Data Variable | Zeigt den intern empfangenen Statuscode an. |

Verfügbare Statuscodes:

```
0-16 => 0% – 100%
17 => an, alter Wert (entspricht Wert 11 bei PIRI)
18 => swap
19 => dim up (Einfache Kanalzahl)
20 => dim down (Einfache Kanalzahl)
21 => dim up/down (Doppelte Kanalzahl)
```

```
24 => PIRI; aus für die Einschaltdauer, danach an
25 => PIRI; an, 100% für die Einschaltdauer, danach aus
26 => PIRI; an, alter Wert für die Einschaltdauer, danach aus
30 => PIRI; an, 100% für die Einschaltdauer, danach alter Zustand
31 => PIRI; an, danach alter Zustand für die Einschaltdauer
```

# FS20_DimDown

```
boolean FS20_DimDown (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Dimmt das Gerät mit der ID **InstanzID** eine Stufe runter.



Da das FS20 System unidirektonal ist, kann nicht festgestellt werden, ob der Befehl erfolgreich war. Fehlermeldungen werden nur gesendet, wenn ein systeminterner Fehler aufgetreten ist, z.B., wenn die FHZ nicht angeschlossen ist.

## BEISPIEL

```
FS20_DimDown(12345); //Gerät runterdimmen
```

# FS20_DimUp

```
boolean FS20_DimUp (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Dimmt das Gerät mit der ID **InstanzID** eine Stufe höher.



Da das FS20 System unidirektonal ist, kann nicht festgestellt werden, ob der Befehl erfolgreich war. Fehlermeldungen werden nur gesendet, wenn ein systeminterner Fehler aufgetreten ist, z.B., wenn die FHZ nicht angeschlossen ist.

## BEISPIEL

```
FS20_DimUp(12345); //Gerät hochdimmen
```

# FS20_SetIntensity

```
boolean FS20_SetIntensity (int $InstanzID, int $Intensität, int
$Dauer)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Intensität | 0-16 (0=0%, 16=100%) |
| Dauer | Dauer des Dimmvorgangs |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Dimmt das Gerät mit der ID **InstanzID** auf Wert **Intensität** in **Dauer** Sekunden.



Da das FS20 System unidirektonal ist, kann nicht festgestellt werden, ob der Befehl erfolgreich war. Fehlermeldungen werden nur gesendet, wenn ein systeminterner Fehler aufgetreten ist, z.B., wenn die FHZ nicht angeschlossen ist.

## BEISPIEL

```
FS20_SetIntensity(12345, 16, 5); //Gerät auf 100% in 5 sekunden
dimmen
```

# FS20_SwitchDuration

```
boolean FS20_SwitchDuration (int $InstanzID, boolean $Status, int $Dauer)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Status | TRUE für An, FALSE für Aus |
| Dauer | Ein-/Ausschaltdauer in Sekunden |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Schaltet das Gerät mit der ID **InstanzID** auf Wert **Status** für **Dauer** Sekunden.



Die Anzahl der Sekunden wird auf den nächstmöglichen Wert gerundet, den die Timer-Funktion im FS0 Gerät erlaubt. Mögliche Werte entnehmen Sie z.B. der Wertetabelle Ihres FS20 PIR Handbuchs auf Seite 28.



Da das FS20 System unidirektonal ist, kann nicht festgestellt werden, ob der Befehl erfolgreich war. Fehlermeldungen werden nur gesendet, wenn ein systeminterner Fehler aufgetreten ist, z.B., wenn die FHZ nicht angeschlossen ist.

## BEISPIEL

```
FS20_SwitchDuration(12345, true, 60); //Gerät für 60 sek einschalten
```

# FS20_SwitchMode

```
boolean FS20_SwitchMode (int $InstanzID, boolean $Status)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Status | TRUE für An, FALSE für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schaltet das Gerät mit der ID **InstanzID** auf Wert **Status**.



Da das FS20 System unidirektonal ist, kann nicht festgestellt werden, ob der Befehl erfolgreich war. Fehlermeldungen werden nur gesendet, wenn ein systeminterner Fehler aufgetreten ist, z.B., wenn die FHZ nicht angeschlossen ist.

## BEISPIEL

```
FS20_SwitchMode(12345, true); //Gerät einschalten
```

# Geräteliste

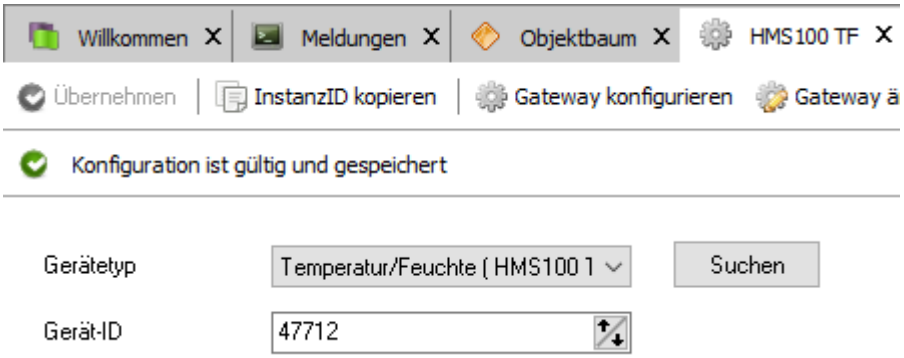
## LISTE DER UNTERSTÜTZTEN GERÄTE

FS20 Sender und Empfänger
Alle FS20 Sender und Empfänger
HMS Sensoren und Aktoren
Alle HMS-Alaram Sensoren und Aktoren
FHT Komponenten
FHT80B/FHT80B2 (FHT8 ist nicht kompatibel!)

# HMS

## EINRICHTEN IN IP-SYMCON

Es muss eine HMS Instanz hinzugefügt werden (Geräte einbinden ). Der Typ kann noch nachträglich auf der Konfigurationsseite geändert werden.




Nach dem "Suchen" muss das Gerät ausgewählt werden. Es wird automatisch eine Geräte-Instanz mit den passenden Variablen erstellt.
Hier anhand des Beispiels eines HMS100 TF.



# HMS_ReleaseFI

```
boolean HMS_ReleaseFI (int $InstanzID, int $Auslöseverzögerung)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Auslöseverzögerung | Verzögerung in Sekunden |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Löst den FI Trenner über das HMS Gerät mit der ID **InstanzID** aus. Der **Auslöseverzögerung** Parameter wird in Sekunden angegeben.



Da das FS20 System unidirektonal ist, kann nicht festgestellt werden, ob der Befehl erfolgreich war. Fehlermeldungen werden nur gesendet, wenn ein systeminterner Fehler aufgetreten ist, z.B., wenn die FHZ nicht angeschlossen ist.



Wenn der FI Trenner keinen FI auslösen kann, wird seine Status Variable auf **FALSE** gesetzt.

## BEISPIEL

```
FS20_ReleaseFI(12345, 0); //FI sofort auslösen
```

# KS300

## INSTALLATION

Falls Sie Ihr Gerät noch nicht eingerichtet haben, folgen Sie bitte den Schritten auf dieser Seite: Geräte einbinden

## TIPPS & TRICKS

Es ist bekannt, dass die FHZ1300PC vom KS300 nur unregelmäßig empfängt.
Eine Diskussion dazu kann im Forum verfolgt werden. Link

# FS10 Wetter

FS10 ist ein Funksystem, welches mit IP-Symcon über einen serielle Schnittstelle verbunden werden kann.

## BESCHREIBUNG

Das **FS10 Funk-System** ist das Vorgängersystem vom FS20-System und arbeitet im 433 MHz Bereich.
Sämtliche Geräte sind nicht mehr im freien Handel verfügbar.
Die interne Codierung ermöglicht bis zu 392 Codevarianten. Um die FS10 Wetter-Komponenten der WS2000er Serie mit IP-Symcon auswerten zu können, benötiget man den PC-Wettersensor der Firma ELV. Dieser wird einfach an die serielle Schnittstelle angeschlossen und somit können leicht bestehende Wetterkomponenten mit in das Haussteuerungssystem integriert werden. Zur Erhöhung der Empfangsqualität empfiehlt sich der Einbau eines Superhet Empfängers.

## INSTALLATION

Für den Wetterempfänger ist keine besondere Installation nötig. Es ist nur ein Anschluss an den Seriellen Port notwendig.

## EINBINDUNG IN IP-SYMCON

Bevor die Suche gestartet werden kann, muss die übergeordnete Instanz konfiguriert werden, indem der korrekte Com Port angegeben und die Konfiguration übernommen wird. Die Einstellung von Baudrate ect. ist nicht notwendig, da dies vom System automatisch vorgenommen wird.

Der Suchdialog zeigt ein Gerät erst an, wenn Daten empfangen wurden. Bei einer Fernbedienung können die Tasten gedrückt, um einen Sendevorgang aufzurufen. Bei Temperatursensoren muss das nächste Intervall abgewartet oder die ID manuell angegeben werden.

Sobald eine ID angegeben wurde, muss die Konfiguration übernommen werden. Dadurch werden auch die passenden Status-Variablen automatisch erstellt.

# GARDENA smart system

Benötigt: IP-Symcon >= 6.0

## BESCHREIBUNG

Das GARDENA-Modul ermöglicht es unterstützte Geräte von GARDENA smart system in IP-Symcon einzubinden.

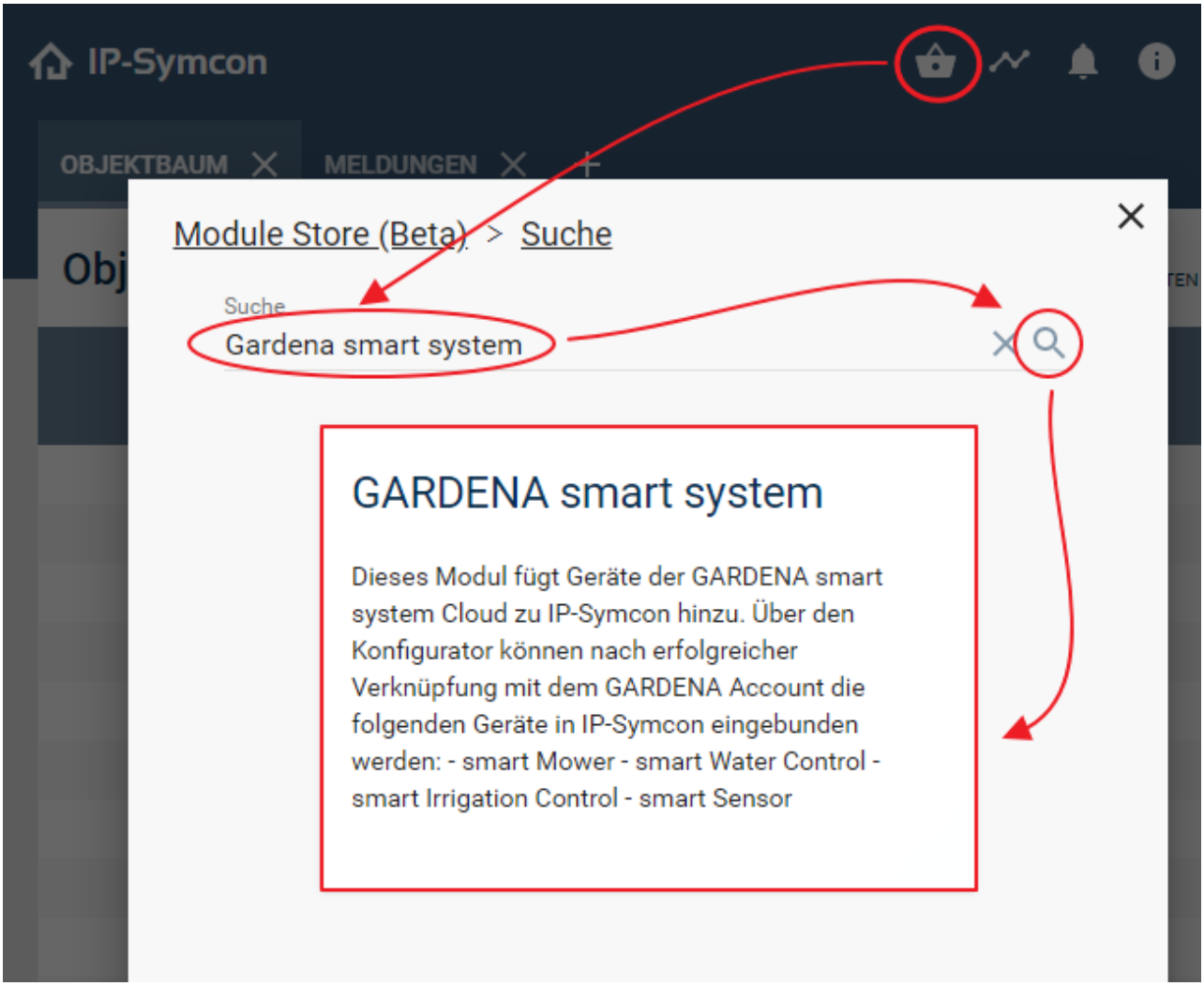
## EINBINDUNG IN IP-SYMCON

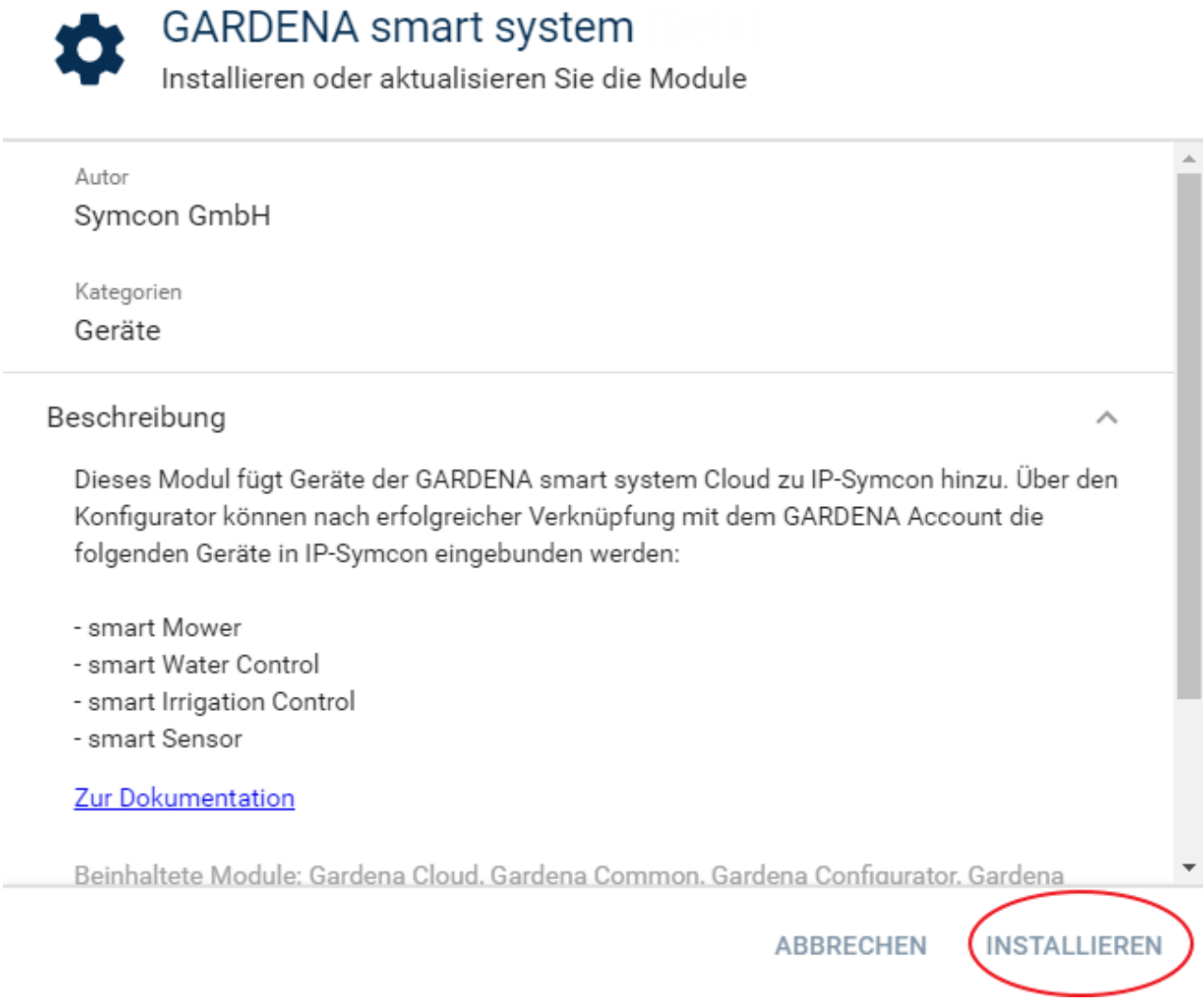


Die Verwendung des GARDENA-Moduls erfordert eine aktivierte Verbindung zum Connect-Dienst, welche wiederum eine laufende Subskription erfordert.

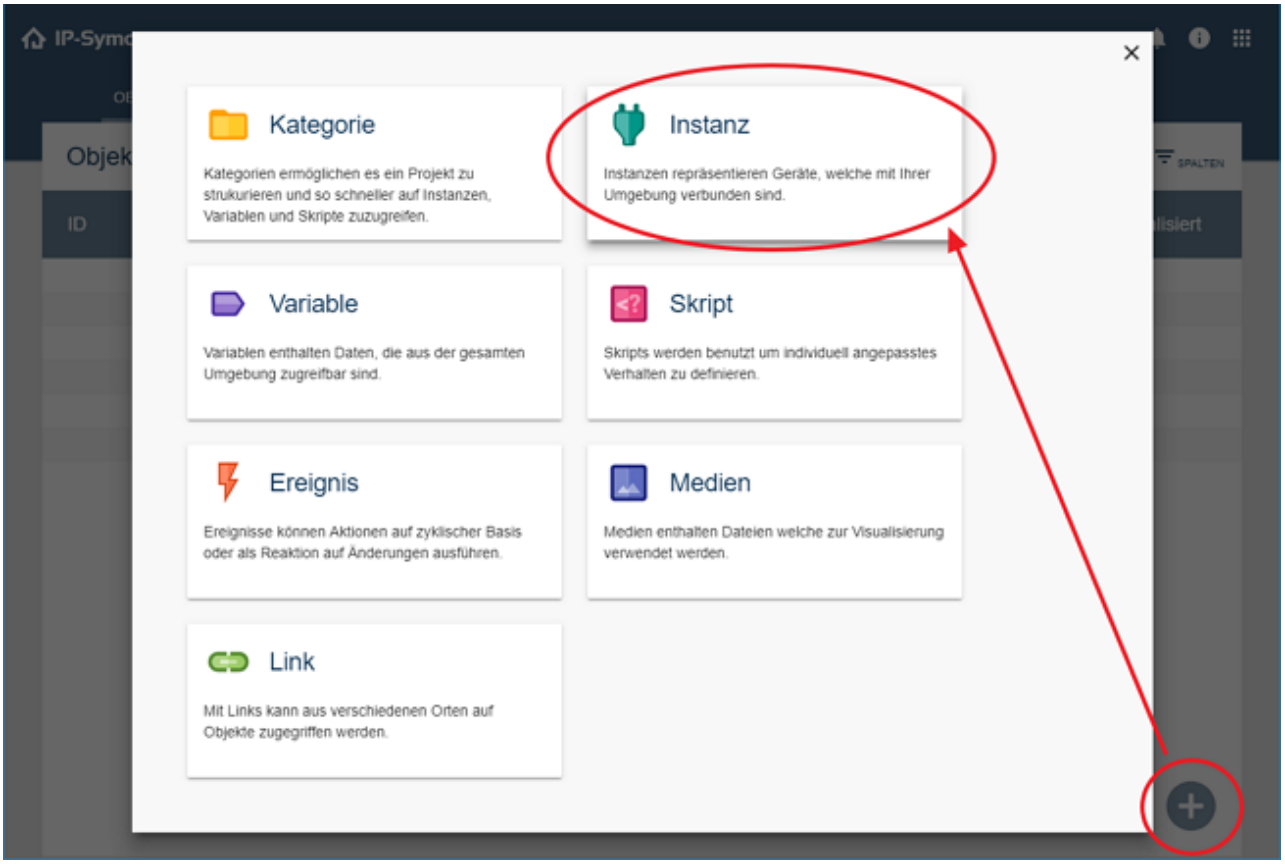
### GARDENA KONFIGURATOR INSTANZ ERSTELLEN

Damit das GARDENA-Modul verwendet werden kann, muss dieses zuerst über den Module Store installiert werden. Hierfür muss der Module Store geöffnet werden. Dieser befindet sich im oberen rechten Bereich. Über das Suchfeld kann via "Gardena smart system" das Modul Gardena smart system gefunden werden. Beim öffnen des gefundenen Modules, kann im folgenden Dialog die Installation des Moduls via "Installieren"-Knopf angestoßen werden.

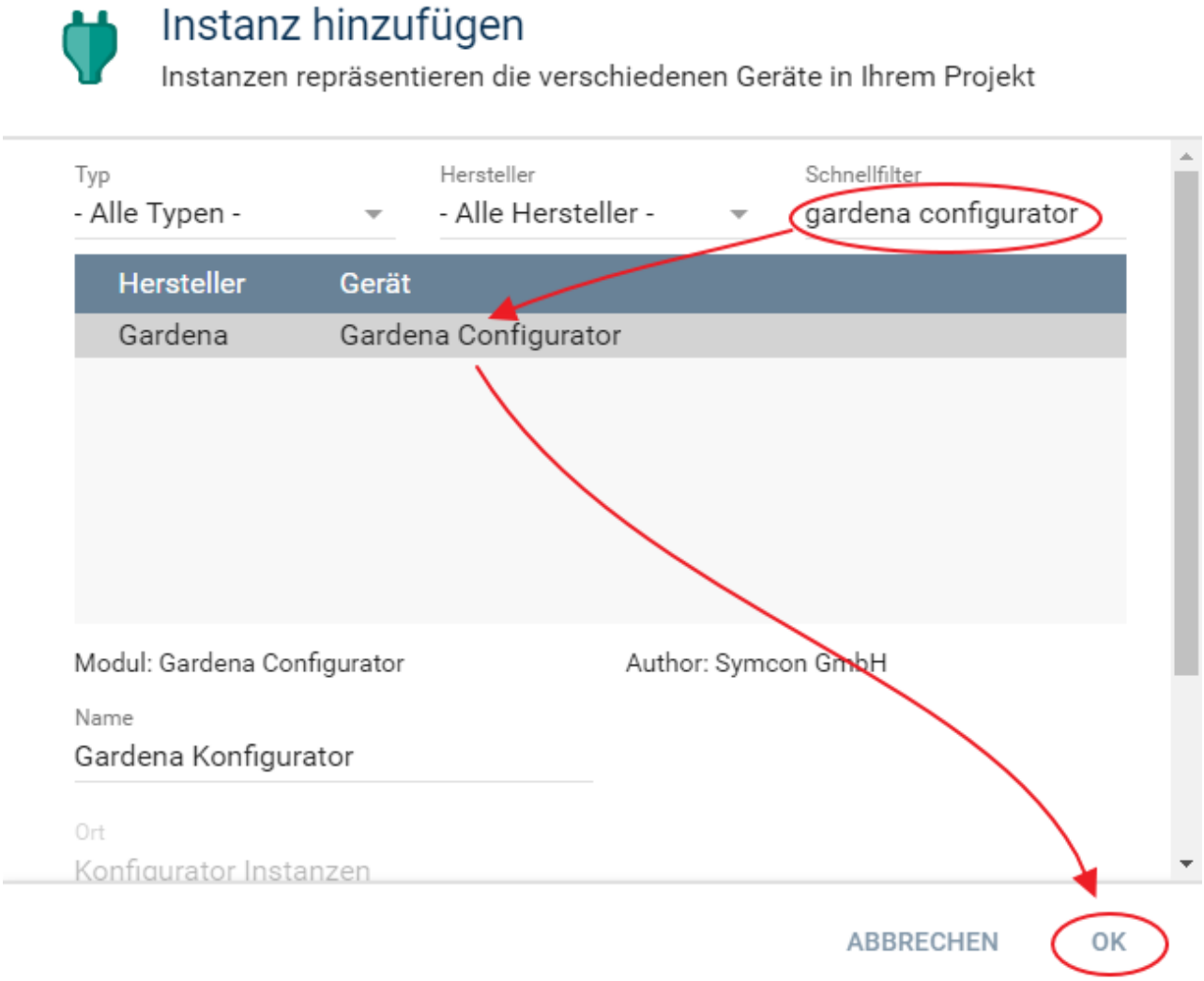

IP-Symcon
OBJEKTBAUM
MELDUNGEN
Module Store (Beta) > Suche
Suche
Gardena smart system
GARDENA smart system
Dieses Modul fügt Geräte der GARDENA smart system Cloud zu IP-Symcon hinzu. Über den Konfigurator können nach erfolgreicher Verknüpfung mit dem GARDENA Account die folgenden Geräte in IP-Symcon eingebunden werden: - smart Mower - smart Water Control - smart Irrigation Control - smart Sensor




Im Anschluss öffnet sich der Hinzufügendialog zum Erstellen einer Gardena Konfigurator Instanz.
Soll die Instanz manuell hinzugefügt werden muss eine Instanz vom Gardena Konfigurator erstellt werden. Dazu muss zuerst der Objektbaum geöffnet werden. In diesem den Hinzufügen-Button "+" unten rechts betätigen und Instanz auswählen.



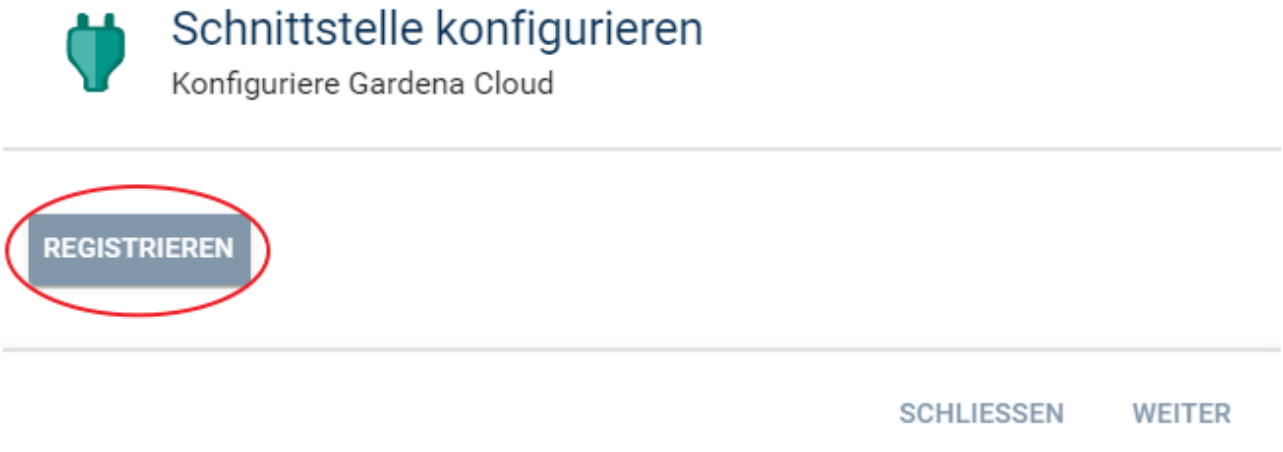

Mithilfe der Schnellsuche kann der "Gardena Configurator" vom Hersteller "Gardena" gefunden werden. Der Ort sollte nicht verändert werden, der Name kann beliebig gewählt werden. Abschließend mit "OK" bestätigen.




Nach der Erstellung der Konfigurator-Instanz öffnet sich diese automatisch und kann eingerichtet werden.

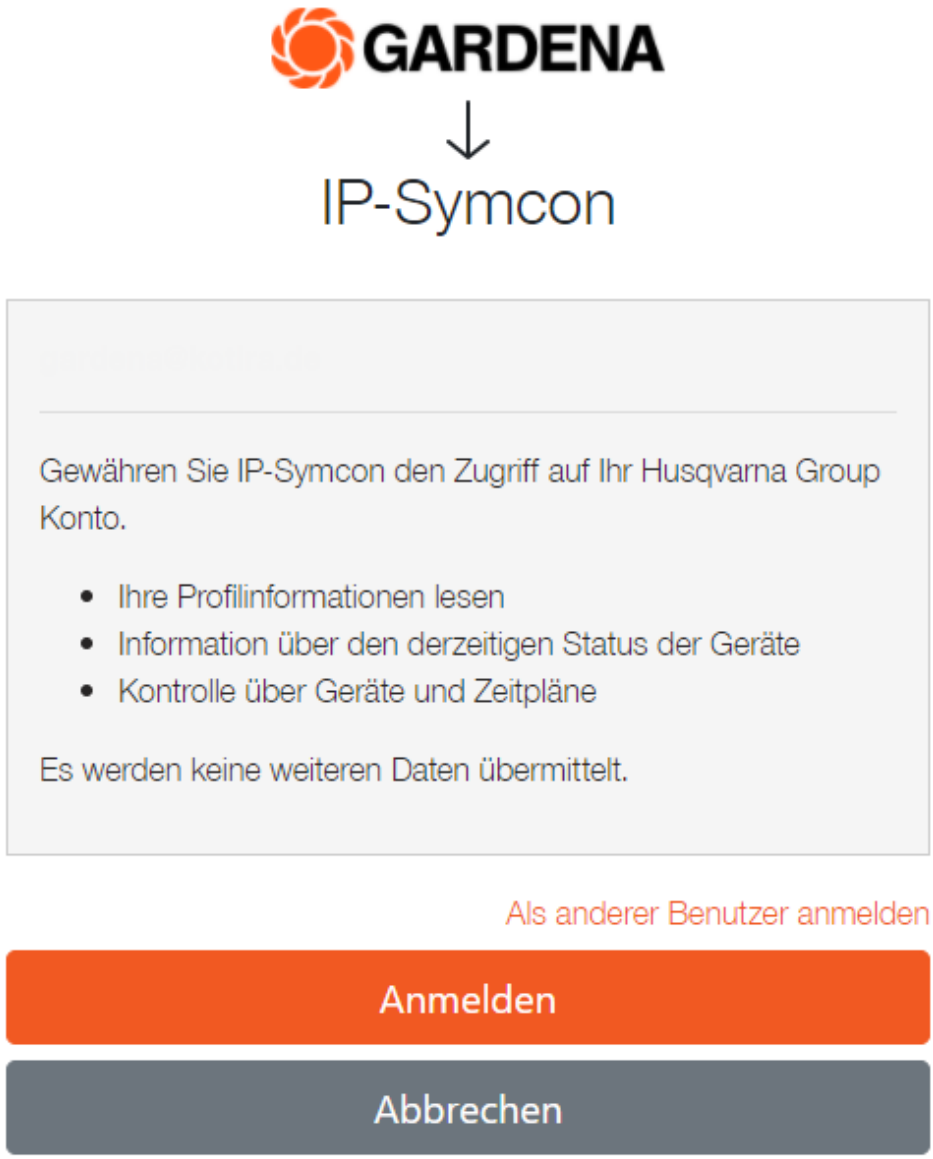
## GARDENA KONFIGURATOR-INSTANZ EINRICHTEN

Bevor Geräte erstellt werden können, muss sich mit einem Klick auf Registrieren registriert werden, um IP-Symcon mit dem Husqvarna Group Konto zu verbinden.

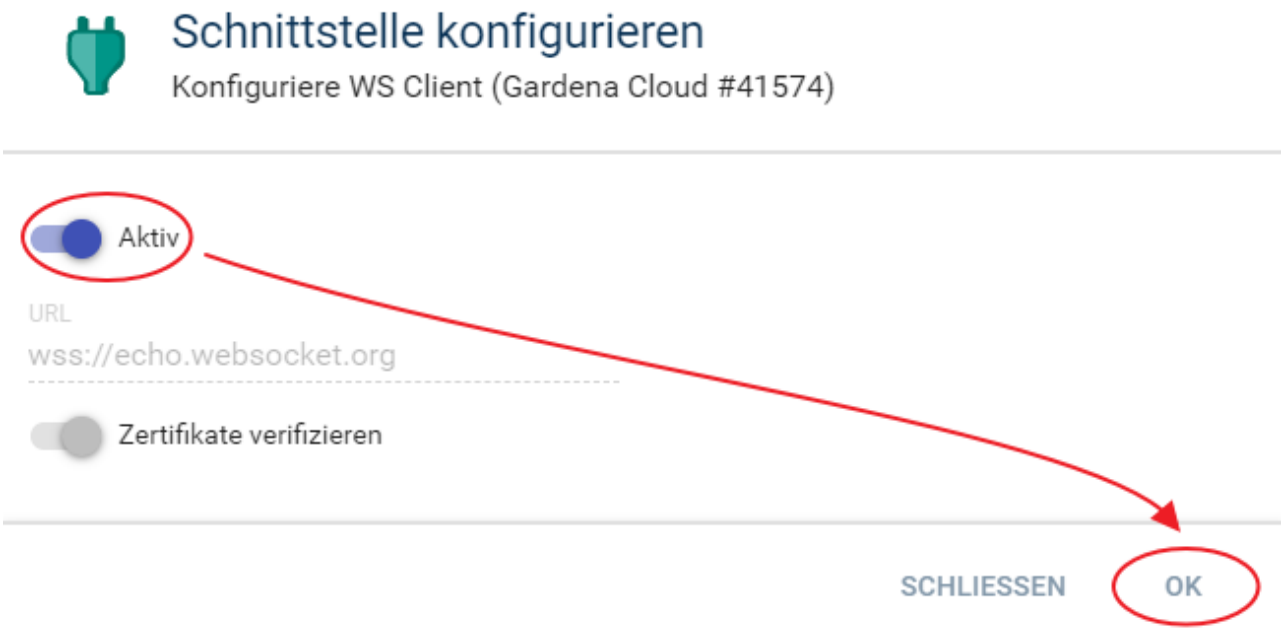


Es muss sich mit dem Husqvarna Group Konto angemeldet werden, um IP-Symcon die nötigen Berechtigungen einzuräumen.

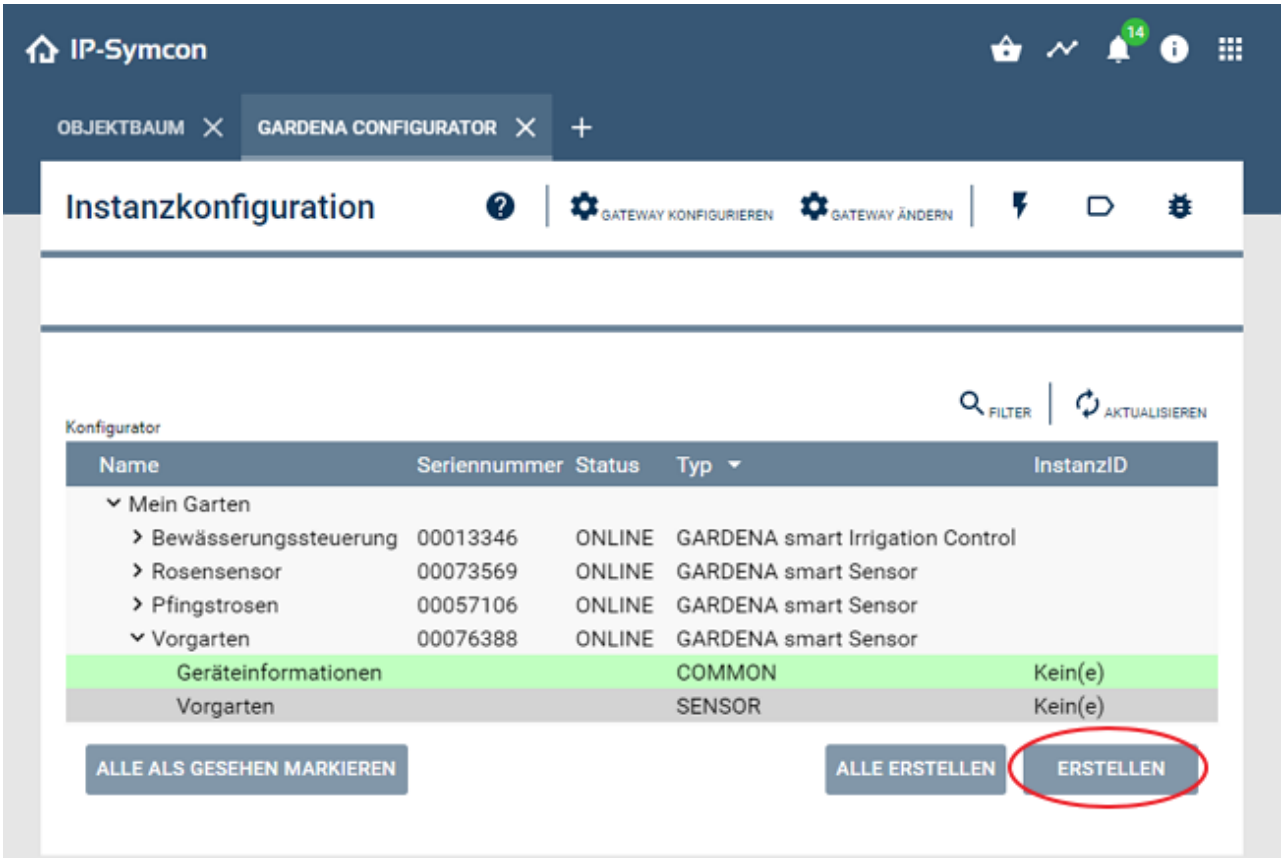





Daraufhin kann der "Schnittstelle Konfigurieren"-Dialog mit Weiter bestätigt werden. Nun muss noch der erstellte WebSocket Client aktiviert werden.




Jetzt öffnet sich der Konfigurator und es können alle erkannten Geräte mit einem Klick auf "Erstellen" erstellt werden.




## PROBLEMBEHANDLUNG

### VARIABLEN WERDEN NICHT MEHR AKTUALISIERT

Die Verbindung zum Husqvarna Dienst kann in der Gardena Cloud Instanz mit einem Klick auf 'WebSocket zurücksetzten' erneut hergestellt werden.

## UNTERSTÜTZTE GERÄTE

smart Irrigation Control
smart Water Control
smart Pump
smart Sensor
smart Mower
smart Power Socket

# smart Irrigation Control

Benötigt: IP-Symcon >= 6.0



Da die Gardena API die in der App erstellten Zeitpläne nicht übermittelt, können Bewässerungspläne mithilfe des Wochenplans und der bereitgestellten Aktionen umgesetzt werden.

## BESCHREIBUNG

Das Gerät smart Irrigation Control bietet in IP-Symcon verschieden Services, welche es zum Beispiel ermöglichen den Status des Gerätes zu erkennen oder die verschieden Ventile zu schalten.

## SERVICES

Für jeden Service kann mit dem Konfigurator eine Instanz erstellt werden.
In der Instanzkonfiguration kann der Schalter 'Letzte Übertragung anzeigen' aktiviert werden um für jeden Wert, wenn verfügbar, eine Variable zu erstellen, welche den Zeitpunkt der letzten Datenübertragung enthält.

| SERVICE | BESCHREIBUNG |
|---|---|
| Allgemein | Enthält generelle Informationen über das Gerät |
| Ventil | Repräsentiert eines der 6 Ventil der Bewässerungssteuerung |
| Hauptventil | Das Hauptventil mit dem alle Ventile geschlossen werden können |

### ALLGEMEIN

Die folgenden Statusvariablen werden für diese Instanz erstellt

| VARIABLE | TYP | BESCHREIBUNG |
|---|---|---|
| Batterie Status | String | Der Status der Batterie des Gerätes (Dieses Gerät hat keine Batterie) |
| RF Link Level | Integer | Die Qualität der Funkverbindung in Prozent |
| RF Link Status | String | Der Status der Funkverbindung des Gerätes |

### VENTIL

Die folgenden Statusvariablen werden für diese Instanz erstellt

| VARIABLE | TYP | BESCHREIBUNG |
|---|---|---|
| Aktivität | String | Die aktuelle Aktivität des Gerätes |
| Verbleibend | Integer | Die verbleibende Zeit in Sekunden bis das Ventil geschlossen wird |
| Fehlercode | String | Der letzte übertragene Fehler |
| Status | String | Der aktuelle Zustand des Gerätes |
| Öffnungsdauer | Integer | Die Dauer für die das Ventil manuell geöffnet werden soll |
| Aktion | String | Ermöglicht es das Ventil für eine bestimmte Öffnungsdauer zu öffnen oder zu schließen |

Die folgenden Aktionen sind für diese Instanz verfügbar

| AKTION | PARAMETER | BESCHREIBUNG |
|---|---|---|
| Ventil öffnen | Dauer (Minuten) | Ermöglicht es das Ventil für eine bestimmte dauer zu öffnen |
| Ventil schließen | (keine) | Ermöglicht das Ventil zu schließen |

## HAUPTVENTIL

Die folgenden Statusvariablen werden für diese Instanz erstellt

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Fehlercode | String | Der letzte übertragene Fehler |
| Status | String | Der aktuelle Zustand des Gerätes |
| Aktion | String | Schließt alle Ventile |

Die folgenden Aktionen sind für diese Instanz verfügbar

| AKTION | PARAMETER | BESCHREIBUNG |
|---|---|---|
| Hauptventil schließen | (keine) | Ermöglicht es das Hauptventil zu schließen |

# smart SILENO Mower

Benötigt: IP-Symcon >= 6.0

## BESCHREIBUNG

Das Gerät smart SILENO Mower bietet in IP-Symcon verschieden Services, welche es zum Beispiel ermöglichen die aktuelle Aktivität des Gerätes zu erkennen oder den Mähvorgang zu starten oder stoppen.

## SERVICES

Für jeden Service kann mit dem Konfigurator eine Instanz erstellt werden.
In der Instanzkonfiguration kann der Schalter 'Letzte Übertragung anzeigen' aktiviert werden um für jeden Wert, wenn verfügbar, eine Variable zu erstellen, welche den Zeitpunkt der letzten Datenübertragung enthält.

| SERVICE | BESCHREIBUNG |
| --- | --- |
| Allgemein | Enthält generelle Informationen über das Gerät |
| Mäher | Repräsentiert den SILENO Mäher |

### ALLGEMEIN

Die folgenden Statusvariablen werden für diese Instanz erstellt
| Variable | Typ | Beschreibung | |
| ---------- | ------- | ---------- |
| Batterie Status | String | Der Status der Batterie des Gerätes |
| RF Link Level | Integer | Die Qualität der Funkverbindung in Prozent |
| RF Link Status | String | Der Status der Funkverbindung des Gerätes |

### MÄHER

Die folgenden Statusvariablen werden für diese Instanz erstellt

| VARIABLE | TYP | BESCHREIBUNG |
| --- | --- | --- |
| Aktivität | String | Die aktuelle Aktivität des Gerätes |
| Betriebsstunden | Integer | Die Zeit in Stunden welche der Mäher schon aktiv ist |
| Fehlercode | String | Der letzte übertragene Fehler |
| Status | String | Der aktuelle Zustand des Gerätes |
| Mähdauer | Integer | Die Zeit in Minuten für manuell gemäht werden soll |
| Mähen | String | Ermöglicht den Mäher für eine bestimmte Mähdauer oder nach dem Zeitplan mähen zu lassen |
| Parken | String | Erlaubt es den Mäher zur Ladestation fahren zu lassen |

Die folgenden Aktionen sind für diese Instanz verfügbar

| AKTION | PARAMETER | BESCHREIBUNG |
|---|---|---|
| Automatisch mähen | (keine) | Lässt den Mähe nach Zeitplan mähen |
| Manuell mähen | Dauer (Minuten) | Lässt den Mäher für die festgelegte Dauer mähen |
| Parken bis zur nächsten Aufgabe | (keine) | Lässt den Mäher zur Ladestation fahren und bis zur nächsten Aufgabe warten |
| Parken und Zeitplan ignorieren | (keine) | Lässt den Mäher zur Ladestation fahren und den Zeitplan ignorieren |

# smart Power Socket

Benötigt: IP-Symcon >= 6.0

## BESCHREIBUNG

Das Gerät smart Power Socket bietet in IP-Symcon verschieden Services, welche es zum Beispiel ermöglichen den Status des Gerätes zu erkennen oder den Zwischenstecker zu schalten.

## SERVICES

Für jeden Service kann mit dem Konfigurator eine Instanz erstellt werden.
In der Instanzkonfiguration kann der Schalter 'Letzte Übertragung anzeigen' aktiviert werden um für jeden Wert, wenn verfügbar, eine Variable zu erstellen, welche den Zeitpunkt der letzten Datenübertragung enthält.

| SERVICE | BESCHREIBUNG |
|---|---|
| Allgemein | Enthält generelle Informationen über das Gerät |
| Zwischenstecker | Repräsentiert den Zwischenstecker, welcher geschaltet werden kann |

### ALLGEMEIN

Die folgenden Statusvariablen werden für diese Instanz erstellt

| VARIABLE | TYP | BESCHREIBUNG |
|---|---|---|
| Batterie Status | String | Der Status der Batterie des Gerätes (Dieses Gerät hat keine Batterie) |
| RF Link Level | Integer | Die Qualität der Funkverbindung in Prozent |
| RF Link Status | String | Der Status der Funkverbindung des Gerätes |

### ZWISCHENSTECKER

Die folgenden Statusvariablen werden für diese Instanz erstellt

| VARIABLE | TYP | BESCHREIBUNG |
|---|---|---|
| Aktivität | String | Die aktuelle Aktivität des Gerätes |
| Verbleibend | Integer | Die verbleibende Zeit in Sekunden bis der Zwischenstecker ausgeschaltet wird |
| Fehlercode | String | Der letzte übertragene Fehler |
| Status | String | Der aktuelle Zustand des Gerätes |
| Aktivitätsdauer | Integer | Die Dauer für die der Zwischenstecker manuell angeschaltet werden soll |
| Aktion | String | Die Aktion welche ausgeführt werden soll |

Die folgenden Aktionen sind für diese Instanz verfügbar

| AKTION | PARAMETER | BESCHREIBUNG |
|---|---|---|
| Zwischenstecker einschalten | Dauer (Minuten) | Ermöglicht es den Zwischenstecker für eine definierte Dauer einzuschalten |
| Zwischenstecker dauerhaft einschalten | (keine) | Ermöglicht es den Zwischenstecker dauerhaft einzuschalten |
| Zwischenstecker ausschalten | (keine) | Ermöglicht es den Zwischenstecker auszuschalten |

# smart Pressure Pump

Benötigt: IP-Symcon >= 6.0

## BESCHREIBUNG

Das Gerät smart Pump bietet in IP-Symcon verschieden Services, welche es zum Beispiel ermöglichen den Status des Gerätes zu erkennen oder das Ventil zu schalten.

## SERVICES

Für jeden Service kann mit dem Konfigurator eine Instanz erstellt werden.
In der Instanzkonfiguration kann der Schalter 'Letzte Übertragung anzeigen' aktiviert werden um für jeden Wert, wenn verfügbar, eine Variable zu erstellen, welche den Zeitpunkt der letzten Datenübertragung enthält.

| SERVICE | BESCHREIBUNG |
|---|---|
| Allgemein | Enthält generelle Informationen über das Gerät |
| Ventil | Repräsentiert das Ventil der Pumpe |
| Hauptventil | Das Hauptventil mit dem alle Ventile geschlossen werden können |

### ALLGEMEIN

Die folgenden Statusvariablen werden für diese Instanz erstellt

| VARIABLE | TYP | BESCHREIBUNG |
|---|---|---|
| Batterie Status | String | Der Status der Batterie des Gerätes |
| RF Link Level | Integer | Die Qualität der Funkverbindung in Prozent |
| RF Link Status | String | Der Status der Funkverbindung des Gerätes |

### VENTIL

Die folgenden Statusvariablen werden für diese Instanz erstellt

| VARIABLE | TYP | BESCHREIBUNG |
|---|---|---|
| Aktivität | String | Die aktuelle Aktivität des Gerätes |
| Verbleibend | Integer | Die verbleibende Zeit in Sekunden bis das Ventil geschlossen wird |
| Fehlercode | String | Der letzte übertragene Fehler |
| Status | String | Der aktuelle Zustand des Gerätes |
| Öffnungsdauer | Integer | Die Dauer für die das Ventil manuell geöffnet werden soll |
| Aktion | String | Ermöglicht es das Ventil für eine bestimmte Öffnungsdauer zu öffnen oder zu schließen |

Die folgenden Aktionen sind für diese Instanz verfügbar

| AKTION | PARAMETER | BESCHREIBUNG |
| --- | --- | --- |
| Ventil öffnen | Dauer (Minuten) | Ermöglicht es das Ventil für eine bestimmte dauer zu öffnen |
| Ventil schließen | (keine) | Ermöglicht das Ventil zu schließen |

## HAUPTVENTIL

Die folgenden Statusvariablen werden für diese Instanz erstellt

| NAME | TYP | BESCHREIBUNG |
| --- | --- | --- |
| Fehlercode | String | Der letzte übertragene Fehler |
| Status | String | Der aktuelle Zustand des Gerätes |
| Aktion | String | Schließt alle Ventile |

Die folgenden Aktionen sind für diese Instanz verfügbar

| AKTION | PARAMETER | BESCHREIBUNG |
| --- | --- | --- |
| Hauptventil schließen | (keine) | Ermöglicht es das Hauptventil zu schließen |

# smart Sensor

Benötigt: IP-Symcon >= 6.0

## BESCHREIBUNG

Das Gerät smart Sensor bietet in IP-Symcon verschieden Services, welche es zum Beispiel ermöglichen die unterschiedlichen Werte der Sensoren, wie etwa die Bodenfeuchte oder die Lichtintensität, auszulesen.

## SERVICES

Für jeden Service kann mit dem Konfigurator eine Instanz erstellt werden.
In der Instanzkonfiguration kann der Schalter 'Letzte Übertragung anzeigen' aktiviert werden um für jeden Wert, wenn verfügbar, eine Variable zu erstellen, welche den Zeitpunkt der letzten Datenübertragung enthält.

| SERVICE | BESCHREIBUNG |
|---|---|
| Allgemein | Enthält generelle Informationen über das Gerät |
| Sensor | Repräsentiert die Werte des Sensors |

### ALLGEMEIN

Die folgenden Statusvariablen werden für diese Instanz erstellt

| VARIABLE | TYP | BESCHREIBUNG |
|---|---|---|
| Batterie Status | String | Der Status der Batterie des Gerätes |
| RF Link Level | Integer | Die Qualität der Funkverbindung in Prozent |
| RF Link Status | String | Der Status der Funkverbindung des Gerätes |

### SENSOR

Die folgenden Statusvariablen werden für diese Instanz erstellt

| VARIABLE | TYP | BESCHREIBUNG |
|---|---|---|
| Bodenfeuchte | Integer | Die vom Sensor gemessene Bodenfeuchte in Prozent |
| Bodentemperatur | Float | Die vom Sensor gemessene Bodentemperatur in °C |
| Lichtintensität | Float | Die vom Sensor gemessene Beleuchtungsstärke in lux |
| Umgebungstemperatur | Float | Die vom Sensor gemessene Umgebungstemperatur in °C |

# smart Water Control

Benötigt: IP-Symcon >= 6.0

## BESCHREIBUNG

Das Gerät smart Water Control bietet in IP-Symcon verschieden Services, welche es zum Beispiel ermöglichen den Status des Gerätes zu erkennen oder das Ventil zu schalten.

## SERVICES

Für jeden Service kann mit dem Konfigurator eine Instanz erstellt werden.
In der Instanzkonfiguration kann der Schalter 'Letzte Übertragung anzeigen' aktiviert werden um für jeden Wert, wenn verfügbar, eine Variable zu erstellen, welche den Zeitpunkt der letzten Datenübertragung enthält.

| SERVICE | BESCHREIBUNG |
|---|---|
| Allgemein | Enthält generelle Informationen über das Gerät |
| Ventil | Repräsentiert das Ventil der Bewässerungssteuerung |
| Hauptventil | Das Hauptventil mit dem alle Ventile geschlossen werden können |

### ALLGEMEIN

Die folgenden Statusvariablen werden für diese Instanz erstellt

| VARIABLE | TYP | BESCHREIBUNG |
|---|---|---|
| Batterie Status | String | Der Status der Batterie des Gerätes |
| RF Link Level | Integer | Die Qualität der Funkverbindung in Prozent |
| RF Link Status | String | Der Status der Funkverbindung des Gerätes |

### VENTIL

Die folgenden Statusvariablen werden für diese Instanz erstellt

| VARIABLE | TYP | BESCHREIBUNG |
|---|---|---|
| Aktivität | String | Die aktuelle Aktivität des Gerätes |
| Verbleibend | Integer | Die verbleibende Zeit in Sekunden bis das Ventil geschlossen wird |
| Fehlercode | String | Der letzte übertragene Fehler |
| Status | String | Der aktuelle Zustand des Gerätes |
| Öffnungsdauer | Integer | Die Dauer für die das Ventil manuell geöffnet werden soll |
| Aktion | String | Ermöglicht es das Ventil für eine bestimmte Öffnungsdauer zu öffnen oder zu schließen |

Die folgenden Aktionen sind für diese Instanz verfügbar

| AKTION | PARAMETER | BESCHREIBUNG |
| --- | --- | --- |
| Ventil öffnen | Dauer (Minuten) | Ermöglicht es das Ventil für eine bestimmte dauer zu öffnen |
| Ventil schließen | (keine) | Ermöglicht das Ventil zu schließen |

## HAUPTVENTIL

Die folgenden Statusvariablen werden für diese Instanz erstellt

| NAME | TYP | BESCHREIBUNG |
| --- | --- | --- |
| Fehlercode | String | Der letzte übertragene Fehler |
| Status | String | Der aktuelle Zustand des Gerätes |
| Aktion | String | Schließt alle Ventile |

Die folgenden Aktionen sind für diese Instanz verfügbar

| AKTION | PARAMETER | BESCHREIBUNG |
| --- | --- | --- |
| Hauptventil schließen | (keine) | Ermöglicht es das Hauptventil zu schließen |

# Geofency

Benötigt: IP-Symcon >= 4.2
Das Modul dient zum empfangen von Geofency Daten.

## FUNKTIONSUMFANG

Pro Gerät eine eigene Standortliste
Benutzername und Passwort Identifikation innerhalb von IP-Symcon.
Richtet automatisch den Webhook "/hook/geofency" ein.
Es wird empfohlen dies in Kombination mit dem Connect Modul zu nutzen.
Optional kann der aktuelle Standort von Geofency übermittelt werden

## VORAUSSETZUNGEN

Geofency App für Apple iOS

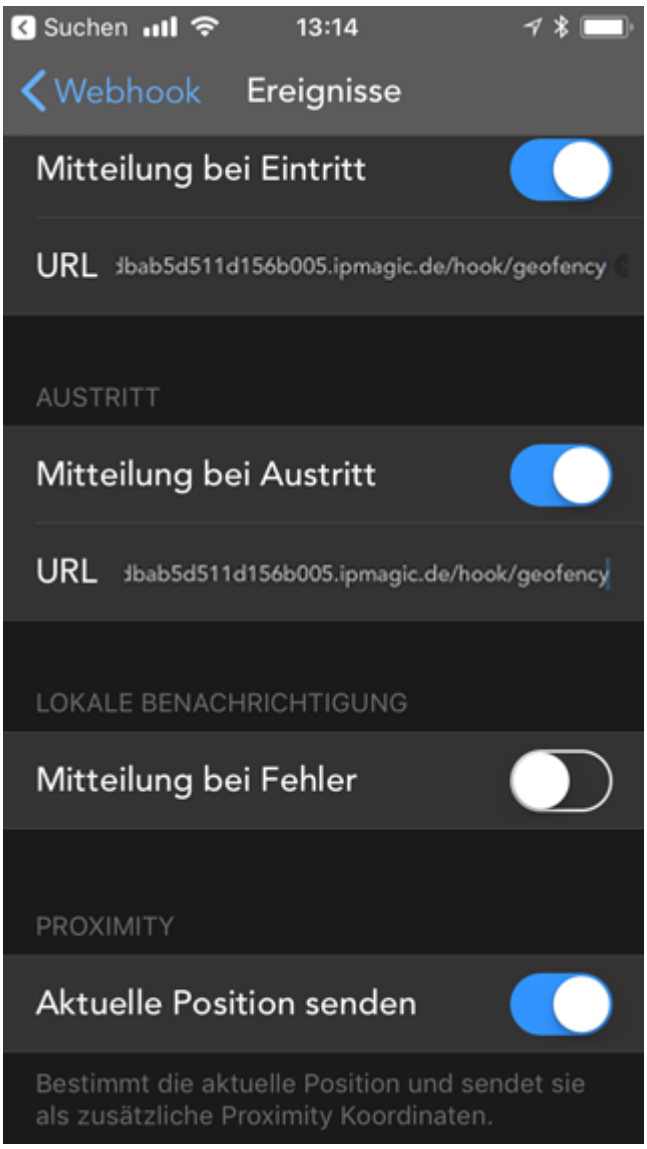
## SOFTWARE-INSTALLATION

Über den Module Store das Modul Geofency installieren.

## GEOFENCY KONFIGURATION

In der Geofency App mit + einen neuen Standort hinzufügen. Dann auf die 3 Punkte rechts klicken um das Menü zu öffnen. Hier ist Webhook auszuwählen.

### EVENT




Als URL wird jeweils die Domain eingetragen unter der IP-Symcon zu erreichen ist gefolgt von /hook/geofency Am einfachtsen ist es dort die IP-Symcon Connect Adresse einzutragen gefolgt von /hook/geofency
Optional kann Aktuelle Position senden aktiviert werden.

### POST FORMAT

Einstellung bleibt auf Default, JSON-enkodiert ist deaktiv und wird nicht ausgewählt

### AUTHENTIFIZIERUNG

| NAME | HTTP BASIC AUTHENTIFIZIERUNG |
|---|---|
| Username | Der Webhook Benutzername der dann in IP-Symcon hinterlegt wird |
| Passwort | Das Webhook Passwort das dann in IP-Symcon hinterlegt wird |

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Geofency'-Modul mithilfe des Schnellfilters gefunden

werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
| --- | --- |
| Benutzername | Benutzername, welcher in der Geofency App angegeben werden muss, um IP-Symcon Daten schicken zu können. |
| Passwort | Passwort, welches in der Geofency App angeben werden muss. |

*Werden diese Daten Leer gelassen kann jeder über den Hook Daten an IP-Symcon schicken.*

## STATUSVARIABLEN

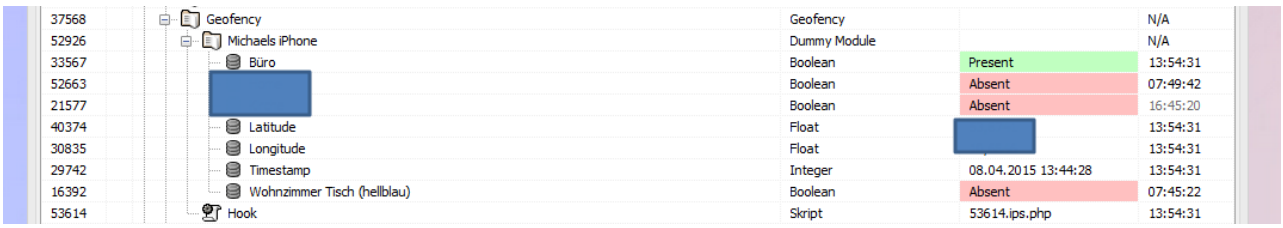
Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

Die Variablen werden anhand der Geräte ID und beim erstmaligen senden innerhalb des Geofency Moduls automatisch angelegt. Es können mehrere Geräte über einen Hook laufen. Jedes Gerät wird unter seiner eigenen "Kategorie" eingerichtet.

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Gerätename | Instanz (Dummy) | Dient als "Kategorie" in der sich alle überwachten Standorte, sowie der Zeitstempel und Längen-/ Breitengrad befinden. Wird pro Gerät erstellt. |
| Latitude | Float | Breitengrad (Mitte des Geofences) der letzten Aktivität |
| Longitude | Float | Längengrad (Mitte des Geofences) der letzten Aktivität |
| CurrentLatitude | Float | Proximity-Koordinaten, also die aktuellen Koordinaten bei Ein- oder Austritt (äusserer Umkreis des Geofences), falls „Aktuelle Position senden“ in der App beim Webhook gewählt wurde |
| CurrentLongitude | Float | Proximity-Koordinaten, also die aktuellen Koordinaten bei Ein- oder Austritt (äusserer Umkreis des Geofences), falls „Aktuelle Position senden“ in der App beim Webhook gewählt wurde |
| Direction | Integer | Azimutwinkel (Richtung in Grad) des aktuellen Ein-/ Austrittspunktes bezogen auf die Mitte des Geofence-Kreises, falls „Aktuelle Position senden“ in der App beim Webhook gewählt wurde |
| Distance | Float | Abstand (in Meter) des aktuellen Ein-/Austrittspunktes bezogen auf die Mitte des Geofence-Kreises, falls „Aktuelle Position senden“ in der App beim Webhook gewählt wurde |
| Orientation | Float | Himmelsrichtung (als Name) des aktuellen Ein-/ Austrittspunktes bezogen auf die Mitte des Geofence-Kreises, falls „Aktuelle Position senden“ in der App beim Webhook gewählt wurde |
| Timestamp | Integer | UnixTimestamp der letzten Aktivität. |
| Beispielstandort (Büro) | Boolean | Present oder Absent. Information wird von Gefency geliefert. |
| aktuelle Longitude | Float | Aktueller Breitengrad |
| aktuelle Latitude | Float | Aktueller Längengrad |
| Motion | Integer | aktueller Bewegungszustand wie Gehen, Autofahren etc. |
| WifiBSSD | String | BSSID des verbundenen WLANs |
| WifiSSID | String | SSID des verbundenen WLANs |

Beispiel:



# Heating Control

## BESCHREIBUNG

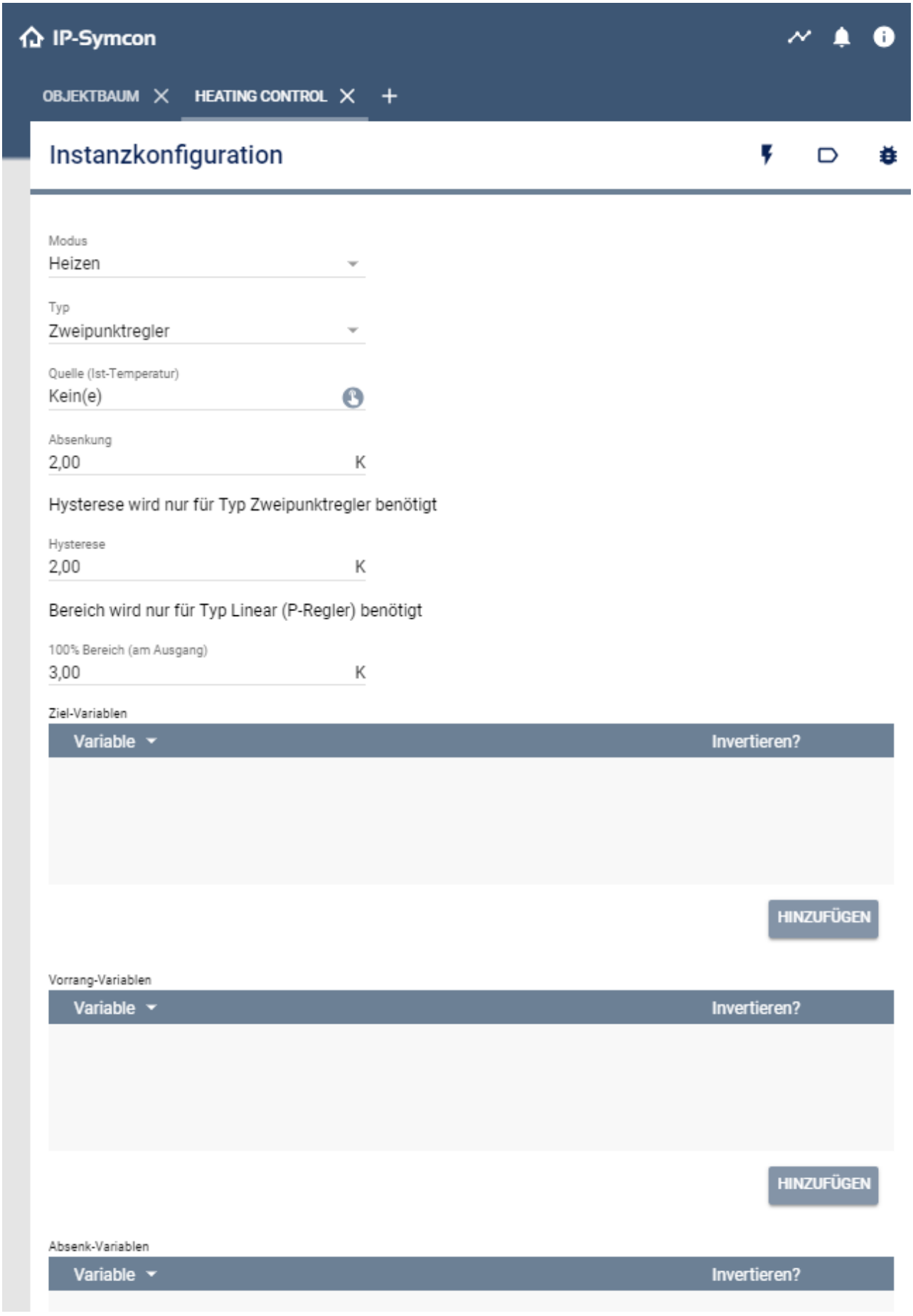
Das Heizungsmodul dient der Regelung von Temperaturen in Wohnräumen.
Es werden 2-Punkt-Regler und Linear-Regler (z.B. P-Regler) unterstützt.
Dazu werden Soll-Wert und Ist-Wert verglichen und der Aktor angesteuert.

## VORAUSSETZUNG

Den Ist-Wert erhält man über entsprechende Temperatursensoren, deren Wert als Variable IP-Symcon bekannt sein muss. Desweiteren muss die Heizquelle über einen Aktor schaltbar sein. Beispiele: Heizkörper mit einem thermischen Stellantrieb oder ein Heizlüfter und eine Funksteckdose.

## EINBINDUNG IN IP-SYMCON

Das "Heating Control" Modul (Hersteller: None) kann innerhalb der Verwaltungskonsole über den Objektbaum hinzugefügt werden und optional sollte ein sprechender Name (z.B. "Heizungsmodul Wohnzimmer") vergeben werden.




Skripte, die durch das Heating Control ausgeführt werden, beinhalten diese Systemvariablen

| VARIABLE | BESCHREIBUNG |
|---|---|
| Modus | Auswahl, ob das Heating Control Heizen oder Kühlen soll. |
| Typ | Auswahl, welche Art von Regler genutzt wird. |
| Quelle (Ist-Temperatur) | Variable, welche den Ist-Wert enthält. |
| Absenkung | Temperatur in °C, um die abgesenkt werden soll, wenn eine oder mehrere der "Absenk-Variablen" TRUE sind |
| Hysterese (Nur Zweipunktregler) | Temperaturdifferenz zwischen Einschalt- und Ausschaltpunkt; Beispiel: T-Soll = 20,0 Grad und Hysterese = 0,4; Der Aktor wird bei Unterschreiten von 19,8 Grad eingeschaltet und bei Überschreiten von 20,2 Grad ausgeschaltet. Eine Hysterese von 0 bewirkt ein “Toggeln” des Aktors bei jeder Aktualisierung der Ist-Variablen. |
| 100% Bereich (Nur P-Regler) | Beschreibt die Größe eines Bereichs vor dem Sollwert. In diesem wird eine linear abgestufte Steuerung berechnet; Beispiel Heizen: T-Soll = 20,0 Grad und 100% Bereich = 1; Der Aktor bleibt bei 19°C (Ist-Wert) auf 100%; Bei 19,3°C auf 70%; Bei 19,9°C auf 10%. Beispiel Kühlen: T-Soll = 10,0 Grad und 100% Bereich = 5; Der Aktor bleibt bei 15°C (Ist-Wert) auf 100%; Bei 14,0°C auf 80%; Bei 11,0°C auf 20%. |
| Ziel-Variablen | Hier werden die zu schaltenden Variablen eingetragen. Bei thermischen Stellantrieben, die bei stromlos geöffnet sind, muss “Invertieren” aktiviert werden. |
| Vorrang-Variablen | Können z.B. Status-Variablen von Fensterkontakten sein. Bei "True" bleibt der Ausgang des Reglers ausgeschaltet. |
| Absenk-Variablen | Können z.B. Präsenz-Variablen von Bewegung- oder Präsenz-Melder sein. Wenn sich in einem Raum keine Person aufhält, kann durch eine Temperatur-Absenkung so Energie gespart werden. |
| Sende Intervall | Falls ein (Funk)-Befehl nicht korrekt verarbeitet worden ist, kann dieser zyklisch wiederholt werden (nicht öfters als 15 Minuten). |
| Experteneinstellungen | Für Sonderfälle kann hier ein eigenes Skript ausgewählt werden. Es wird bei jedem Schaltvorgang des Reglers aufgerufen. Alle notwendigen Daten sind in den Systemvariablen hinterlegt. |
| Test Center | Um die Funktion des Reglers zu überprüfen, kann hier eine Soll-Temperatur eingetragen werden. Der Aktor muss dann wie gewünscht ein- oder ausschalten. |

| STATUSVARIABLEN | IDENT | BESCHREIBUNG |
|---|---|---|
| Absenkung | SetBackVariable | Bei True ist die Absenkung aktiv. |
| Heizen | HeatingVariable | Spiegelt den Ausgang des Reglers wieder. Ein Invertieren der Sende-Instanzen hat auf diese Variable keinen Einfluss. |
| Soll-Wert | TargetValueVariable | Enthält die Soll-Temperatur. |
| Vorrang | OverrideVariable | Bei True ist mindestens eine Vorrang-Variable ebenfalls True (Oder-Verknüpfung) und der Aktor bleibt ausgeschaltet. |

## TIPPS & TRICKS

Die Absenk-Funktion wird verwendet, um in unbewohnten Räumen Energie zu sparen.
Praxis-Beispiel:
Ein im Raum installierter Bewegungsmelder liefert für einige Sekunden in IP-Symcon eine Variablenänderung auf TRUE. Das folgende Skript verlängert diesen Impuls auf z.B. 15 Minuten und schreibt den Status in die Variable "Präsenz" (vorher anlegen!). Wenn innerhalb der gewünschten Zeit keine Bewegung im Raum mehr erkannt wird, wird die Variable auf FALSE gesetzt.
Diese kann nun unter den "Absenk-Variablen" hinzugefügt werden. Allerdings muss zusätzlich "Invertieren" aktiviert werden, da ja bei NICHT-Präsenz die Temperatur abgesenkt werden soll. Diese muss nun nur noch unter "Absenkung" eingetragen werden: z.B. 2 °C.
Ein geöffnetes Fenster mit einer "Vorrang-Variable" hat auch hier Vorrang und schaltet die Heizung komplett aus.

```
$id_prae = 12345 /*[Präsenz]*/; // anlegen!

if($_IPS['SENDER'] == "Variable"){
if($_IPS['VALUE'] == True){ //Timer starten
IPS_SetScriptTimer($_IPS['SELF'], 15 * 60);
$prae = GetValue($id_prae );
if($prae == False){
SetValue($id_prae, True);
}
}
}

if($_IPS['SENDER'] == "TimerEvent"){
IPS_SetScriptTimer($_IPS['SELF'], 0);
SetValue($id_prae , False);
}
```

## WEITERFÜHRENDE LINKS

Zweipunktregler: de.wikipedia.org/wiki/Zweipunktregler
P-Regler: de.wikipedia.org/wiki/Regler
Hysterese: de.wikipedia.org/wiki/Hysterese



Da zu jeder Zeit der PC (Regelung) ausfallen kann, muss zusätzlich eine Temperaturbegrenzung / Sicherheitsabschaltung installiert werden, um Personen- oder Sachschäden zu vermeiden.

# HC_TargetValue

```
boolean HC_TargetValue (int $InstanzID, float $Soll-Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Soll-Wert | Neuer Soll-Wert |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet über die Instanz mit der ID **InstanzID** einen Soll-Wert an das Heating-Modul.

## BEISPIEL

```
HC_TargetValue(12345, 22.5); //Auf 22.5°C
```

# Home Connect

Benötigt: IP-Symcon >= 6.0

## BESCHREIBUNG

Das Home Connect-Modul ermöglicht es Geräte der Hersteller Bosch, Siemens, Gaggenau und Neff, welche den Home Connect Dienst unterstützen, in IP-Symcon einzubinden.

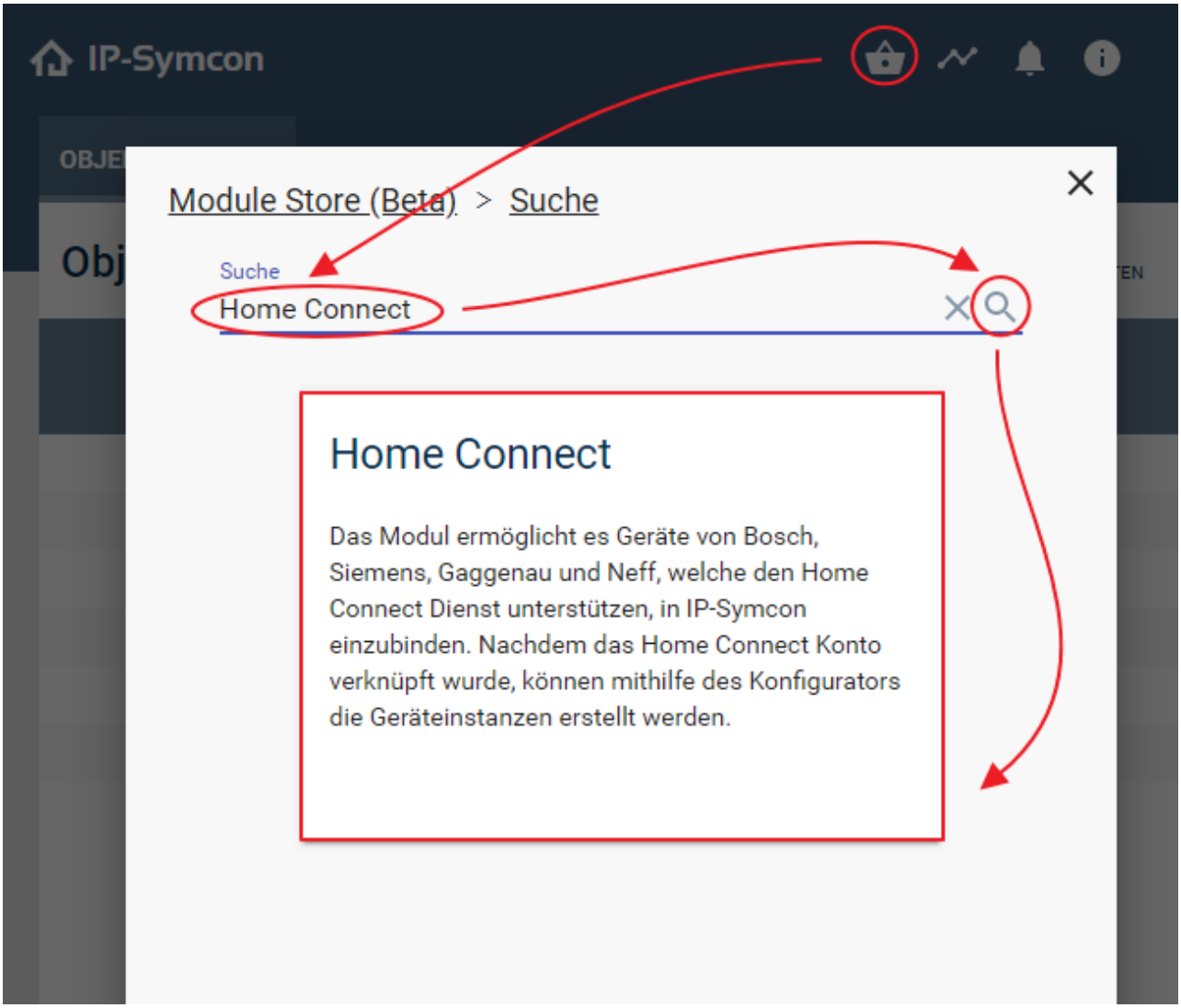
## EINBINDUNG IN IP-SYMCON

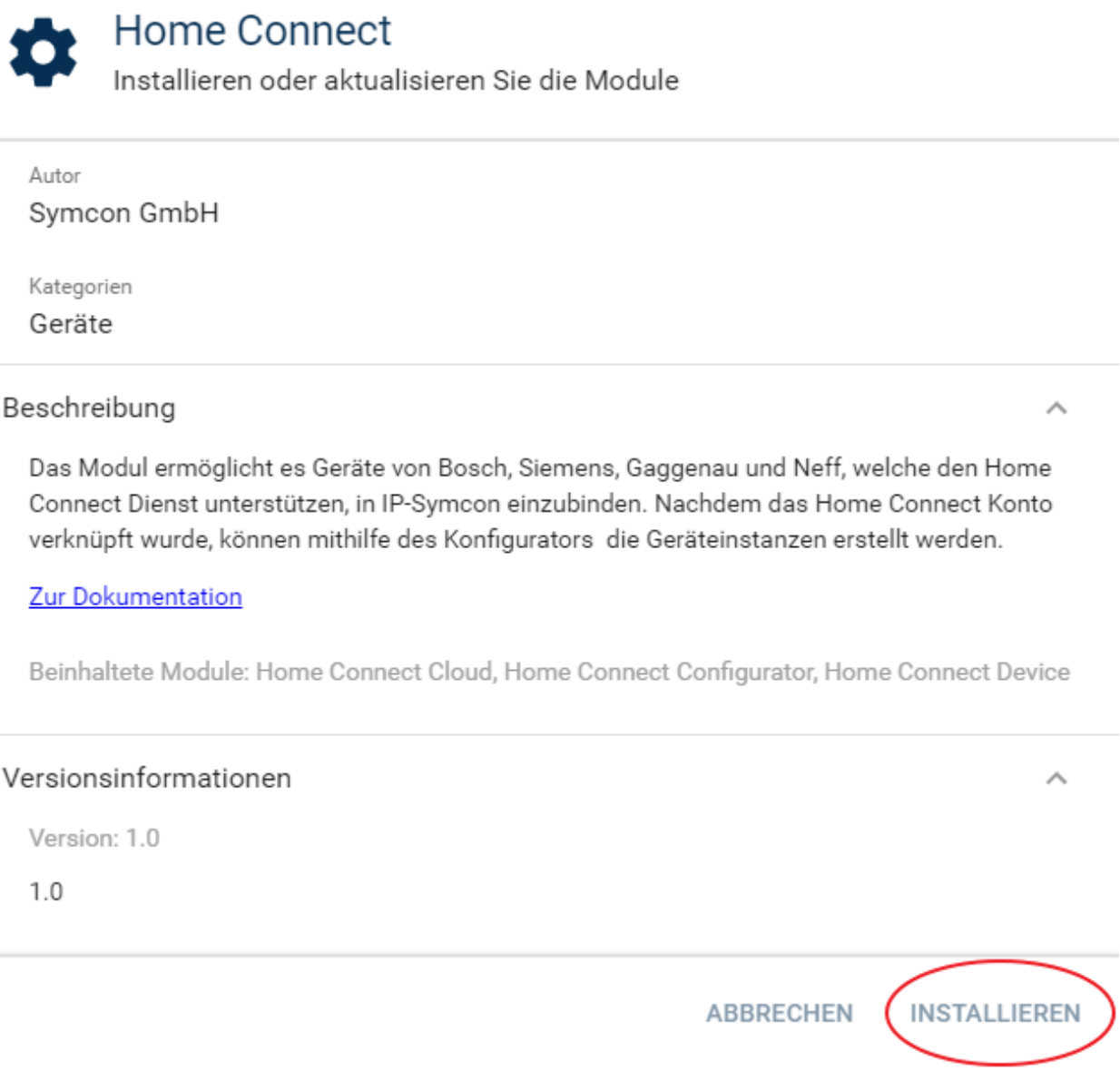


Die Verwendung des Home Connect-Moduls erfordert eine aktivierte Verbindung zum Connect-Dienst, welche wiederum eine laufende Subskription erfordert.

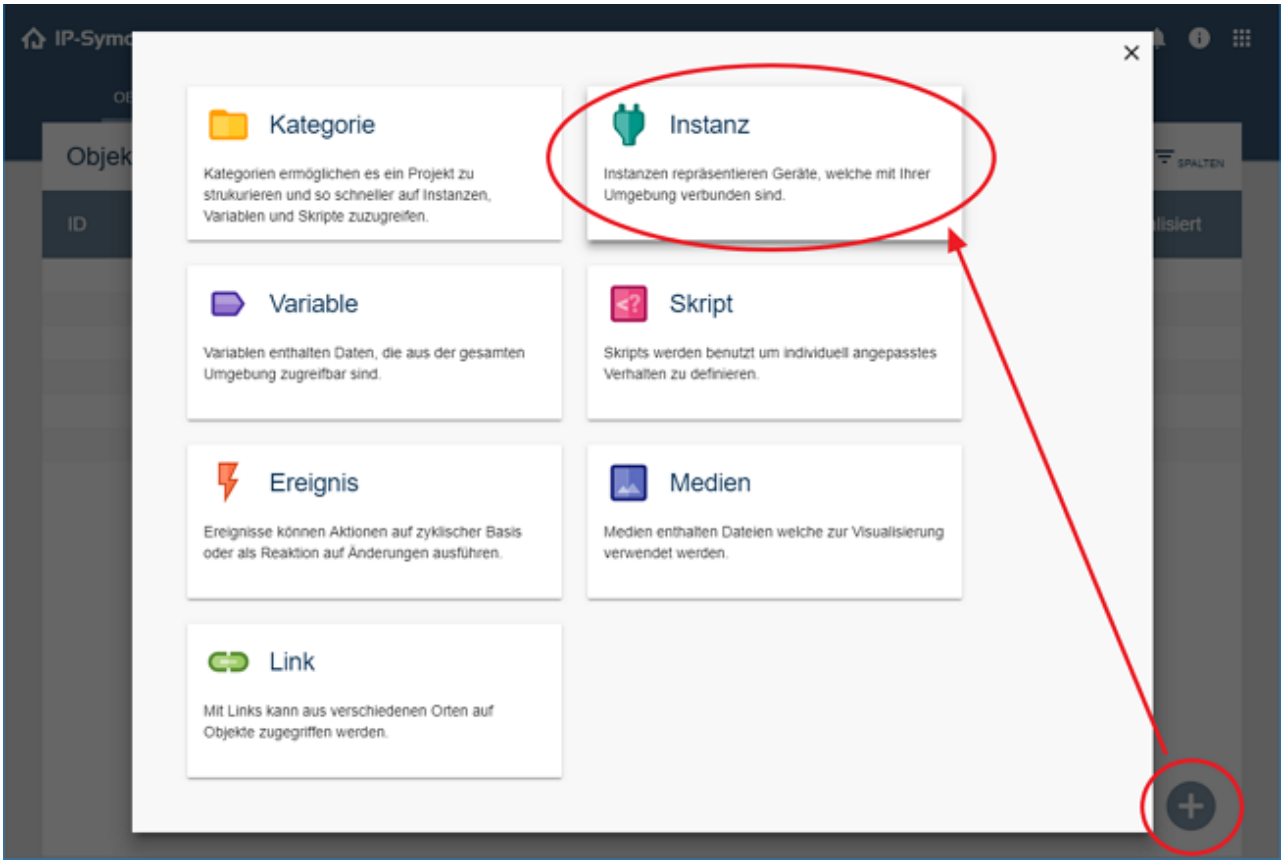
### HOME CONNECT KONFIGURATOR INSTANZ ERSTELLEN

Damit das Home Connect-Modul verwendet werden kann, muss dieses zuerst über den Module Store installiert werden. Hierfür muss der Module Store geöffnet werden. Dieser befindet sich im oberen rechten Bereich. Über das Suchfeld kann via "Home Connect" das Modul Home Connect gefunden werden. Beim öffnen des gefundenen Modules, kann im folgenden Dialog die Installation des Moduls via "Installieren"-Knopf angestoßen werden.

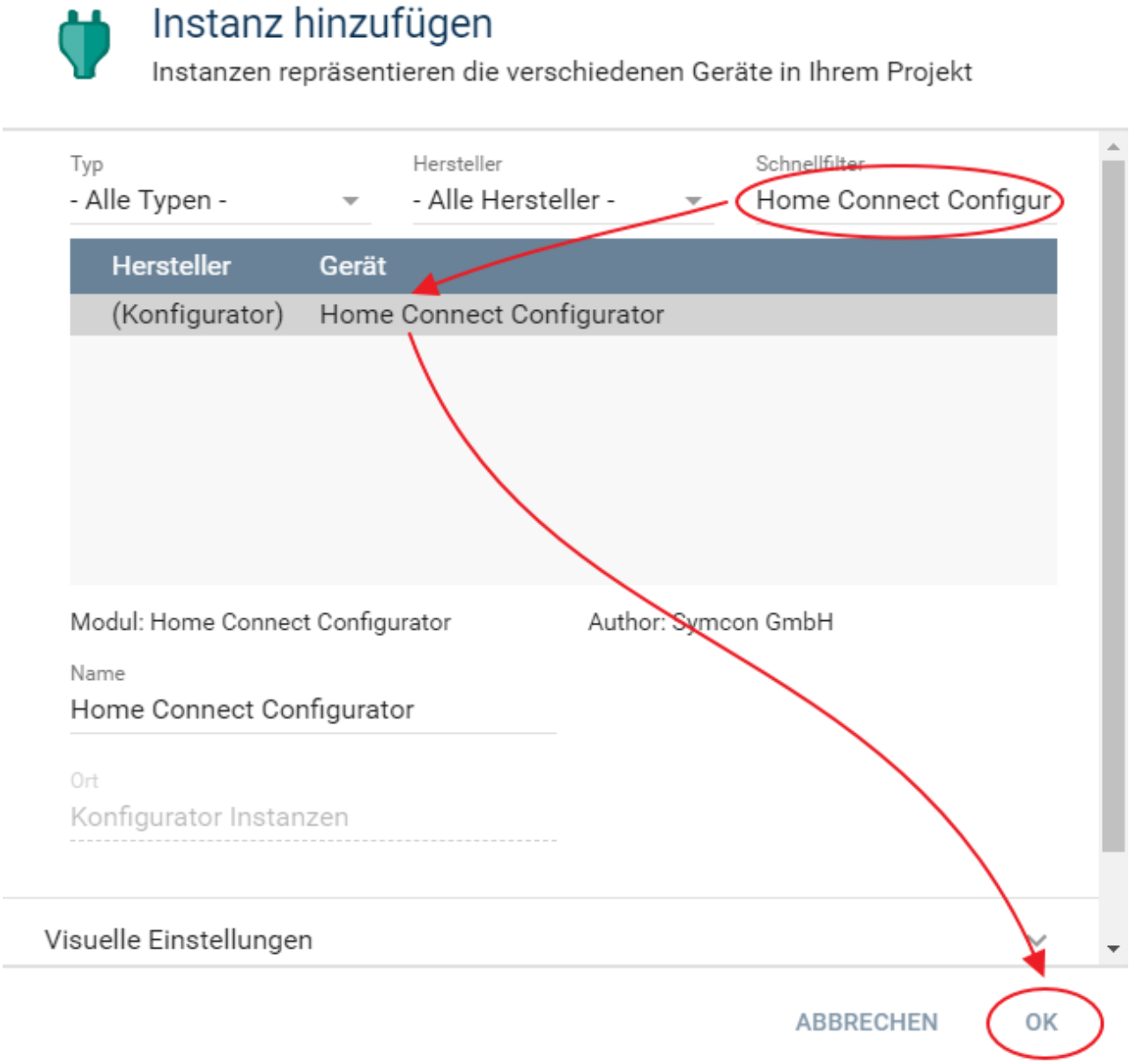

IP-Symcon
Module Store (Beta) > Suche
Suche
Home Connect
Home Connect
Das Modul ermöglicht es Geräte von Bosch, Siemens, Gaggenau und Neff, welche den Home Connect Dienst unterstützen, in IP-Symcon einzubinden. Nachdem das Home Connect Konto verknüpft wurde, können mithilfe des Konfigurators die Geräteinstanzen erstellt werden.




Im Anschluss öffnet sich der Hinzufügendialog zum Erstellen einer Home Connect Konfigurator Instanz.
Soll die Instanz manuell hinzugefügt werden muss eine Instanz vom Home Connect Konfigurator erstellt werden. Dazu muss zuerst der Objektbaum geöffnet werden. In diesem den Hinzufügen-Button "+" unten rechts betätigen und Instanz auswählen.



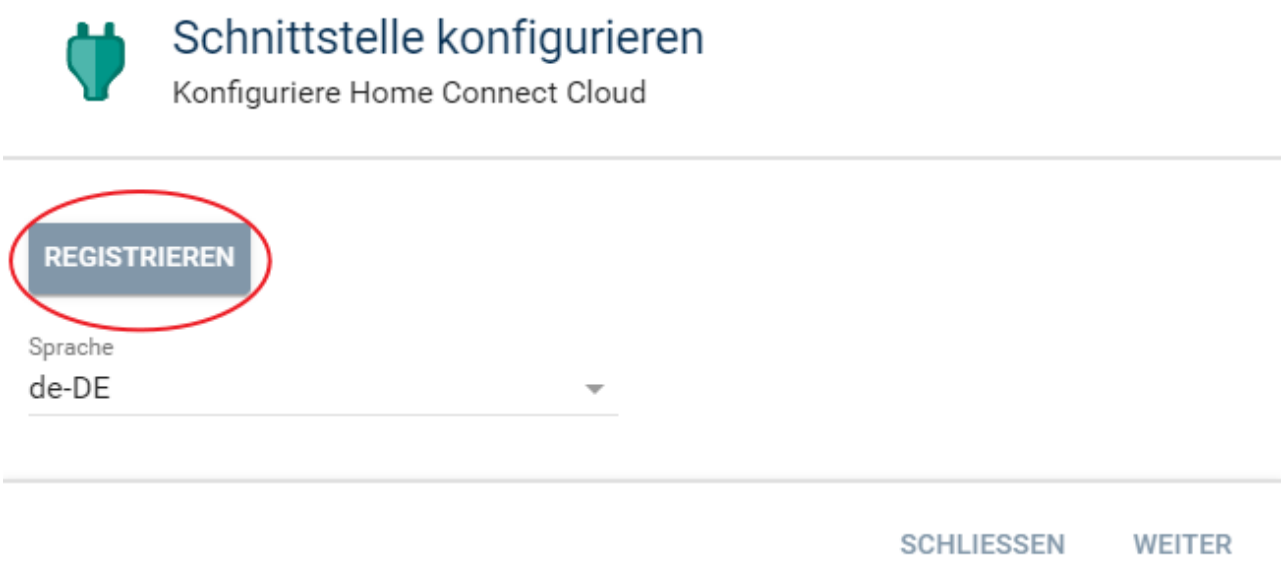

Mithilfe der Schnellsuche kann der "Home Connect Configurator" vom Hersteller "(Konfigurator)" gefunden werden. Der Ort sollte nicht verändert werden, der Name kann beliebig gewählt werden. Abschließend mit "OK" bestätigen.




Nach der Erstellung der Konfigurator-Instanz öffnet sich diese automatisch und kann eingerichtet werden.

## HOME CONNECT KONFIGURATOR-INSTANZ EINRICHTEN

Bevor Geräte erstellt werden können, muss sich mit einem Klick auf Registrieren registriert werden, um IP-Symcon mit dem Home Connect Konto zu verbinden.
In dem Feld 'Sprache' kann ausgewählt werden, in welcher Sprache die Informationen über die Geräte verarbeitet werden. Die hier gewählte Sprache steht nicht im Zusammenhang mit der in IP-Symcon genutzten Sprache.




Es muss sich mit dem Home Connect Konto angemeldet werden, um IP-Symcon die nötigen Berechtigungen einzuräumen.

Home Connect

Bitte melde dich hier mit deinem Home Connect Nutzerkonto an, um dieser Applikation den Zugriff auf deine Hausgeräte zu erlauben.

E-Mail

max@mustermann.com

••••••••••••••••••

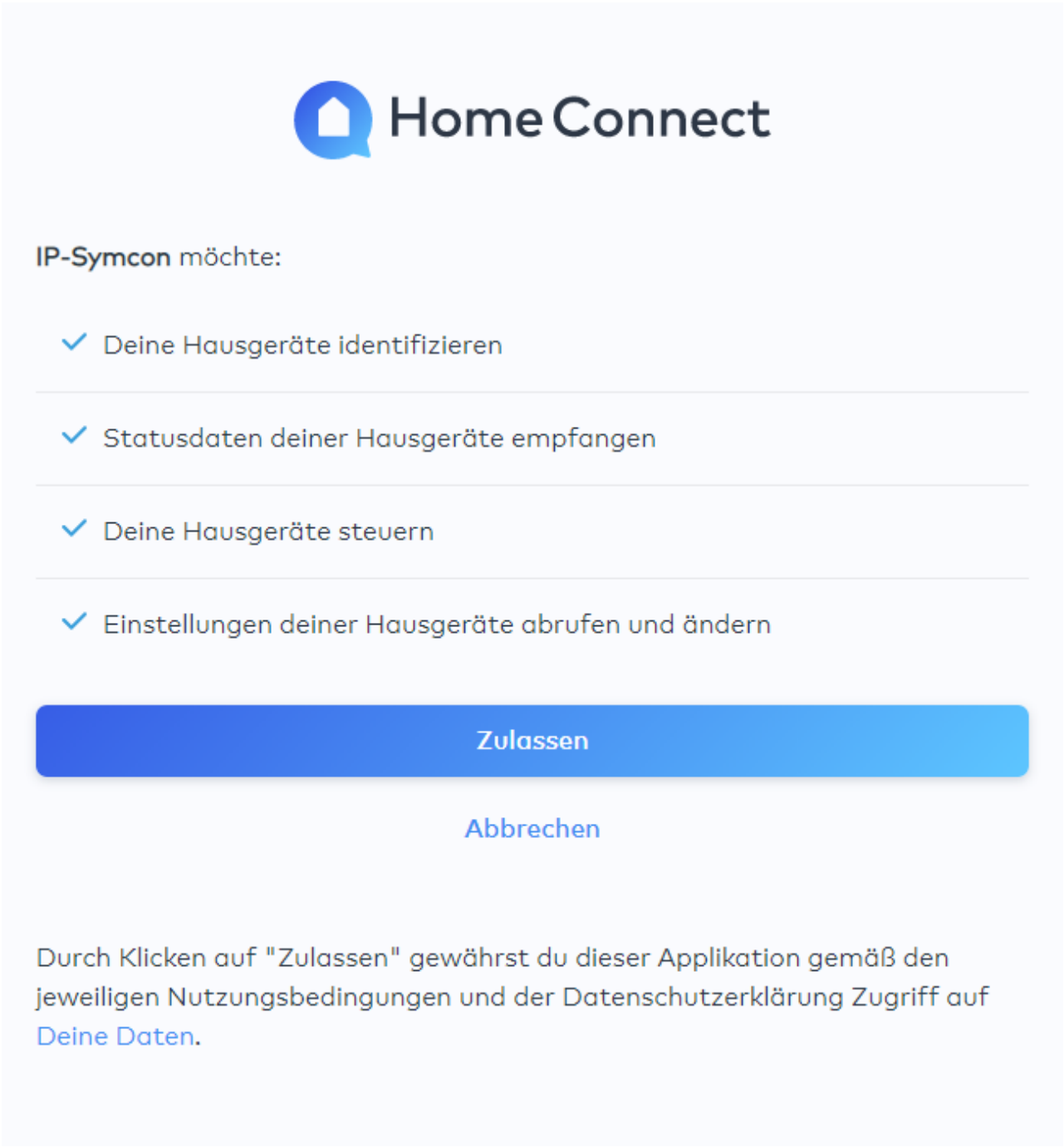
Anmelden

Abbrechen

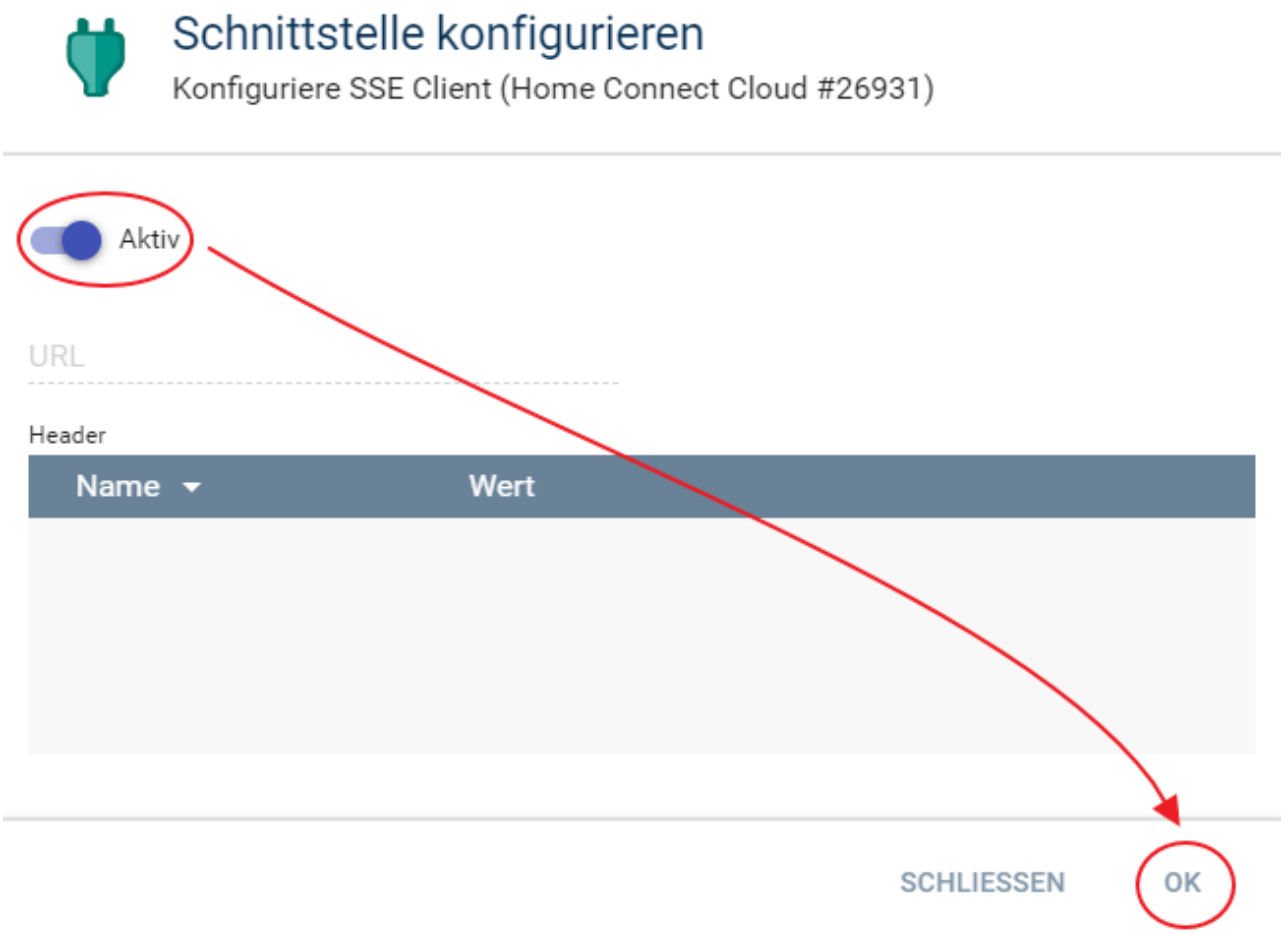
Passwort vergessen? Klicke hier, um dein Passwort zurückzusetzen.

Falls du noch kein Benutzerkonto hast, verwende bitte die Home Connect App um dich zu registrieren.

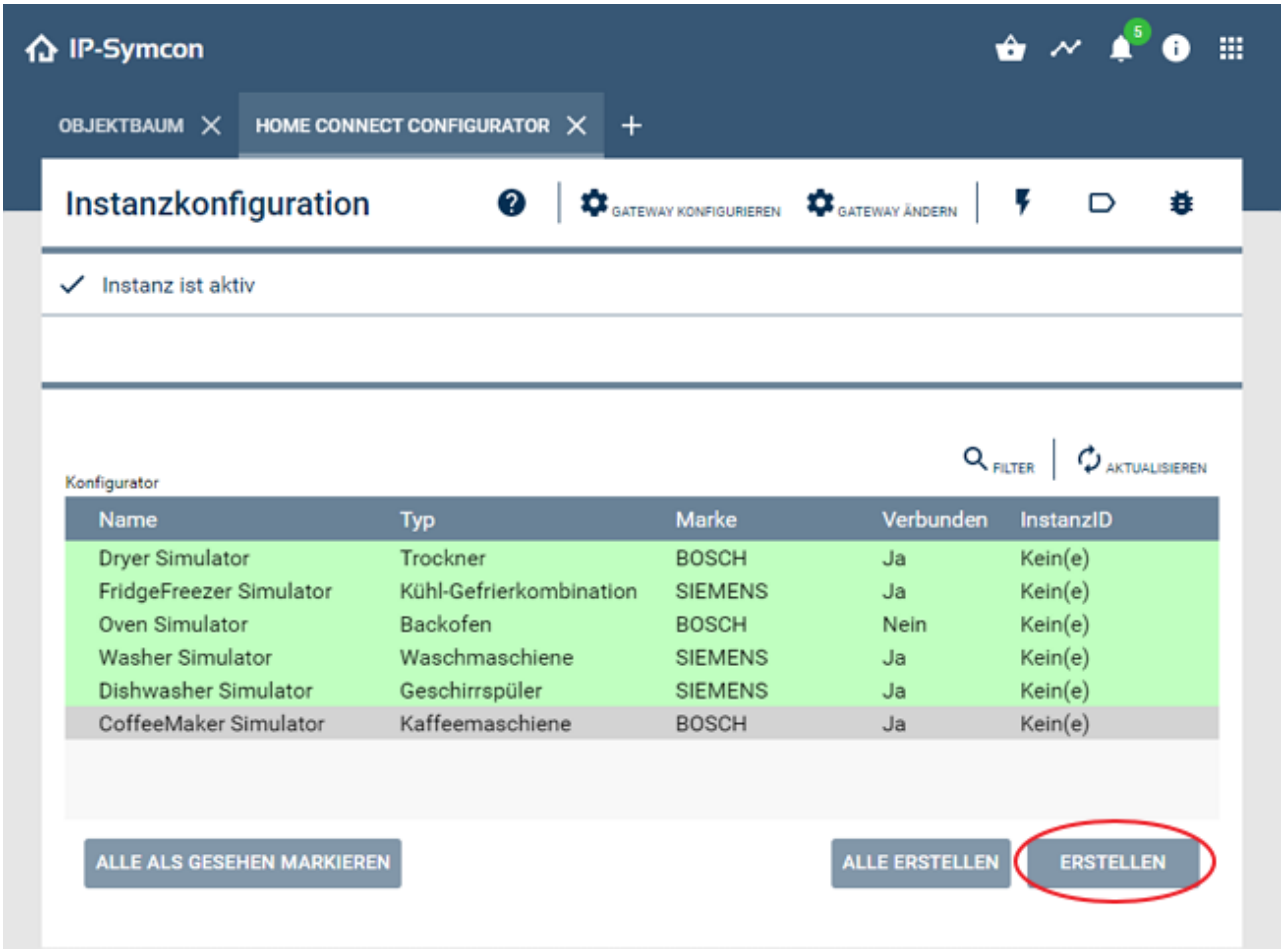
Erfahre mehr über Home Connect.




Daraufin kann der "Schnittstelle Konfigurieren"-Dialog mit Weiter bestätigt werden. Nun muss noch der erstellte SSE Client aktiviert werden.




Jetzt öffnet sich der Konfigurator und es können alle erkannten Geräte mit einem Klick auf "Erstellen" erstellt werden.


IP-Symcon
5
OBJEKTBAUM
HOME CONNECT CONFIGURATOR
Instanzkonfiguration
GATEWAY KONFIGURIEREN
GATEWAY ÄNDERN
Instanz ist aktiv
FILTER
AKTUALISIEREN
Konfigurator
Name
Typ
Marke
Verbunden
InstanzID
Dryer Simulator
Trockner
BOSCH
Ja
Kein(e)
FridgeFreezer Simulator
Kühl-Gefrierkombination
SIEMENS
Ja
Kein(e)
Oven Simulator
Backofen
BOSCH
Nein
Kein(e)
Washer Simulator
Waschmaschiene
SIEMENS
Ja
Kein(e)
Dishwasher Simulator
Geschirrspüler
SIEMENS
Ja
Kein(e)
CoffeeMaker Simulator
Kaffeemaschiene
BOSCH
Ja
Kein(e)
ALLE ALS GESEHEN MARKIEREN
ALLE ERSTELLEN
ERSTELLEN

# Geräteliste

## UNTERSTÜTZTE GERÄTE

Den Home Connect Dienst bieten die Geräte der Hersteller Bosch, Siemens, Gaggenau und Neff.
| Geräte Typ | |
| ---------- | |
| Kaffemaschiene | |
| Backofen | |
| Trockner | |
| Waschmaschiene | |
| Geschirrspüler | |
| Kühlschrank | |
| Gefrierschrank | |
Sollte ein Gerät nicht in der obigen Liste aufgeführt sein, so kann es trotzdem in IP-Symcon hinzugefügt werden. Wenn es bei der Einrichtung zu Problemen kommen, hilft unser Support gerne weiter.

# Home Connect Gerät

Benötigt: IP-Symcon >= 6.0

## BESCHREIBUNG

Wenn durch den Home Connect Konfigurator ein Gerät erstellt wird, wird eine Home Connect Geräte Instanz erstellt, welche abhängig vom Gerätetyp verschiedene Statusvariablen besitzt.

## STEUERUNG DURCH IP-SYMCON

Um die Geräte im vollen Umfang nutzen zu können, muss direkt am Gerät der Fernstart und/oder die Fernsteuerung aktiviert werden. Wenn nur die Fernsteuerung aktiv ist können Programme und Optionen verändert, aber das Programm nicht gestartet werden. Während das Gerät lokal bedient wird kann es durch IP-Symcon nicht gesteuert werden.

## STATUSVARIABLEN

Jedes Gerät kann 4 verschiedene Arten von Variablen haben.

### STATUS

Variablen des Typs Status sind Variablen, welche aus IP-Symcon heraus nicht geschaltet werden können.
**Beispiele:** Betriebsstatus, Lokale Bedienung aktiv

### EINSTELLUNG

Variablen des Typs Einstellung können meist geschaltet werden und dienen dazu verschiedene Einstellungen des Gerätes zu verändern.
**Beispiele:** Temperatur (Kühlschrank), Energiezustand

### PROGRAMM

Jedes Gerät hat nur eine Variable des Typs Programm. Über diese Variable kann festgelegt werden, welches Programm von dem Gerät ausgeführt werden soll.



Wenn ein Programm, welches in der Home Connect App verfügbar ist, nicht in IP-Symcon angezeigt wird, dann wird es von der API nicht unterstützt. Sollte über die App oder das Gerät manuell ein solches Programm ausgewählt werden, dann werden alle Einstellungsmöglichkeiten ausgegraut

### OPTIONEN

Wenn ein Programm ausgewählt ist werden entsprechende Option Variablen erstellt, welche die Details für das ausgewählte Programm verändern.
**Beispiele:** Füllmenge (Kaffeemaschine), Dauer (Backofen)
Wenn ein Programm ausgewählt wurde und die verfügbaren Optionen gesetzt wurden, kann das Programm über die Variable Steuern gestartet werden.
Sobald ein Programm gestartet wurde werden, ja nach Gerät, Variablen angelegt, die Auskunft über den Fortschritt geben.
**Beispiele:** Programmfortschritt, Verbleibende Laufzeit

## BESONDERHEITEN

Bei der Erstellung eines Gerätes sollte dieses Online und somit angeschaltet sein
Der Kühlschrank hat keine Programme

## PROBLEMBEHANDLUNG

### EINE GERÄTEINSTANZ WIRD NICHT KORREKT ERSTELLT

Alle Daten eines Gerätes können in der Geräteinstanz mit einem Klick auf den Button 'Gerät Initialisieren' erneut angefragt werden.

### VARIABLEN WERDEN NICHT AKTUALISIERT

Durch einen Klick auf 'Server Events registrieren' wird die Verbindung mit dem Home Connect Dienst erneut hergestellt.

# HomeMatic

Symcon unterstützt die Integration über die HCU und über die CCU.

- HomeMatic CCU
- HomeMatic HCU (Beta)

# Geräteliste

## UNTERSTÜTZTE GATEWAYS

| GATEWAY | BESCHREIBUNG |
|---|---|
| CCU1 | Zentrale für HomeMatic Funk- und Wired-Komponenten |
| CCU2 | Zentrale für HomeMatic Funk-, Wired- und IP-Komponenten |
| CCU3 | Zentrale für HomeMatic Funk-, Wired- und IP-Komponenten |
| HM-LAN (über BidCos-Service) | HomeMatic-LAN-Adapter |
| HM-USB (über BidCos-Service) | HomeMatic-USB-Adapter |
| HCU | Zentrale für HomeMatic IP Komponenten |

## UNTERSTÜTZTE KOMPONENTEN

IP-Symcon unterstützt alle kabel- und funkgebundenen Komponenten.

# HomeMatic CCU

Benötigt: IP-Symcon >= 4.1
HomeMatic ist ein primär funkbasierendes System. Eine Verbindung zu IP-Symcon wird via LAN über die Central Control Unit (CCU) hergestellt. Alternativ funktioniert dies auch mit einem "LAN-Konfiguration-Adapter".



Folgende Geräte werden von IP-Symcon unterstützt:
Unterstützte Geräte

## INSTALLATION

Die HomeMatic Geräte werden via CCU (LAN) oder BidCoS Service an IP-Symcon angeschlossen.
Die Anleitung zur Einrichtung via BidCoS Service befindet sich hier.



Vorteile der CCU:
HomeMatic Wired RS485 Komponenten werden unterstützt.
Programme können in der CCU ausgeführt werden.
Status der Aktoren können per Skript abgefragt werden.

### HOMEMATIC WIRED

HomeMatic Wired ist eine kabelgebundene Version von HomeMatic. Hierfür sind eigene Schalt- und Dimmaktoren sowie Ein- und Ausgabemodule erhältlich.
"HomeMatic Wired" wird von IP-Symcon aber in gleicher Art und Weise angesteuert, wie die auf Funk basierenden Komponenten.
Dieses kann auf der Konfigurationsseite des Gateways eingeschaltet werden.

### HOMEMATIC IP

HomeMatic IP ist der Nachfolger von HomeMatic und bietet Funk-, sowie auch Wired-Geräte. HomeMatic IP wird von IP-Symcon in gleicher Art und Weise angesteuert, wie die Ursprünglichen HomeMatic Komponenten. Ein paralleler Betrieb von HomeMatic und HomeMatic IP Komponenten ist problemlos möglich.

## VIDEO-TUTORIAL ZUR EINRICHTUNG

## ANSCHLUSS

Die CCU wird durch ein Netzwerkkabel mit dem Netzwerk verbunden.

Die Antenne der CCU dient nur der Funkverbindung mit den einzelnen Aktoren. Eine WLAN-Verbindung mit einem PC ist jedoch nicht möglich.

Sollte kein LAN-Netzwerk vorhanden sein, ist eine Verbindung auch über USB möglich. Diese benötigt extra USB-Treiber, welche sich auf der mitgelieferten CD befinden.

## INSTALLATION

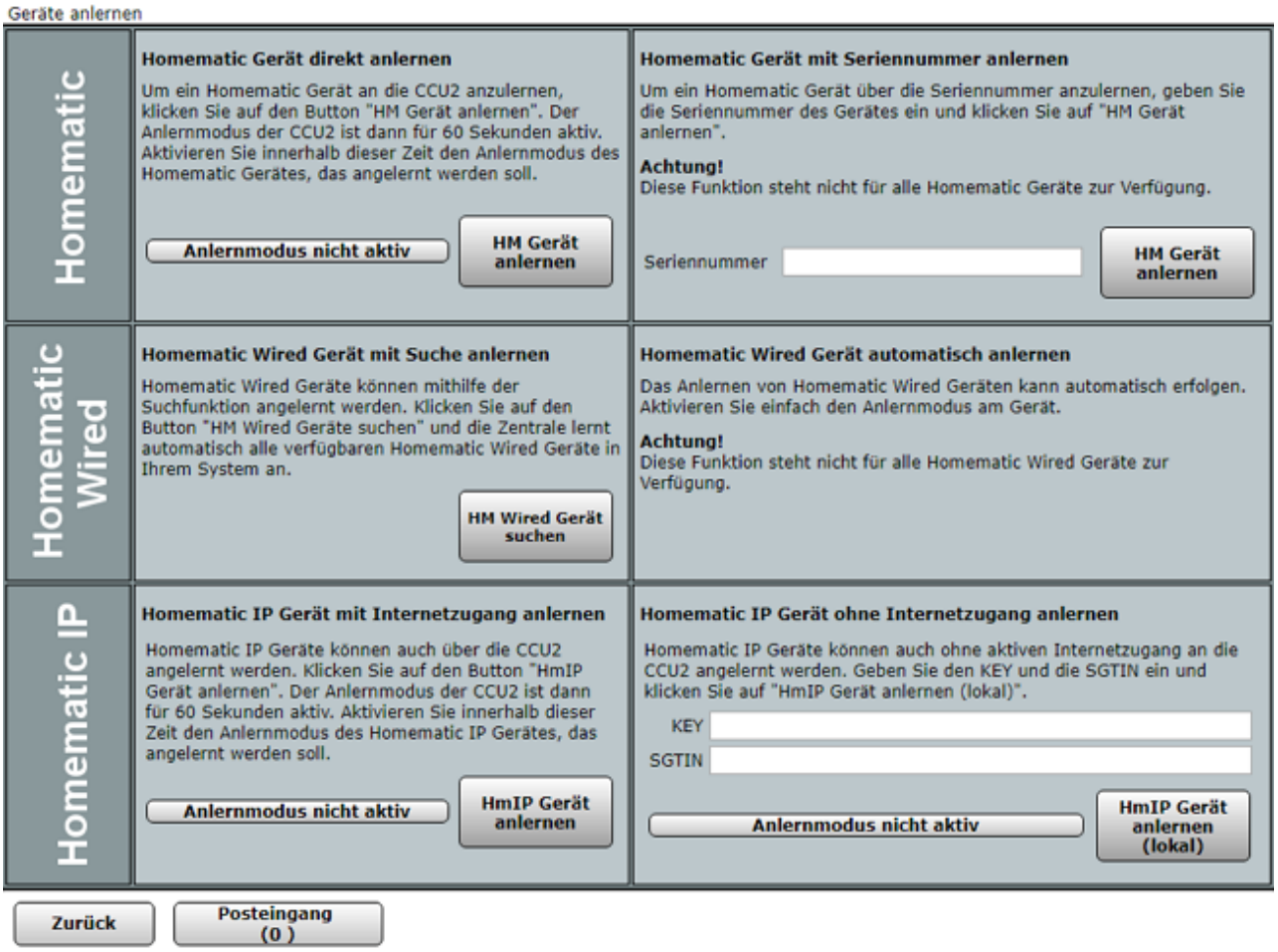
Wenn die CCU korrekt angeschlossen ist und ein DHCP-Server verwendet wird, erhält die CCU automatisch eine IP-Adresse und ist mit dem Netzwerk verbunden.
Steht kein DHCP-Server zur Verfügung, kann eine IP-Adresse auch manuell über das Menü der CCU, unter dem Menüpunkt „Network“, eingegeben werden.
Hierbei müssen auch die Subnetzmaske und das Standardgateway vergeben werden. Bei korrekter Einrichtung ist nun in der Netzwerkübersicht das Gerät „HomeMatic Central“ zu finden.

Sollte die CCU nicht im Netzwerk erscheinen, hilft ein Trennen der Stromversorgung der CCU für 30 Sekunden. Nach dem Hochfahren der CCU und Aktualisierung der Netzwerkumgebung ist diese nun erreichbar.

Ein Doppelklick auf „HomeMatic Central“ öffnet ein Fenster in dessen unteren Bereich die IP-Adresse der CCU zu finden ist. (Optional ist diese auch im Menü der CCU zu finden.)
Durch kopieren dieser Adresse in einen Browser, öffnet sich das HomeMatic WebUI.

## HINZUFÜGEN SEKUNDÄRER GERÄTE

Im HomeMatic WebUI können nun die sekundären HomeMatic-Geräte (z.B. Temperaturfühler, Zwischenstecker, Handsender usw.) mit der CCU verbunden werden. Dazu muss auf „Geräte anlernen“ geklickt (rechts oben) und den weiteren Anweisungen gefolgt werden.

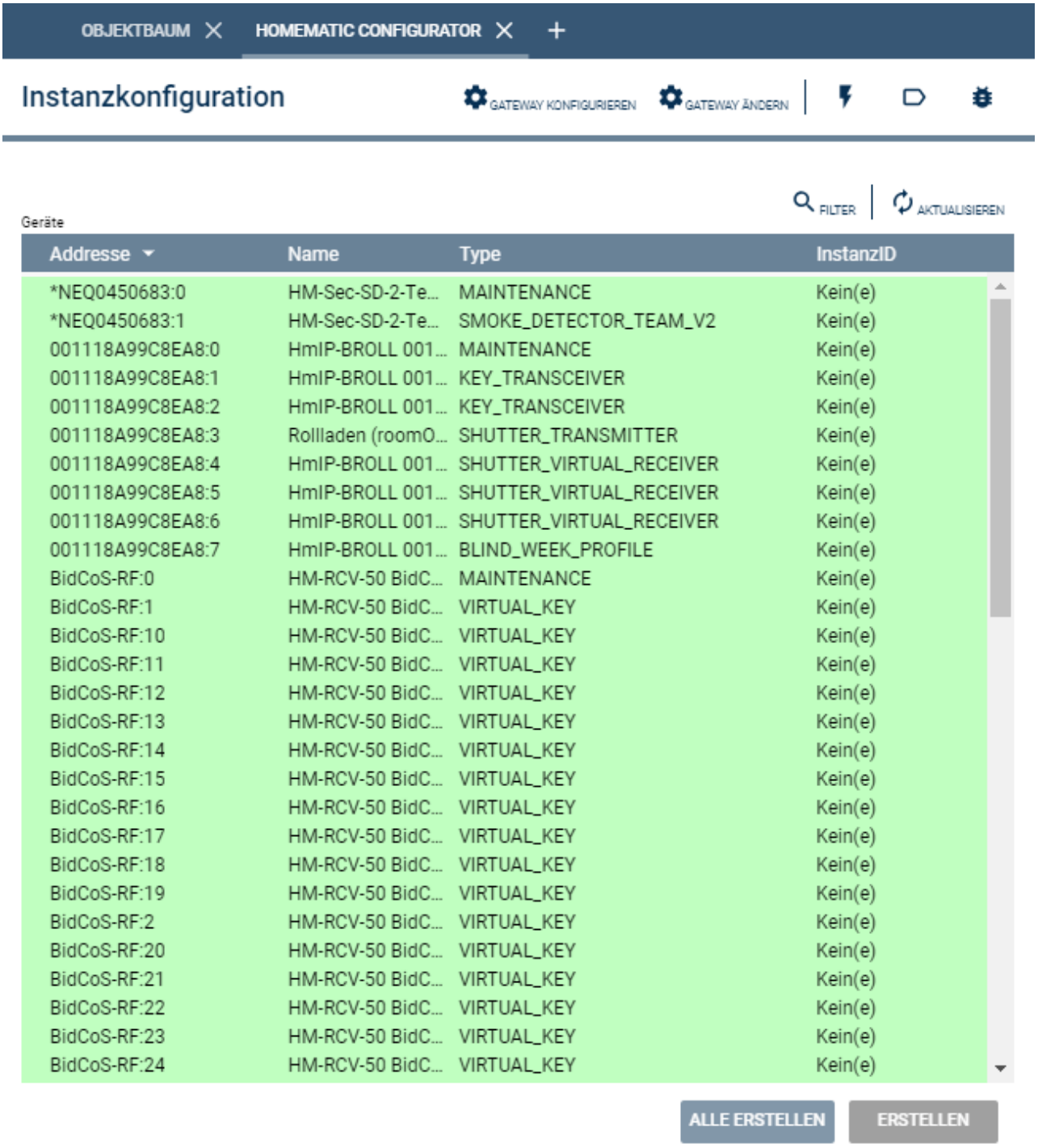



## EINBINDUNG IN IP-SYMCON

Die CCU kann über die Gerätesuche eingebunden werden. Hierfür muss als System "Homematic Discovery" gewählt werden. Die Discovery-Instanz bietet dann die Erstellung eines Homematic Konfigurators an. Nachdem der Konfigurator erstellt wurde, können über diesen die einzelnen Geräte wie unten beschrieben eingebunden werden. Innerhalb des Konfigurators kann über "Gateway konfigurieren" die Konfigurationsseite des "HomeMatic Sockets" geöffnet werden. Um den Socket zu aktivieren muss der Haken bei "Socket öffnen" gesetzt werden. Die angezeigten Optionen sagen, ob HomeMatic Funk/Wired/IP genutzt werden soll. Bei aktivierter Authentifizierung kann hier der Benutzername und das Passwort eingetragen werden.




Im mittig erscheinenden Fenster sind jetzt verschiedene Geräte sichtbar. Nach der Auswahl eines Geräts kann eine Geräteinstanz mit "Erstellen" erstellt und anschließend die Konfigurationsseite über "Konfigurieren" aufgerufen werden.

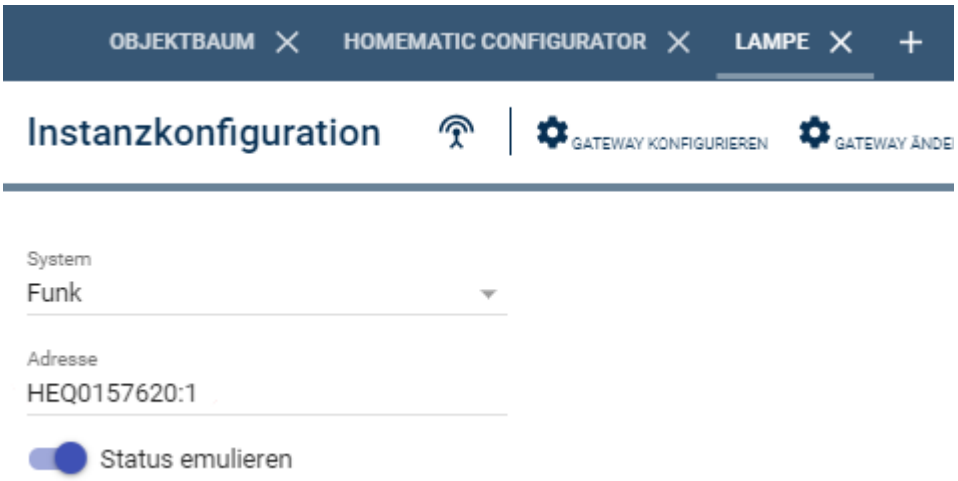



Um zu testen ob z.B. ein Dimmer funktioniert, kann die ordnungsgemäße Funktion, z.B. in der Visualisierung, überprüft werden.

Unter Umständen funktioniert der Rückkanal nicht. Dies ist dadurch bemerkbar, dass IP-Symcon nicht visualisiert wenn z.B. eine Lampe über einen Lichtschalter geschaltet wurde.
Dieses Problem liegt oftmals an der Windows-Firewall. Diese kann in Windows unter "Systemsteuerung" -> "System und Sicherheit" -> "Windows-Firewall" konfiguriert werden.
Im Menü "Ein Programm oder Feature durch die Windows-Firewall zulassen" -> "Einstellungen ändern" -> "Anderes Programm zulassen..." befindet sich der Button "Durchsuchen".
Durch auswählen von der IP-Symcon (IP-Symcon Service) Datei und anschließendem "öffnen", funktioniert auch der Rückkanal!
Eine andere Möglichkeit die Funktionalität des Rückkanals zu gewährleisten, ist darauf zu achten, dass unter den Firewall-Einstellungen in der CCU “HomeMatic XML-RPC API” und “Remote HomeMatic-Script API” Vollzugriff ausgewählt oder die IP-Adresse des IP-Symcon PCs eingetragen sind.

## GERÄTEKONFIGURATION

Es ist empfohlen den HomeMatic Konfigurator zur Einrichtung der einzelnen Geräte zu nutzen.
Auf der Konfigurationsseite der Geräteinstanz sind folgende Einstellungen möglich.



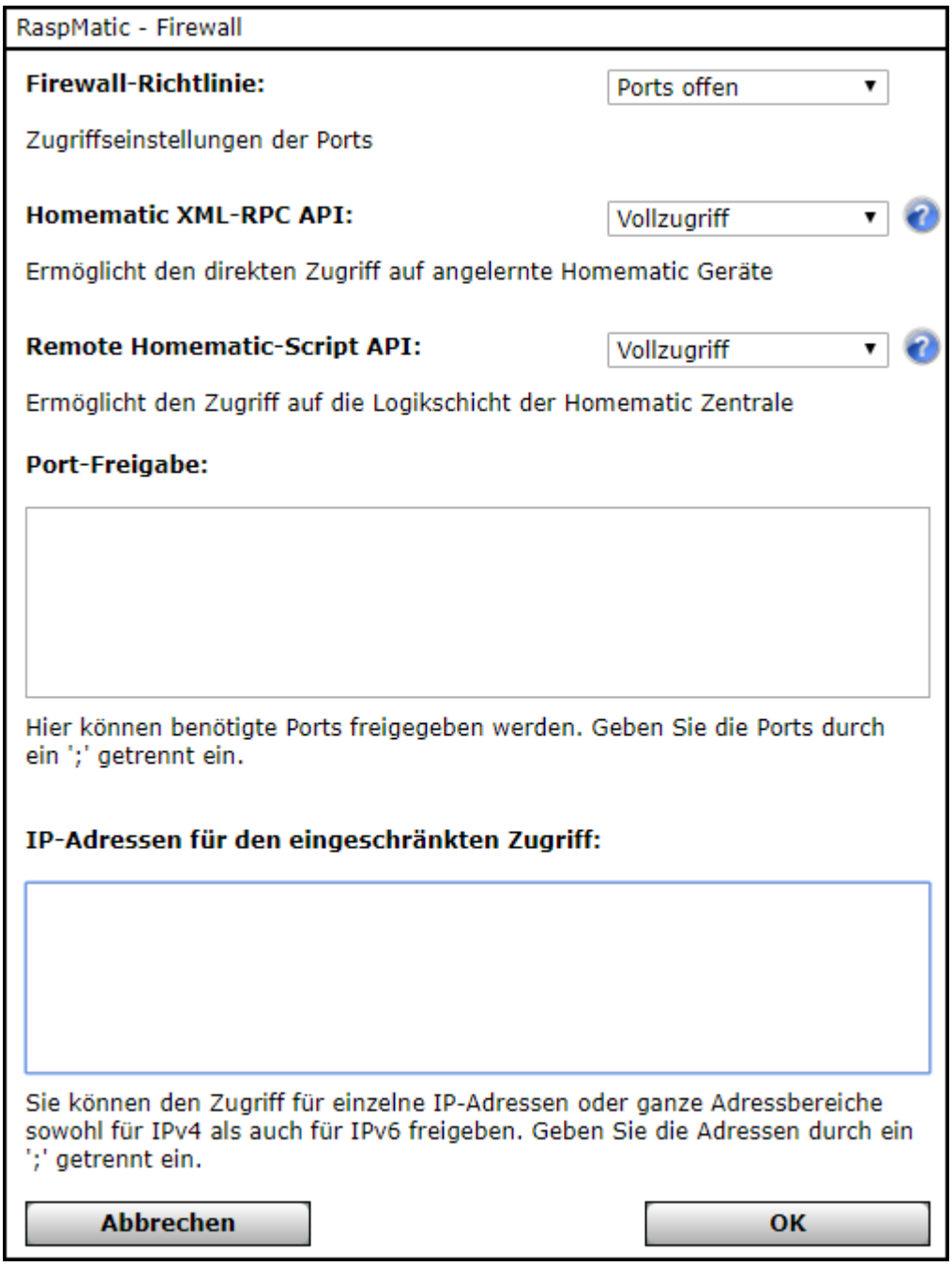

| OPTION | BESCHREIBUNG |
|---|---|
| System | Auswahl ob das Gerät via Funk, Wired oder IP angesprochen wird. |
| Adresse | HomeMatic Adresse des Geräts. Diese kann über den "Suchen" Dialog in der oberen Leiste der Konfigurationleiste auch gesucht werden. |
| Status emulieren | Wenn eingeschaltet wird der Status des Geräts bei erfolgreicher Befehlsübermittlung an die CCU aktualisiert. Wenn ausgeschaltet wird auf die tatsächliche Antwort des zu schaltenden Geräts gewartet. Dies kann zu Verzögerungen führen. Zum Beispiel bei Geräten, welche nur alle x Minuten aufwachen. |

## FIREWALL-, AUTHENTIFIZIERUNG- UND PORTEINSTELLUNGEN

Ab einer CCU3 oder RaspberryMatic in Kombination mit IP-Symcon 5.1 oder höher ist eine Authentifizierung möglich.
Diese kann im Webinterface der CCU eingestellt werden. Unter "Einstellungen" -> "Systemsteuerung" -> "Sicherheit" kann die Authentifizierung aktiviert werden. Diese nutzt dann die Benutzer, welche in der Benutzerverwaltung eingestellt sind.



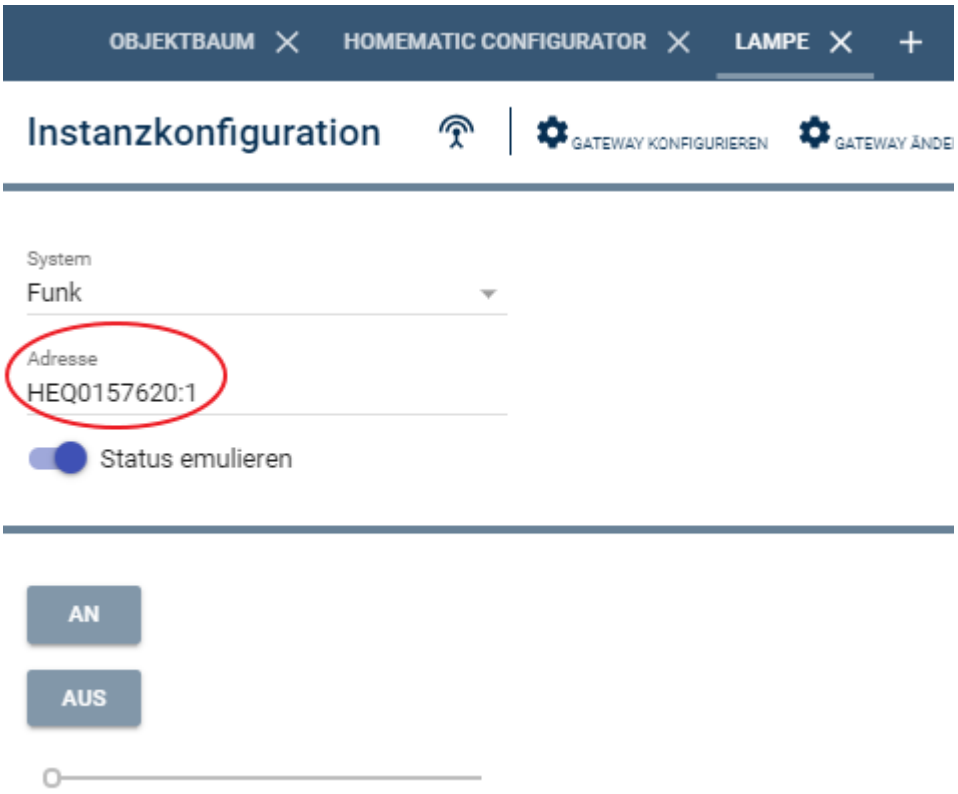

Desweiteren kann unter "Einstellungen" -> "Systemsteuerung" -> "Firewall konfigurieren" die Firewall eingerichtet werden. Weitere Informationen dazu befinden sich in der Anleitung der CCU.




## AUSTAUSCH VON DEFEKTEN GERÄTEN

Allgemeine Informationen wie ein Austausch in IP-Symcon funktioniert kann hier

gefunden werden
Zum Austausch muss in IP-Symcon die Konfiguration des defekten Geräts geöffnet und die Adresse des alten Geräts gegen die des Neuen ausgetauscht werden. Die Adresse steht entweder auf dem Gerät selbst oder kann via "Suchen" ausgewählt werden.




Die Adresse im markierten Bereich muss ausgetauscht und mit "Übernehmen" gespeichert werden.

## HOMEMATIC MIT QNAP, SYNOLOGY ODER DOCKER

Soll HomeMatic mit QNAP, Synology oder Docker verwendet werden, so muss der NATSupport eingerichtet werden um Antworten des Systems zu erhalten. Dieser Vorgang ist hier erklärt.

## TIPPS & TRICKS

Dokumentation Homematic Datenpunkte
Dokumentation HomematicIP Datenpunkte
HomeMatic-Homepage
Es kann das Display der 19 Tasten Fernbedienung verändert werden.
Um den Status aller Aktoren und Sensoren anzufragen (nur mit der CCU möglich!).
Die Einschaltdauer für einen Aktor kann mit “ON_TIME” und bei einem Dimmer kann zusätzlich eine Rampe angegeben werden.

Hier ein fertiges Beispiel-Skript für drei Dimmer, die eine Licht-Szene gestalten:

```
$id_bar = 54392 /*[EG\Tisch Bar]*/;
$id_ecken = 24601 /*[EG\Strahler Ecken]*/;
```

```
$id_tisch = 38758 /*[EG\Tischlampe]*/;

$ramp = 2;
HM_WriteValueFloat($id_bar, "ON_TIME", 60*10); // x Minuten AN
HM_WriteValueFloat($id_bar, "RAMP_TIME", $ramp); // X Sekunden Rampe
HM_WriteValueFloat($id_bar , "LEVEL" , .4); // und ausführen auf X%

HM_WriteValueFloat($id_ecken , "RAMP_TIME", $ramp); // X Sekunden
Rampe
HM_WriteValueFloat($id_ecken , "LEVEL" , .4); // und ausführen auf
X%

HM_WriteValueFloat($id_tisch , "RAMP_TIME", $ramp); // X Sekunden
Rampe
HM_WriteValueFloat($id_tisch , "LEVEL" , .4); // und ausführen auf
x%
```

Beispiel-Skript WinMatic 60 Minuten Belüften:

```
$id_aktor = 49712 /*[OG\Schlafzimmer\HM WinMatic]*/;
HM_WriteValueFloat($id_aktor, "SPEED" , 1.0); // Maximum Speed ;
HM_WriteValueFloat($id_aktor, "RELOCK_DELAY" , 60*60); // Fenster
nach XX Minuten wieder zu
HM_WriteValueFloat($id_aktor, "LEVEL" , 0.7); // Fenster Auf
```

Beispiel-Skript KeyMatic öffnen:

```
HM_WriteValueBoolean($id, "OPEN", true);
```

Beispiel-Skript um eine Rückmeldung zu erhalten, ob ein Aktor den Schaltbefehl korrekt ausgeführt hat:

```
$id_aktor = 12345;
$err = HM_WriteValueBoolean($id_aktor, "STATE" , False);
//echo "Err: " .(int) $err . "\n";
if ($err === False){
echo "Fehler: Schaltaktor - Befehl wurde nicht ausgeführt\n";
SetValue($id_done, False);
} else {
SetValue($id_done, True);
echo "OK: Schaltaktor - Befehl wurde ausgeführt\n";
}
```

Beispiel-Skript, um einen Aktor (Lampe) zu toggeln (ein- und ausschalten):

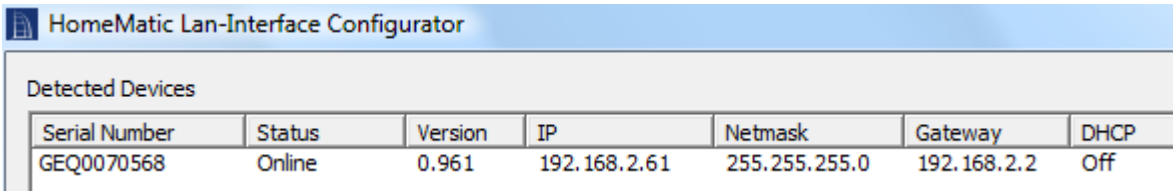
```
$id_aktor = 25404 /*[Hof Garten\Lampe]*/;
$id_state = 55194 /*[Hof Garten\Lampe\STATE]*/;
HM_WriteValueBoolean($id_aktor, "STATE" , !GetValue($id_state));
```

# BidCos Service

Der BidCos Service ist eine Alternative zur CCU-gestützten Ansteuerung von HomeMatic Geräten.

## KONFIGURATION DES LAN-ADAPTERS

Als erstes muss die aktuelle Usersoftware von der HomeMatic-Seite installiert werden. Wir empfehlen, eine feste IP-Adresse (DHCP: Off) zu vergeben:




Nach erfolgreicher Installation wurde ein neues Verzeichnis angelegt: "x:\ProgramData\Bidcos-Service\"
Darin muss in der Datei 'bidcos.conf' die Serial Number (siehe Etikett) sowie die IP-Addresse des Adapters angepasst werden.
**Beispiel:**

```
[Interface 0]
Type = Lan Interface
Serial Number = GEQ0123456

# Key ist auf der Rückseite des Interfaces.
# Funktion kann über das LAN Tool auch deaktiviert werden.
# Dann kann das Feld leergelassen werden
Encryption Key =
Description = First Lan Interface

# IP-Adresse vom LAN Interface
IP Address = 192.168.2.61
```

Weitere Einstellungen in der bidcos.conf sind nicht erforderlich. Essenziell ist, dass folgender Eintrag unverändert bleibt:

```
# TCP Port for XmlRpc connections
Listen Port = 2001
```

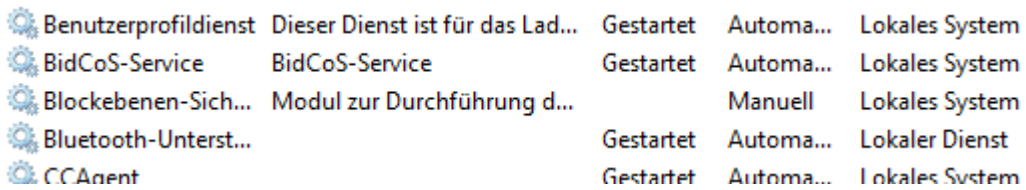
## TESTEN DES ADAPTERS

Eine erfolgreiche Konfiguration kann nun mit dem Programm "Test-Run BidCos-Service" überprüft werden. Die letzte Zeile im DOS-Fenster lautet dann: "… Connected to Lan

Interface …".

> Eine blinkende Power LED deutet auf einen nicht verbundenen BidCos-Service hin.

Bei Bedarf können die Startoptionen angepasst werden (Systemsteuerung > Verwaltung > Dienste) oder der Service manuell gestartet werden:

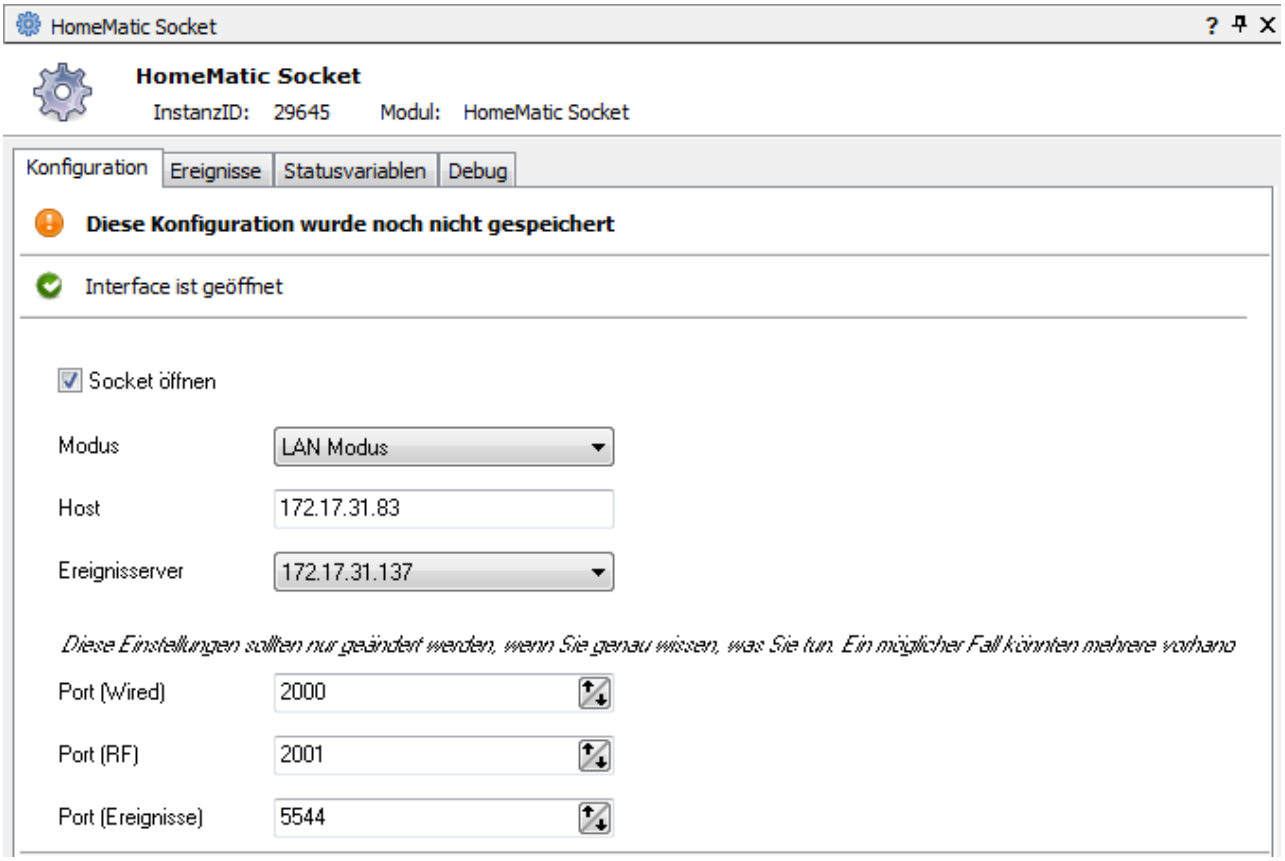



## HOMEMATIC-KOMPONENTEN KONFIGURIEREN

Im Programm "HomeMatic Konfigurator" muss unter "Datei > Einstellungen" die Verbindung "entfernter BidCoS Service" ausgewählt werden. Danach folgt noch die Eingabe der IP-Adresse (localhost) sowie des Ports (2001).
Zum Schluss können nun nacheinander im Menü "Geräte anlernen" die HomeMatic-Komponenten hinzugefügt werden.

## CLIENT SOCKET IN IP-SYMCON

Der HomeMatic Socket (in den I/O Instanzen) muss konfiguriert werden. Als Modus muss "LAN Mode" ausgewählt werden. Sollte der BidCoS Service auf einem anderen PC laufen, so ist als Host noch dessen IP-Adresse einzutragen. Andernfalls muss "localhost" als Adresse angegeben bleiben. Falls mehrere Netzwerk-Adapter vorhanden sein sollten, so muss im Event-Server die IP-Adresse des Netzwerk-Adapers ausgewählt werden, an dem der LAN-Apapter angeschlossen ist. Der Port (5544) muss nicht verändert werden, sondern nur in einer ggf. vorhandenen Firewall freigeschaltet werden.


HomeMatic Socket
HomeMatic Socket
InstanzID: 29645 Modul: HomeMatic Socket
Konfiguration
Ereignisse
Statusvariablen
Debug
Diese Konfiguration wurde noch nicht gespeichert
Interface ist geöffnet
Socket öffnen
Modus
LAN Modus
Host
172.17.31.83
Ereignisserver
172.17.31.137
Diese Einstellungen sollten nur geändert werden, wenn Sie genau wissen, was Sie tun. Ein möglicher Fall könnten mehrere vorhano
Port (Wired)
2000
Port (RF)
2001
Port (Ereignisse)
5544

# HM_ReadServiceMessages

```
array HM_ReadServiceMessages (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der HomeMatic Socket Instanz |

## RÜCKGABEWERT

Folgende Informationen stehen als **key => value** Paare zur Verfügung:

| INDEX | TYP | BESCHREIBUNG |
|---|---|---|
| **Address** | string | Adresse des betreffenden Geräts |
| **Message** | string | Kryptische Beschreibung der Fehlermeldung. Text kann teilweise der stringtable_de.txt aus dem HomeMatic Konfigurationstool entnommen werden. |
| **Value** | variant | Wert der Fehlermeldung |

## BESCHREIBUNG

Liest die Service-Meldungen des CCU/LAN-Adapters mit der ID **InstanzID** aus.

## BEISPIEL

```
//Erstellt eine Variable die in der Visualisierung alle
ServiceMeldungen anzeigt. Einfach in ein Skript kopieren und
ausführen.

//Ab hier nichts mehr ändern
$object = IPS_GetObject($_IPS['SELF']);
$parentID = $object['ParentID'];

//Installer
if ($_IPS['SENDER'] == "Execute")
{
IPS_SetHidden($_IPS['SELF'], true);
IPS_SetName($_IPS['SELF'], "Auslese-Skript");
$parentObject = IPS_GetObject($parentID);
if ($parentObject['ObjectType'] !== 1)
{
$instanceID = IPS_CreateInstance("{485D0419-BE97-4548-AA9C-
C083EB82E61E}");
```

```
IPS_SetParent($instanceID, $parentID);
$parentID = $instanceID;
IPS_SetParent($_IPS['SELF'], $parentID);
IPS_SetName($instanceID, "Servicemeldungen");
}
IPS_SetScriptTimer($_IPS['SELF'], 300);
}

$texte = Array(
"CONFIG_PENDING" => "Konfigurationsdaten stehen zur Übertragung an",
"LOWBAT" => "Batterieladezustand gering",
"STICKY_UNREACH" => "Gerätekommunikation war gestört",
"UNREACH" => "Gerätekommunikation aktuell gestört"
);

$str = "<table width='90%' align='center'>"; // Farbe anpassen oder
style entfernen
$str .= "<tr><td><b>Gerätname</b></td><td><b>GeräteID</b></
td><td><b>Meldung</b></td></tr>";

$ids = IPS_GetInstanceListByModuleID("{A151ECE9-D733-4FB9-
AA15-7F7DD10C58AF}");
if(sizeof($ids) == 0)
die("Keine HomeMatic Socket Instanz gefunden!");

$msgs = HM_ReadServiceMessages($ids[0]);
if($msgs === false)
die("Verbindung zur CCU fehlgeschlagen");

if(sizeof($msgs) == 0)
$str .= "<tr><td colspan=3><br/>Keine Servicemeldungen!</td></tr>";

foreach($msgs as $msg)
{
if(array_key_exists($msg['Message'], $texte)) {
$text = $texte[$msg['Message']];
} else {
$text = $msg['Message'];
}

$id = GetInstanceIDFromHMID($msg['Address']);
if(IPS_InstanceExists($id)) {
$name = IPS_GetLocation($id);
} else {
$name = "Gerät nicht in IP-Symcon eingerichtet";
}

$str .= "<tr><td>".$name."</td><td>".$msg['Address']."</td><td>".
$text."</td></tr>";
}
$str .= "</table>";

$vid = CreateVariableByName($parentID, "Content", 3);
IPS_SetIcon($vid, "Information");
IPS_SetVariableCustomProfile($vid, "~HTMLBox");
```

```
SetValue($vid, $str);

function GetInstanceIDFromHMID($sid)
{
$ids = IPS_GetInstanceListByModuleID("{EE4A81C6-5C90-4DB7-AD2F-
F6BBD521412E}");
foreach($ids as $id)
{
$a = explode(":", IPS_GetProperty($id, 'Address'));
$b = explode(":", $sid);
if($a[0] == $b[0])
{
return $id;
}
}
return 0;
}

function CreateVariableByName($id, $name, $type)
{
$vid = @IPS_GetVariableIDByName($name, $id);
if($vid === false)
{
$vid = IPS_CreateVariable($type);
IPS_SetParent($vid, $id);
IPS_SetName($vid, $name);
}
return $vid;
}
```

# HM_RequestStatus

```
boolean HM_RequestStatus (int $InstanzID, string $Parameter)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz des zu schaltenden Gerätes |
| Parameter | |

| WERT | BESCHREIBUNG |
|---|---|
| STATE | Status eines Aktors |
| LEVEL | Dimmwert |
| ... | Eine Übersicht aller abfragbaren "lesend" Parameter kann der Dokumentation der Homematic Datenpunkte unter Tipps & Tricks entnommen werden. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Fragt die Statusvariable mit dem Namen **Parameter** des Gerätes mit der ID **InstanzID** ab.



Den einzutragenen **Parameter** können Sie anhand des Idents der Statusvariablen der Instanz ableiten. In der Parameterliste werden einige Beispiele gegeben.



**Diese Funktion belastet den Funkverkehr erheblich.**
Sie sollte ausschliesslich verwendet werden, um kritische Geräte zum IP-Symcon-Start zu aktualisieren. Ein erneutes Abfragen zur Laufzeit wird nicht benötigt, da die HomeMatic alle Statusänderungen direkt an IP-Symcon weiterleitet.



Bei batteriebetriebenen Geräten wird nur der gespeicherte Status des CCU/LAN-Adapters abgefragt. Netzbetriebene Geräte werden per Funk direkt abgefragt.

## BEISPIEL

```
//Dies fragt eine Aktualisierung aller Parameter aller Homematic-
Instanzen an.
//Dies belastet den Funkverkehr extrem und ist nicht als allgemeine
Aktualisierung empfohlen.
$ids = IPS_GetInstanceListByModuleID("{EE4A81C6-5C90-4DB7-AD2F-
F6BBD521412E}");
echo "Geräte: ".sizeof($ids)."\n";
foreach($ids as $id)
{
$cids = IPS_GetChildrenIDs($id);
foreach($cids as $cid)
{
$o = IPS_GetObject($cid);
if($o['ObjectIdent'] != "") {
if(@HM_RequestStatus($id, $o['ObjectIdent']) === false) {
echo "Fehler: ".IPS_GetLocation($id)."\n";
}
break;
}
}
}
```

# HM_WriteValueBoolean

```
boolean HM_WriteValueBoolean (int $InstanzID, string $Parameter,
boolean $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz des zu schaltenden Gerätes |
| Parameter | Parameter welcher gesetzt werden soll. Der Name des Parameters kann in der Datenpunktliste oder als "Ident" im Reiter "Statusvariablen" in IP-Symcon nachgesehen werden. |
| Wert | true oder false |

## RÜCKGABEWERT

default

## BESCHREIBUNG

Setzt den **Parameter** des Gerätes mit der ID **InstanzID** auf den Wert **Wert**.

Den einzutragenen **Parameter** können Sie anhand des Idents der Statusvariablen der Instanz ableiten. In der Parameterliste werden einige Beispiele gegeben. Anhand des Typs der Variable können Sie auch die notwendige **HM_WriteValue** Funktion erkennen.

Weitere Datenpunkte können der HomeMatic Datenpunktliste entnommen werden.

Sollte die Fehlermeldung **Failure** auftreten, so ist dies im Normalfall auf eine mangelhafte Funkverbindung zurückzuführen.

## BEISPIEL

```
//Gerät einschalten
HM_WriteValueBoolean(12345, "STATE", true);
```

```
//Gerät sperren
HM_WriteValueBoolean(12345, "INHIBIT", true);

//Display Beleuchtung einschalten
HM_WriteValueBoolean(12345, "BACKLIGHT", true);
```

# HM_WriteValueFloat

```
boolean HM_WriteValueFloat (int $InstanzID, string $Parameter, float $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz des zu schaltenden Gerätes |
| Parameter | Parameter welcher gesetzt werden soll. Der Name des Parameters kann in der Datenpunktliste oder als "Ident" im Reiter "Statusvariablen" in IP-Symcon gefunden werden. |
| Wert | Fließkommazahl die gesetzt werden soll. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den **Parameter** des Gerätes mit der ID **InstanzID** auf Wert **Wert**.

Den einzutragenen **Parameter** können Sie anhand des Idents der Statusvariablen der Instanz ableiten. In der Parameterliste werden einige Beispiele gegeben. Anhand des Typs der Variable können Sie auch die notwendige **HM_WriteValue** Funktion erkennen.

Weitere Datenpunkte können der HomeMatic Datenpunktliste entnommen werden.

Sollte die Fehlermeldung **Failure** auftreten, so ist dies im Normalfall auf eine mangelhafte Funkverbindung zurückzuführen.

## BEISPIEL

```
//Gerät auf 25% dimmen. Werte erlaubt zwischen 0.0 und 1.0
HM_WriteValueFloat(12345, "LEVEL", 0.25);
```

```
// Setzt die Einschaltdauer auf 1 Stunde. Werte erlaubt zwischen 0 und 85825945.6 in Sekunden
HM_WriteValueFloat(12345, "ON_TIME", 3600);
```

# HM_WriteValueInteger

```
boolean HM_WriteValueInteger (int $InstanzID, string $Parameter, int
$Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz des zu schaltenden Gerätes |
| Parameter | Parameter welcher gesetzt werden soll. Der Name des Parameters kann in der Datenpunktliste oder als "Ident" im Reiter "Statusvariablen" in IP-Symcon gefunden werden. |
| Wert | Ganzzahliger Wert der gesetzt werden soll. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den **Parameter** des Gerätes mit der ID **InstanzID** auf Wert **Wert**.

Den einzutragenen **Parameter** können Sie anhand des Idents der Statusvariablen der Instanz ableiten. In der Parameterliste werden einige Beispiele gegeben. Anhand des Typs der Variable können Sie auch die notwendige **HM_WriteValue** Funktion erkennen.

Weitere Datenpunkte können der HomeMatic Datenpunktliste entnommen werden.

Sollte die Fehlermeldung **Failure** auftreten, so ist dies im Normalfall auf eine mangelhafte Funkverbindung zurückzuführen.

## BEISPIEL

```
// Setzt die Anzahl der Alarme auf den Wert 25. Werte erlaubt
zwischen 0 und 255.
```

```
HM_WriteValueInteger(12345, ALARM_COUNT, 25);

// Setzt die Anzahl der Service auf den Wert 78. Werte erlaubt
zwischen 0 und 255.
HM_WriteValueInteger(12345, SERVICE_COUNT, 78);
```

# HM_WriteValueString

```
boolean HM_WriteValueString (int $InstanzID, string $Parameter,
string $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz des zu schaltenden Gerätes |
| Parameter | Parameter welcher gesetzt werden soll. Der Name des Parameters kann in der Datenpunktliste oder als "Ident" im Reiter "Statusvariablen" in IP-Symcon gefunden werden. |
| Wert | String der gesetzt werden soll. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den **Parameter** des Gerätes mit der ID **InstanzID** auf Wert **Wert**.



Den einzutragenen **Parameter** können Sie anhand des Idents der Statusvariablen der Instanz ableiten. In der Parameterliste werden einige Beispiele gegeben. Anhand des Typs der Variable können Sie auch die notwendige **HM_WriteValue** Funktion erkennen.



Weitere Datenpunkte können der HomeMatic Datenpunktliste entnommen werden.



Sollte die Fehlermeldung **Failure** auftreten, so ist dies im Normalfall auf eine mangelhafte Funkverbindung zurückzuführen.

## BEISPIEL

```
//Übermittelt den Wert "Hallo Welt". Beliebige Zeichenkette erlaubt.
HM_WriteValueString(12345, SUBMIT, "Hallo Welt");
```

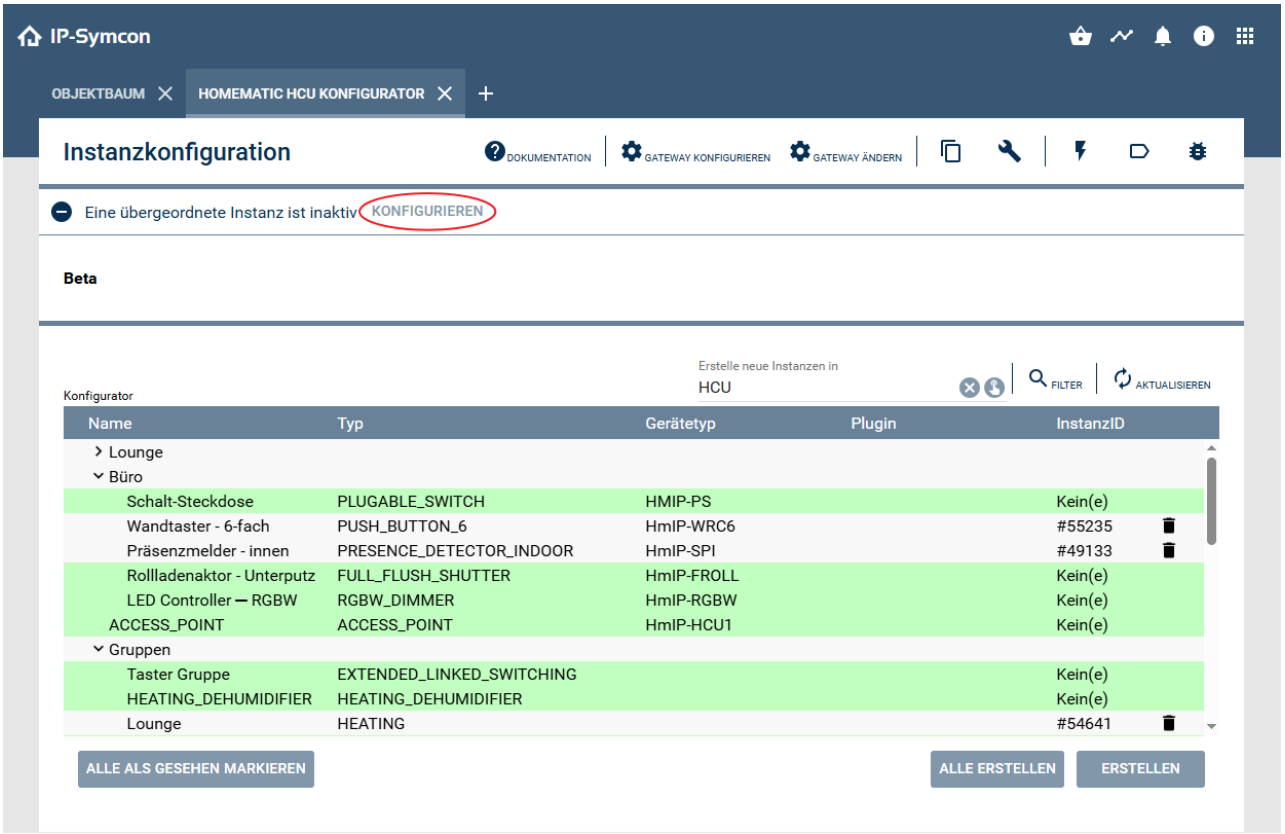
# HomeMatic HCU (Beta)

Benötigt: IP-Symcon >= 8.1

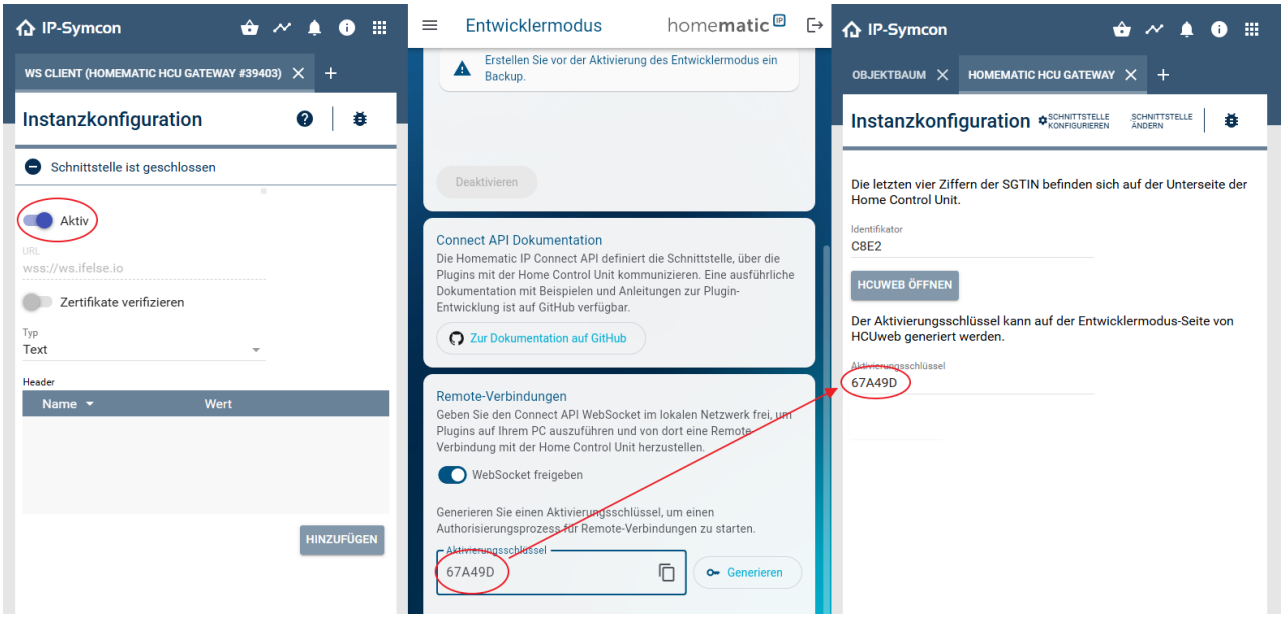
Dieses Modul befindest sich aktuell noch in der Betaphase. Um es nutzen zu können muss der Entwicklermodus über HCUWeb aktiviert werden. Wenn der Modus aktiviert ist, kann eQ-3 keinen technischen Support mehr leisten. Mehr Informationen: Connect API FAQ

## EINBINDUNG IN IP-SYMCON

Die HCU kann über die Gerätesuche eingebunden werden. Hierfür muss als System "HomeMatic HCU Discovery" gewählt werden. Die Discovery-Instanz bietet dann die Erstellung eines HomeMatic HCU Konfigurators an. Nachdem der Konfigurator erstellt wurde, können über diesen die einzelnen Geräte und Gruppen wie unten beschrieben eingebunden werden.

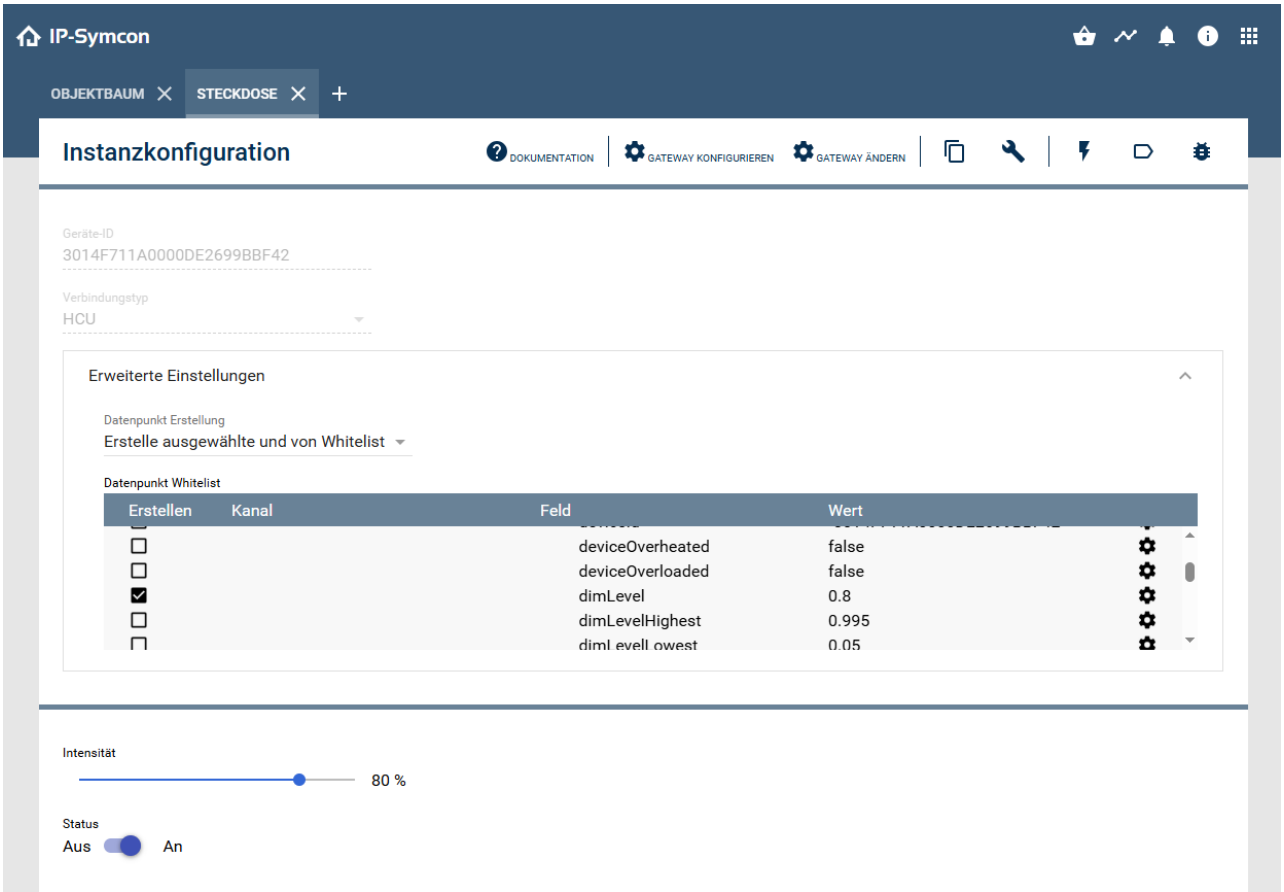



Innerhalb des Konfigurators kann über "Konfigurieren" die Konfigurationsseite des "HomeMatic HCU Gateway" geöffnet werden. Dort muss der Aktivierungsschlüssel aus der HCUWeb eingetragen werden. Hierfür muss in der HCUWeb der Entwicklermodus aktiviert und der WebSocket freigegeben werden. Nachdem der Schlüssel generiert und im Gateway eingetragen wurde, kann die Einrichtung über einen Klick auf "Änderungen übernehmen" abgeschlossen werden.




Sollte zu einen späteren Zeitpunkt die HCU erneut registriert werden sollen, kann ein neuer Aktivierungsschlüssel generiert und in das entsprechende Feld eingegeben werden. Zum Abschluss des Vorgangs muss die Schaltfläche "Erneut Registrieren" in den Expertenoptionen des Gateways betätigt werden.
Im Konfigurator sind jetzt verschiedene Geräte sichtbar und in Räume aufgeteilt. Die Gruppen sind unter einem eigenen Punkt zusammengefasst. Nach der Auswahl eines Geräts oder einer Gruppe kann eine Instanz mit "Erstellen" erstellt und anschließend die Konfigurationsseite über "Konfigurieren" aufgerufen werden.

## GERÄTEKONFIGURATION



Auf der Konfigurationsseite wird neben der Geräte-ID auch der Verbindungstyp angezeigt. Aktuell gibt es nur die Verbindung über die HCU. Die "Erweiterten Einstellungen" erlauben es die Erstellung der Datenpunkte eines Gerätes zu beeinflussen. Standardmäßig ist die Option "Erstelle nur ausgewählte" aktiv. Sollte das Gerät Symcon nicht bekannt sein, werden alle Datenpunkte erstellt.

| OPTION | BESCHREIBUNG |
|---|---|
| Erstelle nur ausgewählte | Wenn das Gerät Symcon bekannt ist, können die erstellten Datenpunkte auf die eingegrenzt werden, die für die Funktion des Gerätes am relevantesten sind. |
| Erstelle nur ausgewählte und von Whitelist | Zusätzlich zu den von Symcon gewählten Datenpunkten können aus einer Liste aller weitere Datenpunkte ausgewählt werden. |
| Erstelle nur von Whitelist | Jeder Datenpunkt, der erstellt werden soll, kann einzeln angewählt werden. |
| Alle erstellen | Es werden alle verfügbaren Datenpunkte erstellt. Diese Option wird gewählt, wenn das Gerät Symcon nicht bekannt ist |

Unten im Aktionsbereich werden alle schaltbaren Werte des Gerätes angezeigt.



Die Konfiguration von Gruppen ist identisch aufgebaut

# Image Grabber

Der Image Grabber speichert von einer bestimmten Adresse Webcam-Bilder. Diese werden automatisch als Medien-Objekte hinzugefügt und stehen somit auch in der Visualisierung zur Verfügung.

Ab IP-Symcon 4.0 werden die vom Image Grabber erstellten Medien nur noch beim Beenden von IP-Symcon auf die Festplatte geschrieben. Durch das intelligente Caching wird die Festplatte/ Flashspeicher geschont, und der Inhalt nur im Arbeitsspeicher vorgehalten. Über IPS_GetMediaContent kann zur Laufzeit auf den aktuellen Inhalt zugegriffen werden.

## EINBINDUNG IN IP-SYMCON

Eine neue Instanz über "Objekt hinzufügen" -> "Instanz hinzufügen" -> Hersteller: "(Sonstige)" einrichten.
Im Konfigurationreiter können folgende Einstellungen vorgenommen werden.

Mit einem MJPG-Stream ist es möglich eine WebCam mit bewegten Bildern anzuzeigen.

Im Gegensatz zu den meisten anderen Instanzen, braucht dieses Modul keine übergeordnete Instanz

## EINSTELLUNGEN

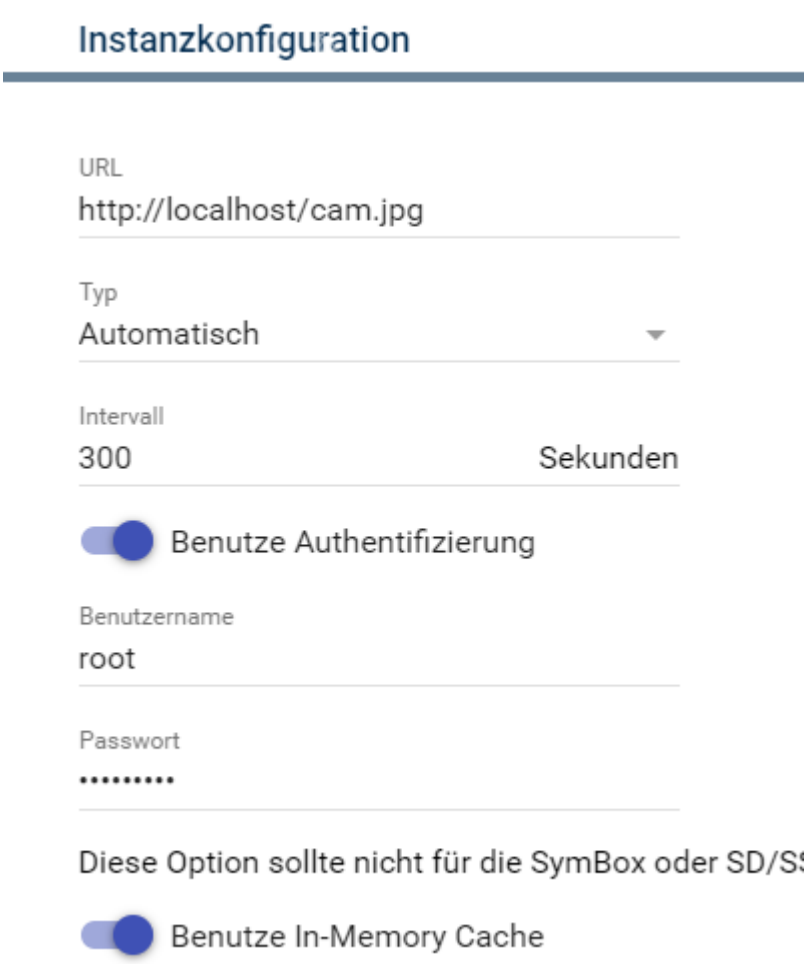
| OPTION | BESCHREIBUNG |
|---|---|
| Adresse | Die Adresse der Webcam und Pfad/URL zum Bild. Siehe Liste unten |
| Typ | In welchem Format das Bild gespeichert werden soll. Sollte die Dateiendung in der Adresse nicht ".jpg", ".gif", ".bmp", ".png" sein, muss das Dateiformat explizit angegeben werden. |
| Intervall | In welchem Zyklus in Sekunden das Bild abgefragt soll |
| Benutze Authentifizierung | Ob eine Authentifizierung genutzt werden soll: Ja/Nein |
| Benutzername | Benutzername zur Authentifizierung |
| Passwort | Passwort zur Authentifizierung |

## ADRESSEN

Unter dem hier aufgeführten Link sind alle gängigen Hersteller/Modelle angegeben.

Unter dem jeweiligen Modell sind auch die Standard-Stream-Adressen zu erkennen, welche gegebenfalls über einen Stream eingebunden werden müssen.
Liste der Hersteller

## BEISPIEL



# IG_UpdateImage

```
boolean IG_UpdateImage (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz des zu schaltenden Gerätes |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Aktualisiert das Bild des Gerätes mit der ID InstanzID.

## BEISPIEL

```
// Das Bild der Instanz 12345 wird aktualisiert
IG_UpdateImage(12345);
```

# IPS-868

Das IP-Symcon Funksystem 868 ergänzt mit seinen bidirektionalen Funkkomponenten Anwendungsbereiche, die von anderen Systemen oft nicht zufriedenstellend abgedeckt werden. Hierzu zählen die Verbrauchserfassung per S0-Eingang, die Anwesenheitskontrolle per AnwesehenheitsKontrollModul mitsamt Tracker, die Analyse der Raumluft via LuftGüteSensor, sowie die Ansteuerung von LED-Stripes.



Folgende Geräte werden von IP-Symcon unterstützt:
Unterstützte Geräte

## INHALT


Installation
Einbindung in IP-Symcon
Konfigurator
Konfiguration


## INSTALLATION

Das **USB Gateway** wird per USB mit dem PC verbunden. Die notwendigen Treiber stehen bei SiLabs zum Download bereit. Nach dessen Installation ist das Gateway einsatzbereit. Das **LAN Gateway** wird per LAN mit dem PC verbunden. Standardmäßig ist DHCP und Port 5000 eingerichtet. Sollte eine individuelle IP Adresse gefordert sein, so wird das IP-Symcon "Network Configuration Tool" benötigt. Dieses steht zum Download bereit. Eine einfache Beschreibung zur Konfiguration des Gateways steht zur Verfügung. Im Anschluss ist das Gateway über die entsprechende IP-Adresse und Port erreichbar.



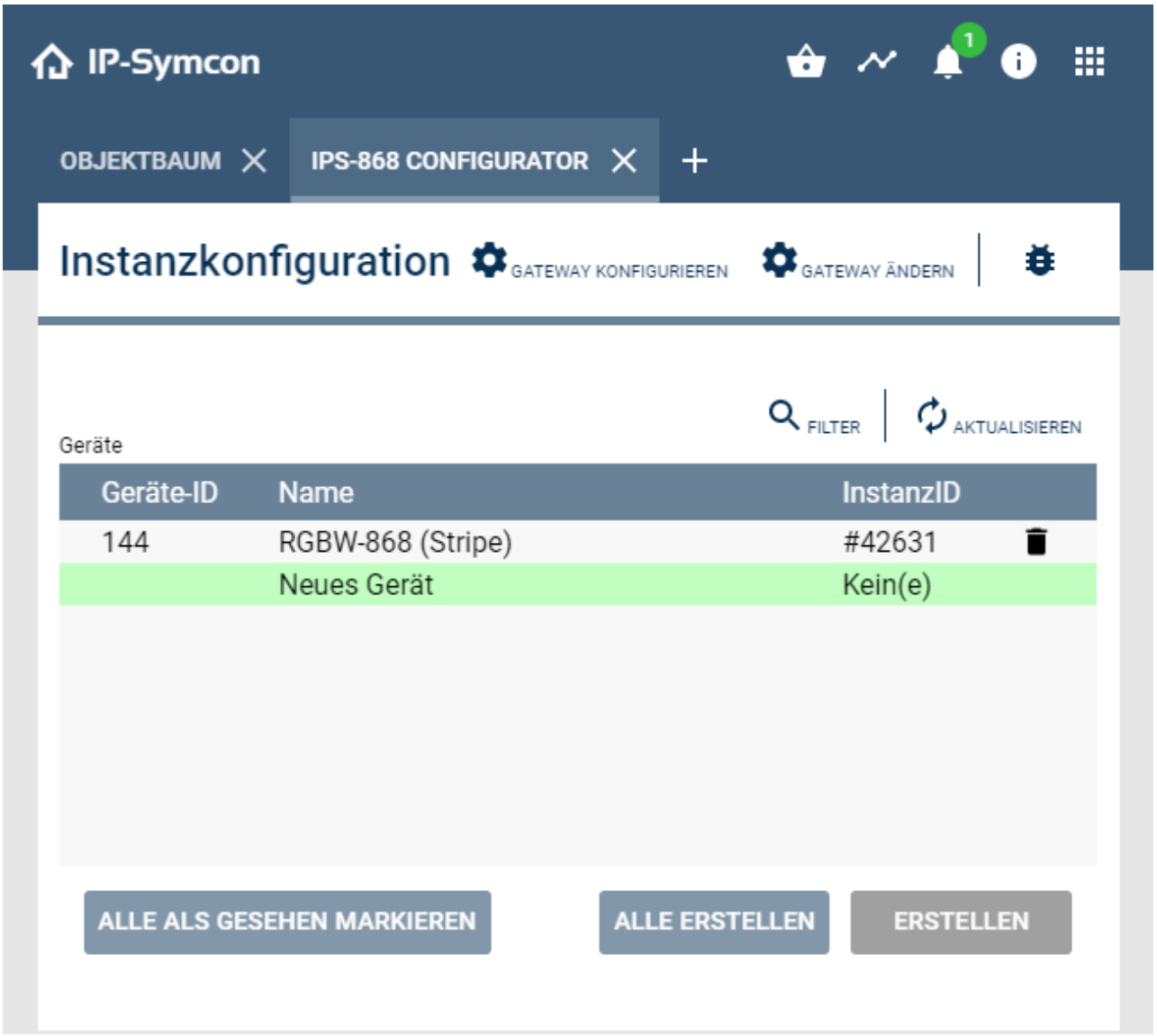
Wenn kein IP-Symcon Haus auf der Vorderseite des LAN-Gateways abgedruckt ist, ist dies ein altes Modell des Gateways. Die Installationsanleitung für dieses Modell kann hier gefunden werden.

## EINBINDUNG IN IP-SYMCON

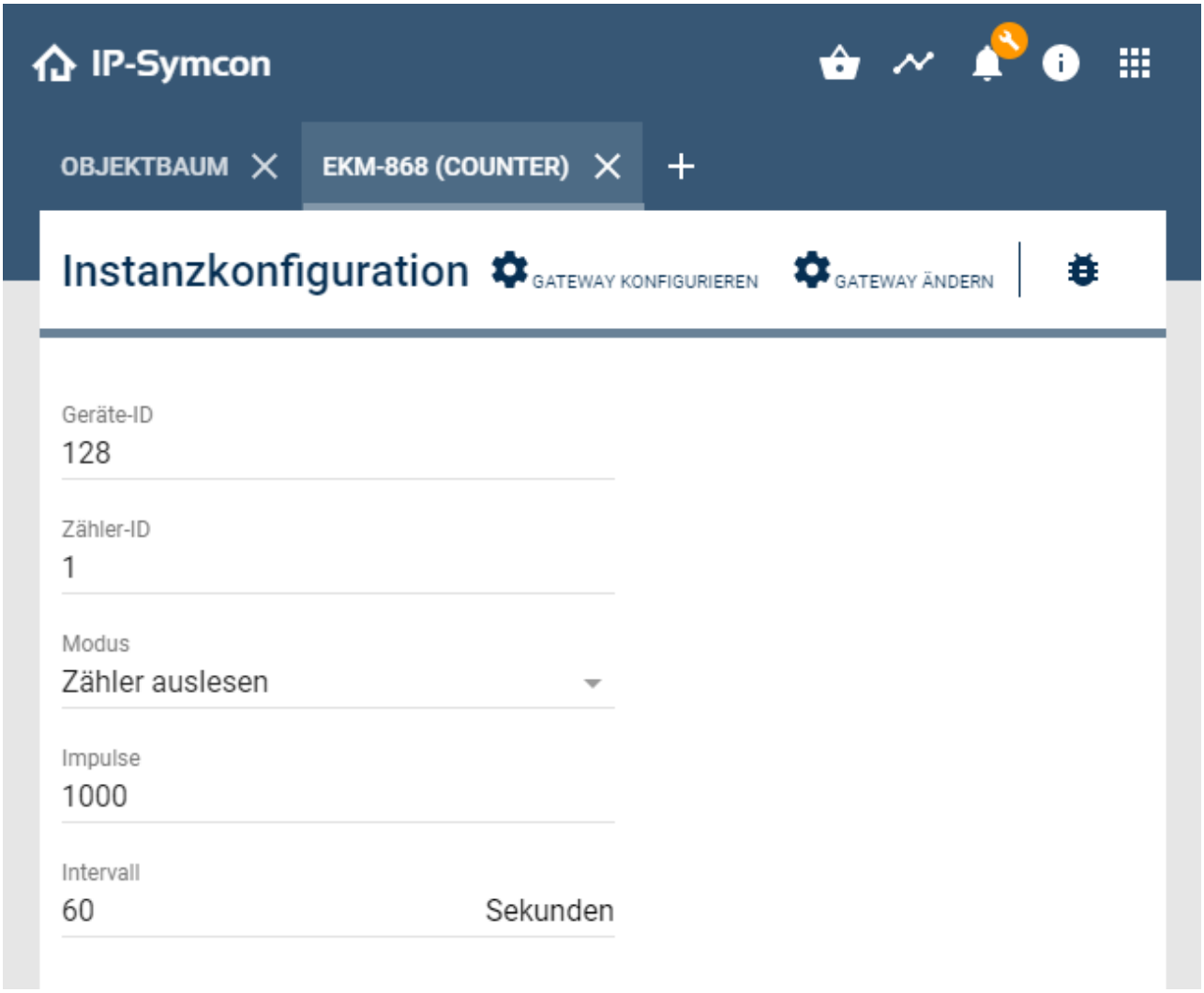
Bei Verwendung des LAN-Gateways kann dies über die Gerätesuche eingebunden werden. Hierfür muss als System "IPS-868 Discovery" gewählt werden. Die Discovery-Instanz bietet dann die Erstellung eines IPS-868 Konfigurators an. Nachdem der Konfigurator erstellt wurde, können dort die einzelnen Geräte wie unten beschrieben eingebunden werden.
Das USB-Gateway wird direkt als Konfigurator angelegt, da es über die Gerätesuche nicht gefunden werden kann. Hierfür wird im Objektbaum ein "IPS-868 Configurator" erstellt. Bei der Konfiguration der Schnittstelle muss der Modus "IPS-868 USB Gateway" gewählt und anschließend der pasende Port ausgewählt und geöffnet werden.

## KONFIGURATOR

Über den IPS-868 Konfigurator können alle IPS-868-Geräte als Instanz erstellt werden. Dazu muss "Neues Gerät" aus der Liste ausgewählt und auf "Erstellen" geklickt werden.




Neue Instanzen werden im Objektbaum in der Hauptkategorie erstellt. Diese erstellten Instanzen können dann entsprechend umbenannt und an anderer Stelle einsortiert werden. Über "Konfigurieren" kann ein neu eingelegtes Gerät konfiguriert werden. Hier muss zum einen die Geräte-ID eingetragen und je nach Gerät möglicherweise noch weitere gereätespezifische Eigenschaften definiert werden. Das genaue Verfahren und die Geräte-ID kann der Dokumentation des jeweiligen Gerätes entnommen werden. (Geräteliste )




## KONFIGURATION

Auf der Konfigurationsseite des Gerätes können verschiedene Eigenschaften eingestellt werden.
Diese können genauer beschrieben den jeweiligen Gerätedokumentationen entnommen werden.

| GERÄTE INSTANZ | GERÄTBESCHREIBUNG |
|---|---|
| ACC-868 | ---------- |
| AKM-868 (Tracker) | AKM-868 |
| ANA-868 (Analog-Ausgang) | ANA-868 |
| ANA-868 (Analog-Eingang) | ANA-868 |
| DATA-868 | ---------- |
| EKM-868 (Counter) | EKM-868 |
| FD-868 (Display Input) | FD-868 |
| FD-868 (Display Output) | FD-868 |
| FD-868 (Display) | FD-868 |
| JKM-868 (LevelJET) | JKM-868-LevelJET |
| JKM-868 (ThermoJET) | JKM-868-ThermoJET |
| LGS-868 (Luftgütesensor) | LGS-868 |
| RGBW-868 (Stripe) | RGBW-868 |
| RGBW-868 (Stripe Input) | RGBW-868 |
| SERVO-868 | SERVO-868 |
| SERVO-868 (Input) | SERVO-868 |
| TFK-868 (Input) | ---------- |
| WDT-868 (WatchDog) | WDT-868 |

# AKM-868

Das Anwesenheitskontrollmodul (AKM-868) registriert die Anwesenheit beliebig vieler Tracker in seiner Nähe.
Dafür verbindet das AKM die Tracker mit IP-Symcon.

## TRACKER

Der Tracker ist eine kleine Fernbedienung im Schlüsselanhängerformat in einem ovalen Gehäuse (Größe: 55 *44* 15 mm). Dieser besitzt eine Taste mit einer Doppelfunktion: kurzer und langer Tastendruck. Somit lassen sich zwei Geräte (z.B. Licht und ein Garagentor) ein- oder ausschalten. Das Besondere des Trackers ist, dass er ca. alle 75 Sekunden einen "Ping" sendet. Dieser besteht aus einer eindeutigen ID, die wiederum von IP-Symcon ausgewertet wird. Befindet sich der Tracker nicht mehr in Reichweite, so wird nach fünf Minuten eine entsprechende Variable auf FALSE gesetzt. Somit können bestimmte Szenarien beim Verlassen oder Betreten der Wohnung gestartet werden.
Die Batterie hält mindestens sechs Monate – nach heutigen Erkenntnissen sogar über ein Jahr (reiner Ping-Betrieb). Die Datenübermittlung erfolgt über das 868 MHz Frequenzband.

## INSTALLATION

Um Daten empfangen zu können, muss zuvor ein 868 Gateway (LAN oder USB) angeschlossen und eingerichtet werden. In IP-Symcon muss "AKM-868 (Tracker)" hinzugefügt werden. Die übergeordneten Instanzen (z.B. das Gateway) werden automatisch angelegt.

## KONFIGURATION

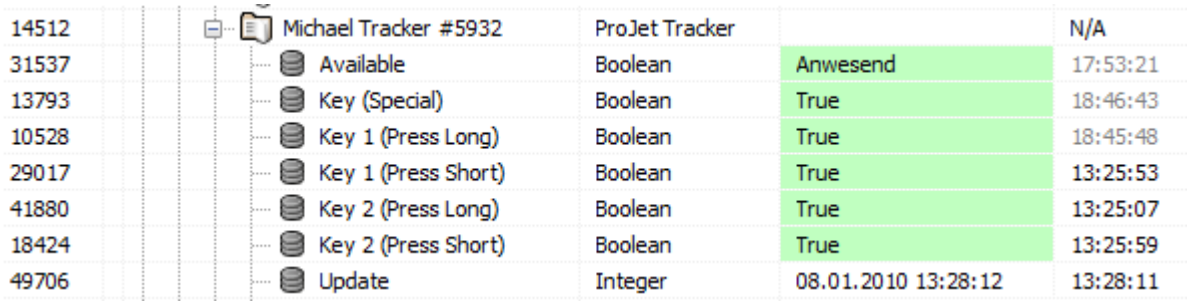
Um das Gerät in IP-Symcon verwenden zu können, muss eine neue AKM-868 Instanz (Hersteller: IP-Symcon / ProJet) hinzugefügt werden:

**GeräteID:**
Standardmäßig wird 0 vorgegeben. Dies bedeutet, dass von allen AKM Daten empfangen werden können. Um gezielt ein Gerät auszuwählen, kann z.B die 160 gewählt werden.
Jumper-Belegung (Blick in das Gerät, Antenne oben, untere 2 Jumper-Paare):
160: beide offen
161: rechter geschlossen
162: linker geschlossen
163: beide geschlossen

**TrackerID:**
Um die ID zu ändern, muss beim einlegen der Batterie die Taste gedrückt gehaltenwerden. Die ID wird von der Tracker automatisch generiert. Diese bleibt auch nach einem Batteriewechsel erhalten.
Meistens ist die TrackerID: 8191 (Vorgabe)

Wird ein "Ping" empfangen, wird "Available" automatisch auf TRUE gesetzt und zusätzlich der Zeitstempel in "Update" aktualisiert. Werden keine "Ping"-Signale über eine Dauer von mehr als fünf Minuten (fest, nicht änderbar) empfangen, wird "Available" auf FALSE gesetzt. Zusätzlich können noch die Tasten-Impulse ausgewertet werden. (Bei "Special" wurden beide Tasten gleichzeitig gedrückt – alte Version: mit zwei Tasten.)

Es ist zu beachten, dass die Key-Variablen immer auf TRUE verbleiben und sich nur der Zeitstempel ändert.



| | | | | |
|---|---|---|---|---|
| 14512 | Michael Tracker #5932 | ProJet Tracker | | N/A |
| 31537 | Available | Boolean | Anwesend | 17:53:21 |
| 13793 | Key (Special) | Boolean | True | 18:46:43 |
| 10528 | Key 1 (Press Long) | Boolean | True | 18:45:48 |
| 29017 | Key 1 (Press Short) | Boolean | True | 13:25:53 |
| 41880 | Key 2 (Press Long) | Boolean | True | 13:25:07 |
| 18424 | Key 2 (Press Short) | Boolean | True | 13:25:59 |
| 49706 | Update | Integer | 08.01.2010 13:28:12 | 13:28:11 |

## TIPPS & TRICKS

Die Empfangsempfindlichkeit des AKM lässt sich auf “HI / LO” einstellen. Dazu muss der Jumper neben dem ATMEL Prozessor gebrückt werden (niedrige Empfangsempfindlichkeit).
Ist die Sendeleistung niedrig (LO) gewählt, kann mit einem zweiten oder dritten AKM die ungefähre Position der Fernbedienung bestimmt werden. Dies ist natürlich stark von den jeweiligen örtlichen Gegebenheiten abhängig. Eine Unterscheidung - wie z.B. Keller, Garten oder Obergeschoss - sollte jedoch möglich sein.
Um die Reichweite der Fernbedienung nochmals deutlich zu erhöhen, kann die SMD-Antenne durch eine Drahtantenne (*Lambda-Viertel: 87mm*) ersetzt werden. Auf Anfrage ist diese bei uns erhältlich.

## PROBLEMLÖSUNG

Wird trotz einer neuen Batterie der Tracker nicht erkannt, dann muss die TrackerID neu vergeben (siehe oben) und anschließend im Tracker-Modul mit “Suchen” neu eingelernt werden.

# ANA-868

Das Analog-Digital/Digital-Analog-Wandler-Modul hat 2 Analog-Eingänge (10V, Eingangsimpedanz 10kOhm) und 2 Analog-Ausgänge (10V/10mA).

## INSTALLATION

Um die Daten von gemessenen Spannungen einlesen oder um Spannungen an den Ausgängen setzen zu können, muss zuvor ein 868 Gateway angeschlossen und eingerichtet werden. In IP-Symcon muss ein "ANA-868 (Analog-Eingang/ Analog-Ausgang)" Modul hinzugefügt werden. Die übergeordneten Instanzen (z.B. das Gateway) werden automatisch angelegt/verbunden.

## KONFIGURATION

| WERT | BESCHREIBUNG |
|---|---|
| Geräte-Adresse | Standardmäßig wird die ID:184 vorgegeben. Sind mehr Controller in einem System vorhanden, muss eine neue ID vergeben werden. Die Jumper-Belegung kann der Bedienungsanleitung entnommen werden. |
| Kanal-Adresse | (1-4) Gibt an welcher der jeweils zwei Ausgangs-/Eingangskanäle benutzt werden soll. |

# PJ_RequestStatus

```
boolean PJ_RequestStatus (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Fragt die aktuellen Werte für das Gerät mit der ID **InstanzID** ab und schreibt diese in die entsprechenden Status-Variablen.



**Diese Funktion belastet den Funkverkehr erheblich.**
Sie sollte ausschließlich verwendet werden, wenn der interne Timer nicht eingesetzt werden soll, um ungleichmäßige Intervalle zu realisieren. Eine ständige Abfrage der Werte blockiert ggf. den Funkverkehr anderer Teilnehmer.

## BEISPIEL

```
//Fragt die aktuellen Werte der Instanz 12345 ab.
PJ_RequestStatus(12345);
```

# PJ_SetVoltage

```
boolean PJ_SetVoltage (int $InstanzID, float $Volt)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |
| Volt | Spannung auf den der Ausgang gesetzt werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt für das Gerät mit der ID **InstanzID** den Ausgang auf die Spannung **Volt**. Höhere Werte als 10V werden automatisch begrenzt.

## BEISPIEL

```
//Setzt die Ausgangsspannung auf 5,30 Volt
PJ_SetVoltage(12345, 5.30);
```

# EKM-868

## BESCHREIBUNG

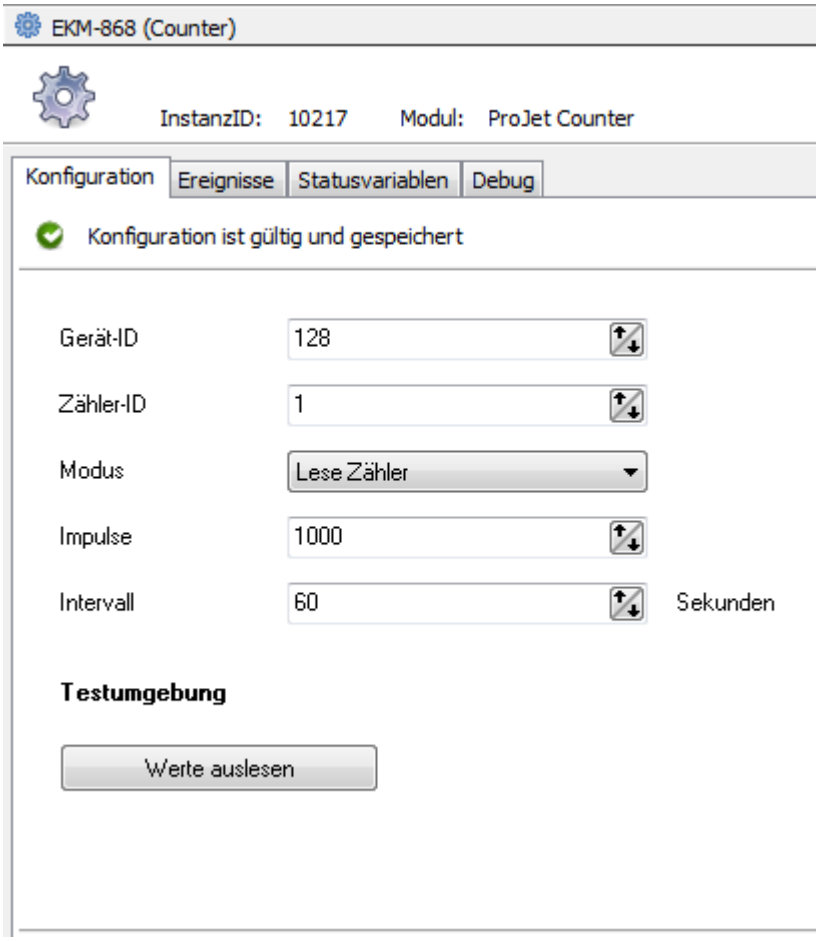
Das Energiekontrollmodul (EKM-868) ist ein 4-Kanal-Zählermodul mit je 32 Bit, das Zählimpulse von vier Eingängen sowie deren Dauer erfasst. Die bidirektionale Datenübermittlung erfolgt über einen USB-Funktransceiver im 868 MHz Frequenzband. Es dient der Visualisierung und Hochrechnung von Verbrauchskosten im Zusammenspiel mit IP-Symcon. Der Zählerstand bleibt solange erhalten, wie eine Spannungsversorgung (9V= Steckernetzteil) besteht. Eine Löschung (Reset) des Zählerstand durch eine Spannungsunterbrechung wird durch IP-Symcon erkannt; eine Batterie-Pufferung ist somit nicht erforderlich. Den eingeschalteten PC benötigt man nur, um einzelne Werte mit einem Zeitstempel zu versehen und Grafiken zu generieren. Eine Besonderheit des EKM 868 ist es, dass zusätzlich die aktuelle Leistung (Verbrauch) erfasst wird.

## INSTALLATION

Um Zählerdaten abfragen zu können, muss zuvor unter Windows der virtuelle COM-Port Treiber installiert werden und der USB-Funktransceiver angeschlossen werden.

## KONFIGURATION

Es muss eine neue Instanz EKM-868 (Hersteller: IP-Symcon / ProJet) erstellt werden:



| WERT | BESCHREIBUNG |
|---|---|
| GeräteID | Standardmäßig wird 128 vorgegeben. Sind zwei EKM in einen System vorhanden, kann im Gerät ein Jumper gesetzt werden: die beiden unteren Pins (Richtung Steckleiste, obere PIN: NC). Die neue ID ist dann 129. ACHTUNG: Sind rote LEDs verbaut (1 bis 4), so ist die ID 129 bei geöffnetem Jumper ! Geräte mit der ID 130/131 usw. sind auf Anfrage erhältlich. |
| ZählerID | Die ID entspricht den jeweiligen Eingängen (Kanälen): 1,2,3,4 |
| Modus: Lese Zähler | Es wird der Zählerstand (Variable: Counter) sowie der zeitliche Abstand in 1/100 Sekunden zwischen zwei Impulsen (Current) ausgelesen. |
| Modus: Lese An-Zeit | Die Einschaltdauer (Kontakt geschlossen) wird in 1/10 Sekunden zurückgegeben (bei 32 Bit ergibt dies eine Zeitspanne von mehr als 13 Jahren!) |
| Modus: Lese Aus-Zeit | Die Ausschaltdauer (Kontakt geöffnet) wird in 1/10 Sekunden zurückgegeben |
| Impulse | Bei Stromzählern werden hier die Impulse pro kWh eingetragen. Die Einheit der Variable "Current" ist dann ebenfalls kWh. Bei einer 0 ist die Einheit 1/100 Sekunde – also ideal um z.B. den Volumenstrom bei Gas- oder Wasser-Zähler zu ermitteln. |
| Timer aktiviert | Bei einem gesetztem Häkchen werden Werte im angegebenen Zeitintervall abgefragt. Diese ist auf ein Minimum von 60 Sekunden limitiert. |
| Test Center | Ein Befehl zur Ausgabe von Werten wird an das EMK gesendet. Bei ordnungsgemäßer Installation werden die Variablen im Objektbaum entsprechend aktualisiert. |



Achtung:
Die Eingänge sind nicht galvanisch getrennt – sie dürfen nicht +5V überschreiten. Bei Draufsicht ist die gemeinsame Masse der vier Eingänge jeweils der rechte Anschluss. An jedem Eingang wurde ein RC-Glied verbaut: 1K als Pull-Up und 100nF an Masse. Ein Schalten z.B. mit einem Reed-Kontakt gegen Masse bewirkt einen Zählimpuls. Die Spannungsversorgung darf =12V nicht überschreiten. Höhere Spannungen oder eine Verpolung führen zu einer Zerstörung des Gerätes. Dann ist eine externe Schutzschaltung (Z-Diode, Vorwiderstand) vorzusehen.

## TIPPS & TRICKS

Der interne Zeitmessung (Current) hat eine Auflösung von 16-Bit und 1/100 Taktung. Dies bedeutet, es können Impulsfolgen von einer Länge von maximal 655 Sekunden (fast 11 Minuten) erfasst werden.
Bei einem Überlauf wird eine -0- zurückgegeben.
Intern wird folgende Formel für die Umrechnung der aktuellen Leistung in Watt verwendet:
P = (3600 *1000) / (Current / 100* Impulses) [Watt]
Wobei "Current" die Zeit zwischen zwei Impulsen in 1/100 Sekunden ist.
Zur Anzeige des aktuellen Energieverbrauchs (Energie-Ampel) haben sich in der Praxis

Intervalle von 20 Minuten als optimal herausgestellt.
Wird der Zähler z.B. durch eine Spannungsunterbrechung gelöscht, so ist dies nicht relevant, da IP-Symcon dies automatisch erkennt und bei der Generierung der Grafiken berücksichtigt.



Problem: LED leuchtet permanent
Lösung: Ggf. die Adern des betreffenden Zählereingangs tauschen (elektronisch)
oder warten bis der Zähler sich weiter dreht und den Reed-Kontakt freigibt (mechanisch).

## TECHNISCHE DATEN

Abmessungen (ohne Antenne): 70 *70* 25 mm
Gewicht: 100g
Stromversorgung: 9V bis 12V DC / 100 mA / Anschluss: DC-Buchse für Hohlstecker 2,1mm, der innere Kontakt der DC-Buchse ist der Pluspol
4 *S0 Eingänge / 4* 32-Bit Zähler (TTL – 5V)
maximale Zähl-Frequenz: 100 Hz
minimale Impulslänge: 20 msec (intern für mechanische Kontakte entprellt)
Schutzklasse: IP 20
Funkmodul: 868MHz (IPS-868 Gateway (LAN oder USB) ist nicht im Lieferumfang enthalten)

# PJ_RequestStatus

```
boolean PJ_RequestStatus (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Fragt die aktuellen Werte für das Gerät mit der ID **InstanzID** ab und schreibt diese in die entsprechenden Status-Variablen.



**Diese Funktion belastet den Funkverkehr erheblich.**
Sie sollte ausschließlich verwendet werden, wenn der interne Timer nicht eingesetzt werden soll, um ungleichmäßige Intervalle zu realisieren. Eine ständige Abfrage der Werte blockiert ggf. den Funkverkehr anderer Teilnehmer.

## BEISPIEL

```
//Fragt die aktuellen Werte der Instanz 12345 ab.
PJ_RequestStatus(12345);
```

# FD-868

## FUNKTION

Das Funkdisplay 868 dient der Anzeige von Text-Meldungen auf einem 2*16-stelligen LC-Display.
Zusätzlich können drei LEDs und ein Tongeber geschaltet werden.
Optional ist eine IO-Box (Relais-Platine) mit zwei Ausgängen (10A/250V) sowie zwei Eingängen (TTL +5V) erhältlich.
Die bidirektionale Datenübermittlung erfolgt über einen USB-Funktransceiver im 868 MHz Frequenzband.
Dabei können alle Tastendrücke ausgewertet werden und zusätzlich die Displaybeleuchtung an/-ausgeschaltet werden.

## INSTALLATION

Um Daten empfangen zu können, muss zuvor unter Windows der virtuelle COM-Port Treiber von Silicon Labs (CP210x) installiert und der USB-T 868 Funktransceiver angeschlossen werden. In IP-Symcon muss "FD-868 (Display)" hinzugefügt werden. Die übergeordneten Instanzen (z.B. das Gateway) werden automatisch angelegt.

## KONFIGURATION

Die folgenden Eigenschaften können wie folgt verändert werden (nicht per Befehl): linke Taste am Display gedrückt halten und Gerät dabei einschalten. Weitere Anweisungen erscheinen auf dem Display.

| WERT | BESCHREIBUNG |
|---|---|
| Geräte-Adresse | Standardmäßig wird die ID:1 vorgegeben. Sind mehr Displays in einem System vorhanden, muss eine neue ID vergeben werden. |
| Gruppen-Adresse | Hier können alle Displays z.B. des Erdgeschoss gemeinsam angesprochen werden. Diese Funktion ist bei der ersten Generation nicht verfügbar. |
| Broadcast-Adresse | Fest: 255 – es werden alle Displays gemeinsam angesprochen |
| Tastenklick | Ein / Aus |



Es darf nur jeweils ein Gerät mit derselben ID in einem Funknetzwerk vorhanden sein.

Optional ist eine externe IO-Box (Relais-Platine) mit zwei Ausgängen (10A/250V) sowie zwei Eingängen (TTL +5V) erhältlich.
RJ-11-Buchse-Belegung (Display):
1 – Eingang 1 / TTL
2 – GND
3 – Ausgang 1 / Open Collector
4 – Ausgang 2 / Open Collector
5 – VCC /+5V
6 – Eingang 2 / TTL
IO-BOX Buchse über der Klemme von ‘Ausgang 1’

## TIPPS & TRICKS

Skripte für eine Menü-Steuerung sind in unserem Forum verfügbar:
Link zum Forenbeitrag

Der Tasten-Klick lässt sich bei der ersten Generation nicht abstellen. Er signalisiert dem Bediener, dass der Befehl vom USB-Funk-Transceiver angenommen worden ist (Quittierung).

# PJ_Backlight

```
boolean PJ_Backlight (int $InstanzID, boolean $Status)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Status | TRUE für An, FALSE für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Schaltet die Displaybeleuchtung im FD-868 mit der ID InstanzID an oder aus.

## BEISPIEL

```
PJ_Backlight(12345, true);
```

# PJ_Beep

```
boolean PJ_Beep (int $InstanzID, int $ZehntelSekunden)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| ZehntelSekunden | Länge des Tones in 1/10 Sekunden |

## RÜCKGABEWERT

default

## BESCHREIBUNG

Gibt ein Tonsignal im FD-868 mit der ID **InstanzID** aus.

## BEISPIEL

```
PJ_Beep($id_lcd, 50); // Beep fuer 0,5 Sekunden
```

# PJ_LCDText

```
boolean PJ_LCDText (int $InstanzID, int $Line, string $Text)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Line | Zeilennummer des Display (1 oder 2) |
| Text | Zeilentext, maximal 16 Zeichen – der Rest wird abgeschnitten |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet einen Text an das FD-868 der ID **InstanzID**.
Da das 'Komma' als Steuerzeichen verwendet wird, darf es im Text nicht vorkommen.
Der o.g. Befehl ersetzt es automatisch in einen 'Punkt'.
Sonderzeichen wie z.B. 'äöü' dürfen nicht vorkommen.

## BEISPIEL

```
PJ_LCDText($id_lcd, 1, "ALARM – Meldung"); // Text in Zeile 1
ausgeben
```

# PJ_SwitchLED

```
boolean PJ_SwitchLED (int $InstanzID, int $LED, boolean $Status)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| LED | 1 |
| Status | True = Ein / False = Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schaltet eine Leuchtdiode vom FD-868 mit der ID **InstanzID** ein oder aus.

## BEISPIEL

```
PJ_SwitchLED($id_lcd,1,False); // Rot
PJ_SwitchLED($id_lcd,3,True); // Gruen
```

# PJ_SwitchMode

```
boolean PJ_SwitchMode (int $InstanzID, boolean $Status)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Relais |
| Status | TRUE für An, FALSE für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Schaltet das Relais mit der ID InstanzID auf Wert Status.

## BEISPIEL

```
PJ_SwitchMode($id_lcd, True); //Relais einschalten
```

# Geräteliste

## UNTERSTÜTZTE GATEWAYS

| GATEWAY | | BESCHREIBUNG |
|---|---|---|
| Symcon LAN-T 868 | Produktinformation | IP Gateway mit LAN-Anschluss |
| Symcon USB-T 868 | | USB Gateway mit USB-Anschluss |

## UNTERSTÜTZTE KOMPONENTEN

| KOMPONENTEN | | BESCHREIBUNG |
|---|---|---|
| AKM-868 | Produktinformation | AnwesenheitsKontrollModul |
| EKM-868 | Produktinformation | EnergieKontrollModul (4-Kanal-Zählermodul) |
| FD-868 | | Funkdisplay |
| JKM-868 LevelJET | | LevelJET |
| JKM-868 ThermoJET | | ThermoJET |
| LGS-868 | Produktinformation | LuftGüteSensor |
| RGBW-868 | Produktinformation | LED-Stripe Controller (12V/24V) |
| SERVO-868 | | 4-fach Servocontroller |
| Tracker-868 | Produktinformation | Schlüsselanhänger (wird in Kombination mit dem AKM-868 genutzt) |
| WDT-868 | | Watch-Dog-Timer |

# JKM-868 LevelJET

## BESCHREIBUNG

Der JKM-868 mit angeschlossenem LevelJET dient dem Auslesen der einzelnen Werte der LevelJET Füllstandsanzeige.

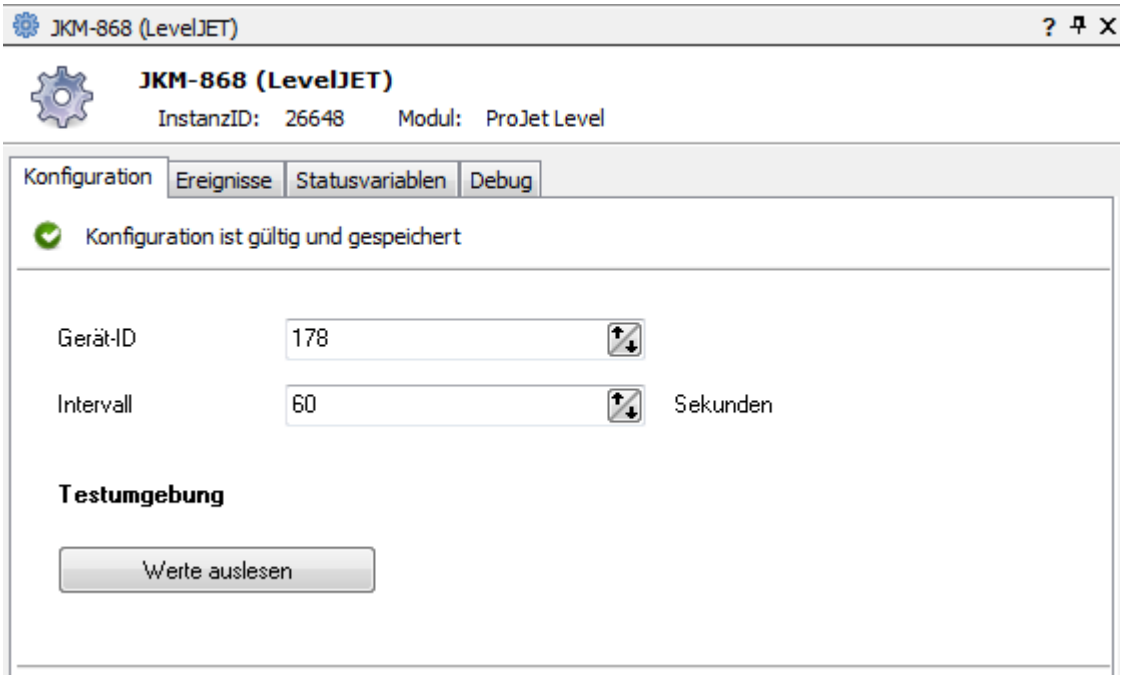
## INSTALLATION

Um Daten empfangen zu können, muss zuvor unter Windows der virtuelle COM-Port Treiber von Silicon Labs (CP210x) installiert und der USB-T 868 Funktransceiver angeschlossen werden. In IP-Symcon muss "JKM-868 (LevelJET)" hinzugefügt werden. Die übergeordneten Instanzen (z.B. das Gateway) werden automatisch angelegt.
Alternativ kann auch das 868-LAN-Gateway verwendet werden. Als "Übergeordnete Instanz" muss ein Client Socket mit den entsprechenden Einstellungen ausgewählt werden.

## KONFIGURATION

Bei der GeräteID wird standardmäßig 166 vorgegeben und muss zu der Adresse des jeweiligen Gerät hinzuaddiert werden. Beispiel: der LevelJET hat die Adresse 12 - also muss hier die 178 eingetragen werden. Die Parametrisierung der Adresse im LevelJET erfolgt gemäß der Anleitung des Herstellers.

| VARIABLE | BESCHREIBUNG |
|---|---|
| Distance | Abstand in cm |
| Output (1) | Zustand des 1. Relais |
| Output (2) | Zustand des 2. Relais |



Der Distance-Wert (Eingang) wird zyklisch im eingestellten „Intervall“ abgefragt. Der Status beider Ausgänge kann nicht aktiv abgefragt werden. Er wird automatisch gesendet, nachdem die im LevelJET eingestellten Distanzschwellen über- bzw. unterschritten wurden.

## TIPPS & TRICKS

Um das Volumen in Liter zu berechnen müssen eigene PHP-Skripte erstellt werden.

# PJ_RequestStatus

```
boolean PJ_RequestStatus (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Fragt die aktuellen Werte für das Gerät mit der ID **InstanzID** ab und schreibt diese in die entsprechenden Status-Variablen.



**Diese Funktion belastet den Funkverkehr erheblich.**
Sie sollte ausschließlich verwendet werden, wenn der interne Timer nicht eingesetzt werden soll, um ungleichmäßige Intervalle zu realisieren. Eine ständige Abfrage der Werte blockiert ggf. den Funkverkehr anderer Teilnehmer.

## BEISPIEL

```
//Fragt die aktuellen Werte der Instanz 12345 ab.
PJ_RequestStatus(12345);
```

# JKM-868 ThermoJET

## BESCHREIBUNG

Der JKM-868 mit angeschlossenem ThermoJET dient dem Auslesen der einzelnen Werte des ThermoJET Temperaturreglers.

## INSTALLATION

Um Daten empfangen zu können, muss zuvor unter Windows der virtuelle COM-Port Treiber von Silicon Labs (CP210x) installiert und der USB-T 868 Funktransceiver angeschlossen werden. In IP-Symcon muss "JKM-868 (ThermoJET)" hinzugefügt werden. Die übergeordneten Instanzen (z.B. das Gateway) werden automatisch angelegt. Alternativ kann auch das 868-LAN-Gateway verwendet werden. Als "Übergeordnete Instanz" muss ein "Client Socket" mit den entsprechenden Einstellungen ausgewählt werden.

## KONFIGURATION

Bei der GeräteID wird standardmäßig 166 vorgegeben und muss zu der Adresse des jeweiligen Gerät hinzuaddiert werden. Beispiel: der ThermoJET hat die Adresse 17 - also muss hier die 183 eingetragen werden. Der ThermoJET besitzt 8 Eingänge - sind mehr als 4 Temperaturführer angeschlossen, so muss eine zweite JKM-Instanz mit der Modus-Einstellung: "Eingang (5-8)" in IP-Symcon hinzugefügt werden.

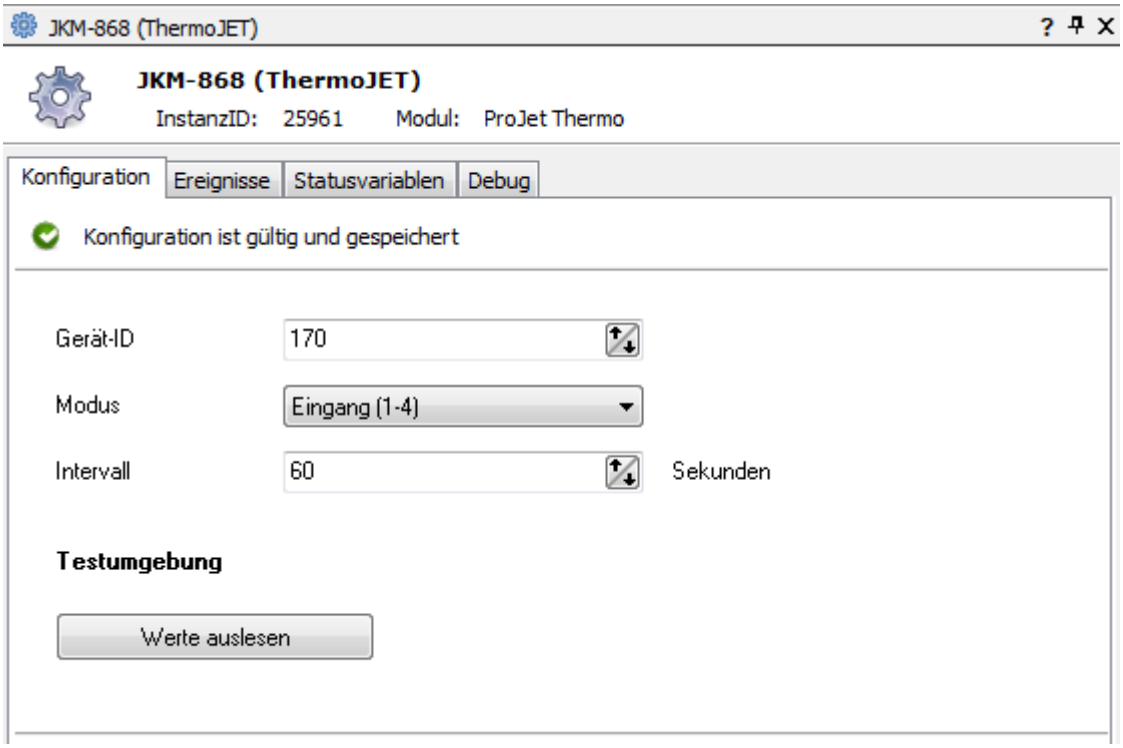
| VARIABLE | BESCHREIBUNG |
|---|---|
| Output (1) | Zustand des 1. Relais |
| Output (2) | Zustand des 2. Relais |
| Temperature(1..4) | Temperatur-Wert in °C |



Hinweis: Ist kein Fühler angeschlossen, wird -99,9°C ausgegeben



Wird eine zweite Instanz für den Eingang 5-8 erstellt, so heißen die neuen Variablen dennoch Temperatur 1-4. Diese können bei Bedarf umbenannt werden. Die Variablen für die Ausgänge 1+2 sind in beiden Modi gleich.




Die vier Temperatur-Kanäle (Eingänge) werden zyklisch im eingestellten „Intervall“ abgefragt. Der Status beiden Ausgänge kann nicht aktiv abgefragt werden. Dieser wird automatisch gesendet, nachdem die im ThermoJET eingestellten Temperaturschwellen über- bzw. unterschritten wurden.

# PJ_RequestStatus

```
boolean PJ_RequestStatus (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Fragt die aktuellen Werte für das Gerät mit der ID **InstanzID** ab und schreibt diese in die entsprechenden Status-Variablen.



**Diese Funktion belastet den Funkverkehr erheblich.**
Sie sollte ausschließlich verwendet werden, wenn der interne Timer nicht eingesetzt werden soll, um ungleichmäßige Intervalle zu realisieren. Eine ständige Abfrage der Werte blockiert ggf. den Funkverkehr anderer Teilnehmer.

## BEISPIEL

```
//Fragt die aktuellen Werte der Instanz 12345 ab.
PJ_RequestStatus(12345);
```

# LGS-868

Der Luftgütesensor (LGS-868) analysiert die Raumluft und fasst die Konzentration von Mischgasen in der Luft zu einem verwertbaren Wert (ppm - parts per million) zusammen.



Technische Hinweise sowie Ratschläge zum richtigen Lüften können in der folgenden Broschüre gefunden werden: Download

## INSTALLATION

Um Daten empfangen zu können, muss zuvor ein IPS-868 Gateway (LAN oder USB) angeschlossen und eingerichtet werden. Die übergeordneten Instanzen (z.B. das Gateway) werden automatisch angelegt.

## KONFIGURATION

Um das Gerät in IP-Symcon verwenden zu können, muss eine neue LGS-868 Instanz (Hersteller: IP-Symcon / ProJet) hinzugefügt werden.
Falls übergeordnete Instanzen noch nicht vorhanden sind, werden diese automatisch angelegt.

| WERT | BESCHREIBUNG |
|---|---|
| Geräte-Adresse | Standardmäßig wird die ID:176 vorgegeben. Sind mehrere Controller in einem System vorhanden, muss eine neue ID vergeben werden. Die Jumper-Belegung kann der Bedienungsanleitung entnommen werden. |



Es darf nur jeweils ein Gerät mit der selben ID in einem Funknetzwerk vorhanden sein.

## ANWENDUNG

Das Gerät liefert einen Integer Wert zurück, welcher die gemessenen ppm (parts per million / millionstel Anteil) zurückgibt. Die offiziellen Grenzwerte können der folgenden Tabelle entnommen werden.
Der LGS-868 besitzt 3 LEDs (grün/gelb/rot), welche automatisch bei den Grenzwerten (450-1000 / 1001-1500 / 1501-2100) von IP-Symcon geschaltet werden. Die Automatik kann abgeschaltet werden, indem "Limit (Gelb)" und "Limit (Rot)" in der Konfigurationsseite auf 0 gesetzt werden.

| PPM | LUFTQUALITÄT |
|---|---|
| >2100 | Sehr Schlecht (stark verschmutzte Raumluft; Lüftung erfolderlich) |
| 2100 ... 1501 | Schlecht (stark verschmutzte Raumluft; Lüftung erforderlich) |
| 1500 ... 1001 | Mittel (verschmutzte Raumluft; Lüftung empfohlen) |
| 1000 ... 801 | Befriedigend |
| 800 ... 601 | Gut |
| 600 ... 450 | Hervorragend |



Extrem hohe Werte über 2100 ppm enstehen selten und beruhen auf z.B. mögliche Alkohole in Putzmitteln oder typische Gerüche auf einer Toilette.

# PJ_RequestStatus

```
boolean PJ_RequestStatus (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Fragt die aktuellen Werte für das Gerät mit der ID **InstanzID** ab und schreibt diese in die entsprechenden Status-Variablen.



**Diese Funktion belastet den Funkverkehr erheblich.**
Sie sollte ausschließlich verwendet werden, wenn der interne Timer nicht eingesetzt werden soll, um ungleichmäßige Intervalle zu realisieren. Eine ständige Abfrage der Werte blockiert ggf. den Funkverkehr anderer Teilnehmer.

## BEISPIEL

```
//Fragt die aktuellen Werte der Instanz 12345 ab.
PJ_RequestStatus(12345);
```

# PJ_SetLEDs

```
boolean PJ_SetLEDs (int $InstanzID, boolean $Grün, boolean $Gelb,
boolean $Rot)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Grün | Schaltet die grüne LED |
| Gelb | Schaltet die gelbe LED |
| Rot | Schaltet die rote LED |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Schaltet die LEDs an dem Controller mit der ID InstanzID.

## BEISPIEL

```
//Schaltet die grüne und gelbe LED ein und die rote LED aus.
PJ_SetLEDs(12345,true,true,false);
```

# RGBW-868

## BESCHREIBUNG

Der LED-Stripe Controller dient der Ansteuerung von LED-Stripes, um unterschiedliche Licht-Szenarien und Stimmungen in Wohnräumen zu ermöglichen.



Bedienungsanleitung als PDF

## INSTALLATION

Um mit dem Controller zu kommunizieren, muss zuvor der virtuellen COM-Port Treiber installiert und der USB-Funktransceiver angeschlossen werden. In IP-Symcon muss "RGBW-868 (Stripe)" hinzugefügt werden. Die übergeordneten Instanzen (z.B. das Gateway) werden automatisch angelegt. Es muss geprüft werden, ob der richtige COM-Port ausgewählt wurde.

## KONFIGURATION

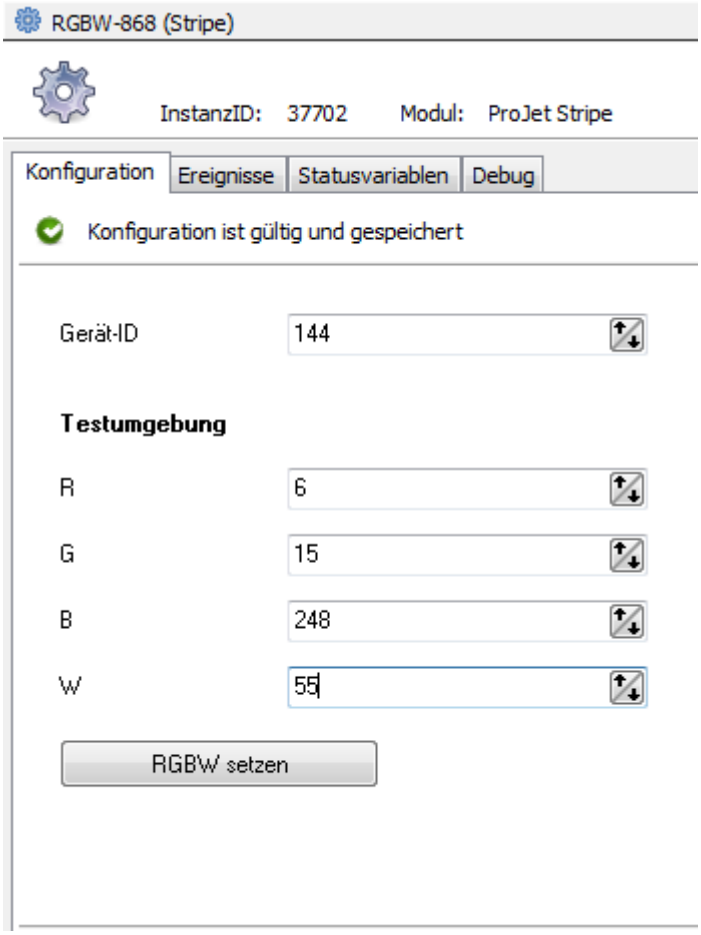
| WERT | BESCHREIBUNG |
|---|---|
| Geräte-Adresse | Standardmäßig wird die ID:144 vorgegeben. Sind mehr Controller in einem System vorhanden, muss eine neue ID vergeben werden. Die Jumper-Belegung kann der Bedienungsanleitung entnommen werden. Hinweis: Ggf. sind die Adressen: 149,150 und 151 vom System belegt. |
| Broadcast-Adresse | Fest: 254 – Es werden alle Controller gemeinsam angesprochen. |



Es darf nur jeweils ein Gerät mit der selben ID in einem Funknetzwerk vorhanden sein.

## EIN ERSTER TEST

Für einen Funktionstest kann in IP-Symcon die Konfigurationsseite mit einem Rechtsklick (oder per Doppelklick) auf die "RGBW-868 (Stripe)" Instanz und anschließend auf "Objekt öffnen" geöffnet werden. Dort kann eine Farbe, bzw. (wenn angeschlossen) ein Helligkeitswert für den Weiß-Kanal gewählt werden. Anschließend muss auf "RGBW setzen" geklickt werden. Die LED-Stripe leuchtet nun in der gewünschten Farbe.




## BEI PROBLEMEN ÜBERPRÜFEN

Beim Einschalten der 12V Spannungsversorgung blitzt der LED-Stripe kurz auf.
Die LED auf der Platine, sichtbar durch die Lüftungsschlitze, leuchtet ebenfalls hell auf und blinkt dann gleichmäßig (sehr) schwach weiter.
Alle Jumper offen bedeutet Adresse 144.
Werden Befehle empfangen, leuchtet die LED zusätzlich.
Der Splitter "ProJet Gateway" muss mit dem richtigen COM-Port (USB-T) bzw. der richtigen IP-Adresse (LAN-T) verbunden sein

## TIPPS & TRICKS

Zusätzlich zu den Lichtstimmungen lassen sich auch z.B. blinkende Warnmeldungen oder blitzende Hinweis ausgeben. Da das Funksystem Bidirektional ist können durch die beiden Tasteingänge des Controllers solche Meldungen quittiert werden. Für diese Funktionalität muss die "RGBW-868 (Stripe) Eingang" Instanz in IP-Symcon hinzugefügt werden.

# PJ_DimRGBW

```
boolean PJ_DimRGBW (int $InstanzID, int $R, int $RZeit, int $G, int $GZeit, int $B, int $BZeit, int $W, int $WZeit)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| R | Helligkeitswert 0 (aus) bis 255 (an, maximale Helligkeit) |
| RZeit | In Sekunden in die der gewünschte Helligkeitswert erreicht werden soll |
| G | Helligkeitswert 0 (aus) bis 255 (an, maximale Helligkeit) |
| GZeit | In Sekunden in die der gewünschte Helligkeitswert erreicht werden soll |
| B | Helligkeitswert 0 (aus) bis 255 (an, maximale Helligkeit) |
| BZeit | In Sekunden in die der gewünschte Helligkeitswert erreicht werden soll |
| W | Helligkeitswert 0 (aus) bis 255 (an, maximale Helligkeit) |
| WZeit | In Sekunden in die der gewünschte Helligkeitswert erreicht werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Dimmt einen **RGB**-Farbwert mit der Helligkeit **W** mit individueller Zeitspanne in Sekunden an dem RGBW-868 Controller mit der ID **InstanzID**.



Der Wert für die Zeit ist intern in zwei Bereiche unterteilt. Werden Zeiten über 255 Sekunden angegeben, werden automatisch alle Zeiten, die kleiner als 255 waren auf 255 gesetzt. Dieser Bereich ist für langsame Szenarien zum Aufwecken oder in Terrarien gedacht.

## BEISPIEL

```
//in 2 Sekunden auf Gelb dimmen
```

```
PJ_DimRGBW(12345,255,2,255,2,0,0,0,0);
```

# PJ_RunProgram

```
boolean PJ_RunProgram (int $InstanzID, int $Programm)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Programm | 0 – alle RGBW auf Aus; 1 – alle RGBW auf den letzten Helligkeits-Wert; 2- Blinken langsam; 3- Blinken schnell; 4- Blitzen (Flash) langsam; 5- Blitzen (Flash) schnell; 6- An- u. Abschwellen (Fade) langsam |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Startet ein Lichtprogramm **Programm** an dem RGBW-868 Controller mit der ID **InstanzID**.

## BEISPIEL

```
//LEDs schnell blinken lassen
PJ_RunProgram(12345, 3);
```

# PJ_SetRGBW

```
boolean PJ_SetRGBW (int $InstanzID, int $R, int $G, int $B, int $W)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| R | Helligkeitswert 0 (aus) bis 255 (an, maximale Helligkeit) |
| G | Helligkeitswert 0 (aus) bis 255 (an, maximale Helligkeit) |
| B | Helligkeitswert 0 (aus) bis 255 (an, maximale Helligkeit) |
| W | Gesamthelligkeitswert 0 (aus) bis 255 (an, maximale Helligkeit) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt einen **RGB**-Farbwert mit der Helligkeit **W** an dem Controller mit der ID **InstanzID**.

## BEISPIEL

```
// Auf Gelb bei voller Helligkeit schalten
PJ_SetRGBW(12345,255,255,0,255);
```

# SERVO-868

## BESCHREIBUNG

Der 4-fach Servocontroller dient der Ansteuerung von handelsüblichen Modellbauservos mit dem RJ-Stecksystem. Als zweite Funktion können bis zu vier Taster als Steuereingänge angeschlossen werden.

## INSTALLATION

Um mit dem Controller zu kommunizieren, muss zuvor das USBT-868 oder LAN-T-868 Gateway installiert worden sein. In IP-Symcon muss die Instanz "Servo-868" hinzugefügt werden. Die übergeordneten Instanzen z.B. das "ProJetGateway" werden automatisch angelegt. Es muss überprüft werden, ob der richtige virtuelle COM-Port vom USB Gateway ausgewählt wurde. Beim LAN-T-868 Gateway muss im "Client Socket" die korrekte IP-Adresse und der Port angegeben werden.

## KONFIGURATION

Als Geräte-Adresse wird standardmäßig die ID:160 vorgegeben. Sind mehr Controller in einem System vorhanden, muss eine neue ID vergeben werden. Die Jumper-Belegung kann der Bedienungsanleitung entnommen werden.

## EIN ERSTER TEST

Für einen schnellen ersten Funktionstest muss das Gerät an eine passende Stromquelle angeschlossen werden. Es ist auf die richtige Polung zu achten. Anschließend wird ein Servo mit Flachbahnkabel und einem Stecker im Rastermaß 2,54 mm mit dem Servo-Ausgang verbunden. In der Regel ist die braune Litze "Minus", rot "Plus" und orange das "PWM" Signal. In IP-Symcon kann die Konfigurationsseite mit einem Rechtsklick (oder per Doppelklick) auf die "Servo-868" Instanz und anschließend auf "Objekt öffnen" geöffnet werden. In der 'Testumgebung' geben kann in dem jeweiligen "Channel" ein Wert zwischen 0 und 255 eingegeben werden. Nach "Set Channel n" bewegt sich der Servo in eine neue Position.

## TIPPS & TRICKS

de.wikipedia.org/wiki/Servo
mikrocontroller.net/articles/Modellbauservo_Ansteuerung
electronicsplanet.ch/Roboter/Servo/intern/intern.htm

# PJ_DimServo

```
boolean PJ_DimServo (int $InstanzID, int $Kanal, int $Wert, int
$Schritte)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Kanal | Servo-Kanal 1..4 |
| Wert | Wert von 0..255 |
| Schritte | Schritte von 0 (sofort) über 1 (langsam) bis 255 (schnell) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt einen Servo-Kanal **Kanal** an dem Controller mit der ID **InstanzID** auf den Wert **Wert** in der Geschwindigkeit **Schritte**.

## BEISPIEL

```
// Setzt sofort den ServoKanal 1 auf den Wert 255
PJ_DimServo(12345, 1, 255, 0);
```

# PJ_SetServo

```
boolean PJ_SetServo (int $InstanzID, int $Kanal, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Kanal | Servo-Kanal 1..4 |
| Wert | Wert von 0..255 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt einen Servo-Kanal **Kanal** an dem Controller mit der ID **InstanzID** auf den **Wert**.

## BEISPIEL

```
PJ_SetServo(12345, 1, 255);
```

# WDT-868

## BESCHREIBUNG

Der Watch-Dog-Timer dient der Überwachung von PC-Systemen bzw. einzelnen Programmen. Hierzu wird zyklisch ein “Lebenszeichen” vom PC an den WDT gesendet. Wird dies nicht korrekt empfangen, kann über einen potentialfreien Kontakt z.B. eine Sirene angesteuert oder ein Hardware-Reset ausgeführt werden.
Zusätzlich kann das UM-Relais (5A/230V) ein- und ausgeschaltet oder über einen Timer (bis 65.535 Sek.) angesprochen werden. Das Gerät verfügt außerdem über einen Tast-Eingang zur freien Verfügung. Dessen Status wird als Variable in IP-Symcon dargestellt.

## INSTALLATION

Um Daten empfangen zu können, muss zuvor unter Windows der virtuelle COM-Port Treiber von Silicon Labs (CP210x) installiert und der USB-T 868 Funktransceiver angeschlossen werden. In IP-Symcon muss "WDT-868" hinzugefügt werden. Die übergeordneten Instanzen (z.B. das Gateway) werden automatisch angelegt.

## KONFIGURATION

Bei der GeräteID wird standardmäßig 165 vorgegeben (Position 0). Diese kann durch Drehen des Codier-Schalters bis 174 (Position 9) verändert werden.




| VARIABLE | BESCHREIBUNG |
|---|---|
| Input | False: Taster offen / True: Taster gedrückt |
| Status | False: Relais stromlos / True: Relais angezogen |
| Timer | Zeit in Sekunden, nach der das Relais seinen Status wechselt |

Nach Ablauf des Timers sendet der WDT den neuen Zustand des Relais. Die Status-Variable wird entsprechend aktualisiert (keine Simulation).
Bei einem aktiven Timer blinkt die LED im Sekundentakt.
Ein kurzes Blitzen signalisiert Funkbefehle im IP-Symcon-868-Funksystem.

Es darf nur jeweils ein Gerät mit derselben ID in einem Funknetzwerk vorhanden sein.

## TIPPS & TRICKS

An die beiden Kontakte das Tasters kann z.B. auch ein externer Schalter (Klingel, Bewegungsmelder, Lichtschranke usw.) angeschlossen werden.

# PJ_SwitchDuration

```
boolean PJ_SwitchDuration (int $InstanzID, boolean $Status, int
$Dauer)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Status | **TRUE** für An, **FALSE** für Aus |
| Dauer | Ein- oder Ausschaltdauer in Sekunden |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schaltet das Relais vom WDT mit der ID **InstanzID** auf Wert **Status** für **Dauer** Sekunden



Das IP-Symcon 868 Funksystem ist bidirektional. Dadurch sendet der WDT den neuen Status des Relais nach Ablauf des Timer.

## BEISPIEL

```
//Der folgende Befehl schaltet das Relais 60 Sekunden ein:
PJ_SwitchDuration(12345, True, 60);
```

# PJ_SwitchMode

```
boolean PJ_SwitchMode (int $InstanzID, boolean $Status)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Status | TRUE für An, FALSE für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Schaltet das Gerät mit der ID InstanzID auf Wert Status.

## BEISPIEL

```
PJ_SwitchMode(12345, True);
```

# IR Trans

IRTrans ist ein Infarotgerät, welches via USB, LAN oder WLAN mit IP-Symcon verbunden wird.

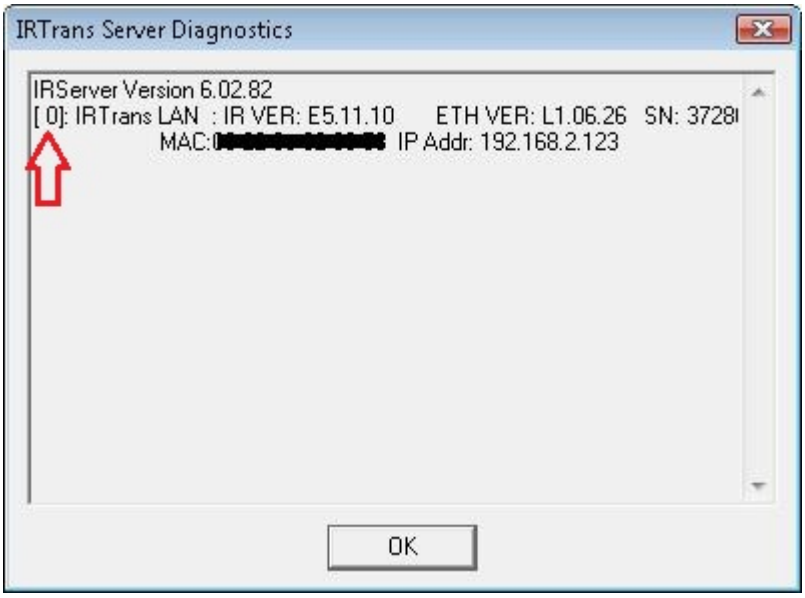
Folgende Geräte werden von IP-Symcon unterstützt:
Unterstützte Geräte

## BESCHREIBUNG

Mit Hilfe von Infrarotsignalen lassen sich existierende Audio- und Videogeräte preiswert und problemlos in die Hausautomation einbinden. Außerdem können z.B. Licht-Szenarien über die Infrarotfernbedienung aktiviert oder die Rollläden heruntergelassen werden. Dazu ist im Prinzip nur eine Sende- und Empfangsdiode notwendig – je nach Wunsch als Fertiggerät oder zum Basteln…
Hersteller: IRTrans: "… macht Ihren PC zu einer programmierbaren Fernbedienung"
irtrans.de/de/index.php

## INSTALLATION

Um IR-Signale über IRTrans empfangen/senden zu können, wird ein IRTrans kompatibles Gerät, wie die USB, LAN oder WLAN Variante, benötigt. Sobald die Geräte innerhalb der IRTrans Tray Software funktionieren, kann die Installation innerhalb von IP-Symcon beginnen. www.irtrans.de/download/setup.exe
Über das IRTrans Tray Icon (Rechte Maustaste -> Diagnostics) können die Geräte IDs abgefragt werden, die von der IRTrans Server Applikation gefunden wurden und die dann über IP-Symcon angesteuert werden können. Das folgende Bild zeigt ein IRTrans LAN Gerät, welches die ID 0 hat.




## EINLERNEN DER FERNBEDIENUNG

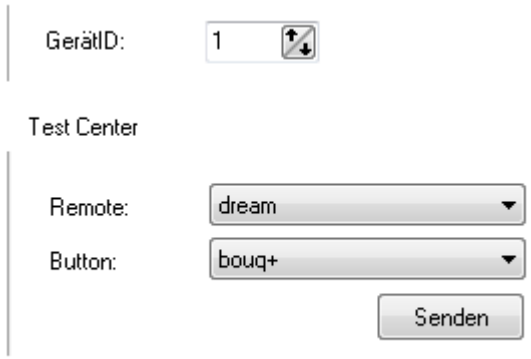
Der Hersteller empfiehlt das Einlernen der Befehle über den ”IRTrans GUI Client”. Dazu muss zunächst der Name der Fernbedienung eingegeben werden (z.B. “sat”) und die betreffende Taste (z.B. “power”). Ein Einlernen über IP-Symcon ist nicht möglich.

## EINBINDUNG IN IP SYMCON




Wenn ein Gerät eine IRDB besitzt, kann statt dem localhost auch direkt die IP-Adresse des Geräts angeben werden. Die Kommunikation mit dem IRTrans Server ist nicht ziwngend notwendig. Anderfalls muss **localhost** angegeben werden!

In der Konfiguration der IRTrans Instanz muss die GeräteID angegeben werden, die aus dem oberen Diagnose Fenster entnommen werden kann. Jede Instanz kann nur ein einzelnes Gerät ansteuern. Es können jedoch beliebig viele IRTrans Instanzen erstellt werden. Wenn später an mehrere Geräte gesendet werden soll, können einfach mehrere PHP Befehle an mehrere Instanzen hintereinander versendet werden. Über das Test-Center kann die Funktion überprüft werden.




Das IRTrans Modul stellt im Objektbaum zwei Variablen bereit, über die auf die aktuell gedrückte Fernbedienung und Taste reagiert werden kann. Dazu muss ein ausgelöstes Ereignis erstellt werden, dass auf die Aktualisierung der “Taste”-Variable reagiert.



# Geräteliste

## UNTERSTÜTZTE GATEWAYS

| GATEWAY | BESCHREIBUNG |
| --- | --- |
| IRTrans Ethernet Modul | IR-Trans LAN-Adapter |
| IRTrans USB Modul | IR-Trans USB-Adapter |
| IRTrans WLAN Modul | IR-Trans WLAN-Adapter |

## UNTERSTÜTZTE KOMPONENTEN

Die Liste der unterstützten Infrarot Geräte kann der Homepage HIER entnommen werden.



Die SymBox unterstützt nur IR-Trans Geräte mit IR Datenbank.

# IRT_SendOnce

```
boolean IRT_SendOnce (int $InstanzID, string $Fernbedienung, string
$Knopf)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Fernbedienung | Name der Fernbedienung, der in der Fernbedienungsdatenbank eingetragen ist/wurde. |
| Knopf | Name der Fernbedienungsfunktion, die in der Fernbedienungsdatenbank eingetragen ist/wurde. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Sendet einen Fernbedienungsbefehl über die Instanz mit der ID InstanzID ab. Die Werte Fernbedienung und Knopf geben an, welcher Befehl gesendet werden soll. Diese müssen in der Client Software (IRTrans Tray) entsprechend eingerichtet worden sein.

## BEISPIEL

```
IRT_SendOnce(37279,"ccf", "v+");
```

# KEBA

KEBA bietet verschiedene Wallboxen/Ladesäulen an. Über ModBus RTU/TCP können diese ausgelesen werden. Eine Verbindung mit IP-Symcon ist über ein LAN oder RS485, je nach Ausführung, möglich.

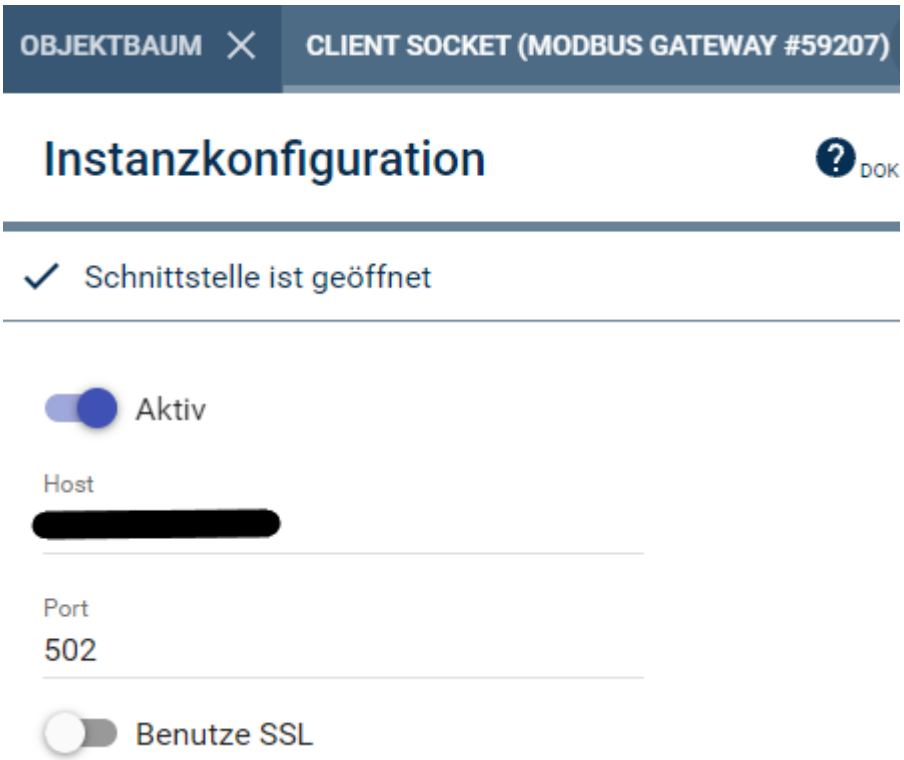
Folgende Geräte werden von IP-Symcon unterstützt:
Unterstützte Komponenten

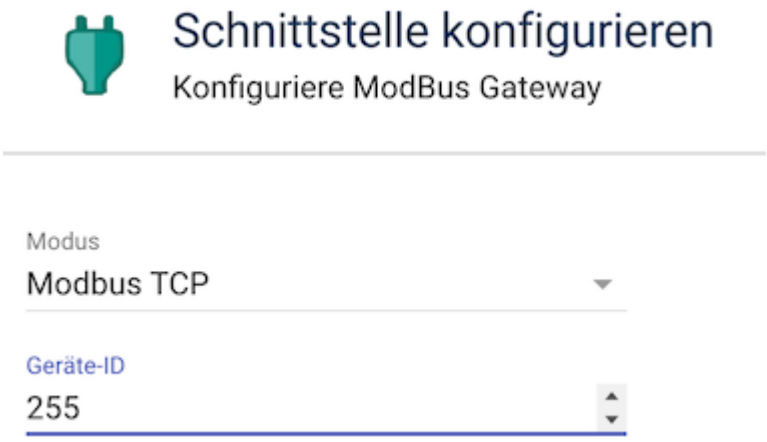
## INSTALLATION

Um KEBA Wallboxen/Ladesäulen in IP-Symcon nutzen zu können, muss eine Verbindung via Ethernet oder RS485 zur Wallbox vorhanden sein.

## EINBINDUNG IP-SYMCON

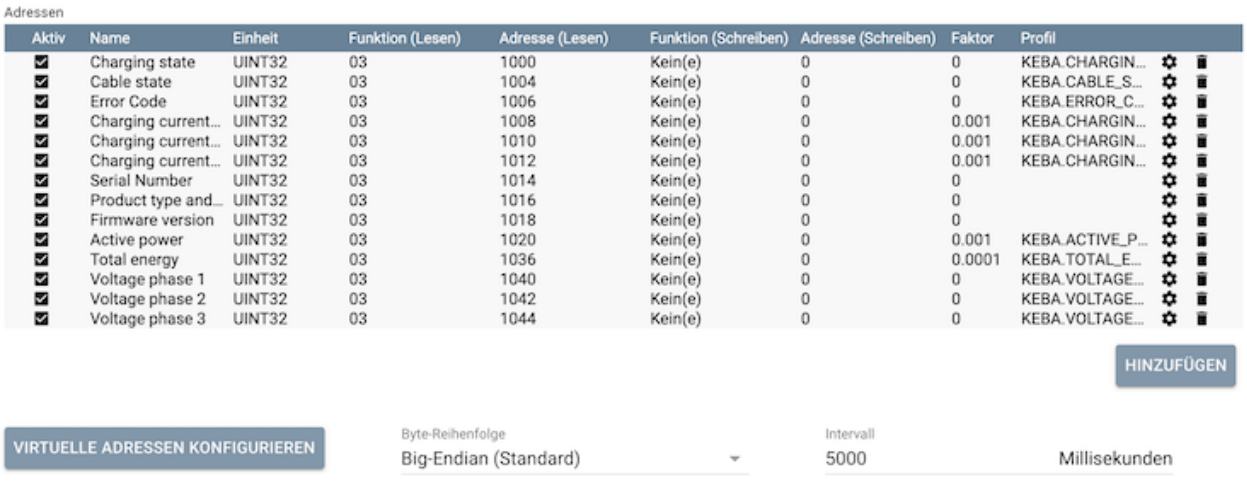
Zuerst muss eine "ModBus Device" Instanz innerhalb des Objektbaums von IP-Symcon hinzugefügt werden. Dabei muss die DeviceID auf 255 gestellt werden und im folgenden Dialog muss die IP-Adresse der Wallbox eingetragen werden. Der Port ist Standardmässig 502. Beide Informationen können auf der Weboberfläche/App der WallBox/Ladesäule eingesehen werden.






Im Anschluss kann die ModBus Vorlage für KEBA Wallboxen/Ladesäulen heruntergeladen werden. Diese beinhaltet die ganze Konfiguration des ModBus Geräts. Nach dem Herunterladen, kann die "KebaModBus_vx.json" über "Importieren" geladen werden. Anschließend sind alle ModBus Adressen der gängigen KEBA Wallboxen/Ladesäulen eingerichtet.

ModBus Vorlage für KEBA Wallboxen/Ladesäulen: Download




## HINZUFÜGEN VON ADRESSEN

Sollen weitere Adressen hinzugefügt werden, kann dies über "Hinzufügen" realisiert werden.
Je nach Ausführung der Wallbox kann es sein, dass einzelne Adressen de-/aktiviert werden müssen. Dies kann über die Aktiv Spalte gesteuert werden.

## PROTOKOLLBESCHREIBUNG

Protokollbeschreibung für KEBA Wallboxen/Ladesäulen:

KEBA KeContact P30: Download

KEBA KeContact P40: Download

# Geräteliste

Benötigt: IP-Symcon >= 7.0

## UNTERSTÜTZTE KOMPONENTEN

| SYSTEM | BESCHREIBUNG |
| --- | --- |
| KEBA KeContact P30 | In allen Versionen unterstützt |
| KEBA KeContact P40 | In allen Versionen unterstützt |

# KNX

KNX ist ein kabelgebundenes sowie funkbasierendes System (KNX-RF). Eine Verbindung mit IP-Symcon ist über ein serielles oder KNX IP Interface möglich. Zur Konfiguration des Systems wird eine Engineering-Tool-Software (kurz "ETS") benötigt.



Folgende Geräte werden von IP-Symcon unterstützt:
Unterstützte Komponenten (Auszug)
KNX Data Secure und KNX IP Secure werden unterstützt!

## INHALTSVERZEICHNIS


Einleitung
Installation
- XML Export in ETS
- Einbindung IP-Symcon
  - Einbindung ohne KNX Data Secure
  - Einbindung mit KNX Data Secure
  - Einzelne Aktoren einrichten
  - Automatische Erkennung der Rückmeldeadressen

Beispiele
- Beispiel: Temperaturwerte auf den Bus senden
- Bessere Rollladensteuerung mit KNX Shutter

Tipps und Tricks
- Umrechnen von Gruppenadressen
- KNX mit QNAP, Synology oder Docker
- Auslesen von Zuständen
- Ausgabe der Gruppenadressen

Legacy
- OPC Export in ETS
- Zuordnung der einzelnen Funktionen zom EIS-Typ/DPT


## EINLEITUNG

KNX ist eine Vernetzung von Haus- und Gebäudesystemtechnik gemäß EN 50090 und ISO/IEC 14543. Es werden gewerkeübergreifend zum Beispiel Heizung, Beleuchtung, Jalousien, Belüftung und Sicherheitstechnik gesteuert. Das System nutzt in der Regel Leitungen (Bezeichnung u.a. J-Y (St) Y 2x2x0,8 EIB bzw. YCYM 2x2x0,8) mit zwei Aderpaaren (rot-schwarz und weiß-gelb). Die KNX-Anlage wird von einer Spannungsversorgung mit 30 V Nennspannung [Gleichspannung] versorgt. Diese Spannung versorgt die Busankoppler, über die jedes KNX-Gerät mit den anderen vernetzten KNX-Geräten kommuniziert. Der Datenaustausch zwischen den KNX-Geräten erfolgt über Telegramme.
Die Programmierung der Teilnehmer und das Zuweisen der Gruppenadressen erfolgt mit der Engineering-Tool-Software (ETS). Die ETS wird von der Dachorganisation Konnex bereitgestellt und sichert die problemlose Zusammenarbeit von Komponenten verschiedener Hersteller.
Die Anbindung an IP-Symcon erfolgt über ein serielles oder IP-Gateway.

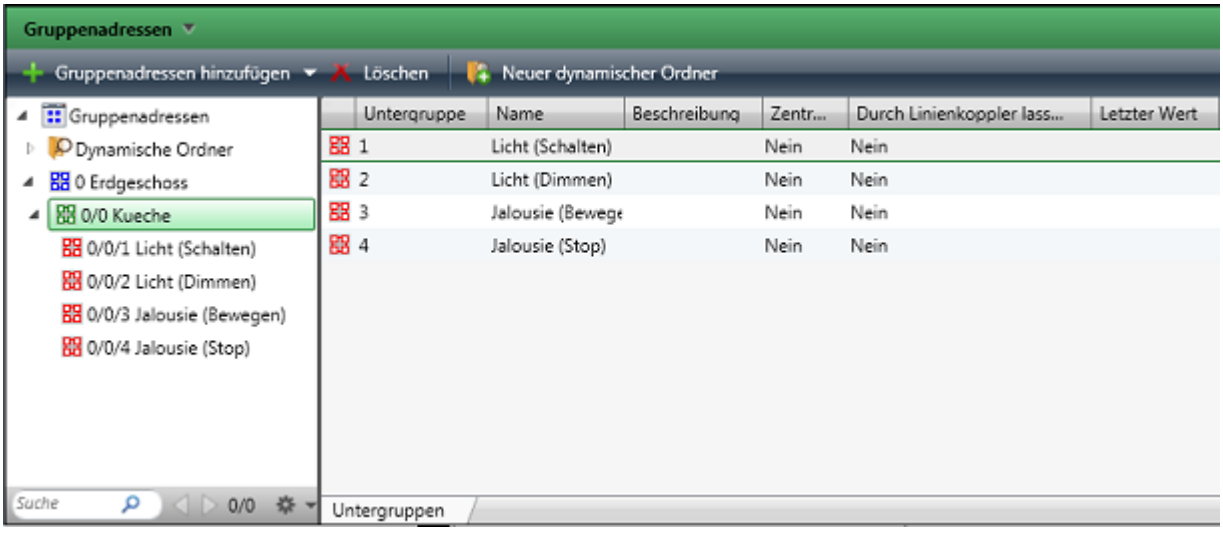
Als Regel gilt, dass alle seriellen Gateways “PEI Typ 10” mit FT1.2 Protokoll (Interfaces des Typs PEI 16 funktionieren nicht) unterstützt werden und alle KNX/IP Gateways, die das Tunneling Protokoll anbieten. Sollte ein KNX Gateway diese Voraussetzungen erfüllen und nicht funktionieren, ist der Symcon Support zu kontaktieren. Das Problem wird so schnell wie möglich gelöst.
Seit IP-Symcon 6.4 wird ebenfalls KNX Data Secure und KNX IP Secure vollständig unterstützt. Zusätzlich unterstützen alle KNX IP Secure Interface auch das neue KNX/IP over TCP, welches wesentlich robuster als das ältere KNX/IP over UDP ist.

## INSTALLATION

Wenn alle KNX-Geräte korrekt aufgestellt und angeschlossen sind, müssen diese mit der ETS konfiguriert werden. Dazu wird die dreistufige Adressvergabe genutzt.

> Zur Zeit werden nur dreistufige Adressen (z.B. 10/2/5) unterstützt. Zweistufige oder freie Adressen unterstützt IP-Symcon zur Zeit nicht. Über die Umrechnung kann jede zweistufige Adresse über die übersetzte dreistufige Adresse Angesteuert werden.

Falls die Installation durch einen Elektriker vorgenommen wurde, so sollte dieser den nachfolgend erklärten OPC/XML Export und die damit verbundene Datei vom Typ XML mitliefern.
Um KNX in IP-Symcon nutzen zu können, ist entweder ein serielles FT1.2 Gateway oder ein kompatibles KNX/IP Gateway notwendig. Die Einrichtung der verschiedenen Gateways unterscheidet sich geringfügig. Es kann direkt eine KNX Instanz in IP-Symcon erstellt und von dieser aus das Gateway konfiguriert werden. Alternativ kann der XML Export aus der ETS genutzt werden.

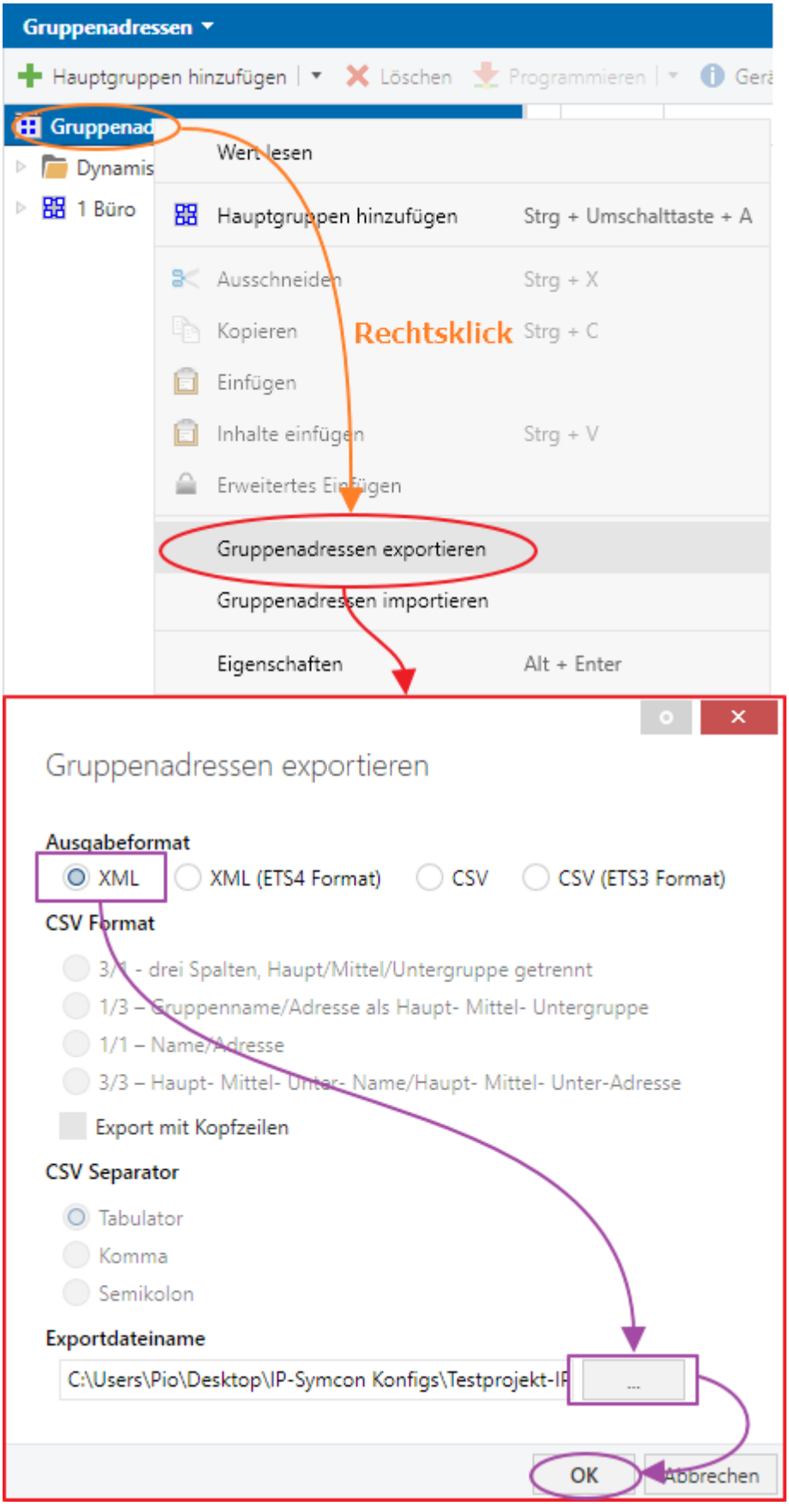



Die Nutzung des XML Exports mit Hilfe des IP-Symcon Konfigurators erlaubt eine schnelle Einbindung aller KNX Komponenten.

> Ab IP-Symcon 5.3 wird der Export als XML empfohlen, da dieser die neuen KNX DPT-Instanzen unterstützt.

## XML EXPORT IN ETS

Ein XML-Export kann innerhalb der ETS im geöffneten Projekt via Rechtsklick auf die Gruppenadressen gespeichert werden.



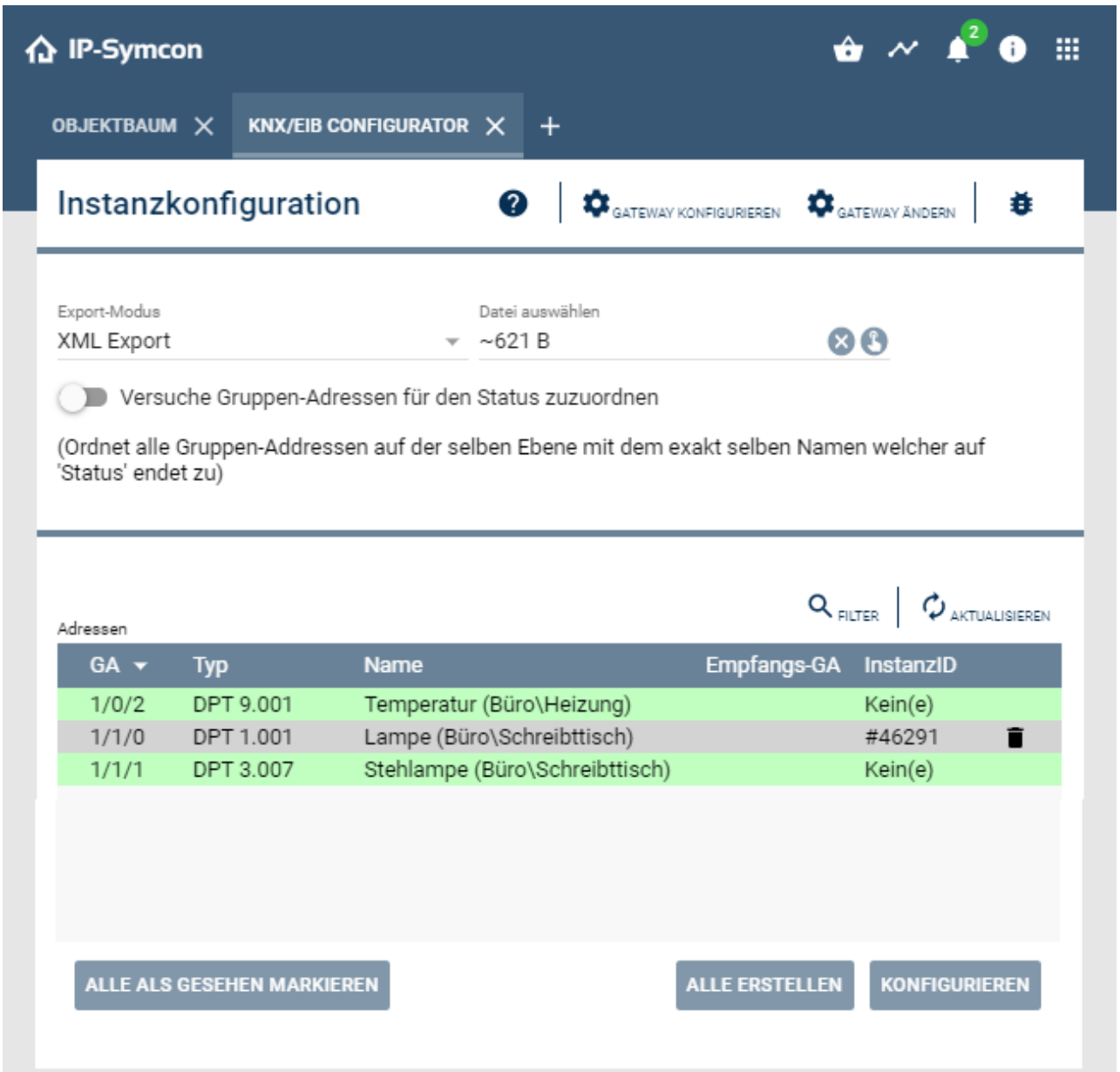

Videoanleitung zum Export:

Ab Version 6.0.0

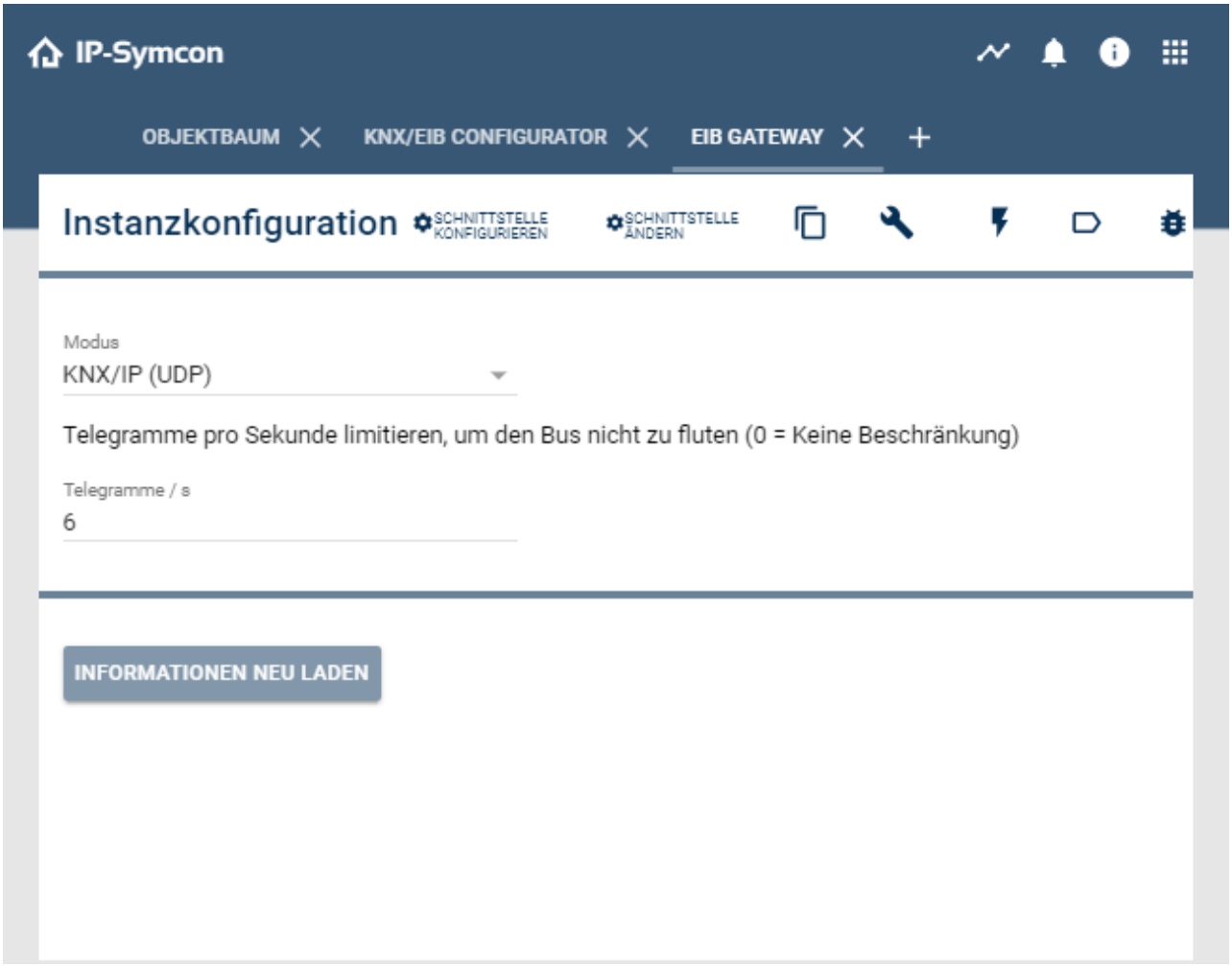
Ab Version 5.6.5

## EINBINDUNG IN IP-SYMCON

Die KNX Schnittstelle kann über die Gerätesuche eingebunden werden. Hierfür muss als System "KNX Discovery" gewählt werden. Die Discovery-Instanz bietet dann die Erstellung eines KNX Konfigurators an. Nachdem der Konfigurator erstellt wurde, können über diesen die einzelnen Geräte wie unten beschrieben hinzugefügt werden.




Bevor mit dem Import der Gruppenadressen begonnen werden kann, muss die Schnittstelle korrekt ausgwählt und konfiguriert werden. Dazu muss im oberen Bereich der Konfiguration auf das "Zahnrad" geklickt werden. Es sollte sich nachfolgender Dialog öffnen.




Wenn KNX Data Secure genutzt werden soll, kann im Kapitel KNX Data Secure weitergelesen werden.

## EINBINDUNG OHNE KNX DATA SECURE

Unter Modus muss der Typ des Gateways ausgewählt werden. Für alle gängigen IP-Gateways kann der Modus "KNX/IP (UDP)" ausgewählt werden. Für die KNX IP Interfaces mit Secure Unterstützung kann auch der zuverlässigere Modul "KNX/IP (TCP)" gewählt werden oder das passende Secure equivalent. Änderungen werden mit dem Button "Übernehmen" gespeichert werden. Um für die Schnittstelle die Adresse eintragen zu können, muss erneut auf das "Zahnrad" im oberen Bereich geklickt werden. Je nach Modus muss entweder die IP-Adresse eingetragen oder der serielle Port der zu der Schnittstelle gehört eingetragen werden. Im Zweifelsfall kann die IP-Adresse auch mit Hilfe der ETS herausgefunden werden. Weitere Einstellungen sind nicht notwendig. Die Änderungen werden mit "Übernehmen" bestätigt. In der Konfiguration des Gateways können nun über "Informationen neu laden" die Parameter des Gateways ausgelesen werden. Wenn diese korrekt Ausgelesen wurden, ist die Konfiguration vollständig. Videoanleitung:

## EINBINDUNG IN IP-SYMCON MIT KNX IP SECURE / KNX DATA SECURE

Um KNX IP Secure und/oder KNX Data Secure in IP-Symcon zu nutzen, wird zusätzlich zu

der XML Datei ein Backup von einem Schlüsselbund benötigt.
Videoanleitung zum Exportieren des Schlüsselbundes:

Im KNX Gateway muss der Modus "KNX/IP Secure (TCP)" ausgewählt werden. In das Feld "Schlüsselbund auswählen" wird die Schlüsselbund-Datei importiert und das Kennwort in das Feld "Schlüsselbund Kennwort" eingeben. Änderungen werden mit dem Button "Übernehmen" gespeichert werden. Nun erscheint das Feld "Sicherer Tunnel" über welchem der konfigurierte Tunnel des Interfaces ausgewählt werden kann. Um für die Schnittstelle die Adresse eintragen zu können, muss erneut auf das "Zahnrad" im oberen Bereich geklickt werden. Hier muss die IP-Adresse eingetragen gehört eingetragen werden. Im Zweifelsfall kann die IP-Adresse auch mit Hilfe der ETS herausgefunden werden. Weitere Einstellungen sind nicht notwendig. Die Änderungen werden mit "Übernehmen" bestätigt. In der Konfiguration des Gateways können nun über "Informationen neu laden" die Parameter des Gateways ausgelesen werden. Wenn diese korrekt Ausgelesen wurden, ist die Konfiguration vollständig.
Videoanleitung für KNX Data Secure mit KNX IP Secure Interface:

Videoanleitung für KNX Data Secure mit SymBox mit KNX Erweiterung:

## IP-ADRESSE HERRAUSFINDEN

## EINZELNE AKTOREN EINRICHTEN

Im KNX Konfigurator können nach dem Hochladen der XML-Datei alle zuvor konfigurierten KNX-Aktoren in IP-Symcon eingebunden und visualisiert werden.
Dazu müssen Geräte durch drücken von "Erstellen" erstellt werden. Die Geräte sind jetzt bedienbar.
Bei der danach erscheinenden ID auf der rechten Seite handelt es sich um die von IP-Symcon zugeordnete InstanzID. Diese ist einmalig und unveränderbar.
Nach einem Klick auf ein ausgewähltes KNX-Gerät und dann einen Klick auf “Konfigurieren” kann die Funktionalität jedes einzelnen Gerätes getestet werden.
Die Geräte sind im Objektbaum wieder zu finden.
Über die Visualisierung können nun die erstellten Geräte geschaltet werden.

## AUTOMATISCHE ERKENNUNG VON RÜCKMELDEADRESSEN

Wenn ein Gerät eine Rückmeldeadresse eingerichtet hat, kann diese automatisch erkannt und als Rückmeldeadresse gesetzt werden. Dazu muss im KNX Konfigurator auf den Button "Erkenne Rückmelde-GA automatisch" geklickt und über den Punkt "Aktiviere Erkennung" muss dies aktiviert werden.
Danach muss ausgewählt werden über welche Ebenen die Gruppenadressen verknüpft werden sollen. Liegt die Rückmeldeadresse und das Gerät in der selben Mittelgruppe, so wird der Punkt "Verknüpfe auf der selben Ebene (1/1/x)" ausgewählt. Liegen Rückmeldeadresse und Gerät in unterschiedlichen Mittelgruppen so wird der Punkt "Verknüpfe über zwei Ebenen (1/x/x)" ausgewählt. Liegen die Adressen in verschiedenen Hauptgruppen wird der Punkt "Verknüpfe auf allen Ebenen (x/x/x)" ausgewählt.
Die Verknüpfung der Gruppenadressen erfolgt über die angegebenen Schlüsselwörter. Ist der Name einer Rückmeldeadresse nach Entfernen des Schlüsselwortes identisch mit dem Namen einer Gruppenadressen so werden diese miteinander verknüpft. Die Groß-/Kleinschreibung wird dabei ignoriert. Die Rückmeldeadresse wird dann als

Rückmeldeadresse der Gruppenadresse gesetzt.
Video zur Einrichtung mit automatischen Rückmeldeadressen:

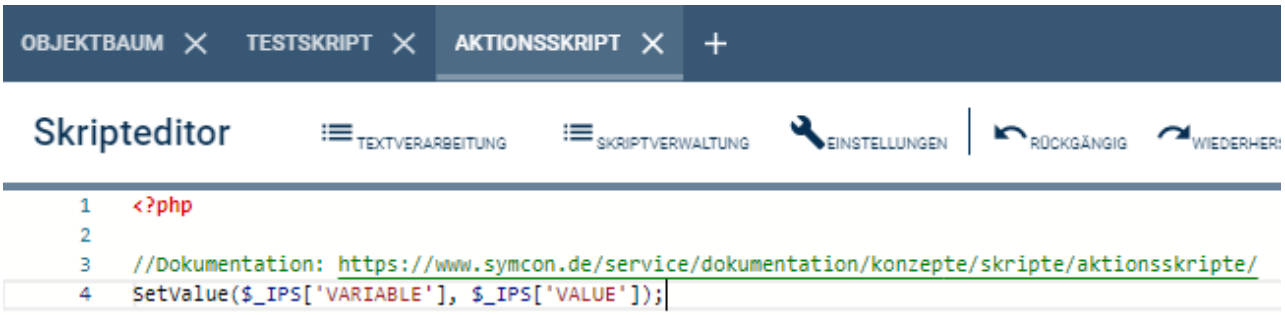
## BEISPIELE

### TEMPERATURWERTE AUF DEN KNX-BUS SENDEN

Um Werte wie z.B. die Temperatur auf den KNX-Bus senden zu können, ist ein Aktionsskript erforderlich. Dieses bildet die Verknüpfung zwischen der Visualisierung und den Aktoren. Es muss ein Skript mit folgendem Inhalt erstellt werden:

```
EIB_Value(IPS_GetParent($_IPS['VARIABLE']), $_IPS['VALUE']);
```

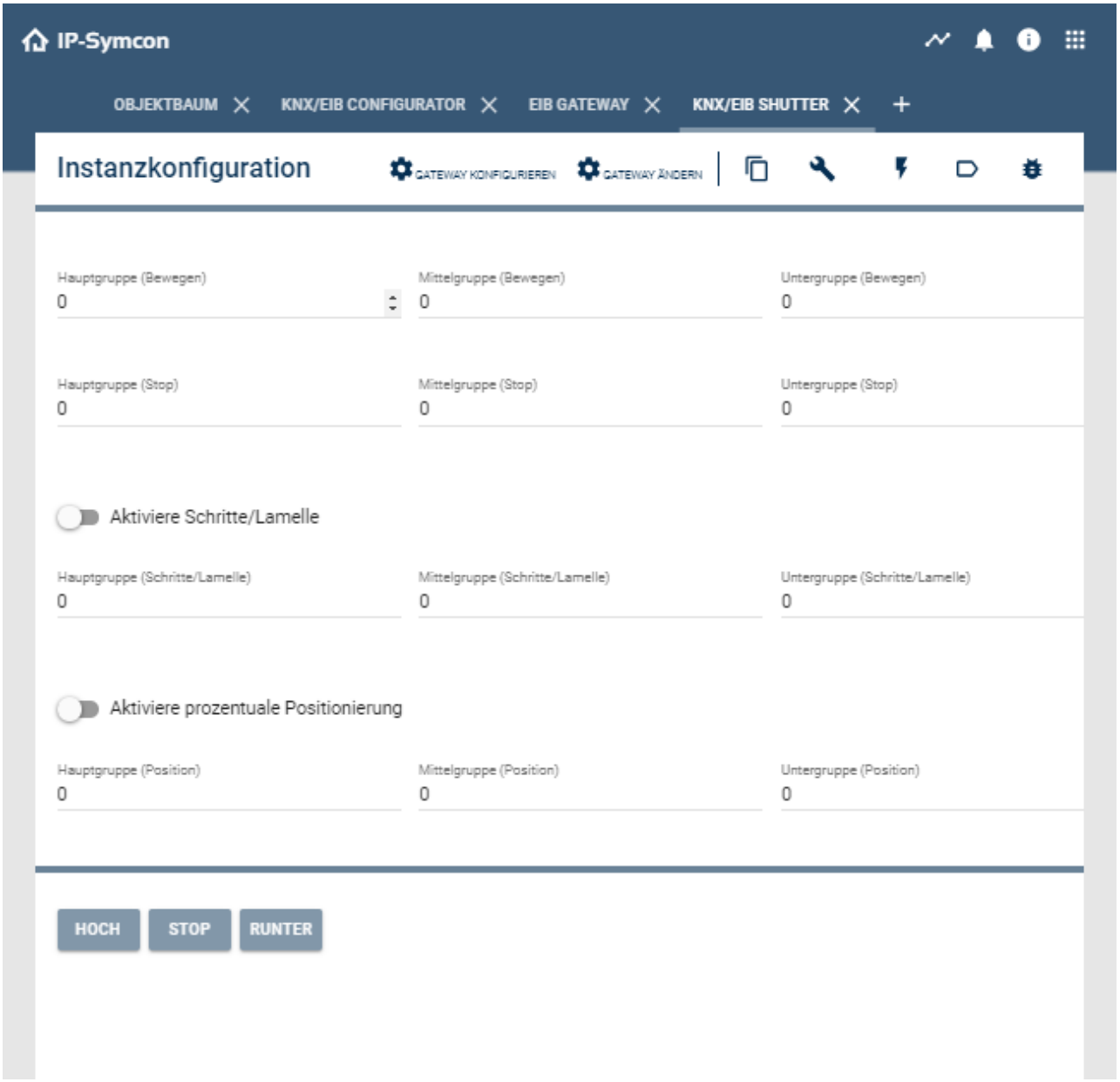
Für jedes gewünschte Gerät, dass veränderbar sein soll, muss die entsprechende Variable mit diesem Aktionsskript verknüpft werden. Dies muss in der Konfiguration der jeweiligen Variable geschehen. Dazu muss die Variable doppelt angeklickt werden. Im Bereich "Profileinstellungen" befindet sich die Auswahl des Aktionsskripts. Hier muss das eben erstelle Skript ausgewählt werden. Über "Eigenes Profil" sollte zusätzlich ein passendes Profil (z.B. ~Temperature) ausgewählt werden.




Der letzte Schritt ist beliebig oft wiederholbar. Es ist nur erforderlich, das Skript einmalig mit den o.g. Inhalt anzulegen.

### BESSERE ROLLLADENSTEUERUNG MIT KNX SHUTTER

Aufgrund der vielen Einstellungsmöglichkeiten und verknüpften Adressen können Shutter nicht über den Konfigurator hinzugefügt werden.




## BEWEGEN

Hier muss die Adresse für die Auf- und Abbewegung des Rollladen eingetragen werden. Im Bereich "Mehr?" können weitere Adressen eingetragen werden, auf welche die Instanz zum Auf-/Abbewegen hören soll.

## STOP

Hier muss die Adresse für das Stoppen des Rollladen eingetragen werden. Im Bereich "Mehr?" können weitere Adressen eingetragen werden, auf welche die Instanz zum Stoppen hören soll.

## AKTIVIERE SCHRITTE/LAMELLE

Diese Adresse kann zugeschaltet werden und steuert das Schrittweise bewegen des Rollladen. Dies dient dazu, die Lamellen des Rollladen zu verstellen. Im Bereich "Mehr?" können weitere Adressen eingetragen werden, auf welche die Instanz zur

Schrittsteuerung hören soll.

### AKTIVIERE PROZENTUALE POSITIONIERUNG

Diese zwei Adressen können zugeschaltet werden und steuern entweder das prozentuale Positionieren des Rollladen oder der Lamellen.

### TEST CENTER

Im Test Center können grundlegende Funktionen adhoc getestet werden.

## TIPPS UND TRICKS

Ist die ETS zwingend notwendig?
Uhrzeit mit KNX Synchronisieren

### UMRECHNEN VON GRUPPENADRESSEN

Umrechnen von zweistufigen Gruppenadressen zu dreistufigen Gruppenadressen
Falls in der ETS eine zweistufige Darstellung der Gruppenadressen gewählt wurde, können die Adressen mit Hilfe dieses Tools umgerechnet werden. Wenn der OPC-Export genutzt wird, übernimmt die ETS diesen Prozess automatisch.

Umrechnen >>>

Umrechnen von freien Gruppenadressen zu dreistufigen Gruppenadressen
Falls Sie in der ETS die freie Darstellung der Gruppenadressen gewählt wurde, so ist leider ein OPC-Export nicht möglich. Mit Hilfe dieses Tools können die Adressen umgerechnet werden.

Umrechnen >>>

### KNX MIT QNAP, SYNOLOGY ODER DOCKER

Soll KNX mit QNAP, Synology oder Docker verwendet werden, so muss der NATSupport eingerichtet werden um Antworten des Systems zu erhalten. Dieser Vorgang ist hier erklärt.

### SKRIPT ZUM AUSLESEN DER ZUSTÄNDE VOM BUS

Da nur Variablenänderungen von IP-Symcon wahrgenommen werden ist beim Neustart von von IP-Symcon abzufragen ob es Statusänderungen gab, um eine konsistente Darstellung zu ermöglichen.
Hierzu kann ein Startskript im Event-Control hinzugefügt werden.

```
$gatewayIDs = IPS_GetInstanceListByModuleID("{1C902193-
```

```
B044-43B8-9433-419F09C641B8}");
$instanceIDs = IPS_GetInstanceList();
foreach($gatewayIDs as $gatewayID) {
foreach($instanceIDs as $instanceID) {
$i = IPS_GetInstance($instanceID);
if($i['ConnectionID'] == $gatewayID) {
switch($i['ModuleInfo']['ModuleID']) {
case "{24A9D68D-7B98-4D74-9BAE-3645D435A9EF}": //Shutter
case "81F54858-72B1-4C2C-8CE3-7E00A3168378": //RGB (Legacy until IP-
Symcon 5.0)
case "{81F54858-72B1-4C2C-8CE3-7E00A3168378}": //RGB (IP-Symcon
5.1+)
case "4D7F7548-0979-4ABD-9BB3-81F9477C0903": //RGBW (Legacy until
IP-Symcon 5.0)
case "{4D7F7548-0979-4ABD-9BB3-81F9477C0903}": //RGBW (IP-Symcon
5.1+)
EIB_RequestStatus($instanceID);
break;

case "{D62B95D3-0C5E-406E-B1D9-8D102E50F64B}": //Group
if(IPS_GetProperty($instanceID, "GroupCapabilityRead")) {
EIB_RequestStatus($instanceID);
}
break;
case "{FB223058-3084-C5D0-C7A2-3B8D2E73FE8A}": //Device
KNX_RequestStatus($instanceID);
break;
default:
//DPTs
if(strpos($i['ModuleInfo']['ModuleName'], "DPT") !== false) {
if(IPS_GetProperty($instanceID, "CapabilityRead")) {
KNX_RequestStatus($instanceID);
}
}
break;
}
}
}
}
```

### SKRIPT ZUR AUSGABE DER GRUPPENADRESSEN

Das folgende Skript kann dazu genutzt werden die Gruppenadressen samt Id, Name und Location auszugeben.

```
echo "ID,Name,Location,GA,MoreGA" . PHP_EOL;

$gatewayIDs = IPS_GetInstanceListByModuleID("{1C902193-
B044-43B8-9433-419F09C641B8}");
$instanceIDs = IPS_GetInstanceList();
foreach($gatewayIDs as $gatewayID) {
foreach($instanceIDs as $instanceID) {
$i = IPS_GetInstance($instanceID);
if($i['ConnectionID'] == $gatewayID) {
switch($i['ModuleInfo']['ModuleID']) {
```

```
case "{24A9D68D-7B98-4D74-9BAE-3645D435A9EF}": //Shutter
printGA($instanceID, "GroupMove");
if (IPS_GetProperty($instanceID, "EnableStep")) {
printGA($instanceID, "GroupStep");
}
printGA($instanceID, "GroupStop");
if (IPS_GetProperty($instanceID, "EnablePosition")) {
printGA($instanceID, "GroupPosition");
}
if (IPS_GetProperty($instanceID, "EnableBladePosition")) {
printGA($instanceID, "GroupBladePosition");
}
break;
case "81F54858-72B1-4C2C-8CE3-7E00A3168378": //RGB (Legacy until IP-
Symcon 5.0)
case "{81F54858-72B1-4C2C-8CE3-7E00A3168378}": //RGB (IP-Symcon
5.1+)
case "4D7F7548-0979-4ABD-9BB3-81F9477C0903": //RGBW (Legacy until
IP-Symcon 5.0)
case "{4D7F7548-0979-4ABD-9BB3-81F9477C0903}": //RGBW (IP-Symcon
5.1+)
case "{D62B95D3-0C5E-406E-B1D9-8D102E50F64B}": //Group
printGA($instanceID, "Group");
break;
case "{FB223058-3084-C5D0-C7A2-3B8D2E73FE8A}": //Device
printGAs($instanceID);
break;
default:
//DPTs
if(strpos($i['ModuleInfo']['ModuleName'], "DPT") !== false) {
printGA($instanceID, "");
}
break;
}
}
}
}

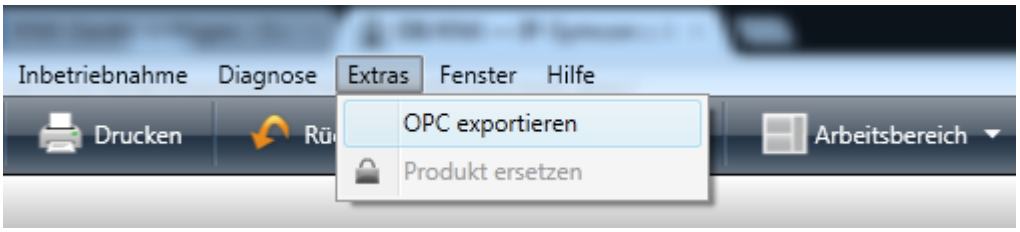
function printGA($id, $prefix) {
$c = json_decode(IPS_GetConfiguration($id), true);
echo sprintf("%d,%s,%s,", $id, IPS_GetName($id),
IPS_GetLocation($id));
echo sprintf("%d/%d/%d", $c[$prefix . "Address1"], $c[$prefix .
"Address2"], $c[$prefix . "Address3"]);
foreach (json_decode($c[$prefix . "Mapping"], true) as $a) {
echo "," . sprintf("%d/%d/%d", $a[$prefix . "Address1"], $a[$prefix
. "Address2"], $a[$prefix . "Address3"]);
}
echo PHP_EOL;
}

function printGAs($id) {
$c = json_decode(IPS_GetConfiguration($id), true);
foreach(json_decode($c['GroupAddresses'], true) as $ga) {
echo sprintf("%d,%s,%s,", $id, IPS_GetName($id),
IPS_GetLocation($id));
echo sprintf("%d/%d/%d", $ga["Address1"], $ga["Address2"],
$ga["Address3"]);
foreach ($ga["Mapping"] as $a) {
```

```
echo "," . sprintf("%d/%d/%d", $a["Address1"], $a["Address2"],
$a["Address3"]);
}
echo PHP_EOL;
}
}
```

## LEGACY

OPC Export in ETS




Videoanleitung zum Export:

Einrichten mit OPC

### ZUORDNUNG DER EINZELNEN FUNKTIONEN ZUM EIS-TYP/DPT

| FUNKTION | EIS-TYP | DPT |
|---|---|---|
| EIB_Char | EIS13 | DPT4 |
| EIB_Counter8bit | EIS14 | DPT5, DPT6 |
| EIB_Counter16bit | EIS10 | DPT7, DPT8 |
| EIB_Counter32bit | EIS11 | DPT12, DPT13 |
| EIB_Date | EIS4 | DPT11 |
| EIB_DimControl | EIS2 | DPT3 |
| EIB_DimValue | EIS6 | DPT5 |
| EIB_DriveBladeValue | EIS6 | DPT5 |
| EIB_DriveMove | EIS7 | DPT1 |
| EIB_DriveShutterValue | EIS6 | DPT5 |
| EIB_DriveStep | EIS7 | DPT1 |
| EIB_FloatValue | EIS9 | DPT14 |
| EIB_Move | EIS7 | DPT1 |
| EIB_Position | EIS6 | DPT5 |
| EIB_PriorityControl | EIS8 | DPT2 |
| EIB_PriorityPosition | EIS1 | DPT1 |
| EIB_Scale | EIS6 | DPT5 |
| EIB_Str | EIS15 | DPT16 |
| EIB_Switch | EIS1 | DPT1 |
| EIB_Time | EIS3 | DPT10 |
| EIB_Value | EIS5 | DPT9 |
| N/A | EIS12 | DPT15 |

# EIB_Char

```
boolean EIB_Char (int $InstanzID, string $Buchstabe)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Buchstabe | Einzelner Buchstabe |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet einen einzelnen Buchstaben **Buchstabe** an das Gerät mit der ID **InstanzID**.

## BEISPIEL

```
EIB_Char(12345, "A"); //Sendet den Buchstaben A auf den Bus
```

# EIB_Counter8Bit

```
boolean EIB_Counter8Bit (int $InstanzID, float $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | Standard = 0..255; Signed = -127..127 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet den Zählerwert **Wert** an das Gerät mit der ID **InstanzID**.

## BEISPIEL

```
EIB_Counter8Bit(12345, 10); //Sendet den Wert 10 auf den Bus
```

# EIB_Counter16Bit

```
boolean EIB_Counter16Bit (int $InstanzID, float $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | Standard = 0..65535; Signed = -32768..32767 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet den Zählerwert **Wert** an das Gerät mit der ID **InstanzID**.

## BEISPIEL

```
EIB_Counter16Bit(12345, 10); //Sendet den Wert 10 auf den Bus
```

# EIB_Counter32Bit

```
boolean EIB_Counter32Bit (int $InstanzID, float $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | Standard = -2147483648..2147483647 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet den Zählerwert **Wert** an das Gerät mit der ID **InstanzID**.

## BEISPIEL

```
EIB_Counter32Bit(12345, 10); //Sendet den Wert 10 auf den Bus
```

# EIB_Date

```
boolean EIB_Date (int $InstanzID, string $Datum)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Datum | Standard = yyyymmdd |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet den Datumswert **Datum** an das Gerät mit der ID **InstanzID**.

## BEISPIEL

```
EIB_Date(12345, date("Ymd")); //Sendet das aktuelle Datum auf den
Bus
```

# EIB_DimControl

```
boolean EIB_DimControl (int $InstanzID, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | Dimmwertverstellung zwischen 0..15 oder -7..7 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet eine Dimmwertverstellung um **Wert** an das Gerät mit der ID **InstanzID**. Je nach Konfiguration kann 0..15 (Standard) oder -7..7 (Enhanced) gesendet werden.



Da diese Funktion nur eine relative Veränderungen des Dimmers vornimmt, kann die Helligkeit nicht absolut angegeben werden. Es wird empfohlen eine 8-Bit Adresse anzulegen, um direkte Dimmwerte setzen zu können.

## BEISPIEL

```
EIB_DimControl(12345, 0);
```

# EIB_DimValue

```
boolean EIB_DimValue (int $InstanzID, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | Wert 0..255, 0..100% |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet den Dimmwert **Wert** an das Gerät mit der ID **InstanzID**. Je nach Konfiguration kann 0..255 (Standard) oder 0..100 (Percent) als Wert Parameter übergeben werden.

## BEISPIEL

```
EIB_DimValue(12345, 100); //Auf 100% dimmen
```

# EIB_DriveBladeValue

```
boolean EIB_DriveBladeValue (int $InstanID, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanID | ID des zu schaltenden Geräts |
| Wert | Wert 0..255, 0..100% |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet den Positionswert **Wert** an das Gerät mit der ID **InstanzID**. Je nach Konfiguration kann 0..255 (Standard) oder 0..100 (Percent) als Wert Parameter übergeben werden.

## BEISPIEL

```
EIB_DriveBladeValue(12345, 100); //Auf 100% fahren
```

# EIB_DriveMove

```
boolean EIB_DriveMove (int $InstanzID, boolean $Richtung)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Richtung | Standart |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Bewegt das Gerät mit ID **InstanzID** in eine bestimmt **Richtung**. Je nach Konfiguration kann die Bewegungsrichtung invertiert werden.

## BEISPIEL

```

```

# EIB_DriveShutterValue

```
boolean EIB_DriveShutterValue (int $InstanzID, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | Wert 0..255, 0..100% |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet den Positionswert **Wert** an das Gerät mit der ID **InstanzID**. Je nach Konfiguration kann 0..255 (Standard) oder 0..100 (Percent) als Wert Parameter übergeben werden.

## BEISPIEL

```
EIB_DriveShutterValue(12345, 100); //Auf 100% fahren
```

# EIB_DriveStep

```
boolean EIB_DriveStep (int $InstanzID, boolean $SchrittRichtung)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| SchrittRichtung | Standard |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Bewegt das Gerät *schrittweise* mit ID **InstanzID** in eine bestimmt **Richtung** in Bewegung. Je nach Konfiguration kann die Bewegungsrichtung invertiert werden.

## BEISPIEL

```
EIB_DriveStep(12345, true); //Rollladen schrittweise bewegen
```

# EIB_FloatValue

```
boolean EIB_FloatValue (int $InstanzID, float $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | Fließkommawert |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet den Wert **Wert** an das Gerät mit der ID **InstanzID**.

## BEISPIEL

```
EIB_FloatValue(12345, 22.1); //Sendet 22.1°C an den Regler
```

# EIB_Move

```
boolean EIB_Move (int $InstanzID, int $Richtung)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der KNX Shutter Instanz |
| Richtung | 0 = Hoch, 1 = Schritt (Hoch), 2 = Stop, 3 = Schritt (Runter), 4 = Runter |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Bewegt den Rollladen mit der InstanzID in die gewünschte Richtung.

## BEISPIEL

```
EIB_Move(12345, 3); //Rollladen anhalten
```

# EIB_Position

```
boolean EIB_Position (int $InstanzID, int $Position)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der KNX Shutter Instanz |
| Position | 0-100% |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Bewegt den Rollladen mit der **InstanzID** auf die gewünschte **Position**.



Dieser Befehl funktioniert nur, wenn Ihr KNX-Gerät eine absolute Positionierung unterstützt

## BEISPIEL

```
EIB_Position(12345, 50); //Rollladen auf 50% fahren
```

# EIB_PriorityControl

```
boolean EIB_PriorityControl (int $InstanzID, int $Zustand)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Zustand | Standard |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt das Gerät mit ID **InstanzID** auf einen bestimmten **Zustand**.

## BEISPIEL

```
EIB_PriorityControl(12345, 0);
```

# EIB_PriorityPosition

```
boolean EIB_PriorityPosition (int $InstanzID, boolean $Richtung)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Richtung | Standard |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt das Gerät mit ID **InstanzID** auf eine bestimmte **Richtung**. Je nach Konfiguration kann die Richtung invertiert werden.

## BEISPIEL

```
EIB_PriorityPosition(12345, true); //Bewegt Gerät
```

# EIB_RequestStatus

```
boolean EIB_RequestStatus (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schickt eine Lesenanfrage für Gerät mit der ID **InstanzID** auf den Bus.



Die Leseanfrage wird nur für die Hauptadresse versendet.



Der Befehl EIB_RequestStatus funktioniert für EIB-Instanzen, also Instanzen mit dem Herseteller "KNX/EIB". Für DPT-Instanzen, also Instanzen mit dem Hersteller "KNX", kann analog die Funktion KNX_RequestStatus verwendet werden.

## BEISPIEL

```
EIB_RequestStatus(12345); //Leseanfrage abschicken
```

# EIB_Scale

```
boolean EIB_Scale (int $InstanzID, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | Wert 0..255, 0..100, 0..360 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet den Wert **Wert** an das Gerät mit der ID **InstanzID**. Je nach Konfiguration kann 0..255 (Standard), 0..100 (Percent) oder 0..360 (Degree) als Wert Parameter übergeben werden.

## BEISPIEL

```
EIB_Scale(12345, 360);
```

# EIB_SetRGB

```
boolean EIB_SetRGB (int $InstanzID, int $Rot, int $Grün, int $Blau)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Rot | 0 .. 255 (0 = aus, 255 = hell) |
| Grün | 0 .. 255 (0 = aus, 255 = hell) |
| Blau | 0 .. 255 (0 = aus, 255 = hell) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den RGB Stripe des Geräts mit der ID **InstanzID** auf den Rotwert **Rot**, den Grünwert **Grün** und den Blauwert **Blau**.

## BEISPIEL

```
EIB_SetRGB(12345, 255, 0, 0); //Setzt den RGB Stripe auf Rot
```

# EIB_SetRGBW

```
boolean EIB_SetRGBW (int $InstanzID, int $Rot, int $Grün, int $Blau,
int $Weiss)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Rot | 0 .. 255 (0 = aus, 255 = hell) |
| Grün | 0 .. 255 (0 = aus, 255 = hell) |
| Blau | 0 .. 255 (0 = aus, 255 = hell) |
| Weiss | 0 .. 255 (0 = aus, 255 = hell) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den RGBW Stripe des Geräts mit der ID **InstanzID** auf den Rotwert **Rot**, den Grünwert **Grün**, den Blauwert **Blau** und den Weißanteil auf **Weiss**.

## BEISPIEL

```
EIB_SetRGBW(12345, 255, 0, 0, 0); //Setzt den RGB Stripe auf Rot
```

# EIB_Str

```
boolean EIB_Str (int $InstanzID, string $Zeichenkette)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Zeichenkette | Zu sendender Text (maximal 14 Zeichen) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet einen Text **Zeichenkette** an das Gerät mit der ID **InstanzID**.

## BEISPIEL

```
EIB_Str(12345, "Es klingelt!"); //Sendet die Zeichenkette auf den
Bus
```

# EIB_Switch

```
boolean EIB_Switch (int $InstanzID, boolean $Status)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Status | TRUE für An, FALSE für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Schaltet das Gerät mit der ID InstanzID auf Wert Status

## BEISPIEL

```
EIB_Switch(12345, true); //Schaltet das Gerät ein
```

# EIB_Time

```
boolean EIB_Time (int $InstanzID, string $Zeit)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Zeit | Standard = dhhmmss; TimeOnly = hhmmss |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet den Zeitwert **Zeit** an das Gerät mit der ID **InstanzID**. Der d Wert ist dabei zwischen 1 (Montag) und 7 (Sonntag).

## BEISPIEL

```
EIB_Time(12345, date("NHis")); //Sendet die aktuelle Zeit auf den
Bus (Standard)
EIB_Time(12345, date("His")); //Sendet die aktuelle Zeit auf den Bus
(TimeOnly)
```

# EIB_Value

```
boolean EIB_Value (int $InstanzID, float $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | -671088,63 .. 670760,96 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet den Wert **Wert** an das Gerät mit der ID **InstanzID**.

## BEISPIEL

```
EIB_Value(12345, 22.1); //Sendet 22.1°C an den Regler
```

# Geräteliste

## UNTERSTÜTZTE GATEWAYS

Es werden alle seriellen Gateways mit FT1.2 Protokoll (und cEMI Nachrichten) und alle KNX/IP Gateways, die das Tunneling Protokoll anbieten, unterstützt. Bei KNX Gateways, welche diese Voraussetzungen erfüllen und nicht funktionieren, kontaktieren Sie bitte unseren Support. Wir lösen das Problem schnellstmöglich.
Seit IP-Symcon 6.4 wird ebenfalls KNX Data Secure und KNX IP Secure unterstützt.

| KOMPONENTE | HERSTELLER | ANSCHLUSSART. | BESCHREIBUNG |
|---|---|---|---|
| SymBox mit KNX Erweiterung | Symcon GmbH | Seriell (Secure) | integriertes Modul von Weinzierl Engineering GmbH |
| 1a EIB KNX IP Schnittstelle PoE | EIBMARKT | IP | EIB KNX IP Schnittstelle PoE (ArtNr. N000401) |
| i-bus® KNX / IP-Schnittstelle | ABB | IP | REG IPS/S 2.1, 2CDG 110 098 R0011 |
| MDT SCN-IP000.02 | MDT | IP | |
| MDT SCN-IP000.03 | MDT | IP (Secure) | |
| 1168 KNX IP Secure Interface | Enertex | IP (Secure) | |
| KNX BAOS Module 838 kBerry | Weinzierl Engineering GmbH | IP (Secure) | |
| KNX BAOS Module 838 kBerry | Weinzierl Engineering GmbH | Seriell | |
| KNX IP Interface 730 | Weinzierl Engineering GmbH | IP | |
| KNX IP Interface 731 | Weinzierl Engineering GmbH | IP | |
| KNX IP Interface 732 Secure | Weinzierl Engineering GmbH | IP (Secure) | |
| KNX IP Interface 740 wireless | Weinzierl Engineering GmbH | IP | |
| KNX Router Weinzierl 750 | Weinzierl Engineering GmbH | IP | |
| N146 | Siemens | IP | |
| N148 | Siemens | IP | |

## UNTERSTÜTZTE DPT

ab Version 5.0

| DPT-TYP | VERSION |
| --- | --- |
| DPT 001.x | ab 5.0 |
| DPT 002.x | ab 5.0 |
| DPT 003.x | ab 5.0 |
| DPT 004.x | ab 5.0 |
| DPT 005.x | ab 5.0 |
| DPT 006.x | ab 5.0 |
| DPT 007.x | ab 5.0 |
| DPT 008.x | ab 5.0 |
| DPT 009.x | ab 5.0 |
| DPT 010.x | ab 5.0 |
| DPT 011.x | ab 5.0 |
| DPT 012.x | ab 5.0 |
| DPT 013.x | ab 5.0 |
| DPT 014.x | ab 5.0 |
| DPT 015.x | ab 5.0 |
| DPT 016.x | ab 5.0 |
| DPT 017.x | ab 5.0 |
| DPT 018.x | ab 5.0 |
| DPT 019.x | ab 5.0 |
| DPT 020.x | ab 5.0 |
| DPT 021.x | ab 5.0 |
| DPT 022.x | ab 5.0 |
| DPT 023.x | ab 5.0 |
| DPT 024.x | ab 5.0 |
| DPT 025.x | ab 5.0 |
| DPT 026.x | ab 5.0 |
| DPT 027.x | ab 5.0 |
| DPT 028.x | ab 5.0 |
| DPT 029.x | ab 5.0 |
| DPT 030.x | ab 5.0 |
| DPT 031.x | ab 5.0 |
| DPT 200.x | ab 5.0 |
| DPT 201.x | ab 5.0 |
| DPT 202.x | ab 5.0 |
| DPT 203.x | ab 5.0 |
| DPT 204.x | ab 5.0 |
| DPT 205.x | ab 5.0 |
| DPT 206.x | ab 5.0 |

## DPT 001.X

| DPT | DPT_NAME |
|---|---|
| 1.001 | DPT_Switch |
| 1.002 | DPT_Bool |
| 1.003 | DPT_Enable |
| 1.004 | DPT_Ramp |
| 1.005 | DPT_Alarm |
| 1.006 | DPT_BinaryValue |
| 1.007 | DPT_Step |
| 1.008 | DPT_UpDown |
| 1.009 | DPT_OpenClose |
| 1.010 | DPT_Start |
| 1.011 | DPT_State |
| 1.012 | DPT_Invert |
| 1.013 | DPT_DimSendStyle |
| 1.014 | DPT_InputSource |
| 1.015 | DPT_Reset |
| 1.016 | DPT_Ack |
| 1.017 | DPT_Trigger |
| 1.018 | DPT_Occupancy |
| 1.019 | DPT_Window_Door |
| 1.021 | DPT_LogicalFunction |
| 1.022 | DPT_Scene_AB |
| 1.023 | DPT_ShutterBlinds_Mode |
| 1.024 | DPT_DayNight |
| 1.100 | DPT_Heat/Cool |

## DPT 002.X

| DPT | DPT_NAME |
|---|---|
| 2.001 | DPT_Switch_Control |
| 2.002 | DPT_Bool_Control |
| 2.003 | DPT_Enable_Control |
| 2.004 | DPT_Ramp_Control |
| 2.005 | DPT_Alarm_Control |
| 2.006 | DPT_BinaryValue_Control |
| 2.007 | DPT_Step_Control |
| 2.008 | DPT_Direction1_Control |
| 2.009 | DPT_Direction2_Control |
| 2.010 | DPT_Start_Control |
| 2.011 | DPT_State_Control |
| 2.012 | DPT_Invert_Control |

### DPT 003.X

| DPT | DPT_NAME |
|---|---|
| 3.007 | DPT_Control_Dimming |
| 3.008 | DPT_Control_Blinds |

### DPT 004.X

| DPT | DPT_NAME |
|---|---|
| 4.001 | DPT_Char_ASCII |
| 4.002 | DPT_Char_8859_1 |

### DPT 005.X

| DPT | DPT_NAME |
|---|---|
| 5.001 | DPT_Scaling |
| 5.003 | DPT_Angle |
| 5.004 | DPT_Percent_U8 |
| 5.005 | DPT_DecimalFactor |
| 5.006 | DPT_Tariff |
| 5.010 | DPT_Value_1_Ucount |

### DPT 006.X

| DPT | DPT_NAME |
|---|---|
| 6.001 | DPT_Percent_V8 |
| 6.010 | DPT_Value_1_Count |
| 6.020 | DPT_Status_Mode3 |

### DPT 007.X

| DPT | DPT_NAME |
|---|---|
| 7.001 | DPT_Value_2_Ucount |
| 7.002 | DPT_TimePeriodMsec |
| 7.003 | DPT_TimePeriod10MSec |
| 7.004 | DPT_TimePeriod100MSec |
| 7.005 | DPT_TimePeriodSec |
| 7.006 | DPT_TimePeriodMin |
| 7.007 | DPT_TimePeriodHrs |
| 7.010 | DPT_PropDataType |
| 7.011 | DPT_Length_mm |
| 7.012 | DPT_UElCurrentmA |
| 7.013 | DPT_Brightness |

### DPT 008.X

| DPT | DPT_NAME |
|---|---|
| 8.001 | DPT_Value_2_Count |
| 8.002 | DPT_DeltaTimeMsec |
| 8.003 | DPT_DeltaTime10MSec |
| 8.004 | DPT_DeltaTime100MSec |
| 8.005 | DPT_DeltaTimeSec |
| 8.006 | DPT_DeltaTimeMin |
| 8.007 | DPT_DeltaTimeHrs |
| 8.010 | DPT_Percent_V16 |
| 8.011 | DPT_Rotation_Angle |
| 8.012 | DPT_Length_m |

### DPT 009.X

| DPT | DPT_NAME |
|---|---|
| 9.001 | DPT_Value_Temp |
| 9.002 | DPT_Value_Tempd |
| 9.003 | DPT_Value_Tempa |
| 9.004 | DPT_Value_Lux |
| 9.005 | DPT_Value_Wsp |
| 9.006 | DPT_Value_Pres |
| 9.007 | DPT_Value_Humidity |
| 9.008 | DPT_Value_AirQuality |
| 9.010 | DPT_Value_Time1 |
| 9.011 | DPT_Value_Time2 |
| 9.020 | DPT_Value_Volt |
| 9.021 | DPT_Value_Curr |
| 9.022 | DPT_PowerDensity |
| 9.023 | DPT_KelvinPerPercent |
| 9.024 | DPT_Power |
| 9.025 | DPT_Value_Volume_Flow |
| 9.026 | DPT_Rain_Amount |
| 9.027 | DPT_Value_Temp_F |
| 9.028 | DPT_Value_Wsp_kmh |

### DPT 010.X

| DPT | DPT_NAME |
|---|---|
| 10.001 | DPT_TimeOfDay |

### DPT 011.X

| DPT | DPT_NAME |
|---|---|
| 11.001 | DPT_Date |

### DPT 012.X

| DPT | DPT_NAME |
|---|---|
| 12.001 | DPT_Value_4_Ucount |
| 12.1200 | DPT_VolumeLiquid_Litre |
| 12.1201 | DPT_Volume_m3 |

## DPT 013.X

| DPT | DPT_NAME |
|---|---|
| 13.001 | DPT_Value_4_Count |
| 13.002 | DPT_FlowRate_m3/h |
| 13.010 | DPT_ActiveEnergy |
| 13.011 | DPT_ApparantEnergy |
| 13.012 | DPT_ReactiveEnergy |
| 13.013 | DPT_ActiveEnergy_kWh |
| 13.014 | DPT_ApparantEnergy_kVAh |
| 13.015 | DPT_ReactiveEnergy_kVARh |
| 13.100 | DPT_LongDeltaTimeSec |
| 13.1200 | DPT_DeltaVolumeLiquid_Litre |
| 13.1201 | DPT_DeltaVolume_m3 |
| 13.16 | DPT_ActiveEnergy_MWh |

## DPT 014.X

| DPT | DPT_NAME |
|---|---|
| 14.000 | DPT_Value_Acceleration |
| 14.001 | DPT_Value_Acceleration_Angular |
| 14.002 | DPT_Value_Activation_Energy |
| 14.003 | DPT_Value_Activity |
| 14.004 | DPT_Value_Mol |
| 14.005 | DPT_Value_Amplitude |
| 14.006 | DPT_Value_AngleRad |
| 14.007 | DPT_Value_AngleDeg |
| 14.008 | DPT_Value_Angular_Momentum |
| 14.009 | DPT_Value_Angular_Velocity |
| 14.010 | DPT_Value_Area |
| 14.011 | DPT_Value_Capacitance |
| 14.012 | DPT_Value_Charge_DensitySurface |
| 14.013 | DPT_Value_Charge_DensityVolume |
| 14.014 | DPT_Value_Compressibility |
| 14.015 | DPT_Value_Conductance |
| 14.016 | DPT_Value_Electrical_Conductivity |
| 14.017 | DPT_Value_Density |
| 14.018 | DPT_Value_Electric_Charge |
| 14.019 | DPT_Value_Electric_Current |
| 14.020 | DPT_Value_Electric_CurrentDensity |
| 14.021 | DPT_Value_Electric_DipoleMoment |
| 14.022 | DPT_Value_Electric_Displacement |
| 14.023 | DPT_Value_Electric_FieldStrength |
| 14.024 | DPT_Value_Electric_Flux |
| 14.025 | DPT_Value_Electric_FluxDensity |
| 14.026 | DPT_Value_Electric_Polarization |
| 14.027 | DPT_Value_Electric_Potential |
| 14.028 | DPT_Value_Electric_PotentialDifference |
| 14.029 | DPT_Value_ElectromagneticMoment |
| 14.030 | DPT_Value_Electromotive_Force |
| 14.031 | DPT_Value_Energy |
| 14.032 | DPT_Value_Force |
| 14.033 | DPT_Value_Frequency |
| 14.034 | DPT_Value_Angular_Frequency |
| 14.035 | DPT_Value_Heat_Capacity |
| 14.036 | DPT_Value_Heat_FlowRate |
| 14.037 | DPT_Value_Heat_Quantity |

### DPT 015.X

| DPT | DPT_NAME |
|---|---|
| 15.000 | DPT_Access_Data |

### DPT 016.X

| DPT | DPT_NAME |
|---|---|
| 16.000 | DPT_String_ASCII |
| 16.001 | DPT_String_8859_1 |

### DPT 017.X

| DPT | DPT_NAME |
|---|---|
| 17.001 | DPT_SceneNumber |

### DPT 018.X

| DPT | DPT_NAME |
|---|---|
| 18.001 | DPT_SceneControl |

### DPT 019.X

| DPT | DPT_NAME |
|---|---|
| 19.001 | DPT_DateTime |

### DPT 020.X

| DPT | DPT_NAME |
|---|---|
| 20.001 | DPT_SCLOMode |
| 20.002 | DPT_BuildingMode |
| 20.003 | DPT_OccMode |
| 20.004 | DPT_Priority |
| 20.005 | DPT_LightApplicationMode |
| 20.006 | DPT_ApplicationArea |
| 20.007 | DPT_AlarmClassType |
| 20.008 | DPT_PSUMode |
| 20.011 | DPT_ErrorClass_System |
| 20.012 | DPT_ErrorClass_HVAC |
| 20.013 | DPT_Time_Delay |
| 20.014 | DPT_Beaufort_Wind_Force_Scale |
| 20.017 | DPT_SensorSelect |
| 20.020 | DPT_ActuatorConnectType |
| 20.100 | DPT_FuelType |
| 20.101 | DPT_BurnerType |
| 20.102 | DPT_HVACMode |
| 20.103 | DPT_DHWMode |
| 20.104 | DPT_LoadPriority |
| 20.105 | DPT_HVACContrMode |
| 20.106 | DPT_HVACEmergMode |
| 20.107 | DPT_ChangeoverMode |
| 20.108 | DPT_ValveMode |
| 20.109 | DPT_DamperMode |
| 20.110 | DPT_HeaterMode |
| 20.111 | DPT_FanMode |
| 20.112 | DPT_MasterSlaveMode |
| 20.113 | DPT_StatusRoomSetp |
| 20.120 | DPT_ADAType |
| 20.121 | DPT_BackupMode |
| 20.122 | DPT_StartSynchronization |
| 20.600 | DPT_Behaviour_Lock_Unlock |
| 20.601 | DPT_Behaviour_Bus_Power_Up_Down |
| 20.602 | DPT_DALI_Fade_Time |
| 20.603 | DPT_BlinkingMode |
| 20.604 | DPT_LightControlMode |
| 20.605 | DPT_SwitchPBModel |
| 20.606 | DPT_PBAction |

## DPT 021.X

| DPT | DPT_NAME |
|---|---|
| 21.001 | DPT_StatusGen |
| 21.002 | DPT_Device_Control |
| 21.100 | DPT_ForceSign |
| 21.101 | DPT_ForceSignCool |
| 21.102 | DPT_StatusRHC |
| 21.103 | DPT_StatusSDHWC |
| 21.104 | DPT_FuelTypeSet |
| 21.105 | DPT_StatusRCC |
| 21.106 | DPT_StatusAHU |
| 21.601 | DPT_LightActuatorErrorInfo |
| 21.1000 | DPT_RF_ModeInfo |
| 21.1001 | DPT_RF_FilterInfo |
| 21.1010 | DPT_Channel_Activation_8 |

## DPT 022.X

| DPT | DPT_NAME |
|---|---|
| 22.100 | DPT_StatusDHWC |
| 22.101 | DPT_StatusRHCC |
| 22.1000 | DPT_Media |
| 22.1010 | DPT_Channel_Activation_16 |

## DPT 023.X

| DPT | DPT_NAME |
|---|---|
| 23.001 | DPT_OnOff_Action |
| 23.002 | DPT_Alarm_Reaction |
| 23.003 | DPT_UpDown_Action |
| 23.102 | DPT_HVAC_PB_Action |

### DPT 024.X

| DPT | DPT_NAME |
|---|---|
| 24.001 | DPT_VarString_8859_1 |

### DPT 025.X

| DPT | DPT_NAME |
|---|---|
| 25.1000 | DPT_DoubleNibble |

### DPT 026.X

| DPT | DPT_NAME |
|---|---|
| 26.001 | DPT_SceneInfo |

### DPT 027.X

| DPT | DPT_NAME |
|---|---|
| 27.001 | DPT_CombinedInfoOnOff |

### DPT 028.X

| DPT | DPT_NAME |
|---|---|
| 28.001 | DPT_UTF-8 |

### DPT 029.X

| DPT | DPT_NAME |
|---|---|
| 29.010 | DPT_ActiveEnergy_V64 |
| 29.011 | DPT_ApparantEnergy_V64 |
| 29.012 | DPT_ReactiveEnergy_V64 |

### DPT 030.X

| DPT | DPT_NAME |
|---|---|
| 30.1010 | DPT_Channel_Activation_24 |

### DPT 031.X

| DPT | DPT_NAME |
|---|---|
| 31.101 | DPT_PB_Action_HVAC_Extended |

### DPT 200.X

| DPT | DPT_NAME |
|---|---|
| 200.100 | DPT_Heat/Cool_Z |
| 200.101 | DPT_BinaryValue_Z |

### DPT 201.X

| DPT | DPT_NAME |
|---|---|
| 201.100 | DPT_HVACMode_Z |
| 201.102 | DPT_DHWMode_Z |
| 201.104 | DPT_HVACContrMode_Z |
| 201.105 | DPT_EnablH/Cstage_Z DPT_EnablH/CStage |
| 201.107 | DPT_BuildingMode_Z |
| 201.108 | DPT_OccMode_Z |
| 201.109 | DPT_HVACEmergMode_Z |

### DPT 202.X

| DPT | DPT_NAME |
|---|---|
| 202.001 | DPT_RelValue_Z |
| 202.002 | DPT_UCountValue8_Z |

### DPT 203.X

| DPT | DPT_NAME |
|---|---|
| 203.002 | DPT_TimePeriodMsec_Z |
| 203.003 | DPT_TimePeriod10Msec_Z |
| 203.004 | DPT_TimePeriod100Msec_Z |
| 203.005 | DPT_TimePeriodSec_Z |
| 203.006 | DPT_TimePeriodMin_Z |
| 203.007 | DPT_TimePeriodHrs_Z |
| 203.011 | DPT_UFlowRateLiter/h_Z |
| 203.012 | DPT_UCountValue16_Z |
| 203.013 | DPT_UElCurrentµA_Z |
| 203.014 | DPT_PowerKW_Z |
| 203.015 | DPT_AtmPressureAbs_Z |
| 203.017 | DPT_PercentU16_Z |
| 203.100 | DPT_HVACAirQual_Z |
| 203.101 | DPT_WindSpeed_Z DPT_WindSpeed |
| 203.102 | DPT_SunIntensity_Z |
| 203.104 | DPT_HVACAirFlowAbs_Z |

### DPT 204.X

| DPT | DPT_NAME |
|---|---|
| 204.001 | DPT_RelSignedValue_Z |

### DPT 205.X

| DPT | DPT_NAME |
|---|---|
| 205.002 | DPT_DeltaTimeMsec_Z |
| 205.003 | DPT_DeltaTime10Msec_Z |
| 205.004 | DPT_DeltaTime100Msec_Z |
| 205.005 | DPT_DeltaTimeSec_Z |
| 205.006 | DPT_DeltaTimeMin_Z |
| 205.007 | DPT_DeltaTimeHrs_Z |
| 205.017 | DPT_Percent_V16_Z |
| 205.100 | DPT_TempHVACAbs_Z |
| 205.101 | DPT_TempHVACRel_Z |
| 205.102 | DPT_HVACAirFlowRel_Z |

### DPT 206.X

| DPT | DPT_NAME |
|---|---|
| 206.100 | DPT_HVACModeNext |
| 206.102 | DPT_DHWModeNext |
| 206.104 | DPT_OccModeNext |
| 206.105 | DPT_BuildingModeNext |

### DPT 207.X

| DPT | DPT_NAME |
|---|---|
| 207.100 | DPT_StatusBUC |
| 207.101 | DPT_LockSign |
| 207.102 | DPT_ValueDemBOC |
| 207.104 | DPT_ActPosDemAbs |
| 207.105 | DPT_StatusAct |
| 207.600 | DPT_StatusLightingActuator |

### DPT 209.X

| DPT | DPT_NAME |
|---|---|
| 209.100 | DPT_StatusHPM |
| 209.101 | DPT_TempRoomDemAbs |
| 209.102 | DPT_StatusCPM |
| 209.103 | DPT_StatusWTC |

### DPT 210.X

| DPT | DPT_NAME |
|---|---|
| 210.100 | DPT_TempFlowWaterDemAbs |

### DPT 211.X

| DPT | DPT_NAME |
|---|---|
| 211.100 | DPT_EnergyDemWater |

### DPT 212.X

| DPT | DPT_NAME |
|---|---|
| 212.100 | DPT_TempRoomSetpSetShift[3] |
| 212.101 | DPT_TempRoomSetpSet[3] |

### DPT 213.X

| DPT | DPT_NAME |
|---|---|
| 213.100 | DPT_TempRoomSetpSet[4] |
| 213.101 | DPT_TempDHWSetpSet[4] |
| 213.102 | DPT_TempRoomSetpSetShift[4] |

### DPT 214.X

| DPT | DPT_NAME |
|---|---|
| 214.100 | DPT_PowerFlowWaterDemHPM |
| 214.101 | DPT_PowerFlowWaterDemCPM |

### DPT 215.X

| DPT | DPT_NAME |
|---|---|
| 215.100 | DPT_StatusBOC |
| 215.101 | DPT_StatusCC |

### DPT 216.X

| DPT | DPT_NAME |
|---|---|
| 216.100 | DPT_SpecHeatProd |

### DPT 217.X

| DPT | DPT_NAME |
|---|---|
| 217.001 | DPT_Version |

### DPT 218.X

| DPT | DPT_NAME |
|---|---|
| 218.001 | DPT_VolumeLiter_Z |
| 218.002 | DPT_FlowRate_m3/h_Z |

### DPT 219.X

| DPT | DPT_NAME |
|---|---|
| 219.001 | DPT_AlarmInfo |

### DPT 220.X

| DPT | DPT_NAME |
|---|---|
| 220.100 | DPT_TempHVACAbsNext |

### DPT 221.X

| DPT | DPT_NAME |
|---|---|
| 221.001 | DPT_SerNum |

### DPT 222.X

| DPT | DPT_NAME |
|---|---|
| 222.100 | DPT_TempRoomSetpSetF16[3] |
| 222.101 | DPT_TempRoomSetpSetShiftF16[3] |

### DPT 223.X

| DPT | DPT_NAME |
|---|---|
| 223.100 | DPT_EnergyDemAir |

### DPT 224.X

| DPT | DPT_NAME |
|---|---|
| 224.100 | DPT_TempSupply AirSetpSet |

### DPT 225.X

| DPT | DPT_NAME |
|---|---|
| 225.001 | DPT_ScalingSpeed |
| 225.002 | DPT_Scaling_Step_Time |
| 225.003 | DPT_TariffNext |

### DPT 229.X

| DPT | DPT_NAME |
|---|---|
| 229.001 | DPT_MeteringValue |

### DPT 230.X

| DPT | DPT_NAME |
|---|---|
| 230.1000 | DPT_MBus_Address |

### DPT 231.X

| DPT | DPT_NAME |
|---|---|
| 231.001 | DPT_Locale_ASCII |

### DPT 232.X

| DPT | DPT_NAME |
|---|---|
| 232.600 | DPT_Colour_RGB |

### DPT 234.X

| DPT | DPT_NAME |
|---|---|
| 234.001 | DPT_LanguageCodeAlpha2_ASCII |
| 234.002 | DPT_RegionCodeAlpha2_ASCII |

### DPT 235.X

| DPT | DPT_NAME |
|---|---|
| 235.001 | DPT_Tariff_ActiveEnergy |

### DPT 236.X

| DPT | DPT_NAME |
| --- | --- |
| 236.001 | DPT_Prioritised_Mode_Control |

### DPT 237.X

| DPT | DPT_NAME |
| --- | --- |
| 237.600 | DPT_DALI_Control_Gear_Diagnostic |

### DPT 238.X

| DPT | DPT_NAME |
| --- | --- |
| 238.001 | DPT_SceneConfig |
| 238.600 | DPT_DALI_Diagnostics |

### DPT 239.X

| DPT | DPT_NAME |
| --- | --- |
| 239.001 | DPT_FlaggedScaling |

### DPT 240.X

| DPT | DPT_NAME |
| --- | --- |
| 240.800 | DPT_CombinedPosition |

### DPT 241.X

| DPT | DPT_NAME |
| --- | --- |
| 241.800 | DPT_StatusSAB |

### DPT 242.X

| DPT | DPT_NAME |
| --- | --- |
| 242.600 | DPT_Colour_xyY |

### DPT 249.X

| DPT | DPT_NAME |
| --- | --- |
| 249.600 | DPT_Brightness_Colour_Temperature_Transition |

### DPT 251.X

| DPT | DPT_NAME |
| --- | --- |
| 251.600 | DPT_Colour_RGBW |

## UNTERSTÜTZTE EIS

| EIS-TYP | VERSION | PASSENDE DPT |
| --- | --- | --- |
| EIS1 | ab 3.4 | DPT 001.x |
| EIS2 | ab 3.4 | DPT 003.x |
| EIS3 | ab 3.4 | DPT 010.x |
| EIS4 | ab 3.4 | DPT 011.x |
| EIS5 | ab 3.4 | DPT 009.x |
| EIS6 | ab 3.4 | DPT 005.x |
| EIS7 | ab 3.4 | DPT 001.x |
| EIS8 | ab 3.4 | DPT 002.x |
| EIS9 | ab 3.4 | DPT 014.x |
| EIS10 | ab 3.4 | DPT 007.x , DPT 008.x |
| EIS11 | ab 3.4 | DPT 012.x , DPT 013.x |
| EIS12 | ab 3.4 | DPT 015.x |
| EIS13 | ab 3.4 | DPT 004.x |
| EIS14 | ab 3.4 | DPT 005.x , DPT 006.x |
| EIS15 | ab 3.4 | DPT 016.x |

# KNX_RequestStatus

```
boolean KNX_RequestStatus (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schickt eine Lesenanfrage für Gerät mit der ID **InstanzID** auf den Bus.



Die Leseanfrage wird nur für die Hauptadresse versendet.



Der Befehl KNX_RequestStatus funktioniert für DPT-Instanzen, also Instanzen mit dem Herseteller "KNX". Für EIB-Instanzen, also Instanzen mit dem Hersteller "KNX/EIB", kann analog die Funktion EIB_RequestStatus verwendet werden.

## BEISPIEL

```
KNX_RequestStatus(12345); //Leseanfrage abschicken
```

# LCN

LCN (Local Control Network) ist ein kabelgebundenes Bussystem.



Folgende Geräte werden von IP-Symcon unterstützt:
Unterstützte Geräte



Die neuen Module mit 12 Variablen (ab Firmware 160B13) werden ab IP-Symcon 3.0 unterstützt.

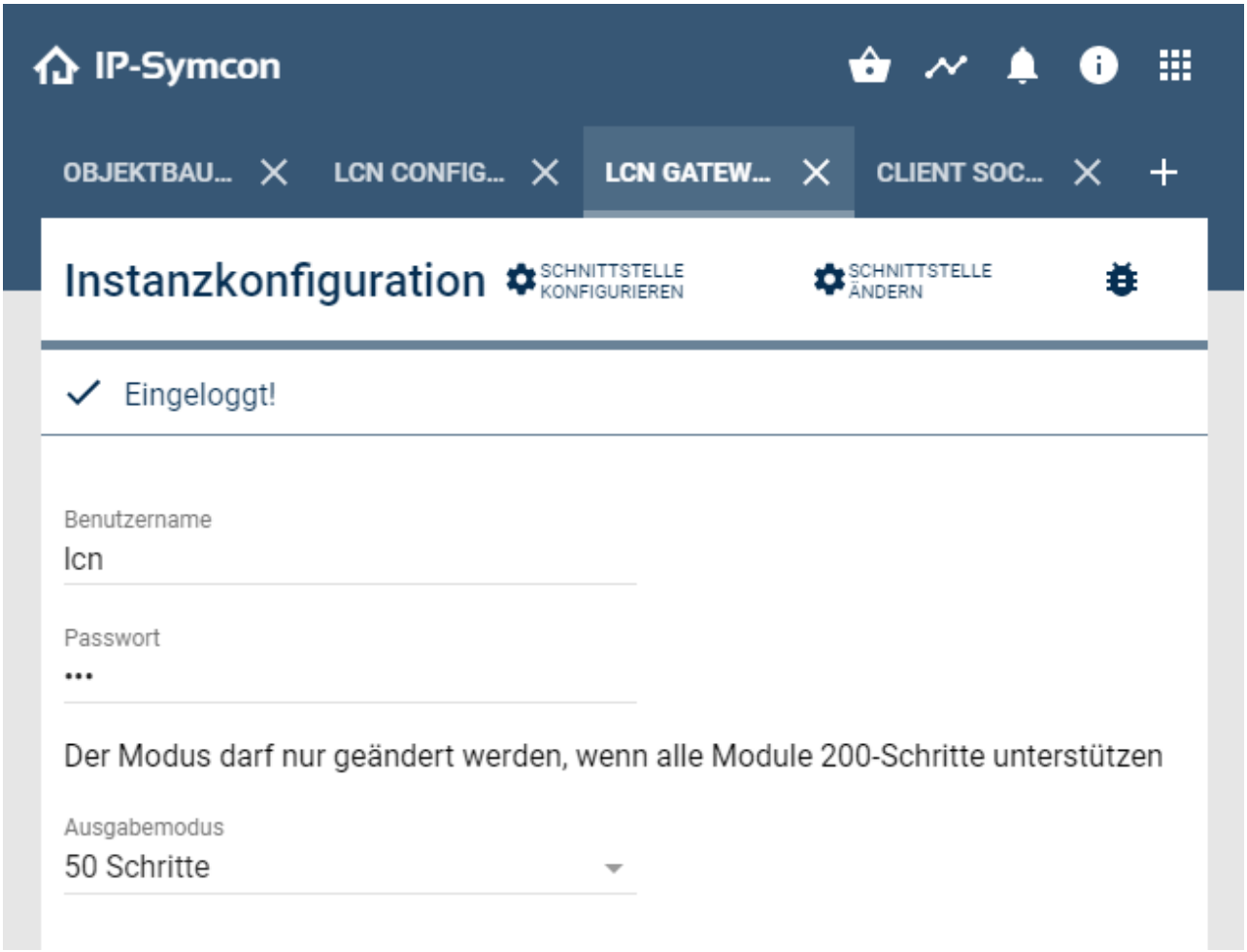
## INSTALLATION

Um einen LCN-Bus an IP-Symcon anschließen zu können, wird ein Bus-Koppler (LCN-PKU ) mit Koppelprogramm PCHK oder ein Ethernet-Koppler (LCN-VISU ) benötigt.

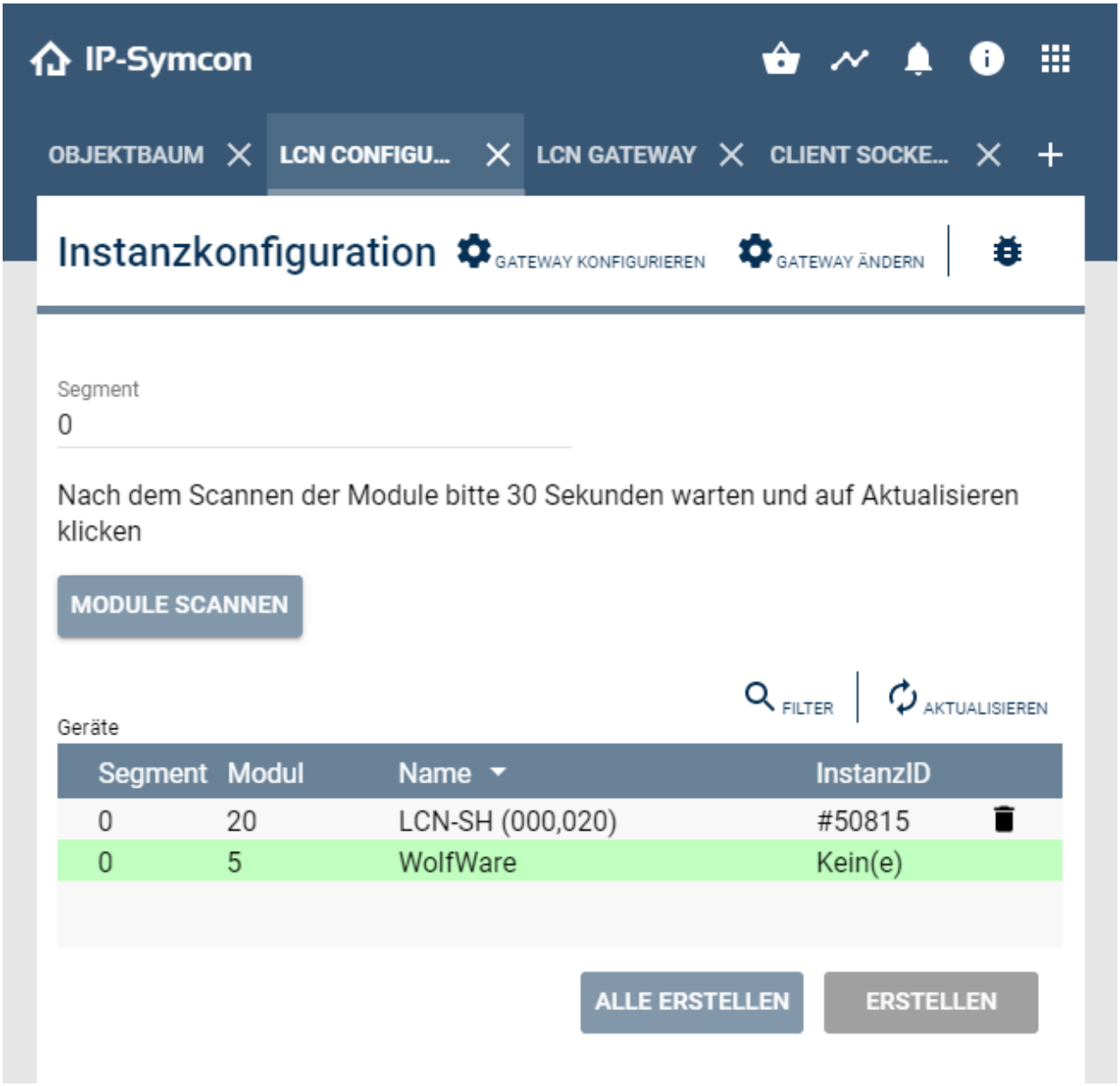
## VIDEO-TUTORIAL ZUR EINRICHTUNG

## EINBINDUNG IN IP-SYMCON

Über die Verwaltungskonsole kann ein Konfigurator für das LCN System erstellt werden. Nachdem ein Konfigurator erstellt wurde, kann unter Umständen eine rote Meldung erscheinen. Diese weist auf eine fehlerhafte oder unvollständige Konfiguration hin. Um den Fehler zu beheben muss in der übergeordneten Instanz der Benutzername und das Passwort angegeben werden. Mit einem Klick auf "Übernehmen" wird die Änderung gespeichert.

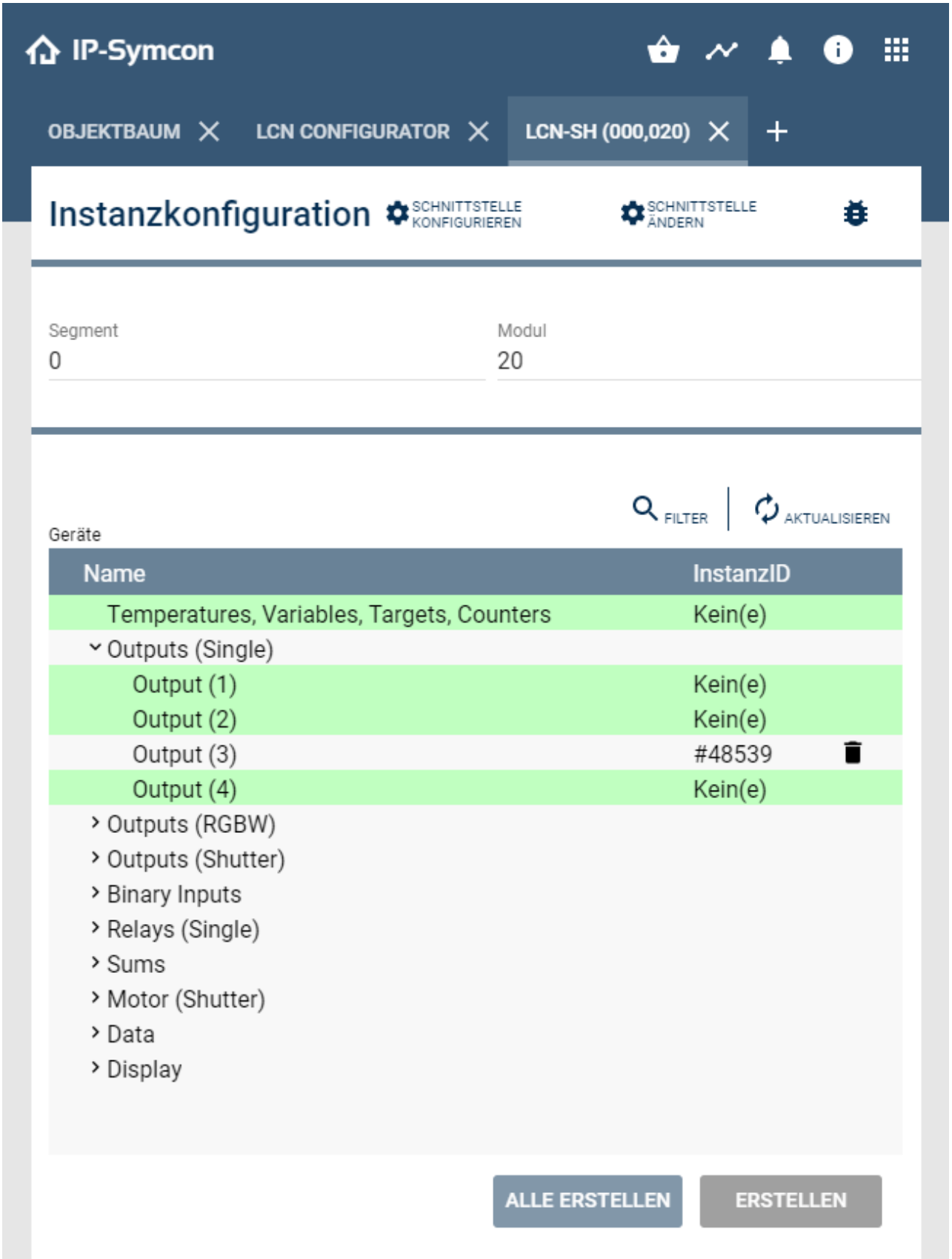



Falls nach dem Übernehmen kein Einloggen möglich ist, muss die Netzwerkkonfiguration innerhalb Schnittstelle überprüft werden. Auch kann im Gateway der Ausgabemodus zwischen 50 und 200 Schritten ausgewählt werden.
Nach dem erfolgreichen Einloggen, kann im Konfigurator auf dem Bus nach Modulen gesucht, diese in IP-Symcon erstellt und die einzelnen Module als IP-Symcon Splitter-Instanzen anlegt werden.

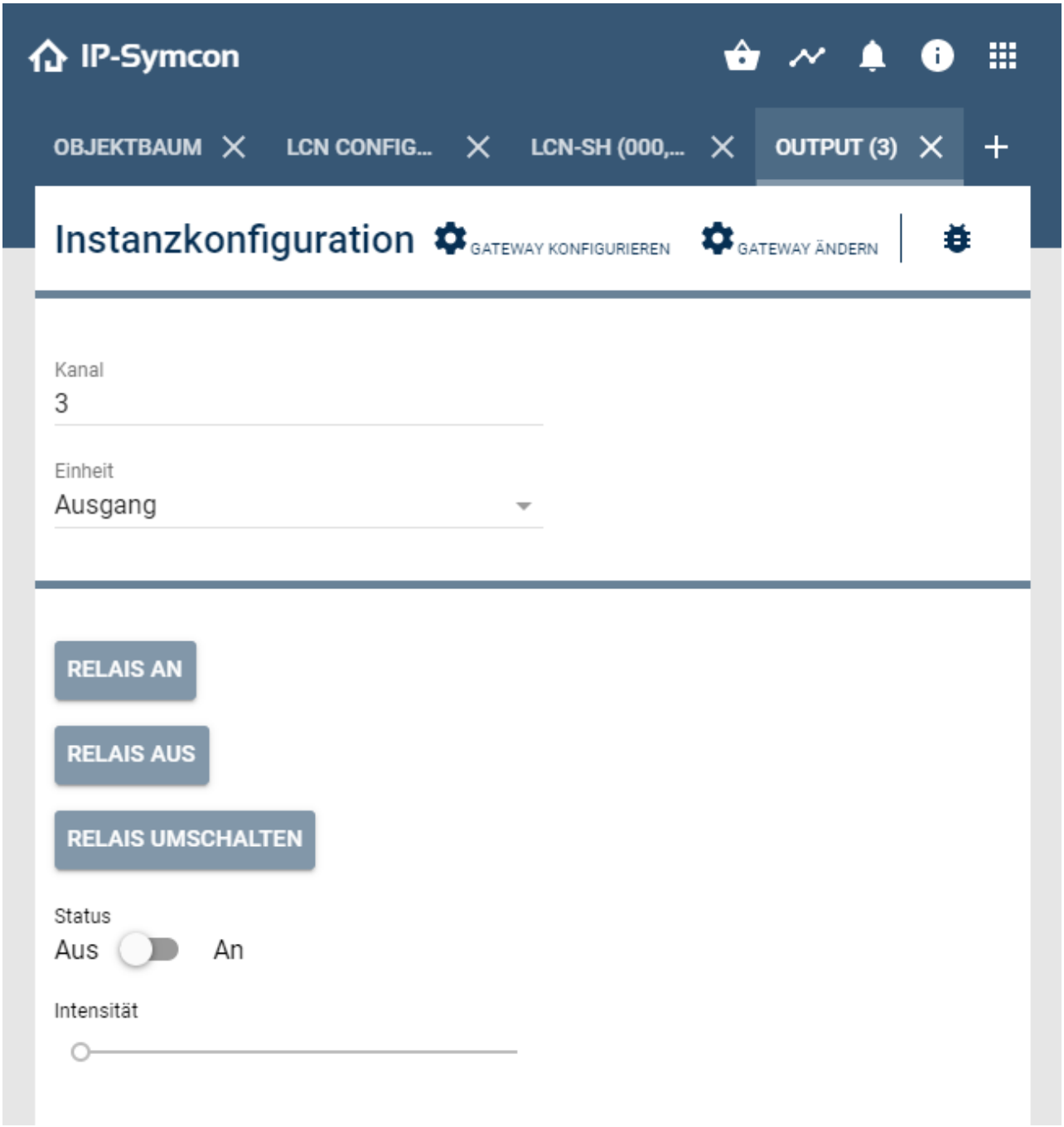



## MODULE KONFIGURIEREN

Sobald im Konfigurator ein Modul erstellt und auf die Konfigurationstaste gedrückt wurde, öffnet sich ein neuer Dialog.




Um die einzelnen Instanzen zu erstellen, müssen die gewünschten Relais, Ein-/Ausgänge, etc ausgewählt und auf "Erstellen" geklickt werden.
Neue Geräte werden im Objektbaum in der Kategorie erstellt und einsortiert, welches namentlich zum Modul passt. Diese erstellten Instanzen können dann entsprechend umbenannt und an anderer Stelle einsortiert werden. Auch besteht die Möglichkeit via "Konfigurieren" die jeweilige Instanzkonfiguration aufzurufen.


IP-Symcon
OBJEKTBAUM
LCN CONFIG...
LCN-SH (000,...
OUTPUT (3)
Instanzkonfiguration
GATEWAY KONFIGURIEREN
GATEWAY ÄNDERN
Kanal
3
Einheit
Ausgang
RELAIS AN
RELAIS AUS
RELAIS UMSCHALTEN
Status
Aus
An
Intensität

# Geräteliste

## UNTERSTÜTZTE GATEWAYS

| GATEWAY | BESCHREIBUNG |
|---|---|
| LCN-PK (über LCN PCHK) | Serieller Anschluss, Kommunikation über PCHK mit IP-Symcon |
| LCN-PKU (über LCN PCHK) | USB Anschluss, Kommunikation über PCHK mit IP-Symcon |
| LCN-PKE | Ethernet Anschluss, Kommunikation direkt mit IP-Symcon |
| LCN-VISU | Ethernet Anschluss, Kommunikation direkt mit IP-Symcon |

## UNTERSTÜTZTE KOMPONENTEN

IP-Symcon unterstützt alle Komponenten.



Beim LCN-HL4 wird ausschließlich das RGB Farbmodell (Schalterstellung 2) unterstützt

# LCN_AddGroup

```
boolean LCN_AddGroup (int $InstanzID, int $Gruppe)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz; die ID der Splitter-Instanz, nicht der Geräte-Instanz |
| Gruppe | Gruppennummer (1..255) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Fügt das Gerät mit der ID **InstanzID** in die Gruppe mit der Gruppennummer **Gruppe** hinzu.

## BEISPIEL

```
//Das Gerät mit der ID 12345 wird der Gruppe 123 hinzugefügt
LCN_AddGroup(12345, 123);
```

# LCN_AddIntensity

```
boolean LCN_AddIntensity (int $InstanzID, int $Intensität)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Intensität | 0-100% |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Dimmt das Gerät mit der ID **InstanzID** um den Wert **Intensität** heller.

## BEISPIEL

```
LCN_AddIntensity(12345, 50); //Gerät um 50 Prozentpunkte heller
dimmen
```

# LCN_AddThresholdCurrent

```
boolean LCN_AddThresholdCurrent (int $InstanzID, int $Register, int
$Schwellwert, float $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Register | Register des zu schaltenden Schwellwerts (1x5 |
| Schwellwert | zu setzender Schwellwert (1x5 |
| Wert | Wert des Schwellwerts (0..1000) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Addiert den Wert **Wert** des Schwellwerts **Schwellwert** im Register **Register** relativ zu dem aktuellen Wert des LCN-Moduls mit der ID **InstanzID**.

## BEISPIEL

```
// 1x5 Schwellwert
// Addiert den Wert 2,5 zu dem Schwellwert 2 der Instanz mit der ID
12345
LCN_AddThresholdCurrent(12345, 0, 2, 2.5);

// 4x4 Schwellwert
// Addiert den Wert 2,5 zu dem Schwellwert 2 im Register 3 der
Instanz mit der ID 12345
LCN_AddThresholdCurrent(12345, 3, 2, 2.5);
```

# LCN_AddThresholdDefined

```
boolean LCN_AddThresholdDefined (int $InstanzID, int $Register, int
$Schwellwert, float $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Register | Register des zu schaltenden Schwellwerts (1x5 |
| Schwellwert | zu setzender Schwellwert (1x5 |
| Wert | Wert des Schwellwerts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Addiert den Wert **Wert** des Schwellwerts **Schwellwert** im Register **Register** relativ zu dem programmierten Wert des LCN-Moduls mit der ID **InstanzID**.

## BEISPIEL

```
// 1x5 Schwellwert
// Addiert den Wert 2,5 relativ zum programmierten Wert in den
Schwellwert 2 der Instanz mit der ID 12345
LCN_AddThresholdDefined(12345, 0, 2, 2.5);

// 4x4 Schwellwert
// Addiert den Wert 2,5 relativ zum programmierten Wert in den
Schwellwert 2 im Register 3 der Instanz mit der ID 12345
LCN_AddThresholdDefined(12345, 3, 2, 2.5);
```

# LCN_Beep

```
boolean LCN_Beep (int $InstanzID, boolean $SonderTon, int $Anzahl)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz; die ID der Splitter-Instanz, nicht der Geräte-Instanz |
| SonderTon | TRUE für An, FALSE für Aus |
| Anzahl | Anzahl zwischen 1..15 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Lässt die Instanz mit der ID InstanzID piepen

## BEISPIEL

```
LCN_Beep(12345, true, 1); //Ein Mal piepen (Sonderton)
```

# LCN_DeductIntensity

```
boolean LCN_DeductIntensity (int $InstanzID, int $Intensität)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Intensität | 0-100% |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Dimmt das Gerät mit der ID InstanzID um den Wert Intensität dunkler.

## BEISPIEL

```
LCN_DeductIntensity(12345, 10); //Gerät um 10 Prozentpunkte dunkler
dimmen
```

# LCN_DeductThresholdCurrent

```
boolean LCN_DeductThresholdCurrent (int $InstanzID, int $Register,
int $Schwellwert, float $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Register | Register des zu schaltenden Schwellwerts (1x5 |
| Schwellwert | zu setzender Schwellwert (1x5 |
| Wert | Wert des Schwellwerts (0..1000) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Subtrahiert den Wert **Wert** des Schwellwerts **Schwellwert** im Register **Register** relativ zu dem aktuellen Wert des LCN-Moduls mit der ID **InstanzID**.

## BEISPIEL

```
// 1x5 Schwellwert
// Subtrahiert den Wert 2,5 zu dem Schwellwert 2 der Instanz mit der
ID 12345
LCN_DeductThresholdCurrent(12345, 0, 2, 2.5);

// 4x4 Schwellwert
// Subtrahiert den Wert 2,5 zu dem Schwellwert 2 im Register 3 der
Instanz mit der ID 12345
LCN_DeductThresholdCurrent(12345, 3, 2, 2.5);
```

# LCN_DeductThresholdDefined

```
boolean LCN_DeductThresholdDefined (int $InstanzID, int $Register,
int $Schwellwert, float $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Register | Register des zu schaltenden Schwellwerts (1x5 |
| Schwellwert | zu setzender Schwellwert (1x5 |
| Wert | Wert des Schwellwerts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Subtrahiert den Wert **Wert** des Schwellwerts **Schwellwert** im Register **Register** relativ zu dem programmierten Wert des LCN-Moduls mit der ID **InstanzID**.

## BEISPIEL

```
// 1x5 Schwellwert
// Subtrahiert den Wert 2,5 relativ zum programmierten Wert in den
Schwellwert 2 der Instanz mit der ID 12345
LCN_AddThresholdDefined(12345, 0, 2, 2.5);

// 4x4 Schwellwert
// Subtrahiert den Wert 2,5 relativ zum programmierten Wert in den
Schwellwert 2 im Register 3 der Instanz mit der ID 12345
LCN_AddThresholdDefined(12345, 3, 2, 2.5);
```

# LCN_FadeOut

```
boolean LCN_FadeOut (int $InstanzID, int $Intensität, int $Rampe)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Intensität | 0-100% |
| Rampe | 0-200 Sekunde |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Dimmt das Gerät mit der ID **InstanzID** auf Wert **Intensität** und dimmt dann mit der **Rampe** Sekunden auf 0% herunter.

## BEISPIEL

```
LCN_FadeOut(12345, 100, 5); //Gerät von 100% in 5 sekunden
runterdimmen auf 0%
```

# LCN_FlipRelay

```
boolean LCN_FlipRelay (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Relais |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Schaltet das Relais mit der ID **InstanzID** um

## BEISPIEL

```
LCN_FlipRelay(12345); //Relais umschalten
```

# LCN_LimitOutput

```
boolean LCN_LimitOutput (int $InstanzID, int $Wert, int $Zeit,
string $ZeitTyp)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Ausgangs |
| Wert | Prozentualer Wert auf den begrenzt werden soll (0..100) |
| Zeit | Zeitraum wielange der Ausgang begrenzt werden soll |
| ZeitTyp | Gibt an ob die Begrenzung für Sekunden, Minuten, Stunden oder Tage gelten soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Begrenzt den Ausgang mit der ID **InstanzID** für den Zeitraum **Zeit** und Zeittyp **ZeitTyp** auf den Wert **Wert**.

| ZEITTYP | MÖGLICHER ZEITRAUM |
|---|---|
| 'S' (Sekunden) | 1..60 |
| 'M' (Minuten) | 1..90 |
| 'H' (Stunden) | 1..50 |
| 'D' (Tage) | 1..45 |

## BEISPIEL

```
//Begrenzt den Ausgang mit der ID 12345 für 8 Stunden auf 36%
LCN_LimitOutput(12345, 36, 8, 'H');
```

# LCN_LoadScene

```
boolean LCN_LoadScene (int $InstanzID, int $Szene)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Ausgangs |
| Szene | Welche Szene geladen werden soll (0..9, 15) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Ruft die Szene **Szene** des Ausgangs mit der ID **InstanzID** auf.



Wird als Szene der Sonderwert 15 angegeben, wird der Wert aus dem Zähler Tvar/Var1 genommen

## BEISPIEL

```
// Ruft die Szene 5 des Ausgangs mit der ID 12345 auf
LCN_LoadScene(12345, 5);
```

# LCN_LockTargetValue

```
boolean LCN_LockTargetValue (int $InstanzID, int $Ziel)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Ziel | Welcher Regler gesperrt werden soll (0,1,2,3) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sperrt den Regler **Ziel** des Geräts mit der ID **InstanzID**.

## BEISPIEL

```
//Sperrt den Regler 0 des Geräts 12345
LCN_LockTargetValue(12345, 0);
```

# LCN_RampStop

```
boolean LCN_RampStop (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Stoppt den Dimmvorgang von Gerät mit der ID InstanzID

## BEISPIEL

```
LCN_RampStop(12345);
```

# LCN_ReleaseTargetValue

```
boolean LCN_ReleaseTargetValue (int $InstanzID, int $Ziel)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Ziel | Welcher Regler entsperrt werden soll (0,1,2,3) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Entsperrt den Regler **Ziel** des Geräts mit der ID **InstanzID**.

## BEISPIEL

```
// Entsperrt den Regler 0 des Geräts 12345
LCN_ReleaseTargetValue(12345, 0);
```

# LCN_RemoveGroup

```
boolean LCN_RemoveGroup (int $InstanzID, int $Gruppe)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz; die ID der Splitter-Instanz, nicht der Geräte-Instanz |
| Gruppe | Gruppennummer (0..255) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Entfernt das Gerät mit der ID **InstanzID** aus der Gruppe mit der Gruppennummer **Gruppe**.



Wenn die Gruppe 0 angegeben wird, wird das Gerät aus allen Gruppen entfernt

## BEISPIEL

```
//Das Gerät mit der ID 12345 wird aus der Gruppe 123 entfernt
LCN_RemoveGroup(12345, 123);
```

# LCN_RequestLights

```
boolean LCN_RequestLights (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des abzufragenden Tableaus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Fragt die LEDs (Tableaulämpchen) des Tableau mit der ID InstanzID ab.

## BEISPIEL

```
//Die LEDs des Tableau mit der ID 12345 werden abgefragt
LCN_RequestLights(12345);
```

# LCN_RequestRead

```
boolean LCN_RequestRead (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des abzufragenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Fragt die Variablen, Targets und Zähler der Temperatur-Instanz mit der ID InstanzID ab.

## BEISPIEL

```
//Das Gerät mit der ID 12345 abfragen
LCN_RequestRead(12345);
```

# LCN_RequestStatus

```
boolean LCN_RequestStatus (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des abzufragenden Moduls |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Fragt des Status aller Ausgänge, Relais und binären Sensoren des Moduls mit der ID **InstanzID** ab.

## BEISPIEL

```
// Abfragen aller Status des Moduls mit der ID 12345 ab
LCN_RequestStatus(12345);
```

# LCN_RequestThresholds

```
boolean LCN_RequestThresholds (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des abzufragenden Thresholds |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Fragt die Schwellenwerte und Hysterese ID InstanzID ab.

## BEISPIEL

```
//Die Schwellenwerte und Hysterese des Thresholds 12345 abfragen
LCN_RequestThresholds(12345);
```

# LCN_SaveScene

```
boolean LCN_SaveScene (int $InstanzID, int $Szene)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Ausgangs |
| Szene | In welcher Szene gespeichert werden soll (0..9, 15) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Speichert die Szene **Szene** für den Ausgang mit der ID **InstanzID**.



Wird als Szene der Sonderwert 15 angegeben, wird der Wert aus dem Zähler Tvar/Var1 genommen

## BEISPIEL

```
// Speichert die Szene 5 des Ausgangs mit der ID 12345
LCN_SaveScene(12345, 5);
```

# LCN_SelectSceneRegister

```
boolean LCN_SelectSceneRegister (int $InstanzID, int $Register)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Register | welches Register ausgewählt werden soll (0..9) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Wählt das Register mit der Nummer **Register** des Geräts mit der ID **InstanzID** aus.

## BEISPIEL

```
// Wählt das Register 2 des Geräts mit der ID 12345
LCN_SelectSceneRegister(12345, 2);
```

# LCN_SendCommand

```
boolean LCN_SendCommand (int $InstanzID, string $Funktion, string
$Daten)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz; die ID der Splitter-Instanz, nicht der Geräte-Instanz |
| Funktion | Die ersten beiden (2) Zeichen des PCHK Befehls (z.B. A1) |
| Daten | Die restlichen Zeichen des PCHK Befehls (z.B. DI000000). Eventuelle Zahlen müssen in der dezimalen Darstellung (3 Zeichen) gesendet werden. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Sendet einen benutzerdefinierten Befehl an die Instanz der ID InstanzID. Mögliche Befehle entnehmen Sie bitte der PCHK Dokumentation. Falls Sie diese nicht besitzen, können Sie diese u.U. von Ihrem Fachhändler beziehen. Wir haben leider keine Möglichkeit diese bereitzustellen.

## BEISPIEL

```
LCN_SendCommand(12345, "A1", "DI000000"); //Ausgang 1 sofort auf 0%
dimmen.
```

# LCN_SetDisplayText

```
boolean LCN_SetDisplayText (int $InstanzID, int $Zeile, string
$Text)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Zeile | Displayzeile zwischen 1..4 |
| Text | Text mit maximal 60 Zeichen |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Zeigt den Text **Text** auf Zeile **Zeile** des Displays mit der ID **InstanzID** an.

## BEISPIEL

```
//Zeigt auf Zeile 1 des Displays, mit der InstanzID 12345, "Hallo
Welt" an.
LCN_SetDisplayText(12345, 1, "Hallo Welt");
```

# LCN_SetDisplayTime

```
boolean LCN_SetDisplayTime (int $InstanzID, int $Zeile, int
$Anzeigedauer)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Zeile | Displayzeile zwischen 1..4 |
| Anzeigedauer | Zeiteinheiten zwischen 1..30 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Konfiguriert die Zeile **Zeile** des Displays mit der ID **InstanzID** auf die Anzeigedauer **Anzeigedauer**.



Ob die Anzeigedauer als Sekunden/Minuten/Stunden/Tage definiert ist, wird im LCN-Modul eingestellt, sprich: Im Gerät selbst.

## BEISPIEL

```
//Konfiguriert die Zeile 1 der Displayinstanz 12345 auf 30
Zeiteinheiten
LCN_SetDisplayTime(12345, 1, 30);
```

# LCN_SetIntensity

```
boolean LCN_SetIntensity (int $InstanzID, int $Intensität, int $Rampe)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Intensität | 0-100% |
| Rampe | 0 = Sofort; n = Sekunden |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Dimmt das Gerät mit der ID **InstanzID** auf Wert **Intensität** in **Rampe** Sekunden

## BEISPIEL

```
LCN_SetIntensity(12345, 100, 0); //Gerät auf 100% sofort dimmen
```

# LCN_SetLamp

```
boolean LCN_SetLamp (int $InstanzID, int $Lampe, string $Aktion)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz; die ID der Splitter-Instanz, nicht der Geräte-Instanz |
| Lampe | Auswahl der zu schaltenden Tableau-Lämpchen (1..12) |
| Aktion | Welche Aktion ausgeführt werden soll. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schaltet das Tableau-Lämpchen **Lampe** der Instanz mit der ID **InstanzID** auf den Wert **Aktion**.

| AKTION | BESCHREIBUNG |
|---|---|
| 'A' | Ausschalten |
| 'B' | Blinken |
| 'E' | Einschalten |
| 'F' | Flackern |

## BEISPIEL

```
// Das Tableau-Lämpchen 2 des Moduls mit der ID 12345 blinken
lassen.
LCN_SetLamp(12345, 2, 'B');
```

# LCN_SetRelay

```
boolean LCN_SetRelay (int $InstanzID, string $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz; die ID der Splitter-Instanz, nicht der Geräte-Instanz |
| Wert | 8fache Zeichenkombination der folgenden Werte |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt alle Relais der Instanz mit der ID **InstanzID** auf **Wert**

## BEISPIEL

```
LCN_SetRelay(12345, "--1100UU");
```

# LCN_SetRGBW

```
boolean LCN_SetRGBW (int $InstanzID, int $Rot, int $Grün, int $Blau,
int $Weiß)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Rot | Intensität zwischen 1..100 |
| Grün | Intensität zwischen 1..100 |
| Blau | Intensität zwischen 1..100 |
| Weiß | Intensität zwischen 1..100 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den Farbwert und Weißwert der Instanz mit der ID **InstanzID** auf **Rot**, **Grün** und **Blau**, sowie die Intensität auf **Weiß**.

## BEISPIEL

```
//Setz die Instanz 12345 auf Rot mit voller Intensität.
LCN_SetRGBW(12345, 100, 0, 0, 100);
```

# LCN_SetTargetValue

```
boolean LCN_SetTargetValue (int $InstanzID, int $Ziel, float $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| Ziel | 0 = Regler R1; 1 = Regler R2; 2 = Regler S1; 3 = Regler S2 |
| Wert | Wert |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den Regler **Ziel** der Instanz mit der ID **InstanzID** auf den Wert **Wert**

## BEISPIEL

```
LCN_SetTargetValue(12345, 2, 24.3); //Regler S1 auf 24.3°C setzen
```

# LCN_ShiftTargetValue

```
boolean LCN_ShiftTargetValue (int $InstanzID, int $Ziel, float
$RelativerWert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Ziel | Welcher Regler geshiftet werden soll (0,1,2,3) |
| RelativerWert | Wert um den geshiftet werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Shiftet den aktuellen Sollwert des Reglers **Ziel** des Geräts mit der ID **InstanzID** um den Wert **RelativerWert**.

## BEISPIEL

```
// Shiftet den Sollwert des Reglers 1 um -1,7°C
LCN_ShiftTargetValue(12345, 1, -1.7);
```

# LCN_ShutterMove

```
boolean LCN_ShutterMove (int $InstanzID, int $Position)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Position | Prozentuale Position des Rollladen (0-100%) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Startet die Bewegung des Rollladen mit der ID **InstanzID** auf die Position **Position**.

## BEISPIEL

```
//Rollladenbewegung auf 80% (zu 20% geschlossen)
LCN_ShutterMove(12345, 80);
```

# LCN_ShutterMoveDown

```
boolean LCN_ShutterMoveDown (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Startet die Bewegung des Rollladen mit der ID **InstanzID** nach unten.

## BEISPIEL

```
//Rollladenbewegung nach unten starten
LCN_ShutterMoveDown(12345);
```

# LCN_ShutterMoveUp

```
boolean LCN_ShutterMoveUp (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Startet die Bewegung des Rollladen mit der ID **InstanzID** nach oben.

## BEISPIEL

```
//Rollladenbewegung nach oben starten
LCN_ShutterMoveUp(12345);
```

# LCN_ShutterStop

```
boolean LCN_ShutterStop (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Stoppt die Bewegung des Rollladen mit der ID **InstanzID**.

## BEISPIEL

```
//Rollladen stoppen
LCN_ShutterStop(12345);
```

# LCN_StartFlicker

```
boolean LCN_StartFlicker (int $InstanzID, string $FlackerTiefe,
string $Geschwindigkeit, int $AnzahlFlackerhübe)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| FlackerTiefe | G = Gering; M = Mittel; S = Start |
| Geschwindigkeit | L = Langsam (ca. 0,5x / Sek); M = Mittel (ca. 1x / Sek); S = Schnell (ca. 2x / Sek) |
| AnzahlFlackerhübe | Anzahl zwischen 1..15 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Das Gerät mit der ID **InstanzID** auf Flackern lassen

## BEISPIEL

```
LCN_StartFlicker(12345, "G", "L", 5);
```

# LCN_StopFlicker

```
boolean LCN_StopFlicker (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Flackern des Geräts mit der InstanzID sofort beenden

## BEISPIEL

```
LCN_StopFlicker(12345);
```

# LCN_SwitchDurationMin

```
boolean LCN_SwitchDurationMin (int $InstanzID, int $Minuten, string
$Abdimmzeit, boolean $Erhaltend)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Minuten | Nach wievielen Minuten Abgedimmt werden soll |
| Abdimmzeit | Wie schnell abgedimmt werden soll |
| Erhaltend | Das Gerät wird nach Ablauf der Zeit nur ausgeschaltet, wenn vorher noch nicht an war |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Dimmt das Gerät mit der ID **InstanzID** nach Minuten **Minuten** heller. Weitere Optionen sind die Dimmgeschwindigkeit **Abdimmzeit** und ob es **Erhaltend** ist.

| ABDIMMZEIT | BESCHREIBUNG |
|---|---|
| 'K' | Kurze Abdimmzeit |
| 'M' | Mittlere Abdimmzeit |
| 'L' | Lange Abdimmzeit |

## BEISPIEL

```
// Das Gerät 12345 langsam nach 10 Minuten dimmen
LCN_SwitchDurationMin(12345, 10, 'L', false);
```

# LCN_SwitchDurationSec

```
boolean LCN_SwitchDurationSec (int $InstanzID, int $Sekunden, string
$Abdimmzeit, boolean $Erhaltend)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Sekunden | Nach wie vielen Sekunden Abgedimmt werden soll |
| Abdimmzeit | Wie schnell abgedimmt werden soll |
| Erhaltend | Das Gerät wird nach Ablauf der Zeit nur ausgeschaltet, wenn vorher noch nicht an war |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Dimmt das Gerät mit der ID **InstanzID** nach Sekunden **Sekunden** heller. Weitere Optionen sind die Dimmgeschwindigkeit **Abdimmzeit** und ob es **Erhaltend** ist.

| ABDIMMZEIT | BESCHREIBUNG |
|---|---|
| 'K' | Kurze Abdimmzeit |
| 'M' | Mittlere Abdimmzeit |
| 'L' | Lange Abdimmzeit |

## BEISPIEL

```
// Das Gerät 12345 schnell nach 5 Sekunden dimmen
LCN_SwitchDurationSec(12345, 5, 'K', false);
```

# LCN_SwitchMemory

```
boolean LCN_SwitchMemory (int $InstanzID, int $Rampe)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Rampe | Anzahl Sekunden, 0 = Sofort |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schaltet das Gerät mit der ID **InstanzID** auf einen abgespeicherten Wert

## BEISPIEL

```
LCN_SwitchMemory(12345, 0); //Gerät sofort schalten
```

# LCN_SwitchMode

```
boolean LCN_SwitchMode (int $InstanzID, int $Rampe)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Rampe | Anzahl Sekunden, 0 = Sofort |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schaltet das Gerät mit der ID **InstanzID** um.

## BEISPIEL

```
LCN_SwitchMode(12345, 0); //Gerät sofort umschalten
```

# LCN_SwitchRelay

```
boolean LCN_SwitchRelay (int $InstanzID, boolean $Status)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Status | TRUE für An, FALSE für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Schaltet das Relais mit der ID InstanzID auf Wert Status

## BEISPIEL

```
LCN_SwitchRelay(12345, true); //Relais einschalten
```

# LCN_SwitchRelayTimer

```
boolean LCN_SwitchRelayTimer (int $InstanzID, int $Zeitfaktor)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Zeitfaktor | Zeitfaktor zwischen 1..255 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schaltet das Relais mit der ID **InstanzID** auf Zeitfaktor **Zeitfaktor**



Der Zeitfaktor ist kein linearer Wert. 1 = 30ms und 255 = 240s

## BEISPIEL

```
//Relais Zeitfaktor auf 255 stellen.
LCN_SwitchRelayTimer(12345, 255);
```

# LJQuick

Benötigt: IP-Symcon >= 7.2
Das KNX quick (Lingg&Janke) Modul hilft dabei die KNX quick Instanzen schnell und unkompliziert hinzuzufügen. Durch auswählen einer Gruppe, Kanal und einem Klick auf den Button 'Erstellen' werden die Instanzen mit den richtigen Adressen erstellt. Datum und Uhrzeit werden alle 10 Minuten an die KNX Adresse 30/3/254 gesendet.

## FUNKTIONSUMFANG

Erstellung von Instanzen anhand eines Konfigurators

## SOFTWARE-INSTALLATION

Über den Module Store das 'KNX quick (Lingg&Janke)'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'KNX quick (Lingg&Janke)'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

Der Konfigurator ist als Baum auf gebaut. Zuerst die Geräteart, darunter die verschiedenen Gruppen unter denen die verschiedenen Kanäle sind. Ist ein oder mehrere Kanäle ausgewählt, können die dazugehörigen KNX Geräte Instanzen mit den dazugehörigen Gruppenaddressen mit einem Klick auf den Button 'Erstellen' erstellt werden.
Verfügbare Geräte:
- Schalter
- Dimmen
- Beschattung
- Energie
- Wasser, Gas, Öl
- Wärmemengenzähler
- Temperatur
- Temperatur/Luftfeuchtigkeit

Jedes Gerät beinhaltet die Gruppe 1-F.
Jede Gruppe enthält die Kanäle 1-9.
Die Gerätegruppen Schalter, Dimmen, Beschattung, Temperatur und Temperatur/Luftfeuchtigkeit enthalten ebenfalls den Kanal 0.

# LJ_SendDateTime

Benötigt: IP-Symcon >= 7.2

```
boolean LJ_SendDateTime (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet das aktuelle Datum/Uhrzeit auf den KNX-Bus

## BEISPIEL

```
LJ_SendDateTime(12345);
```

# M-Bus

M-Bus ist ein standartisiertes Protokoll und Bussystem zur Verbrauchsdatenerfassung. Zur Verbindung mit IP-Symcon wird ein LAN-, seriell- oder USB-Gateway benötigt.



Folgende Geräte werden von IP-Symcon unterstützt:
Unterstützte Geräte

## INHALT


Installation
Einbindung in IP-Symcon
Konfigurator
Konfiguration


## INSTALLATION

Das **M-Bus LAN Gateway** wird per LAN mit dem PC verbunden. Standardmäßig ist DHCP und Port 5000 eingerichtet. Sollte eine individuelle IP Adresse gefordert sein, so wird das IP-Symcon "Network Configuration Tool" benötigt. Dieses steht zum Download bereit. Eine einfache Beschreibung zur Konfiguration des Gateways steht zur Verfügung. Im Anschluss ist das Gateway über die eingerichtete IP-Adresse und Port erreichbar.

## EINBINDUNG IN IP-SYMCON

Das LAN-Gateway kann über die Gerätesuche eingebunden werden. Hierfür muss als System "M-Bus Discovery" gewählt werden. Die Discovery-Instanz bietet dann die Erstellung eines M-Bus Konfigurators an. Nachdem der Konfigurator erstellt wurde, können dort Geräte gesucht und eingebunden werden.

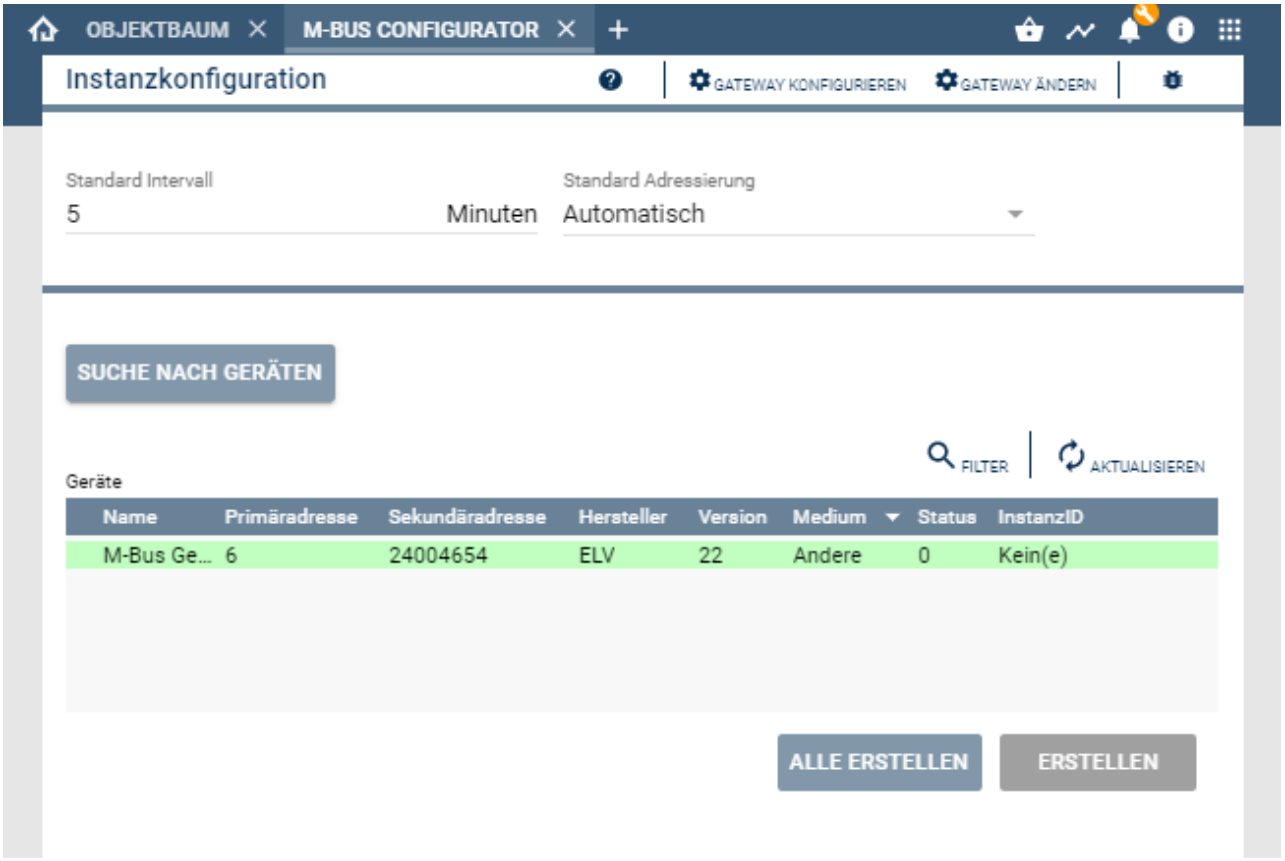
## KONFIGURATOR

Im Konfigurator können Standardeinstellungen für neu erstellte Geräte definiert werden:

| EIGENSCHAFT | BESCHREIBUNG |
|---|---|
| Standard Adressierung | Auswahl ob Geräte über die Primäradresse oder Sekundäradresse angesprochen werden sollen, bei "Automatisch" wird die Sekundäradresse verwendet, falls die Primäradresse 0 ist, ansonsten die Sekundäradresse |
| Standard Intervall | In welchen Abständen die Werte in Minuten abgefragt werden sollen |

Die Standardeinstellungen werden nur bei der Erstellung von neuen Instanzen angewandt. Eine Änderung wirkt sich also nicht auf bereits existierende Instanzen aus.
Nach der Definition der Standardeinstellungen kann durch einen Klick auf den Button "Suche nach Geräten" die Suche nach M-Bus-Geräten gestartet werden. Die Suche kann einige Minuten dauern.
Nach der Suche erscheinen die gefundenen Geräte als erstellbare Instanzen im

Konfigurator. Dazu muss ein Gerät aus der Liste ausgewählt und auf "Erstellen" geklickt werden.

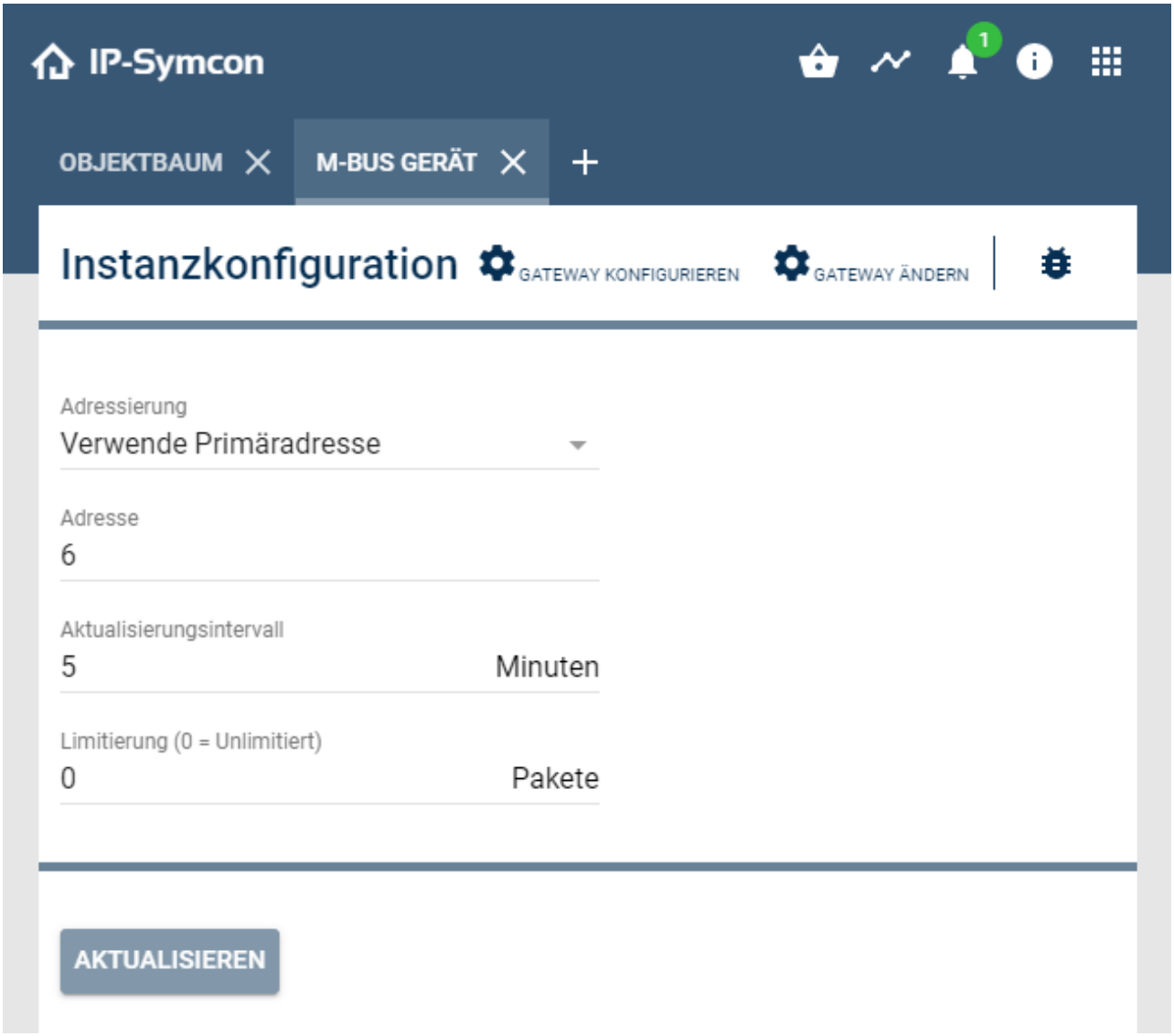



Neue Instanzen werden im Objektbaum in der Hauptkategorie erstellt. Diese erstellten Instanzen können dann entsprechend umbenannt und an anderer Stelle einsortiert werden. Auch besteht die Möglichkeit via "Konfigurieren" im Konfigurator die jeweilige Instanzkonfiguration aufzurufen.

## KONFIGURATION

Auf der Konfigurationsseite des Gerätes können verschiedene Eigenschaften eingestellt werden.

| EIGENSCHAFT | BESCHREIBUNG |
|---|---|
| Adressierung | Auswahl ob das Gerät über die Primäradresse oder Sekundäradresse angesprochen werden soll |
| Adresse | Adresse bzw. ID den Geräts |
| Aktualisierungsintervall | In welchen Abständen die Werte in Minuten abgefragt werden sollen |
| Limitierung | Limitierung der Pakete auf dem Bus (0 = Unlimitiert) |


IP-Symcon
1
OBJEKTBAUM
M-BUS GERÄT
Instanzkonfiguration
GATEWAY KONFIGURIEREN
GATEWAY ÄNDERN
Adressierung
Verwende Primäradresse
Adresse
6
Aktualisierungsintervall
5
Minuten
Limitierung (0 = Unlimitiert)
0
Pakete
AKTUALISIEREN

# Geräteliste

## UNTERSTÜTZTE GATEWAYS

Generell werden alle M-Bus Gateways unterstützt.
Explizit getestete Gateways:

| HERSTELLER | BESCHREIBUNG |
|---|---|
| SymBox mit M-Bus Erweiterung Jetzt bestellen | Seriell integriertes Modul |
| Symcon M-Bus LAN Gateway<br>Jetzt bestellen | IP Gateway |
| PiiGAB | M-Bus 800 |
| Relay | M-Bus Micro-Master |
| Relay | Pegelwandler PW3 |
| Relay | Pegelwandler PW20 |
| Relay | Pegelwandler PW60 |
| Relay | RelAir R2M HOME |
| Relay | RelAir R2M Pro |

## UNTERSTÜTZTE KOMPONENTEN

IP-Symcon unterstützt alle Geräte, die das M-Bus Protokoll nutzen.

# MBUS_UpdateValues

```
boolean MBUS_UpdateValues (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu aktualisierenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Aktualisiert die Werte mit der ID InstanzID

## BEISPIEL

```
MBUS_UpdateValues(12345);
```

# Mennekes

Benötigt: IP-Symcon >= 7.0
Mennekes bietet verschiedene Wallboxen/Ladesäulen an. Über ModBus TCP können diese ausgelesen werden. Eine Verbindung mit IP-Symcon ist über ein LAN-IP Gateway oder W-LAN, je nach Ausführung, möglich.

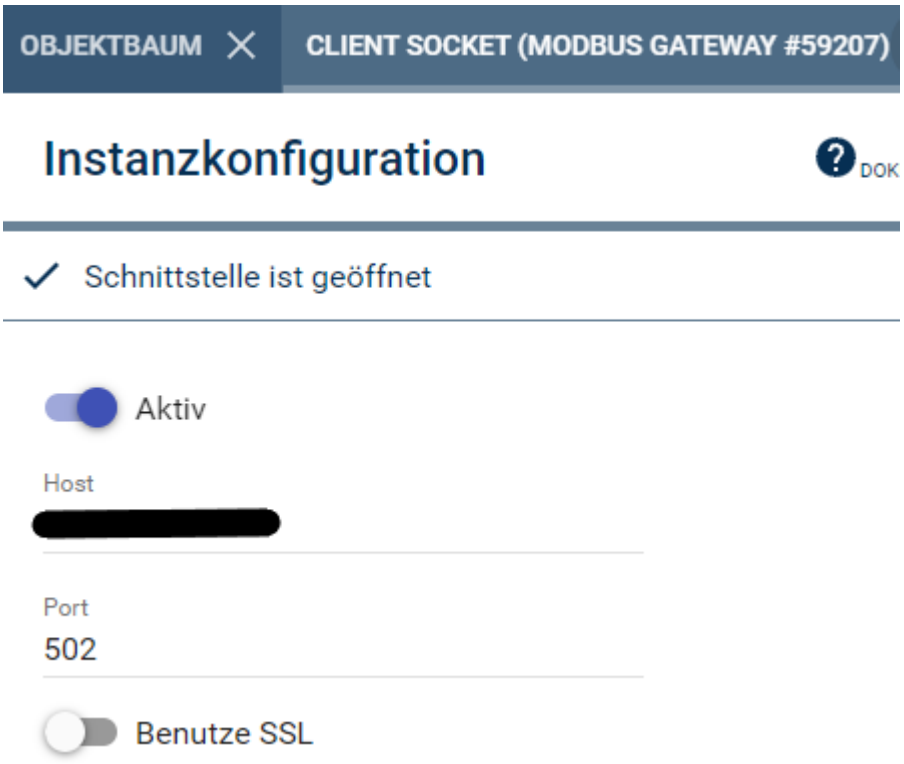
Folgende Geräte werden von IP-Symcon unterstützt:
Unterstützte Komponenten

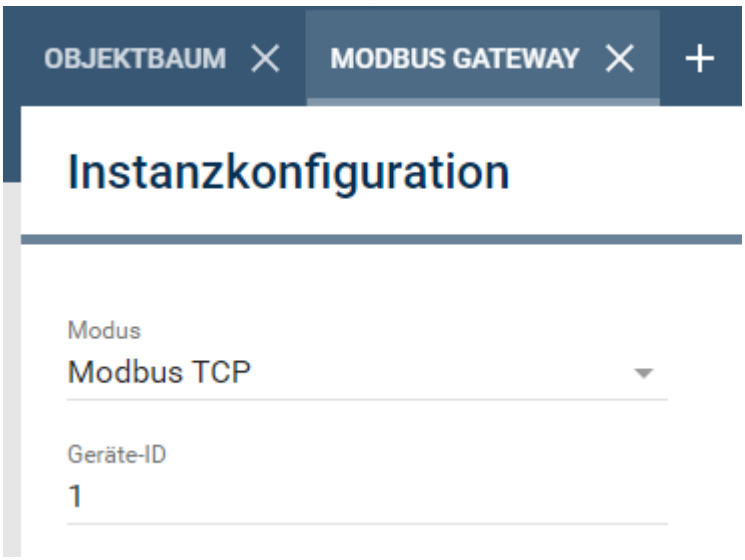
## INSTALLATION

Um Mennekes Wallboxen/Ladesäulen in IP-Symcon nutzen zu können, muss eine Verbindung via Ethernet oder W-LAN zur Wallbox vorhanden sein. Innerhalb der Wallbox/Ladesäulen ist für eine kabelgebundene Lösung ein USB->LAN Converter nötig, welcher im Lieferumfang der Wallbox/Ladesäulen mitgeliefert wird.

## EINBINDUNG IP-SYMCON

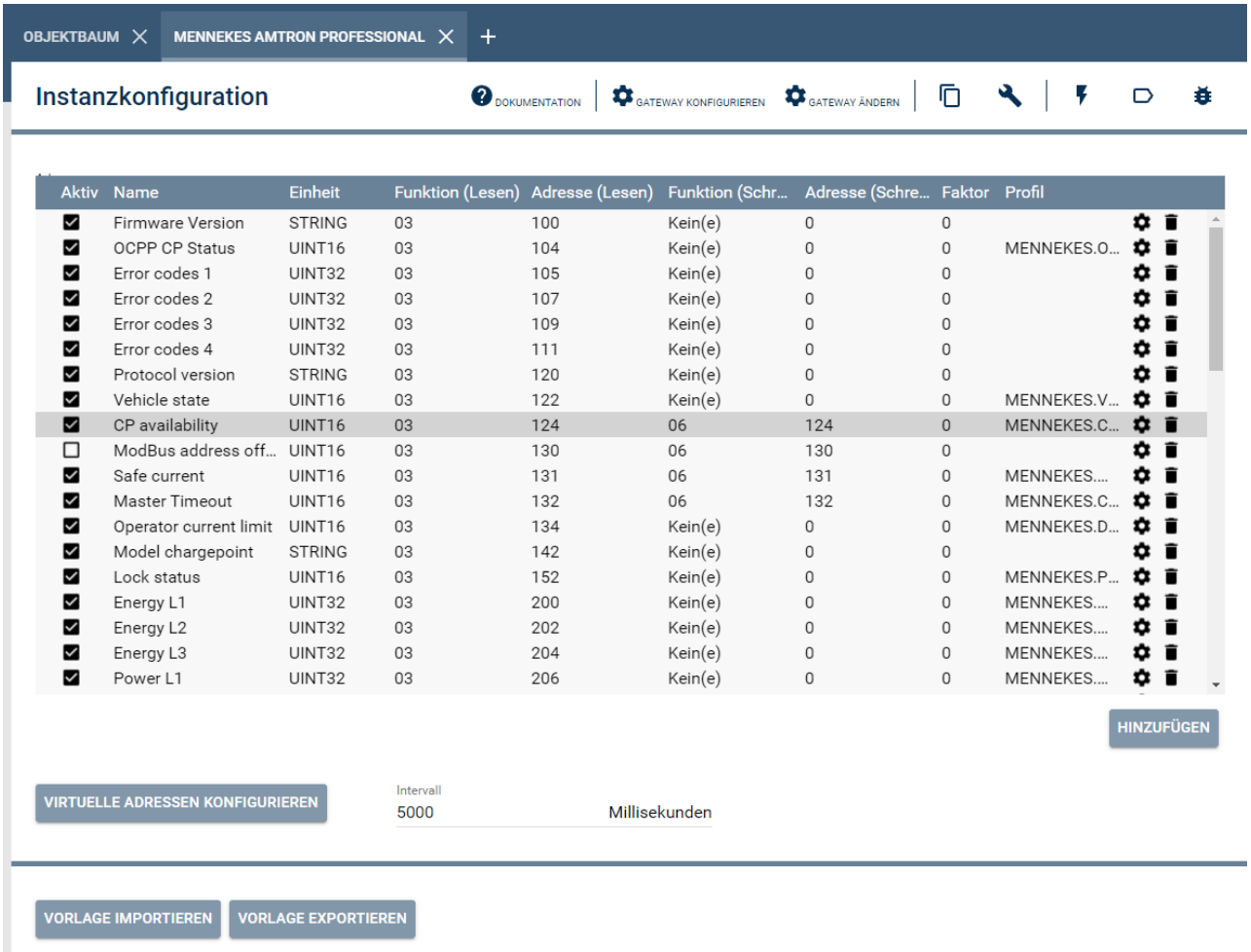
Zuerst muss eine "ModBus Device" Instanz innerhalb des Objektbaums von IP-Symcon hinzugefügt werden. Im folgenden Dialog muss die IP-Adresse der Wallbox eingetragen werden. Der Port ist Standardmässig 502. Beide Informationen können auf der Weboberfläche der WallBox/Ladesäule eingesehen werden.






Im Anschluss kann die ModBus Vorlage für Mennekes Wallboxen/Ladesäulen heruntergeladen werden. Diese beinhaltet die ganze Konfiguration des ModBus Geräts. Nach dem Herunterladen, kann die "MennekesModBusx_vx.json" über "Importieren" geladen werden. Anschließend sind alle ModBus Adressen der gängigen Mennekes Wallboxen/Ladesäulen eingerichtet.

ModBus Vorlagen für Mennekes Wallboxen/Ladesäulen:
AMTRON Xtra: Download
AMTRON Professional: Download
AMEDIO Professional: Download



## HINZUFÜGEN VON ADRESSEN

Sollen weitere Adressen hinzugefügt werden, kann dies über "Hinzufügen" realisiert werden.
Je nach Ausführung der Wallbox kann es sein, dass einzelne Adressen de-/aktiviert werden müssen. Dies kann über die Aktiv Spalte gesteuert werden.

## PROTOKOLLBESCHREIBUNG

Protokollbeschreibung für Mennekes Wallboxen/Ladesäulen:
AMTRON Professional: Download
AMEDIO Professional: Download
AMTRON Xtra: Download
AMTRON 4you: Download

# Geräteliste

Benötigt: IP-Symcon >= 7.0

## UNTERSTÜTZTE KOMPONENTEN

| SYSTEM | BESCHREIBUNG |
|---|---|
| AMEDIO Professional | In allen Versionen unterstützt |
| AMEDIO Professional+ | In allen Versionen unterstützt |
| AMEDIO Professional PnC | In allen Versionen unterstützt |
| AMEDIO Professional+ PnC | In allen Versionen unterstützt |
| AMEDIO Professional mit Shutter | In allen Versionen unterstützt |
| AMEDIO Professional PnC mit Shutter | In allen Versionen unterstützt |
| AMTRON Charge Control 11 | In allen Versionen unterstützt |
| AMTRON Charge Control 11 C2 | In allen Versionen unterstützt |
| AMTRON Professional | In allen Versionen unterstützt |
| AMTRON Professional PnC | In allen Versionen unterstützt |
| AMTRON Professional+ | In allen Versionen unterstützt |
| AMTRON Professional+ PnC | In allen Versionen unterstützt |
| AMTRON Xtra | In allen Versionen unterstützt |
| AMTRON 4you | 310, 510, 560 (In Vorbereitung) |

# Modbus RTU/TCP

ModBus ist ein Protokoll welches auf RTU (seriell binär) und TCP/IP Paketen basiert. Eine Verbindung mit IP-Symcon ist via LAN- oder Seriell-Gateway möglich.

Folgende Geräte werden von IP-Symcon unterstützt:
Unterstützte Geräte

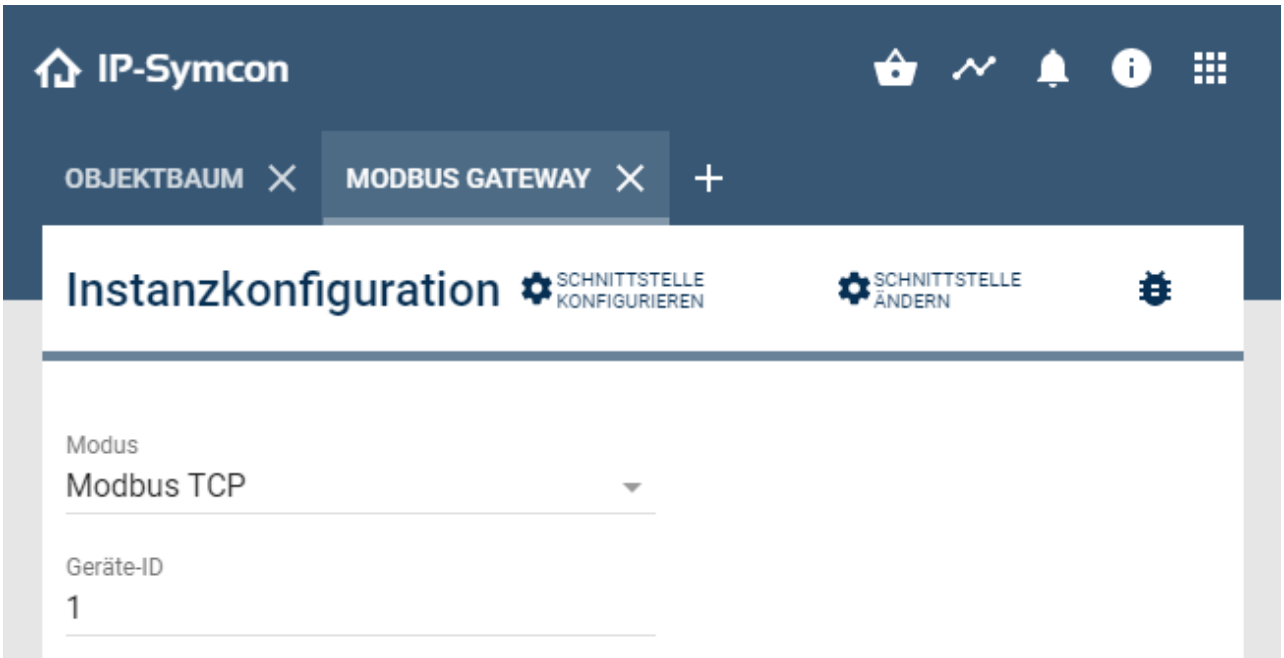
## EINBINDUNG IN IP-SYMCON

Zuerst muss eine "ModBus" Instanz innerhalb von IP-Symcon hinzugefügt werden. Dabei gibt es verschiedene Varianten, die erstellt werden können:

| INSTANZ | BESCHREIBUNG |
|---|---|
| ModBus Gerät | Instanz, welche mehrere Adressen (Coils/Register) abbilden kann und Vorlagen unterstützt (ab 7.0) |
| ModBus Coil | Instanz, welche eine einzelne Adresse (Coil) darstellt |
| ModBus Adresse | Instanz, welche eine einzelne Adresse (Register) darstellt |

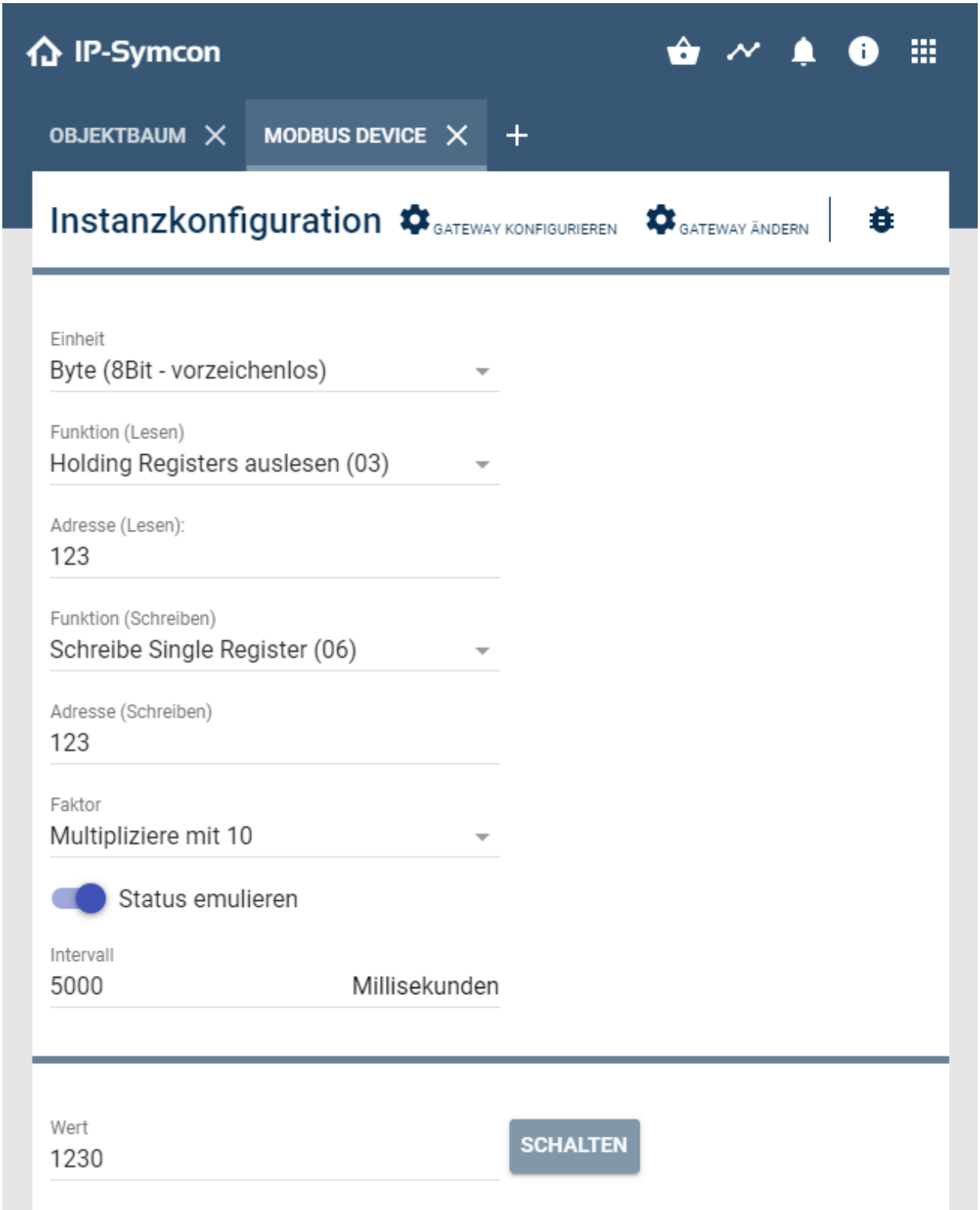
Viele Vorlagen sind in unserer Community zu finden: Vorlagen anzeigen

Anschließend muss die Konfiguration des übergeordneten Gateways und I/O Instanz an das angeschlossene Gerät angepasst werden. Es muss ausgewählt werden über welches Protokoll und Geräte-ID das Gerät angesprochen wird. Die Geräte-ID ist vorallem bei RTU angeschlossenen Geräten wichtig. Bei TCP angeschlossenen Geräten ist diese oftmals die Geräte-ID 1.



Um eine Verbindung zum Gerät herstellen zu können, müssen in der dem Gateway übergeordneten I/O Instanz, IP-Adressse und Port (Default: 502) eingetragen werden. Weitere Konfiguration (Screenshot siehe unten) ist wiefolgt.

| OPTION | BESCHREIBUNG |
|---|---|
| Einheit | Datentyp der Variable des Devices |
| Funktion (Lesen) | Funktion nach der die Werte gelesen werden sollen |
| Adresse (Lesen) | Geräteadresse des Registers aus dem gelesen werden soll |
| Funktion (Schreiben) | Funktion nach der die Werte geschrieben werden sollen |
| Adresse (Schreiben) | Geräteadresse des Registers in das Geschrieben werden soll |
| Faktor (Zahlenwerte) | Der Faktor multipliziert oder teilt den empfangenen Wert und schreibt diesen in die Variable des Devices |
| Länge (Strings) | Wenn die Länge größer 0 gesetzt ist, bestimmt es die Anzahl der Zeichen (Byte), welche abgefragt werden. Hinweis: 1 Register = 2 Zeichen (Byte) |
| Byte-Reihenfolge | Je nach Gerät muss die Byte-Reihenfolge angepasst werden. Oft wird dies nicht dokumentiert, sondern muss ausprobiert werden. Big Endian und Little Endian (Bytes vertauscht) sind die gängigen Werte. |
| Status emulieren | "Status emulieren" bedeutet, dass der Wert der Variable bei erfolgreichem Schreibbefehl auf den neuen Wert gesetzt wird und nicht auf einen Lesebefehl angewiesen ist. |
| Intervall | Wenn das Intervall größer 0 gesetzt ist, wird die Adresse (Lesen) beim ModBus-Gerät zyklisch abgefragt und die Variable des Devices aktualisiert. |




Durch die Tatsache, dass wir noch 32 Bit Systeme unterstützen werden Int64 Werte als Float64 abgebildet.

## MODBUS GERÄT

Sollen mehrere Adressen für ein Gerät abgefragt werden, sollte das Modbus-Gerät genutzt werden. Hier können mehrere Adressen als Tabelle eigetragen werden, sowie virtuelle Adressen definiert werden.

Konfiguration:

| NAME | BESCHREIBUNG |
|---|---|
| Adressen | Tabellarische Aufführung der definierten Adressen |
| Virtuelle Adressen konfigurieren | Button mit dem eine Liste virtueller Adressen definiert werden kann |
| Byte-Reihenfolge | Je nach Gerät muss die Byte-Reihenfolge angepasst werden. Oft wird dies nicht dokumentiert, sondern muss ausprobiert werden. Big Endian und Little Endian (Bytes vertauscht) sind die gängigen Werte. |
| Intervall | Wenn das Intervall größer 0 gesetzt ist, wird die Adresse (Lesen) beim ModBus-Gerät zyklisch abgefragt und die Variable des Devices aktualisiert. |
| Vorlage importieren | Durch diesen Button, kann eine Vorlage der Tabellen schnell eingefügt werden. |
| Vorlage exportieren | Durch diesen Button, kann eine Vorlage aus den vorgenommenen Einstellungen erstellt werden. Die Profile werden dabei ebenfalls exportiert. |

Inhalt der Adressen-Liste:

| NAME | BESCHREIBUNG |
|---|---|
| Aktiv | Gibt an, ob die Variable erstellt werden soll |
| Name | Name, der als Variablenname genutzt wird |
| Einheit | Datentyp des Registers |
| Funktion (Lesen) | Funktion nach der die Werte gelesen werden sollen |
| Adresse (Lesen) | Registers aus dem gelesen werden soll |
| Funktion (Schreiben) | Funktion nach der die Werte geschrieben werden sollen |
| Adresse (Schreiben) | Registers in das Geschrieben werden soll |
| Profil | Profil, welches die Variable bekommen soll |

Unter den Expertenoptionen kann der Ident individuell angepasst und auch eine Übersetzung für den Variablennamen mitgegeben werden.
Inhalt der virtuellen Adressen Liste:

| NAME | BESCHREIBUNG |
|---|---|
| Aktiv | Die Variable wird erstellt |
| Name | Name der als Variablenname genutzt wird |
| Variablentyp | Typ der Variable |
| Profil | Profil der Variable |
| Lesen | Skript, welches die Werte der virtuelle Adresse zu Verfügung stellt |
| Schreiben | Skript, welches die Werte an die Variablen weitergibt. |

Beim Lese-Skript sind in $VALUES alle Werte mit den jeweiligen Idents verfügbar. Der neue Wert, der in der Variable der virtuellen Adresse laden soll, wird einfach mit return zurückgegeben. Sofern kein Wert geschrieben werden soll (z.B. weil ungültig) kann null zurückgegeben werden. (ab 7.1)

```
return ($VALUES["A_3_3_23296"] + $VALUES["A_3_3_23298"] +
$VALUES["A_3_3_23300"])/3;
```

Beim Schreibt-Skript wird ein Assoziatives Array als return erwartet, welches alle Ident enthält, in die geschrieben werden soll. $VALUE enthält den neuen Wert, der per RequestAction angefordert wurde.

```
return ["A_3_3_23296" => $VALUE];
```

## FUNCTIONCODES

Je nach FunctionCode muss auch eine passende Adresse eingetragen werden. Diese kann der Tabelle entnommen werden.
Die nachfolgende Tabelle bietet ebenfalls einen Überblick, welcher FunctionCode bei welcher Parameterierung innerhalb von IP-Symcon gesendet wird:

| FUNCTIONCODE | GERÄTEADRESSE | LESE-/ SCHREIBADRESSE | NAME |
|---|---|---|---|
| 0x01 (1) | 1 - 10000 | Geräteadresse - 1 | Read Coils |
| 0x05 (5) | 1 - 10000 | Geräteadresse - 1 | Write Single Coil |
| 0x02 (2) | 10001 - 20000 | Geräteadresse - 10001 | Read Discrete Inputs |
| 0x03 (3) | 40001 - 50000 | Geräteadresse - 40001 | Read Holding Registers |
| 0x10 (16) | 40001 - 50000 | Geräteadresse - 40001 | Write Multiple registers |
| 0x04 (4) | 30001 - 40000 | Geräteadresse - 30001 | Read Input Registers |



Wenn Beispielsweise auf der Adresse 40123 geschrieben (FunctionCode = 0x10) werden soll, muss der passende Datentyp (Einheit) ausgewählt werden und die Schreibadresse 122 (40123 - 40001) eingetragen werden.



Manche Hersteller halten sich nicht an diese Konvention. Es muss in der jeweiligen Anleitung oder Datenblätter nachgesehen werden, ob die Geräteadresse je nach Functioncode abgezogen werden muss oder absolute Adressen gelten

## DATENTYPEN

| DATENTYP | VORZEICHEN | BITS |
|---|---|---|
| BOOL | vorzeichenlos | 1 |
| UINT8MSB | vorzeichenlos | 8 |
| UINT8LSB | vorzeichenlos | 8 |
| UINT16 | vorzeichenlos | 16 |
| UINT32 | vorzeichenlos | 32 |
| UINT64 | vorzeichenlos | 64 |
| INT8MSB | vorzeichenbehaftet | 8 |
| INT8LSB | vorzeichenbehaftet | 8 |
| INT16 | vorzeichenbehaftet | 16 |
| INT32 | vorzeichenbehaftet | 32 |
| INT64 | vorzeichenbehaftet | 64 |
| FLOAT32 | vorzeichenbehaftet | 32 |
| FLOAT64 | vorzeichenbehaftet | 64 |
| STRING | | |

# Geräteliste

## UNTERSTÜTZTE GATEWAYS/KOMPONENTEN

IP-Symcon unterstützt alle Geräte, die das ModBus RTU oder ModBus TCP Protokoll nutzen.
Insbesondere Geräte und SPS von Wago/Beckhoff/ABB.

## UNTERSTÜTZTE KOMPONENTEN

IP-Symcon unterstützt alle Komponenten.

| HERSTELLER | BESCHREIBUNG |
| --- | --- |
| SymBox mit RS485 (ModBus RTU) Erweiterung<br>Jetzt bestellen | Seriell integriertes Modul |
| Exsys EX-6051 (unverbindliche Empfehlung) | Seriell auf LAN Wandler für Modus "ModBus RTU über TCP" |
| Waveshare RS485 to Ethernet Converter (unverbindliche Empfehlung) | Seriell auf LAN Wandler für Modus "ModBus RTU über TCP" |

# ModBus_RequestRead

```
boolean ModBus_RequestRead (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Führt einen erneuten Lesevorgang für das Gerät mit der ID **InstanzID** aus.
Sobald die Funktion ausgeführt wurde, sind die neuen Werte in den Status Variablen abrufbar.

## BEISPIEL

```
ModBus_RequestRead(12345); //Gerät auslesen
```

# ModBus_WriteCoil

```
boolean ModBus_WriteCoil (int $InstanzID, boolean $Status)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Status | TRUE für An, FALSE für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Schreibt den Wert **Wert** in die konfigurierte Adresse (Schreiben) der Instanz mit der ID **InstanzID**.

## BEISPIEL

```
//Schaltet die Instanz mit der ID 12345 ein
ModBus_WriteCoil(12345, true);
```

# ModBus_WriteRegister

```
boolean ModBus_WriteRegister (int $InstanzID, float $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | 32Bit Gleitkommawert nach IEEE754 oder Ganzzahlig |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schreibt den Wert **Wert** in die konfigurierte Adresse (Schreiben) der Instanz mit der ID **InstanzID**, ohne den genauen Datentyp zu wissen.



Diese Funktion kann nicht für Bit-Werte genutzt werden. Für Bit-Werte muss ModBus_WriteCoil genutzt werden.

## BEISPIEL

```
////Schreibt auf das Register der Instanz mit der ID 12345 den Wert
23,5
ModBus_WriteRegister(12345, 23.5);

////Schreibt auf das Register der Instanz mit der ID 23456 den Wert
12
ModBus_WriteRegister(23456, 12);
```

# ModBus_WriteRegisterByte

```
boolean ModBus_WriteRegisterByte (int $InstanzID, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | 0-255 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schreibt den Wert **Wert** in die konfigurierte Adresse (Schreiben) der Instanz mit der ID **InstanzID**.

## BEISPIEL

```
//Schreibt auf das Register der Instanz mit der ID 12345 den Wert
123
ModBus_WriteRegisterByte(12345, 123);
```

# ModBus_WriteRegisterChar

Benötigt: IP-Symcon >= 4.4

```
boolean ModBus_WriteRegisterChar (int $InstanzID, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | -128 bis 127 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG



Die Funktion hieß bis einschließlich Version 4.3
ModBus_WriteRegisterShortInt

Schreibt den Wert **Wert** in die konfigurierte Adresse (Schreiben) der Instanz mit der ID **InstanzID**.

## BEISPIEL

```
//Schreibt auf das Register der Instanz mit der ID 12345 den Wert
-123
ModBus_WriteRegisterChar(12345, -123);
```

# ModBus_WriteRegisterDWord

```
boolean ModBus_WriteRegisterDWord (int $InstanzID, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | 0-4294967295 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schreibt den Wert **Wert** in die konfigurierte Adresse (Schreiben) der Instanz mit der ID **InstanzID**.



Aufgrund einer Beschränkung auf 32-Bit Integer innerhalb von PHP, kann nur der Bereich bis 2147483647 genutzt werden.

## BEISPIEL

```
//Schreibt auf das Register der Instanz mit der ID 12345 den Wert
123
ModBus_WriteRegisterDWord(12345, 123);
```

# ModBus_WriteRegisterInt64

```
boolean ModBus_WriteRegisterInt64 (int $InstanzID, float $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | -9223372036854775808 bis 9223372036854775807 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schreibt den Wert **Wert** in die konfigurierte Adresse (Schreiben) der Instanz mit der ID **InstanzID**.

## BEISPIEL

```
//Schreibt auf das Register der Instanz mit der ID 12345 den Wert
235689
ModBus_WriteRegisterInt64(12345, 235689);
```

# ModBus_WriteRegisterInteger

```
boolean ModBus_WriteRegisterInteger (int $InstanzID, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | -2147483648 bis 2147483647 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schreibt den Wert **Wert** in die konfigurierte Adresse (Schreiben) der Instanz mit der ID **InstanzID**.

## BEISPIEL

```
//Schreibt auf das Register der Instanz mit der ID 12345 den Wert
-123
ModBus_WriteRegisterInteger(12345, -123);
```

# ModBus_WriteRegisterReal

```
boolean ModBus_WriteRegisterReal (int $InstanzID, float $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | 32Bit Gleitkommawert nach IEEE754 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schreibt den Wert **Wert** in die konfigurierte Adresse (Schreiben) der Instanz mit der ID **InstanzID**.

## BEISPIEL

```
//Schreibt auf das Register der Instanz mit der ID 12345 den Wert
23,5
ModBus_WriteRegisterReal(12345, 23.5);
```

# ModBus_WriteRegisterReal64

```
boolean ModBus_WriteRegisterReal64 (int $InstanzID, float $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | 64Bit Gleitkommawert nach IEEE754 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schreibt den Wert **Wert** in die konfigurierte Adresse (Schreiben) der Instanz mit der ID **InstanzID**.

## BEISPIEL

```
//Schreibt auf das Register der Instanz mit der ID 12345 den Wert
23,5
ModBus_WriteRegisterReal64(12345, 23.5);
```

# ModBus_WriteRegisterShort

Benötigt: IP-Symcon >= 4.4

```
boolean ModBus_WriteRegisterShort (int $InstanzID, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | -32768 bis 32767 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG



Die Funktion hieß bis einschließlich Version 4.3
ModBus_WriteRegisterSmallInt

Schreibt den Wert **Wert** in die konfigurierte Adresse (Schreiben) der Instanz mit der ID **InstanzID**.

## BEISPIEL

```
//Schreibt auf das Register der Instanz mit der ID 12345 den Wert
-123
ModBus_WriteRegisterShort(12345, -123);
```

# ModBus_WriteRegisterString

```
boolean ModBus_WriteRegisterString (int $InstanzID, string $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | String, welcher auf Register geschrieben werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Schreibt den Wert **Wert** in die konfigurierte Adresse (Schreiben) der Instanz mit der ID **InstanzID**.

## BEISPIEL

```
//Schreibt auf das Register der Instanz mit der ID 12345 den String
"Hallo Welt"
ModBus_WriteRegisterString(12345, "Hallo Welt");
```

# ModBus_WriteRegisterWord

```
boolean ModBus_WriteRegisterWord (int $InstanzID, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | 0-65535 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schreibt den Wert **Wert** in die konfigurierte Adresse (Schreiben) der Instanz mit der ID **InstanzID**.

## BEISPIEL

```
//Schreibt auf das Register der Instanz mit der ID 12345 den Wert
123
ModBus_WriteRegisterWord(12345, 123);
```

# Möhlenhoff Alpha 2

Benötigt: IP-Symcon >= 4.0
Das Modul dient zum Empfangen und Schalten von Möhlenhoff Alpha2 Daten.

## FUNKTIONSUMFANG

Setzen von Variablen via Skriptbefehl oder Visualisierung.
Auslesen der Daten der Möhlenhoff Alpha 2

## SOFTWARE-INSTALLATION

Über den Module Store das Modul Möhlenhoff Alpha 2 installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Möhlenhoff Alpha 2'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| IP-Adresse | IP-Adresse der Alpha 2 |
| Intervall | Einstellbares Intervall in Sekunden in denen die Daten der Alpha 2 abgefragt werden sollen. (Standard: 0) |
| Heizzone 1..12 | Aktiviert das Auslesen der jeweiligen Heizzone. (Die Heizkreise werden immer alle ausgelesen) |
| "Gerät auslesen" | Liest die Daten manuell aus. Muss auch einmalig zu Beginn gemacht werden. |

## STATUSVARIABLEN UND PROFILE

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

Die Variablen werden automatisch angelegt.

| NAME (BASIS) | TYP | BESCHREIBUNG |
|---|---|---|
| Basis Absenkdifferenztemperatur | Float | Dient als "Kategorie" in der sich alle überwachten Standorte, sowie der Zeitstempel und Längen-/Breitengrad befinden. Wird pro Gerät erstellt. |
| Basis Absenkeingang | Boolean | |
| Basis Automatische Zeitumstellung | Boolean | Ein/Aus der automatischen Zeitumstellung |
| Basis Betriebsart Heizen/ Kühlen (CO-Eingang) | Boolean | Steuert die Betriebsart (Ein = Kühlen, Aus = Heizen) |
| Basis Frostschutz | Boolean | Ein/Aus des Frostschutzes |
| Basis Frostschutz Temperatur | Integer | Temperatur des Frostschutzes |
| Basis HW Version | String | Hardware Version der Basis |
| Basis ID | String | ID der Basis |
| Basis Kühlen-Modus | Boolean | Ein/Aus des Kühlenmodus |
| Basis Rang im Systemverbund | Integer | Rand der Basis (0 = Standalone, 1 = Master, 2 = Slave) |
| Basis Smartstart ein/aus | Boolean | Ein/Aus des Smartstart |
| Basis SW ETH Version | String | Software Version Ethernet |
| Basis SW STM Version | String | Software Version STM |
| Basis Taupunktsensor | Boolean | Ein/Aus des Taupunktsensors |
| Basis Temperaturbegrenzer | Boolean | Ein/Aus des Temperaturbegrenzers |
| Basis Temperatureinheit | Integer | Einstellung ob Fahrenheit oder Celsius (0 = °C, 1 = °F) |
| Basis Urlaub Status | Integer | Urlaubstatus (0 = aus, 1 = geplant, 2 = aktiv) |
| Basis Urlaub Temperatur Heizen | Float | Temperatur während des Urlaubs |

| NAME (HEIZZONE 1..12) | TYP | BESCHREIBUNG |
|---|---|---|
| Heizzone 01..12 Ist Temperatur | Float | Aktuelle Ist Temperatur |
| Heizzone 01..12 Soll Temperatur | Float | Soll Temperatur |
| Heizzone 01..12 Betriebsmodus | Integer | Betriebsmodus der Heizzone (0 = Auto, 1 = Tag, 2 = Nacht) |
| Heizzone 01..12 Ist Temperatur Ext. Sensor | Float | Aktuelle Ist Temperatur des externen Sensors |
| Heizzone 01..12 Anwesenheit | Boolean | Ein/Aus der Anwesenheit |
| Heizzone 01..12 Heizsystem | Integer | Das verwendete Heizsystem (0 = FBH-Standard, 1 = FHB-Niedrigenergie, 2 = Radiator, 3 = Konvektor passiv, 4 = Konvektor aktiv) |
| Heizzone 01..12 Korrektur Ist-Werterfassung | Float | Korrektur des Ist-Werts (-2.0 .. +2.0) |
| Heizzone 01..12 Name | String | Name der Heizzone |
| Heizzone 01..12 Partyschaltung (Stunden) | Integer | Wieviele Stunden die Partyschaltung laufen soll |
| Heizzone 01..12 Partyschaltung Restzeit (min) | Integer | Wielange die Partyschaltung noch aktiv ist |
| Heizzone 01..12 Programm Wochenende | Integer | Welches Zeitprogramm am Wochenende aktiv ist |
| Heizzone 01..12 Programm wochentags | Integer | Welches Zeitprogramm unter der Woche aktiv ist |
| Heizzone 01..12 Soll Temperatur Max | Float | Einstellbereich Soll-Temperatur (maximal) |
| Heizzone 01..12 Soll Temperatur Min | Float | Einstellbereich Soll-Temperatur (minimal) |
| Heizzone 01..12 Status | Boolean | Statusanzeige (0 = OK, 1 = Error) |

| NAME (HEIZKREIS 1..12) | TYP | BESCHREIBUNG |
|---|---|---|
| Heizkreis 1..12 Aktiv | Boolean | Ob der Heizkreis aktiv ist (0 = Nein, 1 = Ja) |
| Heizkreis 1..12 Aktor | Boolean | Ein/Aus des Stellantriebs |
| Heizkreis 1..12 Aktor Prozent | Integer | Prozentuale Aktivität des Aktors (Nur bei SW STM Version 2.02 oder höher) |
| Heizkreis 1..12 Status | Integer | Status (0 = Aus, 1 = An, 2 = Fehler) |
| Heizkreis 1..12 zugewiesene Heizzone | Integer | Welche Heizzone mit dem Heizkreis verbunden ist |

**PROFILE:**

| NAME | TYP |
|---|---|
| MH.AntifreezeTemp | Integer |
| MH.TemperatureUnit | Integer |
| MH.SummerWinter | Boolean |
| MH.ChangeOver | Boolean |
| MH.Mode | Integer |
| MH.EcoDiff | Float |
| MH.VacationState | Integer |
| MH.PumpTime | Integer |
| MH.RelaisTime | Integer |
| MH.EmergencyTime | Integer |
| MH.PWMCycle | Integer |
| MH.PWMPercent | Integer |
| MH.HeatAreaMode | Integer |
| MH.HeatAreaProgram | Integer |
| MH.HeatAreaParty | Integer |
| MH.HeatAreaPartyRemainingTime | Integer |
| MH.HeatAreaState | Boolean |
| MH.HeatAreaRPMMotor | Integer |
| MH.HeatingSystem | Integer |
| MH.HeatAreaTActualTemp | Float |
| MH.HeatAreaTTarget | Float |
| MH.HeatAreaTHeatCool | Float |
| MH.HeatAreaOffset | Float |
| MH.HeatAreaBlockHC | Integer |
| MH.HeatAreaHeatCTRLState | Integer |
| MH.HeatAreaNr | Integer |
| MH.HeatCtrlActorPercent | Integer |

## VISUALISIERUNG

Über die Visualisierung oder in den mobilen Apps werden Werte angezeigt und können wenn möglich geändert werden.
Wir empfehlen aufgrund der Menge der Variablen die Anzeige der gesamten Alpha 2 in der Verwaltungskonsole zu deaktivieren und alle gewünschten Variablen via Links in einer eigenen Kategorie anzeigen zu lassen.

# MA2_RequestStatus

Benötigt: IP-Symcon >= 4.0

```
boolean MA2_RequestStatus (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Ruft die in der Alpha2 mit der InstanzID **InstanzID** gespeicherten Werte auf und setzt die dazugehörigen Variablen.

## BEISPIEL

```
MA2_RequestStatus(12345);
```

# MA2_WriteValue

Benötigt: IP-Symcon >= 4.0

```
boolean MA2_WriteValue (int $InstanzID, string $Ident, variant
$Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| Ident | Ident der Variable |
| Wert | Wert der Variable |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schreibt den Wert **Wert** in die Variable mit dem Ident **Ident** in der Möhlenhoff Alpha 2 mit der InstanzID **InstanzID**.

## BEISPIEL

```
MA2_WriteValue(12345 , "HEATAREA1_T_TARGET", 23.3);
```

# MQTT

MQTT ist ein offenes Nachrichtenprotokoll für Machine-to-Machine-Kommunikation (M2M).
IP-Symcon agiert in diesem Zusammenhang sowohl als Server (Broker) als auch Client.



Folgende Geräte werden von IP-Symcon unterstützt:
Unterstützte Geräte

## EINBINDUNG IN IP-SYMCON

Zur Verwendung von MQTT stehen IP-Symcon sowohl Clients als auch Server (Broker) zur Verfügung.
Einrichtung von MQTT-Server
Einrichtung von MQTT-Client

# Geräteliste

## UNTERSTÜTZTE GATEWAYS/KOMPONENTEN

IP-Symcon unterstützt alle MQTT Client Geräte.

## UNTERSTÜTZTE FUNKTIONEN

### AB IP-SYMCON 5.1

MQTT 3.1/3.1.1 kompatibel
Clients können sich verbinden/trennen
Clients können Nachrichten auf Topics schicken (QoS 0-2)
Clients können Topics (inkl. Filter +/#) abonnieren (Diese werden immer mit QoS 0 gesendet)
Clients können sich mit Benutzername/Kennwort authentifizieren

### AB IP-SYMCON 5.3

Retain (ab 5.5 auch Persistent)
Session Management (Siehe Timeout und Queue Limit Einstellungen im Server)
Last Will/Testament
Publish eines Clients wird nur in IP-Symcon ausgewertet und nicht an die anderen Clients weitergeleitet

### AB IP-SYMCON 5.5

Persistentes Retain
MQTT Client
TLS (Server Socket)
Keep Alive (+ Last Will bei Timeout versenden)

### NOCH FEHLENDE FUNKTIONEN

Folgende Funktionen fehlen und sind geplant
QoS 1-2 beim Senden

# MQTT Client

MQTT Clients haben einen MQTT Server abonniert und können Werte von diesem Empfangen und darstellen.

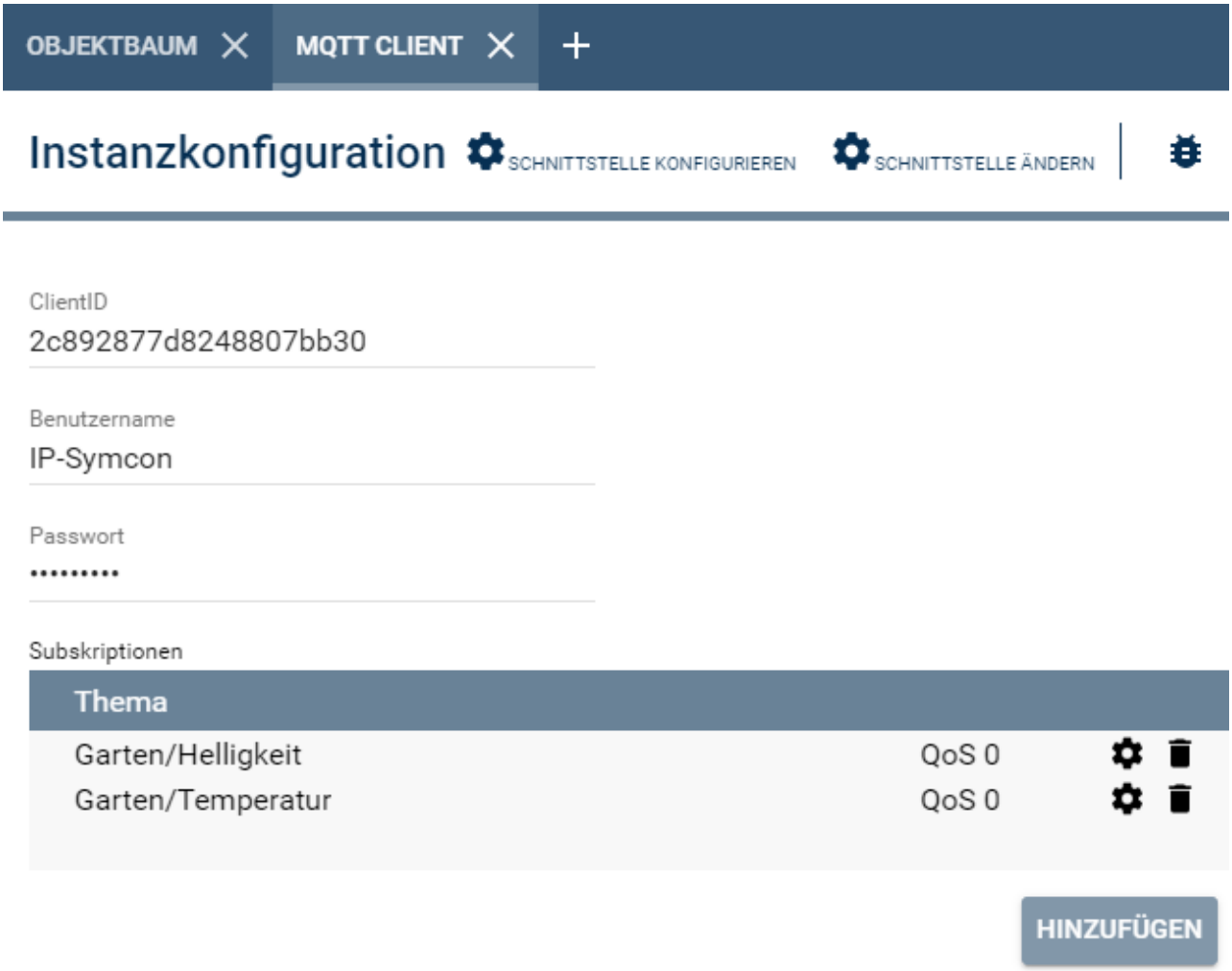
## VIDEO-TUTORIAL ZUR EINRICHTUNG

## EINBINDUNG IN IP-SYMCON

Um eine einfache und komfortable Einbindung zu ermöglichen, wird empfohlen den Konfigurator für MQTT Client zu nutzen. Dieser kann im Objektbaum über "+"->"Instanz" via Schnellfilter "MQTT Konfigurator" hinzugefügt werden.
In der Instanzkonfiguration des dazughörigen MQTT Client (Gateway) können die gewünschten Subskriptions angelegt werden. Eine "#" ist der Standard und es werden alle Daten des MQTT Servers abonniert.
In der Instanzkonfiguration des dazugehörigen Client Sockets kann die Hostadresse und der Port definiert sowie die Verbindung aktiv gesetzt werden. Der Port muss identisch mit dem Port des MQTT-Servers sein.
Nach der Einrichtung des MQTT-Client Konfigurators ist der nächste Schritt das Hinzufügen und Erstellen von Geräteinstanzen.

### HINZUFÜGEN VON GERÄTEN

Der Client empfängt die Werte der abonnierten Themen des MQTT Servers. Der Konfigurator kann diese Daten lesen und zeigt alle Themen, welche sich gemeldet haben an.
Über "Aktualisieren" wird die Anzeige aller empfangenen Themen und deren zuletzt empfangenen Nutzdaten aktualisiert.
Über "Erstellen" kann die ausgewählte Instanz erstellt werden.
Der MQTT Client hat eine voreingerichtete ClientID. Der Benutzername und das Passwort müssen mit denen des Servers übereinstimmen.




Wenn als Subkription die "#" eingetragen wird, werden alle Themen eines MQTT Servers Empfangen.

Es können dann über den Konfigurator oder manuell erstellten Devices Nachrichten an den MQTT Server gesendet werden. Dieser leitet diese dann alle anderen Clients weiter.




## BEISPIEL

Skriptbeispiel für das Publishen eines Wertes auf ein Thema.
Das Thema ist durch die MQTT Device Instanz definiert.

```
//Publishen von "EinTollerWertZumPublishen" auf die Variable mit der
ID 12345
RequestAction(12345, "EinTollerWertZumPublishen");
```

# MQTT-Server

Der MQTT-Server in IP-Symcon agiert als sogenannter MQTT-Broker. Clients können sich bei diesem anmelden und bekommen ihre Daten über diesen.

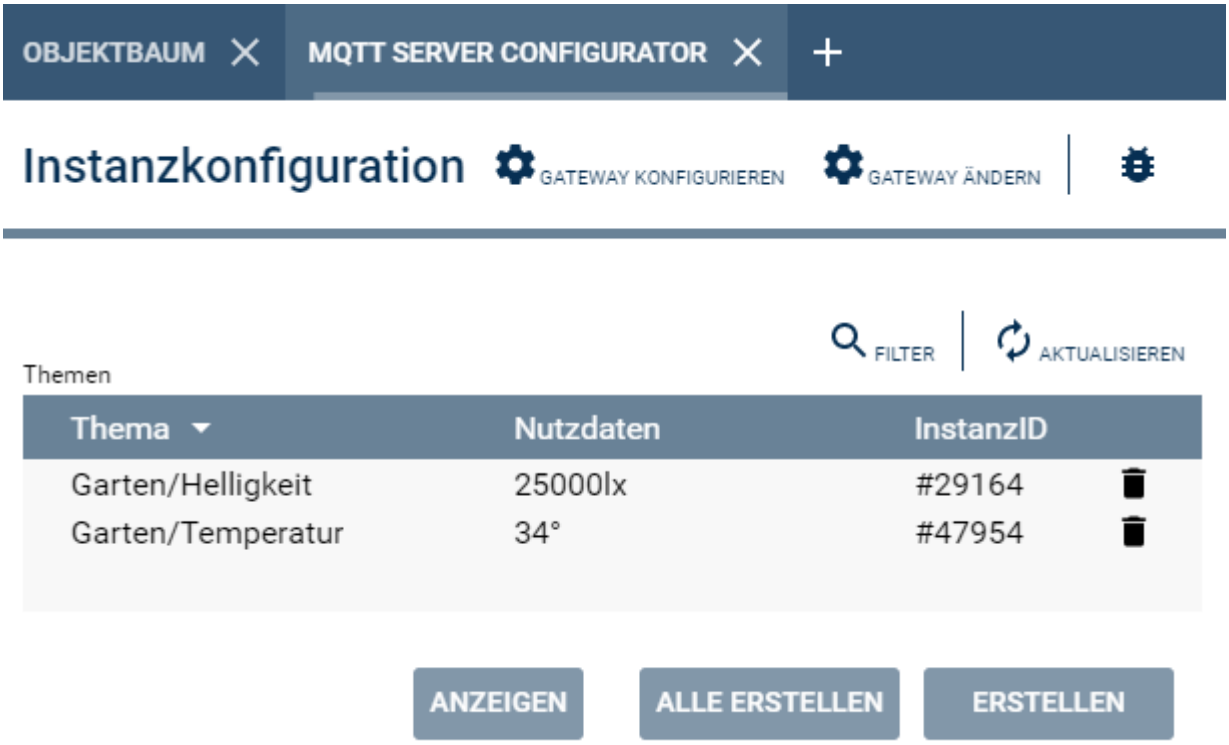
## VIDEO-TUTORIAL ZUR EINRICHTUNG

## EINBINDUNG IN IP-SYMCON

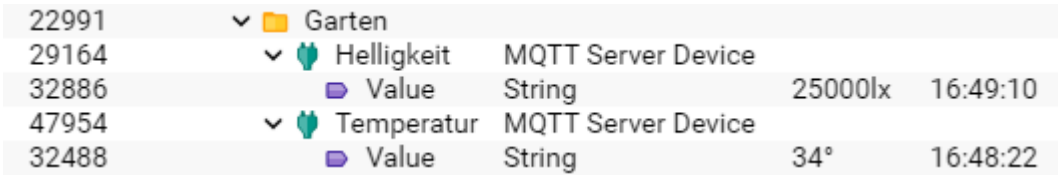
Um eine einfache und komfortable Einbindung zu ermöglichen, wird empfohlen den Konfigurator für MQTT-Server zu nutzen. Dieser kann im Objektbaum über "+"->"Instanz" via Schnellfilter "MQTT Server" hinzugefügt werden.
IP-Symcon agiert als MQTT-Server (Broker) und benötigt dafür eine geöffnete Server Socket Schnittstelle. In der Instanzkonfiguration des dazughörigen Server Sockets kann der Port definiert und die Verbindung aktiv gesetzt werden.
Des Weiteren kann in der Instanzkonfiguration des MQTT-Servers ein Benutzername und festgelegt werden. Die MQTT Clients müssen dann auf den eingerichteten Port senden.
Nach der Einrichtung des MQTT Konfigurators ist der nächste Schritt das Hinzufügen und Erstellen von Geräteinstanzen.

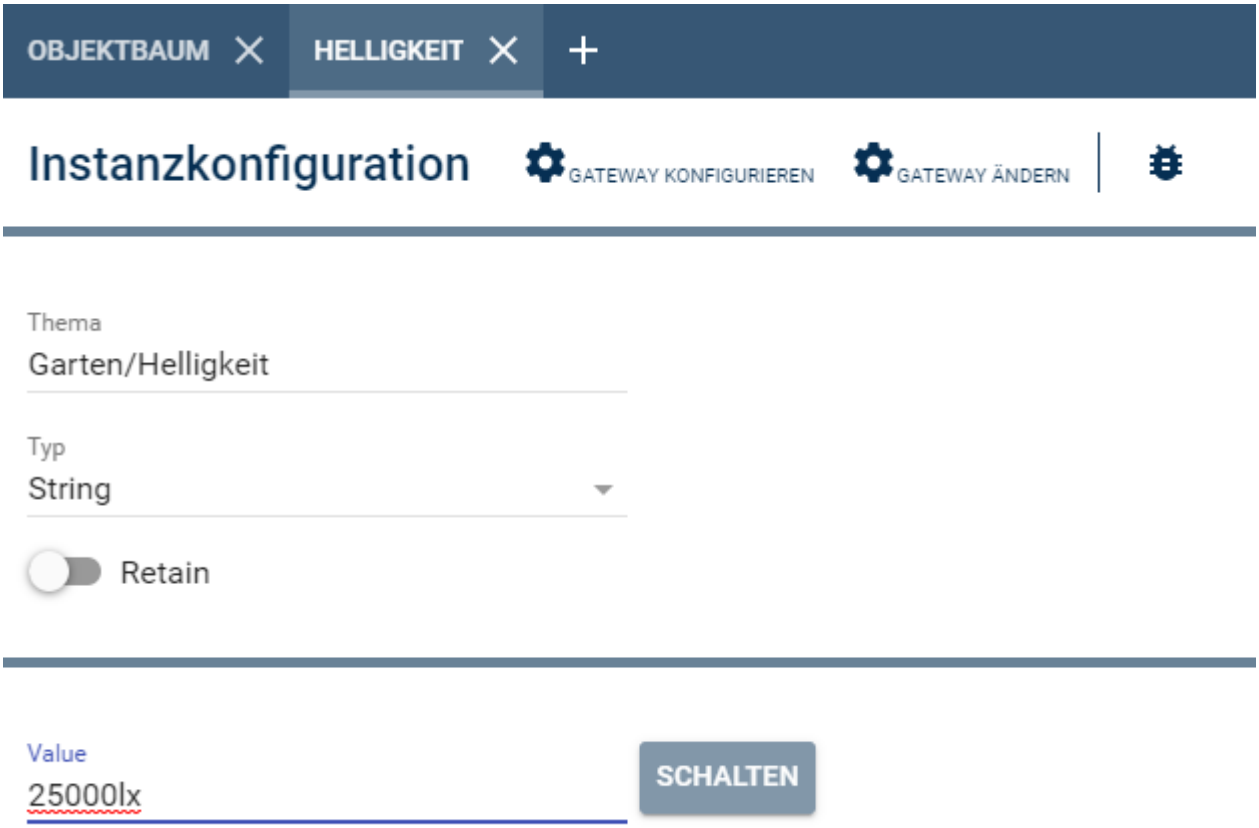
### HINZUFÜGEN VON GERÄTEN

Der Server (Broker) empfängt die Werte und Eigenschaften der MQTT Geräte. Der Konfigurator kann diese Daten lesen und zeigt alle Geräte, welche sich gemeldet haben an.
Sollte ein Gerät nicht angezeigt werden, muss sichergestellt werden, dass dieses Gerät eine Nachricht an den Server geschickt hat.
Über "Aktualisieren" wird die Anzeige aller empfangenen Themen und deren zuletzt empfangenen Nutzdaten aktualisiert.
Über "Erstellen" kann die ausgewählte Instanz erstellt werden.




Der Konfigurator erstellt den Pfad des Themas mit den passenden Kategorien und die Nutzdatenvariable vom Datentyp "String". Sofern ein anderer Datentyp gewünscht wird, kann dieser in der Instanz umgestellt werden.






Geräte, welche selbst nur Empfangen ("Subscriben") und nicht senden ("Publishen") können, können nicht über den Konfigurator erstellt werden, da dieser niemals etwas von den Geräten empfängt. Für diese Geräte muss ein MQTT-Device erstellt, ein Thema eingetragen und Datentyp festgelegt werden. (Siehe auch: Instanz erstellen)

## BEISPIEL

Skriptbeispiel für das Publishen eines Wertes auf ein Thema.
Das Thema ist durch die MQTT Device Instanz definiert.

```
//Publishen von "EinTollerWertZumPublishen" auf die Variable mit der
ID 12345
RequestAction(12345, "EinTollerWertZumPublishen");
```

# NEA Smart

Benötigt: IP-Symcon >= 4.0



Dieses Modul ist nur kompatibel mit der NEA Smart 1.0
und nicht kompatibel mit der NEA Smart 2.0 oder neuer

Das Modul dient zum Empfangen und Schalten von NEA Smart Daten.

## FUNKTIONSUMFANG

Setzen von Variablen via Skriptbefehl oder Visualisierung.
Auslesen der Daten der NEA Smart

## SOFTWARE-INSTALLATION

Über den Module Store das Modul REHAU Nea Smart installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'NEA Smart'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| IP-Adresse | IP-Adresse der NEA Smart |
| Intervall | Einstellbares Intervall in Sekunden in denen die Daten der NEA Smart abgefragt werden sollen. (Standard: 0) |
| Heizzone 1..12 | Aktiviert das Auslesen der jeweiligen Heizzone. (Die Heizkreise werden immer alle ausgelesen) |
| "Gerät auslesen" | Liest die Daten manuell aus. Muss auch einmalig zu Beginn gemacht werden. |

## STATUSVARIABLEN UND PROFILE

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

Die Variablen werden automatisch angelegt.

| NAME (BASIS) | TYP | BESCHREIBUNG |
|---|---|---|
| Basis Absenkdifferenztemperatur | Float | Dient als "Kategorie" in der sich alle überwachten Standorte, sowie der Zeitstempel und Längen-/Breitengrad befinden. Wird pro Gerät erstellt. |
| Basis Absenkeingang | Boolean | |
| Basis Automatische Zeitumstellung | Boolean | Ein/Aus der automatischen Zeitumstellung |
| Basis Betriebsart Heizen/ Kühlen (CO-Eingang) | Boolean | Steuert die Betriebsart (Ein = Kühlen, Aus = Heizen) |
| Basis Frostschutz | Boolean | Ein/Aus des Frostschutzes |
| Basis Frostschutz Temperatur | Integer | Temperatur des Frostschutzes |
| Basis HW Version | String | Hardware Version der Basis |
| Basis ID | String | ID der Basis |
| Basis Kühlen-Modus | Boolean | Ein/Aus des Kühlenmodus |
| Basis Rang im Systemverbund | Integer | Rand der Basis (0 = Standalone, 1 = Master, 2 = Slave) |
| Basis Smartstart ein/aus | Boolean | Ein/Aus des Smartstart |
| Basis SW ETH Version | String | Software Version Ethernet |
| Basis SW STM Version | String | Software Version STM |
| Basis Taupunktsensor | Boolean | Ein/Aus des Taupunktsensors |
| Basis Temperaturbegrenzer | Boolean | Ein/Aus des Temperaturbegrenzers |
| Basis Temperatureinheit | Integer | Einstellung ob Fahrenheit oder Celsius (0 = °C, 1 = °F) |
| Basis Urlaub Status | Integer | Urlaubstatus (0 = aus, 1 = geplant, 2 = aktiv) |
| Basis Urlaub Temperatur Heizen | Float | Temperatur während des Urlaubs |

| NAME (HEIZZONE 1..12) | TYP | BESCHREIBUNG |
|---|---|---|
| Heizzone 01..12 Ist Temperatur | Float | Aktuelle Ist Temperatur |
| Heizzone 01..12 Soll Temperatur | Float | Soll Temperatur |
| Heizzone 01..12 Betriebsmodus | Integer | Betriebsmodus der Heizzone (0 = Auto, 1 = Tag, 2 = Nacht) |
| Heizzone 01..12 Ist Temperatur Ext. Sensor | Float | Aktuelle Ist Temperatur des externen Sensors |
| Heizzone 01..12 Anwesenheit | Boolean | Ein/Aus der Anwesenheit |
| Heizzone 01..12 Heizsystem | Integer | Das verwendete Heizsystem (0 = FBH-Standard, 1 = FHB-Niedrigenergie, 2 = Radiator, 3 = Konvektor passiv, 4 = Konvektor aktiv) |
| Heizzone 01..12 Korrektur Ist-Werterfassung | Float | Korrektur des Ist-Werts (-2.0 .. +2.0) |
| Heizzone 01..12 Name | String | Name der Heizzone |
| Heizzone 01..12 Partyschaltung (Stunden) | Integer | Wieviele Stunden die Partyschaltung laufen soll |
| Heizzone 01..12 Partyschaltung Restzeit (min) | Integer | Wielange die Partyschaltung noch aktiv ist |
| Heizzone 01..12 Programm Wochenende | Integer | Welches Zeitprogramm am Wochenende aktiv ist |
| Heizzone 01..12 Programm wochentags | Integer | Welches Zeitprogramm unter der Woche aktiv ist |
| Heizzone 01..12 Soll Temperatur Max | Float | Einstellbereich Soll-Temperatur (maximal) |
| Heizzone 01..12 Soll Temperatur Min | Float | Einstellbereich Soll-Temperatur (minimal) |
| Heizzone 01..12 Status | Boolean | Statusanzeige (0 = OK, 1 = Error) |

| NAME (HEIZKREIS 1..12) | TYP | BESCHREIBUNG |
|---|---|---|
| Heizkreis 1..12 Aktiv | Boolean | Ob der Heizkreis aktiv ist (0 = Nein, 1 = Ja) |
| Heizkreis 1..12 Aktor | Boolean | Ein/Aus des Stellantriebs |
| Heizkreis 1..12 Aktor Prozent | Integer | Prozentuale Aktivität des Aktors (Nur bei SW STM Version 2.02 oder höher) |
| Heizkreis 1..12 Status | Integer | Status (0 = Aus, 1 = An, 2 = Fehler) |
| Heizkreis 1..12 zugewiesene Heizzone | Integer | Welche Heizzone mit dem Heizkreis verbunden ist |

**PROFILE:**

| NAME | TYP |
|---|---|
| NEAS.AntifreezeTemp | Integer |
| NEAS.TemperatureUnit | Integer |
| NEAS.SummerWinter | Boolean |
| NEAS.ChangeOver | Boolean |
| NEAS.Mode | Integer |
| NEAS.EcoDiff | Float |
| NEAS.VacationState | Integer |
| NEAS.PumpTime | Integer |
| NEAS.RelaisTime | Integer |
| NEAS.EmergencyTime | Integer |
| NEAS.PWMCycle | Integer |
| NEAS.PWMPercent | Integer |
| NEAS.HeatAreaMode | Integer |
| NEAS.HeatAreaProgram | Integer |
| NEAS.HeatAreaParty | Integer |
| NEAS.HeatAreaPartyRemainingTime | Integer |
| NEAS.HeatAreaState | Boolean |
| NEAS.HeatAreaRPMMotor | Integer |
| NEAS.HeatingSystem | Integer |
| NEAS.HeatAreaTActualTemp | Float |
| NEAS.HeatAreaTTarget | Float |
| NEAS.HeatAreaTHeatCool | Float |
| NEAS.HeatAreaOffset | Float |
| NEAS.HeatAreaBlockHC | Integer |
| NEAS.HeatAreaHeatCTRLState | Integer |
| NEAS.HeatAreaNr | Integer |
| NEAS.HeatCtrlActorPercent | Integer |

## VISUALISIERUNG

Über die Visualisierung oder in den mobilen Apps werden Werte angezeigt und können wenn möglich geändert werden.
Wir empfehlen aufgrund der Menge der Variablen die Anzeige der gesamten NEA Smart in der Verwaltungskonsole zu deaktivieren und alle gewünschten Variablen via Links in einer eigenen Kategorie anzeigen zu lassen.

# NEAS_RequestStatus

Benötigt: IP-Symcon >= 4.0

```
boolean NEAS_RequestStatus (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Ruft die in der NEA Smart mit der InstanzID **InstanzID** gespeicherten Werte auf und setzt die dazugehörigen Variablen.

## BEISPIEL

```
NEAS_RequestStatus(12345);
```

# NEAS_WriteValue

Benötigt: IP-Symcon >= 4.0

```
boolean NEAS_WriteValue (int $InstanzID, string $Ident, variant
$Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| Ident | Ident der Variable |
| Wert | Wert der Variable |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schreibt den Wert **Wert** in die Variable mit dem Ident **Ident** in der NEA Smart mit der InstanzID **InstanzID**.

## BEISPIEL

```
NEAS_WriteValue(12345 , "HEATAREA1_T_TARGET", 23.3);
```

# OCPP

Über das OCPP Protokoll können Wallboxen bequem über Symcon überwacht und geschaltet werden.

## FUNKTIONSUMFANG

Anzeige des Verbrauchs einzelner Ladepunkte
Anzeige, ob gerade eine Transaktion läuft bei Ladepunkten

## SOFTWAREINSTALLATION

Über den Module Store das 'OCPP'-Modul installieren.
Unter 'Instanz hinzufügen' kann das 'OCPP Configurator'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

Auf der Konfigurationsseite des Splitters können die Informationen zum Einrichten eines Ladepunktes entnommen werden.
Je nach Wallbox müssen die Parameter einzeln oder als komplette URL angegeben werden.
Ist die Verbindung hergestellt werden die Ladepunkte im Konfigurator angezeigt.

Hinweise

Bei der Pulsar Plus müssen alle Zeitpläne vor der Aktivierung von OCPP entfernt werden.
Die Ansteuerung über OCPP kann die konfigurierten Zeitpläne nicht überschreiben, da diese immer Vorrang haben.

## STATUSVARIABLEN: LADEPUNKT

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Hersteller | string | Name des Herstellers |
| Verbrauchswert | float | Anzeige des Verbrauchs des Ladepunktes |
| Model | string | Modell des Ladepunktes |
| Seriennummer | string | Seriennummer des Ladepunktes |
| Transaktion ausführen | boolean | Anzeige, ob gerade ein Ladepunkt genutzt wird |

# OCPP_RemoteStartTransaction

```
boolean OCPP_RemoteStartTransaction (int $InstanzID, int
$ConnectorID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| ConnectorID | ID des Connectors |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Wenn der Status auf 'Preparing' oder 'Finishing' steht, kann durch diesen Befehl das Laden freigeschaltet werden.

## BEISPIEL

```
OCPP_RemoteStartTransaction(12345, 1);
```

# OCPP_RemoteStopCurrentTransaction

```
boolean OCPP_RemoteStopCurrentTransaction (int $InstanzID, int
$ConnectorID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| ConnectorID | ID des Connectors |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Es wird dabei die aktuelle TransactionId genommen, welche in der passenden Variable zur ConnectorId steht!

## BEISPIEL

```
OCPP_RemoteStopCurrentTransaction(12345, 1);
```

# OCPP_RemoteStopTransaction

```
boolean OCPP_RemoteStopTransaction (int $InstanzID, int
$TransactionID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| TransactionID | ID der Transaction |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Wenn der Zustand auf 'Charging' steht, kann durch diesen Befehl das Laden beendet werden. Dazu muss jedoch nicht die ConnectorId, sondern die TransactionId, welche in der passenden Variable steht, übergeben werden!

## BEISPIEL

```
OCPP_RemoteStopTransaction(12345, 4479);
```

# OPC UA

Benötigt: IP-Symcon >= 6.4



Das OPC UA Modul ist eine kostenpflichtige Erweiterung, welche zu jeder bestehenden IP-Symcon Lizenz im Shop erworben werden kann. Bei jeder Symcon Lizenz können bis zu 3 OPC UA Knoten kostenlos erstellt werden.

Die OPC UA (Client) Anbindung erlaubt das komfortable Einrichten von OPC UA Geräten innerhalb von IP-Symcon. Dabei kann über einen OPC UA Konfigurator der OPC UA Address Space eines Geräts ermittelt und traversiert werden. Es können beliebige OPC UA Adressen (Knoten) erstellt werden. Die Knoten zeigen dabei den aktuellen Wert an, welcher über auch automatisch überwacht und aktualisiert wird, sofern das Gerät dies unterstützt. Alternativ kann der Wert auch zyklisch abgefragt werden. Der Wert kann zusätzlich per Visualisierung, Skript und Ablaufplan verändert werden.

## VIDEO-TUTORIAL ZUR EINRICHTUNG

Folgt

### UNTERSTÜTZTES PROTOKOLL

OPC UA (Binary Encoding)

### UNTERSTÜTZTER TRANSPORT

TCP/IP

### UNTERSTÜTZTE SECURITY MODES

None
Sign
SignAndEncrypt

### UNTERSTÜTZTE AUTHENTIFIZIERUNGSMETHODEN

Anonymous
Username/Password

### UNTERSTÜTZTE SECURITYPOLICIES

None
Basic256
Basic256Sha256

## UNTERSTÜTZTE DATENTYPEN

Boolean, SByte, Byte, Int16, UInt16, Int32, UIn32, Int64, UIn64, Float, Double, String, DateTime, GUID, ByteString, XMLElement, NodeId, StatusCode, QualifiedName, LocalizedText, Variant
Array von den jeweiligen Typen (werden als eigene Variablen dargestellt)

# SageGlass (BACnet)

Benötigt: IP-Symcon >= 6.0
Dieses Modul erlaubt die Einbindung SageGlass SIM II Controller über BACnet/IP.

## FUNKTIONSUMFANG

Integriert die Ansteuerung von SageGlass Zonen und zeigt den aktuellen Zustand inkl. der übermittelten Lux-Werte an.

## VORAUSSETZUNGEN

IP-Symcon Enterprise ab Version 6.0 mit BACnet Erweiterung

## SOFTWARE-INSTALLATION

Über den Module Store das 'SageGlass'-Modul installieren.

## EINRICHTEN IN IP-SYMCON

Es wird empfohlen alle BACnet Objekte auf dem SageGlass SIM II Controller automatisch erstellen zu lassen, wodurch die einzelnen Datenpunkte in IP-Symcon verfügbar gemacht werden. Im zweiten Schritt kann nachfolgendes Skript verwendet werden, um die SageGlass Instanzen für die Zonen zu erstellen, welche die Informationen schöner aufbereiten. Es ist auch möglich die Verknüpfungen manuell durchzuführen, indem alle Eigenschaften auf die korrekten BACnet Objekte verknüpft werden. Die Namen der Eigenschaften entsprechen dabei den Beschreibungen der BACnet Objekte und sollten auf die "Preset Value" Variable verknüpft werden.

```
<?php

// Not every zone has a sensor - Normally each facade will have a
vertical/horizontal sensor
// Fill these arrays with the sensor -> zone information
// e.g. Sensor 1 (Vertical) is used for Zone 10, Zone 20 and Zone 25
$sensorVertical = [
1 => [10, 20, 25]
];
$sensorHorizontal = [];

$zones = 250;
for ($i = 1; $i <= $zones; $i++) {
$variableTintID = searchBACnet(2 /* Analoge Value */, $i + 1 /* Zone
1 = 2 */);
$automodeStateID = searchBACnet(2 /* Analoge Value */, $i + 2001 /*
Zone 1 = 2002 */);
$luxLevelSetPointID = searchBACnet(2 /* Analoge Value */, $i + 4001
/* Zone 1 = 4002 */);
$statusID = searchBACnet(0 /* Analoge Input */, $i + 4000 /* Zone 1
```

```
= 4001 */);

// Search for the our sensor the vertical sensor matrix
$verticalSensorID = searchSensorID($i, $sensorVertical);
if ($verticalSensorID) {
$verticalSensorID = searchBACnet(0 /* Analoge Input */,
$verticalSensorID + 2000 /* Sensor 1 = 2001 */);
}

// Search for the our sensor the vertical sensor matrix
$horizontalSensorID = searchSensorID($i, $sensorHorizontal);
if ($horizontalSensorID) {
$horizontalSensorID = searchBACnet(0 /* Analoge Input */,
$horizontalSensorID + 2000 /* Sensor 1 = 2001 */);
}

if (!@IPS_GetObjectIDByIdent("SageGlassZone" . $i)) {
if ($variableTintID && $automodeStateID && $luxLevelSetPointID &&
$verticalSensorID && $horizontalSensorID && $statusID) {
echo "Creating Zone " . $i . "..." . PHP_EOL;
$id = IPS_CreateInstance("{67CEA419-A625-703E-2BE6-BF51B3C913B9}");
IPS_SetName($id, "Zone " . $i);
IPS_SetIdent($id, "SageGlassZone" . $i);
IPS_SetProperty($id, "VariableTint", $variableTintID);
IPS_SetProperty($id, "AutomodeState", $automodeStateID);
IPS_SetProperty($id, "LuxLevelSetPoint", $luxLevelSetPointID);
IPS_SetProperty($id, "Status", $statusID);
IPS_SetProperty($id, "SensorVertical", $verticalSensorID);
IPS_SetProperty($id, "SensorHorizontal", $horizontalSensorID);
IPS_ApplyChanges($id);
}
}
}

function searchSensorID($zone, $sensorMatrix) {
foreach($sensorMatrix as $sensor => $matrix) {
foreach($matrix as $sensorZone) {
if ($zone == $sensorZone) {
return $sensor;
}
}
}
return 0;
}

function searchBACnet($objectType, $instanceNumber) {
$ids = IPS_GetInstanceListByModuleID("{CD5D5D10-3743-DA88-
F16C-8B65CF4103F9}");
foreach ($ids as $id) {
if ((IPS_GetProperty($id, "ObjectType") == $objectType) &&
(IPS_GetProperty($id, "InstanceNumber") == $instanceNumber)) {
return IPS_GetObjectIDByIdent("PresentValue", $id);
}
}
return 0;
}
```

# Shutter Control

Benötigt: IP-Symcon >= 5.0
Das "Shutter Control" bietet die Möglichkeit einen Rollladen/Markise auf eine bestimmte prozentuale Position zu fahren.
Desweiteren wird die Position aktualisiert, wenn der Shutter via Befehl oder Manuell angesteuert wird.
Im Gegensatz zum Shutter Control (legacy) kann auch während der Fahrt die Position angepasst werden.

## VORAUSSETZUNGEN

Das IP-Symcon Shutter-Modul benötigt eine Variable oder eine Instanz mit einer Statusvariable, welche das Profil ~ShutterMoveStop oder ~ShutterMoveStep nutzt.
Auch müssen die Fahrzeiten und Anfahrverzögerung in das Modul eingetragen werden.
Dies funktioniert am einfachsten mit einer Stoppuhr.



Standardmäßig unterstützen dS Shutter, EnOcean Shutter, KNX Shutter, LCN Shutter, xComfort Shutter und Z-Wave Shutter diese Profile. Das heißt bei deren Erstellung ist eine Variable mit dem benötigten Profil vorhanden.

## EINSTELLUNGEN

| OPTION | POSITION | BESCHREIBUNG |
|---|---|---|
| Typ | -------- | Rollladen oder Markise |
| Verzögerung | -------- | Anfahrverzögerung des Motors nach dem Schalten |
| Ziel | -------- | Aktorvariable, welcher angesteuert werden soll |
| Runterfahren (Mitte) | 50% | Gemessene Zeit von Offen zu Mitte. |
| Runterfahren (Unten) | 99% | (nur Rollladen) Gemessene Zeit von Offen zu Unten. (Stäbe noch offen) |
| Runterfahren (Geschlossen) | 100% | Gemessene Zeit von Offen bis Endposition erreicht. Lamellen geschlossen. |
| Hochfahren (Unten) | 99% | (nur Rollladen) Gemessene Zeit von Geschlossen bis Stäbe geöffnet |
| Hochfahren (Mitte) | 50% | Gemessene Zeit von Geschlossen bis Mitte erreicht. |
| Hochfahren (Offen) | 0% | Gemessene Zeit von Geschlossen bis Endposition erreicht. Komplett geöffnet |

## EIGENES GERÄT HINZUFÜGEN

Falls das Gerät nicht zu einem der untersützten Varianten gehört, kann selber eine Integervariable angelegt werden.
Diese muss das Profil ~ShutterMoveStop oder ~ShutterMoveStep zugewiesen haben.
Desweiteren muss folgendes Aktionsskript hinterlegt werden.

```
// Vorlage für Aktionsskript
switch($_IPS['VALUE']) {

case 0:
// Hier der Befehl für Öffnen
break;

case 1: // Nur bei ~ShutterMoveStep
// Hier der Befehl für Schritt-Öffnen
break;

case 2:
// Hier der Befehl für Stoppen
break;

case 3: // Nur bei ~ShutterMoveStep
// Hier der Befehl für Schritt Schließen
break;

case 4:
// Hier der Befehl für Schließen
break;
}

SetValue($_IPS['VARIABLE'], $_IPS['VALUE']);
```

# SC_Move

```
boolean SC_Move (int $InstanzID, int $Position)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Shutter Control Instanz |
| Position | 0%-100%, 99% = Auf Spalt fahren |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Bewegt den Rollladen für das Gerät mit der ID **InstanzID** auf die **Position**.



Da einige Funk-Systeme unidirektonal sind, kann nicht festgestellt werden, ob ein Befehl erfolgreich war. Dadurch kann die genaue Positionierung nicht garantiert werden.

## BEISPIEL

```
SC_Move(12345, 50); //Auf 50% fahren
```

# SC_MoveDown

```
boolean SC_MoveDown (int $InstanzID, int $Dauer)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Shutter Control Instanz |
| Dauer | 0 = Endlagenschalter, >1 = Dauer in ms |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Bewegt den Rollladen für das Gerät mit der ID **InstanzID** für **Dauer** Millisekunden nach unten.



Die Statusvariable wird unabhängig von der gefahrenen Zeit auf 100 zurückgesetzt.

## BEISPIEL

```
SC_MoveDown(12345, 0); //Nach unten fahren
```

# SC_MoveUp

```
boolean SC_MoveUp (int $InstanzID, int $Dauer)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Shutter Control Instanz |
| Dauer | 0 = Endlagenschalter, >1 = Dauer in ms |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Bewegt den Rollladen für das Gerät mit der ID **InstanzID** für **Dauer** Millisekunden nach oben.



Die Statusvariable wird unabhängig von der gefahrenen Zeit auf 0 zurückgesetzt.

## BEISPIEL

```
SC_MoveUp(12345, 0); //Nach oben fahren
```

# SC_Stop

```
boolean SC_Stop (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Shutter Control Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Stoppt einen Bewegungsvorgang für das Gerät mit der ID InstanzID.
Die Statusvariable wird auf -1 zurückgesetzt.

## BEISPIEL

```
SC_Stop(12345); //Anhalten
```

# Siemens OZW

Siemens OZW ist ein Web-Server, welcher via Web eine Fernbedienung und Fernüberwachung von KNX Geräten ermöglicht. IP-Symcon wird via LAN(IP) mit Siemens OZW verbunden.



Folgende Geräte werden von IP-Symcon unterstützt:
Unterstützte Geräte

## EINLEITUNG

Diese Anleitung soll helfen, einen **Siemens OZW 672 / Siemens OZW 772 Web-Server** möglichst schnell und unkompliziert mit dem Computer zu verbinden. Dies soll die Voraussetzung schaffen, mit IP-Symcon verschiedenste Anwendungsbereiche abzudecken und diese zu visualisieren. Der Web-Server OZW ermöglicht die Fernbedienung und Fernüberwachung von Anlagen über das Web. Er ist in vier Ausführungen erhältlich: für den Anschluss von eins, vier, 16 oder 250 KNX Geräten der Sortimente Synco 700, Synco RXB/RXL, Raumthermostaten RDG/RDF/RDU und der Synco living Wohnungszentralen QAX9.

## VIDEO-TUTORIAL ZUR EINRICHTUNG

## ANSCHLUSS

Es sollten alle KNX-Geräte korrekt angeschlossen sein bevor das Gerät durch ein Netzwerkkabel mit dem Netzwerk und schließen Sie es an die Spannungsversorgung an. Auch eine Verbindung mit dem PC per USB ist möglich, sofern eine Verbindung zum Internet gegeben ist, damit der RNDIS-Treiber beim Verbinden via USB automatisch installiert werden kann (sofern der Microsoft online update Dienst freigeschaltet ist). Ein manuelles Installieren des Treibers ist ebenfalls möglich.

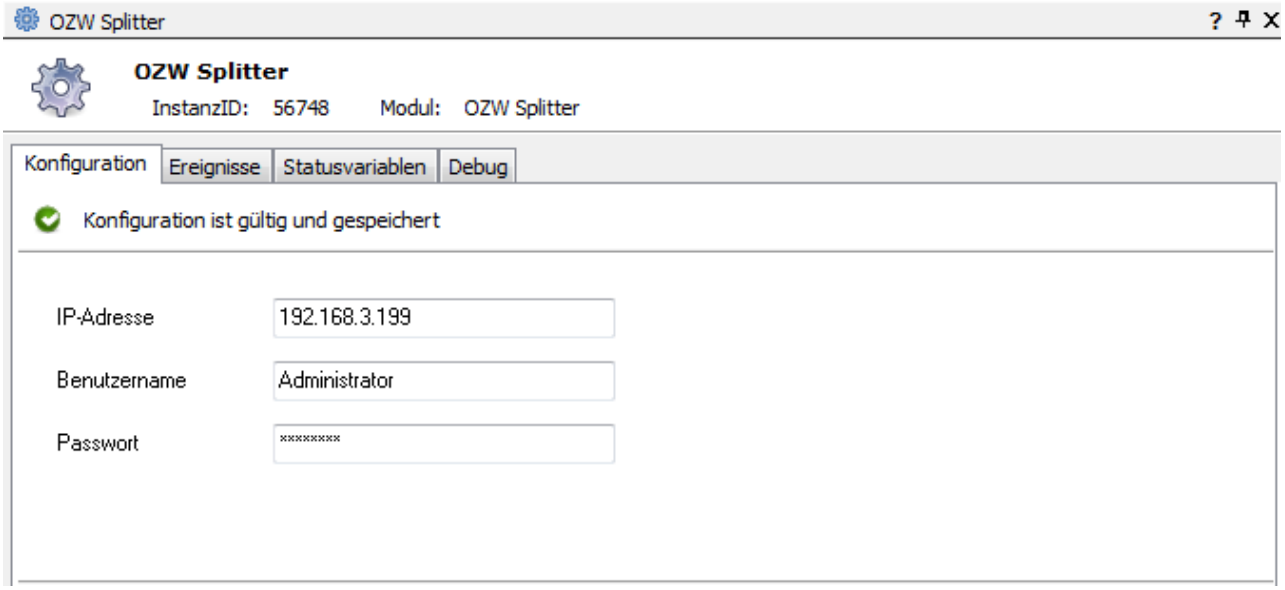
## INSTALLATION

Wenn der OZW korrekt angeschlossen ist und ein DHCP-Server verwendet wird, ist jener nun mit dem lokalen Netzwerk verbunden. *(Der DHCP-Server übernimmt automatisch die Adressvergabe für die Netzwerkteilnehmer.)*
Unter den Netzwerkgeräten erscheint nun der OZW erscheint in der Kategorie "Andere Geräte".
Durch anklicken des "OZW..." öffnet sich ein Fenster, in dessen unteren Bereich die IP-Adresse des OZWs angezeigt wird. Durch Kopieren dieser Adresse in einen beliebigen Browser öffnet sich die Siemens Webpage.
Beim ersten Login gilt: **"Username" = "Administrator" und "Password" = "Password".**
Das Passwort sollte nach dem ersten Login geändert werden.
Groß- und Kleinschreibung muss beachtet werden. Nach dem ersten Login ist die Web-Server-Sprache "Englisch", die sich aber anpassen lässt. Auf der linken Seite sind alle angeschlossenen KNX-Geräte.
Sofern kein DHCP-Server benutzet wird, ist die IP-Adresse des OZWs im

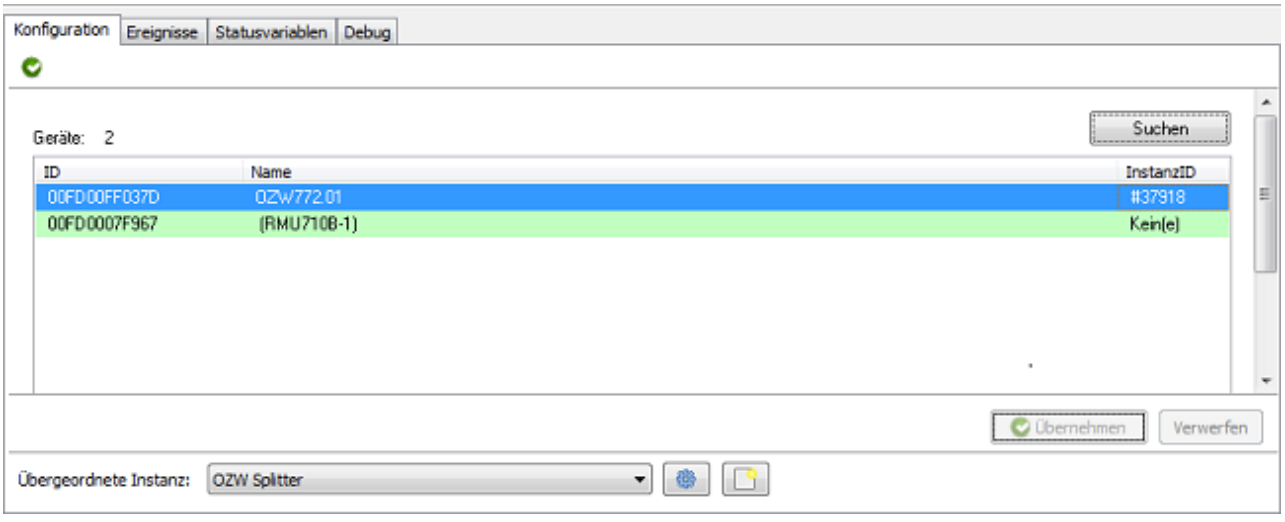
Auslieferungszustand 192.168.2.10 (USB: 192.168.250.1 (nicht veränderbar)).

## EINBINDUNG IN IP-SYMCON

In der Verwaltungskonsole kann über "Konfigurator erstellen" ein "OZW Konfigurator" erstellt werden. Dazu muss zweimal auf „Weiter“, einmal auf „OK“ geklickt werden. Zur folgenden Konfigurationsseite gelangt man via Zahnrad (unten links).

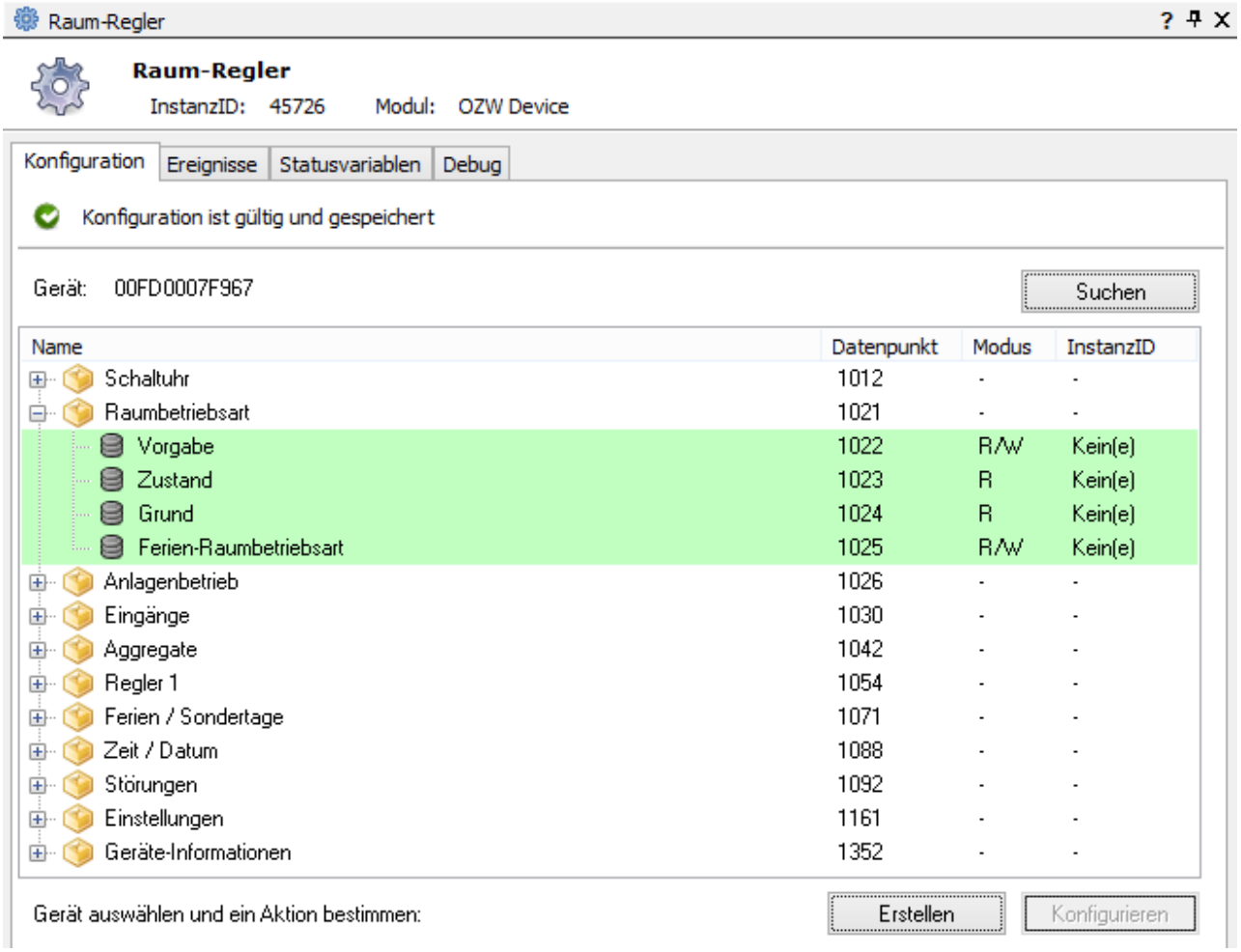



Es öffnet sich die Konfiguration des "OZW Splitters". Hier sind bereits die Standardwerte von Siemens voreingestellt. Gegebenfalls können hier veränderte Passwörter oder eine andere IP eingestellt werden. Mit dem Klick auf "Übernehmen" werden die Änderungen erst gespeichert. Im OZW Konfigurator kann jetzt auf "Suchen" geklickt werden. Anschließend erscheinen der OZW und alle an diesen angeschlossenen KNX-Geräte.




Bei der ID auf der linken Seite handelt es sich um die Identifikationsnummer, die sich auch auf der Rückseite der einzelnen Geräte wiederfinden lässt. Im Falle einer großen Anzahl von installierten KNX-Geräten ist damit eine problemlose Zuordnung und Konfiguration gewährleistet.
Bei der ID auf der rechten Seite handelt es sich um die von IP-Symcon zugeordnete InstanzID. Diese ist einmalig und unveränderbar.
Jedes Gerät muss nacheinander ausgewählt und durch "Erstellen" in IP-Symcon

eingerichtet werden.
Diese Geräte stehen nun innerhalb von IP-Symcon zur Verfügung und sind bedienbar. Nach einem Klick auf ein ausgewähltes KNX-Gerät und dann einen Klick auf "Konfigurieren" kann darüber hinaus das Gerät genau eingestellt werden. Auch kann eingestellt werden, was in der Visualisierung angezeigt werden soll.




Dafür muss auf das "+" geklickt und der jeweilige Datenpunkt ausgewählt werden, der in der Visualisierung angezeigt werden soll. Durch ein Klicken auf "Erstellen", wird dieser in IP-Symcon eingerichtet. Die Ansicht kann nun geschlossen werden.

## ABFRAGEINTERVALLE KONFIGURIEREN

Sobald die einzelnen gewünschten Datenpunkte ausgewählt wurden, können diese über "Konfigurieren" geöffnet und ein Abfrageintervall eingestellt werden. Dieses ist notwendig, da IP-Symcon die Werte aktiv vom OZW abfragen muss.



Eine Abfrage dauert ca. 800ms. Es können also z.B. 13 Datenpunkte alle 10 Sekunden abgefragt werden. Wenn die Intervalle zu kurz sind, werden die Abfragen nacheinander, verzögert abgearbeitet.

# Geräteliste

## UNTERSTÜTZTE GATEWAYS/KOMPONENTEN

Es werden alle KNX Geräte, die mit dem Siemens OZW 672 / Siemens OZW 772 Web-Server kompatibel sind, unterstützt.

# OZW_RequestStatus

```
boolean OZW_RequestStatus (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des abzufragenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Fragt den Status des Geräts mit InstanzID **InstanzID** ab.
Setzt den Wert des Datenpunktes mit der ID **InstanzID** auf angegebenen Wert.

## BEISPIEL

```
OZW_RequestStatus(12345);
```

# OZW_WriteDataPoint

```
boolean OZW_WriteDataPoint (int $InstanzID, variant $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu verändernden Geräts |
| Wert | z.B. 23.5 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den Wert des Datenpunktes mit der ID **InstanzID** auf angegebenen Wert.



Sofern das angezeigte Profil über Assoziationen verfügt, muss der Integer-Wert der gewünschten Assoziation angegeben werden (z.B. 0) anstatt des String-Wertes (z.B. Aus).

## BEISPIEL

```
OZW_WriteDataPoint(12345, 1);
```

# SNMP

Benötigt: IP-Symcon >= 6.0
Das SMNP Modul liest alle verfügbaren OIDs eines Geräts aus und stellt beliebige davon als Variable in IP-Symcon dar. Diese werden zyklisch Aktualisierung und können bei Bedarf auch beschrieben werden. Unterstützung für OIDLib Dateien gibt den OIDs mehr Informationen/Kontext, um die Einrichtung zu vereinfachen.

## FUNKTIONSUMFANG

Automatischer Walk aller OIDs einges Geräts
Erstellung einzelner OIDs als Variable
Schreiben von Werten auf die OIDs
Weitere Informationen sofern OIDLib Dateien vorhanden
SNMPv1/SNMPv2/SNMPv3 inkl. Authentifizerung und Verschlüsselung werden unterstützt

## SOFTWARE-INSTALLATION

Über den Module Store das 'SMNP'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'SMNP'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE

| NAME | BESCHREIBUNG |
|---|---|
| Host | Adresse des Host |
| Version | Auswahl zwischen den Versionen SNMPv1, SNMPv2, SNMPv3 |
| Starte bei | OID bei dem der Walk starten soll |
| Benutzer | Benutzername |
| Community | Für SNMPv1, SNMPv2 nötig |
| Authentifizierung aktivieren | Aktiviert die Möglichkeit der Authentifizierung |
| Authentifizierungspasswort | Passwort für die Authentifizierung |
| Autentifizierungsart | Auswahl der Autentifizierungsart: MD5, SHA1, SHA224, SHA256, SHA384, SHA512 |
| Verschlüsselung aktivieren | Aktiviert die Verschlüsselung |
| Verschlüsselungspasswort | Passwort für die Verschlüsselung |
| Verschlüsselungsart | Auswahl für die Verschlüsselungsart: DES, AES128, 3DES, AES192, AES256, AES192blue, AES256blue |
| Zeige nur bekannte OIDs aus den OIDLibs an | Bei Aktivierung werdern nur OIDs angezeigt, welche in den OIDLibs vorhanden sind |
| OIDLibs | Liste von Dateien, welche die OIDs mit Beschreibung und Namen versieht. ÜBer folgendes Tool können MIB Dateien in OIDLib Dateien konvertiert werden: MIB-Dateien zu OIDLibs |
| Aktualisierungsintervall | Intervall in Sekunden, in welchem zeitlichen Abstand die Werte aktualisiert werden |

| NAME | BESCHREIBUNG |
|---|---|
| Walk starten / Walk stoppen | Button um den Walk zu starten und zu stoppen |
| OID | Anzeige der OID |
| Name | Name, welcher durch die OIDLib gegeben wurde |
| Beschreibung | Beschreibung , welche durch die OIDLib gegeben wurde |
| Wert | Wert der OID, welcher zum Zeitpunkt des Walks |
| Aktiv? | Erstellt eine Variable für die gewählte OID |
| Schreibbar? | Aktiviert das Schreiben für die gewählte OID |

# SNMP_UpdateValues

Benötigt: IP-Symcon >= 6.0

```
boolean SNMP_UpdateValues (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Updated die Variablenwerte, welche unterhalb der Instanz liegen.

## BEISPIEL

```
SNMP_UpdateValues(12345);
```

# Snom

Benötigt: IP-Symcon >= 7.0



Das Snom Modul wurde in Kooperation zwischen Symcon und Snom erstellt. Das Modul ist als Open Source verfügbar und die aktuelle Dokumentation steht auf GitHub bereit.

# SPS: Siemens, Vipa, Logo

## BESCHREIBUNG

Speicherprogrammierbare Steuerungen (SPS, engl. Programmable Logic Controller, PLC) sind eigenständige elektronische Baugruppen zur Steuerung von Maschinen und Anlagen. Diese arbeiten unabhängig von einem PC. Die Befehle zum Steuern und Regeln sind direkt in der SPS abgelegt.



Folgende Geräte werden von IP-Symcon unterstützt:
Unterstützte Geräte



Durch die Tatsache, dass PHP Integer als größten ganzzahligen Wert anbietet, können DWORDs (unsigend ints) in IP-Symcon zwar gelesen werden, sie werden jedoch auf Integer abgebildet, was bedeutet, nur die unteren 31Bit können ausgewertet werden.

## INHALT


Einbindung in IP-Symcon
S7 Verbindungskonfiguration
Logo Verbindung
Logo Konfigurator
Beispiel mit TIA Portal
Beispiel mit Logo Soft Comfort
Datentypen
Logo VM-Adressen


## EINBINDUNG IN IP-SYMCON

Die Siemens PLCs können nur via Ethernet Anschluss eingebunden werden.



Die Unterstützung von MPI und ProfiBus via LIBNODAVE wurde ab Version 4.0 eingestellt



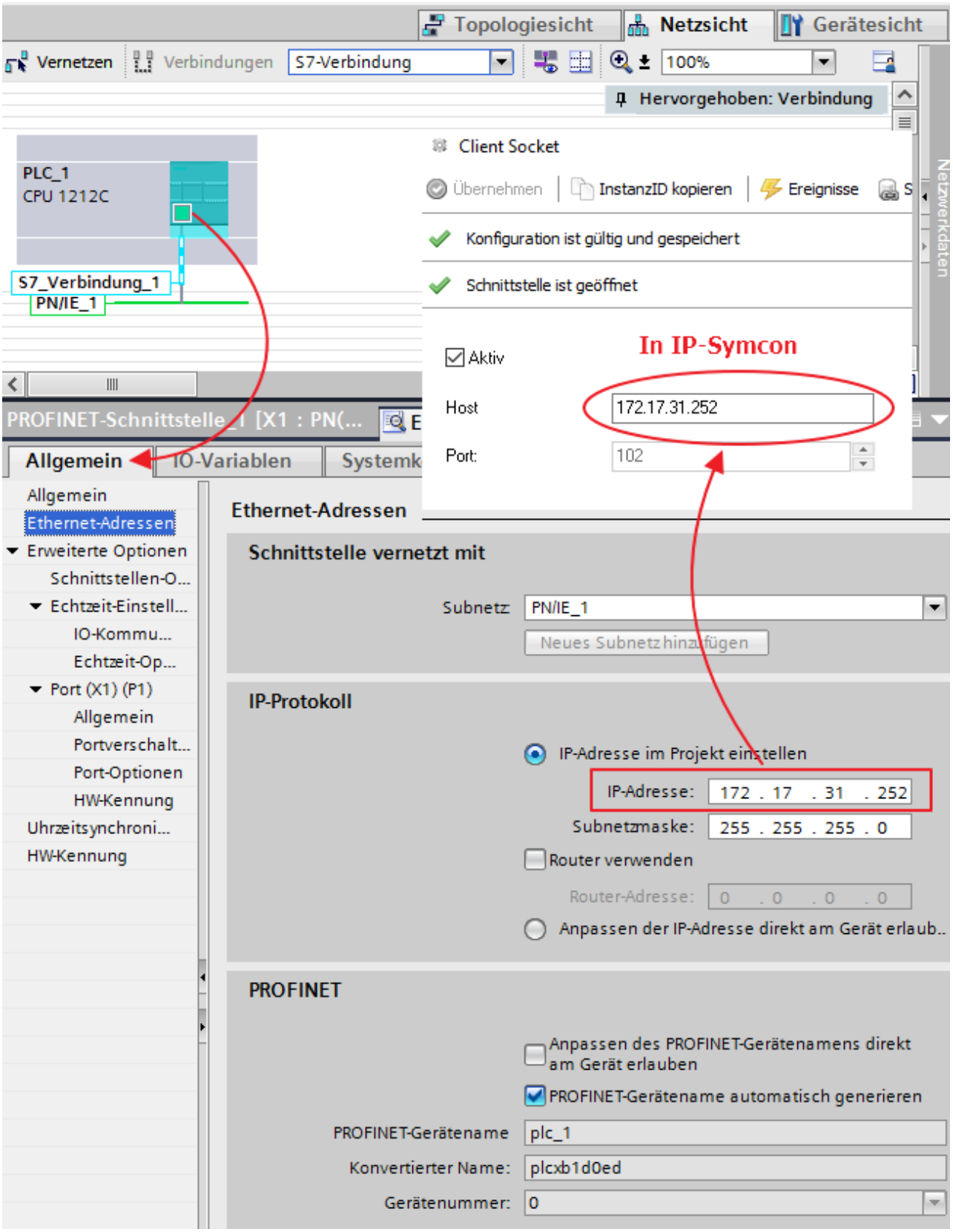
Bei neueren S7 (1200/1500) muss das "Optimized block access" Attribut deaktiviert werden. Weitere Informationen befinden sich HIER

### S7 VERBINDUNGSKONFIGURATION

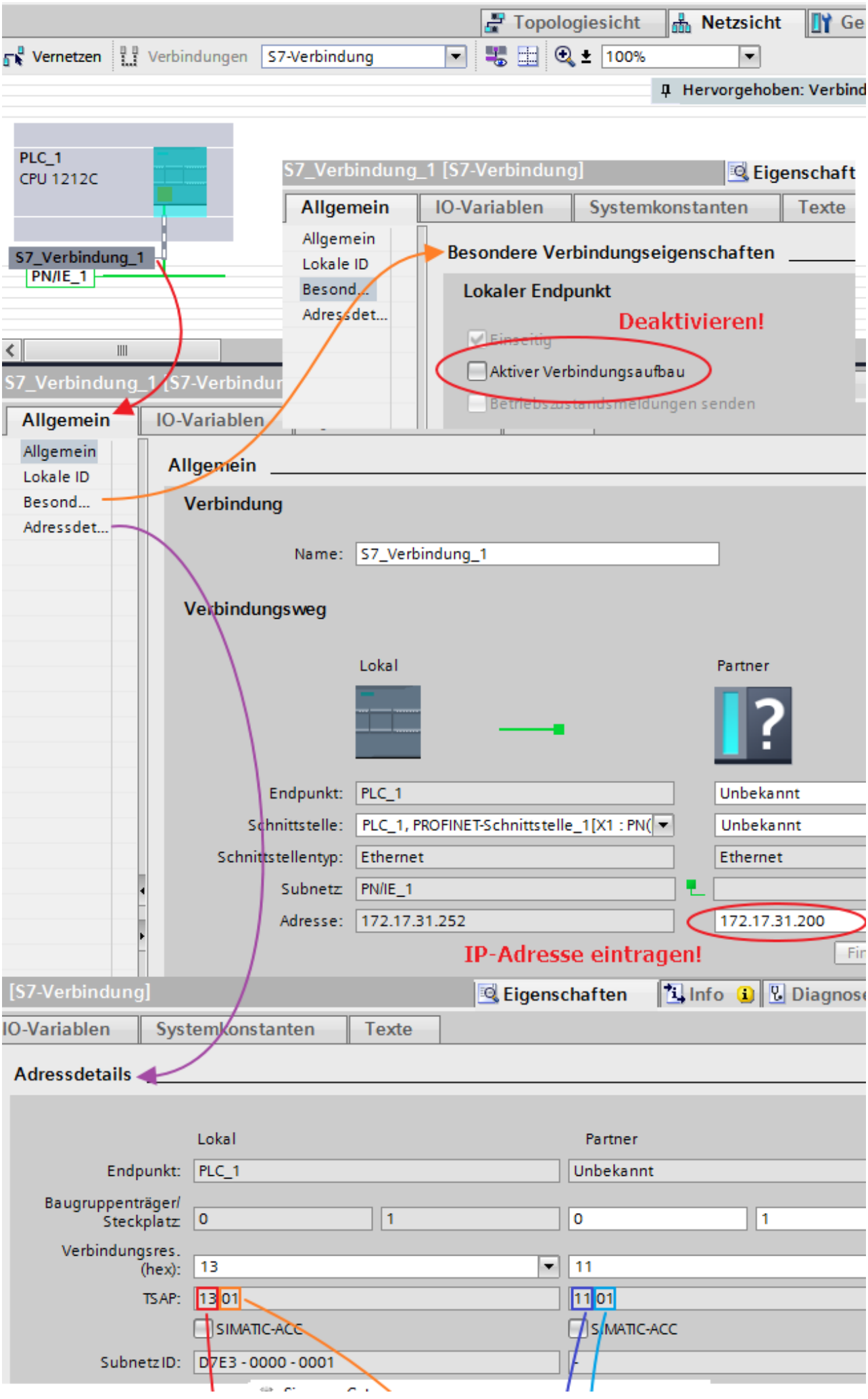
Die Verbindungskonfiguration erfolgt via "TIA Portal"-Software (Totally Integrated Automation) von Siemens.
Folgende Einstellungen müssen gemacht werden.

- IP-Adresse in PROFINET Schnittstelle eintragen (Siehe Screenshot)
- PN/IE Subnetz anbinden
- S7-Verbindung an PROFINET einfügen und folgende Punkte konfigurieren (Siehe Screenshot)
- Verbindungsdaten in IP-Symcon eintragen (Siehe Screenshot)

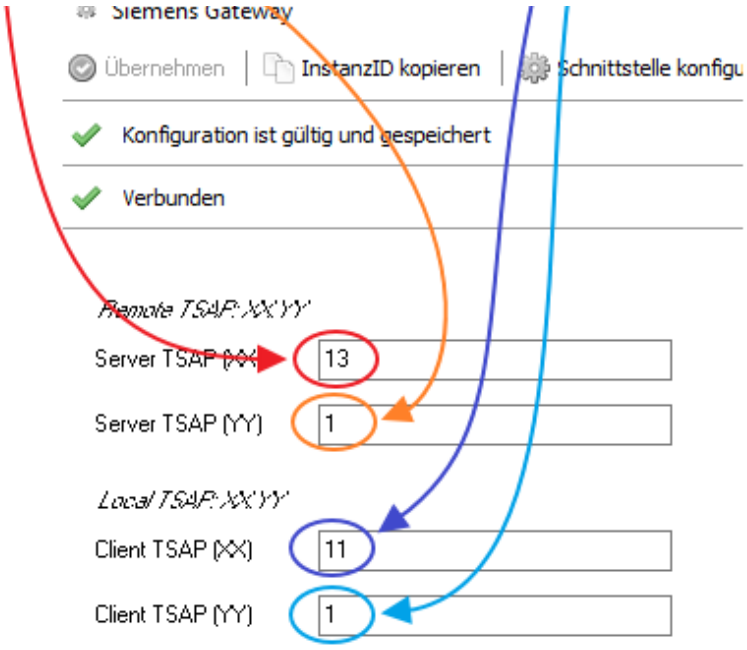
PROFINET Schnittstelle




S7 Verbindung

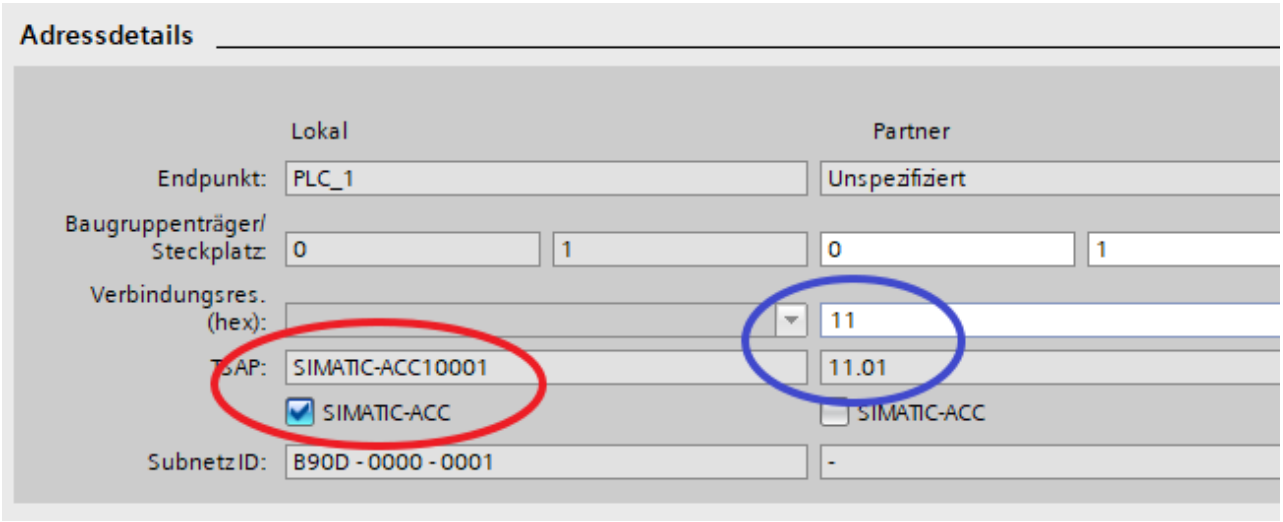

Topologiesicht
Netzsicht
Vernetzen
Verbindungen
S7-Verbindung
100%
Hervorgehoben: Verbind
PLC_1
CPU 1212C
S7_Verbindung_1
PN/IE_1
S7_Verbindung_1 [S7-Verbindung]
Eigenschaft
Allgemein
IO-Variablen
Systemkonstanten
Texte
Allgemein
Lokale ID
Besond...
Adressdet...
Besondere Verbindungseigenschaften
Lokaler Endpunkt
Deaktivieren!
Einseitig
Aktiver Verbindungsaufbau
Betriebszustandsmeldungen senden
Verbindung
Name:
S7_Verbindung_1
Verbindungsweg
Lokal
Partner
Endpunkt:
PLC_1
Unbekannt
Schnittstelle:
PLC_1, PROFINET-Schnittstelle_1 [X1 : PN(
Unbekannt
Schnittstellentyp:
Ethernet
Ethernet
Subnetz:
PN/IE_1
Adresse:
172.17.31.252
172.17.31.200
IP-Adresse eintragen!
[S7-Verbindung]
Eigenschaften
Info
Diagnose
Adressdetails
Baugruppenträger/ Steckplatz:
0
1
Verbindungsres. (hex):
13
11
TSAP:
13 01
11 01
SIMATIC-ACC
Subnetz ID:
D7E3 - 0000 - 0001

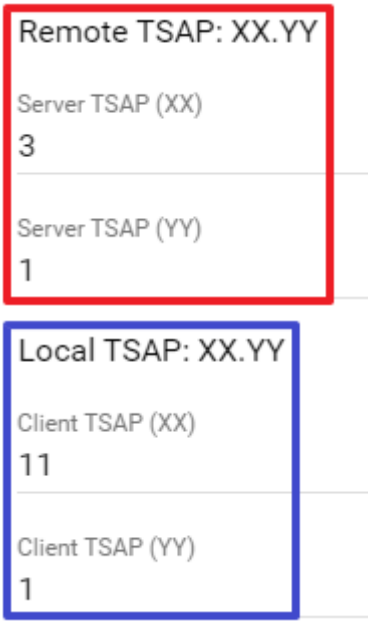
**Einstellungen in IP-Symcon:**




S7 Verbindung Secure

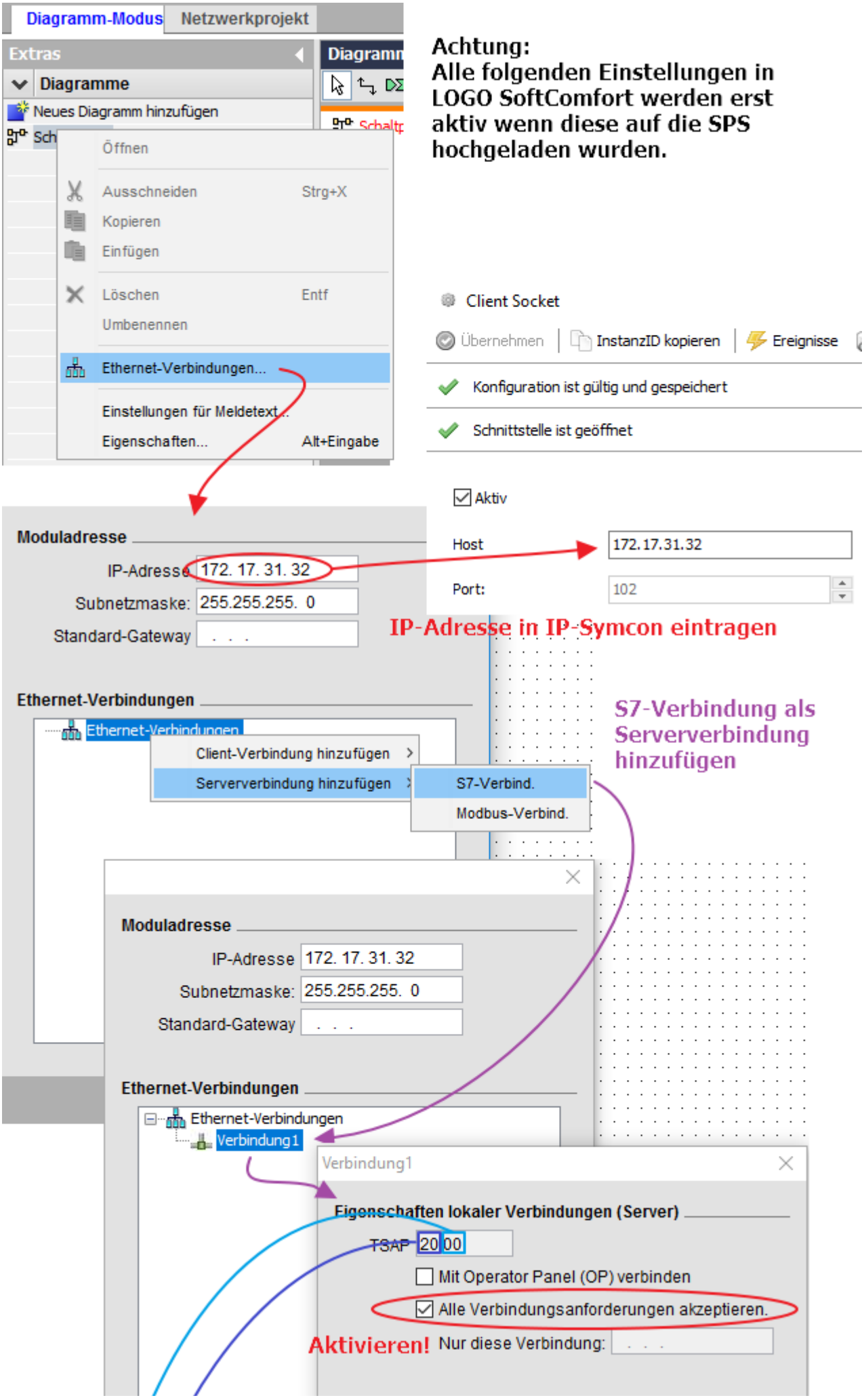
Wenn eine Siemens SPS mit zusätzlichen Sicherheitseinstellungen genutzt wird (z.B. S7-1517f), kann eine Verbindung nur mit aktivierten SIMATIC-ACC auf der SPS Seite genutzt werden. Das heißt in der S7-Verbindung der SPS muss für die lokale TSAP die Option "SIMATIC-ACC" aktiviert werden. Daraus resultiert in der Gatewaykonfiguration in IP-Symcon die Remote TSAP 03.01.

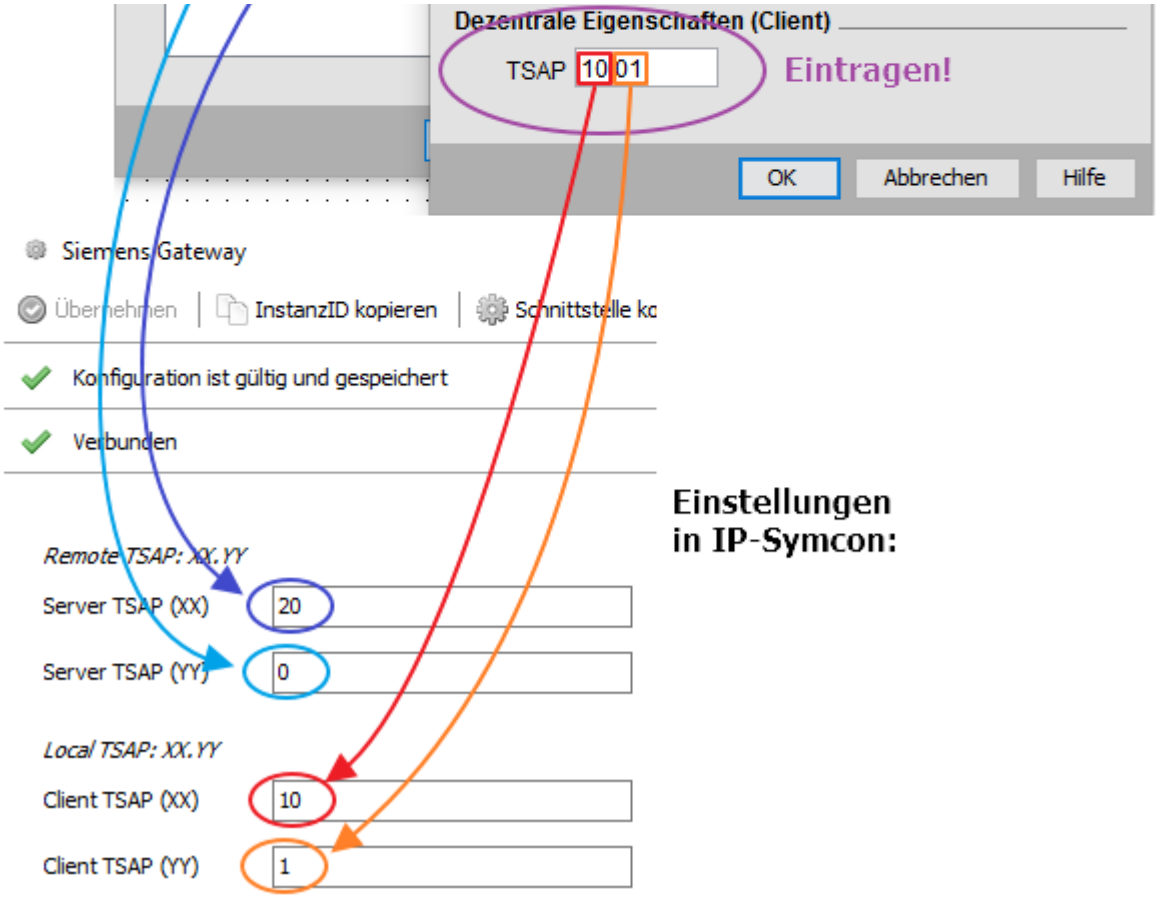




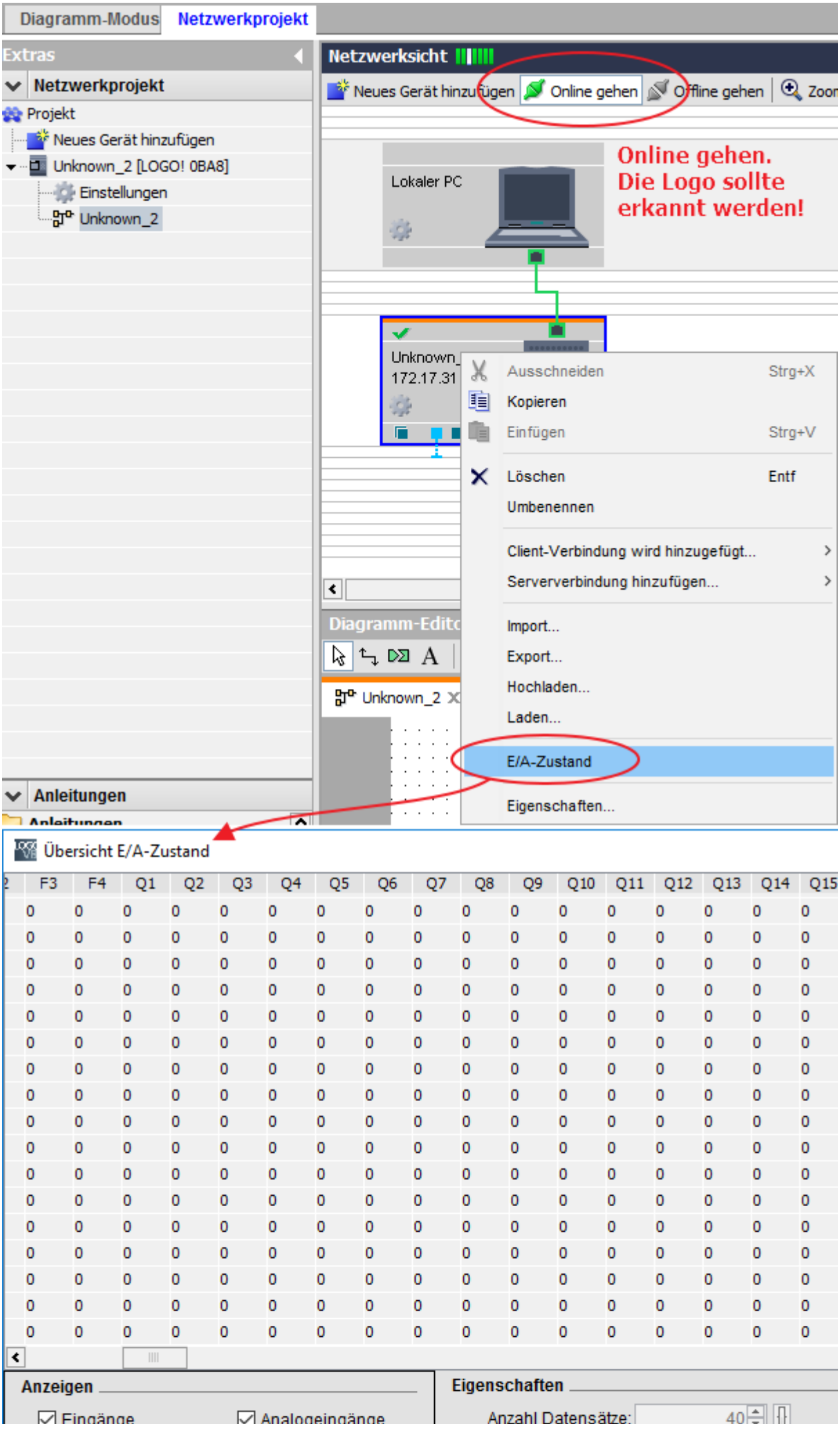

## LOGO 7/8 KONFIGURATION

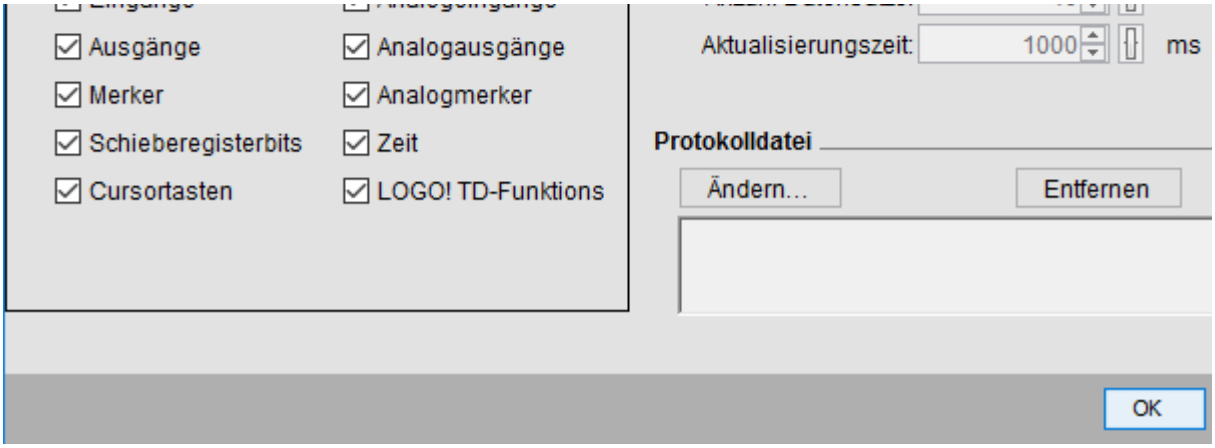
Die Konfiguration der SPS und des Gateways muss wie folgt eingerichtet werden.
Konfiguration über LOGO!Soft Comfort 8.1.


Diagramm-Modus
Netzwerkprojekt
Extras
Diagramme
Neues Diagramm hinzufügen
Öffnen
Ausschneiden Strg+X
Kopieren
Einfügen
Löschen Entf
Umbenennen
Ethernet-Verbindungen...
Einstellungen für Meldetext...
Eigenschaften... Alt+Eingabe
Achtung:
Alle folgenden Einstellungen in LOGO SoftComfort werden erst aktiv wenn diese auf die SPS hochgeladen wurden.
Client Socket
Übernehmen
InstanzID kopieren
Ereignisse
Konfiguration ist gültig und gespeichert
Schnittstelle ist geöffnet
Aktiv
Host 172.17.31.32
Port: 102
Moduladresse
IP-Adresse 172. 17. 31. 32
Subnetzmaske: 255.255.255. 0
Standard-Gateway
IP-Adresse in IP-Symcon eintragen
Ethernet-Verbindungen
Client-Verbindung hinzufügen
Serververbindung hinzufügen
S7-Verbind.
Modbus-Verbind.
S7-Verbindung als Serververbindung hinzufügen
Verbindung1
Verbindung1
Eigenschaften lokaler Verbindungen (Server)
TSAP 20 00
Mit Operator Panel (OP) verbinden
Alle Verbindungsanforderungen akzeptieren.
Aktivieren!
Nur diese Verbindung:



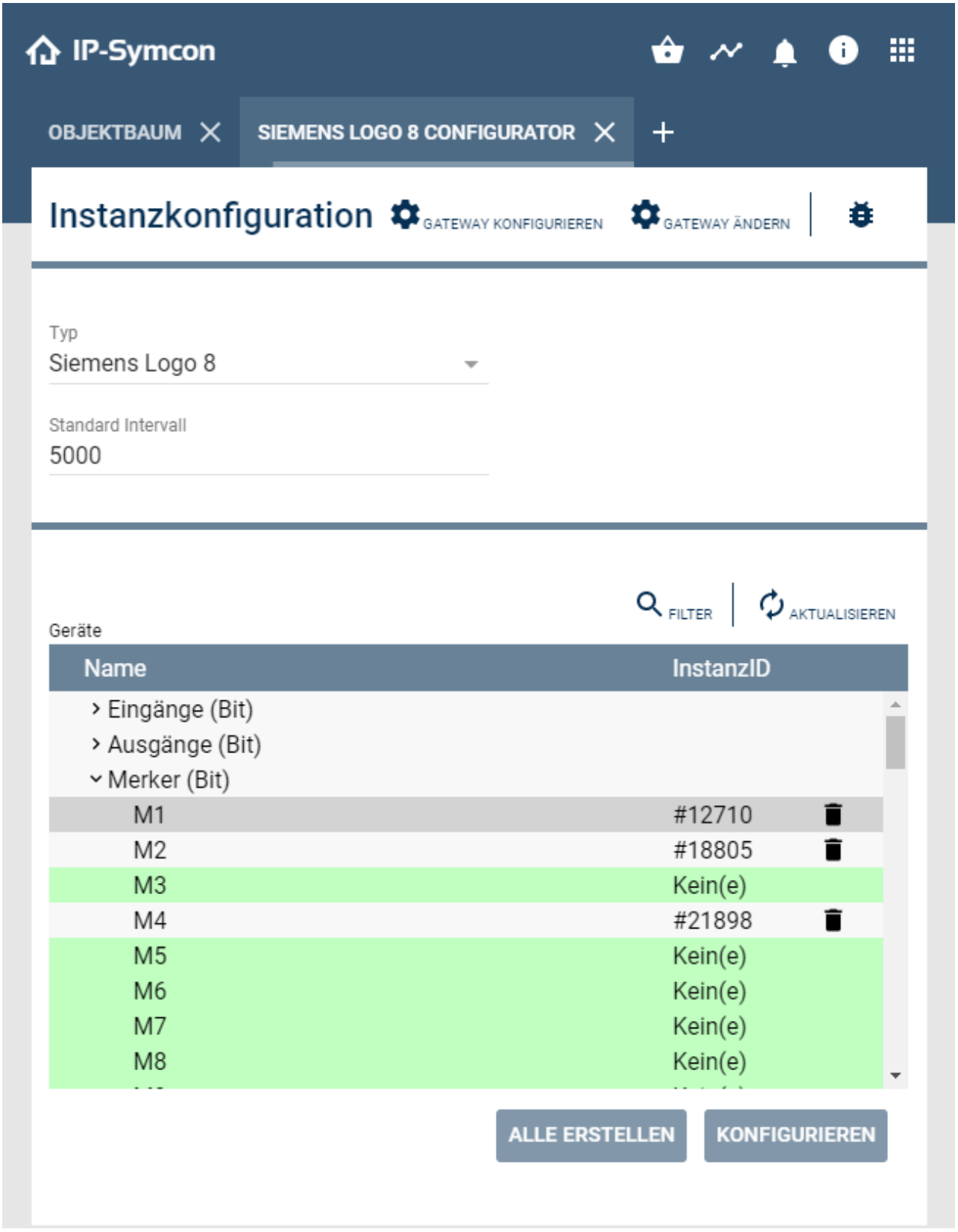

Im Anschluss können die Befehle getestet werden. Zur einfachen Übersicht ob alles korrekt funktioniert, bietet die SoftComfort Software eine Übersicht der E/A-Zustände. Diese kann wiefolgt aufgerufen werden.


Diagramm-Modus
Netzwerkprojekt
Extras
Netzwerkprojekt
Projekt
Neues Gerät hinzufügen
Unknown_2 [LOGO! 0BA8]
Einstellungen
Unknown_2
Netzwerksicht
Neues Gerät hinzufügen
Online gehen
Offline gehen
Lokaler PC
Online gehen.
Die Logo sollte
erkannt werden!
Unknown_
172.17.31
Ausschneiden Strg+X
Kopieren
Einfügen Strg+V
Löschen Entf
Umbenennen
Client-Verbindung wird hinzugefügt...
Serververbindung hinzufügen...
Import...
Export...
Hochladen...
Laden...
E/A-Zustand
Eigenschaften...
Diagramm-Editc
Unknown_2
Anleitungen
Übersicht E/A-Zustand
F3 F4 Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14 Q15
Anzeigen
Eingänge
Analogeingänge
Eigenschaften
Anzahl Datensätze:
40

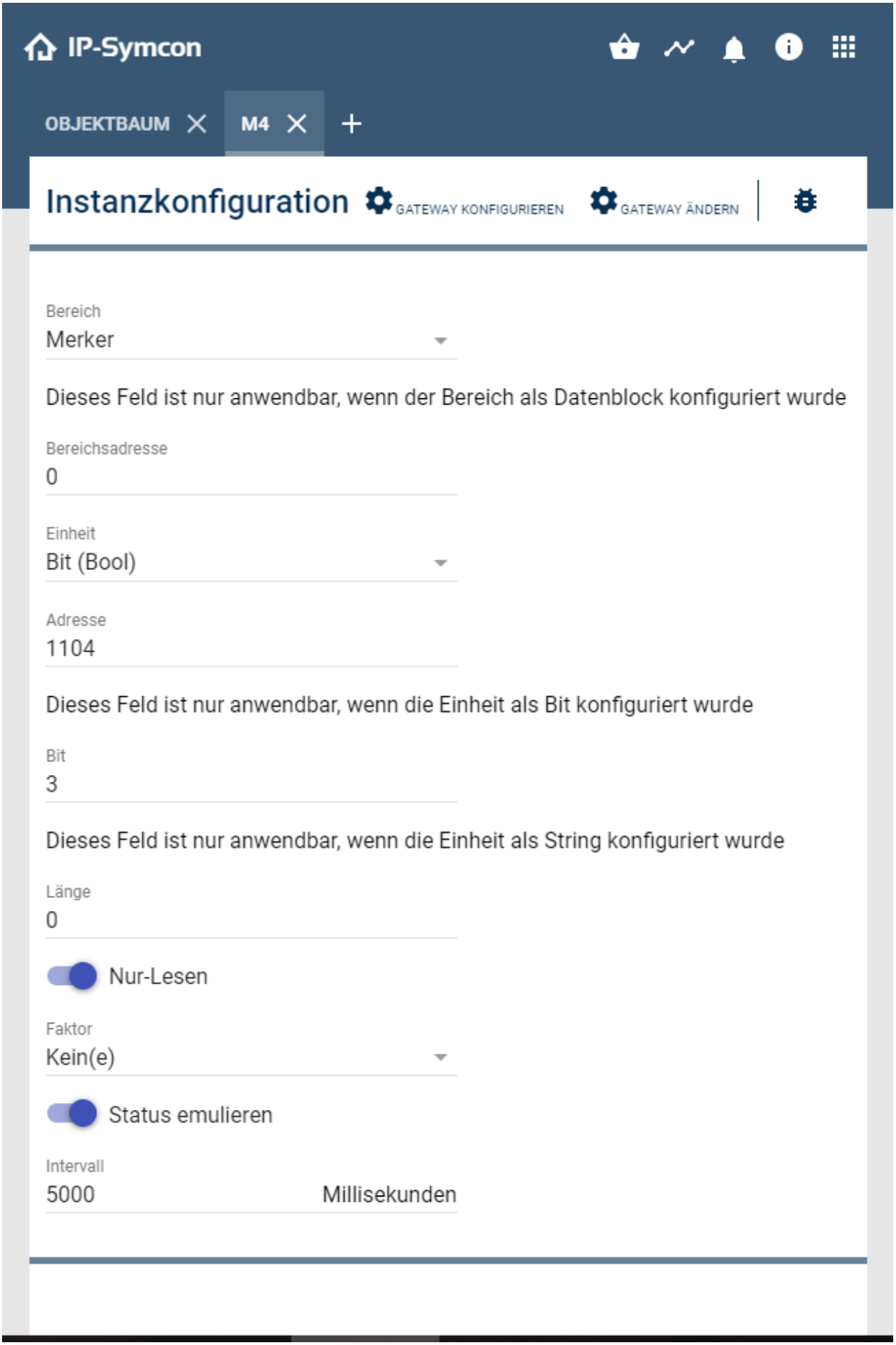



## LOGO 7/8 KONFIGURATOR

Wenn die Logo korrekt eingebunden wurde, können über den Konfigurator die jeweiligen Speicheradressen via Logo 7/8 Konfigurator als Instanz erstellt werden. Dazu muss die der jeweilige Eingang, Ausgang oder Merker ausgewählt und auf "Erstellen" geklickt werden.




Sobald im Konfigurator ein Modul erstellt und auf die Konfigurationstaste gedrückt wurde, öffnet sich ein neuer Dialog.
Neue Instanzen werden im Objektbaum in der Hauptkategorie erstellt. Diese erstellten Instanzen können dann entsprechend umbenannt und an anderer Stelle einsortiert werden. Auch besteht die Möglichkeit via "Konfigurieren" im Konfigurator die jeweilige Instanzkonfiguration aufzurufen.

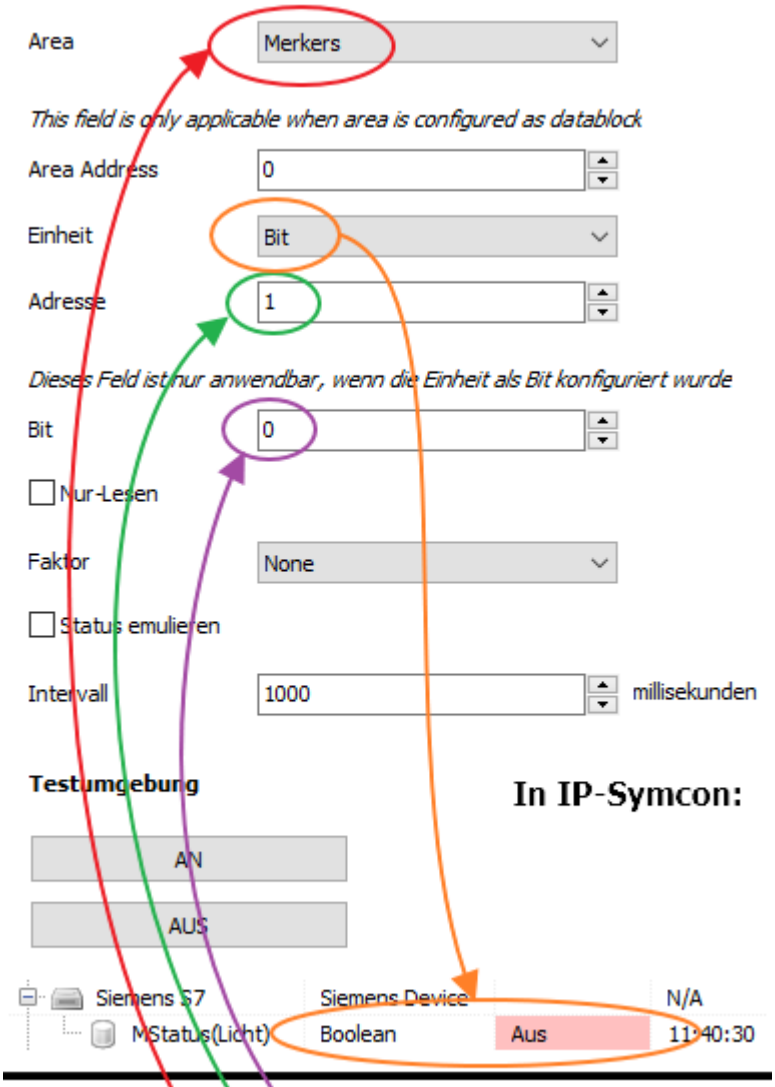



## BEISPIEL MIT TIA PORTAL

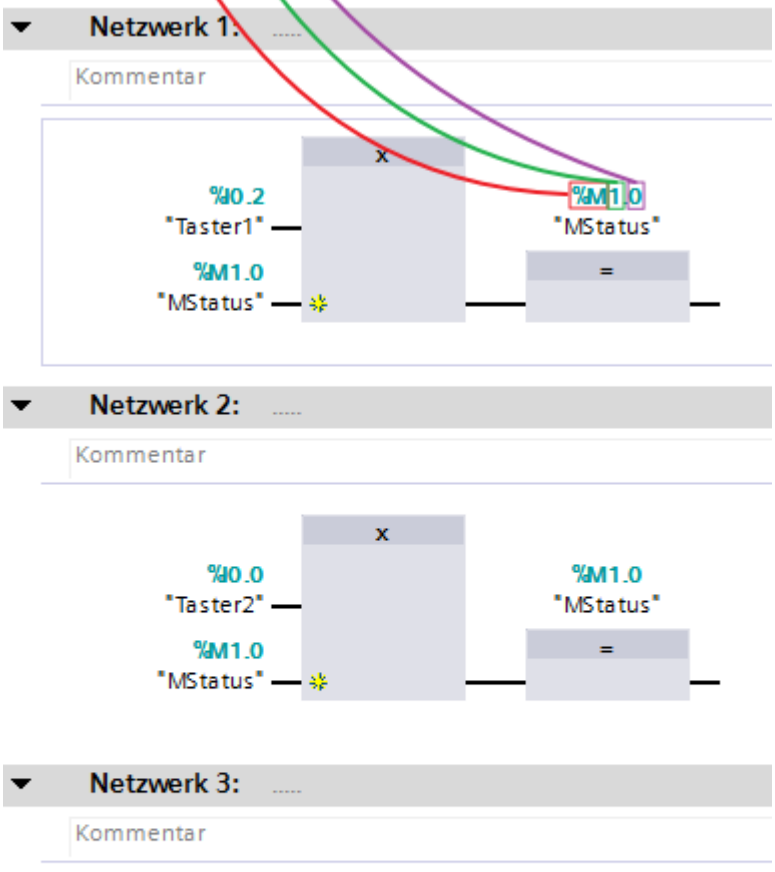
Dieses Beispiel auf der S7-1200 spiegelt 2 Taster, welche unabhängig voneinander ein Licht ein-/ausschalten sollen. Gleichzeitig soll es möglich sein, das Licht ebenfalls via IP-Symcon zu steuern.

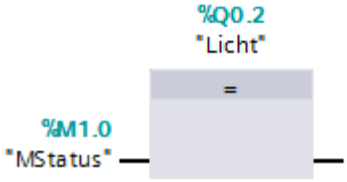
- Main (OB1) wie auf dem Screenshot konfigurieren. (Siehe Screenshot)
- Programm auf die SPS laden und starten
- In IP-Symcon "Siemens S7" Instanz hinzufügen
- Konfigurationsseite der "Siemens S7" Instanz konfigurieren (Siehe Screenshot)
- Adresse und Bit ergeben sich aus dem TIA Portal (Siehe Screenshot)
- Der Typ wird unter dem dem Feld "Area" angegeben (Siehe Screenshot)
- Die Art der Variable in IP-Symcon wird über Einheit angegeben (Siehe Screenshot)
- Der Status des Merkers wird im eingetragengen Intervall abgefragt. (Siehe Screenshot)
- Über Testumgebung (AN/AUS) kann direkt getestet werden, ob der Merker korrekt geschaltet wird

Durch die XOR-Schaltung und den Merker ist es per Taster möglich umzuschalten und IP-Symcon bekommt die Variablenänderung mit. Somit zeigt IP-Symcon immer den Status des Lichtes an, ohne das eine seperate Variable zum Auslesen des SPS-Ausgangs genutzt werden muss. Auch ist es IP-Symcon möglich direkt über den Merker das Licht zu schalten. Teils bessere Ergebnisse erzielt man bei diesem Beispiel, wenn man ein AND (Flanke) vor das jeweilige XOR setzt um nur einen Zyklus zu bekommen. Ob dies benötigt wird muss aber vor Ort getestet werden und wird in diesem Beispiel nicht mit angeführt.




## Im TIA-Portal:





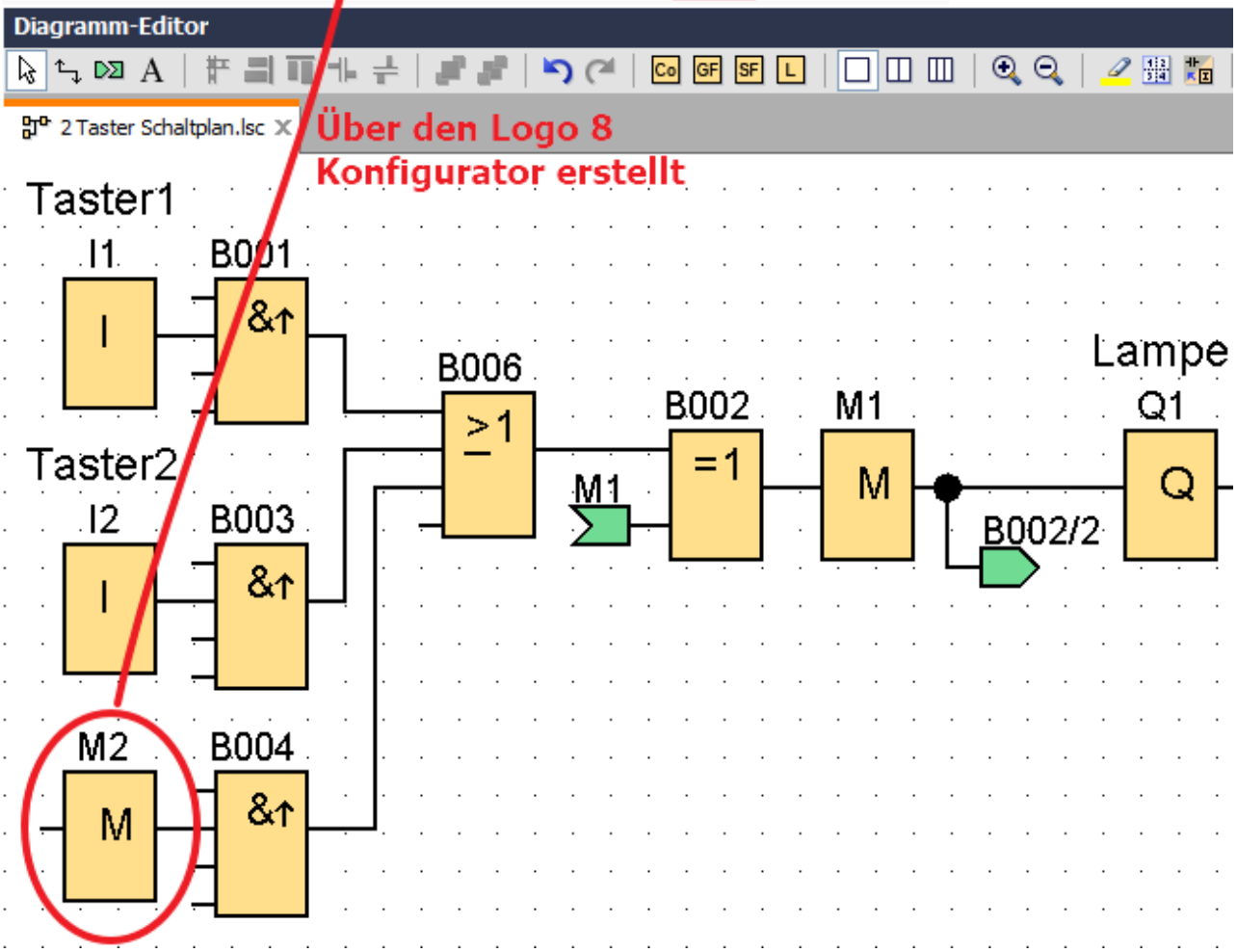

## BEISPIEL MIT LOGO SOFT COMFORT

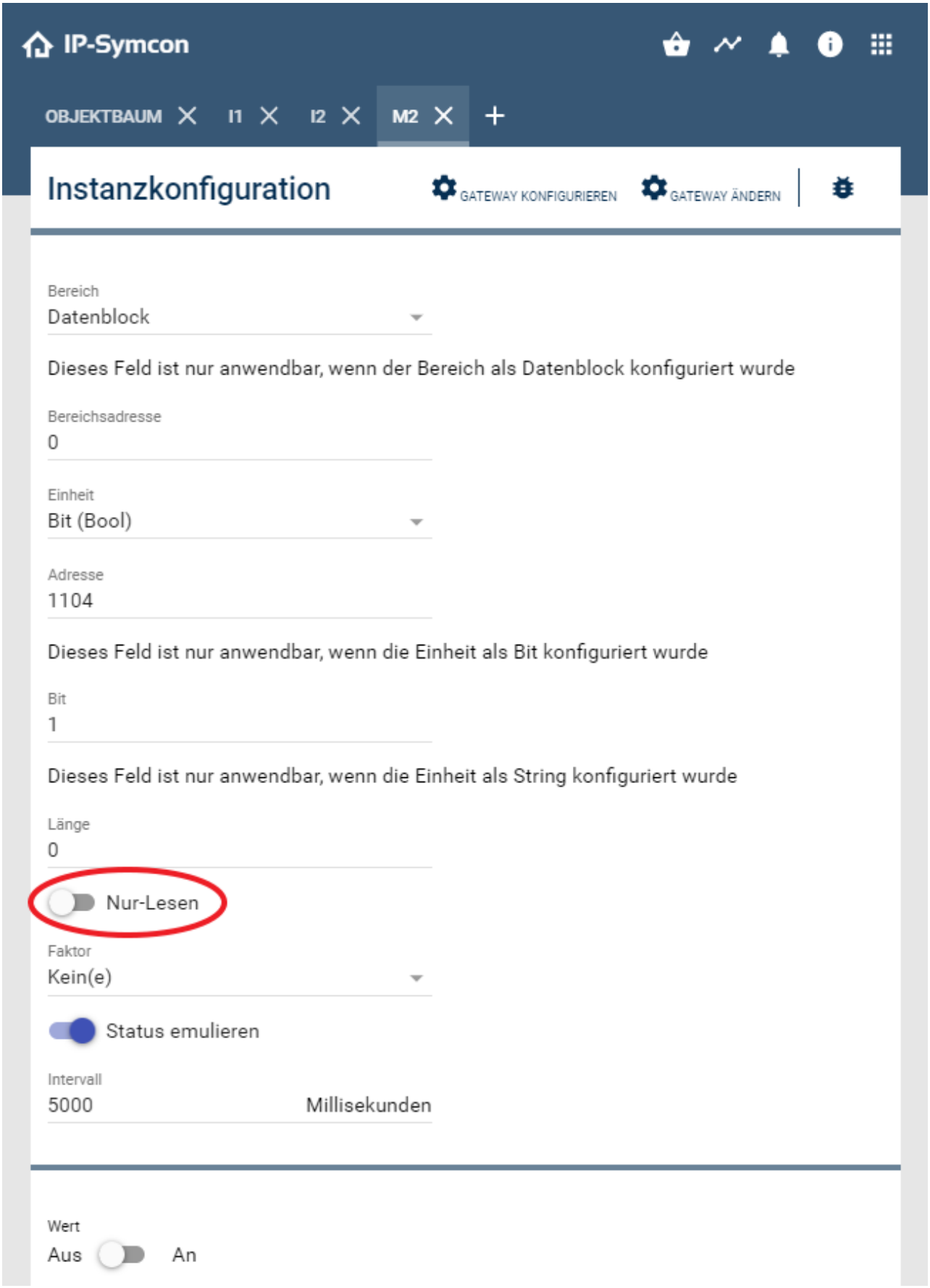
Dieses Beispiel auf der Logo 8 spiegelt 2 Taster, welche unabhängig voneinander ein Licht ein-/ausschalten sollen. Gleichzeitig soll es möglich sein, das Licht ebenfalls via IP-Symcon zu steuern.

2 Taster Schaltplan wie auf dem Screenshot konfigurieren. (Siehe Screenshot)
Programm auf die SPS laden und starten
In IP-Symcon "Logo Configurator" Instanz hinzufügen und Schnittstelle wie oben beschrieben konfigurieren
Die gewünschten Variablen über den Konfigurator erstellen
Beim schalbaren Merker M2 die Option "Nur-Lesen" deaktivieren
Die Art der Variable in IP-Symcon wird über Einheit angegeben
Der Status des Merkers wird im eingetragengen Intervall abgefragt. (Siehe Screenshot)
Über Testumgebung (AN/AUS) kann direkt getestet werden, ob der Merker korrekt geschaltet wird

Durch die XOR-Schaltung und den Merker ist es per Taster möglich umzuschalten und IP-Symcon bekommt bei eingerichtetem Q2 die Variablenänderung mit. Auch ist es IP-Symcon möglich direkt über den Merker M2 das Licht zu schalten. Dies ist nicht über den Merker M1 möglich, da die Logo das in dieser Konfiguration nicht unterstützt. Der Merker M2 fungiert in IP-Symcon als Schalter und in der Logo als Taster.

| | | | | | |
|---|---|---|---|---|---|
| 26668 | I1 | Siemens Device | | |
| 23990 | Wert | Boolean | Aus | 28.07.2020 |
| 32394 | I2 | Siemens Device | | |
| 46964 | Wert | Boolean | Aus | 28.07.2020 |
| 45374 | M1 | Siemens Device | | |
| 45574 | Wert | Boolean | Aus | 09:36:01 |
| 53760 | M2 | Siemens Device | | |
| 20011 | Wert | Boolean | Aus | 09:35:56 |
| 33360 | Q1 | Siemens Device | | |
| 41412 | Wert | Boolean | Aus | 09:36:01 |




IP-Symcon
OBJEKTBAUM
I1
I2
M2
Instanzkonfiguration
GATEWAY KONFIGURIEREN
GATEWAY ÄNDERN
Bereich
Datenblock
Dieses Feld ist nur anwendbar, wenn der Bereich als Datenblock konfiguriert wurde
Bereichsadresse
0
Einheit
Bit (Bool)
Adresse
1104
Dieses Feld ist nur anwendbar, wenn die Einheit als Bit konfiguriert wurde
Bit
1
Dieses Feld ist nur anwendbar, wenn die Einheit als String konfiguriert wurde
Länge
0
Nur-Lesen
Faktor
Kein(e)
Status emulieren
Intervall
5000
Millisekunden
Wert
Aus
An

## DATENTYPEN

| DATENTYP | VORZEICHEN | BIT | NAME IP-SYMCON |
|---|---|---|---|
| BOOL | unsigned | 1 | Bit |
| BYTE | abhängig | 8 | Char/Byte |
| WORD | abhängig | 16 | Short/Word |
| DWORD | abhängig | 32 | Integer/DWord |
| SINT | signed | 8 | Char |
| INT | signed | 16 | Short |
| DINT | signed | 32 | Integer |
| USINT | unsigned | 8 | Byte |
| UINT | unsigned | 16 | Word |
| UDINT | unsigned | 32 | DWord |
| REAL | signed | 32 | Real |

## LOGO VM-ADRESSEN

Die Logo hat für Ein-/Ausgänge und Merker feste Speicheradressen im VM-Speicher oberhalb des 850sten Bytes reserviert.
Diese sind je nach vorliegendem Modell (Logo7/8) unterschiedlich.
Wenn die Logo mit IP-Symcon ausgelesen werden soll, ist es hilfreich diese Adressen zu kennen.
Eine gesamte Liste befindet sich hier: LogoVMAdressen

# Geräteliste

## UNTERSTÜTZTE GATEWAYS/KOMPONENTEN

| HERSTELLER | GERÄTENAME | BESCHREIBUNG |
|---|---|---|
| Siemens | Logo 7 | |
| Siemens | Logo 8 | |
| Siemens | S7 200/300/400 | Über Ethernet angeschlossen |
| Siemens | S7 1200/1500 | Weitere Informationen können HIER gefunden werden |
| Siemens | S7 1517f | Weitere Informationen können HIER gefunden werden. Des Weiteren ist die besondere S7 Konfiguration der S7 zu beachten |

# Logo-VM-Adressen

Die Logo hat für Ein-/Ausgänge und Merker feste Speicheradressen.
Diese sind je nach vorliegendem Modell (Logo7/8) unterschiedlich. (Quelle LOGO!Soft Comfort Online Hilfe )



Diese Adressen müssen innnerhalb der Logokonfiguration in IP-Symcon unter "Bereichsadresse" eingetragen werden. Die jeweiligen Bits unter "Bit"

## LOGO 7


Eingänge Bit
Ausgänge Bit
Analoge Eingänge Word
Analoge Ausgänge Word
Merker Bit
Analoge Merker Word


## LOGO 8


Eingänge Bit
Ausgänge Bit
Analoge Eingänge Word
Analoge Ausgänge Word
Merker Bit
Analoge Merker Word
Netzwerkeingänge Bit
Netzwerkausgänge Bit
Analoge Netzwerkeingänge Word
Analoge Netzwerkausgänge Word


### LOGO 7 EINGÄNGE BIT

| EINGANG | ADRESSE | BIT |
|---|---|---|
| I1 | 923 | 0 |
| I2 | 923 | 1 |
| I3 | 923 | 2 |
| I4 | 923 | 3 |
| I5 | 923 | 4 |
| I6 | 923 | 5 |
| I7 | 923 | 6 |
| I8 | 923 | 7 |
| I9 | 924 | 0 |
| I10 | 924 | 1 |
| I11 | 924 | 2 |
| I12 | 924 | 3 |
| I13 | 924 | 4 |
| I14 | 924 | 5 |
| I15 | 924 | 6 |
| I16 | 924 | 7 |
| I17 | 925 | 0 |
| I18 | 925 | 1 |
| I19 | 925 | 2 |
| I20 | 925 | 3 |
| I21 | 925 | 4 |
| I22 | 925 | 5 |
| I23 | 925 | 6 |
| I24 | 925 | 7 |

## LOGO 7 AUSGÄNGE BIT

| AUSGANG | ADRESSE | BIT |
|---|---|---|
| Q1 | 942 | 0 |
| Q2 | 942 | 1 |
| Q3 | 942 | 2 |
| Q4 | 942 | 3 |
| Q5 | 942 | 4 |
| Q6 | 942 | 5 |
| Q7 | 942 | 6 |
| Q8 | 942 | 7 |
| Q9 | 943 | 0 |
| Q10 | 943 | 1 |
| Q11 | 943 | 2 |
| Q12 | 943 | 3 |
| Q13 | 943 | 4 |
| Q14 | 943 | 5 |
| Q15 | 943 | 6 |
| Q16 | 943 | 7 |

## LOGO 7 ANALOGE EINGÄNGE WORD

| EINGANG | ADRESSE | BIT |
|---|---|---|
| AI1 | 926 | 0 |
| AI2 | 928 | 0 |
| AI3 | 930 | 0 |
| AI4 | 932 | 0 |
| AI5 | 934 | 0 |
| AI6 | 936 | 0 |
| AI7 | 938 | 0 |
| AI8 | 940 | 0 |

## LOGO 7 ANALOGE AUSGÄNGE WORD

| WORD | ADRESSE | BIT |
|---|---|---|
| AQ1 | 944 | 0 |
| AQ2 | 946 | 0 |

## LOGO 7 MERKER BIT

| MERKER | ADRESSE | BIT |
|---|---|---|
| M1 | 948 | 0 |
| M2 | 948 | 1 |
| M3 | 948 | 2 |
| M4 | 948 | 3 |
| M5 | 948 | 4 |
| M6 | 948 | 5 |
| M7 | 948 | 6 |
| M8 | 948 | 7 |
| M9 | 949 | 0 |
| M10 | 949 | 1 |
| M11 | 949 | 2 |
| M12 | 949 | 3 |
| M13 | 949 | 4 |
| M14 | 949 | 5 |
| M15 | 949 | 6 |
| M16 | 949 | 7 |
| M17 | 950 | 0 |
| M18 | 950 | 1 |
| M19 | 950 | 2 |
| M20 | 950 | 3 |
| M21 | 950 | 4 |
| M22 | 950 | 5 |
| M23 | 950 | 6 |
| M24 | 950 | 7 |
| M25 | 951 | 0 |
| M26 | 951 | 1 |
| M27 | 951 | 2 |

## LOGO 7 ANALOGE MERKER WORD

| MERKER | ADRESSE | BIT |
|---|---|---|
| AM1 | 952 | 0 |
| AM2 | 954 | 0 |
| AM3 | 956 | 0 |
| AM4 | 958 | 0 |
| AM5 | 960 | 0 |
| AM6 | 962 | 0 |
| AM7 | 964 | 0 |
| AM8 | 966 | 0 |
| AM9 | 968 | 0 |
| M10 | 970 | 0 |
| AM11 | 972 | 0 |
| AM12 | 974 | 0 |
| AM13 | 976 | 0 |
| AM14 | 978 | 0 |
| AM15 | 980 | 0 |
| AM16 | 982 | 0 |

## LOGO 8

### LOGO 8 EINGÄNGE BIT

| EINGANG | ADRESSE | BIT |
|---|---|---|
| I1 | 1024 | 0 |
| I2 | 1024 | 1 |
| I3 | 1024 | 2 |
| I4 | 1024 | 3 |
| I5 | 1024 | 4 |
| I6 | 1024 | 5 |
| I7 | 1024 | 6 |
| I8 | 1024 | 7 |
| I9 | 1025 | 0 |
| I10 | 1025 | 1 |
| I11 | 1025 | 2 |
| I12 | 1025 | 3 |
| I13 | 1025 | 4 |
| I14 | 1025 | 5 |
| I15 | 1025 | 6 |
| I16 | 1025 | 7 |
| I17 | 1026 | 0 |
| I18 | 1026 | 1 |
| I19 | 1026 | 2 |
| I20 | 1026 | 3 |
| I21 | 1026 | 4 |
| I22 | 1026 | 5 |
| I23 | 1026 | 6 |
| I24 | 1026 | 7 |

## LOGO 8 AUSGÄNGE BIT

| AUSGANG | ADRESSE | BIT |
|---|---|---|
| Q1 | 1064 | 0 |
| Q2 | 1064 | 1 |
| Q3 | 1064 | 2 |
| Q4 | 1064 | 3 |
| Q5 | 1064 | 4 |
| Q6 | 1064 | 5 |
| Q7 | 1064 | 6 |
| Q8 | 1064 | 7 |
| Q9 | 1065 | 0 |
| Q10 | 1065 | 1 |
| Q11 | 1065 | 2 |
| Q12 | 1065 | 3 |
| Q13 | 1065 | 4 |
| Q14 | 1065 | 5 |
| Q15 | 1065 | 6 |
| Q16 | 1065 | 7 |
| Q17 | 1066 | 0 |
| Q18 | 1066 | 1 |
| Q19 | 1066 | 2 |
| Q20 | 1066 | 3 |

## LOGO 8 ANALOGE EINGÄNGE WORD

| EINGANG | ADRESSE | BIT |
|---|---|---|
| AI1 | 1032 | 0 |
| AI2 | 1034 | 0 |
| AI3 | 1036 | 0 |
| AI4 | 1038 | 0 |
| AI5 | 1040 | 0 |
| AI6 | 1042 | 0 |
| AI7 | 1044 | 0 |
| AI8 | 1046 | 0 |

## LOGO 8 ANALOGE AUSGÄNGE WORD

| AUSGÄNGE | ADRESSE | BIT |
|---|---|---|
| AQ1 | 1072 | 0 |
| AQ2 | 1074 | 0 |
| AQ3 | 1076 | 0 |
| AQ4 | 1078 | 0 |
| AQ5 | 1080 | 0 |
| AQ6 | 1082 | 0 |
| AQ7 | 1084 | 0 |
| AQ8 | 1086 | 0 |

## LOGO 8 MERKER BIT

| MERKER | ADRESSE | BIT |
|---|---|---|
| M1 | 1104 | 0 |
| M2 | 1104 | 1 |
| M3 | 1104 | 2 |
| M4 | 1104 | 3 |
| M5 | 1104 | 4 |
| M6 | 1104 | 5 |
| M7 | 1104 | 6 |
| M8 | 1104 | 7 |
| M9 | 1105 | 0 |
| M10 | 1105 | 1 |
| M11 | 1105 | 2 |
| M12 | 1105 | 3 |
| M13 | 1105 | 4 |
| M14 | 1105 | 5 |
| M15 | 1105 | 6 |
| M16 | 1105 | 7 |
| M17 | 1106 | 0 |
| M18 | 1106 | 1 |
| M19 | 1106 | 2 |
| M20 | 1106 | 3 |
| M21 | 1106 | 4 |
| M22 | 1106 | 5 |
| M23 | 1106 | 6 |
| M24 | 1106 | 7 |
| M25 | 1107 | 0 |
| M26 | 1107 | 1 |
| M27 | 1107 | 2 |
| M28 | 1107 | 3 |
| M29 | 1107 | 4 |
| M30 | 1107 | 5 |
| M31 | 1107 | 6 |
| M32 | 1107 | 7 |
| M33 | 1108 | 0 |
| M34 | 1108 | 1 |
| M35 | 1108 | 2 |
| M36 | 1108 | 3 |
| M37 | 1108 | 4 |
| M38 | 1108 | 5 |

LOGO 8 ANALOGE MERKER WORD

| MERKER | ADRESSE | BIT |
|---|---|---|
| AM1 | 1118 | 0 |
| AM2 | 1120 | 0 |
| AM3 | 1122 | 0 |
| AM4 | 1124 | 0 |
| AM5 | 1126 | 0 |
| AM6 | 1128 | 0 |
| AM7 | 1130 | 0 |
| AM8 | 1132 | 0 |
| AM9 | 1134 | 0 |
| AM10 | 1136 | 0 |
| AM11 | 1138 | 0 |
| AM12 | 1140 | 0 |
| AM13 | 1142 | 0 |
| AM14 | 1144 | 0 |
| AM15 | 1146 | 0 |
| AM16 | 1148 | 0 |
| AM17 | 1150 | 0 |
| AM18 | 1152 | 0 |
| AM19 | 1154 | 0 |
| AM20 | 1156 | 0 |
| AM21 | 1158 | 0 |
| AM22 | 1160 | 0 |
| AM23 | 1162 | 0 |
| AM24 | 1164 | 0 |
| AM25 | 1166 | 0 |
| AM26 | 1168 | 0 |
| AM27 | 1170 | 0 |
| AM28 | 1172 | 0 |
| AM29 | 1174 | 0 |
| AM30 | 1176 | 0 |
| AM31 | 1178 | 0 |
| AM32 | 1180 | 0 |
| AM33 | 1182 | 0 |
| AM34 | 1184 | 0 |
| AM35 | 1186 | 0 |
| AM36 | 1188 | 0 |
| AM37 | 1190 | 0 |
| AM38 | 1192 | 0 |

# LOGO 8 NETZWERKEINGÄNGE BIT

| EINGANG | ADRESSE | BIT |
|---|---|---|
| NI1 | 1246 | 0 |
| NI2 | 1246 | 1 |
| NI3 | 1246 | 2 |
| NI4 | 1246 | 3 |
| NI5 | 1246 | 4 |
| NI6 | 1246 | 5 |
| NI7 | 1246 | 6 |
| NI8 | 1246 | 7 |
| NI9 | 1247 | 0 |
| NI10 | 1247 | 1 |
| NI11 | 1247 | 2 |
| NI12 | 1247 | 3 |
| NI13 | 1247 | 4 |
| NI14 | 1247 | 5 |
| NI15 | 1247 | 6 |
| NI16 | 1247 | 7 |
| NI17 | 1248 | 0 |
| NI18 | 1248 | 1 |
| NI19 | 1248 | 2 |
| NI20 | 1248 | 3 |
| NI21 | 1248 | 4 |
| NI22 | 1248 | 5 |
| NI23 | 1248 | 6 |
| NI24 | 1248 | 7 |
| NI25 | 1249 | 0 |
| NI26 | 1249 | 1 |
| NI27 | 1249 | 2 |
| NI28 | 1249 | 3 |
| NI29 | 1249 | 4 |
| NI30 | 1249 | 5 |
| NI31 | 1249 | 6 |
| NI32 | 1249 | 7 |
| NI33 | 1250 | 0 |
| NI34 | 1250 | 1 |
| NI35 | 1250 | 2 |
| NI36 | 1250 | 3 |
| NI37 | 1250 | 4 |
| NI38 | 1250 | 5 |

## LOGO 8 NETZWERKAUSGÄNGE BIT

| AUSGANG | ADRESSE | BIT |
|---|---|---|
| NQ1 | 1390 | 0 |
| NQ2 | 1390 | 1 |
| NQ3 | 1390 | 2 |
| NQ4 | 1390 | 3 |
| NQ5 | 1390 | 4 |
| NQ6 | 1390 | 5 |
| NQ7 | 1390 | 6 |
| NQ8 | 1390 | 7 |
| NQ9 | 1391 | 0 |
| NQ10 | 1391 | 1 |
| NQ11 | 1391 | 2 |
| NQ12 | 1391 | 3 |
| NQ13 | 1391 | 4 |
| NQ14 | 1391 | 5 |
| NQ15 | 1391 | 6 |
| NQ16 | 1391 | 7 |
| NQ17 | 1392 | 0 |
| NQ18 | 1392 | 1 |
| NQ19 | 1392 | 2 |
| NQ20 | 1392 | 3 |
| NQ21 | 1392 | 4 |
| NQ22 | 1392 | 5 |
| NQ23 | 1392 | 6 |
| NQ24 | 1392 | 7 |
| NQ25 | 1393 | 0 |
| NQ26 | 1393 | 1 |
| NQ27 | 1393 | 2 |
| NQ28 | 1393 | 3 |
| NQ29 | 1393 | 4 |
| NQ30 | 1393 | 5 |
| NQ31 | 1393 | 6 |
| NQ32 | 1393 | 7 |
| NQ33 | 1394 | 0 |
| NQ34 | 1394 | 1 |
| NQ35 | 1394 | 2 |
| NQ36 | 1394 | 3 |
| NQ37 | 1394 | 4 |
| NQ38 | 1394 | 5 |

# LOGO 8 ANALOGE NETZWERKEINGÄNGE WORD

| EINGANG | ADRESSE | BIT |
|---|---|---|
| NAI1 | 1262 | 0 |
| NAI2 | 1264 | 0 |
| NAI3 | 1266 | 0 |
| NAI4 | 1268 | 0 |
| NAI5 | 1270 | 0 |
| NAI6 | 1272 | 0 |
| NAI7 | 1274 | 0 |
| NAI8 | 1276 | 0 |
| NAI9 | 1278 | 0 |
| NAI10 | 1280 | 0 |
| NAI11 | 1282 | 0 |
| NAI12 | 1284 | 0 |
| NAI13 | 1286 | 0 |
| NAI14 | 1288 | 0 |
| NAI15 | 1290 | 0 |
| NAI16 | 1292 | 0 |
| NAI17 | 1294 | 0 |
| NAI18 | 1296 | 0 |
| NAI19 | 1298 | 0 |
| NAI20 | 1300 | 0 |
| NAI21 | 1302 | 0 |
| NAI22 | 1304 | 0 |
| NAI23 | 1306 | 0 |
| NAI24 | 1308 | 0 |
| NAI25 | 1310 | 0 |
| NAI26 | 1312 | 0 |
| NAI27 | 1314 | 0 |
| NAI28 | 1316 | 0 |
| NAI29 | 1318 | 0 |
| NAI30 | 1320 | 0 |
| NAI31 | 1322 | 0 |
| NAI32 | 1324 | 0 |

## LOGO 8 ANALOGE NETZWERKAUSGÄNGE WORD

| AUSGANG | ADRESSE | BIT |
|---|---|---|
| NAQ1 | 1406 | 0 |
| NAQ2 | 1408 | 0 |
| NAQ3 | 1410 | 0 |
| NAQ4 | 1412 | 0 |
| NAQ5 | 1414 | 0 |
| NAQ6 | 1416 | 0 |
| NAQ7 | 1418 | 0 |
| NAQ8 | 1420 | 0 |
| NAQ9 | 1422 | 0 |
| NAQ10 | 1424 | 0 |
| NAQ11 | 1426 | 0 |
| NAQ12 | 1428 | 0 |
| NAQ13 | 1430 | 0 |
| NAQ14 | 1432 | 0 |
| NAQ15 | 1434 | 0 |
| NAQ16 | 1436 | 0 |

# S7_RequestRead

```
boolean S7_RequestRead (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Führt einen erneuten Lesevorgang für das Gerät mit der ID **InstanzID** aus.
Sobald die Funktion ausgeführt wurde, sind die neuen Werte in den Status Variablen abrufbar.

## BEISPIEL

```
// Gerät auslesen
S7_RequestRead(12345);
```

# S7_Write

```
boolean S7_Write (int $InstanzID, float $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | 32Bit Gleitkommawert nach IEEE754 oder Ganzzahlig |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schreibt den Wert **Wert** in die konfigurierte Adresse der Instanz mit der ID **InstanzID**, ohne den genauen Datentyp zu wissen.



Diese Funktion unterstützt auch das Setzen von Werten, wenn ein Faktor gesetzt wurde.



Diese Funktion kann nicht für Bit-Werte genutzt werden. Für Bit-Werte muss S7_WriteBit genutzt werden.

## BEISPIEL

```
// Gleitkommawert
S7_Write(12345, 23.5);

// Ganzzahlig
S7_Write(12345, 123);
```

# S7_WriteBit

```
boolean S7_WriteBit (int $InstanzID, boolean $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | TRUE für An, FALSE für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Schreibt den Wert **Wert** in die konfigurierte Adresse der Instanz mit der ID **InstanzID**.

## BEISPIEL

```
S7_WriteBit(12345, true); //Gerät einschalten
```

# S7_WriteByte

```
boolean S7_WriteByte (int $InstanzID, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | 0-255 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schreibt den Wert **Wert** in die konfigurierte Adresse der Instanz mit der ID **InstanzID**.

## BEISPIEL

```
S7_WriteByte(12345, 123);
```

# S7_WriteChar

Benötigt: IP-Symcon >= 4.4

```
boolean S7_WriteChar (int $InstanzID, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | -128 bis 127 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG



Die Funktion hieß bis einschließlich Version 4.3 **S7_WriteShortInt**

Schreibt den Wert **Wert** in die konfigurierte Adresse der Instanz mit der ID **InstanzID**.

## BEISPIEL

```
S7_WriteChar(12345, -123);
```

# S7_WriteDWord

```
boolean S7_WriteDWord (int $InstanzID, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | 0-4294967295 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schreibt den Wert **Wert** in die konfigurierte Adresse der Instanz mit der ID **InstanzID**.



Aufgrund von einer Beschränkung auf 32-Bit Integer innerhalb von PHP kann nur der Bereich bis 2147483647 genutzt werden.

## BEISPIEL

```
S7_WriteDWord(12345, 123);
```

# S7_WriteInteger

```
boolean S7_WriteInteger (int $InstanzID, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | -2147483648 bis 2147483647 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schreibt den Wert **Wert** in die konfigurierte Adresse der Instanz mit der ID **InstanzID**.

## BEISPIEL

```
S7_WriteInteger(12345, -123);
```

# S7_WriteReal

```
boolean S7_WriteReal (int $InstanzID, float $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | 64Bit Gleitkommawert nach IEEE754 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schreibt den Wert **Wert** in die konfigurierte Adresse der Instanz mit der ID **InstanzID**.

## BEISPIEL

```
S7_WriteReal(12345, 23.5);
```

# S7_WriteShort

Benötigt: IP-Symcon >= 4.4

```
boolean S7_WriteShort (int $InstanzID, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | -32768 bis 32767 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG



Die Funktion hieß bis einschließlich Version 4.3 **S7_WriteSmallInt**

Schreibt den Wert **Wert** in die konfigurierte Adresse der Instanz mit der ID **InstanzID**.

## BEISPIEL

```
S7_WriteShort(12345, -123);
```

# S7_WriteWord

```
boolean S7_WriteWord (int $InstanzID, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | 0-65535 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schreibt den Wert **Wert** in die konfigurierte Adresse der Instanz mit der ID **InstanzID**.

## BEISPIEL

```
S7_WriteWord(12345, 123);
```

# SPS: Wago, Beckhoff, ABB

SPS die ModBus RTU/TCP unterstützen, können in IP-Symcon eingebunden werden. Eine Verbindung wird über ModBus RTU/TCP hergestellt.

Folgende Geräte werden von IP-Symcon unterstützt:
Unterstützte Geräte

Zum Beispiel
Wago 750-841 – Der Anschluss an IP-Symcon erfolgt über die LAN Schnittstelle.

## BESCHREIBUNG

Speicherprogrammierbare Steuerungen (SPS, engl. Programmable Logic Controller, PLC) sind eigenständige elektronische Baugruppen zur Steuerung von Maschinen und Anlagen. Diese arbeiten unabhängig von einem PC. Die Befehle zum Steuern und Regeln sind in einem Programm hinterlegt. IP-Symcon dient zur Visualisierung sowie des Human-Maschine Interface (HMI), also der Schnittstelle zwischen dem Bediener und der Maschine.

## EINRICHTUNG IN IP-SYMCON

Die Wago, Beckhoff und ABB SPS können über das IP-Symcon ModBus RTU/TCP Modul angesteuert werden. Da ModBus das übergeordnete Protokoll zur Ansteuerung ist, werden die Namen der Endgerätehersteller nicht mehr explizit in dem Ansteuerungsmodul erwähnt. Die Dokumentation dazu befindet sich hier: ModBus RTU/TCP

## IEC 61131 NACH MODBUS REGISTER UMRECHNEN

Sie können mit diesem Tool die WAGO (IEC 61131) Adressen schnell in die jeweilige ModBus Adresse umrechnen.
%MW1000

Convert >>>

## TIPPS & TICKS

WAGO ModBus Register Mapping Tabelle

# Sync Remote

Benötigt: IP-Symcon >= 7.1



Das Sync Remote Modul ist eine kostenpflichtige Erweiterung, welche zu jeder bestehenden Symcon Lizenz erworben werden kann. Bei jeder Symcon Lizenz kann 1 Remote System kostenlos integriert werden. Die Erweiterung kann direkt im Shop erworben werden.

Das Sync Remote ermöglicht es einem Server sich mit einem entfernten Client zu synchronisieren. Dabei wird der gesamte Inhalt des Objektbaums des Client in den Objektbaum des Servers integriert. Der Server kann dann den Objektbaum des Clients nutzen als wäre es Teil des eigenen Objektbaums und kann die Objekte darin sowohl auslesen als auch modifizieren.

## FUNKTIONSUMFANG

Vollzugriff auf den Client vom Server aus
Objektbaum des Clients nahtlos in den des Servers integriert
Server kann Client auslesen und modifizieren

## SOFTWARE-INSTALLATION

Sofern die kostenpflichtige Erweiterung für die Lizenz des Servers erworben wurde, kann das Modul dort verwendet werden.

## EINRICHTEN DER INSTANZEN IN SYMCON

Unter "Instanz hinzufügen" kann das 'Sync Remote'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen in der Dokumentation der Instanzen.

### KONFIGURATION

Auf dem Client muss zur Synchronisation der Fernzugriff aktiviert werden. Auf dem Server muss das Sync Remote konfiguriert werden.

| NAME | BESCHREIBUNG |
| --- | --- |
| Aktiviere Synchronisation von entferntem Symcon | Die Synchronisation wird durchgeführt, sofern dieser Schalter aktiv ist |
| Host | Die URL unter welcher der Client erreichbar ist |
| Port | Der Port unter welchem der Client Symcon bereitstellt |
| Benutze SSL | Ist der Schalter gesetzt, wird die Verbindung zum Client via SSL verschlüsselt |
| Überprüfe Peer | Verifiziere, dass das SSL Zertifikat des Hosts korrekt ist (nur sichtbar falls Benutze SSL aktiv ist) |
| Überprüfe Host | Verifiziere, dass die Host URL aus dem SSL Zertifikat mit dem Host aus der Eigenschaft übereinstimmt (nur sichtbar falls Benutze SSL aktiv ist) |
| Benutzername | Der Benutzername, also die Lizenzadresse, des Client |
| Passwort | Das Passwort für den Fernzugriff des Client |
| Timeout | Der Timeout bei Anfragen an den Client in Millisekunden. Benötigt eine Anfrage länger, wird die Verbindung als fehlerhaft markiert und keine Synchronisation durchgeführt |
| Strategie (Profile) | Auswahl zur Strategie zur Synchronisation von Variablenprofilen des Client<br>**Ausschließlich vorhandene Profile zuweisen**: Es werden keine neuen Profile auf dem Server erstellt oder existierende modifiziert. Somit erfolgt die Darstellung von Variablen auf Basis der Profile des Servers und kann von der des Clients abweichen<br>**Erstelle Profile nur sofern nicht existent**: Profile des Clients, welche nicht auf dem Server existieren, werden auch auf dem Server erstellt. Damit werden Variablen mit vorher unbekannten Profilen wie auf dem Client dargestellt. Existierende Profile werden allerdings beibehalten. Somit können diese Variablen unter Umständen unterschiedlich dargestellt werden<br>**Erstelle/Aktualisiere passende Profile**: Erstelle Profile des Clients welche auf dem Server fehlen und modifiziere existierende Profile des Servers, damit Sie dem Client entsprechen. Damit ist sichergestellt, dass die Darstellung von Variablen der des Clients entspricht. Dies könnte allerdings Darstellung von Variablen außerhalb des synchronisierten Bereichs ungewollt modifizieren |
| Intervall | Diese Option gibt an, wie häufig der Server sich mit dem Client synchronisiert |
| Reduzierung der Protokollierung für zyklische Aktualisierungen | Ist dieser Schalter aktiviert, werden die Protokollierungen im Meldungsfenster reduziert |
| Nur Änderungen laden um Datentransferkosten zu sparen (Benutzt mehr CPU Resourcen auf beiden Seiten) | Ist dieser Schalter aktiviert, sendet der Client nur die Änderungen seit der letzten Synchronisation an den Client. Dies verringert die Datenmenge welche übertragen wird, erfordert aber zur Berechnung und Integration der Änderungen mehr Rechenleistung bei beiden Systemen (ab Symcon 7.2) |
| Burst Intervall, wenn | Falls der Server eine synchronisierte Variable schaltet, so wird kurzzeitig die Synchronisation häufiger durchgeführt, damit die dazugehörigen Änderungen schneller sichtbar werden |



Die Systeme sollten die gleiche Version von Symcon verwenden. Bei einem Upgrade der Systeme ist es empfohlen zuerst den Server und dann den Client zu aktualisieren. Ansonsten kann es zu ungewollten Wechselwirkungen kommen.

# Technische Alternative

Technische Alternative bietet eine Steuerung von verschiedenen Reglern und Anlagenüberwachung. Eine Verbindung zu IP-Symcon wird mithilfe von CMI oder BL-NET hergestellt.

## BESCHREIBUNG

IP-Symcon unterstützt, sowohl die BL-NET (UVR1611), als auch die CMI Varianten von der Technischen Alternative.
CMI Installation
BL-NET Installation

## CMI

Die Anbindung an IP-Symcon erfolgt über LAN.

### EINBINDUNG IN IP-SYMCON

Nachdem eine passende CMI Instanz erstellt wurde, muss die IP-Adresse des C.M.I. eingetragen werden.
Ebenso muss das Device, Knotennummer und das Intervall angeben werden.
Ist eine Authentifizierung nötig, muss diese in der übergeordnete Instanz des WWW Reader eingetragen werden.
Über "Update" kann die Verbindung getestet werden.
Variablen werden automatisch bei erstmaligen Abfragen erstellt und bei jedem Intervall aktualisiert.

## BL-NET

Die Anbindung an IP-Symcon erfolgt über LAN oder USB mit Hilfe des "BOOTLOADER" (BL-NET).
In jeden Falle muss die DL (Datenleitung) benutzt werden.



Das Auslesen über den CAN-Bus ist nicht möglich.

### EINBINDUNG IN IP-SYMCON

Nachdem eine Instanz in IP-Symcon erstellt wurde, muss die IP-Adresse des BL-NET angegeben werden. Falls die USB Version genutzt werden soll, muss eine neue übergeordnete Instanz erstellt und bei der Auswahl der SerialPort ausgewählt werden.
Um eine oder mehrere Variablen in der Visualisierung anzeigen zu lassen, muss diesen jeweils ein Profil (z.B. Temperatur) zugeordnet werden um der Visualisierung dadurch erkennbar zu machen, um welche Art von Wert es sich handelt. Da die Regelung und die Eingänge der UVR1611 vom Benutzer definiert werden, kann dies nicht automatisch vom

System vorgenommen werden. Mehr Informationen über Variablenprofile kann hier nachgelesen werden: Variablenprofile
Sollen **zwei (2)** UVR1611 angeschlossen werden, so müssen zwei Instanzen erstellt werden. Einmal mit der ID = 1 und einmal mit der ID = 2. Der Timer darf jedoch nur in einer der beiden Instanzen aktiviert werden, da eine Anfrage beide Datenpakete liefert und somit beide Instanzen aktualisiert.

## TIPPS & TRICKS

Da der BL-NET nicht sehr belastbar ist, sollten so wenig Applikationen wie möglich auf das Gerät zugreifen. Andernfalls kann es zu Ausfällen kommen, in denen der BL-NET nicht erreichbar ist.
Folgende Einträge im Logfile sind normal und kein Fehler:

```
Client Socket | Socket: Verbinde…. #1
Client Socket | Socket: Verbunden
Client Socket | Socket: Getrennt
```

# Geräteliste

## CMI

| GERÄTE | BESCHREIBUNG |
|---|---|
| CAN-BC2 | Schnittstellengerät für CAN-Bus Geräte |
| CAN-EZ2 | Energiezähler, Wärmemengenzähler |
| CAN-EZ3 | Energiezähler, Wärmemengenzähler |
| CAN-I/O45 | RSM610/UVR16x2 Erweiterungsgerät für weitere Ein-/Ausgänge |
| CAN-MTx2 | RSM610/UVR16x2 Erweiterungsgerät mit 4,3“ Touch-Display, eine Bedien- und Anzeigeeinheit |
| RSM610 | Frei programmierbare Regel- und Schaltmodul. Mögliche Erweiterung von UVR1611/16x2 |
| UVR1611 | Universalregelung |
| UVR16x2 | Frei programmierbare Universalregelung |
| UVR610 | Frei programmierbare Universalregelung |
| UVR65* | Universalregelung |
| UVR67* | Universalregelung |

## BL-NET

| GERÄTE | BESCHREIBUNG |
|---|---|
| UVR1611 | Universalregelung |

# Voice over IP

Das VoIP Modul kann Telefonverbindungen aufbauen/entgegennehmen, DTMF Signale verarbeiten, Verbindungsgebundene Daten buffern und Sounds abspielen.



Folgende Geräte werden von IP-Symcon unterstützt:
Unterstützte Geräte

## VORAUSSETZUNGEN

Es wird ein aktives Portal zur Telefonie benötigt. Dies kann eine Fritzbox oder ein Dienst wie Sipgate sein.

## EINBINDUNG IN IP-SYMCON

Zur Einbindung kann via Objekt hinzufügen im Objektbaum ein "VoIP" Gerät hinzugefügt werden.
In der Instanzkonfiguration müssen Domain und Benutzername zum Telefonservice angegeben werden.
Falls benötigt müssen noch AuthID und Passwort eingetragen werden.



Wenn der Rechner mehrere Netzwerkkarten oder -interfaces hat (z.B. auch virtuelle Adapter wie VPN, Docker, Hyper-V) kann es erforderlich sein die IP-Adresse über die die VoIP Verbindung aufgebaut wird zu definieren. Dazu gibt es den passenden Spezialschalter "VoIPInterface"




Um zu verwalten was bei einer aktiven Verbindung passieren soll, muss ein Abarbeitungsskript hinzugefügt werden.

## ABARBEITUNGSSKRIPT

Das Abarbeitungsskript bestimmt das Verhalten bei jeder eingehenden Verbindung. Dazu sind auch die Systemvariablen wichtig.

## BEISPIELE

## SKRIPT FÜR EINGEHENDE ANRUFE (ABARBEITUNGSSKRIPT)

```
if($_IPS['SENDER'] == "VoIP") {
// Wir wollen nur eingehende Anrufe verarbeiten
// $_IPS["INSTANCE"] ist ab IP-Symcon 5.4 verfügbar
if(VoIP_GetConnection($_IPS["INSTANCE"], $_IPS["CONNECTION"])
["Direction"] == 1 /* Ausgehend */) {
return;
}
switch($_IPS["EVENT"]) {
case "Incoming":
IPS_LogMessage("VoIP", "Ein eingehender Anruf");
break;
case "Connect":
IPS_LogMessage("VoIP", "Es wurde eine Verbindung aufgebaut");
break;

case "Disconnect":
IPS_LogMessage("VoIP", "Es wurde eine Verbindung beendet");
break;
case "DTMF":
IPS_LogMessage("VoIP", "Es wurde ein DTMF Signal empfangen");

switch($_IPS["DATA"]) {
case '1':
case '2':
case '3':
case '4':
case '5':
case '6':
IPS_LogMessage("VoIP", "Es wurde eine der Tasten 1 bis 6 gedrückt");
break;
case '#':
IPS_LogMessage("VoIP", "Es wurde die Taste # gedrückt");
break;
default:
IPS_LogMessage("VoIP", "Es wurde die Taste ". $_IPS["DATA"] ."
gedrückt");
break;
}
break;
case "PlayFinish":
IPS_LogMessage("VoIP", "Es wurde eine Sounddatei abgespielt");
break;
default:
IPS_LogMessage("VoIP", "Ein unbekanntes Event wurde ausgelöst");
break;
}
}
```

## SKRIPT FÜR AUSGEHENDE ANRUFE MIT TTS-MODULE "AWS POLLY" AUS DEM MODULE STORE

```
if($_IPS['SENDER'] == "Execute") {
$id = VoIP_Connect(12345, "0451305005xx");
//Maximal 10 Sekunden warten, dass einer abnimmt
for($i = 0; $i < 10; $i++) {
IPS_Sleep(1000);
$c = VoIP_GetConnection(12345, $id);
if($c['Connected']) {
// VoIP_Playwave() unterstützt ausschließlich WAV im Format: 16 Bit,
8000 Hz, Mono.
VoIP_PlayWave(12345, $id, TTSAWSPOLLY_GenerateFile(23456, "IP-Symcon
wünscht einen wunderbaren Tag"));
return;
}
}
//Auflegen, falls keiner abnimmt
VoIP_Disconnect(12345, $id);
}
```

## BEISPIELEINRICHTUNG IN DER FRITZBOX

Mithilfe der FritzBox kann das VoIP-Modul genutzt werden. Hierzu ist eine seperate freie Telefonnummer in der FritzBox empfohlen und folgende Schritte sind zu beachten.

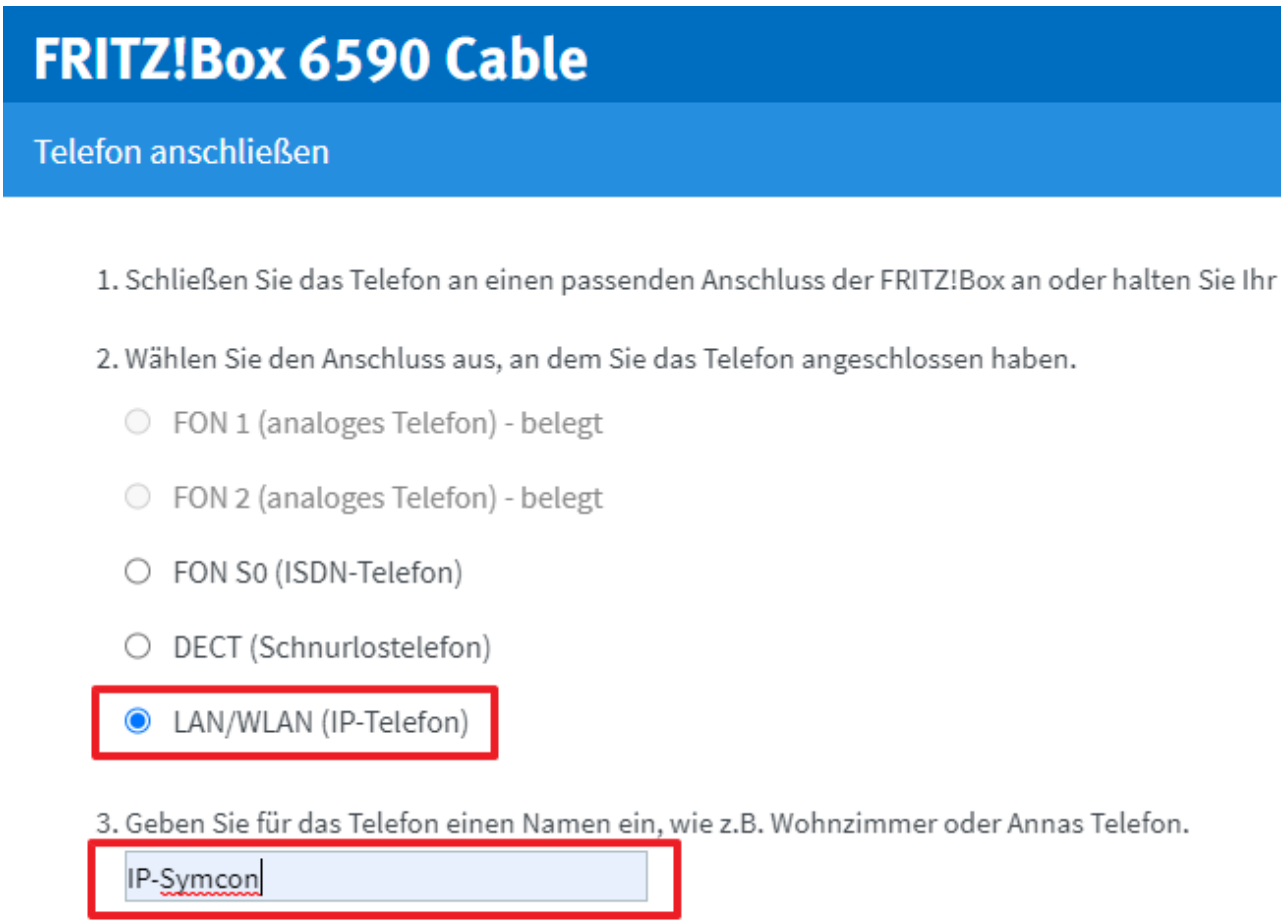


Gegebenfalls muss unter "Telefonie" -> "Eigene Rufnummer" -> "Neue Rufnummer" eine neue Rufnummer eingerichtet werden

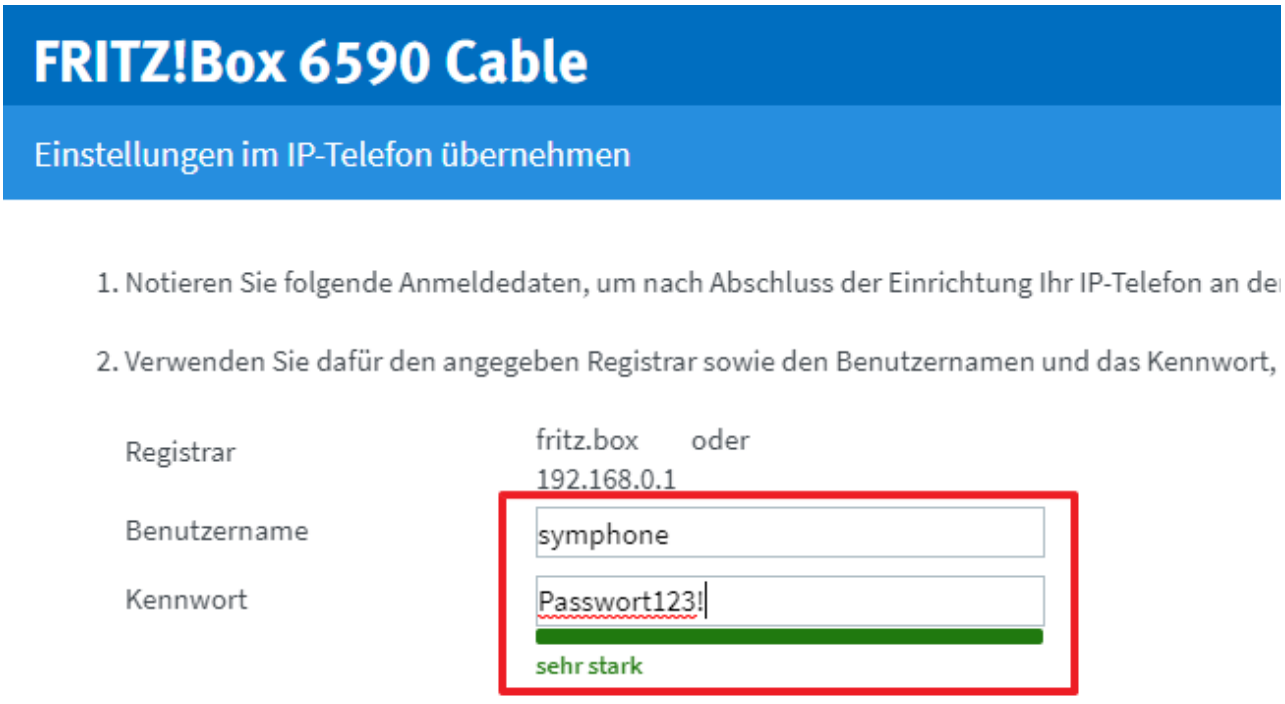
Als erster Schritt muss in der FritzBox unter "Telefonie" -> "Telefoniegeräte" ein neues Telefoniegerät hinzugefügt werden.
Dieses muss als neues "Telefon" hinzugefügt werden.



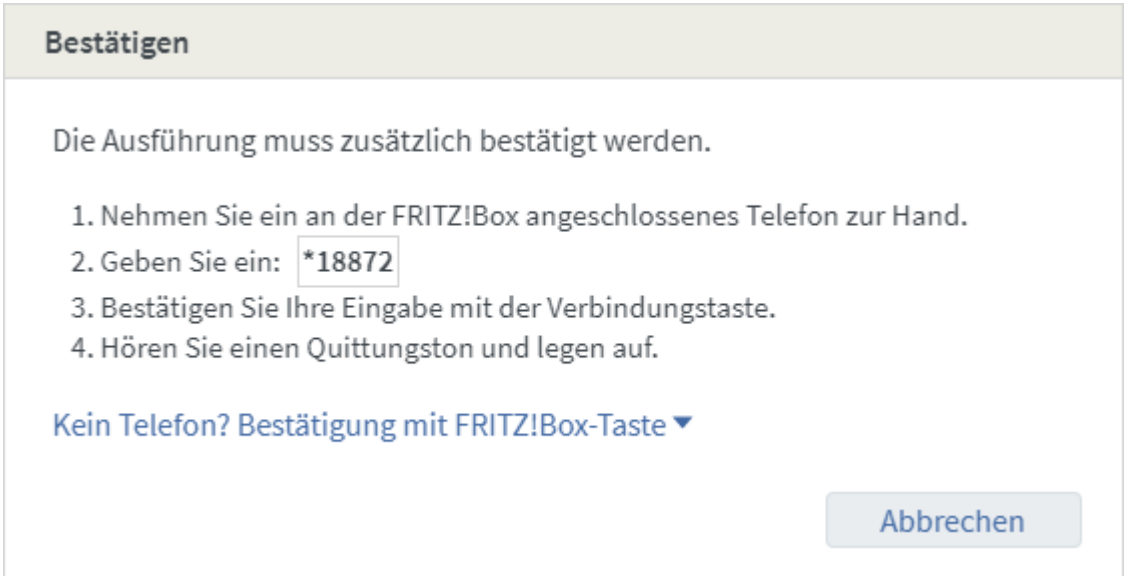

Da IP-Symcon als IP-Telefon fungiert, muss dieses und anschließend eine seperate Telefonnummer ausgewählt werden. Es sollte ein sprechender Name für das Telefoniegerät gewählt werden.

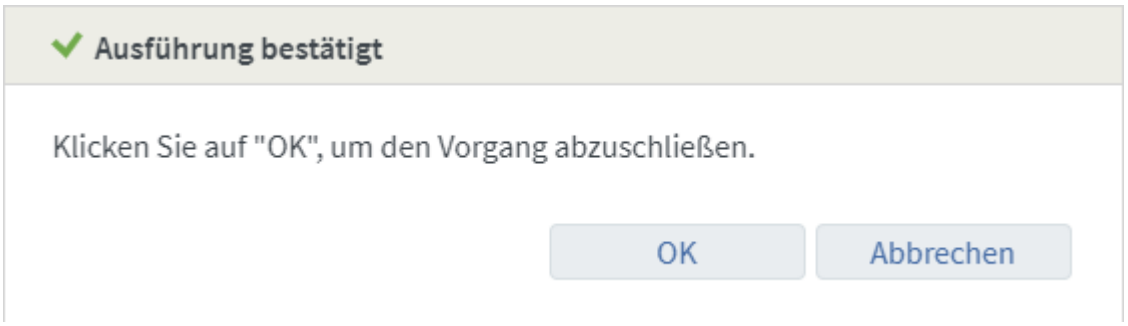



Anschließend müssen ein Benutzername und Passwort festegelegt werden, diese werden später in das VoIP in IP-Symcon eingetragen.

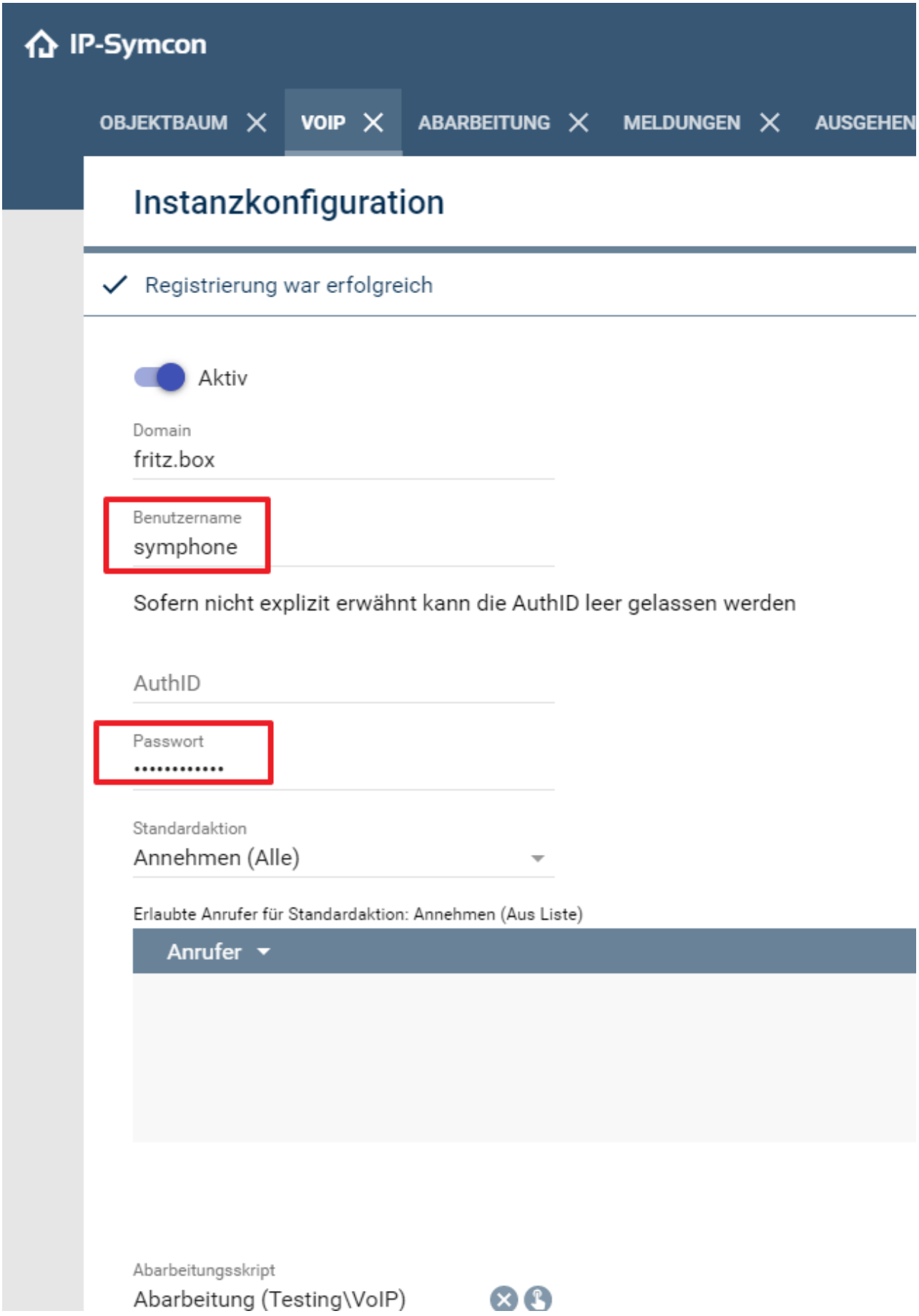



Das einzurichtende Telefoniegerät muss bestätigt und dessen Einrichtung im Anschluss abgeschlossen werden.




Nun kann das Telefoniegerät in IP-Symcon innerhalb des VoIP Modul übernommen werden.
Wenn alles geklappt hat wird der Status "Registrierung war erfolgreich" angezeigt.




Mit eingerichtetem Abarbeitungsskript kann ein Testanruf getätigt werden. In den Meldungen sollten folgende Nachrichten beim Drücken von Zifferntasten auf dem Telefon ersichtlich werden.

OBJEKTBAUM ✕ VOIP ✕ ABARBEITUNG ✕ MELDUNGEN ✕ AUSGEHEND

# Meldungen

| Zeitpunkt | Absender | Nachricht |
|---|---|---|
| 20.07.2020, 16:16:11 | VoIP | Ein eingehender Anruf |
| 20.07.2020, 16:16:11 | VoIP | Es wurde eine Verbindung aufgebaut |
| 20.07.2020, 16:16:15 | VoIP | Es wurde ein DTMF Signal empfangen |
| 20.07.2020, 16:16:15 | VoIP | Es wurde eine der Tasten 1 bis 6 gedrückt |
| 20.07.2020, 16:16:16 | VoIP | Es wurde ein DTMF Signal empfangen |
| 20.07.2020, 16:16:16 | VoIP | Es wurde eine der Tasten 1 bis 6 gedrückt |
| 20.07.2020, 16:16:17 | VoIP | Es wurde ein DTMF Signal empfangen |
| 20.07.2020, 16:16:17 | VoIP | Es wurde eine der Tasten 1 bis 6 gedrückt |
| 20.07.2020, 16:16:18 | VoIP | Es wurde ein DTMF Signal empfangen |
| 20.07.2020, 16:16:18 | VoIP | Es wurde die Taste 7 gedrückt |
| 20.07.2020, 16:16:19 | VoIP | Es wurde ein DTMF Signal empfangen |
| 20.07.2020, 16:16:19 | VoIP | Es wurde die Taste 8 gedrückt |
| 20.07.2020, 16:16:20 | VoIP | Es wurde ein DTMF Signal empfangen |
| 20.07.2020, 16:16:20 | VoIP | Es wurde die Taste 9 gedrückt |
| 20.07.2020, 16:16:20 | VoIP | Es wurde ein DTMF Signal empfangen |
| 20.07.2020, 16:16:20 | VoIP | Es wurde die Taste 9 gedrückt |
| 20.07.2020, 16:16:27 | VoIP | Es wurde eine Verbindung beendet |

# Geräteliste

## UNTERSTÜTZTE KOMPONENTEN

Ein Auszug unterstützter Komponenten.

- Fritzbox
- Sipgate



Dies sind nur die getesteten Geräte. Es können auch weitere Geräte funktionieren.

# VoIP_AcceptCall

Benötigt: IP-Symcon >= 5.4

```
boolean VoIP_AcceptCall (int $InstanzID, int $VerbindungsID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| VerbindungsID | ID der zu schaltenden Verbindung |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Akzeptiert bei der VoIP-Instanz mit der ID **InstanzID** die Verbindung mit der ID **VerbindungsID**.

## BEISPIEL

```
// Akzeptiert die Verbindung mit der ID 3 der Instanz mit der ID
12345
VoIP_AcceptCall(12345, 3);
```

# VoIP_Connect

Benötigt: IP-Symcon >= 5.2

```
int VoIP_Connect (int $InstanzID, string $Nummer)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Nummer | Telefonnummer mit der eine Verbindung aufgebaut werden soll |

## RÜCKGABEWERT

Gibt die ID der Verbindung zurück

## BESCHREIBUNG

Baut bei der VoIP-Instanz mit der ID **InstanzID** eine Verbindung mit der Nummer **Nummer** auf.

## BEISPIEL

```
// Baut eine Verbindung mit der Telefonnummer 0451305005xx auf
VoIP_Connect(12345, "0451305005xx");
```

# VoIP_Disconnect

Benötigt: IP-Symcon >= 5.2

```
boolean VoIP_Disconnect (int $InstanzID, int $VerbindungsID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| VerbindungsID | ID der zu schaltenden Verbindung |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Trennt bei der VoIP-Instanz mit der ID **InstanzID** die Verbindung mit der ID **VerbindungsID**.

## BEISPIEL

```
// Trennt die Verbindung mit der ID 3 der Instanz mit der ID 12345
VoIP_Disconnect(12345, 3);
```

# VoIP_GetConnection

Benötigt: IP-Symcon >= 5.2

```
array VoIP_GetConnection (int $InstanzID, int $VerbindungsID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| VerbindungsID | ID der zu schaltenden Verbindung |

## RÜCKGABEWERT

Array mit folgenden Informationen

| FELD | DATENTYP | BESCHREIBUNG |
|---|---|---|
| ID | integer | VerbindungsID |
| TimeStamp | integer | Zeitstemmpel des Anrufs |
| Number | string | Nummer mit der eine Verbindung aufgebaut wurde |
| Direction | integer | Anrufrichtung: 0 = Eingehender Anruf, 1 = Ausgehender Anruf |
| Connected | boolean | Verbindungsstatus: FALSE = nicht Verbunden, TRUE = Verbunden |
| Disconnected | boolean | Trennungssstatus: FALSE = nicht Getrennt, TRUE = Getrennt |

## BESCHREIBUNG

Gibt die Informationen der Verbindung mit der ID **VerbindungsID** der VoIP-Instanz mit der ID **InstanzID** zurück.
---return---
Array mit folgenden Informationen

| FELD | DATENTYP | BESCHREIBUNG |
|---|---|---|
| ID | integer | VerbindungsID |
| TimeStamp | integer | Zeitstem |

## BEISPIEL

```
// Gibt Informationen der Verbindung mit der ID 3 wieder.
print_r(VoIP_GetConnection(12345, 3));

//Beispielausgabe
```

```
/*
Array
(
[ID] => 3
[TimeStamp] => 1566915689
[Number] => 045130500511
[Data] =>
[Direction] => 1
[Connected] => 1
[Disconnected] =>
)
*/
```

# VoIP_GetData

Benötigt: IP-Symcon >= 5.2

```
string VoIP_GetData (int $InstanzID, int $VerbindungsID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| VerbindungsID | ID der zu schaltenden Verbindung |

## RÜCKGABEWERT

Inhalt der Daten der Verbindung als String

## BESCHREIBUNG

Gibt den Inhalt der Daten der Verbindung mit der ID **VerbindungsID** der VoIP-Instanz mit der ID **InstanzID** zurück.

## BEISPIEL

```
// Gibt den Inhalt der Daten der Verbindung mit der ID 3 zurück.
VoIP_SetData(12345, 3, "Hallo Welt");
print_r(VoIP_GetData(12345, 3));

//Beispielausgabe
/*
Hallo Welt
*/
```

# VoIP_PlayWave

Benötigt: IP-Symcon >= 5.2

```
boolean VoIP_PlayWave (int $InstanzID, int $VerbindungsID, string
$Dateiname)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| VerbindungsID | ID der zu schaltenden Verbindung |
| Dateiname | Pfad und Name der Datei, welche abgespielt werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Spielt auf der VoIP-Instanz mit der ID **InstanzID** die Sounddatei **Dateiname** auf der Verbindung mit der ID **VerbindungsID** ab.



Es wird ausschließlich WAV im Format: 16 Bit, 8000 Hz, Mono unterstützt.



Es wird nicht auf das Beenden des Abspielens der Sounddatei gewartet. Dies wird durch die Systemvariable $_IPS['EVENT'] == "PlayFinish" behandelt.



Es können mithilfe des TTS Moduls oder AWS Polly aus dem Module-Store auch dynamische Sounddateien erstellt werden.

## BEISPIEL

```
// Spielt auf der Verbindung mit der ID 3 die Sounddatei
"willkommen.wav" ab
```

```
VoIP_PlayWave(12345, 3, IPS_GetKernelDir() . "/media/
willkommen.wav");
```

# VoIP_RejectCall

Benötigt: IP-Symcon >= 5.4

```
boolean VoIP_RejectCall (int $InstanzID, int $VerbindungsID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| VerbindungsID | ID der zu schaltenden Verbindung |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Lehnt bei der VoIP-Instanz mit der ID **InstanzID** die Verbindung mit der ID **VerbindungsID** ab.

## BEISPIEL

```
// Lehnt die Verbindung mit der ID 3 der Instanz mit der ID 12345 ab
VoIP_RejectCall(12345, 3);
```

# VoIP_SendDTMF

Benötigt: IP-Symcon >= 5.3

```
boolean VoIP_SendDTMF (int $InstanzID, int $VerbindungsID, string
$DTMF)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| VerbindungsID | ID der zu schaltenden Verbindung |
| DTMF | Zeichen, welche als DTMF gesendet werden sollen |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet auf der VoIP-Instanz mit der ID **InstanzID** die Zeichen **DTMF** auf der Verbindung mit der ID **VerbindungsID**.



Es sind ausschließlich diese Zeichen erlaubt: 0-9, *, #

## BEISPIEL

```
// Sendet auf der Verbindung mit der ID 3 die DTMF Signale für
"*123#"
VoIP_SendDTMF(12345, 3, "*123#");
```

# VoIP_SetData

Benötigt: IP-Symcon >= 5.2

```
boolean VoIP_SetData (int $InstanzID, int $VerbindungsID, string
$Daten)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| VerbindungsID | ID der zu schaltenden Verbindung |
| Daten | Daten, für die Verbindung gesetzt werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt auf der VoIP-Instanz mit der ID **InstanzID** die Daten **Daten** der Verbindung mit der ID **VerbindungsID**.

## BEISPIEL

```
// Setzt den Inhalt der Daten der Verbindung mit der ID 3 auf "Hallo
Welt"
VoIP_SetData(12345, 3, "Hallo Welt");
```

# WinLIRC

Mit Hilfe von Infrarotsignalen lassen sich existierende Audio und Video Geräte in die Hausautomation einbinden. Außerdem können z.B. Licht-Szenarien über die Infrarotfernbedienung aktiviert oder die Rollläden herunter gelassen werden. Dazu ist nur eine Sende- und Empfangsdiode notwendig – je nach Wunsch als Fertiggerät oder zum Basteln…
Wissenswertes über WinLIRC:
winlirc.sourceforge.net
Für alle, die gern mal wieder zum Lötkolben greifen wollen, finden sich hier weitere Informationen:
lirc.org/receivers.html
lirc.org/transmitters.html

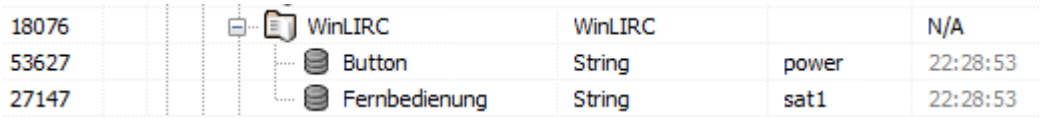
## INSTALLATION

Um IR-Signale über IP-Symcon empfangen/senden zu können, wird ein LIRC kompatibles Gerät benötigt.

WinLIRC Applikation muss installiert und gestartet sein
Download: winlirc.sourceforge.net/de/
Fernbedienung/Tasten müssen in der WinLIRC Applikation angelernt sein
WinLIRC Modul in IP-Symcon anlegen.
Host: localhost
Port: 8765

## EINBINDUNG IN IP-SYMCON

Das WinLIRC Modul stellt sich folgendermaßen im Objektbaum dar:




Eine erfolgreiche Installation ist an dem geöffnetem Socket zu erkennen.




Um IR-Befehle von IP-Symcon auszugeben, wird die WinLIRC_SendOnce-Funktion verwendet.

```
$id = 18076 /*[Media IR\WinLIRC]*/;
// ZDF einschalten
WinLIRC_SendOnce($id, "humax", "power");
IPS_Sleep(1000);
WinLIRC_SendOnce($id, "humax", "power");
IPS_Sleep(200);
WinLIRC_SendOnce($id, "sat", "power");
// hier können noch Befehle zur Beleuchtung stehen ...
```

# WinLIRC_SendOnce

```
boolean WinLIRC_SendOnce (int $InstanzID, string $Fernbedienung,
string $Knopf)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Fernbedienung | Name der Fernbedienung, der in der Fernbedienungsdatenbank eingetragen ist/wurde. |
| Knopf | Name der Fernbedienungsfunktion, die in der Fernbedienungsdatenbank eingetragen ist/wurde. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Sendet einen Fernbedienungsbefehl über die Instanz mit der ID InstanzID ab. Die Werte Fernbedienung und Knopf geben an, welcher Befehl gesendet werden soll. Diese müssen in der Client Software (WinLIRC/IRTrans Tray) entsprechend eingerichtet worden sein.

## BEISPIEL

```
WinLIRC_SendOnce(37279,"soundmaster", "power");
```

# Wireless M-Bus

Wireless M-Bus ist ein von der OMS Group standartisiertes Protokoll zur Verbrauchsdatenerfassung. Zur Verbindung mit IP-Symcon wird ein LAN-, seriell- oder USB-Gateway benötigt.



Folgende Geräte werden von IP-Symcon unterstützt:
Unterstützte Geräte

## INHALT


Installation
Einbindung in IP-Symcon
Konfigurator
Konfiguration


## INSTALLATION

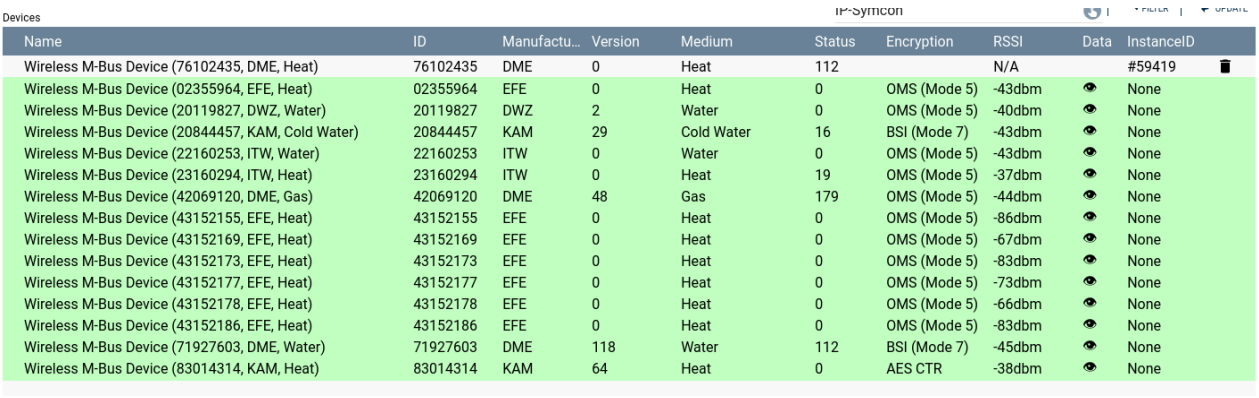
Das **Wireless M-Bus LAN Gateway** wird per LAN mit dem PC verbunden. Standardmäßig ist DHCP und Port 5000 eingerichtet. Sollte eine individuelle IP Adresse gefordert sein, so wird das IP-Symcon "Network Configuration Tool" benötigt. Dieses steht zum Download bereit. Eine einfache Beschreibung zur Konfiguration des Gateways steht zur Verfügung. Im Anschluss ist das Gateway über die eingerichtete IP-Adresse und Port erreichbar.

## EINBINDUNG IN IP-SYMCON

Das LAN-Gateway kann über die Gerätesuche eingebunden werden. Hierfür muss als System "Wireless M-Bus Discovery" gewählt werden. Die Discovery-Instanz bietet dann die Erstellung eines Wireless M-Bus Konfigurators an. Nachdem der Konfigurator erstellt wurde, werden alle empfangenen Geräte gezeigt, welche eingebunden werden.

## KONFIGURATOR

Der Konfigurator zeigt alle empfangenen Geräte an den öffentlichen Informationen an. In den jeweiligen Spalten werden Informationen wie Hersteller, Geräteart, Verschlüsselung und das letzte Empfangsdatum angezeigt. In der Spalte Daten können die empfangenen Rohdaten angesehen werden, um diese ggf. dem Symcon Support zukommen zu lassen.


Devices

IP-Symcon

| Name | ID | Manufactu... | Version | Medium | Status | Encryption | RSSI | Data | InstanceID |
|---|---|---|---|---|---|---|---|---|---|
| Wireless M-Bus Device (76102435, DME, Heat) | 76102435 | DME | 0 | Heat | 112 | | N/A | | #59419 |
| Wireless M-Bus Device (02355964, EFE, Heat) | 02355964 | EFE | 0 | Heat | 0 | OMS (Mode 5) | -43dbm | | None |
| Wireless M-Bus Device (20119827, DWZ, Water) | 20119827 | DWZ | 2 | Water | 0 | OMS (Mode 5) | -40dbm | | None |
| Wireless M-Bus Device (20844457, KAM, Cold Water) | 20844457 | KAM | 29 | Cold Water | 16 | BSI (Mode 7) | -43dbm | | None |
| Wireless M-Bus Device (22160253, ITW, Water) | 22160253 | ITW | 0 | Water | 0 | OMS (Mode 5) | -43dbm | | None |
| Wireless M-Bus Device (23160294, ITW, Heat) | 23160294 | ITW | 0 | Heat | 19 | OMS (Mode 5) | -37dbm | | None |
| Wireless M-Bus Device (42069120, DME, Gas) | 42069120 | DME | 48 | Gas | 179 | OMS (Mode 5) | -44dbm | | None |
| Wireless M-Bus Device (43152155, EFE, Heat) | 43152155 | EFE | 0 | Heat | 0 | OMS (Mode 5) | -86dbm | | None |
| Wireless M-Bus Device (43152169, EFE, Heat) | 43152169 | EFE | 0 | Heat | 0 | OMS (Mode 5) | -67dbm | | None |
| Wireless M-Bus Device (43152173, EFE, Heat) | 43152173 | EFE | 0 | Heat | 0 | OMS (Mode 5) | -83dbm | | None |
| Wireless M-Bus Device (43152177, EFE, Heat) | 43152177 | EFE | 0 | Heat | 0 | OMS (Mode 5) | -73dbm | | None |
| Wireless M-Bus Device (43152178, EFE, Heat) | 43152178 | EFE | 0 | Heat | 0 | OMS (Mode 5) | -66dbm | | None |
| Wireless M-Bus Device (43152186, EFE, Heat) | 43152186 | EFE | 0 | Heat | 0 | OMS (Mode 5) | -83dbm | | None |
| Wireless M-Bus Device (71927603, DME, Water) | 71927603 | DME | 118 | Water | 112 | BSI (Mode 7) | -45dbm | | None |
| Wireless M-Bus Device (83014314, KAM, Heat) | 83014314 | KAM | 64 | Heat | 0 | AES CTR | -38dbm | | None |

Neue Instanzen werden im Objektbaum in der Hauptkategorie erstellt. Diese erstellten Instanzen können dann entsprechend umbenannt und an anderer Stelle einsortiert werden. Auch besteht die Möglichkeit via "Konfigurieren" im Konfigurator die jeweilige Instanzkonfiguration aufzurufen.

## KONFIGURATION

Auf der Konfigurationsseite des Gerätes kann der Geräteschlüssel eingegeben werden. Dieser muss aus 32 Zeichen bestehen und darf nur Großbuchstaben/Zahlen ohne Leerzeichen enthalten. Alternativ können auch Schlüssellisten in der Wireless M-Bus Gateway Instanz importiert werden, welche dann von der Instanz direkt genutzt werden.

## KONFIGURATION

In der Wireless M-Bus Gateway Instanz können Schlüssellisten importiert werden, sodass dieser schneller und automatisierter dem Symcon System zur Verfügung stehen. Dabei werden folgende Formate unterstützt:

```
012345678;00000000000000000000000000000000 (Adresse;Schlüssel)
012345678;TESTTEST;00000000000000000000000000000000 (Beliebige
weitere Spalten)
SOX;012345678;00000000000000000000000000000000 (Beliebige weitere
Spalten)
012345678;00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
(Leerzeichen dürfen beliebig im Schlüssel vorkommen)
012345678,00000000000000000000000000000000 (Komma und Semikolon sind
als Trennzeichen erlaubt)
0SOX012345678;00000000000000000000000000000000 (Adresse im DIN
Format)
```

Eine Unterstützung für das OMS SecProfile A Format ist in Planung.

# Geräteliste

## UNTERSTÜTZTE GATEWAYS

Folgende Wireless M-Bus Gateways sind kompatibel:

| HERSTELLER | BESCHREIBUNG |
|---|---|
| SymBox mit Wireless M-Bus Erweiterung Jetzt bestellen | Seriell integriertes Modul |
| Wireless M-Bus LAN Gateway Jetzt bestellen | IP Gateway |
| Würth Elektronik 2605056083001 | METIS I, USB, AMB8465-M, integrierte Antenne |
| Würth Elektronik 2607056283001 | METIS II, USB, AMB8665-M, externe Antenne |
| IMST iU891A-XL | Neues Modell mit externer Antenne |
| IMST iM871-A | Altes Modell |
| WEPTECH SWAN2 | Über HTTP/HTTPS |
| WEPTECH SWAN3 | Über HTTP/HTTPS |

## UNTERSTÜTZTE KOMPONENTEN

IP-Symcon unterstützt die folgenden Verschlüsselungsmodi nach DIN EN 13757-7
- Mode 0 (Keine Verschlüsselung)
- Mode 5 (OMS)
- Mode 7 (BSI)

IP-Symcon unterstützt die folgenden Verschlüsselungsmodi nach DIN EN 13757-4
- AES-128-CTR

IP-Symcon unterstützt das Frame-Format A/B und als Empfangsmodus C/T. Zusätzlich werden die etwas spezielleren Compact Frames, welche die Kamstrup Geräte oft verwenden, unterstützt.

# WMRS200

WMRS200 ist ein Wetterstationsystem und nutzt funkbasierende Sensorkomponenten. Eine Verbindung mit IP-Symcon ist über ein USB-Gateway möglich.



Folgende Geräte werden von IP-Symcon unterstützt:
Unterstützte Komponenten

## INSTALLATION

Voraussetzung für den Betrieb mit IP-Symcon ist ein korrekt angeschlossener USB-Kommunikationshub.

## EINBINDUNG IN IP-SYMCON

Im Objektbaum kann über das "+" -> "Instanz hinzufügen" oder über Rechtsklick -> "Objekt hinzufügen" -> Instanz im erscheinenden Dialog nach WMRS gesucht werden. Es stehen die einzelnen Geräte zur Auswahl.
Gateway und I/O Instanzen werden automatisch erstellt und es muss einzig der richtige USB Anschluss im Gateway, der Gerätetyp in der Instanz ausgewählt und die passende Geräte-ID eingetragen werden. Über die "Suchen" Funktion innerhalb der Instanzkonfiguration kann die Geräte-ID eingetragen werden.

# Geräteliste

## UNTERSTÜTZTE GATEWAYS

| GATEWAY | HERSTELLER | ANSCHLUSSART | BESCHREIBUNG |
|---|---|---|---|
| USB Kommunikationshub | Oregon Scientific | Seriell / USB | USB-Schnittstelle aus Bundle der Wetterstation |

## UNTERSTÜTZTE KOMPONENTEN

| KOMPONENTE | HERSTELLER | ANSCHLUSSART | BESCHREIBUNG |
|---|---|---|---|
| Windsensor | Oregon Scientific | Funk | Aus Bundle der Wetterstation |
| Regensensor | Oregon Scientific | Funk | Aus Bundle der Wetterstation |
| Luftdrucksensor | Oregon Scientific | Funk | Aus Bundle der Wetterstation |
| Temperatur / Luftfeuchtigkeit | Oregon Scientific | Funk | Aus Bundle der Wetterstation |
| THGR800 | Oregon Scientific | Funk | Thermo-Hygro (3-Kanal) - Temperatur / Luftfeuchtigkeit |
| THGR810 | Oregon Scientific | Funk | Thermo-Hygro (10-Kanal) - Temperatur / Luftfeuchtigkeit |
| UVN800 | Oregon Scientific | Funk | Pyranometer - Remote-UV-Sensor |

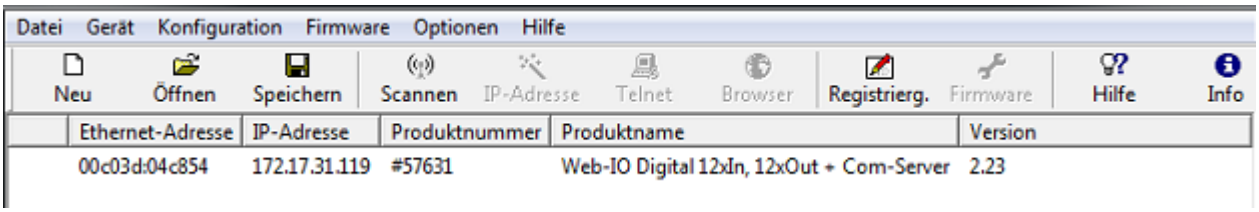
# W&T

Über eine TCP/IP-Ethernet 10/100BaseT Schnittstelle erlauben die W&T Web-IO Digital Boxen den Zugriff über eine Vielzahl von Protokollen und Anwendungen.

## INSTALLATION

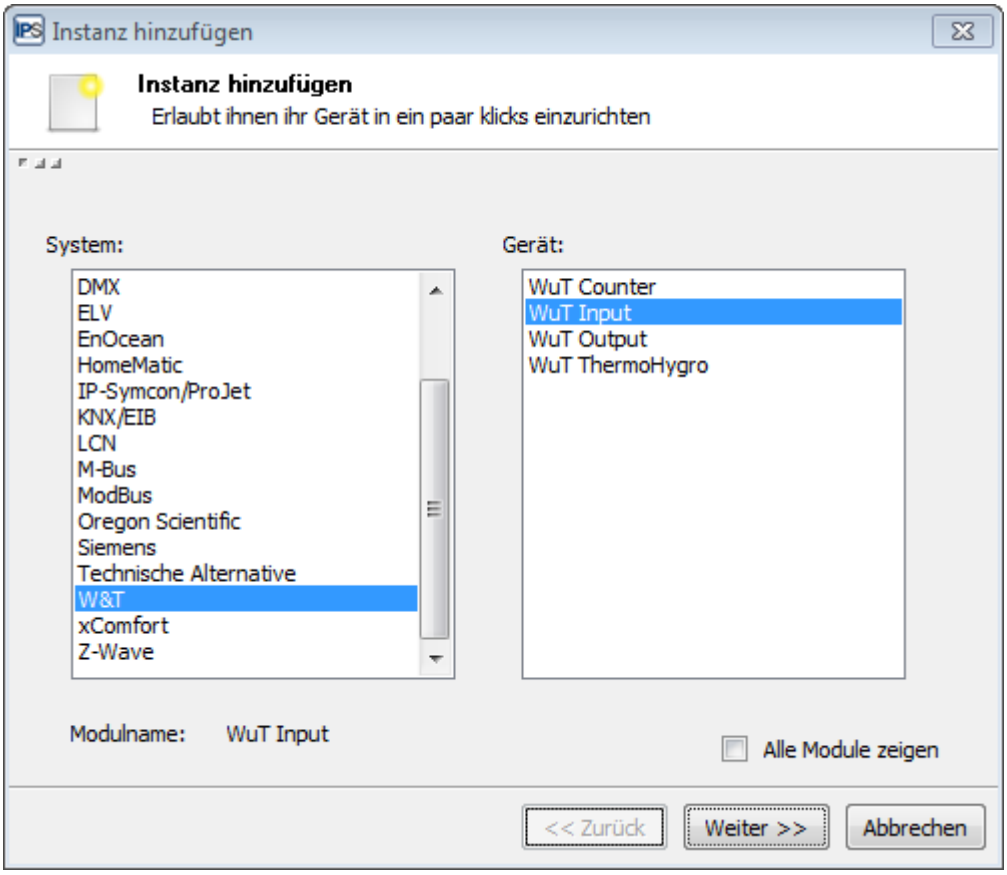
Voraussetzung für den Betrieb mit IP-Symcon ist ein korrekt angeschlossenes Web-IO und ein DHCP-Server. (Der DHCP-Server übernimmt automatisch die Adressvergabe für die Netzwerkteilnehmer)
Erste Konfigurationsmöglichkeiten und Einstellmöglichkeiten (z.B. Neuvergabe der IP-Adresse) ist mithilfe der **Wutility-Software** möglich. Steht diese nicht zur Verfügung, so kann diese unter www.wut.de heruntergeladen werden. Nach der Installation der Software kann das Web-IO erkannt und genutzt werden.

> Sofern kein DHCP Server zur Verfügung steht, kann die IP Adresse auch mit der Wutility-Software konfiguriert werden

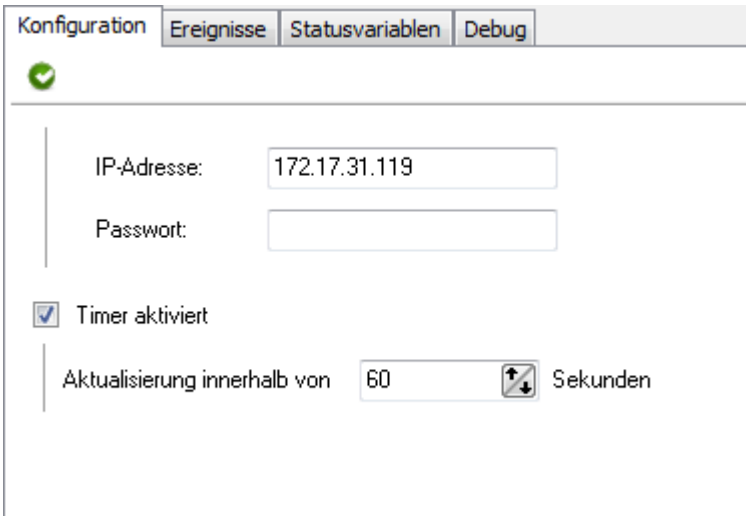



Im Browser kann die Konfigurations-Webpage des Web-IOs durch Eingabe der IP aufgerufen werden. Hier sind erste Einstellungen, vor allem zu Testzwecken, möglich.

## EINBINDUNG IN IP-SYMCON




Es muss in IP-Symcon aus Counter, Input und Ouput, eine dem angeschlossenen Geräten entsprechende Funktion, ausgewählt werden. Wenn mehrere Kanäle des Gerätes eingebunden werden sollen, so kann der Schritt der Anleitung beliebig oft wiederholt werden. Nun muss zweimal auf “Weiter” und einmal auf “OK” geklickt werden. Es öffnet sich beispielsweise die Konfiguration des “WuT Outputs“. Hier sind Einstellungen zum Kanal des Gerätes möglich. Die genaue Belegung der Kanäle kann der Anleitung des W&T Gerätes entnommen werden.
Anschließend muss die IP-Adresse des Web-IOs auf der Konfigurationsseite eingetragen werden. Diese kann durch Klicken auf das Zahnrad (unten links) geöffnet werden.




Ein Passwort ist optional. Die Werkseinstellung beinhaltet kein Passwort.

Der Timer muss aktiviert sein, damit IP-Symcon innerhalb einer manuell eingegebenen Zeit neue Daten vom Web-IO erhält und eine Veränderung sichtbar wird.
(Die beschriebene Konfiguration des Wut Gateways ist nur bei erstmaliger Installation nötig. Beim Hinzufügen weiterer Geräte kann dieser Schritt übersprungen werden.)

# WUT_SwitchMode

```
boolean WUT_SwitchMode (int $InstanzID, boolean $Status)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Status | TRUE für An, FALSE für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Schaltet den Kanal mit der ID InstanzID auf Wert Status.

## BEISPIEL

```
WUT_SwitchMode(12345, true); //Kanal einschalten
```

# XBee

XBee ist ein auf dem ZigBee Funkstandard basierendes Modul. Es ermöglicht als Funkbrücke für eigentlich festverdrahtete Systeme zu dienen.

## INSTALLATION

Zur Verbindung mit IP-Symcon wird ein XBee-Coordinator(Gateway) benötigt, der via Seriell oder Seriell-USB angeschlossen ist.

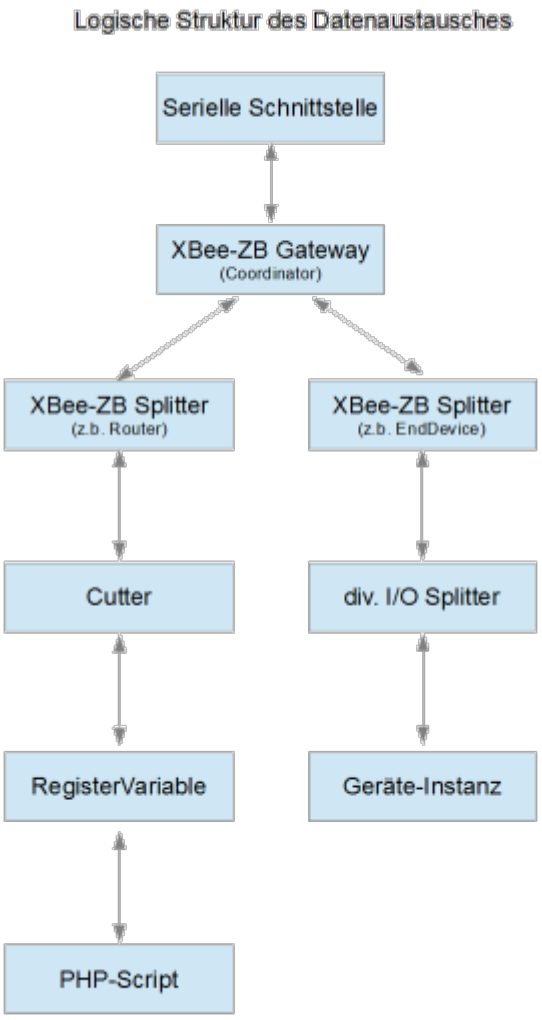
## EINBINDUNG IN IP-SYMCON

Das XBee-Gateway(Coordinator) wird via Serial Port an IP-Symcon eingebunden.
Zu jedem End-Gerät muss eine XBee Splitter Instanz erstellt werden. Diese beinhaltet lediglich die Geräte ID zur Kommunikation innerhalb von XBee.
Für die End-Geräte-Gateways dienen die XBee-Splitter Instanzen als Ersatz übergeordnete Instanz. (Siehe "Beispiel Objektbaum" )

## BEISPIEL

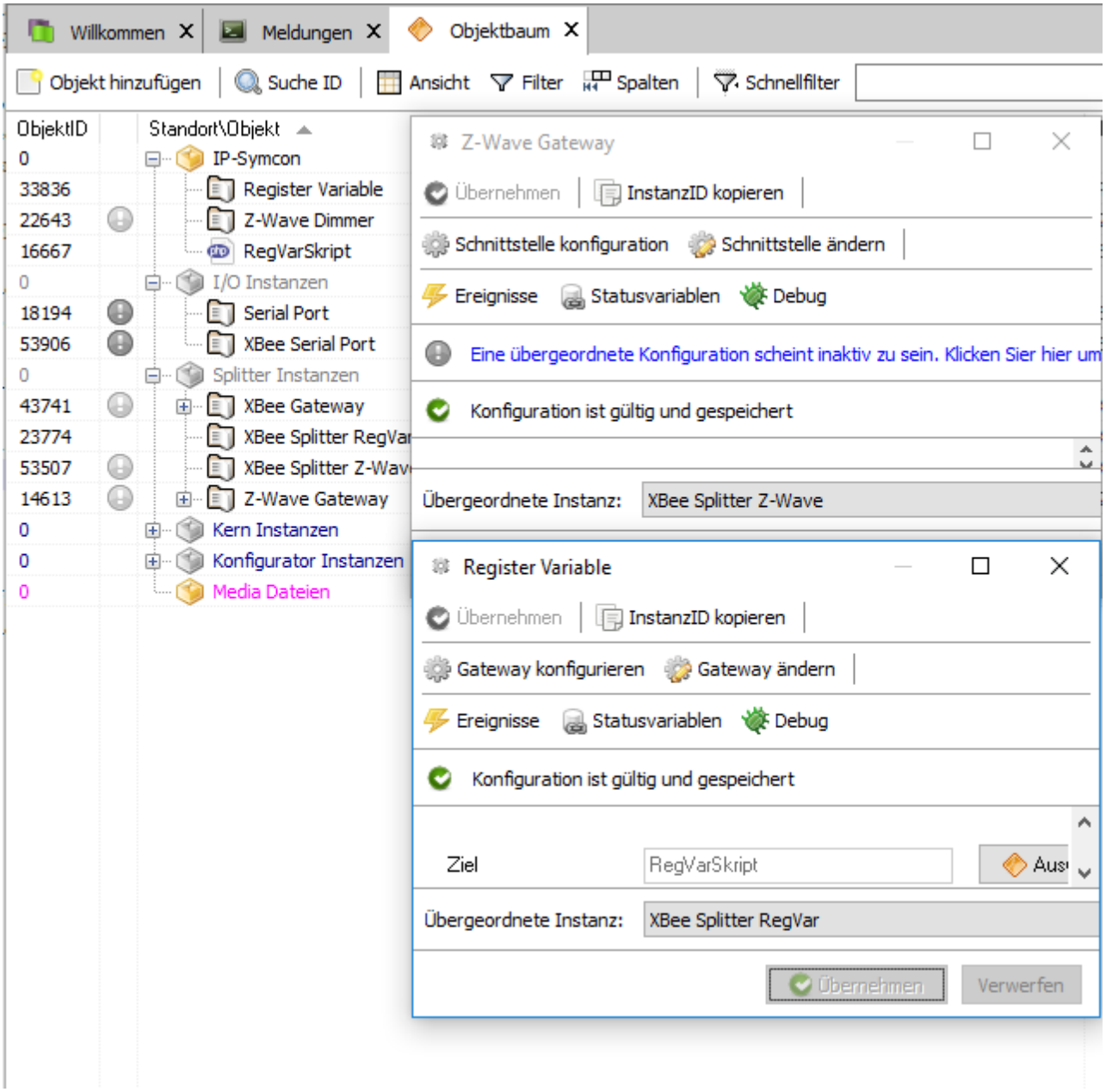
### DATENFLUSS

Zum besseren Verständnis kann man hier die abstrakte Darstellung des Datenflusses sehen.




## BEISPIEL OBJEKTBAUM

In diesem Beispiel sind an einem Coordinator ein Z-Wave Gateway und eine Register Variable angeschlossen.
XBee kümmert sich mithilfe der GeräteID darum, dass sowohl die Daten von der Register Variable im dazugehörigen Skript als Systemvariablen zur Verfügung gestellt werden, als auch die Z-Wave Geräte Daten via Z-Wave Gateway (übergeordnete Instanz: XBee Splitter Z-Wave)aktualisiert werden.




Daraus ergibt sich in diesem Beispiel folgende Topologie:

```
Serial Port -> XBee Gateway -> XBee Splitter Z-Wave -> Z-Wave
Gateway -> Z-Wave Dimmer
|
-> XBee Splitter RegVar -> Register Variable -> RegVarSkript
(Systemvariable)
```

# XBee_SendBuffer

```
boolean XBee_SendBuffer (int $InstanzID, int $ZielGerät, string
$Buffer)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des abzufragenden Geräts |
| ZielGerät | ID des ZielGeräts |
| Buffer | Datenbuffer der versendet werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet den DatenString **Buffer** an das Zielgerät **ZielGerät** und das XBee-Gateway mit der ID **InstanzID**.

## BEISPIEL

```
XBee_SendBuffer(12345, 2, "Beliebiger DatenString");
```

# XBee_SendCommand

```
boolean XBee_SendCommand (int $InstanzID, string $Befehl)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu verändernden Geräts |
| Befehl | Befehl der gesendet werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Sendet den Befehl **Befehl** an den XBee-Splitter mit der ID **InstanzID**.

## BEISPIEL

```
XBee_SendCommand(12345, "Beliebiger CommandString");
```

# xComfort

Eaton xComfort ist ein bidirektionales funkbasierendes System. Um eine Einrichtung in Symcon zu ermöglichen wird die Konfigurationsdatei der Installation oder die Programmierschnittstelle (CRSZ-00/01) benötigt. Die Verbindung zu Symcon wird mit der Kommunikationsschnittstelle CKOZ-00/03 oder einem Eaton USB- bzw. LAN-Gateway hergestellt.

Folgende Geräte werden von Symcon unterstützt:
Unterstützte Geräte

## INSTALLATION

Zur Anbindung an Symcon nutzt Eaton (ehemals Moeller) die **xComfort Kommunikations-Schnittstelle (CKOZ-00/03)**.
Zur Programmierung des Systems im Eaton Komfort-Modus wird die Eaton **xComfort Programmier-Schnittstelle (CRSZ-00/01)** genutzt.
Alternativ ist auch der Anschluss eines Eaton USB-Gateways oder eines LAN-Gateways zur Anbindung an Symcon möglich.
Durch das Eaton xComfort Funksystem lässt sich die Überwachung von Fenstern, Klima- und Beleuchtungsanlagen sowie viele andere Gebäudefunktionen automatisiert steuern. Das System lässt sich nachträglich installieren, ohne dass neue Leitungen verlegt werden müssen. Es handelt sich dabei um ein bidirektionales System, d.h. alle Sensoren und Aktoren (Schalter) können untereinander kommunizieren, da sie je eine Sende- und eine Empfangseinheit enthalten.

## VIDEO-TUTORIAL ZUR EINRICHTUNG

## ANSCHLUSS

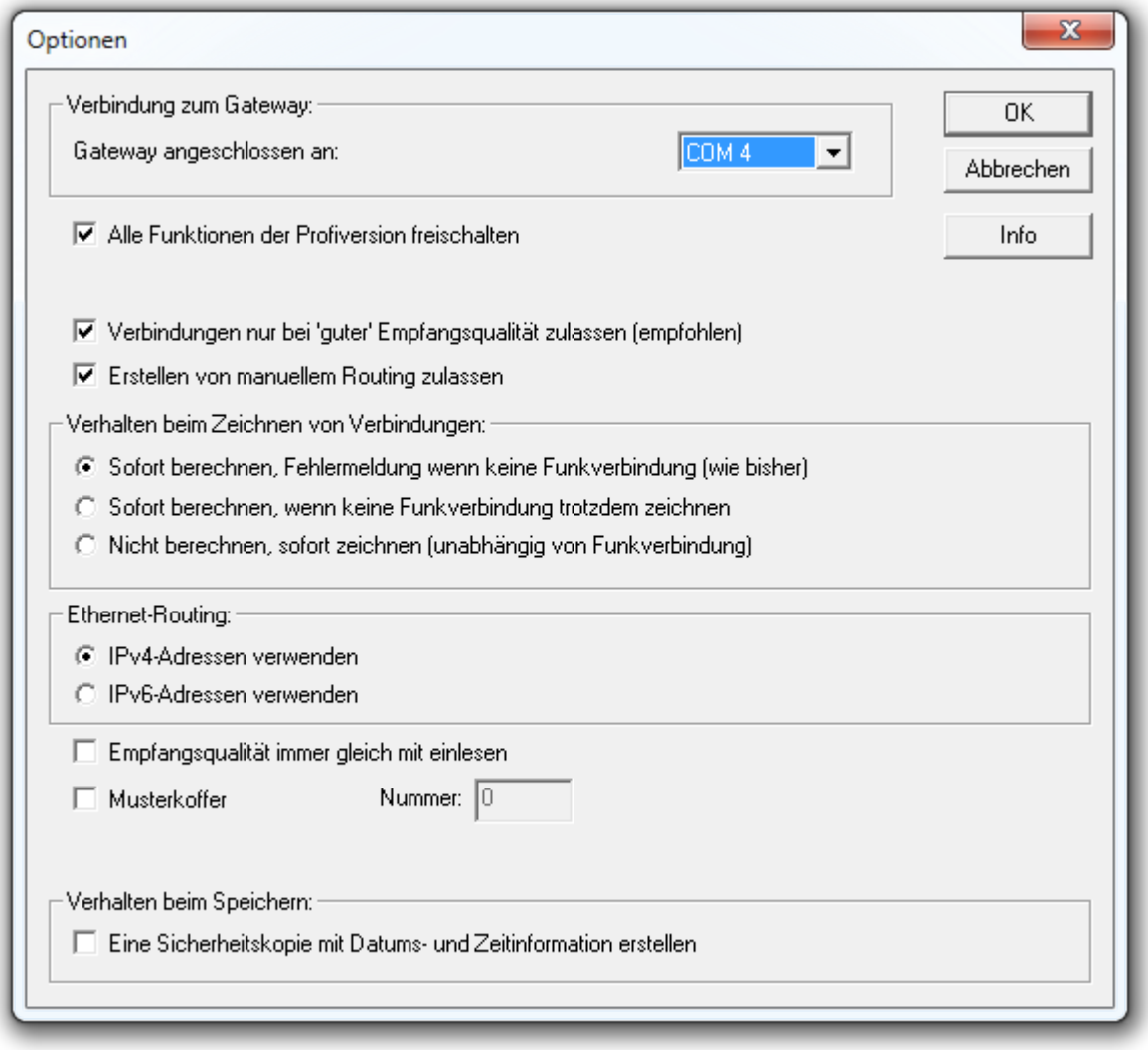
Um das Eaton Xcomfort System mit Symcon verbinden zu können, wird ein entsprechendes Gateway benötigt. In diesem Beispiel ist dies die Kommunikations-Schnittstelle CKOZ-00/03. Die Programmier-Schnittstelle CRZ-00/01 für den Comfort-Modus ist nicht dazu geeignet, wird aber ebenfalls benötigt, da diese zur Programmierung des Systems im Komfort-Modus von Eaton gebraucht wird.

Bei Benutzung des RS232-Anschlusses ist zu beachten, dass zusätzlich Strom über den USB-Port zugeführt werden muss. Dies ist über einen passenden Adapter (230V->USB) möglich.

Die Programmier-Schnittstelle CRZ-00/01 kann ebenfalls über die USB-Schnittstelle (optionales Kabel: RS232->USB) oder über den RS232-Anschluss angeschlossen werden.

## INSTALLATION

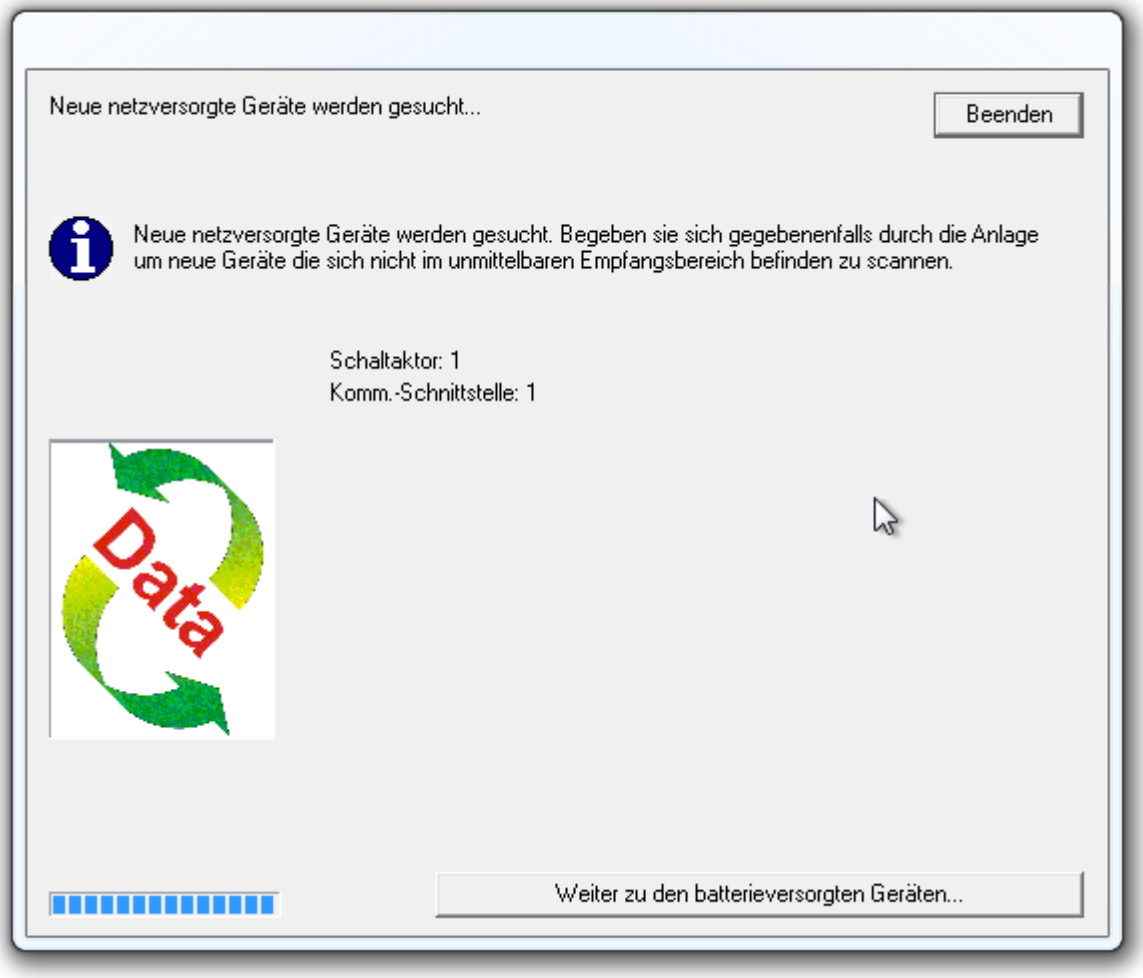
Nach dem korrekten Anschluss des/der Geräts/-e, muss "Systemsteuerung -> Hardware und Sounds -> Geräte und Drucker" geöffnet werden. Dort muss nach "Prolific USB-to-Serial Comm Port" geguckt werden. In der Klammer dahinter kann der genutzte Port (z.B.: COM4) gefunden werden, dieser sollte notiert oder gemerkt werden.
Es muss die Eaton Firmware / MRF Software heruntergeladen und installiert werden.
Nach dem Start der Software muss auf “Bearbeiten -> Optionen” geklickt, der korrekte Port (z.B. COM4) ausgewählt und die passenden Optionen aktiviert werden.




Im oberen Reiter kann nun auf “Gateway -> Einlesen” geklickt werden. Das System kann nun im Komfort-Modus programmiert werden.

## HINZUFÜGEN SEKUNDÄRER GERÄTE

Mit der Eaton Firmware können nun sekundäre Geräte der Firma Eaton (z.B. Zwischenstecker, Raumcontroller, Fernbedienung, usw.) in das System eingebunden werden.
Dies kann im oberen Reiter via “Aktionen -> 2x Einlesen” getan werden.
Jetzt wird zuerst nach den netzversorgten Geräten gesucht. Ggf. muss man sich zum Scannen in die Nähe der sekundären Geräte begeben, falls sich diese nicht im unmittelbaren Empfangsbereich befinden.

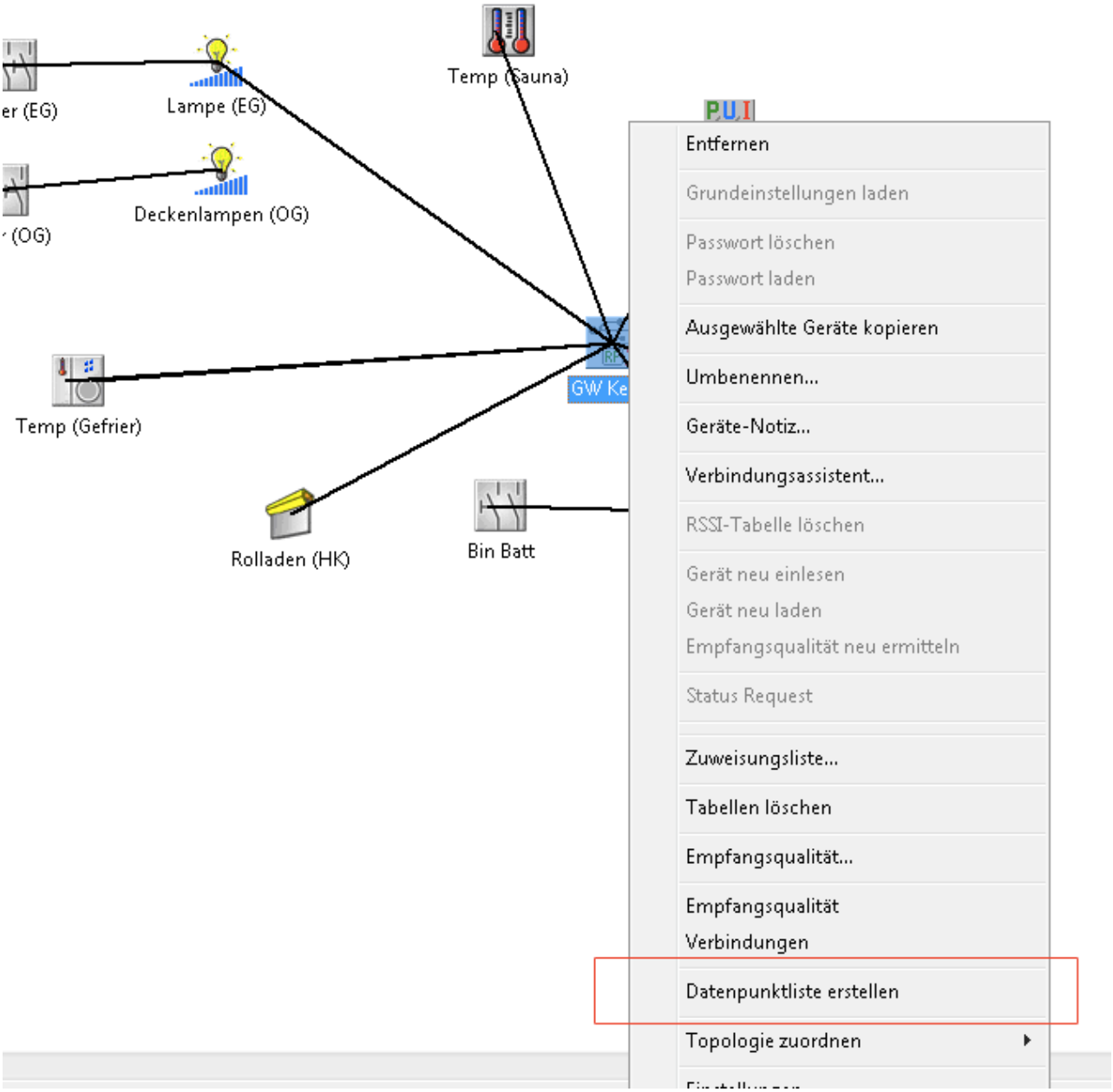



Wenn alle netzversorgten Geräte eingelesen worden sind, muss auf "Weiter zu den batterieversorgten Geräten" geklickt werden. Es müssen nun die batterieversorgten Geräte betätigt werden, damit eine Verbindung zum Gateway hergestellt werden kann. Nachdem dies geschehen ist, muss auf "Beenden" geklickt werden. Jetzt wird eine Passworteingabe verlangt. Es sollte als **Passwort "0000"** eingegeben und bestätigt werden.
Im oberen Reiter "Aktionen" sollte danach auf "Empfangsqualität einlesen..." und dann nochmals auf “Einlesen” geklickt werden.

Sollten noch Geräte mit Fragezeichen angezeigt werden, aber alle sekundären Geräte bereits einem anderen Symbol zugeordnet sein, so können diese gelöscht werden. Dies kann via rechter Maustaste auf das zu löschende Symbol und dann mit der linken auf "Entfernen" getan werden. Der folgende Dialog muss mit "Ja" und dann "Nein" bestätigt werden.
Auch Umbenennungen sind mit einem Klick auf die rechte Maustaste und danach auf "Umbenennen" möglich.

Nun werden die Geräte in der Eaton Firmware angezeigt. Jetzt muss auf das Bleistift-Symbol geklickt und die Geräte via virtuellem Bleistift mit dem Gateway verbunden werden. Danach muss unter "Aktionen" auf "Änderungen laden" geklickt werden. Damit Symcon später weiß, was beispielsweise auf Datenpunkt 1 oder auf Datenpunkt 2 ist, muss mit der rechten Maustaste auf das Gateway und nachfolgend mit der linken auf "Datenpunktliste erstellen" geklickt werden. Diese muss gespeichert werden. Zum Abschluss muss das gesamte Projekt gespeichert werden. Die Eaton Firmware wird nicht

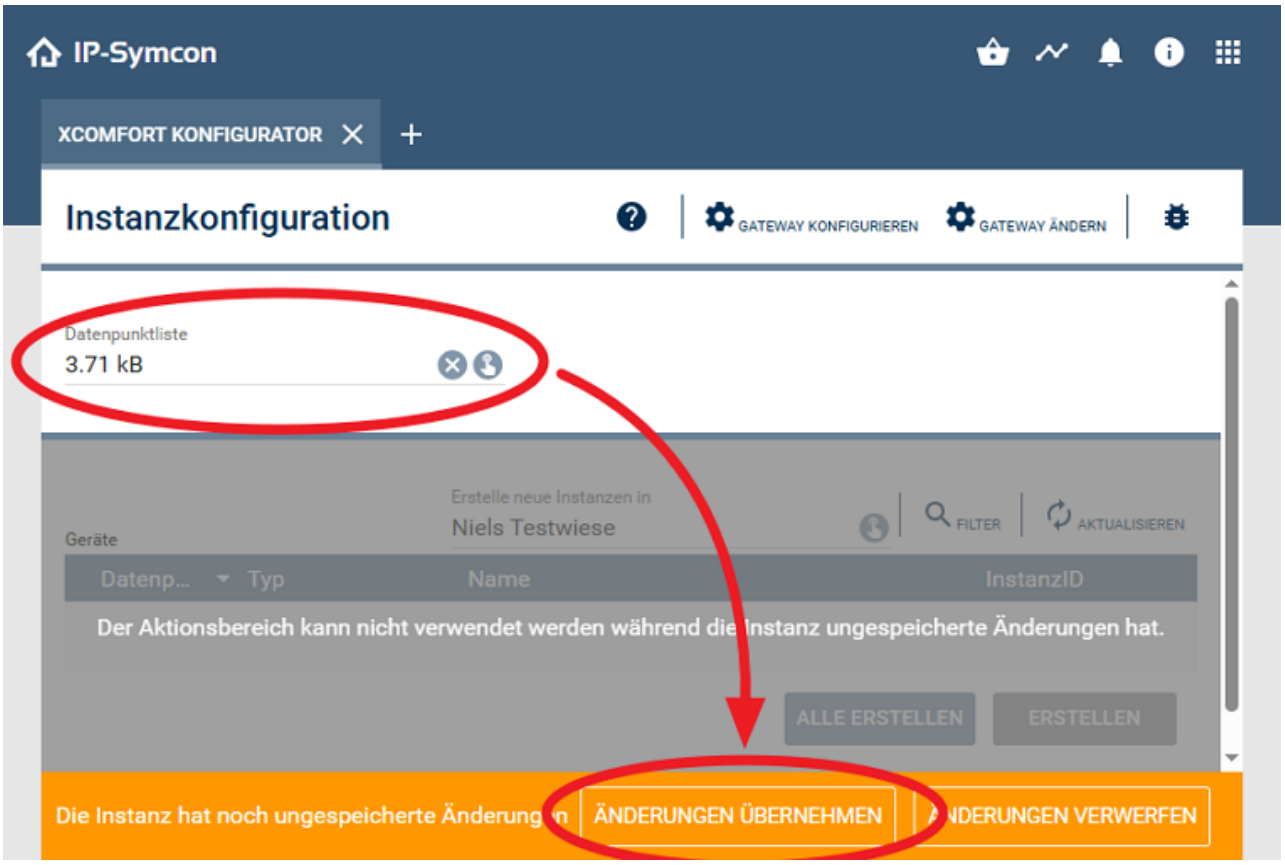
weiter benötigt.

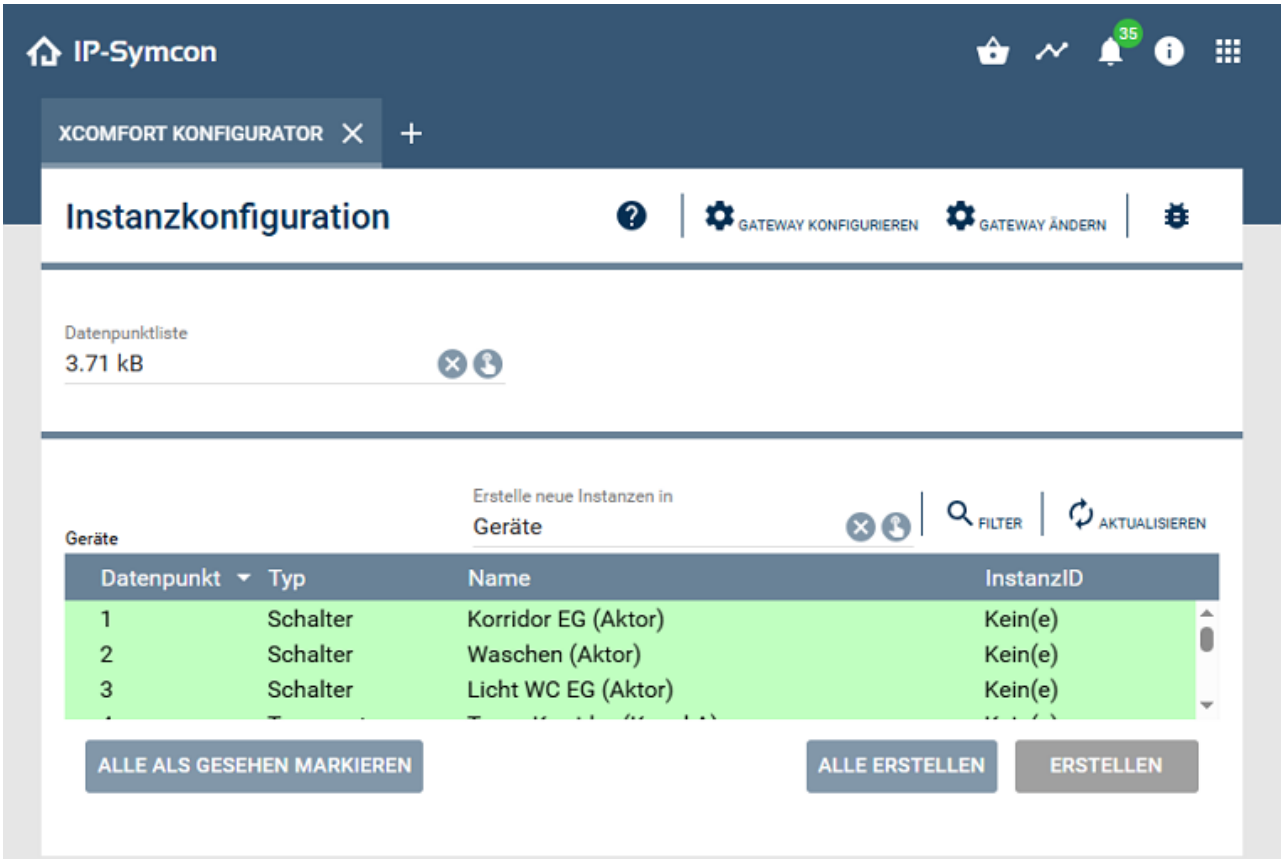



## EINBINDUNG IN SYMCON

In der Verwaltungskonsole kann im Objektbaum über "Objekt hinzufügen -> Instanz" ein "xComfort Konfigurator" erstellt werden. Nach Wahl des Modus passend zum verwendeten Gateway, wird die Verbindung zum Gateway konfiguriert.
Hier muss je nach Modus die IP-Adresse eingegeben, der Comport ausgewählt oder das passende USB-Gerät ausgewählt werden sowie der I/O aktiviert werden. Schließlich wird der Dialog noch bestätigt und der Konfigurator öffnet sich. Über die Dateiauswahl "Datenpunktliste" wird anschließend die zuvor gespeicherte Datenpunktlisten hochgeladen und nachdem die Änderungen übernommen wurden, können die Geräte erstellt werden.

Für den Empfang von routingfähigen Rückmeldungen, muss die Datenpunktliste im Konfigurator geladen worden sein. Nur so ist eine Seriennummerzuweisung zu einer Datenpunktnummer möglich.



## TIPPS & TRICKS

Neue Versionen der Firmware/MRF Software kann hier heruntergeladen werden:
Eaton Downloads
Das Gateway ist mit allen gängigen USB->LAN Umwandlern kompatibel (z.B. von Silex, Lantronix).
Statusvariablen werden vom Eaton Xcomfort System nicht geroutet. Das bedeutet, dass das Gateway an einem Ort positioniert sein muss, dass alle relevanten Geräte in Reichweite sind.
Es können in Symcon mehrere Gateways gleichzeitig verwendet und über Skripte auch Befehle weitergeleitet werden. So ist es üblich, pro Etage ein Gateway zu platzieren.
Erst ab der RF Version **9.2** (ca. Juli/2008) werden die auf der Protokollebene versendeten "Acknowledges" vom USB-Gateway ausgewertet. In den vorherigen Versionen ist im System nicht erkennbar, ob ein Befehl erfolgreich an ein Gerät übermittelt worden ist. Die RF Version ist ein Hardware-Upgrade und darf nicht mit der FW Version verwechselt werden, die per Software aktualisiert werden kann. (Betrifft nur PC-Manager CKOZ-00/03)
Das neue Thermostat benötigt ein ECI LAN Gateway oder ein USB-Gateway mit mindestens Firmware 2.6 und RF Version 9.2. Falls das Gerät zu alt sein sollte, kann im Meldungsfenster folgende Fehlermeldungen auftreten: "Invalid RF-firmware revision"
Der Port für die ClientSocket-Konfiguration vom ECI ist 7153

# Geräteliste

## UNTERSTÜTZTE SCHNITTSTELLEN

| PRODUKT | BESCHREIBUNG |
|---|---|
| CCIA-01/01 | Eaton ECI WLAN Interface |
| CCIA-02/01 | Eaton ECI LAN Interface |
| CCIA-03/01 | Eaton ECI LAN Interface Power over Ethernet |
| CKOZ-00/03 | Kommunikationsschnittstelle (USB/RS232) |
| CKOZ-00/11 | Kommunikationsschnittstelle (USB/RS232) |
| CKOZ-00/14 | USB Kommunikations-Stick |

## UNTERSTÜTZTE KOMPONENTEN

| PRODUKT | BESCHREIBUNG |
|---|---|
| CAAE* | Analogaktor |
| CAEE* | Analogeingang |
| CBEU* | Binäreingang |
| CBMA* | Bewegungsmelder |
| CDAE* | Dimmaktor |
| CDAE* | Schaltaktor |
| CDAP* | Schalt-/Dimmaktor |
| CDAU* | Dimmaktor |
| CDWA*<br>(ab Version 5.4) | Fenster-/Türkontakt |
| CEMU* | Energiemesssensor |
| CHAP* | Heizungsaktor |
| CHAU* | Heizungsaktor |
| CHAZ* | Heizungsaktor für Elektropanele |
| CHAZ* | Heizungsaktor 12-fach |
| CHSZ* | Fernbedienungen |
| CHVZ* | Heizkörperventil |
| CIZE* | Impulszähleingang |
| CJAU* | Jalousieaktor |
| CJAE* | Jalousieaktor Stecktyp |
| CRCA* | Raumcontroller |
| CRMA* | Room-Manager |
| CROU* | Router |
| CSAE* | Schaltaktor |
| CSAP* | Schaltaktor |
| CSAU* | Schaltaktor |
| CSEZ* | Fensterkontakt (über Binäreingang) |
| CSEZ* | Feuchtesensor (über Analogeingang) |
| CSEZ* | Helligkeitssensor (über Analogeingang) |
| CSEZ* | Luftqualitätssensor (über Analogeingang) |
| CSEZ* | Präsenzmelder (über Binäreingang) |
| CSEZ* | Rauchmelder (über Binäreingang) |
| CSEZ* | Temperatursensor |
| CSEZ* | Wasser Leckagensensor (über Binärsensor) |
| CSGZ* | Signalgeber (über Schaltaktor) |
| CTAA* | Taster |
| CTEU* | Temperatureingang |

# MXC_DimBrighter

```
boolean MXC_DimBrighter (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Beginnt den Dimmvorgang für Gerät mit der ID **InstanzID** auf einer höheren Helligkeitsstufe. Der Vorgang kann mit dem Befehl MXC_DimStop beendet werden. Wird die maximale Stufe erreicht, stoppt der Vorgang automatisch. Die Statusmeldung wird von dem Gerät gesendet, sobald der Vorgang beendet wurde. Es werden keine Meldungen zwischendurch versandt.



Die Geschwindigkeit und Stärke der Dimmung muss vorher über die xComfort Programmierschnittstelle festgelegt werden. IP-Symcon ruft nur die Dimmung auf.

## BEISPIEL

```
MXC_DimBrigther(12345); //Heller dimmen
IPS_Sleep(1000): //1sek warten
MXC_DimStop(12345);
```

# MXC_DimDarker

```
boolean MXC_DimDarker (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Beginnt den Dimmvorgang für Gerät mit der ID InstanzID auf einer niedrigeren Helligkeitsstufe. Der Vorgang kann mit dem Befehl MXC_DimStop beendet werden. Wird die maximale Stufe erreicht, stoppt der Vorgang automatisch. Die Statusmeldung wird von dem Gerät gesendet, sobald der Vorgang beendet wurde. Es werden keine Meldungen zwischendurch versandt.



Die Geschwindigkeit und Stärke der Dimmung muss vorher über die xComfort Programmierschnittstelle festgelegt werden. IP-Symcon ruft nur die Dimmung auf.

## BEISPIEL

```
MXC_DimDarker(12345); //Dunkler dimmen
IPS_Sleep(1000);
MXC_DimStop(12345);
```

# MXC_DimSet

```
boolean MXC_DimSet (int $InstanzID, int $Intensität)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Intensität | Wert von 0-100 (in %) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Dimmt das Gerät mit der ID **InstanzID** auf Wert **Intensität**

## BEISPIEL

```
MXC_DimSet(12345, 50); //Auf 50% dimmen
```

# MXC_DimStop

```
boolean MXC_DimStop (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schaltet das Gerät mit der ID **InstanzID** auf Wert **Status**

## BEISPIEL

```
MXC_DimStop(12345, true); //Gerät einschalten
```

# MXC_RequestStatus

```
boolean MXC_RequestStatus (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Fordert ein erneutes Senden der Statusmeldungen für Gerät mit der ID **InstanzID** an.



Da Statusmeldungen im xComfort System nicht geroutet werden, kann es sein, dass der Befehl erfolgreich ausgeführt, aber kein neuer Status empfangen wird. Dieser Befehlt fragt den Status nicht ab, sondern fordert nur einen neue Statusmeldung an.



**Diese Funktion belastet den Funkverkehr erheblich.**
Sie sollte ausschließlich verwendet werden, um kritische Gerät zum IP-Symcon-Start zu aktualisieren. Ein erneutes Abfragen zur Laufzeit wird nicht benötigt, da xComfort alle Statusänderungen direkt an IP-Symcon weiterleitet.

## BEISPIEL

```
MXC_RequestStatus(12345); //Statusmeldung anfordern
```

# MXC_SendBoolean

```
boolean MXC_SendBoolean (int $InstanzID, boolean $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | True für An, False für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet einen booleanischen Wert an die Instanz mit der **InstanzID**, welche an einen Room-Manager angeschlossen ist.

## BEISPIEL

```
MXC_SendBoolean(12345, true);
```

# MXC_SendFloat

```
boolean MXC_SendFloat (int $InstanzID, float $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | Zahlenwert mit Nachkommastelle |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet eine Gleitkommazahl an die Instanz mit der **InstanzID**, welche an einen Room-Manager angeschlossen ist.

## BEISPIEL

```
MXC_SendFloat(12345, 35.34); //35.34 als Wert senden
```

# MXC_SendInteger

```
boolean MXC_SendInteger (int $InstanzID, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Wert | Zahlenwert ohne Nachkommastellen |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet einen booleanischen Wert an die Instanz mit der **InstanzID**, welche an einen Room-Manager angeschlossen ist.

## BEISPIEL

```
MXC_SendInteger(12345, 40); //Wert 40 senden
```

# MXC_SetTemperature

```
boolean MXC_SetTemperature (int $InstanzID, float $Temperatur)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Temperatur | Temperatur in °C |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt die neue Soll-Temperatur des Stellventils mit der **InstanzID** InstanzID auf die neue **Temperatur**.

## BEISPIEL

```
MXC_SetTemperature(12345, 21.5); //Auf 21.5°C stellen
```

# MXC_ShutterMoveDown

```
boolean MXC_ShutterMoveDown (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Bewegt den Rollladen mit der ID **InstanzID** nach unten, bis MXC_ShutterStop ausgeführt wird oder die Endposition erreicht wird.



Beachten Sie, dass die Laufzeitlimitierung über das MRF Tool von Moeller programmiert wird. IP-Symcon kann diese Limitierung nicht beeinflussen.

## BEISPIEL

```
MXC_ShutterMoveDown(12345); //Nach unten bewegen
```

# MXC_ShutterMoveUp

```
boolean MXC_ShutterMoveUp (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Bewegt den Rollladen mit der ID **InstanzID** nach oben, bis MXC_ShutterStop ausgeführt wird oder die Endposition erreicht wird.



Beachten Sie, dass die Laufzeitlimitierung über das MRF Tool von Moeller programmiert wird. IP-Symcon kann diese Limitierung nicht beeinflussen.

## BEISPIEL

```
MXC_ShutterMoveUp(12345); //Nach oben bewegen
```

# MXC_ShutterStepDown

```
boolean MXC_ShutterStepDown (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Bewegt den Rollladen mit der ID **InstanzID** einen Schritt nach unten.

## BEISPIEL

```
MXC_ShutterStepDown(12345); //Einen Schritt nach unten bewegen
```

# MXC_ShutterStepUp

```
boolean MXC_ShutterStepUp (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Bewegt den Rollladen mit der ID **InstanzID** einen Schritt nach oben.

## BEISPIEL

```
MXC_ShutterStepUp(12345); //Einen Schritt nach unten bewegen
```

# MXC_ShutterStop

```
boolean MXC_ShutterStop (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Stoppt eine aktuelle Bewegung des Rollladens.

## BEISPIEL

```
MXC_ShutterStop(12345); //Anhalten
```

# MXC_SwitchMode

```
boolean MXC_SwitchMode (int $InstanzID, boolean $Status)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Status | TRUE für An, FALSE für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Schaltet das Gerät mit der ID InstanzID auf Wert Status

## BEISPIEL

```
MXC_SwitchMode(12345, true); //Gerät einschalten
```

# Z-Wave

Z-Wave ist ein funkgebundenes System, welches von mehreren Herstellern genutzt wird. Zur Verbindung mit IP-Symcon wird entweder ein LAN oder USB Gateway benötigt.

Folgende Geräte werden von IP-Symcon unterstützt:
Unterstützte Geräte

## EINBINDUNG IN IP-SYMCON

Bei Verwendung des LAN-Gateways kann dies über die Gerätesuche eingebunden werden. Hierfür muss als System "Z-Wave Discovery" gewählt werden. Die Discovery-Instanz bietet dann die Erstellung eines Z-Wave Konfigurators an. Nachdem der Konfigurator erstellt wurde, können dort die einzelnen Geräte wie unten beschrieben eingebunden werden.
Soll ein Z-Wave Konfigurator manuell erstellt werden kann im "Instanz hinzufügen" Dialog nach "Z-Wave Configurator" gesucht werden.
Nach der Einrichtung eines Z-Wave Konfigurators ist der nächste Schritt das Anlernen von Geräten.

Es ist darauf zu achten, dass das passende Gateway ausgewählt, der passende Client Socket (Standard-Port: 5000) oder Serial Port eingerichtet und konfiguriert ist.

Dies geschieht innerhalb des Konfigurators durch die "Gerät hinzufügen"-Taste und anschließend mindestens drei Mal schnell hintereinander drücken der Anlerntaste des Geräts.
IP-Symcon findet automatisch das Gerät und zeigt diese Grün in der Liste an.
Wenn dieses ausgewählt und der "Erstellen"-Button gedrückt wird, erstellt IP-Symcon eine Instanz des Gerätes, welches dann konfiguriert werden kann.
Das Konfigurationsfenster der Instanz lässt sich per "Konfigurieren" im Z-Wave Konfigurator oder im Objektbaum per Doppelklick auf die Instanz aufrufen.
Die Eigenschaften (Command-Klassen siehe Geräteliste ) des Geräts lassen sich nach dem "Laden" unter "Zeigen" anzeigen. Statusvariablen werden automatisch hinzugefügt und können im Objektbaum überprüft werden.

Um IP-Symcon den Zugriff auf bestimmte Funktionen (z.B. Z-Wave Gerätedatenbank) von Außerhalb zu gewähren ist es gegebenenfalls nötig diese in der Firewall einzurichten. Weitere Informationen können unter Firewall gefunden werden.

## VIDEO-TUTORIAL ZUR EINRICHTUNG

# SPEZIALGERÄTE – EINRICHTUNG

Link zum Forenbeitrag

# Geräteliste

## UNTERSTÜTZTE GATEWAYS

| PRODUKT | BESCHREIBUNG |
|---|---|
| Symcon Z-Wave-Plus LAN Gateway \| Produktinformation | IP Gateway |
| 300ZW-US-W | TrickleStar USB Stick |
| AEO_USB | USB Stick von Aeon Labs |
| ZW090 | USB Z-Stick von Aeotec |
| RaZberry | Z-Wave Transceiver für Rasperrry PI |
| VIS_ZU1401 | Vision USB Stick |
| ZME_Serial | Serieller Adapterplatte |
| ZME_UZB1 | USB-Stick |
| ZME_ZSTICKC | Z-Wave.Me USB Stick |

## GERÄTE

Alle aufgelisteten Command-Klassen werden von IP-Symcon unterstützt. Zusätzlich bieten wir eine Liste mit Geräten, welche von uns oder unseren Kunden erfolgreich getestet wurden.
Auf Anfrage überprüfen wir gerne weitere, nicht aufgelistete, Geräte auf ihre Kompatibilität.

## UNTERSTÜTZTE COMMAND-KLASSEN

Jedes Z-Wave-Gerät besitzt verschiedene Command-Klassen. Diese Klassen geben an, welche Funktionen von dem jeweiligen Gerät unterstützt werden. In IP-Symcon stehen alle Funktionen mit den folgenden gelisteten Command-Klassen zur Verfügung.

| COMMAND CLASS | CLASS VERSION | IP-SYMCON VERSION |
|---|---|---|
| ALARM | ---------- | Abgelöst durch NOTIFICATION |
| ASSOCIATION | 2 | |
| ASSOCIATION_GRP_INFO | 2 | ab 5.1 |
| BASIC | 2 | |
| BASIC_WINDOW_COVERING | 1 | |
| BATTERY | 1 | |
| CENTRAL_SCENE | 1 | ab 4.1 |
| CLOCK | 1 | |
| COLOR | 1 | ab 4.1 |
| CONFIGURATION | 1 | |
| CRC_16 | 1 | |
| DOOR_LOCK | 2 | ab 4.1 |
| LOCK | 2 | |
| MANUFACTURER_SPECIFIC | 2 | |
| METER | 5 | |
| MULTI_CHANNEL | 4 | |
| MULTI_CHANNEL_ASSOCIATION | 3 | ab 5.1 |
| MULTI_CMD | 1 | |
| MULTI_INSTANCE | ---------- | Abgelöst durch MULTI_CHANNEL |
| NOTIFICATION | 8 | ab 4.3 |
| PROTECTION | 2 | |
| SCENE_ACTIVATION | 1 | ab 4.1 |
| SECURITY | 1 | ab 4.1 |
| SENSOR_ALARM | 1 | |
| SENSOR_BINARY | 2 | |
| SENSOR_MULTILEVEL | 11 | |
| SWITCH_ALL | 1 | ab 4.1 |
| SWITCH_BINARY | 2 | |
| SWITCH_MULTILEVEL | 4 | |
| SWITCH_SHUTTER | 1 | |
| THERMOSTAT_FAN_MODE | 1 | |
| THERMOSTAT_FAN_STATE | 1 | |
| THERMOSTAT_MODE | 1 | |
| THERMOSTAT_OPERATING_STATE | 1 | |
| THERMOSTAT_SETPOINT | 1 | |
| TIME | 2 | ab 4.1 |
| USER_CODE | 1 | ab 4.1 |
| Unknown | 1 | |

## GETESTETE KOMPONENTEN

nach Alphabet sortiert

### ACT

| PRODUKT | BESCHREIBUNG |
|---|---|
| ZRM230 | Wall Switch/Transmitter (2-gang) |
| ZTM230 | RF Wall Transmitter (2-gang) |

### AEON LABS

| PRODUKT | BESCHREIBUNG |
|---|---|
| AEO_DSC20-EU | Door/Window Sensor |
| AEO_DSB54 | Recessed Door Sensor |
| AEO_DSD31 | Siren |
| AEO_EXTENDER | Extender |
| AEO-HEM2, HEM3 | Clamp Meter 40 |
| AEO_KFOB | Key Fob |
| AEO-MEI | Micro Dimmer |
| AEO-MES | Micro Switch |
| AEO-MSEI | Micro Smart Energy Dimmer |
| AEO-MSES | Micro Smart Energy Switch |
| AEO_MULTISENS | Multisensor |
| AEO-SES3 | Smart Energy Switch 3 |
| AEO_ZW088 | Key Fob Gen 5 |
| AEO_ZW089 | recessed Doorsensor |
| DSD31 | Outlet Plugable Siren |

### AEOTEC

| PRODUKT | BESCHREIBUNG |
|---|---|
| ZW078 | Heavy Duty Smart Switch |
| ZW080 | Siren 5 |
| ZW095 | Home Energy Meter |
| ZW096 | Smart Switch 6 |
| ZW100 | MultiSensor 6 |
| ZW111 | Nano Dimmer |
| ZW112 | Door/Window Sensor 6 |
| ZW116 | Nano Switch with Power Metering |
| ZW117 | Range Extender 6 |
| ZW120 | Door/Window Sensor Gen5 with Tamper Sensor |
| ZW121 | LED Strip |
| ZW122 | Water Sensor 6 |
| ZW129 | WallMote |
| ZW130 | WallMote Quad |
| ZW132 | Dual Nano Switch with Power Metering |
| ZW139 | Nano Switch |
| ZW140 | Dual Nano Switch |
| ZW141 | Nano Shutter |
| ZW158 | WallSwipe |
| ZW160 | Water Sensor 6 Dock |
| ZW162 | Doorbell 6 |
| ZW166 | Button |
| ZW164 | Siren 6 |
| ZW175 | Smart Switch 7 |
| ZW189 | Range Extender 7 |
| ZWA001 | LED Bulb 6 Multi-White |
| ZWA002 | LED Bulb 6 Multi-Colour |
| ZWA003 | NanoMote Quad |
| ZWA004 | NanoMote One |
| ZWA005 | TriSensor |
| ZWA006 | Smart Boost Timer Switch |
| ZWA008 | Door/Window Sensor 7 |

| PRODUKT | BESCHREIBUNG |
|---|---|
| DAN_LC-13 (014G0013) | Thermostat Radiator Valve |
| DAN_LIVC_RAK (014G0012) | Thermostat Radiator Valve |

## EVERSPRING

| PRODUKT | BESCHREIBUNG |
|---|---|
| EVR_AD142 | Dimmer Plug |
| EVR_AN145 | Screw-In Lamp Holder |
| EVR_AN157 | On/Off Plug |
| EVR_AN158 | On/Off Plug with power meter |
| EVR_HSM02 | Door/Window Detector |
| EVR_HSP02 | Motion Detector |
| EVR_SE812 | Siren |
| EVR_SF812 | Smoke Detector with Alarm |
| EVR_SP103 | Motion Sensor |
| EVR_SP814 | Motion Detector |
| EVR_ST812 | Flood Sensor |
| EVR_ST814 | Temperature & Humidity Sensor |

## FIBARO

| PRODUKT | BESCHREIBUNG |
|---|---|
| FGBS-001 | Universal Binary Sensor |
| FGD-211 | Universal Dimmer |
| FGFS-101 | Flood Sensor |
| FGK 101-107 Door/Window Sensor 2.5 | Door/Window Sensor |
| FGK-001 | Door/Window Sensor |
| FGMS-001 | Motion Sensor |
| FGRGB-101 | RGBW Controller |
| FGRGBWM-441 | RGBW Controller |
| FGRM-222 | Roller Shutter 2 |
| FGS-211 | 1x3KW Relay |
| FGS-221 | 2x1.5KW Relay |
| FGSS-001 | Smoke Sensor |
| FGSS-101 | Smoke Sensor |
| FGWPE/F-101 | Wall Plug |

## NORTHQ

| PRODUKT | BESCHREIBUNG |
|---|---|
| Energy Guard Electrical Meter | Meter (unvollständige Unterstützung) |
| Energy Guard Gas Meter | Gas Meter (unvollständige Unterstützung) |

## PHILIO

| PRODUKT | BESCHREIBUNG |
|---|---|
| In Wall Dual Relay (1 Way) | 230VAC powered static controller with binary switch capability |
| In Wall Dual Relay(1 Way) | In-wall switch module |
| In Wall Dual Relay(1 Way) | In-wall dual relay switch module |
| In Wall Single Relay(1 way) | In-wall switch module |
| PAN04 | 2 x 1.5Kw Relay with Power Measurement |
| PAN06 | 2 x 1.5Kw Relay |
| PAN08 | In-Wall Roller Shutter Controller |
| PAN11-1 | Plugin Switch with meter functionality PAN11-1 |
| PAT02-1A | Flood multisensor (flood, temperature and humidity) |
| PSE02 | Wireless multiple sound siren |
| PSG01 | Smoke Sensor |
| PSM01 | 3-in-1 Sensor (Door, Light, Temp) |
| PSM02 | 4-in-1 Sensor (PIR, Door, Light, Temp) |
| PSM02-1 | PIR, door/window, temperature and illumination Sensor |
| PSP01 | 3-in-1 Sensor (PIR, Light, Temp) |
| PSR03 | Remote |
| PST02 | Slim Multisensor |

## POLYCONTROL

| PRODUKT | BESCHREIBUNG |
|---|---|
| Danalock Circle | Z-Wave Door Lock |
| Danalock Square | Z-Wave Door Lock |

## POPP

| PRODUKT | BESCHREIBUNG |
|---|---|
| POP_004308 | Smoke Detector and Temp Sensor |
| POP_123610 | Plug-in Switch |
| POP_123658 | Plug-in Switch with Meter |
| POP_123665-58 | Wall Plug Switch/Meter |
| POP_123xxx | On/Off Plug |
| POP_123xxx | Dimmer Plug |

## VISION

| PRODUKT | BESCHREIBUNG |
|---|---|
| PIR Sensor | PIR Motion Sensor |
| Plugin ON/OFF | Light Relay |
| Power Monitor | Power Monitor |
| VIS_ZD2102EU 2013 | Door Window Sensor with external digital input |
| VIS_ZD3102EU 2013 | Motion detector with Temperature sensor and temper protection |
| VIS_ZG8101 | Garage Door Detector |
| VIS_ZL7432 | In-Wall Dual Relay Switch |
| VIS_ZM1601 | Siren (Battery) |
| VIS_ZM1602 | Siren (Mains) |
| VIS_ZS5102 | Shock & Vibration Sensor |
| VIS_ZS6101 | Smoke Detector |
| VIS_ZS6301 | CO Detector |

## ZIPATO

| PRODUKT | BESCHREIBUNG |
|---|---|
| Mini Keypad RFID/Z-Wave | Mini Keypad mit RFID Unterstützung |
| RGBW bulb | RGBW Bulb |

## Z-WAVE.ME

| PRODUKT | BESCHREIBUNG |
|---|---|
| Mini-Fernbedienung KFOB | Mini-Fernbedienung 4 Tasten Z-Wave KFOB, Version 2 Z-WAVE.ME |

## SONSTIGE

| HERSTELLER | PRODUKT | BESCHREIBUNG |
| --- | --- | --- |
| Express Control | EZMotion | Motion detector, luminosity and temperature sensor |



Es folgen die zur Zeit definitiv nicht unterstützten Geräte

## NICHT UNTERSTÜTZTE KOMPONENTEN

nach Alphabet sortiert

### MERTEN

| PRODUKT | BESCHREIBUNG |
| --- | --- |
| Alle | Fast alle Merten Produkte können nicht über den normalen Z-Wave Standard angesprochen werden |

# ZW_Basic

```
boolean ZW_Basic (int $InstanzID, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |
| Wert | 0..255 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schaltet die Basisfunktion vom angegeben Gerät mit der InstanzID **InstanzID**



Die Basisfunktion kann in den meisten Geräten über Parameter bestimmt werden. Bitte entnehmen Sie weitere Informationen der Anleitung ihres spezifischen Gerätes.

## BEISPIEL

```
ZW_Basic(12345, 255); //z.B. Anschalten
```

# ZW_ColorCW

Benötigt: IP-Symcon >= 5.4

```
boolean ZW_ColorCW (int $InstanzID, int $KaltWeiß)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |
| KaltWeiß | KaltWeiß Intensität von 0..255 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schaltet das Gerät mit der ID **InstanzID** im KaltWeiß Kanal auf den Wert **KaltWeiß**.

## BEISPIEL

```
// KaltWeiß auf 50% setzen
ZW_ColorCW(12345, 128);
```

# ZW_ColorRGB

Benötigt: IP-Symcon >= 5.4

```
boolean ZW_ColorRGB (int $InstanzID, int $Rot, int $Grün, int $Blau)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |
| Rot | Rot Intensität von 0..255 |
| Grün | Grün Intensität von 0..255 |
| Blau | Blau Intensität von 0..255 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schaltet das Gerät mit der ID **InstanzID** auf die Farbwertkombination **RGB**.

## BEISPIEL

```
// Dunkles Rot setzen
ZW_ColorRGB(12345, 255, 0, 0);
```

# ZW_ColorRGBWW

Benötigt: IP-Symcon >= 4.1

```
boolean ZW_ColorRGBWW (int $InstanzID, int $Rot, int $Grün, int
$Blau, int $WarmWeiß, int $KaltWeiß)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |
| Rot | Rot Intensität von 0..255 |
| Grün | Grün Intensität von 0..255 |
| Blau | Blau Intensität von 0..255 |
| WarmWeiß | WarmWeiß Intensität von 0..255 |
| KaltWeiß | KaltWeiß Intensität von 0..255 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schaltet das Gerät mit der ID **InstanzID** auf die Farbwertkombination **RGBWW**.



Der Warmweißanteil wirkt nur, wenn alle anderen Farben- und Kaltweißanteile auf 0 gesetzt werden.



Ab IP-Symcon 5.4 wird geprüft, ob das Gerät RGB, Warmweiß und Kaltweiß auch unterstützt

## BEISPIEL

```
// Dunkles Rot setzen
ZW_ColorRGBWW(12345, 255, 0, 0, 0, 50);

// Warmweiß setzen
```

```
ZW_ColorRGBWW(12345, 0, 0, 0, 255, 0);
```

# ZW_ColorWW

Benötigt: IP-Symcon >= 4.1

```
boolean ZW_ColorWW (int $InstanzID, int $WarmWeiß)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |
| WarmWeiß | WarmWeiß Intensität von 0..255 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schaltet das Gerät mit der ID **InstanzID** im WarmWeiß Kanal auf den Wert **WarmWeiß**.

## BEISPIEL

```
// WarmWeiß auf 50% setzen
ZW_ColorWW(12345, 128);
```

# ZW_DimDown

```
boolean ZW_DimDown (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Beginnt das Runterdimmen des Geräts mit der ID InstanzID.

## BEISPIEL

```
//Das Gerät mit der ID 12345 beginnt runterzudimmen
ZW_DimDown(12345);
```

# ZW_DimDownEx

```
boolean ZW_DimDownEx (int $InstanzID, int $Duration)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Duration | Laufzeit bis komplett heruntergedimmt wird (0..255) (Siehe Tabelle) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Beginnt das Runterdimmen des Geräts mit der ID **InstanzID** in der Laufzeit **Duration**.

| DURATION | BESCHREIBUNG |
|---|---|
| 0 | Sofort |
| 1 - 127 | 1 - 127 Sekunden |
| 128 - 254 | 1 - 127 Minuten |
| 255 | Gerätestandard für Duration |

## BEISPIEL

```
//Das Gerät mit der ID 12345 beginnt über eine Laufzeit von 10
Minuten runterzudimmen
ZW_DimDownEx(12345, 138);

//Das Gerät mit der ID 12345 beginnt über eine Laufzeit von 5
Sekunden runterzudimmen
ZW_DimDownEx(12345, 5);
```

# ZW_DimSet

```
boolean ZW_DimSet (int $InstanzID, int $Intensität)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Intensität | Wert von 0-100 (in %) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Dimmt das Gerät mit der ID **InstanzID** auf Wert **Intensität**

## BEISPIEL

```
ZW_DimSet(12345, 50); //Auf 50% dimmen
```

# ZW_DimSetEx

```
boolean ZW_DimSetEx (int $InstanzID, int $Intensität, int $Duration)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Intensität | Wert von 0-100 (in %) |
| Duration | Laufzeit bis die Intensität erreicht wird (0..255) (Siehe Tabelle) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Dimmt das Gerät mit der ID **InstanzID** auf Wert **Intensität** in der Laufzeit **Duration**.

| DURATION | BESCHREIBUNG |
|---|---|
| 0 | Sofort |
| 1 - 127 | 1 - 127 Sekunden |
| 128 - 254 | 1 - 127 Minuten |
| 255 | Gerätestandard für Duration |

## BEISPIEL

```
//Dimmt das Gerät mit der ID 12345 in 10 Minuten auf 50%
ZW_DimSetEx(12345, 50, 138);

//Dimmt das Gerät mit der ID 12345 in 5 Sekunden auf 30%
ZW_DimSetEx(12345, 30, 5);
```

# ZW_DimStop

```
boolean ZW_DimStop (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Unterbricht das Dimmen des Geräts mit der ID InstanzID

## BEISPIEL

```
ZW_DimStop(12345);
```

# ZW_DimUp

```
boolean ZW_DimUp (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Beginnt das Hochdimmen des Geräts mit der ID InstanzID.

## BEISPIEL

```
//Das Gerät mit der ID 12345 beginnt hochzudimmen
ZW_DimUp(12345);
```

# ZW_DimUpEx

```
boolean ZW_DimUpEx (int $InstanzID, int $Duration)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Duration | Laufzeit bis komplett hochgedimmt wird (0..255) (Siehe Tabelle) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Beginnt das Hochdimmen des Geräts mit der ID **InstanzID** in der Laufzeit **Duration**.

| DURATION | BESCHREIBUNG |
|---|---|
| 0 | Sofort |
| 1 - 127 | 1 - 127 Sekunden |
| 128 - 254 | 1 - 127 Minuten |
| 255 | Gerätestandard für Duration |

## BEISPIEL

```
//Das Gerät mit der ID 12345 beginnt über eine Laufzeit von 10
Minuten hochzudimmen
ZW_DimUpEx(12345, 138);

//Das Gerät mit der ID 12345 beginnt über eine Laufzeit von 5
Sekunden hochzudimmen
ZW_DimUpEx(12345, 5);
```

# ZW_DoorLockOperation

Benötigt: IP-Symcon >= 4.1

```
boolean ZW_DoorLockOperation (int $InstanzID, int $Modus)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Modus | Modus auf den das Z-Wave Gerät eingestellt werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den Door Lock Operation Mode des Geräts mit der ID **InstanzID** auf den Modus **Modus**.

| MODUS | WERT |
|---|---|
| Ungesichert | 0 |
| Ungesichert mit Timeout | 1 |
| Ungesichert (Innen) | 16 |
| Ungesichert (Innen) mit Timeout | 17 |
| Ungesichert (Außen) | 32 |
| Ungesichert (Außen) mit Timeout | 33 |
| Gesichert | 255 |

## BEISPIEL

```
//z.B. Schaltet den Door Lock Operation Mode auf Ungesichert mit
Timeout
ZW_DoorLockOperation(12345,1);
```

# ZW_LockMode

```
boolean ZW_LockMode (int $InstanzID, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |
| Wert | 0..255 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den LockMode bei dem Gerät mit der InstanzID **InstanzID**

## BEISPIEL

```
ZW_LockMode(12345, 0);
```

# ZW_MeterReset

```
boolean ZW_MeterReset (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt das Meter vom Gerät mit der InstanzID **InstanzID** zurück

## BEISPIEL

```
ZW_MeterReset(12345);
```

# ZW_Optimize

Benötigt: IP-Symcon >= 4.1

```
boolean ZW_Optimize (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Startet den Optimierungsprozess der Funkverbindung für das Gerät mit der InstanzID **InstanzID**



Der Optimierungsprozess wird zyklisch für alle in IP-Symcon integrierten Geräte automatisch ausgeführt.

## BEISPIEL

```
ZW_Optimize(12345);
```

# ZW_ProtectionSet

```
boolean ZW_ProtectionSet (int $InstanzID, int $Mode)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |
| Mode | 0..255 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt die Protection und dessen Mode **Mode** vom angegeben Gerät mit der InstanzID **InstanzID**

| MODE | INTEGER-WERT |
|---|---|
| Off | 0 |
| Limited | 1 |
| On | 2 |

## BEISPIEL

```
ZW_ProtectionSet(12345, 1);
```

# ZW_RequestStatus

```
boolean ZW_RequestStatus (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Fragt den Status für das Gerät mit der ID **InstanzID** ab. Nur netzbetriebene Geräte können per Befehl direkt abgefragt werden, da batteriebetriebene Geräte nicht dauerhaft dem Abfragen lauschen.



**Diese Funktion belastet den Funkverkehr erheblich.**
Sie sollte ausschließlich verwendet werden, um kritische Gerät zum IP-Symcon-Start zu aktualisieren. Ein erneutes Abfragen zur Laufzeit wird nicht benötigt, da Z-Wave im Normalfall alle Statusänderungen direkt an IP-Symcon weiterleitet. Ältere Geräte die dies nicht tun, sollten durch neuere Geräte ersetzt werden.

## BEISPIEL

```
ZW_RequestStatus(12345);
```

# ZW_ShutterMoveDown

```
boolean ZW_ShutterMoveDown (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Bewegt den Rollladen mit der ID **InstanzID** nach unten, bis ZW_ShutterStop ausgeführt oder die Endposition erreicht wird.



Beachten Sie, dass einige Rollladensteuerungen bei Z-Wave über ZW_DimSet wie Dimmer angesprochen werden müssen.

## BEISPIEL

```
ZW_ShutterMoveDown(12345); //Nach unten bewegen
```

# ZW_ShutterMoveUp

```
boolean ZW_ShutterMoveUp (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Bewegt den Rollladen mit der ID **InstanzID** nach oben, bis ZW_ShutterStop ausgeführt oder die Endposition erreicht wird.



Beachten Sie, dass einige Rollladensteuerungen bei Z-Wave über ZW_DimSet wie Dimmer angesprochen werden müssen.

## BEISPIEL

```
ZW_ShutterMoveUp(12345); //Nach oben bewegen
```

# ZW_ShutterStop

```
boolean ZW_ShutterStop (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Stoppt eine aktuelle Bewegung des Rollladens.



Beachten Sie, dass einige Rollladensteuerungen bei Z-Wave über ZW_DimSet wie Dimmer angesprochen werden müssen.

## BEISPIEL

```
ZW_ShutterStop(12345); //Anhalten
```

# ZW_SwitchAllMode

Benötigt: IP-Symcon >= 4.1

```
boolean ZW_SwitchAllMode (int $InstanzID, int $Modus)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Modus | Modus auf den das Z-Wave Gerät eingestellt werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt die Instanz mit der ID **InstanzID** auf den Modus **Modus**.

| MODUS | WERT |
|---|---|
| All On | 0 |
| Only Off | 1 |
| Only On | 2 |
| All Off | 255 |

## BEISPIEL

```
//Modus wird auf Only On gesetzt
ZW_SwitchAllMode(12345, 2);
```

# ZW_SwitchMode

```
boolean ZW_SwitchMode (int $InstanzID, boolean $Status)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |
| Status | TRUE für An, FALSE für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Schaltet das Gerät mit der ID InstanzID auf Wert Status

## BEISPIEL

```
ZW_SwitchMode(12345, true); //Gerät einschalten
```

# ZW_Test

Benötigt: IP-Symcon >= 4.1

```
boolean ZW_Test (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Testet die Verbindung zu dem Gerät mit der InstanzID **InstanzID**

## BEISPIEL

```
ZW_Test(12345);
```

# ZW_ThermostatFanModeSet

```
boolean ZW_ThermostatFanModeSet (int $InstanzID, int $FanMode)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |
| FanMode | 0..255 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den FanMode **FanMode** vom angegeben Gerät mit der InstanzID **InstanzID**

| FANMODE | INTEGER-WERT |
|---|---|
| AutoLow | 0 |
| OnLow | 1 |
| AutoHigh | 2 |
| OnHigh | 3 |

## BEISPIEL

```
ZW_ThermostatFanModeSet(12345,1); //z.B. Schaltet den FanMode auf
OnLow
```

# ZW_ThermostatModeSet

```
boolean ZW_ThermostatModeSet (int $InstanzID, int $Mode)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |
| Mode | 0..255 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den Mode **Mode** vom angegeben Gerät mit der InstanzID **InstanzID**

| MODE | INTEGER-WERT |
|---|---|
| Off | 0 |
| Heat | 1 |
| Cool | 2 |
| Auto | 3 |
| Auxiliary | 4 |
| Resume | 5 |
| FanOnly | 6 |
| Furnace | 7 |
| DryAir | 8 |
| MoistAir | 9 |
| AutoChangeover | 10 |

## BEISPIEL

```
ZW_ThermostatModeSet(12345, 1); //z.B. Setzt den Mode auf Heat
```

# ZW_ThermostatSetPointSet

```
boolean ZW_ThermostatSetPointSet (int $InstanzID, int $SetPoint,
float $Value)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Geräts |
| SetPoint | 0..255 |
| Value | 0..255 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den PointSet **PointSet** vom angegeben Gerät mit der InstanzID **InstanzID**

| SETPOINT | INTEGER-WERT |
|---|---|
| Invalid | 0 |
| Heating | 1 |
| Cooling | 2 |
| Furnace | 3 |
| DryAir | 4 |
| MoistAir | 5 |
| AutoChangeover | 6 |

## BEISPIEL

```
ZW_ThermostatSetPointSet(12345, 1, 20); //z.B. Anschalten
```

# Zevvy

Benötigt: IP-Symcon >= 7.0
Dieses Module schickt die Daten ausgewählter Variablen zu einem Zevvy Konto, welche dann und unter "Einstellungen" -> "Geräteliste" mit der ID der Variable auftaucht.

## FUNKTIONSUMFANG

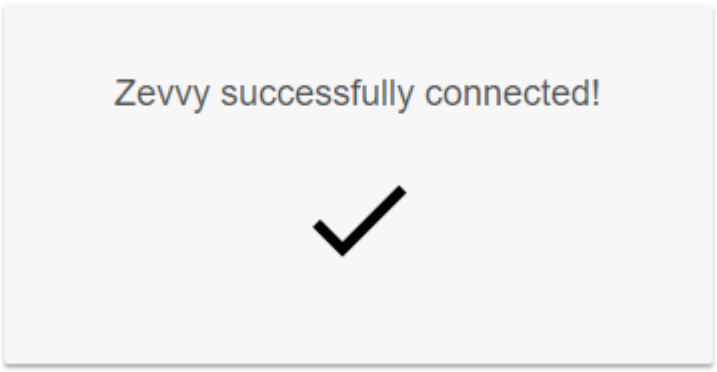
Verschicken der Werte

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Zevvy'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen
Bevor die Variablen eingerichtet werden können, muss auf den Registrieren-Button geklickt werden. Im Browser öffnet sich nun ein neuer Tab im Browser mit der Anmeldungsmaske von Zevvy.




Nach Eingabe der Anmeldedaten wird bei Erfolg folgender Dialog angezeigt.




Nun kann zu IP-Symcon zurückgekehrt werden. In der Zevvy Instanz sollte nun ein Token vorhanden sein.

Konfigurationsseite:

| NAME | BESCHREIBUNG |
| --- | --- |
| Geräte Liste | Liste mit den Variablen IDs, welche verschickt werden sollen. |
| Intervall | Intervall fürs automatische verschicken der Daten |
| Übermittle Daten | Button, welcher die Daten manuell verschickt |
| Registrierung | Verknüpft den Zevvy Account mittels OAuth mit IP-Symcon |

# ZY_SendMeasurements

Benötigt: IP-Symcon >= 7.0

```
boolean ZY_SendMeasurements (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | Id der Zevvy Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet die Daten der eingerichtet Variablen an ein Zevvy Konto.

## BEISPIEL

```
ZY_SendMeasurements(12345);
```

# Logiken

Logiken sind komplexere zusammengefasste Funktionsinstanzen, deren Funktion oder Automatisierung wesentlich komplexer sind, als durch ein Ereignis oder Ablaufplan schnell dargestellt werden kann. Dies vereinfacht die Einrichtung und bietet einen größeren Komfort.

## AKTIVLISTE

Übersicht aller aktiven Variablen in der Visualisierung

| | |
|---|---|
| AL_SwitchOff | Schaltet alle in der Liste vorhandenen aktiven Variablen inaktiv. |

## ANWESENHEITS-SIMULATION

Anwesenheit über geloggte Werte simulieren

| | |
|---|---|
| AS_SetSimulation | De-/Aktiviert die Anwesenheits-Simulation |

## BILD-ARCHIV

Das Modul kopiert bei Auslösung durch eine gewählte Variable ein Bild.

| | |
|---|---|
| BA_AddImage | Kopiert das aktuelle Quellbild in die Kategorie "Bilder" des Moduls "BildArchiv" mit der InstanzID. |

## COUNTDOWN

## CSV ZIP EXPORT

Exportiert aggregierte Werte einer Variable als CSV-Datei in einem ZIP-Archiv

| | |
|---|---|
| CSV_DeleteZip | Entfernt die generierte Datei. |
| CSV_Export | Erzeugt ein Zip-Archiv basierend auf den gegebenen Parametern. |
| CSV_SendMail | Senden durch eine SMTP Instanz eine Mail mit einer erzeugten Zip-Datei |

## DUMMY MODUL

Platzhalter Modul zur Darstellung als Instanz in der Visualisierung

## EIERUHR

Eine Eieruhr, die eine gewisse Zeit läuft und jederzeit wieder aufgezogen werden kann. Ideal für die Verwendung in Ereignissen und Skripten.

## GRUPPENSTEUERUNG

Mithilfe der Gruppensteuerungen können Variablen in Gruppen zusammen geschaltet werden.

## JSON DECODER

Dekodiert JSON und stellt den Inhalt in Variablen zur Verfügung

## JSON EXPORTER

Exportiert beliebige Variablen in einer benutzerdefinierten Struktur über einen HTTP Endpunkt.

## LOGIK GATTER

Verknüpft Variablen durch logische Operationen

## RECHENMODULE

Module, welche verschiedene Berechnungen innerhalb von IP-Symcon ermöglicht

| | |
|---|---|
| Rechen-Modul | Berechnung verschiedener Hilfswerte auf Basis einer Gruppe von Variablen, z.B. Summer oder Durchschnitt. |
| RM_Update | Aktualisiert die berechneten Werte |
| Umrechnen | Rechnet eine Variable über eine selbstdefinierte Formel um. |
| UMR_Calculate | Berechnet die Rückgabe der Formel für den Wert und gibt diesen zurück. |
| UmrechnenMultiGrenzen | Rechnet eine Variable in eingerichteten Grenzen über eine selbstdefinierte Formel um. |
| UMG_Calculate | Berechnet die Rückgabe der Formel für den Wert __Value__ und gibt diesen zurück. |
| WertebereichSkalieren | Skalierung eines Wertes innerhalb eines Wertebereiches |
| VRC_Scale | Skaliert den Eingabevariablenwert |

## RGBMULTIPLEXER

Verbindet einzelne R, G, B Kanäle zu einer Variable, welche mit dem Farbrad angesteuert werden kann.

| | |
|---|---|
| RGBM_RequestStatus | Ermittelt den Wert der einzelnen R, G, B Kanäle und setzt die Color Variable. |
| RGBM_SetRGB | Sendet die R, G, B Werte an die einzelnen Kanäle und aktualisiert die Color Variable |

## SCHLEPPZEIGER

Eine Variable, welche auf dem höchsten oder niedrigsten Wert bis zum Reset stehen bleibt.

| | |
|---|---|
| SZ_Reset | Setzt den Schleppzeiger zurück |

## SPIELESAMMLUNG

Eine Sammlung kleiner Spiele, welche über das Webfront gespielt werden können.

| | |
|---|---|
| Schere Stein Papier | Eine Möglichkeit Schere, Stein, Papier in der Visualisierung zu spielen. |
| Zahlenraten | Ein kleines Spiel, bei dem eine zufällige Zahl erraten werden muss. |
| ZR_Generate | Generiert eine neue Zahl, die erraten werden kann |

## SZENENSTEUERUNG

Speichert den Zustand einer belieben Anzahl an Aktoren und macht diese via Visualisierung/PHP-Funktion abrufbar.

| | |
|---|---|
| SZS_CallScene | Ruft die Szene auf und setzt sie dazugehörigen Variablen |
| SZS_GetActiveScene | Gibt die Nummer der Szene zurück, welche gerade aktiv ist. |
| SZS_SaveScene | Speichert die Werte der in der Liste vorhandenen Variablen in der entsprechenden Szene. |

## TAUPUNKT BERECHNEN

Berechnung der Taupunkttemperatur

## TREPPENHAUSLICHTSTEUERUNG

Beim Aktivieren wird das Licht im Treppenhaus eingeschaltet und wird kurz nach Verlassen des Treppenhauses wieder ausgeschaltet.

| | |
|---|---|
| THL_SetActive | Aktiviert oder deaktiviert die Treppenhauslichtsteuerung. |
| THL_Start | Aktiviert das Licht im Treppenhaus und startet den Timer, welcher das Licht wieder deaktiviert. |
| THL_Stop | Deaktiviert das Licht im Treppenhaus und den Timer. |

## UMBENNENER

Erlaubt das Umbenennen von ausgewählten Objekten über die Visualisierung

## UNWETTERZENTRALE

Holt das aktuelle Regenradarbild von Wetteronline und wertet die Farbpixel der Regenmenge aus.

| | |
|---|---|
| UWZ_RequestInfo | Berechnet die Variable Regenwert des Unwetterzentralemoduls |

## VARIABLENVERGLEICH

Das VariablenVergleichsmodul ermöglicht es aus zwei Variablen eine Punktwolke zu erstellen und mithilfe von linearer Regression vergleichbar zu machen.

| | |
|---|---|
| LR_Download | Generiert das Diagramm neu und gibt eine Adresse aus |
| LR_GenerateChart | Generiert das Diagramm neu und gibt es als SVG und PNG zurück |
| LR_UpdateChart | Generiert das Diagramm neu |

## VIRTUELLE GERÄTE

Module zum Simulieren verschiedener Geräte

| | |
|---|---|
| E-Auto (Simulation) | Simuliert einen Ladevorgang eines E-Autos |
| Heizstab (Simulation) | Simuliert einen Heizstab |
| Licht (Simulation) | Simuliert ein Licht |
| Mediaplayer (Simulation) | Simuliert einen MediaPlayer |
| PV-Anlage (Simulation) | Simulation einer PV-Anlage |
| Rollladen (Simulation) | Simuliert einen Rollladen. |
| Thermostat (Simulation) | Simuliert ein Thermostat |
| Verbrauch/Kosten | Errechnet die Kosten nach Verbrauch |
| Zähler (Simulation) | Simuliert eine einstellbare Anzahl an Zählern. |

## ZUFÄLLIGE BELEUCHTUNG

Das Modul ermöglicht es den Farbwert von Lampen zufällig zwischen verschiedenen Farben zu schalten.

| | |
|---|---|
| ZB_ChangeLight | Wählt zufällige Farben aus und setzt die ausgewählten Variablen auf diese |

# Aktivliste

Benötigt: IP-Symcon >= 5.0
Die Aktivliste zeigt alle aktiven Variablen in der Visualisierung an und bietet die Möglichkeit diese simultan auszuschalten.
Hierzu müssen sie zuvor der Liste auf der Konfigurationsseite hinzugefügt wurden.

## FUNKTIONSUMFANG

Zeigt alle aktiven Variablen in der Visualisierung an und erlaubt das Ausschalten dieser.

## SOFTWARE-INSTALLATION

Über den Module Store das Modul Aktivliste installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Aktivliste'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE

| NAME | BESCHREIBUNG |
|---|---|
| Aktion zum Ausschalten | Erstellt eine Aktion zum simultanen Auschalten der Variablen |
| Variablen | Eine Liste mit Variablen, deren Status überprüft wird. |

Variablen gelten als aktiv, wenn ...
der Wert einer Integer oder Float Variable größer als der Minimalwert ist. Sollte die Variable ein .Reversed Profil haben gilt sie als aktiv, wenn der Wert kleiner als der Maximalwert ist.
der Wert einer Boolean Variable true ist. Sollte die Variable ein .Reversed Profil haben ist false der aktive Zustand.
der Wert einer String Variable nicht leer ist.

Dementsprechend gelten Variablen als inaktiv, wenn ...
der Wert einer Integer oder Float Variable der Minimalwert ist. Sollte die Variable ein .Reversed Profil haben gilt sie als inaktiv, wenn der Wert der Maximalwert ist.
der Wert einer Boolean false ist. Sollte die Variable ein .Reversed Profil haben ist true der inaktive Zustand.
der Wert einer String Variable leer ist.

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner

kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
| --- | --- | --- |
| Ausschalten | Skript | Schaltet alle noch aktiven Variablen inaktiv. |

## VISUALISIERUNG

In der Visualisierung werden alle aktiven Variablen angezeigt. Mit einem Klick auf "Ausschalten" werden alle angezeigten Variablen auf inaktiv geschaltet.

# AL_SwitchOff

Benötigt: IP-Symcon >= 5.0

```
boolean AL_SwitchOff (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Schaltet alle in der Liste vorhandenen aktiven Variablen inaktiv.

## BEISPIEL

```
AL_SwitchOff(12345);
```

# Anwesenheits-Simulation

Benötigt: IP-Symcon >= 5.0
Simuliert die Anwesenheit von Personen im Haushalt. Das Modul bezieht dafür zufällig die Tagesdaten von einem der letzten 4 identischen Wochentagen. Sind an keinem dieser 4 Tage genug Schaltvorgänge geloggt, wird zufällig einer der letzten 30 Tage gewählt.
Ist auch innerhalb dieser 30 Tage kein gültiger Tagesdatensatz vorhanden, ist keine Simulation möglich. Sollte keine Simulation möglich sein, wird dies als Nachricht in der Stringvariable "Simulationsquelle (Tag)" angezeigt.

## FUNKTIONSUMFANG

Schalten von ausgewählten Aktoren/Variablen über geloggte Werte.
Einstellbarkeit der benötigen durchschnittlichen Schaltvorgänge, bevor ein Tag als Quelle zur Simulation zugelassen wird.
Ein-/Ausschaltbarkeit via Visualisierungsbutton oder Skript-Funktion.
Anzeige welcher Tag zur Simulation genutzt wird.
Automatische Aktualisierung bei Tageswechsel.

## SOFTWARE-INSTALLATION

Über den Module Store das Modul Anwesenheits-Simulation installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'AnwesenheitsSimulation'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen in der Dokumentation der Instanzen.
Alle zu schaltenden Variablen müssen der Variablen Liste in der Instanzkonfiguration hinzugefügt werden.

### KONFIGURATIONSSEITE

| NAME | BESCHREIBUNG |
|---|---|
| Variablen | Eine Liste der Variablen, welche für die Simulation der Anwesenheit genutzt werden sollen. Die hinzugefügten Variablen benötigen eine Aktion und müssen geloggt sein. |
| Mindestanzahl | Dies beschreibt die durchschnittliche Mindestanzahl von Variablenschaltungen aller ausgewählten Variablen, die vorhanden sein müssen. |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Simulation aktiv | Boolean | Zeigt an ob ob die Simulation aktiviert ist oder nicht. True = Aktiviert; False = Deaktiviert; |
| Simulationsquelle (Tag) | String | Der String beinhaltet das Datum, nach dem die Simulationsdaten ausgewählt wurden. |
| Simulationsvorschau | String | Zeigt eine Tabelle welche eine Übersicht über die zukünftigen Schaltvorgänge gibt. |

## VISUALISIERUNG

Über die Visualisierung kann die Simulation de-/aktiviert werden.
Es wird zusätzlich die Information angezeigt, welcher Tag zur Simulation genutzt wird. Falls nicht genügend oder ungültige Daten vorhanden sind, wird dieses ebenfalls hier angezeigt. Es wird eine Liste mit allen ausgewählten Variablen angezeigt, welche den aktuellen und nächsten Wert, sowie die Uhrzeit der Schaltung beinhaltet.

# AS_SetSimulation

Benötigt: IP-Symcon >= 5.0

```
boolean AS_SetSimulation (int $InstanzID, boolean $SetActive)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Anwesenheits-Simulation |
| SetActive | Aktiviert/Deaktiviert Anwesenheits-Simulation |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE

## BESCHREIBUNG

SetActive aktiviert (true) oder deaktiviert (false) die Anwesenheits-Simulation mit der InstanzID InstanzID.

## BEISPIEL

```
AS_SetSimulation(12345, true);
```

# Bild-Archiv

Benötigt: IP-Symcon >= 4.2
Das Modul kopiert bei Auslösung durch eine gewählte Variable ein Bild. Es ist einstellbar wieviele Bilder gespeichert werden sollen.

## FUNKTIONSUMFANG

Auswählbares Quellbild und auswählbare Auslöservariable
Einstellbare Anzahl von Bildern, welche gespeichert werden sollen
Wenn eine Auslöservariable ausgewählt ist wird ein Ereignis erstellt, welches auf die Aktualisierung dieser reagiert
Das Ereignis kann bis auf die Auslöservariable vollständig personalisiert werden
Bilder bekommen den Zeitpunkt des Auslösens als Namen
Bilder werden chronologisch von alt -> neu in der Kategorie Bilder angezeigt
Automatisches Löschen der ältesten Bilder, beim Erreichen der eingestellten maximalen Bilderanzahl

## SOFTWARE-INSTALLATION

Über den Module Store das Modul Bild-Archiv installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Bild-Archiv'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE

| NAME | BESCHREIBUNG |
|---|---|
| Bild | Quellbild, welches kopiert werden soll |
| Anzahl | Maximale Anzahl an Bildern, die gespeichert werden sollen |
| Auslösevariable | Variable bei deren Änderung das Quellbild ins Archiv kopiert werden soll |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
| --- | --- | --- |
| Bilder | Kategorie | Hier werden die Quellbilder hineinkopiert |
| AddImage | Ereignis | Wird durch die Auslösevariable ausgelöst und stößt das Kopieren des Quellbildes an |

## VISUALISIERUNG

Über die Visualisierung werden die Medienobjekte als Liste angezeigt. Es ist keine weitere Steuerung oder gesonderte Darstellung integriert.

# BA_AddImage

Benötigt: IP-Symcon >= 4.2

```
boolean BA_AddImage (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Kopiert das aktuelle Quellbild in die Kategorie "Bilder" des Moduls "BildArchiv" mit der InstanzID **InstanzID**. Die Funktion liefert keinerlei Rückgabewert.

## BEISPIEL

```
BA_AddImage(12345);
```

# Countdown

Benötigt: IP-Symcon >= 5.0
Das Modul ermöglicht es einen Countdown bis zu einem Event im Webfront anzuzeigen.

## FUNKTIONSUMFANG

Ermöglicht es einen Countdown im Webfront anzuzeigen, der sich minütlich aktualisiert
Festlegung eines Zeitpunktes und Namen eines Ereignis

## SOFTWARE-INSTALLATION

Über den Modul Store das Modul Countdown installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Countdown'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE

| NAME | BESCHREIBUNG |
|---|---|
| Datum und Uhrzeit | Datum und Uhrzeit |
| Ereignis | Name der in der Visualisierung angezeigt werden soll |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Ereignis erreicht | Boolean | Zeigt an ob das Ereignis erreicht wurde |
| Verbleibende Zeit | String | Beinhaltet die verbleibende Zeit |

## VISUALISIERUNG

Hier wird der Countdown angezeigt und, sowie eine Nachricht, wenn das Ereignis eingetreten ist.

# CSV ZIP Export

Benötigt: IP-Symcon >= 5.5
Dieses Modul bietet die Möglichkeit, die aggregierten Werte einer Variable als CSV-Datei in einem ZIP-Archiv zu exportieren.

## FUNKTIONSUMFANG

Exportieren von aggregierten Daten einer oder mehreren Variablen
Daten in CSV-Datei ein einem ZIP-Archiv exportieren
Auflistung aller geloggten Variablen
Zeitraum der Aggregierung frei wählbar
Aggregierungsstufe kann ausgewählt werden
Zyklisches erstellen und versenden eines Archivs per Mail oder FTP/FTPS/SFTP

## SOFTWARE-INSTALLATION

Über den Module Store das 'CSVZipExport'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'CSV-ZIP-Export'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE

| NAME | BESCHREIBUNG |
|---|---|
| Export Optionen | Option ob eine oder mehrere Variablen pro Datei exportiert werden sollen, oder eine einzelne vom MySQL genutzt Variable exportiert werden soll |
| Ausgewählte Variablen | [Export Optionen: Eine Datei pro Variable, Eine Datei mit mehreren Variablen] Liste an Variablen mit setzbaren für die CSV Spalte |
| Filter | [Export Option: MySQL] Filtriert die Auswahl der geloggten Variablen |
| Geloggte Variablen | [Export Option: MySQL] Auswahl der Variable welche exportiert werden soll |
| Zip Datei erstellen | Checkbox, ob die Datei in ein Zip-Archiv komprimiert werden soll |
| Dezimaltrennzeichen | Wählbar, ob Punkt oder Komma als Dezimaltrennzeichen genutzt werden soll |
| Start der Aggregation | Beginn des Aggregationszeitraums |
| Ende der Aggregation | Ende des Aggregationszeitraums |
| Dezimaltrennzeichen | Auswahl vom Dezimaltrennzeichen Komma oder Punkt. |
| Aggregationsstufe | Stufe der Aggregation |
| Exportieren | Die aggregierten Daten der Variable werden exportiert |
| Zyklischen senden | |
| Zyklisches senden aktivieren | Aktiviert die zyklische Versendung der Daten |
| Intervall | Intervall in welchem versendet wird. Bei "Wöchentlich" wird am Montag versendet und bei "Monatlich" wird jeweils am 1. des Monats versendet zum jeweilig gewählten Zeitpunkt |
| Zeitpunkt | Zeitpunkt zu welchem versendet werden soll |
| Mail | |
| Per Mail senden | Checkbox, ob die Daten über Mail versendet werden soll |
| SMPT-Instanz | Auswahl der E-Mail-Instanz |
| Jetzt Mail senden | Sendet manuell eine Mail |
| FTP, FTPS, SFTP | |
| Per SFTP/FTP/FTPS senden | Checkbox, ob die Daten über SFTP, FTP oder FTPS versendet werden sollen |
| Verbindungstyp | Verbindung ob über SFTP, FTP oder FTPS versendet werden soll |
| Host | IP Adresse des Servers |
| Port | Port auf dem der Client sich auf den Server verbindet. Bei FTPS und FTP ist dies normalerweise 21. Bei SFTP ist dies der Port 22 |
| Benutzername | Benutzernamen für die SFTP Verbindung |
| Passwort | Passwort für die SFTP Verbindung |

# CSV_DeleteZip

Benötigt: IP-Symcon >= 5.5

```
boolean CSV_DeleteZip (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Entfernt die generierte Datei.

## BEISPIEL

```
CSV_DeleteZip(12345);
```

# CSV_Export

Benötigt: IP-Symcon >= 5.5

```
string CSV_Export (int $InstanzID, int $ArchiveVariable, int
$AggregationStage, int $AggregationStart, int $AggregationEnd)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| ArchiveVariable | Variablen ID, welche exportiert werden soll |
| AggregationStage | Aggregationsstufe analog zu AC_GetAggregatedValues |
| AggregationStart | Datum/Zeit als Unix Zeitstempel (0 = von Anfang an) |
| AggregationEnd | Datum/Zeit als Unix Zeitstempel (0 = bis jetzt) |

## RÜCKGABEWERT

Der relative Pfad des Zip-Archivs.

## BESCHREIBUNG

Erzeugt ein Zip-Archiv basierend auf den gegebenen Parametern und liefert den relativen Pfad des Archivs als Rückgabewert.

## BEISPIEL

```
CSV_Export(12345, 54321, 4, 2293574400, 3127161600);
```

# CSV_SendMail

Benötigt: IP-Symcon >= 5.5

```
boolean CSV_SendMail (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Senden durch eine SMTP Instanz eine Mail mit einer erzeugten Zip-Datei

## BEISPIEL

```
CSV_SendMail(12345);
```

# Dummy Modul

Das Dummy Modul ist eine Instanz, die keine eigenen Funktion oder Status Variablen beinhaltet. Das Modul kann nur als Platzhalter verwendet werden, um als Gerät in der Visualisierung dargestellt zu werden.

## BEISPIEL

Mehrere erstellte Variablen gehören zu einem Gerät, welches IP-Symcon nicht nativ unterstützt. Um dennoch die Visualisierung als ein Gerät (z.B. in der Visualisierung) zu erhalten, können die erstellten Variablen unter der Dummy Instanz zusammengefasst werden.
In Kombination mit den Variablenprofilen und Variablenaktionen können eigene "Instanzen" erstellt werden, die in der Funktion und automatischen Visualisierung nativer Instanzen eine ähnliche Funktionsvielfalt beinhalten.

## DARSTELLUNG

Wie das Modul dargestellt wird kann man hier nachsehen:
Dummy Darstellung

# Eieruhr

Benötigt: IP-Symcon >= 5.0
Eine Eieruhr, die eine gewisse Zeit läuft und jederzeit wieder aufgezogen werden kann.
Ideal für die Verwendung in Ereignissen und Skripten.

## FUNKTIONSUMFANG

Eine einfache Eieruhr
Jederzeit wieder aufziehbar
Läuft eine einstellbare Zeit
Kontrollierbar über die Visualisierung
Einfache Integration in Ereignisse und Skripte über die Aktiv-Variable

## SOFTWARE-INSTALLATION

Über den Module Store das 'Eieruhr'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' ist das 'Eieruhr'-Modul unter dem Hersteller '(Gerät)' aufgeführt.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Aktualisierungsintervall | Das Intervall in dem die verbleibende Zeit aktualisiert wird (Wenn das gewählte Intervall größer als die verbleibende Zeit ist, wird "Aktiv" erst dann wieder auf false gesetzt, wenn das Intervall abgelaufen ist) |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
| --- | --- | --- |
| Aktiv | boolean | Legt fest ob die Eieruhr läuft oder nicht; Erneutes Setzen startet die Uhr neu |
| Zeit in Sekunden | integer | Die Dauer in Sekunden welche die Eieruhr läuft; Nach Ablauf der Zeit wird "Aktiv" auf false gesetzt |
| Verbleibend | string | Zeigt die verbleibende Zeit der laufenden Eieruhr an |
| Abgebrochen | integer | Zeigt an ob die Variable Aktiv vor Ablauf der Zeit wieder ausgeschaltet und somit der Timer abgebrochen wurde. |

## VISUALISIERUNG

In der Visualisierung kann die Laufzeit der Eieruhr eingestellt und de-/aktiviert werden

# Gruppensteuerung

Benötigt: IP-Symcon >= 5.0
Mithilfe der Gruppensteuerungen können Variablen in Gruppen zusammen geschaltet werden.

## FUNKTIONSUMFANG

Wenn eine Variable in der Gruppe geschaltet wird, werden alle verbleibenden in der Gruppe ebenso geschaltet
Alle Variablen, welche einer Liste hinzugefügt werden, können mit einer separaten Variable gleichzeitig geschaltet werden

## SOFTWARE-INSTALLATION

Über den Module Store das 'Gruppensteuerung'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Gruppensteuerung'-Modul mithilfe des Schnellfilters gefunden werden. - Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Variablen | Die in dieser Liste vorhandenen Variablen gehören zu der Gruppe; Alle Variablen müssen vom gleichen Typ sein und das gleiche Profil haben sowie eine Aktion |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Status | variant | Zeigt den Status der aktuellen Gruppe an |

## VISUALISIERUNG

Hier wird die Statusvariable angezeigt, welche die Gruppe schalten kann.

# JSON Decoder

Benötigt: IP-Symcon >= 6.0
Der JSON Decoder ist ein Modul, welches einen JSON-kodierten Inhalt in seine Bestandteile zerlegt und die Elemente als Variablen zur Verfügung stellt.

## EINBINDUNG IN IP-SYMCON

Der JSON Decoder kann als Instanz über das "+" im Objektbaum gesucht und hinzugefügt werden.
Standardmäßig wird ein HTTP Client erstellt und als Quelle genutzt. Innerhalb der Konfiguration des JSON Decoders kann über "Gateway ändern" ein Gateway, welches den zu bearbeitenden Text zur Verfügung stellt, ausgewählt werden.

## BEISPIEL

Als Beispiel soll die Seite filesamples.com/samples/code/json/sample4.json eingelesen und über den JSON Decoder dekodiert werden.
Dazu muss die URL der Seite im HTTP Client eingetragen werden und anschließend eingelesen werden.
Die Seite beinhaltet folgende JSON.

```
{
"people" : [
{
"firstName": "Joe",
"lastName": "Jackson",
"gender": "male",
"age": 28,
"number": "7349282382"
},
{
"firstName": "James",
"lastName": "Smith",
"gender": "male",
"age": 32,
"number": "5678568567"
},
{
"firstName": "Emily",
"lastName": "Jones",
"gender": "female",
"age": 24,
"number": "456754675"
}
]
}
```

Die über den HTTP Client eingelesene JSON wird nun mit dem JSON Decoder verarbeitet und folgende Informationen werden als Variablen im Objektbaum zur Verfügung gestellt.

| | | | | |
|---|---|---|---|---|
| 23811 | JSON Decoder | JSON Decoder | | |
| 46224 | people | Dummy Module | | |
| 40910 | Index_0 | Dummy Module | | |
| 10707 | age | Integer | 28 | 15:24:55 |
| 14117 | firstName | String | Joe | 15:24:55 |
| 14032 | gender | String | male | 15:24:55 |
| 41550 | lastName | String | Jackson | 15:24:55 |
| 18695 | number | String | 7349282382 | 15:24:55 |
| 22449 | Index_1 | Dummy Module | | |
| 59407 | Index_2 | Dummy Module | | |

# JSON Exporter

Benötigt: IP-Symcon >= 5.5
Das Modul bietet die Möglichkeiten, die aktuelle Werte von Variablen als JSON Export darzustellen. Dabei kann die Objektstruktur vorab definiert werden, sodass z.B. ein Array von Objekten mit ID, Name, Wert und Profilsuffix zurückgegeben wird. Es können im Anschluss beliebig der Objekte in dieser Struktur zurückgegeben werden, wobei die einzelnen Felder mit Variablen verknüpft oder als Freitext definiert werden.

## FUNKTIONSUMFANG

Individuelle Struktur der Objekte kann definiert werden
Ausgabe der aktuellen Werte im JSON Format, wobei ein Array von Objekten zurückgegeben wird

## SOFTWARE-INSTALLATION

Über den Module Store das 'JSON Exporter'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' ist das 'JSON Exporter'-Modul unter dem Hersteller '(Kern)' aufgeführt.

### KONFIGURATIONSSEITE

| NAME | BESCHREIBUNG |
|---|---|
| Konfiguriere Struktur | Über **Konfiguriere Struktur** kann der bereitzustellende Datensatz konfiguriert werden. Jeder Wert wird in die Tabelle **Zu exportierende Variablen** eingetragen |
| Ident | Auswählbarer Name |
| Wert | **Benutzerdefiniert** Auswahl, was von der Variable ausgewählt werden soll |
| Zu exportierende Variablen | Liste, in dem die Variablen nach der Struktur aufgelistet und hinzugefügt werden kann |
| Im Browser öffnen | Öffnet die JSON Struktur in einem Browser |

# Logik Gatter

Benötigt: IP-Symcon >= 5.0
Dieses Modul ermöglicht die Bestimmungen logischer Verknüpfung zwischen mehreren Variablen. Hierbei wird die boolsche Interpretation der Eingangsvariablen verwendet.

## FUNKTIONSUMFANG

Bestimmung der logischen Verknüpfung von Variablenwerten
Unterstützte Operationen:
- OR
- AND
- NOR
- NAND

Ausgang wird aktuell gehalten und aktualisiert sich sobald eine der Eingangsvariablen sich ändert

## SOFTWARE-INSTALLATION

Über den Module Store das 'LogikGatter'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" ist das 'LogikGatter'-Modul unter dem Hersteller '(Sonstige)' aufgeführt
Bei 'Berechnung' die gewünschte Operation auswählen
In der Liste 'Eingabe' die gewünschten Eingabevariablen auswählen
- 'Invertieren' kann aktiviert werden um den Eingabewert dieser Variable zu negieren
- Wird eine nicht-Boolean Variable ausgewählt, so wird diese als boolscher Wert interpretiert, z.B. 0 als false

## STATUSVARIABLEN

Die Statusvariable "Ausgabe" beinhaltet das aktuelle Ergebnis der logischen Operation

## VISUALISIERUNG

Die aktuelle Ausgabe wird angezeigt.

# Rechenmodule

Module, welche verschiedene Berechnungen innerhalb von IP-Symcon ermöglicht

| | |
|---|---|
| Rechen-Modul | Berechnung verschiedener Hilfswerte auf Basis einer Gruppe von Variablen, z.B. Summer oder Durchschnitt. |
| RM_Update | Aktualisiert die berechneten Werte |
| Umrechnen | Rechnet eine Variable über eine selbstdefinierte Formel um. |
| UMR_Calculate | Berechnet die Rückgabe der Formel für den Wert und gibt diesen zurück. |
| UmrechnenMultiGrenzen | Rechnet eine Variable in eingerichteten Grenzen über eine selbstdefinierte Formel um. |
| UMG_Calculate | Berechnet die Rückgabe der Formel für den Wert __Value__ und gibt diesen zurück. |
| WertebereichSkalieren | Skalierung eines Wertes innerhalb eines Wertebereiches |
| VRC_Scale | Skaliert den Eingabevariablenwert |

# Rechen-Modul

Benötigt: IP-Symcon >= 4.3
Dieses Modul kann verschiedene Hilfsberechnungen auf einer Menge von Variablen durchführen, beispielsweise die Summe oder den Durchschnitt.

## FUNKTIONSUMFANG

Berechnungen verschiedener Werte basierend auf einer Gruppe von Variablen:
- Summe
- Minimum
- Maximum
- Durchschnitt
- Anzahl der Variablen

Ausgabe der berechneten Werte in Variablen
Aktualisierung der Werte sobald sich eine der Variablen ändert

## SOFTWARE-INSTALLATION

Über den Module Store das Modul Rechenmodule installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Rechen-Modul' mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Berechnung | Auswahl der durchgeführten Berechnung(en) |
| Variablen | VariablenIDs der Variablen auf denen die Berechnung(en) durchgeführt werden; Für alle Berechnungen außer Anzahl müssen alle Variablen vom Typ Float oder Integer sein |

#### *MÖGLICHE BERECHNUNGEN:*

| NAME | BESCHREIBUNG |
|---|---|
| Alles | Alle in dieser Tabelle vorgestellten Berechnungen werden durchgeführt |
| Summe | Die Summe aller ausgewählten Variablen wird in der Statusvariablen Summe gespeichert |
| Minimum | Der minimale Wert der ausgewählten Variablen wird in der Statusvariablen Minimum gespeichert |
| Maximum | Der maximale Wert der ausgewählten Variablen wird in der Statusvariablen Maximum gespeichert |
| Durchschnitt | Der Durchschnitt der ausgewählten Variablen wird in der Statusvariablen Durchschnitt gespeichert |
| Anzahl | Die Anzahl der ausgewählten Variablen wird in der Statusvariablen Anzahl gespeichert |

## STATUSVARIABLEN

Die Statusvariablen werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

Für jede Berechnung wird eine dazugehörige Statusvariable angelegt.

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Summe | Float | Die Summe aller ausgewählten Variablen |
| Minimum | Float | Der minimale Wert der ausgewählten Variablen |
| Maximum | Float | Der maximale Wert der ausgewählten Variablen |
| Durchschnitt | Float | Der Durchschnitt der ausgewählten Variablen |
| Anzahl | Float | Die Anzahl der ausgewählten Variablen |

# RM_Update

Benötigt: IP-Symcon >= 4.3

```
boolean RM_Update (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Aktualisiert die berechneten Werte

## BEISPIEL

```
RM_Update(12345);
```

# Umrechnen

Benötigt: IP-Symcon >= 4.3

## FUNKTIONSUMFANG

Berechnet über eine eingerichtete Formel einen Wert aus einer ausgewählten Quellvariable.
Bei Variablenänderung der Quellvariable wird der Wert automatisch neuberechnet.

## SOFTWARE-INSTALLATION

Über den Module Store das Modul Rechenmodule installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Umrechnen'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Quelle | Quellvariable, die Berechnung genutzt werden soll. |
| Formel | Formel, bei der Rechnung genutzt werden soll. |
| Wert | Testwert um die Formel zu Testen |
| Berechne | Berechnet den Wert anhand des Test-"Wert" |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Value | Float | Beinhaltet den anhand der eingerichteten Formel berechneten Wert. |

# UMR_Calculate

Benötigt: IP-Symcon >= 4.3

```
float UMR_Calculate (int $InstanzID, float $Value)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| Value | Einzusetzender Wert in die Formel |

## RÜCKGABEWERT

Umgerechneter Wert **Value**

## BESCHREIBUNG

Berechnet die Rückgabe der Formel für den Wert **Value** und gibt diesen zurück.

## BEISPIEL

```
//Funktion in der Konfiguration: $Value/10\n$ausgabe =
UMR_Calculate(12345, 50);\nvar_dump($ausgabe) //Ausgabe ist 5
```

# UmrechnenMultiGrenzen

Benötigt: IP-Symcon >= 4.3

## FUNKTIONSUMFANG

Berechnet über eingerichtete Formeln einen Wert aus einer ausgewählten Quellvariable.
Welche Formel genutzt werden soll, wird über einrichtbare Grenzwerte entschieden.
Bei Variablenänderung der Quellvariable wird der Wert automatisch neuberechnet.
Wenn keine Berechnung/Grenzen zutrifft/zutreffen, wird der Originalwert eingetragen.

## SOFTWARE-INSTALLATION

Über den Module Store das Modul Rechenmodule installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON)

Unter "Instanz hinzufügen" kann das 'UmrechnenMultiGrenzen'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Quelle | Quellvariable, die zur Berechnung genutzt werden soll. |
| Formel 1-10 | Formel, die auf die Quellvariable angewendetet werden soll. |
| Grenze 0-10 | Grenzwerte zwischen denen die eingerichtete Formel genutzt wird. |
| Wert | Testwert um die Formel zu Testen |
| Berechne | Berechnet den Wert anhand des Test-"Wert" |

Der Wert der Quellvariable kann innerhalb der Formel mit "$Value" implementiert werden. Beispiel:

```
10*$Value+20
```

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
| --- | --- | --- |
| Value | Float | Beinhaltet den anhand der eingerichteten Formel berechneten Wert. |

# UMG_Calculate

Benötigt: IP-Symcon >= 4.3

```
float UMG_Calculate (int $InstanzID, float $Value)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| Value | In die Formel einzusetzender Wert |

## RÜCKGABEWERT

Ergebnis der Berechnung.

## BESCHREIBUNG

Berechnet die Rückgabe der Formel für den Wert **Value** und gibt diesen zurück.

## BEISPIEL

```
//Funktion in der Konfiguration: $Value/10\n$ausgabe =
UMG_Calculate(12345, 50);\nvar_dump($ausgabe) //Ausgabe ist 5
```

# WertebereichSkalieren

Benötigt: IP-Symcon >= 4.3

## FUNKTIONSUMFANG

Eingabe vom Eingabewertebereich
Eingabe vom Ausgabewertebereich
Umrechnung von einem Eingabewert
Berechnung bei Wertänderung des Eingabewertes

## SOFTWARE-INSTALLATION

Über den (link: /de/service/dokumentation/komponenten/verwaltungskonsole/module-store text: Module Store das 'WertebereichSkalieren'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'WertebereichSkalieren'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Eingabevariable | Variable, welche als Eingabewert dient |
| Minimaler Eingabewert | Die untere Wertgrenze der Eingabe |
| Maximaler Eingabewert | Die obere Wertgrenze der Eingabe |
| Minimaler Ausgabewert | Die untere Wertgrenze der Ausgabe |
| Maximaler Ausgabewert | Die obere Wertgrenze der Ausgabe |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Ausgabe | float | Ausgabewert der Berechnung |

# VRC_Scale

Benötigt: IP-Symcon >= 4.3

```
boolean VRC_Scale (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | Id der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Skaliert den Wert der Eingabevariable nach den gesetzten Grenzen mit einer linearen Funktion.

## BEISPIEL

```
VRC_Scale(12345)
```

# RGBMultiplexer

Benötigt: IP-Symcon >= 4.2

## FUNKTIONSUMFANG

Bietet ein Farbrad, welches im Hintergrund drei einzelne R, G, B Kanäle ansteuert
Sollten die R, G, B Kanäle den Wert ändern wird der neue Zustand auch ins Farbrad übertragen

## SOFTWARE-INSTALLATION

Über den Module Store das Modul RGB-Multiplexer installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'RGB Multiplexer'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Variable (R) | Variable für den Rot Kanal |
| Variable (G) | Variable für den Grün Kanal |
| Variable (B) | Variable für den Blau Kanal |
| Button "RGB setzen" | Sendet den RGB-Wert an die Kanäle |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Color | Integer | Beinhaltet Wert passend zum Farbrad |

## VISUALISIERUNG

Über die Visualisierung werden die Variablen angezeigt. Über das Farbrad kann die Farbe

eingestellt werden.

# RGBM_RequestStatus

Benötigt: IP-Symcon >= 4.2

```
boolean RGBM_RequestStatus (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Ermittelt den Wert der einzelnen R, G, B Kanäle und setzt die Color Variable. Diese Funktion muss nicht aufgerufen werden. Dies passiert bei Änderung einer der Variable der R, G, B Kanäle automatisch.

## BEISPIEL

```
RGBM_RequestStatus(12345);
```

# RGBM_SetRGB

Benötigt: IP-Symcon >= 4.2

```
boolean RGBM_SetRGB (int $InstanzID, int $Rot, int $Grün, int $Blau)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| Rot | Wert des roten Farbkanals |
| Grün | Wert des grünen Farbkanals |
| Blau | Wert des blauen Farbkanals |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Sendet die R, G, B Werte an die einzelnen Kanäle und aktualisiert die Color Variable

## BEISPIEL

```
RGBM_SetRGB(12345, 255, 255, 255);
```

# Schleppzeiger

Benötigt: IP-Symcon >= 6.0
Eine Variable, welche auf dem höchsten oder niedrigsten Wert bis zum Reset stehen bleibt.

## FUNKTIONSUMFANG

Variable, welche den höchsten/niedrigsten Wert seit dem letztem Reset anzeigt
Reset via Timer oder manuel möglich

## SOFTWARE-INSTALLATION

Über den Module Store das 'Schleppzeiger'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Schleppzeiger'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen
Konfigurationsseite:

| NAME | BESCHREIBUNG |
|---|---|
| Zielvariable | Variable, deren Werte ausgelesen werden |
| Optionen | **Maximum** Gibt an ob der Schleppzeigen den höchsten oder niedrigsten Wert anzeigen soll |
| Intervall | Zeitangabe in Sekunden bis der Schleppzeiger auf den Wert der Zielvariable zurückgesetzt wird. |

## STATUSVARIABLEN UND PROFILE

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Schleppzeiger | float | Zeigt den höchsten/niedrigsten Wert seit dem letzten Reset an |
| Zurücksetzen | boolean | Assoziation zum Zurücksetzen des Schleppzeigers |

### PROFILE

| NAME | TYP |
| --- | --- |
| SZ.Reset | bool |

## VISUALISIERUNG

Anzeige des Schleppzeigers, sowie Zurücksetzen des Schleppzeigers

# SZ_Reset

Benötigt: IP-Symcon >= 6.0

```
boolean SZ_Reset (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Schleppzeiger Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den Schleppzeiger zurück.

## BEISPIEL

```
SZ_Reset(12345);
```

# Spielesammlung

Eine Sammlung kleiner Spiele, welche über das Webfront gespielt werden können.

| | |
|---|---|
| Schere Stein Papier | Eine Möglichkeit Schere, Stein, Papier in der Visualisierung zu spielen. |
| Zahlenraten | Ein kleines Spiel, bei dem eine zufällige Zahl erraten werden muss. |
| ZR_Generate | Generiert eine neue Zahl, die erraten werden kann |

# Schere Stein Papier

Benötigt: IP-Symcon >= 5.0

## FUNKTIONSUMFANG

Schere, Stein, Papier in IP-Symcon

## SOFTWARE-INSTALLATION

Über den Module Store das Modul Spielesammlung installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'SchereSteinPapier'-Modul mithilfe der Schnellsuche einfach gefunden werden.

## STATUSVARIABLEN UND PROFILE

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Deine Wahl | Integer | Wahl des Spielers. |
| Computer wählte | Integer | Zeigt die Wahl des Computers. |
| Ergebnis | String | Zeigt das Ergebnis. |

### PROFILE:

| NAME | TYP |
|---|---|
| SSP.Choice | Integer |

## VISUALISIERUNG

Hier wird gespielt.

# Zahlenraten

Benötigt: IP-Symcon >= 5.0

## FUNKTIONSUMFANG

Ein kleines Spiel, bei dem eine zufällige Zahl erraten werden muss.

## SOFTWARE-INSTALLATION

Über den Module Store das Modul Spielesammlung installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Zahlenraten'-Modul mithilfe der Schnellsuche einfach gefunden werden.

### KONFIGURATIONSSEITE:

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Minimum der generierten Zahl | Integer | Das Minimum der Zahl die generiert wird. |
| Maximum der generierten Zahl | Integer | Maximum der generierten Zahl. |
| Anzahl der erlaubten Versuche | Integer | Die Anzahl der Versuche, die der Spieler zum Erraten der Zahl hat. |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Züge übrig | Integer | Zeigt die Anzahl der verbleibenden Versuche. |
| Deine Zahl ist | String | Zeigt an ob der Tipp größer oder kleiner als die zu erratende Zahl ist. |
| Dein Tipp | Integer | Das Eingabefeld für den Tipp des Spielers. |

## VISUALISIERUNG

Hier wird gespielt.

# ZR_Generate

Benötigt: IP-Symcon >= 5.0

```
boolean ZR_Generate (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Generiert eine neue Zahl, die erraten werden kann.

## BEISPIEL

```
ZR_Generate(12345);
```

# Szenensteuerung

Benötigt: IP-Symcon >= 5.0
Die Szenensteuerung speichert Werte von in einer Liste gespeicherten Variablen in Szenen und kann diese via Knopfdruck aus der Visualisierung und mobilen Apps wieder aufrufen.
Die zu schaltenden Variablen müssen dazu in der Instanzkonfiguration der Liste "Variablen" hinzugefügt werden. Sind alle für eine Szene gewünschten Variablen hinzugefügt und auf den gewünschten Wert gesetzt, können diese mit dem "Speichern"-Knopf (in der Visualisierung) der entsprechenden Szene hinzugefügt werden. Nun kann die Szene mit dem entsprechenden "Ausführen"-Knopf jederzeit abgerufen werden. Die Variablen werden auf die zuvor gespeicherten Werte gesetzt.

## FUNKTIONSUMFANG

Ermöglicht das Speichern und Aufrufen von in einer Liste gespeicherten Variablen über Szenen.
Die aktuell aktive Szene wird angezeigt
Darstellung und Bedienung via Visualisierung und mobilen Apps
Wenn eine Variable ausgetauscht wird, bleiben bereits erstellte Szenen erhalten und schalten das neue Gerät

## SOFTWARE-INSTALLATION

Über den Module Store das Modul Szenen-Steuerung installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Szenensteuerung'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Szenen | Anzahl der Szenen die zur Verfügung gestellt werden. |
| Variablen | Liste mit den zu schaltenden Variablen. |

## STATUSVARIABLEN UND PROFILE

Die Statusvariablen werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

Die Szenen werden 1,2..n aufsteigend durchnummeriert.

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Aktive Szene | String | Zeigt den Namen der gerade Aktiven Szene an. Wenn die Werte der Variablen einer Szene zugeornet werden können wird diese ebenfalls angezeigt. Entsprechen die Werte der Variablen keiner Szene wird 'Unbekann' angezeigt |
| Szene 1..n | Integer | Zur Anzeige in der Visualisierung und den mobilen Apps. Ruft "Speichern" oder "Aufrufen" auf. |

**PROFILE:**

| NAME | TYP |
|---|---|
| SZS.SceneControl | Integer |

## VISUALISIERUNG

Über die Visualisierung können die momentanen Werte der gelisteten Zielvariablen in einer Scene gespeichert werden. Über "Ausführen" können bereits gespeicherte Szenen aufgerufen werden.

# SZS_CallScene

Benötigt: IP-Symcon >= 5.0

```
boolean SZS_CallScene (int $InstanzID, int $SceneNumber)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| SceneNumber | Nummer der Szene |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Ruft die in dem Szenensteuerungsmodul mit der InstanzID **InstanzID** gespeicherten Werte der Szene mit der Nummer **SceneNumber** auf und setzt die dazugehörigen Variablen.

## BEISPIEL

```
SZS_CallScene(12345, 1);
```

# SZS_GetActiveScene

Benötigt: IP-Symcon >= 5.0

```
int SZS_GetActiveScene (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Nummer der aktiven Szene. Wenn eine unbekannte Szene aktiv ist wird 0 zurückgegeben.

## BESCHREIBUNG

Gibt die Nummer der Szene zurück, welche gerade aktiv ist.

## BEISPIEL

```
SZS_GetActiveScene(12345);
```

# SZS_SaveScene

Benötigt: IP-Symcon >= 5.0

```
boolean SZS_SaveScene (int $InstanzID, int $SceneNumber)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| SceneNumber | Nummer der Szene |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Speichert die Werte der in der Liste vorhandenen Variablen in der entsprechenden Szene.

## BEISPIEL

```
SZS_SaveScene(12345, 1);
```

# Taupunkt berechnen

Berechnung des Taupunktes mit der relativen Luftfeuchtigkeit und Raumtemperatur.
Die Berechnung basiert auf den Formeln von der Seite Wetterochs.

## FUNKTIONSUMFANG

Berechnung der Taupunkttemperatur
Berechnung der Temperatur ab dem das Risiko zum Schimmelbefall steigt

## SOFTWARE-INSTALLATION

Über den Module Store das 'TaupunktBerechnung'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'TaupunktBerechnung'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

Konfigurationsseite:

| NAME | BESCHREIBUNG |
|---|---|
| Raumtemperatur | Temperatur, welche im Raum herrscht in °C |
| Luftfeuchtigkeit | Relative Luftfeuchtigkeit, welche herrscht |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Taupunkt | float | Temperatur, bei unterschreiten wird überschüssiger Wasserdampf sich in Wasser wandeln |
| Temperatur für Schimmelgefahr | float | Temperatur, bei unterschreiten dieser steigt das Risiko des Schimmelbefalls |
| Alarm für Schimmelgefahr | boolean | Alarm, welcher auslöst wenn die Temperatur für Schimmelgefahr unterschritten wird. Ausgeschaltet wird dieser, wenn die Raumtemperatur 1°C über der Temperatur für Schimmelgefahr ist. |

# Treppenhauslichtsteuerung

Benötigt: IP-Symcon >= 5.0
Nachdem ein Auslöser aktiviert wird, geht das Licht im Treppenhaus an. Wird der Auslöser wiederholt aktiviert bleibt das Licht an und der Timer wird zurückgesetzt. Erst wenn für eine vorgegebene Zeit keine weitere Auslösung stattfindet wird das Licht ausgeschaltet.

## FUNKTIONSUMFANG

Auswahl von Ein- und Ausgabevariablen in einer Liste.
Auswahl der Dauer bevor das Licht ausgeschaltet wird.
Angabe der Helligkeit in Abhängigkeit einer Nacht-Modus Variable
.Reversed Profile werden sowohl für Ein- als auch Ausgabevariablen unterstützt
Möglichkeit die verbleibende Zeit bis zum Auschalten anzuzeigen.

## SOFTWARE-INSTALLATION

Über den Module Store das Modul Treppenhauslichtsteuerung installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Treppenhauslicht'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
| --- | --- |
| Eingabesensoren | Liste der Eingabesensoren, bei deren Aktivierung das Licht aktiviert werden soll, z.B. Bewegungssensoren oder Taster - Das Licht wird aktiviert sobald eine Variable auf aktiv gesetzt wird. Als aktiv gelten hierbei Variablen mit einem Wert, der nicht false, 0, oder "" ist. Sollte die Variable ein .Reversed Profil haben gelten die genannten Werte als aktiv. |
| Ausgabevariablen | Liste der Variablen, welche aktiv, also auf ihren Maximalwert geschaltet werden und das Licht darstellen. Sollte eine Variable ein .Reversed Profil haben wird diese auf den Minimalwert geschaltet. Variablen des Typs String werden nicht geschaltet. Die Variablen werden akiv geschaltet, wenn ein Sensor aus der Eingabesensor Liste ausgelöst wird. |
| Dauer | Nachdem die ausgewählte Dauer ohne weitere Auslösung eines Eingabesensors vergeht, wird das Licht deaktiviert. |
| Aktion erneut senden | Wenn ein unzuverlässiges Funk-System verwendet wird, so ist es ggf. erforderlich bei jedem Impuls die Aktion zu senden. Im Normalfall sollte diese Option deaktiviert bleiben, da ständiges senden der Aktion bei Funk-Aktoren ggf. den Duty-Cycle aufbrauchen kann. |
| Restlaufzeit anzeigen | Wenn aktiv wird die verbleibende Zeit bis zum Ausschalten in einer Variable angezeigt. |
| Aktualisierungsintervall | Das Intervall, in dem die "Restzeit" Variable aktualisiert wird. |
| Nacht-/Tag-Modus | Ermöglicht es die gewählten Variablen basiernd auf der Tageszeit, oder der Umgebungshelligkeit auf unterschiedliche Werte zu schalten |

## NACHT-/TAG-MODUS - NACHT-/TAG VARAIBLE

| NAME | BESCHREIBUNG |
| --- | --- |
| Tag/Nacht | Eine Variable, die angibt ob Nacht oder Tag ist. |
| Invertiert | Gibt an, ob der Wert der Nacht-Modus Variable invertiert werden soll. Dies ist notwendig, wenn die Ist-Tag Variable von der Location Instanz verwendet werden soll. Diese ist nämlich FALSE, wenn es Dunkel ist. |
| Helligkeit (Nacht-Modus) | Gibt die Helligkeit in Prozent an, die Nachts geschaltet werden soll. |
| Helligkeit (Tag-Modus) | Gibt die Helligkeit in Prozent an, auf die am Tag geschaltet werden soll. |

## NACHT-/TAG-MODUS - UMGEBUNGSHELLIGKEITSVARIABLE

| NAME | BESCHREIBUNG |
| --- | --- |
| Umgebungshelligkeit | Die Variable, die als Umgebungshelligkeit genutzt wird. |
| Umgebungshelligkeitsschwelle | Der Grenzwert, bei dem zwischen den Werten für Tag und Nacht gewechselt wird. |
| Helligkeit (Nacht-Modus) | Gibt die Helligkeit in Prozent an, die bei Unterschreitung der Helligkeit geschaltet werden soll. |
| Helligkeit (Tag-Modus) | Gibt die Helligkeit in Prozent an, die bei Überschreitung der Helligkeit geschaltet werden soll. |

## STATUSVARIABLEN

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
| --- | --- | --- |
| Treppenhaussteuerung aktiv | Boolean | Die Variable gibt an, ob die Treppenhaussteuerung aktiviert ist |
| Restzeit | String | Wenn "Restlaufzeit anzeigen" aktiv ist wird hier die verbleibende Zeit bis zum Auschalten angezeigt |

# THL_SetActive

Benötigt: IP-Symcon >= 5.0

```
boolean THL_SetActive (int $InstanzID, boolean $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| Wert | TODO |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Aktiviert oder deaktiviert die Treppenhauslichtsteuerung. Wurde das Treppenhauslicht durch die Steuerung eingeschaltet und die Steuerung wird deaktiviert, so wird der aktuelle Steuervorgang noch zu Ende geführt. Allerdings wird der Timer bei erneutem Auslösen des Eingabesensors nicht zurückgesetzt. Das Treppenhauslicht wird also trotz deaktivierter Steuerung nach Ablauf des Timers ausgeschaltet.

## BEISPIEL

```
THL_SetActive(12345, true);
```

# THL_Start

Benötigt: IP-Symcon >= 5.0

```
boolean THL_Start (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Aktiviert das Licht im Treppenhaus und startet den Timer, welcher das Licht wieder deaktiviert. Bei wiederholtem Aufruf wird der Timer zurückgesetzt.

## BEISPIEL

```
THL_Start(12345);
```

# THL_Stop

Benötigt: IP-Symcon >= 5.0

```
boolean THL_Stop (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Deaktiviert das Licht im Treppenhaus und den Timer.

## BEISPIEL

```
THL_Stop(12345);
```

# Umbennener

Benötigt: IP-Symcon >= 4.4
Ermöglicht das umbenennen von Objekten aus der Visualisierung heraus.

## FUNKTIONSUMFANG

Erlaubt es mehrere Objekte aus der Visualisierung heraus umzubenennen.
Namensänderungen der Objekte, die nicht durch das Modul gemacht werden, werden nicht an die Statusvariaben weiter gegeben.

## SOFTWARE-INSTALLATION

Über den Module Store das Modul 'Umbennener' installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Umbenenner'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Objekte | Eine Liste, zu der alle Objekte hinzugefügt werden müssen, die im Webfront umbenannt werden können sollen. |

## STATUSVARIABLEN

### STATUSVARIABLEN

Für jedes in der Instanzkonfiguration ausgewähltes Objekt wird eine Statusvariable vom Typ String erstellt. Der Name der Variablen ist der aktuelle Ort des Objektes welches umbenannt wird.

## VISUALISIERUNG

Hier können die zuvor ausgewählten Objekte umbenannt werden.

# Unwetterzentrale

Benötigt: IP-Symcon >= 4.2
Das Modul berechnet einen gewichteten Regenwert in einem ausgewählten Bereich des Radarbilds.

## FUNKTIONSUMFANG

Berechnet in einem selbstbestimmten Bereich des Regenradars die momentane Regenmenge.
Auswahl zwischen bundesländischen oder bundesweiten Radarbildern.
Das Radarbild und die Berechnung findet alle 15 Min automatisch statt.

## SOFTWARE-INSTALLATION

Über den Module Store das Modul Unwetterzentrale installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Unwetterzentrale'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Gebiet | Auswahl des bundesländischen oder bundesweiten Kartenauschschnitts |
| X | X-Position vom Mittelpunkt des Karrees |
| Y | Y-Position vom Mittelpunkt des Karrees |
| Radius | Kantenlänge des Karrees in Pixel |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
| --- | --- | --- |
| Regenwert | Integer | Gewichteter Wert der ausgewerteten Regenmenge. 8-stufige Gewichtung: Leichter Regen = Hellblau = 1 -> ... -> Starker Regen = Dunkelblau = 8 |
| Radarbild | Medien | Radarbild des ausgewählten Kartenausschnitts |

## VISUALISIERUNG

Über die Visualisierung werden die Variablen und das Radarbild angezeigt. Es ist keine weitere Steuerung oder gesonderte Darstellung integriert. Ebenfalls ist auf dem Radarbild der ausgewertete Karreeausschnitt eingezeichnet.

# UWZ_RequestInfo

Benötigt: IP-Symcon >= 4.2

```
boolean UWZ_RequestInfo (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Berechnet die Variable Regenwert des Unwetterzentralemoduls mit der InstanzID **InstanzID**.

## BEISPIEL

```
UWZ_RequestInfo(12345);
```

# VariablenVergleich

Benötigt: IP-Symcon >= 6.0
Dieses Modul wurde gefördert durch IB-Jaetzel.
Das Modul erstellt anhand einer oder mehreren Variablenpaaren ein Diagramm mit Puktewolken und mit Hilfe von einfacher Liniaren Regression eine Linie durch die Punktewolke. Das Diagramm kann als SVG oder PNG ausgegeben werden.

## FUNKTIONSUMFANG

Erstellung eines Diagramms mit ein oder mehreren Wertepaaren
Ausgabe des Diagramms in SVG oder PNG
Ausgabe von Steigung, Y-Achsenabschnitt, Funktion und Bestimmtheitsmaß des Graphen

## SOFTWARE-INSTALLATION

Über den Module Store das 'VariablenVergleich'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'VariablenVergleich'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

Werte:

| NAME | BESCHREIBUNG |
|---|---|
| X Wert | Variablenauswahl für die X-Achse |
| Y Wert | Variablenauswahl für die Y-Achse |
| Punkt | Farbauswahl für die Punktemakierung |
| Linie | Farbauswal für die Linie |
| Aggregationsstufe | Stufe wie detailiert die Daten aus dem Achiv geholt werden |

Diagramm Einstellungen:

| NAME | BESCHREIBUNG |
|---|---|
| Achsen kleine Schritte | Gibt an in welchen Schritten kleine Makierungen sein sollen |
| Achsen große Schritte | Gibt an in welchen Schritten die Makierung mit Beschriftung ist |
| Breite | Gibt an wie breit das Diagramm ist |
| Höhe | Gibt an wie hoch das Diagramm ist |
| Y - Min | Gibt den minimalen Wert auf der Y-Achse an |
| Y - Max | Gibt den maximalen Wert auf der Y-Achse an |
| X - Min | Gibt den minimalen Wert auf der X-Achse an |
| X - Max | Gibt den maximalen Wert auf der X-Achse an |
| Digramm Format | Gibt an ob das Diagramm als SVG oder PNG ausgegeben werden soll |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Funktion | String | Gibt die mathematische Funktion des Graphen wieder |
| b | Float | Beschreibt den y-Achsenabschnitt des Graphen |
| m | Float | Beschreibt die Steigung des Graphen |
| Bestimmtheitsmaß | Float | Beschreibt wie genau der Graph zu der Punktewolke passt |
| Startdatum | Integer | Datum, ab welchem die Punktewolke starten soll |
| Enddatum | Integer | Datum, bis zu welchem die Punktewolke geht |
| Chart | String / Media | Gibt das Diagramm je nach Einstellung als PNG oder SVG |

# LR_Download

Benötigt: IP-Symcon >= 6.0

```
void LR_Download (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Keine Rückgabe.

## BESCHREIBUNG

Generiert das Diagramm neu und gibt eine Adresse aus. Über einen Button im Konfigurationsformular wird im Browser das Diagramm angezeigt.

## BEISPIEL

```
LR_Download(12345);
```

# LR_GenerateChart

Benötigt: IP-Symcon >= 6.0

```
array LR_GenerateChart (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

SVG und PNG Diagramm

## BESCHREIBUNG

Generiert das Diagramm neu und gibt es als SVG und PNG zurück.

## BEISPIEL

```
LR_GenerateChart(12345);
```

# LR_UpdateChart

Benötigt: IP-Symcon >= 6.0

```
void LR_UpdateChart (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Keine Rückgabe.

## BESCHREIBUNG

Generiert das Diagramm neu. Setzt es in das ausgewählte Format und zeigt es im Konfigurationsformular an.

## BEISPIEL

```
LR_UpdateChart(12345);
```

# Virtuelle Geräte

Die virtuellen Geräte simulieren verschieden funktionsweisen von verschiedenen Geräten.

| | |
|---|---|
| E-Auto (Simulation) | Simuliert einen Ladevorgang eines E-Autos |
| Heizstab (Simulation) | Simuliert einen Heizstab |
| Licht (Simulation) | Simuliert ein Licht |
| Mediaplayer (Simulation) | Simuliert einen MediaPlayer |
| PV-Anlage (Simulation) | Simulation einer PV-Anlage |
| Rollladen (Simulation) | Simuliert einen Rollladen. |
| Thermostat (Simulation) | Simuliert ein Thermostat |
| Verbrauch/Kosten | Errechnet die Kosten nach Verbrauch |
| Zähler (Simulation) | Simuliert eine einstellbare Anzahl an Zählern. |

# E-Auto (Simulation)

Benötigt: IP-Symcon >= 6.0
Das E-Auto simuliert einen Ladevorgang eines E-Autos.

## FUNKTIONSUMFANG

Simuliert den Ladevorgang über eine Wallbox
Simuliert vereinfacht einen Ladevorgang mit vorgegebener Leistung

## SOFWARE-INSTALATION

Über den Module Store das 'Virtuelle Geräte'- Modul installieren

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'E-Auto (Simulation)'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Modus | Optionen: 'Simpel', in der nur die Leistung angegeben wird. 'Wallbox', mit umfangreicher Konfiguration um den Ladevorgang an einer Wallbox zu simulieren |
| Varianz | Angabe wie stark der Verbrauch schwanken soll |
| Intervall | Zeit zwischen den Aktualisierungen |

Modus: Simpel

| NAME | BESCHREIBUNG |
|---|---|
| Leistung | Maximale Leistung die gebraucht wird |

Modus: Wallbox

| NAME | BESCHREIBUNG |
|---|---|
| Phasen | Über wie viele Phasen die Wallbox läd |
| Teile die Phasen auf einzelne Variablen auf (Simuliert die Alfen Wallbox) | Aufteilung der Phasen in einzelne Variablen |
| Minimale Stromstäkre (pro Phase) | Stromstärke die minimal pro Phase anliegt |
| Maximale Stromstärke (pro Phase) | Stromstärke die maximal pro Phase anliegt |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

## STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Status | Boolean | Staus, ob der Ladevorgang momentan läuft |
| Intensität | Integer | Im Simpel-Modus verfügbar und schaltet die Stärke des Verbrauchs |
| Verbrauch | Float | Maximale Ladeleistung |
| Akkuladestand (Ist) | Integer | Aktueller Ladestand |
| Leistung (Soll) | Float | Schaltet den Verbrauch des Ladevorganges |
| Stromstärke | Float | Im Wallbox-Modus verfügbar und schaltet die aktuelle Stromstärke der Phasen. Kann sich auf mehrere Variablen aufteilen |

# Heizstab (Simulation)

Simuliert einen Heizstab.

## FUNKTIONSUMFANG

Simuliert einen Heizstab.

## SOFTWARE-INSTALLATION

Über den Module Store das 'Virtuelle Geräte'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Heizstab (Simulation)'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

Konfigurationsseite:

| NAME | BESCHREIBUNG |
|---|---|
| Leistung | Angabe wie viel der Heizstab verbrauchen soll |
| Varianz | Angabe wie stark der Verbrauch schwanken soll |
| Intervall | Zeitinterval der Aktualisierungen |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Status | Boolean | Schaltet den Heizstab an/aus |
| Intensität | Integer | Schaltet die Stärke des Heizstabes |
| Verbrauch | Float | Zeigt den aktuellen Verbrauch an |

# Licht (Simulation)

Simuliert ein Licht.

## FUNKTIONSUMFANG

Simuliert eine Licht

## SOFTWARE-INSTALLATION

Über den Module Store das 'Virtuelle Geräte'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Licht (Simulation)'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Status | Boolean | Schaltet das Licht an/aus. |
| Helligkeit | Integer | Schaltet die Helligkeit des Lichtes |
| Farbe | Integer | Schaltet die Farbe des Lichtes |
| Farbtemperatur | Integer | Schaltet die Farbtemperatur des Lichtes |

## VISUALISIERUNG

Zeigt in der Kachel-Visualisierung initial die Lichtkachel mit mehreren Möglichkeiten an.

# Mediaplayer (Simulation)

Simuliert einen Mediaplayer.

## FUNKTIONSUMFANG

Simuliert einen Mediaplayer

## SOFTWARE-INSTALLATION

Über den Module Store das 'Virtuelle Geräte'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Mediaplayer (Simulation)'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Fortschritt | Float | Zeigt den Fortschritt des Liedes an |
| Künstler | String | Zeigt den Künstler des aktuellen Liedes an |
| Lautstärke | Integer | Zeigt die Lautstärke an |
| Song | String | Zeigt das aktuelle Lied an |
| Stummschaltung | Boolean | Zeigt an ob die Ausgabe de-/aktiviert ist |
| Wiedergabe | Integer | Zeigt den momentanen Status der Wiedergabe an |
| Wiedergabeliste | String | JSON codierter String mit den nächsten Liedern und ihren Informationen |
| Wiederholen | Integer | Zeigt an ob die Playlist, Lied oder nichts wiederholt wird |
| Zufallswiedergabe | Boolean | Schaltet die zufällige Wiedergabe der Lieder ein/aus |
| Cover | Bild | Zeigt das Cover des Liedes |

## VISUALISIERUNG

Zeigt in der Kachel-Visualisierung die Mediaplayer-Kachel mit ihren Funktionen an.

# PV-Anlage (Simulation)

Simulation einer PV-Anlage

## FUNKTIONSUMFANG

Simulation einer PV-Anlage

## SOFTWARE-INSTALLATION

Über den Module Store das 'Virtuelle Geräte'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'PV-Anlage (Simulation)'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

Konfigurationsseite:

| NAME | BESCHREIBUNG |
|---|---|
| Leistung | Angabe wie viel die PV Anlage verbrauchen soll |

## STATUSVARIABLEN UND PROFILE

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Erzeugung | Float | Gibt an wie viel die PV Anlage erzeugt |

### PROFILE

| NAME | TYP |
|---|---|
| WattSlider.10 | float |

# Rollladen (Simulation)

Simuliert einen Rollladen.

## FUNKTIONSUMFANG

Simuliert einen Rollladen

## SOFTWARE-INSTALLATION

Über den Module Store das 'Virtuelle Geräte'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Rollo (Simulation)'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Jalousie | Integer | Höhe der Jalousie |
| Lamella | Integer | Höhe der Lamellen |

## VISULISIERUNG

In der Kachel-Visualisierung wir die Shutter-Kachel angezeigt.

# Thermostat (Simulation)

Simuliert ein Thermostat.

## FUNKTIONSUMFANG

Simuliert ein Thermostat

## SOFTWARE-INSTALLATION

Über den Module Store das 'Virtuelle Geräte'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Thermostat (Simulation)'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

Konfigurationsseite:

| NAME | BESCHREIBUNG |
|---|---|
| Erstelle geloggte Werte für die aktuelle Temperatur | |
| Stunde Wert | Tabelle mit den zu loggenden Werten |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Soll Temperatur | Float | Variable zum Einstellen der Temperatur |
| Ist Temperatur | Float | Momentane Temperatur |

## VISUALISIERUNG

In der Kachel-Visualisierung wird die Heizungskachel angezeigt.

# Verbrauch/Kosten

Errechnet die Kosten mit Verbrauch im Standby.

## FUNKTIONSUMFANG

Errechnen der Kosten anhand des gegebenen Verbrauchs

## SOFTWARE-INSTALLATION

Über den Module Store das 'Virtuelle Geräte'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Verbrauch/Kosten'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

## STATUSVARIABLEN UND PROFILE

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| StandBy Verbrauch | Float | Verbrauch im StandBy-Modus |
| Kosten pro kWh | Float | Kosten pro kWh |
| Gesamtkosten pro Jahr | Float | Gesamtkosten pro Jahr |

### PROFILE

| NAME | TYP |
|---|---|
| WattSlider | float |
| EuroCent | float |

# Zähler (Simulation)

Simuliert eine einstellbare Anzahl an Zählern.

## FUNKTIONSUMFANG

Simuliert mehrere Zähler

## SOFTWARE-INSTALLATION

Über den Module Store das 'Virtuelle Geräte'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Zähler (Simulation)'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

Konfigurationsseite:

| NAME | BESCHREIBUNG |
|---|---|
| Anzahl Zähler | Anzahl der Zähler, die simuliert werden soll. |
| Minimaler Wert pro Stunde | |
| Maximaler Wert pro Stunde | |
| Maximaler Abstand zwischen zwei folgenden Werten | |
| Verwende für jede Variable ein anderes Profil | Jeder Zähler erhält ein anderes Profil |
| Erstelle Graph | Erstellt einen Multigraphen |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Zähler x | Float | Eine Zähler -Variable. x Steht hier für die Nummer des Zählers |
| Graph | Multigraph | Ein Multigraph der Zähler |

# Zufällige Beleuchtung

Benötigt: IP-Symcon >= 5.0
Das Modul ermöglicht es den Farbwert von Lampen zufällig zwischen verschiedenen Farben zu schalten.

## FUNKTIONSUMFANG

Farbvariablen können in einer Liste hinzugefügt werden
Eine Liste mit Farben aus denen gewählt wird, welche erweitert werden kann
Wenn das Modul deaktiviert wird, werden die ausgewählten Farbvariablen auf den Wert vor der Aktivierung gesetzt
Einstellung des Intervalls in dem die Farben gewechselt werden sollen

## SOFTWARE-INSTALLATION

Über den Module Store das 'Zufällige Beleuchtung'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Zufällige Beleuchtung'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Farbvariablen | Liste mit Farbvariablen, welche geschaltet werden sollen |
| Farben | Farben von denen eine zufällig zum Schalten ausgewählt wird |
| Änderungsintervall | Abstand zwischen den einzelenen Schaltungen in Sekunden |
| Gleichzeitiges Schalten | Legt fest ob alle Farbvariablen auf die gleiche Farbe gesetzt werden oder auf verschiedene |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Aktiv | Boolean | Schaltet das Modul an oder aus |

# ZB_ChangeLight

Benötigt: IP-Symcon >= 5.0

```
boolean ZB_ChangeLight (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Wählt zufällige Farben aus und setzt die ausgewählten Variablen auf diese.

## BEISPIEL

```
ZB_ChangeLight(12345);
```

# Energie

Der Abschnitt befasst sich speziell mit allen Instanzen und Modulen rund ums Thema Energie. Von Automatisierungen bis zu vereinfachten Darstellungen gibt es hier umfangreiche Auswahl an Konfiguration und Einrichtungshilfen.

## ARBEITSZAHL

Berechnung des Wirkungsgrades

| | |
|---|---|
| ARZ_Calculation | Berechnet den Wirkungsgrad |

## BETRIEBSSTUNDENZÄHLER

Ermittelt und zeigt die Betriebszeit eines Gerätes an

| | |
|---|---|
| BSZ_Calculate | Die Betriebsstunden-Variable wird auf den errechneten Wert gesetzt |

## ENERGIE DASHBOARD

Eine Übersicht mit mehrern Diagrammen

## ENERGIE MANAGER

Module, welches die Verbraucher an und ausschaltet, wenn Energie übrig ist.

## ENERGIEVERTEILUNG

## ENERGIEZÄHLER

Berechnet den Verbrauch

| | |
|---|---|
| Energiezähler Impuls | Rechnet Zählerwerte in einen Momentanverbrauch um. |
| EZI_Update | Aktualisiert die berechneten Werte des EnergieZählerImpuls-Moduls mit der InstanzID InstanzID. |
| Energiezähler Strom | Rechnet aus Strom/Leistung den momentanen und kumlativen Stromverbrauch. |
| EZS_Update | Aktualisiert die berechneten Werte des EnergieZählerStrom-Moduls mit der InstanzID |

## STROMABRECHNUNGSMODUL

Das Modul liefert eine Kosten-Aufstellung ähnlich der Jahresabrechnung vom Energieversorger.

| | |
|---|---|
| SAM_UpdateCalculations | Berechnet alle angezeigten Werte |

## STROMPREIS

Liest die aktuellen/vorhergesagten Strompreise von aWATTar, Tibber oder Epex Spot DE aus.

| | |
|---|---|
| SPX_Update | Aktualisiert die Daten |

## VERBRAUCH IN KATEGORIE

Zeigt den Verbrauch in Prozent in Kategorie.

| | |
|---|---|
| VIK_CalculationConsumption | Berechnet die Kategorien neu |

## VERBRAUCH IN ZEITSPANNE

Berechnet den Verbrauch zwischen einem Start- und End-Datum

| | |
|---|---|
| VIZ_Calculate | Berechnet den Verbrauch zwischen Start- und End-Datum neu |

## VERBRAUCHSVERHALTEN

Das Verbrauchsverhalten-Modul zeigt den erwarteten Verbrauch an.

| | |
|---|---|
| VBV_UpdateCalculation | Die Funktion aktualisiert die prognostizierten Werte |

## VIRTUELLEMESSSTELLE

Rechnet positive Variablenänderungen nach einstellbaren Regeln mit Veränderungen einer Hauptvariable zusammen und addiert dieses mit dem Wert eine Variable.

| | |
|---|---|
| VM_Update | Berechnet die Steigung der sekundären Messstellen und addiert dieses mit dem Ergebnis |

## VIRTUELLER ZÄHLER

Eingabemaske für manuelle abgelesene Zählerstände.

| | |
|---|---|
| VZ_WriteNewCounterValue | Setzt nach Überprüfung den aktuellen Zählerstand |

## ZÄHLERSTAND (TAG)

Für ein ausgewähltes Datum kann der Zählerstand angezeigt werden

| | |
|---|---|
| ZST_Calculate | Ermittelt den Wert für das gewählte Datum neu |

## ZÄHLERÜBERLAUF

Zähler, welche überlaufen können als Zähler dargestellt werden.

# Arbeitszahl

Benötigt: IP-Symcon >= 6.3
Der Wirkungsgrad wird berechnet.

## FUNKTIONSUMFANG

Der Wirkungsgrad für die vergangenen 365 Tage
Der Wirkungsgrad für die vergangenen 30 Tage

## SOFTWARE-INSTALLATION

Über den Module Store das 'Arbeitszahl'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Arbeitszahl'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE

| NAME | BESCHREIBUNG |
|---|---|
| Wärmeenergie (kWh) | Variable, welche als Zähler geloggt ist |
| Elektrische Energie (kWh) | Variable, welche als Zähler geloggt ist |

## STATUSVARIABLEN

Die Statusvariablen werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| MAZ | Float | Steht für die Monatsarbeitszahl |
| JAZ | Float | Steht für die Jahresarbeitszahl |
| Ereignis | zyklisches Event | Event, damit der Wirkungsgrad um Mitternacht berechnet wird |

# ARZ_Calculation

Benötigt: IP-Symcon >= 6.3

```
boolean ARZ_Calculation (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Berechnet den Wirkungsgrad

## BEISPIEL

```
ARZ_Calculation(12345)
```

# Betriebsstundenzähler

Benötigt: IP-Symcon >= 5.2
Mithilfe des Betriebsstundenzähler-Moduls kann die Betriebszeit eines Gerätes ermittelt und angezeigt werden.

## FUNKTIONSUMFANG

Anzeige der Stunden, welche ein Gerät aktiv ist
Hinzufügen des Geräts durch eine Variable vom Typ Boolean
Anzeigen der Kosten für den aktuellen, den letzten Zeitraum und Vorhersage für das Ende des Zeitraums
Möglichkeit die Kosten mit einem dynamischen Preis zu berechnen

## SOFTWARE-INSTALLATION

Über den Module Store das 'Betriebsstundenzähler'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Betriebsstundenzähler'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE

| NAME | BESCHREIBUNG |
|---|---|
| Aktiv | Legt fest ob die Rechnung auf Basis des eingestellten Intervalls aktualisiert wird |
| Quelle | Die Variable vom Typ Boolean, welche den Aktivitätsstatus eines Gerätes anzeigt, wobei true als aktiv angesehen wird. Um die Betriebsstunden zu errechnen muss diese Variable geloggt sein |
| Stufe | Die Stufe legt den Beginn des Zeitraums fest welcher betrachtet wird (Beginn des Tages, Woche, Monat, Jahr) sowie die Aggregationsstufe |
| Aktualisierungsintervall | Das Intervall in Minuten in dem die Betriebszeit erneut berechnet wird |
| Kostenberechnung | Legt fest ob die Kostenberechnung ausgeführt werden |
| Preistyp | Option, ob der Preis fest gelegt ist oder dynamisch über eine Variable berechnet werden soll |
| Preis | Der Preis welcher pro Betriebsstunde berechnet wird |
| Berechnen | Berechnet die Betriebszeit mit allen angegebenen Parametern |

Ist im Preistyp die Option "Dynamisch" eingestellt, wird eine geloggte Variable gefordert, welche den Kostenpreis beinhaltet.

## STATUSVARIABLEN UND PROFILE

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
| --- | --- | --- |
| Betriebsstunden | float | Die berechneten Betriebsstunden der Quellvariable im ausgewählten Zeitraum |
| Kosten des jetzigen Zeitraums | float | Die berechneten Kosten des laufenden Zeitraums |
| Kosten des letzten Zeitraums | float | Die berechneten Kosten des letzten abgeschlossenen Zeitraums |
| Vorhersge zum Ende des Zeitraums | float | Vorhersage der Kosten zum Ende des laufenden Zeitraums |

### PROFILE

| NAME | TYP |
| --- | --- |
| BSZ.OperatingHours | float |

## VISUALISIERUNG

Im Webfront werden die Betriebsstunden angezeigt.

# BSZ_Calculate

Benötigt: IP-Symcon >= 5.2

```
void BSZ_Calculate (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Keine Rückgabe.

## BESCHREIBUNG

Die Betriebsstunden-Variable wird auf den errechneten Wert gesetzt.

## BEISPIEL

```
BSZ_Calculate(12345);
```

# Energie Dashboard

Benötigt: IP-Symcon >= 8.1
Das Energie Dashboard dient als Übersicht über mehrere Zähler mit mehreren Diagrammen.

Momentan ist das Energie-Dashboard noch in der Entwicklung und daher nur auf Anfrage per Mail zu enthalten.

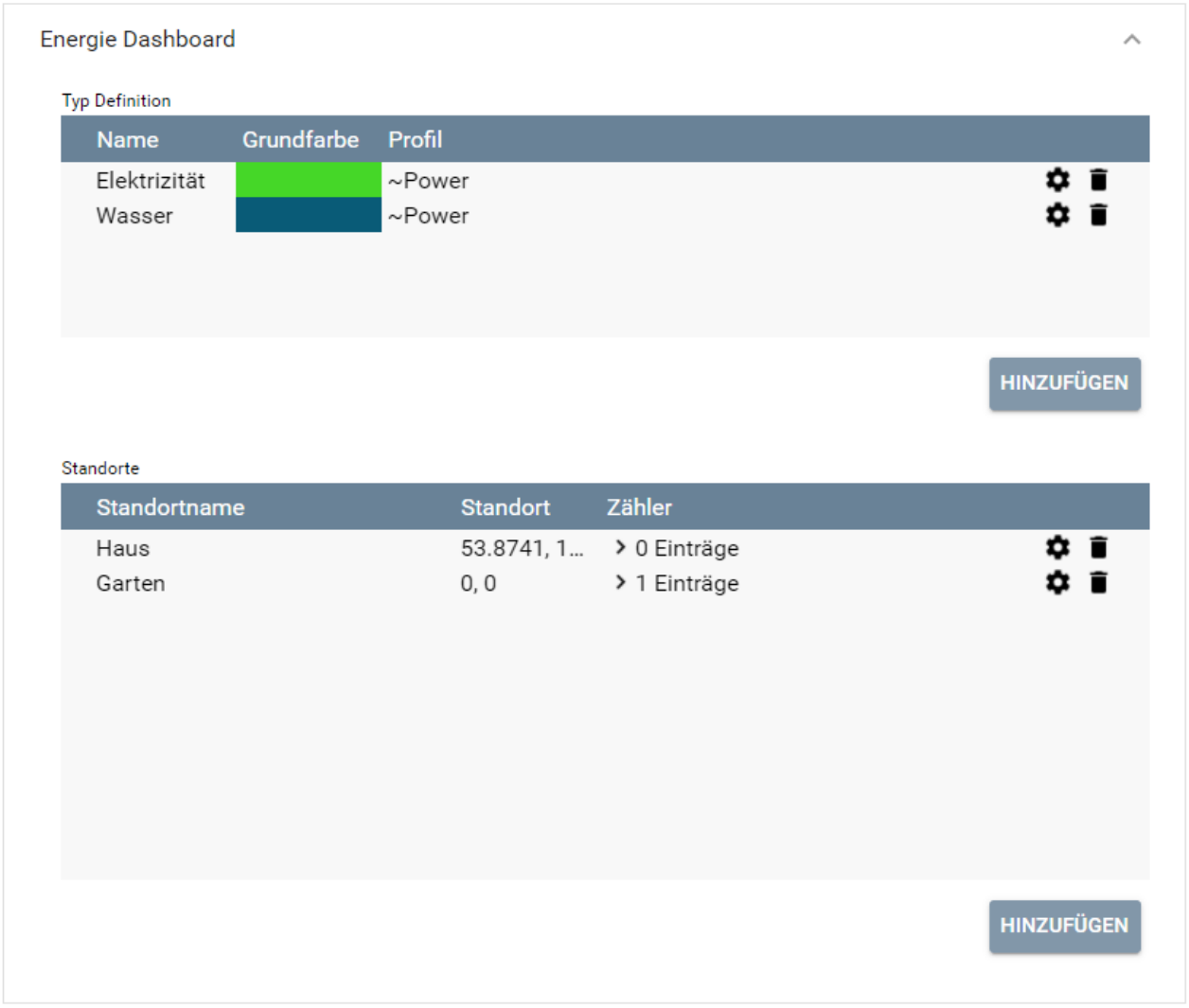
Nachfolgend werden die Enterprise-Features erklärt. Die Features für alle Nutzer sind noch in der Entwicklung.

## ERWERB DES ENERGIE-DASHBOARDS

Die Enterprise-Version des Energie-Dashboard kann als Zusatz für Symcon per Mail erworben werden.

## EINRICHTUNG DES ENERGIE-DASHBOARDS

Um das Energie-Dashboard zu nutzen, muss dieses zuerst eingerichtet werden. Hierfür wird in der Konsole die Instanz einer "Kachel-Visualisierung" aufgerufen. Hier sollte sichergestellt werden, dass im Abschnitt "Appbar" die Checkbox mit dem Label "Energie-Dashboard anzeigen" aktiviert ist. Damit wird in der Visualisierung der Blitz und somit der Zugang angezeigt.
Im Abschnitt "Energie-Dashboard" können nun die Typen und Areale definiert werden.

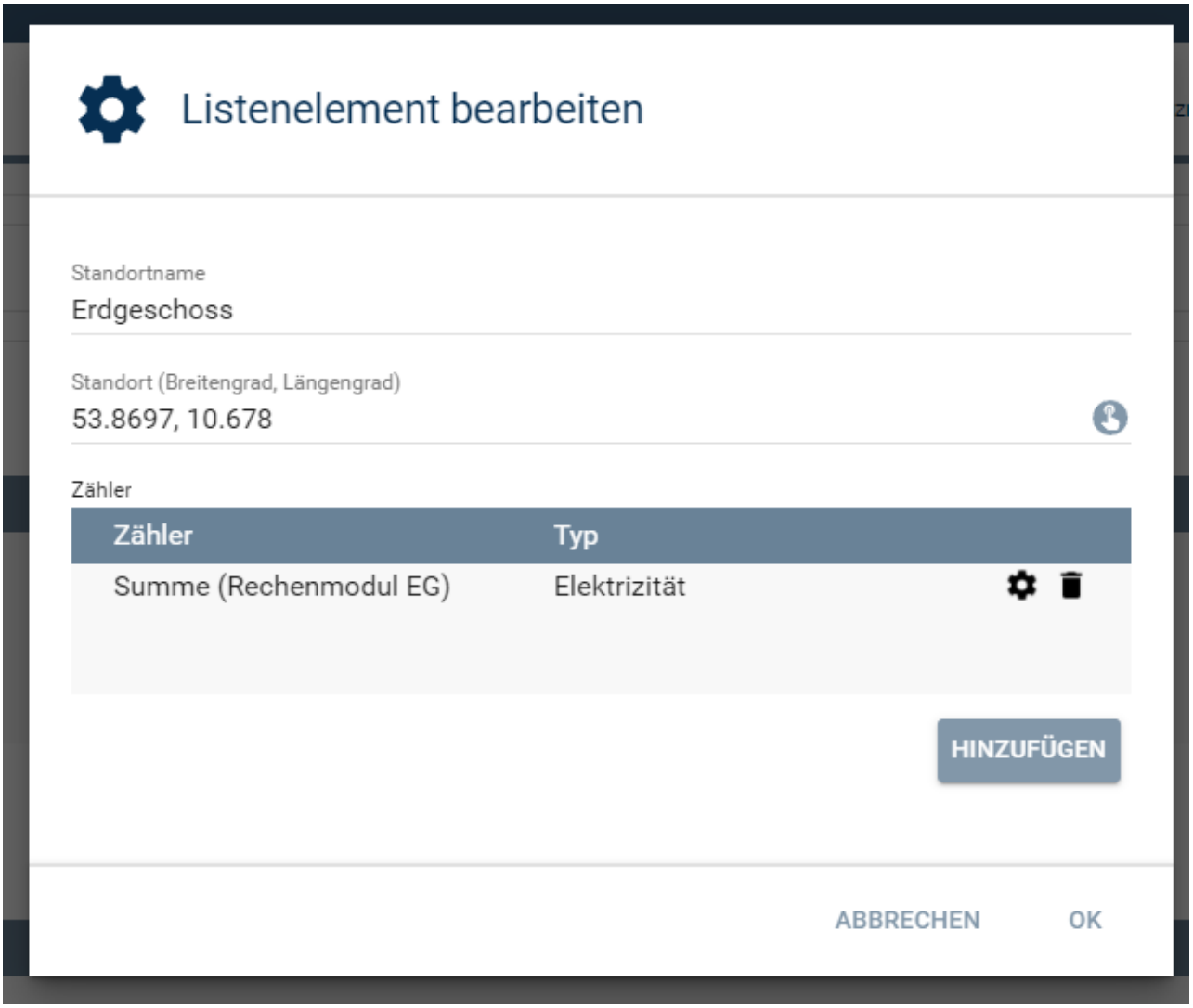



## DEFINITION DER TYPEN

Initial sind die Typen Elektrizität, Wasser und Gas definiert.

| NAME | ART | BESCHREIBUNG |
|---|---|---|
| Name | Freitext | Der Name des Typs, welcher durch den Nutzer frei angegeben werden kann. |
| Grundfarbe | Farbauswahl | Auswahl einer Grundfarbe für die Diagramme. Alle Areale des gleichen Typs werden in einer Variation der Farbe in den Diagrammen dargestellt. |
| Profil | Profilauswahl | Auswahl eines Profils. Hiermit können die Werte der Diagramme mit Suffix/Prefix formatiert werden. |

## KONFIGURATION DER AREALE




Die Areale werden als Baum aufgebaut. Mit einem Klick auf Hinzufügen wird ein Areal auf der obersten Ebene erstellt.
Mit einem Klick auf einen Areal und dann einen Klick auf Hinzufügen wird ein Areal unterhalb des ausgewählten Areals erstellt.
Jeder Areal hat folgende Konfigurationen:

| NAME | ART | BESCHREIBUNG |
|---|---|---|
| Arealname | Freitext | Frei vergebbarer Name für das Areal. Dieser wird später im Energie-Dashboard angezeigt. |
| Standort | Koordinaten Auswahl | Optionale Auswahl der Koordinaten des Areals. |
| Zähler | Liste | Konfiguration der relevanten Zähler |

## KONFIGURATION DER ZÄHLER

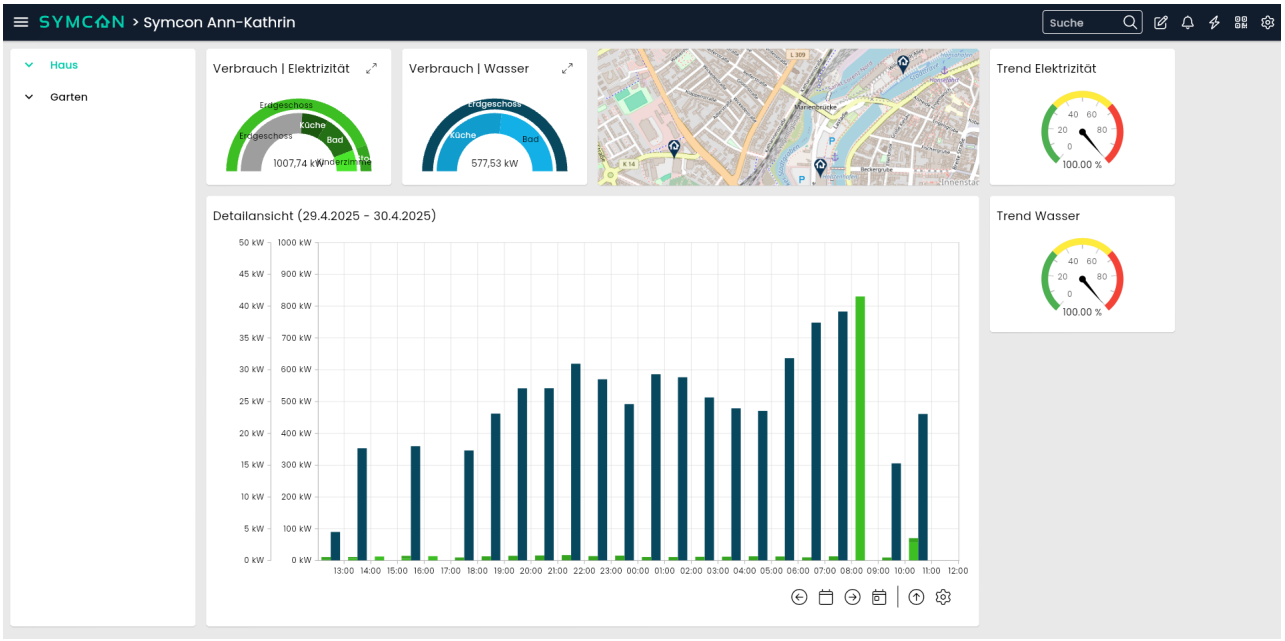
Die Zähler können optional hinzugefügt werden.

| NAME | ART | BESCHREIBUNG |
|---|---|---|
| Zähler | Variablenauswahl | Auswahl der Zählervariable, welche die aggregierten Werte für diesen Areal zur Verfügung stellt. |
| Typ | Typenauswahl | Der Typ des Zählers basierend auf den vorher definierten Typen |

# NUTZUNG DES ENERGIE-DASHBOARDS

Wurde das Energie-Dashboard aktiviert, so erscheint in der Appbar der Visualisierung ein Blitz-Symbol. Durch einen Klick darauf wir das Energie-Dashboard geöffnet.

## DAS ENERGIE-DASHBOARD




### BAUM

Der Baum spiegelt die Konfiguration der Areale durch Namen wieder.
Initial ist das erste Areal ausgewählt. Ist neben dem Areal ein Pfeil, so kann das Areal ausgeklappt werden, wodurch man auf untergeordnete Areale zugreifen kann.
Mit einem Klick auf die verschiedenen Areale werden die Werte der Diagramme sowie der Karte aktualisiert.

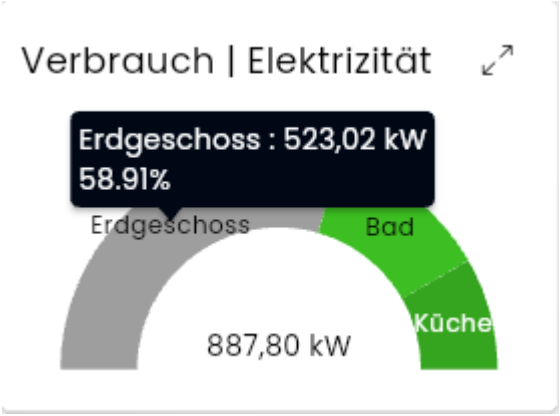
### DIAGRAMME

Beim Aufrufen des Energie-Dashboards wird der aktuelle Tag als Zeitraum und die Zeitspanne Stunde ausgewählt.

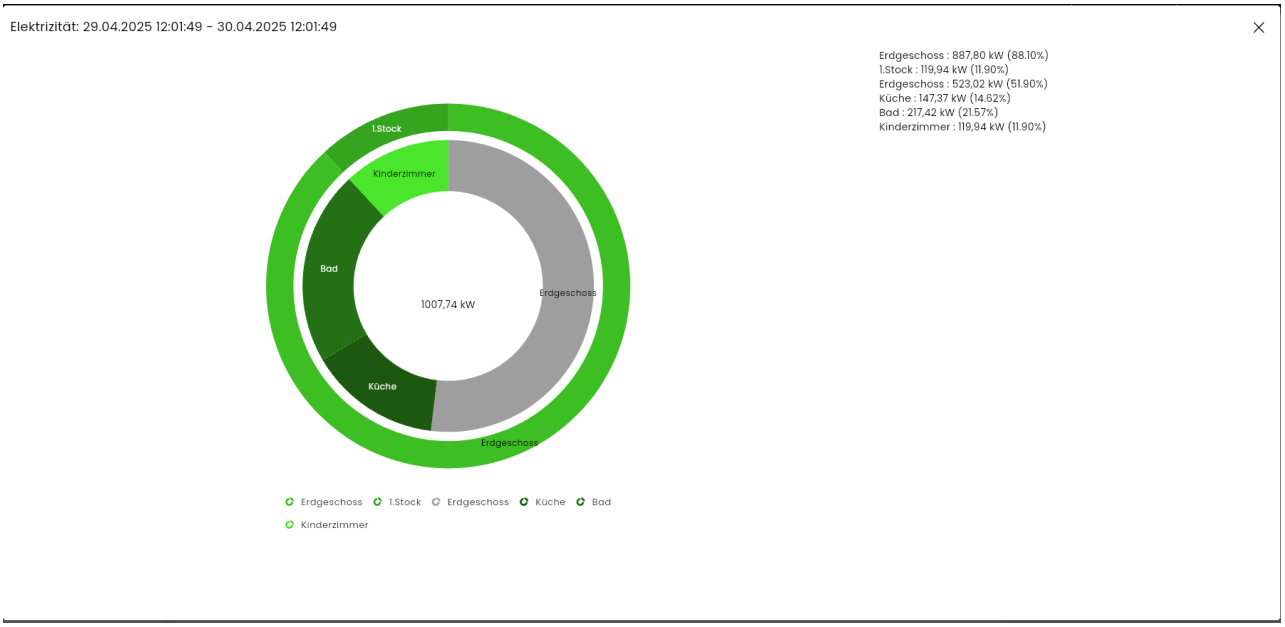
Detailansicht

In der Detailansicht wird der ausgewählte Zeitraum angezeigt.

Verbrauchsdiagramm




Im Verbrauchsdiagramm wird der ausgewählte Zeitraum mit der Zeitspanne vom Tag angezeigt. Der äußere Donut zeigt den ausgewählten Zähler, wenn nötigt, an, dazu die direkt untergeordneten Zähler. Der innere Donut wird angezeigt, wenn unterhalb des ausgewählten Areales noch eine Ebene vorhanden ist.




Mit einem Klick auf den maximieren Button wird der Doughnut vergrößert und eine detaillierte Auflistung der Einträge angezeigt.

Bedingte Anzeige der Zähler

Bei der Detailansicht, wie auch dem Doughnut-Diagramm werden Zähler bedingt angezeigt.

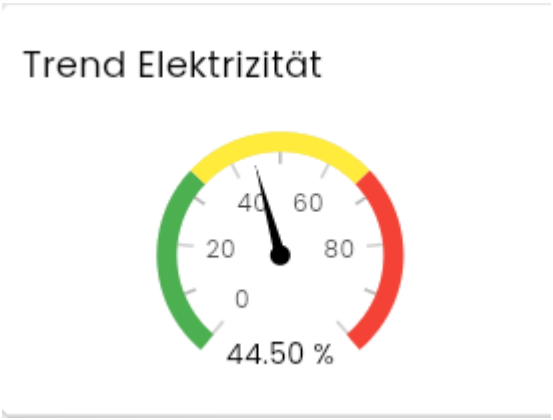
Hat das ausgewählte Areal einen Zähler und keine Areale unter sich, so wird dieser angezeigt.

Hat das ausgewählte Areal keinen Zähler und keine Areale unter sich, so wird dieser nicht angezeigt.

Hat das ausgewählte Areal Areale unter sich, so werden die unterhalbliegenden Arealzähler angezeigt.

Falls der Areal einen Zähler hat und dessen Summe nicht zu den untergeordneten Arealzählern passt, so wird zusätzlich zu den untergeordneten Arealzählern die Differenz in grau angezeigt.

Trend-Diagramm




Im Trend-Diagramm wird der prozentuale Zusammenhang zwischen dem obersten Areal und dem ausgewählten Areal dargestellt.

## KARTE




Die Karte listet die konfigurierten Koordinaten auf. Sind keine Koordinaten beim ausgewählten Areal konfiguriert, so werden die untergeordneten Areale nach Koordinaten durchsucht. Somit können auch mehrere Markierungen auf der Karte sein. Wird kein Areal gefunden wird ein Text anstatt der Karte angezeigt. Für die Karte wird OpenStreetMap genutzt.

# Energie Manager

Benötigt: IP-Symcon >= 6.3
Das Modul schaltet automatisch Variablen, wenn eine Quellenergie unterschritten ist.

## FUNKTIONSUMFANG

Anschalten eines Gerätes, wenn genügend Energie vorhanden ist
Ausschalten eines Gerätes, wenn nicht mehr genügend Energie vorhanden ist
Nächtliches Laden von Geräten um sicherzustellen, dass ein E-Auto oder ein Energiespeicher morgens geladen ist
Günstiges Laden um Geräte automatisch bei niedrigen Strompreisen zu aktivieren
Restriktion von Geräten konform zu §14a EnWG und §9 EEG
Verwaltung von Energiespeichern zum Ausgleichen von Defiziten
Verschiedene Rechenoptionen

## SOFTWARE-INSTALLATION

Über den Module Store das 'Energie Manager'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Energie Manager'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
| --- | --- |
| Modus | Auswahl der Rechenoption<br>**Absolut**: Geräte werden solange angeschaltet, bis die Quellenergie erreicht ist und abgeschaltet, wenn die Leistung die Quellenergie übersteigt<br>**Relativ**: Geräte werden angeschaltet, wenn die Quellenergie über null liegt und ausgeschaltet, wenn die Quellenergie unter null fällt |
| Hysterese | Toleranz bei der nicht geschalten wird |
| Konstellation für Überschuss/Defizit | Auswahl ob Überschuss und Defizit über eine geteilte oder zwei separate Variablen angegeben wird (nur bei Modus Relativ, ab Symcon 8.1) |
| Verfügbare Leistung (W) | Referenzvariable, welche die absolute verfügbare Leistung für den Manager beinhaltet (nur bei Modus Absolut) |
| Überschuss (W) | Referenzvariable, welche den aktuellen Überschuss beinhaltet (nur bei Modus Relativ) |
| Defizit (W) | Referenzvariable, welche den aktuellen Defizit beinhaltet (nur bei Modus Relativ und Konstellation "Zwei Variablen, eine pro Funktion", ab Symcon 8.1) |
| Invertieren | Auswahl ob Verfügbare Leistung bzw. Überschuss positiv oder negativ ist (nicht verfügbar, falls Modus Relativ und Konstellation für Überschuss Defizit auf "Zwei Variablen, eine pro Funktion" |
| Aktualisierungsmodus | **Nach Timer**: Der Zustand wird in einem festen Intervall aktualisiert<br>**Bei Änderung der Quelle**: Der Zustand wird immer aktualisiert, wenn Verfügbare Leistung bzw. Überschuss aktualisiert wird |
| Intervall | Aktualisierungsintervall des Zustands (nur bei Aktualisierungsmodus Nach Timer) |
| Energiepreise | Referenzvariable, welche die aktuellen Strompreise beinhaltet, bereitgestellt vom Modul Strompreis oder einem anderen Modul, welches dessen Format verwendet. Diese Variable wird nur für das Nachtladen bzw. Nächtliche Laufzeit benötigt. Ist sie nicht ausgewählt, kann Nachtladen bzw. Nächtliche Laufzeit nicht verwendet werden |
| Verbraucher | Liste aller Verbraucher, siehe Verbraucher |
| Energiespeicher | Liste aller Energiespeicher, siehe Energiespeicher |
| Elektrofahrzeuge | Liste aller Elektrofahrzeuge, siehe Elektrofahrzeuge |
| Konfiguration für §14a und §9 | Bereich um Restriktionen entsprechend §14a EnWG und §9 EEG zu konfigurieren, siehe Konfiguration für §14a und §9 (ab Symcon 8.1) |

## VERBRAUCHER

| NAME | BESCHREIBUNG |
|---|---|
| Verbraucher | Die Variable um den Verbraucher zu schalten. Ist die Variable vom Typ Integer oder Float, so versucht der Energie Manager diese bei Bedarf zu skalieren. Bei Boolean kann der Verbraucher lediglich An und Aus geschaltet werden. |
| Maximaler Verbrauch | Verbrauch in Watt, wenn der Verbraucher komplett angeschaltet ist |
| Aktueller Verbrauch (Optional) | Variable mit dem aktuellen Verbrauch des Geräts in Watt. Ist keine gesetzt, so wird der Verbrauch auf Basis der Variable Verbraucher und dem Maximalen Verbrauch berechnet (ab Symcon 8.1) |
| Name (Optional) | Der angezeigte Name des Verbrauchers. Ist kein Name angegeben, so wird der Name der Verbraucher Variablen verwendet |
| Bedingung (Optional) | Falls eine Bedingung angegeben ist, wird der Verbraucher nur aktiviert, falls die hier definierte Bedingung erfüllt ist. |
| Hinweis für Blockierung durch Bedingung (Optional) | Ist die Bedingung nicht erfüllt, wird dieser Text in der Visualisierung angezeigt. Ist kein Hinweis definiert, so wird ein Standardtext angezeigt. |
| Mindestlaufzeit | Wird der Verbraucher aktiviert, so läuft er mindestens die hier definierte Zeit in Sekunden. Erst nach Ablauf der Mindestlaufzeit nach der initialen Aktivierung kann der Energie Manager den Verbraucher wieder deaktivieren. Ist der Verbraucher skalierbar, so kann er bei nicht ausreichend vorhandener Energie passend skaliert werden, wird allerdings nie komplett deaktiviert. |
| Nachlaufzeit | Sollte nicht mehr genug Energie für den Verbraucher verfügbar sein, so wird er erst deaktiviert, falls die Energie für die hier definierte Dauer in Sekunden nicht vorhanden ist. So können kurze Einbrüche der verfügbaren Energie überbrückt werden. Ist der Verbraucher skalierbar, so kann er bei nicht ausreichend vorhandener Energie passend skaliert werden, wird allerdings nie komplett deaktiviert. |
| Falls Mindestlaufzeit/ Nachlaufzeit noch aktiv sind, aber die Bedingung nicht mehr erfüllt ist... | Definiere das Verhalten des Geräts falls die Bedingung während der Mindestlaufzeit oder Nachlaufzeit nicht mehr erfüllt ist<br>**...sofort anhalten**: Das Gerät wird bei der nächsten Aktualisierung des Zustands deaktiviert obwohl die Zeit noch läuft<br>**...erst anhalten sobald Mindestlaufzeit/Nachlaufzeit abgelaufen ist**: Die aktuelle Zeit läuft weiter ab. Erst nach Ablauf wird das Gerät deaktiviert. (nur falls sowohl eine Bedingung gesetzt als auch Mindest- oder Nachlaufzeit positiv sind, ab Symcon 8.1) |

## ENERGIESPEICHER

| NAME | BESCHREIBUNG |
| --- | --- |
| Vorlage | Auswahl einer Vorlage, über welche die technischen Parameter des Geräts automatisch eingetragen werden können. Wurde "Benutzerdefiniert" gewählt, so sind alle Parameter freigegeben und können manuell angepasst werden, ansonsten werden diese gesperrt und durch die Vorlage gesetzt (ab Symcon 8.1) |
| Instanz | Auswahl der Instanz passend zur gewählten Vorlage. Nach Wahl einer Instanz werden alle Parameter der Vorlage in der Konfiguration eingetragen. In manchen Vorlagen bleiben spezifische Parameter editierbar, da diese nicht explizit in der Vorlage enthalten sind. In diesem Fall erscheint eine entsprechende Information. (nur falls die Vorlage nicht auf "Benutzerdefiniert" steht, ab Symcon 8.1) |
| Energiespeicher schatbar? | Angabe ob der Energiespeicher über Variablen zum schalten verfügt<br>**Direkt schaltbar**: Die Variablen sind direkt schaltbar. Möchte der Energie Manager Anpassungen vornehmen, werden die entsprechenden Variablen geschaltet<br>**Durch Freilassen von Energie indirekt schalten**: Die Variablen sind nicht schaltbar, der aktuelle Zustand allerdings lesbar. Möchte der Energie Manager Änderungen am Energiespeicher vornehmen, so geschieht dies indirekt, beispielsweise indem bewusst Energie freigelassen und nicht verbraucht wird, sodass die Automatik des Energiespeichers diese zum Laden abgreift (ab Symcon 8.1) |
| Konstellation für Laden/ Entladen | Auswahl ob Laden und Entladen über eine geteilte oder zwei separate Variablen geschaltet werden (nur falls Energiespeicher schaltbar, ab Symcon 8.1) |
| Lade Variable | Die Variable um die Ladung des Energiespeichers zu schalten. Ist die Variable vom Typ Integer oder Float, so versucht der Energie Manager diese bei Bedarf zu skalieren. Bei Boolean kann die Ladung lediglich aktiviert oder deaktiviert werden. (nur falls Energiespeicher schaltbar ist) |
| Entlade Variable | Die Variable um die Entladung des Energiespeichers zu schalten. Ist die Variable vom Typ Integer oder Float, so versucht der Energie Manager diese bei Bedarf zu skalieren. Bei Boolean kann die Entladung lediglich aktiviert oder deaktiviert werden. (nur falls Energiespeicher schaltbar ist und Konstellation "Zwei Variablen, eine pro Funktion", ab Symcon 8.1) |
| Invertieren | Angabe ob Entladung als positiver oder negativer Variablenwert dargestellt wird. Ladung wird entsprechend durch das gegensätzliche Vorzeichen dargestellt. (nur falls Energiespeicher schaltbar ist und Konstellation "Eine kombinierte Variable für beides", ab Symcon 8.1) |
| Schalte Entladung um Einkauf von Energie zu verhindern | Ist dieser Schalter gesetzt, so aktiviert der Energie Manager die Entladung dieses Energiespeichers, falls es einen Defizit gibt und der Energiespeichers noch über Ladung verfügt (nur falls Energiespeicher schaltbar ist, ab Symcon 8.1) |
| Variable zum Umschalten zwischen Laden/Entladen | Falls das Gerät nicht nur über die Variable zum Laden/Entladen geschaltet wird, sondern zusätzlich über eine Statusvariable verfügt, welche auf einen spezifischen Wert für Laden oder Entladen geschaltet werden muss, so kann diese hier angegeben werden. (nur falls Energiespeicher schaltbar ist, ab Symcon 8.1) |
| Wert auf den die Variable zum Umschalten zwischen Laden/ | |

# ELEKTROFAHRZEUGE

| NAME | BESCHREIBUNG |
| --- | --- |
| Vorlage | Auswahl einer Vorlage, über welche die technischen Parameter des Geräts automatisch eingetragen werden können. Wurde "Benutzerdefiniert" gewählt, so sind alle Parameter freigegeben und können manuell angepasst werden, ansonsten werden diese gesperrt und durch die Vorlage gesetzt (ab Symcon 8.1) |
| Instanz | Auswahl der Instanz passend zur gewählten Vorlage. Nach Wahl einer Instanz werden alle Parameter der Vorlage in der Konfiguration eingetragen. In manchen Vorlagen bleiben spezifische Parameter editierbar, da diese nicht explizit in der Vorlage enthalten sind. In diesem Fall erscheint eine entsprechende Information. (nur falls die Vorlage nicht auf "Benutzerdefiniert" steht, ab Symcon 8.1) |
| Leistung (Soll) | Die Variable um die Sollleistung des Elektrofahrzeuges zu schalten. |
| Einheit für Leistung | Angabe ob die Sollleistung in Watt oder Ampere angegeben wird (ab Symcon 8.1) |
| Variable für An/Aus (Optional) | Falls das Gerät nicht nur über den Sollwert geschaltet wird, sondern zusätzlich über eine Variable verfügt welche das Gerät An oder Aus schaltet, so kann diese hier angegeben werden. (ab Symcon 8.1) |
| Phasen | Auswahl, wie viele Phasen das Elektrofahrzeug unterstützt. |
| Setze Leistung | Auswahl, ob die Leistung pro Phase oder insgesamt geschaltet wird (nur falls mindestens 2-phasig, ab Symcon 8.1) |
| Unterstütze Umschaltung zwischen 1-phasigem und mehr-phasigem Laden | Der Schalter gibt an, ob das Elektrofahrzeug eine Umschaltung der Phasen aus der obigen Auswahl zu einer Phase und zurück unterstützt. (nur falls mindestens 2-phasig) |
| Min. Ladestrom pro Phase | Minimaler Ladestrom pro Phase in Ampere |
| Max. Ladestrom pro Phase | Maximaler Ladestrom pro Phase in Ampere |
| Aktueller Verbrauch (Optional) | Variable mit dem aktuellen Verbrauch des Geräts in Watt. Ist keine gesetzt, so wird der Verbrauch in Höhe der Sollleistung angenommen (ab Symcon 8.1) |
| Name (Optional) | Der angezeigte Name des Elektrofahrzeuges. Ist kein Name angegeben, so wird der Name der Instanz zur verwendeten Vorlage verwendet. Wurde keine Vorlage benutzt, so wird der Name der Leistung Variablen verwendet |
| Verzögerung | Wird die Soll-Leistung des Elektrofahrzeuges durch den Energie Manager angepasst, so werden für die hier angegebene Dauer in Sekunden keine weiteren Schaltvorgänge durchgeführt. |
| Ladestand | Variable, welche den aktuellen Ladestand beinhaltet. Dabei wird der Minimalwert der Variable als komplett leer und der Maximalwert als komplett voll interpretiert. Somit können sowohl prozentuale als auch absolute Darstellungen verwendet werden. Die Variable wird nur fürs Nachtladen benötigt. Ist sie nicht gesetzt, so ist Nachtladen nicht möglich. |
| Kapazität | Die Kapazität des Elektrofahrzeuges in kWh. Dieser Wert wird lediglich fürs Nachtladen benötigt. Ist er nicht gesetzt, kann Nachtladen bei dem Elektrofahrzeug nicht verwendet werden. |

# KONFIGURATION FÜR §14A UND §9 (AB SYMCON 8.1)

| NAME | BESCHREIBUNG |
| --- | --- |
| Variable für Einschränkung (§14a) | Auswahl einer Variable vom Typ Boolean. Ist diese gesetzt, so wird eine Einschränkung entsprechend §14a EnWG vorgenommen und verifiziert. Während die Variable aktiv ist wird außerdem das gesamte Verhalten des Energie Managers in einer Mediendatei gespeichert |
| Variable für maximale Einspeisung (§9) | Auswahl einer Variable vom Typ Integer oder Float. Ist diese gesetzt, so gibt sie die maximale Einspeisung entsprechend §9 EEG an. Liegt eine Einschränkung vor, so wird der Energie Manager die Einspeisung einschränken um konform zu §9 EEG zu bleiben. |
| Energieproduktion | Liste aller Energieproduzenten, siehe Energieproduktion |
| Simuliere Einschränkungen nach §14a | Ist dieser Schalter gesetzt, so wird eine Einschränkung entsprechend §14a EnWG angewandt, unabhängig des tatsächlichen Wertes der Variable für Einschränkung |
| Aktueller Verbrauch | Auflistung aller Verbraucher und den relevanten Parametern nach §14a EnWg um zu verifizieren, dass sich der Energie Manager korrekt verhält |

### ENERGIEPRODUKTION (AB SYMCON 8.1)

| NAME | BESCHREIBUNG |
| --- | --- |
| Variable mit Energieproduktion | Variable, welche die aktuelle Energieproduktion des Geräts in Watt angibt |
| Variable für Einschränkung | Variable, über welche die maximale Einspeisung des Geräts in Watt eingeschränkt werden kann |
| Maximale Produktion | Maximale Produktion des Geräts in Watt |

## STATUSVARIABLEN UND PROFILE

Die Statusvariablen werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

Der Begriff "Gerät" fasst hier Verbraucher, Energiespeicher und Elektrofahrzeuge zusammen.

| NAME | TYP | BESCHREIBUNG |
| --- | --- | --- |
| Geplanter Verbrauch | Float | Der aktuelle vom Energie Manager berechnete Verbrauch aller eingetragenen Geräte |
| PMin (§14a) | Float | Der berechnete Wert PMin für Anwendung der Restriktion im Sinne von §14a EnWG (nur falls Variable für Einschränkung (§14a) gesetzt ist, ab Symcon 8.1) |
| Protokoll zu Einschränkungen (§14a/§9) | Medien - Dokument | Protokoll mit allen Operationen während eine Einschränkung durch §14a EnWG aktiv ist sowie sämtlichen Einschränkungen welche nach §9 EEG vorgenommen werden (ab Symcon 8.1) |

Alle weiteren Statusvariablen werden pro Gerät angelegt. Sind also 3 Geräte im Modul, werden je nach Konfiguration 12 bis 27 Variablen angelegt.

| NAME | TYP | BESCHREIBUNG |
| --- | --- | --- |
| Priorität | Integer | Reihenfolge in der die Geräte betrachtet werden, dabei werden niedrigere Prioritätswerte bevorzugt aktiv geschaltet |
| Status | Link | Verlinkung zum Status des Gerätes, also "Verbraucher" für Verbraucher, "Lade Variable" für Energiespeicher und "Leistung (Soll)" für Elektrofahrzeuge (nicht für nicht schaltbare Energiespeicher) |
| Modus | Integer | Bei "Automatisch" wird das Gerät vom Energie Manager verwaltet, ansonsten kann es hiermit übersteuert werden |
| Aktueller Verbrauch | Link | Verlinkung zur Variable mit dem aktuellen Verbrauchs des Gerätes (nur wenn Variable für das Gerät gesetzt wurde, ab Symcon 8.1) |
| Ladung | Link | Verlinkung zur Variable mit dem aktuellen Ladestand des Energiespeichers (nur wenn Variable für den Energiespeicher gesetzt wurde, ab Symcon 8.1) |
| Betriebsmodus | Integer | Angabe ob der Energiespeicher aktuell lädt, entlädt oder passiv ist (nur für Energiespeicher, ab Symcon 8.1) |
| Bedingung | String | Aktueller Zustand der Bedingung des Gerätes |
| Gesperrt | Boolean | Ist diese Variable gesetzt, so ist das Gerät aufgrund von Mindestlaufzeit oder Nachlaufzeit gesperrt und darf nicht deaktiviert werden. Die Variable wird nur erstellt, falls das Gerät eine Mindestlaufzeit oder Nachlaufzeit hat. |
| Gesperrt bis | Integer | Ist das Gerät gesperrt, so beinhaltet diese Variable den Zeitpunkt, wann diese Sperrung wieder aufgehoben wird. Die Variable wird nur erstellt, falls das Gerät eine Mindestlaufzeit oder Nachlaufzeit hat. |
| Nachtladen/ Nächtliche Laufzeit | Boolean | Aktiviert Nachtladen (für Energiespeicher und Elektrofahrzeug) bzw. Nächtliche Laufzeit (für Verbraucher). Diese Variable wird nur erstellt, falls das Gerät Nachtladen bzw. Nächtliche Laufzeit unterstützt. |
| Nachtladen: Reichweite | Float | Gewünschte Reichweite nach dem Nachtladen für das Elektrofahrzeug. Diese Variable wird nur für Elektrofahrzeuge erstellt, die Nachtladen unterstützen und für die das Ziel zum Nachtladen in Kilometer angegeben werden soll. |
| Nachtladen: Reichweite verfügbar um | Integer | Uhrzeit zu welcher die gewünschte Reichweite am nächsten Tag verfügbar sein soll. Diese Variable wird nur für Elektrofahrzeuge erstellt, die Nachtladen unterstützen und für die das Ziel zum Nachtladen in Kilometer angegeben werden soll. |
| Nachtladen: Ladung | Float | Gewünschter prozentualer Füllstand nach dem Nachtladen für den Energiespeicher oder das Elektrofahrzeug. Diese Variable wird nur für Energiespeicher oder Elektrofahrzeuge erstellt, die Nachtladen unterstützen. Elektrofahrzeugen müssen zusätzlich so konfiguriert sein, dass das Ziel zum Nachtladen in Prozent angegeben werden soll. |
| Nachtladen: Ladung | | Uhrzeit zu welcher die gewünschte Ladung am nächsten Tag verfügbar sein soll. Diese Variable wird nur für Energiespeicher oder Elektrofahrzeuge erstellt, die |

### PROFILE

| NAME | TYP |
|---|---|
| EO.Mode | Integer |
| EO.Priority | Integer |
| EO.Range | Float |
| EO.Percentage | Float |
| EO.Duration | Float |
| EO.Cents | Float |
| EO.BatteryMode | Float |

Assoziationen EO.Mode

| NAME | BESCHREIBUNG |
|---|---|
| Aktiv | Die Variable wird aktiv geschaltet und wird von der Betrachtung ausgeschlossen |
| Inaktiv | Die Variable wird ausgeschaltet und wird von der Betrachtung ausgeschlossen |
| Automatisch | Die Variable wird je nach Priorität und Energie geschaltet. |

## FUNKTIONSWEISE

### OPTIMIERUNG

Während des Betriebs des Energie Managers wird die Energie in einzelnen Schritten optimiert, sodass die produzierte Energie möglichst sinnvoll verwendet wird. Je nach Konfiguration erfolgen Schritte in einem festen Zeitintervall oder bei Aktualisierung der verfügbaren Leistung. In jedem Schritt wird nur ein Gerät geschaltet, damit Differenzen zwischen eingetragenen und tatsächlichen Verbräuchen früh erkannt und gehändelt

werden können. Im Energie Manager kann jedes Gerät durch den Benutzer mit einer Priorität eingestuft werden.
In jedem Schritt wird initial geprüft, ob es eine "Lücke" bei den aktivierten Geräten gibt, also ob Geräte mit niedrigerer Priorität aktiv sind, obwohl Geräte mit höherer Priorität noch inaktiv sind und genug Energie zur Verfügung stehen würde um sie zu aktivieren. In diesem Fall wird ein Gerät mit niedrigerer Priorität deaktiviert um die Energie im nächsten Schritt für Geräte mit höherer Priorität freizugeben.
Ansonsten wird der aktuelle Überschuss ermittelt. Ist dieser positiv, so wird ein Gerät mit höchster Priorität ermittelt, welches den Überschuss aufnehmen kann. Das Gerät wird dann entsprechend aktiviert oder skaliert.
Ist der Überschuss negativ, so wird ein Gerät mit der niedrigsten Priorität deaktiviert oder herunter skaliert um die fehlende Energie auszugleichen.

## NACHTLADEN/NÄCHTLICHER ÜBERSCHUSS

Nachtladen bzw. Nächtlicher Überschuss wird betrachtet, sobald der Tag laut Location Control endet. Ist für Geräte Nachtladen bzw. Nächtlicher Überschuss aktiviert, so berechnet der Energie Manager, wie lange das entsprechende Gerät aktiviert sein muss. Diese Dauer wird mit den Energiepreisen vom aktuellen Zeitpunkt bis zur entsprechenden Zielzeit verglichen und bestimmt, zu welchen Zeitpunkten diese Laufzeit am günstigsten genutzt werden kann. Liegt bei einem Optimierungsschritt solch ein günstigster Preis vor, so wird das Gerät auf maximaler Skalierung aktiviert, ansonsten wird es deaktiviert.

## GÜNSTIGES LADEN/GÜNSTIGE LAUFZEIT

Günstiges Laden bzw. Günstige Laufzeit kann genutzt werden, um ein Gerät zu aktivieren, sobald der Strompreis unter einen selbst gewählten Wert fällt. Ist bei einer Aktualisierung der Strompreis ausreichend niedrig, so wird primär eines der betroffenen Geräte aktiviert, unabhängig von dessen eingestellter Priorität und des aktuellen Überschusses. Erst wenn alle entsprechenden Geräte bereits aktiv sind, so wird in einem Aktualisierungsdurchlauf wie gewohnt vorgegangen.

## EINSCHRÄNKUNG NACH §14A ENWG

Wird die Einschränkung nach §14a aktiv, so wird sofort eine Aktualisierung durchgeführt. Diese erfolgt zusätzlich zur üblichen Aktualisierung im Intervalls bzw. bei Aktualisierung der Quellvariable. Ist bei einer Aktualisierung die Einschränkung aktiv, so wird der erlaubte Maximalverbrauch "PMin" für Energiespeicher und Elektrofahrzeuge berechnet. Verbrauchen die Energiespeicher und Elektrofahrzeuge aktuell mehr, so werden diese umgehend herunter geregelt. Hierbei werden bevorzugt Geräte deaktiviert, die nicht per Mindestlaufzeit oder Nachlaufzeit gesperrt sind. Sollte dies allerdings nicht ausreichen, so werden auch Geräte mit aktiver Mindestlaufzeit oder Nachlaufzeit deaktiviert. Für Geräte, welche nicht im automatischen Modus sind, wird der Wert vor der Einschränkung abgespeichert und wiederhergestellt sobald die Einschränkung endet. Bei einem Versuch Energiespeicher oder Elektrofahrzeuge über PMin aus manuell zu aktivieren, werden die Schaltungen rückgängig gemacht. Sie werden allerdings gespeichert, damit diese bei einem Ende der Einschränkung auf den gewünschten Wert geschaltet werden. Für mehr Informationen zu §14a EnWG, siehe hier.

## EINSCHRÄNKUNG NACH §9 EEG

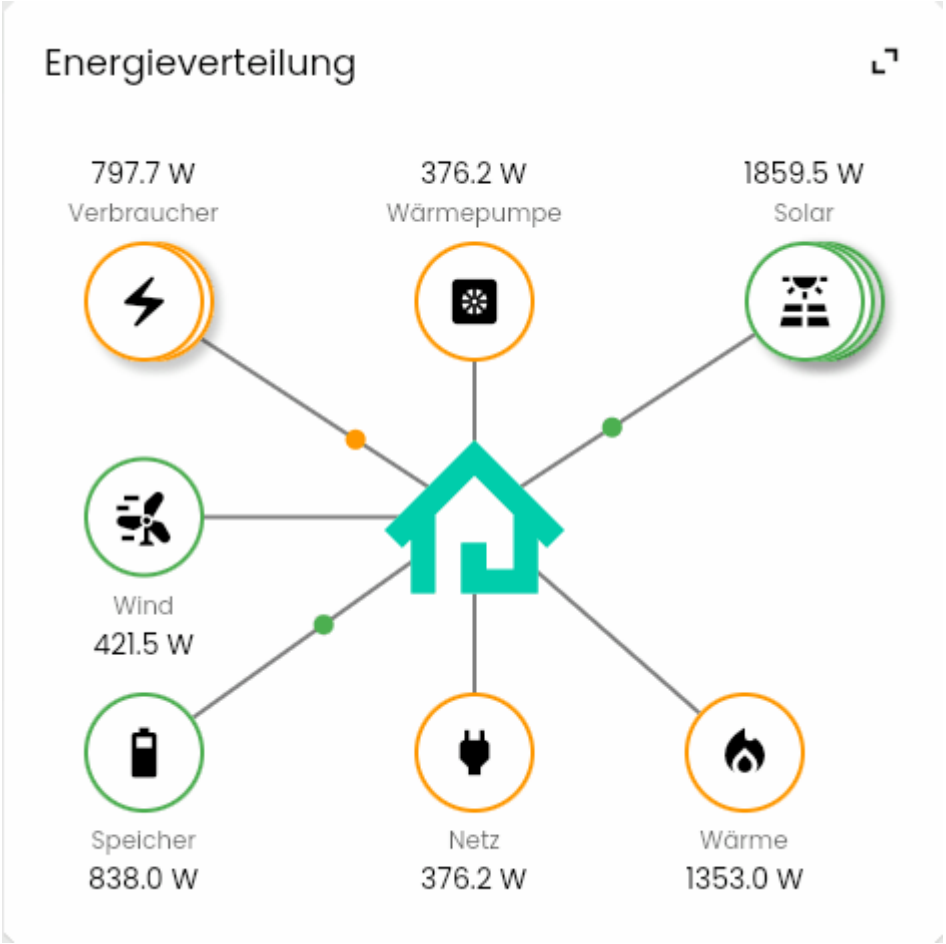
Ist eine Einschränkung nach §9 gegeben, diese also unter 100%, so werden die Energieproduzenten im Anschluss jeden Aktualisierungsschrittes entsprechend eingeschränkt. Hierfür wird die maximale erlaubte Energieproduktion berechnet. Diese setzt sich aus der erlaubten Einspeisung, also der Einschränkungswert multipliziert mit der maximalen Gesamtproduktion, sowie der aktuell selbst verbrauchten Energie zusammen, welche über alle Geräte berechnet wird. Ist die aktuelle Einschränkung der Energieproduzenten über der erlaubten Einschränkung, so wird diese reduziert. Hierbei wird bevorzugt eine Einschränkung bei Geräten angewandt, welche sowieso nicht produzieren. Reicht dies nicht aus, so werden Energieproduzenten eingeschränkt, die aktuell Energie produzieren. Für mehr Informationen zu §9 EEG, siehe hier.

## VISUALISIERUNG

Der Energie Manager verfügt über eine eigene Darstellung in der Visualisierung.

# Energieverteilung

Benötigt: IP-Symcon >= 7.0
Die Energieverteilung bietet eine Darstellung, welche einen Überblick über erzeugte/ verbrauchte und eingespeiste/bezogene Energie ermöglicht.




## FUNKTIONSUMFANG

Grafische Darstellung des Energieflusses zwischen verschiedenen Geräten
Aufteilung von Geräten nach Erzeuger und Verbraucher
Spezielle Darstellungen für Netz und Speicher
Berechnung von Gesamtverbrauch, Gesamterzeugung, Bezug und Überschuss
Gruppierung mehrerer Geräte des gleichen Typs

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Energieverteilung'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE

Alle ausgewählten Variablen sollten die Werte in der gleichen Einheit anbieten, um eine fehlerfreie Darstellung und Funktion zu gewährleisten.

| NAME | BESCHREIBUNG |
|---|---|
| Zusammenfassende Werte berechnen | Wenn aktiviert, werden Statusvariablen erstellt, die anhand der hinzugefügten Variablen Erzeugungs-, Verbrauchs-, Bezugs- und Überschusswerte berechnen. |

## KONFIGURATIONS-LISTE

| NAME | BESCHREIBUNG |
|---|---|
| Typ | Die Auswahl des Typs entscheidet ob das Gerät als Erzeuger oder Verbraucher behandelt wird und bietet entsprechend weitere Konfigurationsmöglichkeiten |
| Name | Der für die Darstellung genutzte Name. Bleibt dieser leer, wird der Variablenname genutzt. |
| Gruppieren | Alle Geräte des selben Typs werden in der Visualisierung zusammengefasst. |
| Zusatzinformation | Ein Variable deren Wert hinter dem eigentlichen Variablenwert in Klammern angezeigt werden kann. Beispielsweise der prozentuale Ladestand eines E-Fahrzeuges oder ein Zählerstand |
| Berechnungstyp | Aus den Werten der Zusatzinformation Variablen einer Gruppe wird entweder die Summe oder der Mittelwert gebildet. (nur wenn 'Gruppieren' aktiviert ist) |

## TYP-KONFIGURATION

Wind/Solar

| NAME | BESCHREIBUNG |
|---|---|
| Erzeugung | Die Variable, welche die erzeuge Energie angibt. |

Verbraucher/Wallbox/Wärme/Wärmepumpe

| NAME | BESCHREIBUNG |
|---|---|
| Verbrauch | Die Variable, welche die verbrauchte Energie angibt. |

Netz

| NAME | BESCHREIBUNG |
|---|---|
| Konfigurationstyp | Bestimmt ob der Wert sich aus einer oder zwei Variablen ergibt. |
| Bezug / Überschuss | Ist der Wert der Variable positiv so wird er als Bezug gezählt. Ist er negativ gilt zählt er zu Überschuss. (nur bei Konfigurationstyp 'Einzelne Variable') |
| Invertiert | Invertiert das Verhalten der einzelnen Variable. (nur bei 'Konfigurationstyp' 'Einzelne Variable') |
| Bezug | Die Variable, welche die bezogene Energie angibt. (nur bei 'Konfigurationstyp' 'Zwei Variablen') |
| Überschuss | Die Variable, welche die überschüssige Energie angibt. (nur bei 'Konfigurationstyp' 'Zwei Variablen') |

Speicher

| NAME | BESCHREIBUNG |
|---|---|
| Konfigurationstyp | Bestimmt, ob der Wert sich aus einer oder zwei Variablen ergibt. |
| Laden / Entladen | Ist der Wert der Variable positiv so wird er als Laden gezählt. Ist er negativ zählt er als Entladen. (nur bei Konfigurationstyp 'Einzelne Variable') |
| Invertiert | Invertiert das Verhalten der einzelnen Variable. (nur bei 'Konfigurationstyp' 'Einzelne Variable') |
| Laden | Die Variable, welche die geladene Energie angibt. (nur bei 'Konfigurationstyp' 'Zwei Variablen') |
| Entladen | Die Variable, welche die entladene Energie angibt. (nur bei 'Konfigurationstyp' 'Zwei Variablen') |

## STATUSVARIABLEN

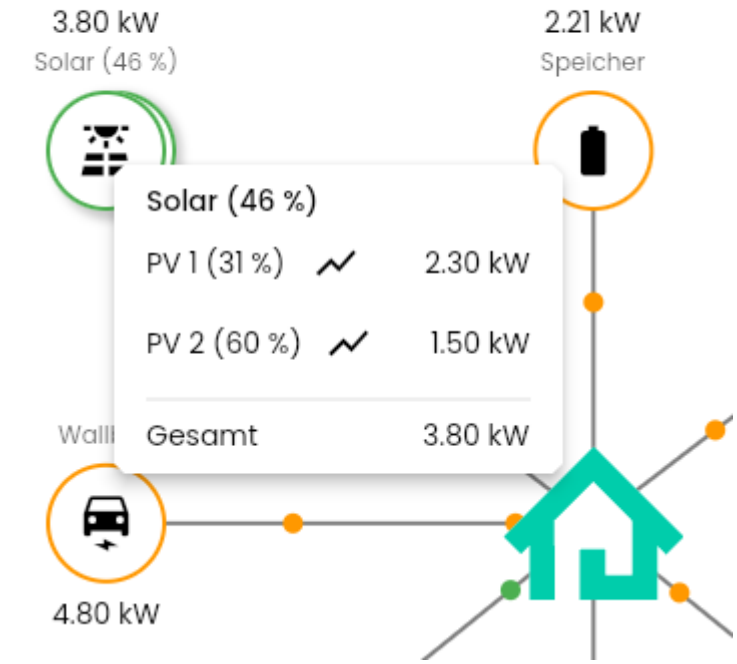
Die Statusvariablen werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

Die Statusvariablen werden nur angelegt, wenn die entsprechende Einstellung auf der Konfigurationsseite aktiviert wurde

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Verbrauch (Gesamt) | Float | Die Summe der Verbräuche des Typs Verbraucher, Wallbox, Wärmepumpe und Wärme. |
| Erzeugung (Gesamt) | Float | Die Summe der Erzeugung des Typs Solar und Wind. |
| Bezug | Float | Differenz zwischen verbrauchter Energie und erzeugter Energie. |
| Überschuss | Float | Differenz zwischen erzeugter Energie und verbrauchter Energie. |

# VISUALISIERUNG




Jedes Gerät und jede Gruppe kann angeklickt werden, um eine Aufschlüsselung der einzelnen Werte anzuzeigen. Sind die gewählten Variablen geloggt kann das entsprechende Diagramm direkt geöffnet werden.
Über einen Klick auf das Haus in der Mitte kann bei bedarf ein Ressourcen schonender Modus aktiviert werden, der die bewegten Punkte durch statische Pfeile ersetzt.

Die Energieverteilung hat im WebFront keine eigene Darstellung

# Energiezähler

Berechnet den Verbrauch

| | |
|---|---|
| Energiezähler Impuls | Rechnet Zählerwerte in einen Momentanverbrauch um. |
| EZI_Update | Aktualisiert die berechneten Werte des EnergieZählerImpuls-Moduls mit der InstanzID InstanzID. |
| Energiezähler Strom | Rechnet aus Strom/Leistung den momentanen und kumlativen Stromverbrauch. |
| EZS_Update | Aktualisiert die berechneten Werte des EnergieZählerStrom-Moduls mit der InstanzID |

# Energiezähler Impuls

Benötigt: IP-Symcon >= 4.2
Das Modul berechnet via eines Stromzählers (z.B. S0-Anschluss) den momentanen und kumlativen Stromverbrauch.

## FUNKTIONSUMFANG

Berechnet den momentanen Stromverbrauch in Watt und den gesamten Stromverbrauch in kWh.
Einstellbarkeit der Impulse des zählenden Geräts.
Einstellbarkeit der Intervalhäufigkeit in Sekunden zur Neuberechnung des Verbrauchs

## SOFTWARE-INSTALLATION

Über den Module Store das Modul Energiezähler installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'EnergiezählerImpuls'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Quelle | Quellvariable (Impulszähler), welche für die Berechnung genutzt werden soll. |
| Impulse | Wieviele Impulse das Gerät pro Kilowatt sendet. Dies muss der jeweiligen Betriebsanleitung des Geräts entnommen werden. |
| Intervall | In welchem Sekunden-Intervall automatisch neuberechnet werden soll. (Hinweis: Zu kurz eingestellte Intervalle (schneller als einkommende Impulse), können zu starken Schwankungen bei der Angabe des aktuellen Verbrauch führen. Es hat keinen Einfluss auf den berechneten Gesamtverbrauch. => Empfehlung: mind. 300 Sekunden) |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
| --- | --- | --- |
| Current | Float | Angabe in W |
| Counter | Float | Angabe in kWh |
| Last Value (Temporary) | Float | Hilfsvariable für letzten Wert. Wird zur Differenzberechnung benötigt. |

# EZI_Update

Benötigt: IP-Symcon >= 4.2

```
boolean EZI_Update (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Aktualisiert die berechneten Werte des EnergieZählerImpuls-Moduls mit der InstanzID **InstanzID**.

## BEISPIEL

```
EZI_Update(12345);
```

# Energiezähler Strom

Benötigt: IP-Symcon >= 4.2
Das Modul berechnet via eines Stromzählers (Strom oder Leistung) den momentanen und kumulativen Stromverbrauch.

## FUNKTIONSUMFANG

Berechnet den momentanen Stromverbrauch in Watt und den gesamten Stromverbrauch in kWh.
Einstellbarkeit von Typ und Quelle des zählenden Geräts.
Einstellbarkeit der Intervalhäufigkeit in Sekunden zur Neuberechnung des Verbrauchs, sofern sich die Quellvariable nicht geändert haben sollte.

## SOFTWARE-INSTALLATION

Über den Module Store das Modul Energiezähler installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'EnergiezählerStrom'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Typ | Ist die Quellvariable vom Typ Leistung (W) oder Strom (A). |
| Quelle | Quellvariable, welche für die Berechnung genutzt werden soll. |
| Spannung | Spannung (V), welche beim Umrechnen angenommen werden soll. |
| Intervall | In welchem Sekunden-Intervall automatisch neuberechnet werden soll, sofern sich die Quellvariable nicht geändert haben sollte. |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | BESCHREIBUNG |
| --- | --- |
| Typ | Ist die Quellvariable vom Typ Leistung (W) oder Strom (A). |
| Quelle | Quellvariable, welche für die Berechnung genutzt werden soll. |
| Spannung | Spannung (V), welche beim Umrechnen angenommen werden soll. |
| Intervall | In welchem Sekunden-Intervall automatisch neuberechnet werden soll, sofern sich die Quellvariable nicht geändert haben sollte. |

# EZS_Update

Benötigt: IP-Symcon >= 4.2

```
boolean EZS_Update (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Aktualisiert die berechneten Werte des EnergieZählerStrom-Moduls mit der InstanzID **InstanzID**.

## BEISPIEL

```
EZS_Update(12345);
```

# Stromabrechnungsmodul

Benötigt: IP-Symcon >= 5.0
Das Modul liefert eine Kosten-Aufstellung ähnlich der Jahresabrechnung vom Energieversorger.

## FUNKTIONSUMFANG

Erzeugt eine Kosten-Aufstellung mithilfe einiger Eckdaten.

## SOFTWARE-INSTALLATION

Über den Module Store das Modul "Stromabrechnung"-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'StromAbrechnung'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Quelle | Die Variable des Hauptzählers, welche als Zähler geloggt wird |
| Grundpreis | Der Grundpreis |
| Arbeitspreis | Der Arbeitspreis |
| Ablesedatum | Datum der letzen Zählerstandsablesung |
| letzter Zählerstand | Der Zählerstand am lezten Ablesetermin |
| geplanter Verbrauch | Der geplante Stromverbrauch/Jahr |

## STATUSVARIABLEN UND PROFILE

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Energiepreis | Float | Energiepreis mit einbezug des Grundpreises |
| Tage bis zur nächsten Ablesung | Integer | Tage bis zur nächsten Ablesung (1 Jahr ausgehend von der letzten Ablesung) |
| Tage seit letzter Ablesung | Integer | Tage seit letzter Ablesung |
| Zählerstand(Soll) | Float | Der Soll-Wert des heutigen Zählerstandes basierend auf dem geplanten Verbrauch |
| Geplanter Verbrauch/Tag | Float | Der Geplante Verbrauch pro Tag basierend auf dem geplanten Jahresverbrauch |
| durchschnittlicher Verbrauch der letzten 30 Tage | Float | Der Mittelwert des Verbrauches der letzten 30 Tage |
| Abweichung | Float | Die Abweichung des tatsächlichen Zählerstandes zum Soll-Wert |
| Gutschrift/Rückzahlung | Float | Menge der Rückzahlung bzw. Gutschrift berechnet durch die Abweichung |

**PROFILE:**

| NAME | TYP |
|---|---|
| SAM.EuroRating | Float |
| SAM.PowerPrice | Float |
| SAM.Calendar | Integer |

## VISUALISIERUNG

Hier werden alle wichtigen errechneten Werte angezeigt.

# SAM_UpdateCalculations

Benötigt: IP-Symcon >= 5.0

```
boolean SAM_UpdateCalculations (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Berechnet alle im Webfront angezeigten Werte basierend auf den Daten auf der Konfigurationsseite.

## BEISPIEL

```
SAM_UpdateCalculations(12345);
```

# Strompreis

Benötigt: IP-Symcon >= 7.1
Liest die aktuellen/vorhergesagten Strompreise von aWATTar, Tibber oder Epex Spot DE aus.

## FUNKTIONSUMFANG

Auslesen von Strompreisen verschiedener Anbieter
Visueller Verlauf der Marktdaten
Manuelle Eingabe von Steuern, Aufschlag und Grundpreis

## SOFTWARE-INSTALLATION

Über den Module Store das 'Strompreis'-Modul installieren.
Alternativ über das Module Control folgende URL hinzufügen

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Strompreis'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen
Konfigurationsseite:

| NAME | BESCHREIBUNG |
|---|---|
| Anbieter | Stromanbieter, von dem die Daten bezogen werden sollen |
| Markt | Auswahl des Landes |
| Grundpreis | Grundpreis des Stromes |
| Aufschlag | Prozentualer Aufschlag zum Marktpreis |
| Steuersatz | Steuersatz, welcher auf den Strom hinzukommt |

## STATUSVARIABLEN UND PROFILE

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Marktdaten | String | Daten zum anzeigen der Grafik |
| Aktueller Preis | Float | Aktueller Strompreis |

### PROFILE

| NAME | TYP |
| --- | --- |
| Cent | Float |

## VISUALISIERUNG

In der Kachelvisualisierung bietet das Module ein Balkendiagramm, welches den Verlauf des Strompreises darstellt.

# SPX_Update

Benötigt: IP-Symcon >= 7.1

```
boolean SPX_Update (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Aktualisiert die Daten.

## BEISPIEL

```
SPX_Update(12345)
```

# Verbrauch in Kategorie

Benötigt: IP-Symcon >= 6.3
Berechnet den Verbrauch in Prozent pro frei wählbarer Kategorie nach angegebener Start- und Enddatum. Die Berechnung basiert auf der täglichen Aggregation.

## FUNKTIONSUMFANG

Auswahl mehrere Variablen
Freitext für Kategorie

## SOFTWARE-INSTALLATION

Über den Module Store das 'Verbrauch in Kategorie'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Verbrauch in Kategorie'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen
Konfigurationsseite:

| NAME | BESCHREIBUNG |
|---|---|
| Intervall | Intervall, in welchem die Berechnung ausgeführt wird. 0 = Deaktiviert |
| Quellen | Liste der Variablen und Kategorien |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Kategorie | Float | Pro Kategorie wird eine Variable angelegt, in welchem der Prozentuale Verbrauch angezeigt wird. Der Kategoriename wird intern auf A-Z, a-z, 0-9 und _ (Unterstrich) reduziert. Das bedeutet, dass die Kategorie "Test" und "Test €" ebenfalls auf "Test" kategorisiert werden. |
| Startzeit | Integer | Datum von wann die Berechnung starten soll |
| Endzeit | Integer | Datum bis wann die Berechnung gehen soll |

# VIK_CalculationConsumption

Benötigt: IP-Symcon >= 6.3

```
boolean VIK_CalculateConsumption (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | Id der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Berechnet die Prozentangabe der Kategorie neu.

## BEISPIEL

```
VIK_CalculateConsumption(12345);
```

# Verbrauch in Zeitspanne

Benötigt: IP-Symcon >= 4.2

## FUNKTIONSUMFANG

Berechnet für eine Zeitspanne den Verbrauch anhand der Aggregation der ausgewählten Quellvariable.
Bei Änderung der Zeitspanne wird der Verbrauch neu ermittelt
Verschiedene Detailgerade für Start- und End-Datum
Einstellbares Intervall für die Aktualisierung der Berechnung

## SOFTWARE-INSTALLATION

Über den Module Store das Modul Verbrauch in Zeitspanne installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Verbrauch in Zeitspanne'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Quelle | Quellvariable, die für Verbrauch genutzt werden soll |
| Detailgrad | Legt fest wie genau der Start- und Endzeitpunkt festgelegt werden kann (Datum, Uhrzeit, Datum/Uhrzeit) |
| Intervall zum Aktualisieren nutzen | Wenn aktiv, wird ein Intervall zum Aktualisieren der Berechnung genutzt |
| Intervall | Das Intervall in Minuten, in dem die Berechnung aktualisiert wird |



Wird im Detailgrad Datum ausgewählt, wird das Startdatum mit der Uhrzeit 00:00 und das Enddatum mit der Uhrzeit 23:59:59 genutzt.

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
| --- | --- | --- |
| Start | Integer | Start-Datum/Start-Zeit für den Verbrauch (Sekunden werden nicht beachtet) |
| Ende | Integer | End-Datum/End-Zeit für den Verbrauch (Sekunden werden nicht beachtet) |
| Verbrauch | Float | Verbrauch zwischen Start- und End-Datum |

# VIZ_Calculate

Benötigt: IP-Symcon >= 4.2

```
boolean VIZ_Calculate (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Berechnet den Verbrauch zwischen Start- und End-Datum neu.

## BEISPIEL

```
VIZ_Calculate(12345);
```

# Verbrauchsverhalten

Benötigt: IP-Symcon >= 6.0
Das Modul errechnet den wahrscheinlichen Verbrauch für die Periode auf Basis einer Außentemperatur-Variable und einer Zähler-Variable mithilfe der Linearen Regression. Je mehr Werte für die Außentemperatur und den Zähler verfügbar sind, desto genauer kann der erwartete Verbrauch ermittelt werden.

## FUNKTIONSUMFANG

Errechnet aus zwei Variablen einen wahrscheinlichen Verbrauch

## SOFTWARE-INSTALLATION

Über den Module Store das 'Verbrauchsverhalten'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Verbrauchsverhalten'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Variable für Außentemperatur | Geloggte Variable für die Außentemperatur |
| Variable für den Zähler | Geloggte Variable für den Zähler |
| Periode | Zeitraum für den die Berechnung durchgeführt werden soll |
| Limit | Maximale Anzahl der Datensätze die für die Regression genutzt werden sollen. 0 = Keine Limitierung |
| Intervall | Zeitintervall des Timers in dem die Variable erneut berechnet werden soll |
| Berechnen | Button, um die Variablen neu zu berechnen |

Für die Zählervariable müssen mindestens 2 Datensätze für 3 aufeinderfolgenden Perioden vorhanden sein, damit eine Berechnung erfolgen kann. Für Periode Tag wird die stündliche Aggregation der Variablen genutzt. Für alle anderen Perioden wird die tägliche Aggregation der Variablen genutzt.

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

## STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Erwartung der aktuelle Periode | float | Zeigt den erwarteten Verbrauch auf Basis der Außentemperatur der laufenden Periode an |
| Erwartung der letzten Periode | float | Zeigt den erwarteten Verbrauch auf Basis der Außentemperatur der letzten Periode an |
| Hochrechnung der aktuelle Periode | float | Zeigt den hochgerechneten Verbrauch der laufenden Periode an |
| Hochrechnung der letzten Periode | float | Zeigt den hochgerechneten Verbrauch der letzten Periode an |
| Wert der aktuellen Periode | float | Zeigt den aktuellen Verbrauch der laufenden Periode an |
| Wert der letzten Periode | float | Zeigt den Verbrauch der letzten Periode an |
| Prozent der aktuellen Periode | float | Zeigt in wie weit sich der hochgerechnete Verbrauch vom erwarteten Wert prozentual unterscheiden |
| Prozent der letzten Periode | float | Zeigt in wie weit sich der hochgerechnete Verbrauch vom erwarteten Wert prozentual unterscheiden |
| Bestimmtheitsmaß der aktuellen Periode | float | Genauigkeit der Erwartungsberechnung der laufenden Periode an |
| Bestimmtheitsmaß der letzten Periode | float | Genauigkeit der Erwartungsberechnung der letzten Periode an |

## MATHEMATISCHE FORMELN

Die Erwartung erfolgt anhand der einfachen linearen Regression. Mathematisch Erklärt.
Hier wird die Summe des positiven Ergebnis aus der Rechnung m *Temperaturdurchschnit (je nach Periodendauer)* b errechnet. m und b werden durch die linearen Regression bestimmt.
Die Hochrechnung besteht wiederrum aus dem Durchschnittswert der Periode multipliziert mit der Periodenlänge.

# VBV_UpdateCalculation

Benötigt: IP-Symcon >= 5.0

```
boolean VBV_UpdateCalculation (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Die Funktion aktualisiert die prognostizierten Werte.

## BEISPIEL

```
VBV_UpdateCalculation(12345);
```

# VirtuelleMessstelle

Benötigt: IP-Symcon >= 5.5
Rechnet positive Variablenänderungen nach einstellbaren Regeln mit Veränderungen einer Hauptvariable zusammen und addiert dieses mit dem Wert einer Variable.
*Das Modul „Smart Energy Box“ wurde in Kooperation zwischen der Symcon GmbH, Stark Elektronik GmbH und der Hochschule Biberach (Institut IGE) im Rahmen des Landesprojektes EnMa HAW: Konzept für automationsgestütztes Energiemanagement an nicht-universitären Hochschulen in Baden-Württemberg entwickelt.*

## FUNKTIONSUMFANG

Auswahl einer geloggten Variable als primären Messstelle
Auswahl beliebig vieler Variablen als sekundären Messstellen
Einstellbar, ob die Veränderungen der sekundären Messtellen zu den Veränderungen der primären Messstellen addiert oder subtrahiert werden sollen

## SOFTWARE-INSTALLATION

Über den Module Store das 'Virtuelle Messstelle'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Virtuelle Messstelle'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

## KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Primäre Messstelle | Auswahl einer geloggten Variable |
| Sekundäre Messstelle | Mehrfachauswahl von Variablen |
| Operation | Addieren/Subtrahieren Option, ob die positive Veränderung der sekundären Messstelle zu der primären Messstelle hinzugefügt oder abgezogen werden soll |
| Variable | Auswahl einer geloggte Variable als sekundäre Messstelle |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

## STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
| --- | --- | --- |
| Ergebnis | Float | Errechnetes Ergebnis |

# VM_Update

Benötigt: IP-Symcon >= 5.5

```
boolean VM_Update (int $InstanzID, float $PrimaryDelta)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| PrimaryDelta | Positive Veränderung der Hauptmessstelle |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Die Steigung der sekundären Messstellen wird je nach ausgewählter Option addiert oder subtrahiert. Das Ergebnis wird mit dem übergebenen Wert zusammen auf den Wert der Variable 'Ergebnis' addiert.

## BEISPIEL

```
VM_Update(12345, 5.3);
```

# Virtueller Zähler

Benötigt: IP-Symcon >= 6.0
Das Modul stellt eine Eingabemaske für manuell abgelesene Zählerstände bereit. Es übernimmt den manuellen Eintrag, prüft diesen auf Plausibilität und übergibt den Wert an eine Zählervariable. Um auf Plausibilität zu prüfen kann ein Grenzwert eingestellt werden.

## FUNKTIONSUMFANG

Prüfung auf Plausibilität
Anzeige des zuletzt eingegebenen Zählerstandes
Eingabemaske für neuen Zählerstand

## SOFTWARE-INSTALATION

Über den Module Store das Modul Virtueller Zähler installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Virtueller Zähler'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Grenzwert | Zahl, um die ein neuer Zählerwert maximal steigen darf. Ist der Grenzwert 0 wird nicht auf diesen überprüft |
| Erfordere Bestätigung zum Übernehmen | Wenn Aktiviert, muss ein Script ausgeführt werden um die Plausibilitätsprüfung anzustoßen |
| Logging Aktivieren | Button, welcher das Logging der Zählervariable aktiviert. Er wird ausgeblendet, wenn das Logging aktiviert ist. |

## STATUSVARIABLEN UND PROFILE

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Aktueller Zählerstand | Float | Zeigt den zuletzt eingetragenen Zählerstand an |
| Neuer Zählerstand | String | Eingabevariable für neuen Zählerstand |
| Sicher neuen Zählerstand zu setzen: Eingabe des neuen Zählerstandes | Boolean | Erscheint, wenn der neue Zählerstand den Aktuellen um den Grenzwert überschreitet. Der Wert kann mit der Bestätigung trotzdem übernommen werden |

### PROFILE

| NAME | TYP |
|---|---|
| VZ.Confirm | Boolean |
| VZ.NewCounterReading | String |

### SKRIPTE

| BEZEICHNUNG | BESCHREIBUNG |
|---|---|
| Zählerstand setzen | Führt die Überprüfung auf Plausibilität aus und schiebt den Wert bei Erfolg in die Vaiable "Aktueller Zählerstand" |

## VISUALISIERUNG

Über die Visualisierung kann der neue Zählerstand eingegeben werden und das Skript ausgeführt werden, zudem wird der zuletzt übernommene Zählerstand angezeigt.

# VZ_WriteNewCounterValue

Benötigt: IP-Symcon >= 6.0

```
void VZ_WriteNewCounterValue (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Die Funktion liefert keinerlei Rückgabewert.

## BESCHREIBUNG

Überprüft ob der Wert in Variable "Neuer Zählerstand" negativ ist, der "Neue Zählerstand" kleiner dem "Aktuellen Zählerstand" oder der "Neue Zählerstand" den "Aktuellen Zählerstand" mit dem "Grenzwert" übersteigt. Wenn diese nicht zutreffen wird der neue Wert in "Aktuellen Zählerstand" gesetzt.

## BEISPIEL

```
VZ_WriteNewCounterValue(12345);
```

# Zählerstand (Tag)

Benötigt: IP-Symcon >= 4.2
Das Modul erlaubt die Auswahl eines Datums und zeigt dann den jeweiligen (Ersten/Letzten) Zählerwert dieses Tages an.

## FUNKTIONSUMFANG

Bei Änderung des Datums wird je nach Wert-Auswahl der Erste/Letzte gelesene Zählerwert der geloggten Variable ausgegeben.

## SOFTWARE-INSTALLATION

Über den Module Store das Modul Zählerstand (Tag) installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Zählerstand (Tag)'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Quelle | Quellvariable, welche als Datenquelle genutzt werden soll. |
| Wert | Erster/Letzter Wert des Tages. |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Zählerstand | Integer/Float | Wert für das ausgewählte Datum. |

## VISUALISIERUNG

Über die Visualisierung wird die Variable angezeigt. Es kann ein Datum gewählt werden für das der Zählerstand (jeweils der Erste/Letzte des Tages) angezeigt wird.

# ZST_Calculate

Benötigt: IP-Symcon >= 4.2

```
boolean ZST_Calculate (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Ermittelt den Wert für das gewählte Datum neu.

## BEISPIEL

```
ZST_Calculate(12345);
```

# Zählerüberlauf

Benötigt: IP-Symcon >= 4.2
Das Modul stellt überlaufende Zähler als kontinuierliche Zähler dar.

## FUNKTIONSUMFANG

Berechnet den Gesamtwert einer Variable und zählt diese hoch, obwohl das Gerät einen Überlauf hat.

## SOFTWARE-INSTALLATION

Über den Module Store das Modul Zähler-Überlauf installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Zählerüberlauf'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Quelle | Quellvariable, welche für die Berechnung genutzt werden soll. |
| Maximalwert | Ab welchem Wert ein Überlauf stattfindet. Der maximale Wert, welcher das Gerät zählt. |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Zähler | Float | Fortlaufend hochzählender Wert. |

# Visualisierungen

Im Abschnitt Visualisierung werden Funktionen und Einrichtung von den gängigen Visualisierungen zusammengefasst.

| | |
|---|---|
| Kachel Visualisierung | Funktionen zum Ansteuern der Kachel Visualisierung auf den Client-Geräten |
| VISU_PostNotification | sendet eine Push-Benachrichtigung auf die Kachel Visualisierung |
| VISU_PostNotificationEx | sendet eine Push-Benachrichtigung auf die Kachel Visualisierung |
| WebFront Visualisierung | Funktionen zum Ansteuern des WebFronts auf den Client-Rechnern |
| WFC_AudioNotification | sendet eine Audio-Meldung auf das WebFront |
| WFC_OpenCategory | Öffnet im WebFront eine Kategorie als Popup |
| WFC_PushNotification | sendet eine Push-Nachricht an die mobilen Apps |
| WFC_Reload | lädt das WebFront auf dem Client-Rechner neu |
| WFC_SendNotification | sendet eine kleine Meldung auf das WebFront |
| WFC_SendPopup | sendet eine große Meldung an das WebFront |
| WFC_SwitchPage | wechselt den Reiter im WebFront |

# Kachel Visualisierung

Benötigt: IP-Symcon >= 7.0

Eine generelle Anleitung kann im Bereich Kachel Visualisierung gefunden werden.

| FUNKTION | BESCHREIBUNG |
| --- | --- |
| Benachrichtigung senden | Push-Benachrichtigungen, können an alle verbundenen Geräte gesendet und mit einem Benachrichtigungstyp versehen werden. |
| Erweiterte Benachrichtigung senden | Push-Benachrichtigungen können an alle verbundenen Geräte gesendet und mit einem eigenen Icon und Ton versehen werden. |

# VISU_PostNotification

Benötigt: IP-Symcon >= 7.0

```
variant VISU_PostNotification (int $InstanzID, string $Titel, string
$Text, string $Type, int $TargetID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Kachel Visualisierung |
| Titel | Titel der Meldung (maximal 32 Zeichen). Kann auch leer sein. |
| Text | Text der Meldung (maximal 256 Zeichen). |
| Type | Der Typ der Meldung beeinflusst den beim Empfangen abgespielten Ton und das Icon. |
| TargetID | Die ID des Objektes welches beim antippen der Meldung geöffnet wird |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis die ID der Benachrichtigung, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet eine Push-Nachricht mit **Titel**, **Text** an alle Geräte, die der Visualisierung mit der ID **InstanzID** zugeordnet sind.



Das TargetID Objekt muss innerhalb der angegeben Kachel Visualisierung verfügbar sein!



Sollen Nachrichten an weitere Gruppen (Sonderfall: auch einzelne Geräte) gesendet werden, so muss pro Gruppe eine weitere Kachel Visualisierung erstellt werden.

Bitte beachten Sie, dass Push-Benachrichtigungen verschlüsselt über den IP-Symcon Server an Apple/Google weitergeleitet werden, und dies eine gewisse Verzögerungszeit impliziert. Bei zu vielen Nachrichten hintereinander verzögern die Dienste von Apple/ Google die Zustellungen automatisch, worauf IP-Symcon keinerlei Einfluss nehmen kann.

Die Nutzung dieser Funktion erfordert, wie schon in der Beschreibung der Visualisierung beschrieben, eine gültige Subskription. Diese können Sie jederzeit über unseren Shop verlängern. Zusätzlich muss das Gerät in IP-Symcon registriert werden. Nähere Informationen dazu finden Sie unter dem Punkt Notification Control

## BEISPIEL

```
VISU_PostNotification(12345, 'Wetter', 'Es regnet bald', 'Info',
54321);
```

# VISU_PostNotificationEx

Benötigt: IP-Symcon >= 7.0

```
variant VISU_PostNotificationEx (int $InstanzID, string $Titel,
string $Text, string $Icon, string $Sound, int $TargetID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Kachel Visualisierung |
| Titel | Titel der Meldung (maximal 32 Zeichen). Kann auch leer sein. |
| Text | Text der Meldung (maximal 256 Zeichen). |
| Icon | Wenn nicht leer, dann wird das Icon angezeigt. Siehe Icons |
| Sound | Beim empfangen der Meldung abgespielter Ton. (alarm bell, boom, buzzer, connected, dark, digital, drums, duck, full, happy, horn, inception, kazoo, roll, siren, space, trickling, turn) |
| TargetID | Die ID des Objektes welches beim antippen der Meldung geöffnet wird. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis die ID der Benachrichtigung, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet eine Push-Nachricht mit **Titel**, **Text**, **Icon**, **Sound** an alle Geräte, die der Visualisierung mit der ID **InstanzID** zugeordnet sind.

Das TargetID Objekt muss innerhalb der angegeben Kachel Visualisierung verfügbar sein!

Sollen Nachrichten an weitere Gruppen (Sonderfall: auch einzelne Geräte) gesendet werden, so muss pro Gruppe eine weitere Kachel Visualisierung erstellt werden.

Bitte beachten Sie, dass Push-Benachrichtigungen verschlüsselt über den IP-Symcon Server an Apple/Google weitergeleitet werden, und dies eine gewisse Verzögerungszeit impliziert. Bei zu vielen Nachrichten hintereinander verzögern die Dienste von Apple/ Google die Zustellungen automatisch, worauf IP-Symcon keinerlei Einfluss nehmen kann.

Die Nutzung dieser Funktion erfordert, wie schon in der Beschreibung der Visualisierung beschrieben, eine gültige Subskription. Diese können Sie jederzeit über unseren Shop verlängern. Zusätzlich muss das Gerät in IP-Symcon registriert werden. Nähere Informationen dazu finden Sie unter dem Punkt Notification Control

## BEISPIEL

```
VISU_PostNotificationEx(12345, 'Haustür', 'Bewegung erkannt',
'Alert', 'alarm', 54321);
```

# WebFront Visualisierung

Eine generelle Anleitung kann im BereichWebfront Visualisierung gefunden werden.

Ab IP-Symcon Version 2.2 ist es möglich, das WebFront von IP-Symcon aus zu steuern und Meldungen an das WebFront zu senden. Dem Benutzer stehen dabei folgende Funktionen zur Verfügung, um das WebFront zu beeinflussen.

| FUNKTION | BESCHREIBUNG |
|---|---|
| Audio-Notification senden | Audio-Benachrichtigungen können an das WebFront gesendet werden, welche dann vom Web-Browser akustisch wiedergegeben werden. |
| Notification senden | Kleine Benachrichtigungen können an das WebFront gesendet werden, die mit einem Icon, Kopfzeile und Text an der rechten Seite im WebFront angezeigt werden. |
| Popup öffnen | Um größere Meldungen anzeigen zu können, kann ein Popup geöffnet werden, welches fast den vollständigen Bildschirm ausfüllt und wichtige Meldungen darstellt, die nicht in eine kleine Notification passen. |
| Push-Notification senden | Push-Benachrichtigungen können an die Mobilen Apps (iOS/Android) gesendet werden, die direkt auf dem Mobilgerät angezeigt und akustisch signalisiert werden. |
| WebFront neu laden | Lädt das WebFront komplett neu (gleicht einem Neuladen der Seite im Browser (F5)). |
| WebFront Reiter auswählen | Wählt den Reiter mit der angegebenen ID aus. Diese Funktion ist nützlich, um z.B. bei einer Bewegung an der Haustür zum Tür-Kamera-Reiter zu springen. |

# WFC_AudioNotification

Benötigt: IP-Symcon >= 3.0

```
boolean WFC_AudioNotification (int $InstanzID, string $Titel, int $MediaID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der WebFront Visualisierung |
| Titel | Titel der Audio-Meldung |
| MediaID | ID des Media-Objekts, welches vom Typ "Ton" ist |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet eine Audio-Nachricht mit **Titel** an das WebFront, das dem Konfigurator mit der ID **InstanzID** zugeordnet ist. Die Audio-Nachricht erscheint als kleiner Kasten in der oberen rechten Ecke des WebFronts und wird für die Abspieldauer angezeigt.

Bitte beachten, dass nicht alle Browser alle Formate unterstützen: http://www.w3schools.com/tags/tag_audio.asp

Um "Media Dateien" im WebFront abspielen zu können, müssen diese im "sichtbaren" Bereich des WebFronts als Links zur Verfügung gestellt werden. Diese können auch unsichtbar hinterlegt werden. Geschieht dies nicht, kann das WebFront nicht auf die Datei zugreifen und es wird ein Fehler geworfen, dass die Instanz nicht gefunden werden kann.

Unter iOS hat diese Funktion keinerlei Auswirkung, da Apple das Abspielen von Sounddateien im WebBrowser unterbindet!



Unter Android Browser und Android Chrome hat diese Funktion keierlei Auswirkung, da Google das Abspielen von Sounddateien im WebBrowser unterbindet!

Unter Android Chrome kann das automatische Abspielen von Sounddateien aktiviert werden. Dazu muss unter about:flags "Bewegungsanforderung für Medienwiedergabe deaktivieren" aktiviert und der Browser neugestartet werden.

## BEISPIEL

```
WFC_AudioNotification(12345, 'Gong!', 55541); //55541 ist die ID des
Media Objekts in IP-Symcon
```

# WFC_OpenCategory

Benötigt: IP-Symcon >= 4.1

```
boolean WFC_OpenCategory (int $InstanzID, int $KategorieID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der WebFront Visualisierung |
| KategorieID | ID der Kategorie |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Öffnet die Kategorie mit der ID **KategorieID** als Popup. Dabei werden alle Unterobjekte der Kategorie in dem Popup angezeigt.



Die zu öffnende Kategorie muss aus Sicherheitsgründen innerhalb des WebFronts verfügbar sein. Ist dies nicht der Fall, kann die Kategorie nicht angezeigt werden.

## BEISPIEL

```
WFC_OpenCategory(12345, 45678);
```

# WFC_PushNotification

Benötigt: IP-Symcon >= 3.4

```
boolean WFC_PushNotification (int $InstanzID, string $Titel, string
$Text, string $Sound, int $TargetID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der WebFront Visualisierung |
| Titel | Titel der Meldung (maximal 32 Zeichen). Kann auch leer sein. |
| Text | Text der Meldung (maximal 256 Zeichen). |
| Sound | Wird in den Apps ab Version 3.0.6 ausgewertet |
| TargetID | Es kann beim Öffnen der Push Nachricht direkt zu einem Objekt gesprungen werden. (ab Version 5.0) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet eine Push-Nachricht mit **Titel**, **Text** an alle mobilen Geräte, die dem Konfigurator mit der ID **InstanzID** zugeordnet sind. Alternativ kann auch einer der folgenden Sounds abgespielt werden:

alarm
bell
boom
buzzer
connected
dark
digital
drums
duck
full
happy
horn
inception
kazoo
roll
siren
space
trickling

turn

Das TargetID Objekt muss innerhalb vom angegeben WebFront Konfigurator verfügbar sein, ansonsten kommt eine Fehlermeldung! (TargetID is not inside the mobile context)

Sollen Nachrichten an weitere Gruppen (Sonderfall: auch einzelne mobile Geräte) gesendet werden, so muss pro Gruppe ein weiterer WebFront Konfigurator erstellt werden.

Bitte beachten Sie, dass Push-Benachrichtigungen verschlüsselt über den IP-Symcon Server an Apple/Google weitergeleitet werden, und dies eine gewisse Verzögerungszeit impliziert. Bei zu vielen Nachrichten hintereinander verzögern die Dienste von Apple/ Google die Zustellungen automatisch, worauf IP-Symcon keinerlei Einfluss nehmen kann.

Die Nutzung dieser Funktion erfordert, wie schon in der Beschreibung der Mobile-Apps beschrieben, eine gültige Subskription. Diese können Sie jederzeit über unseren Shop verlängern. Zusätzlich muss das Gerät in IP-Symcon registriert werden. Nähere Informationen dazu finden Sie unter dem Punkt Notification Control

## BEISPIEL

```
WFC_PushNotification(12345, 'Warnung', 'Es regnet bald!', '', 0);
```

# WFC_Reload

Benötigt: IP-Symcon >= 2.2

```
boolean WFC_Reload (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der WebFront Visualisierung |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Lädt das WebFront, das der Visualisierung mit der ID **InstanzID** zugeordnet ist neu.

## BEISPIEL

```
WFC_Reload(12345);
```

# WFC_SendNotification

Benötigt: IP-Symcon >= 2.2

```
boolean WFC_SendNotification (int $InstanzID, string $Titel, string
$Text, string $Icon, int $Timeout)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der WebFront Visualisierung |
| Titel | Titel der Meldung |
| Text | Text der Meldung. Kann auch leer sein. |
| Icon | Wenn nicht leer, dann wird das Icon angezeigt. Siehe Icons |
| Timeout | Anzuzeigende Zeit in Sekunden. 0 = Nur ausblenden, wenn der Nutzer auf die Meldung klickt. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet eine kleine Nachricht mit **Titel**, **Text** und **Icon** an das WebFront, das der Visualisierung mit der ID **InstanzID** zugeordnet ist. Die Nachricht erscheint als kleiner Kasten in der oberen rechten Ecke des WebFronts und wird für **Timeout** Sekunden angezeigt.

## BEISPIEL

```
WFC_SendNotification(12345, 'Test', 'Dies ist ein toller Test',
'Speaker', 4);
```

# WFC_SendPopup

Benötigt: IP-Symcon >= 2.2

```
boolean WFC_SendPopup (int $InstanzID, string $Titel, string $Text)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der WebFront Visualisierung |
| Titel | Angezeigter Titel |
| Text | Angezeigter Text (HTML wird gefiltert!) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet eine große Meldung an das WebFront, das der Visualisierung mit der ID **InstanzID** zugeordnet ist. Es werden der Inhalt **Text** und der **Titel** angezeigt.



HTML Tags wie z.B. <br> (Zeilenumbruch) werden ohne Probleme verarbeitet

## BEISPIEL

```
WFC_SendPopup(12345, "Test", "Eine nette <br> Meldung");
// Ausgabe:
// "Eine nette
// Meldung"
```

# WFC_SwitchPage

Benötigt: IP-Symcon >= 2.2

```
boolean WFC_SwitchPage (int $InstanzID, string $PageName)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der WebFront Visualisierung |
| PageName | Name des Reiters (z.B. item1234) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Wechselt im WebFront, das der Visualisierung mit der ID **InstanzID** zugeordnet ist, auf den Reiter mit der ID **PageName**.

Die Reiternamen "itemxxx" werden im WebFront Editor bei möglichen Items im Konfigurationsfenster angezeigt. Zum Beispiel als "Konfiguration (item1234)".

Der Befehl WFC_SwitchPage unterbindet bzw. resettet den Idle-Timer vom Idle Widget

## BEISPIEL

```
WFC_SwitchPage(12345, "item1234");
```

# Sprachassistenten

Sprachassistentssysteme lassen Symcon über Sprachbefehle steuern. Die Möglichkeiten und Einrichtung sind hier beschrieben.

| | |
|---|---|
| Amazon Alexa | Modul zur Verknüpfung mit Amazon Alexa |
| Expertenoptionen | Expertenoptionen von Amazon Alexa |
| Fernsehgerät | Fernsehgerät in Amazon Alexa |
| Generischer Schalter | Generische Schalter in Amazon Alexa |
| Generischer Schieberegler | Generische Slider in Amazon Alexa |
| Lautsprecher | Lautsprecher in Amazon Alexa |
| Lautsprecher mit Stummschaltung | Lautsprecher mit Stummschaltung in Amazon Alexa |
| Licht (Dimmer) | Licht in Amazon Alexa, welches gedimmt werden kann |
| Licht (Experte) | Expertenlicht in Amazon Alexa |
| Licht (Farbe) | Farbiges Licht in Amazon Alexa |
| Licht (Schalter) | Licht in Amazon Alexa, welches an- und ausgeschaltet werden kann |
| Medienabspieler | Medienabspieler in Amazon Alexa |
| Rollladen | Rollläden in Amazon Alexa |
| Schloss | Schlösser in Amazon Alexa |
| Szenen | Szenen in Amazon Alexa |
| Szenen (deaktivierbar) | Szenen in Amazon Alexa |
| Temperatursensor | Temperatursensoren in Amazon Alexa |
| Thermostat | Thermostate in Amazon Alexa |
| Google Assistant | Modul zur Verknüpfung mit Google Assistant |
| Expertenoptionen | Expertenoptionen von Google Assistant |
| Generischer Schalter | Generische Schalter in Google Assistant |
| Licht (Dimmer) | Licht in Google Assistant, welches gedimmt werden kann |
| Licht (Experte) | Expertenlicht in Google Assistant |
| Licht (Farbe) | Farbiges Licht in Google Assistant |
| Licht (Schalter) | Licht in Google Assistant, welches an- und ausgeschaltet werden kann |
| Rollladen | Rollläden in Google Assistant |
| Szenen | Szenen in Google Assistant |
| Szenen (deaktivierbar) | Szenen in Google Assistant |
| Thermostat | Thermostate in Google Assistant |

# Amazon Alexa

Benötigt: IP-Symcon >= 5.0

## BESCHREIBUNG

Das Amazon Alexa-Modul ermöglicht die Steuerung von Geräten durch Amazon Alexa. Dies beinhaltet insbesondere die Sprachsteuerung über Echo Dots.

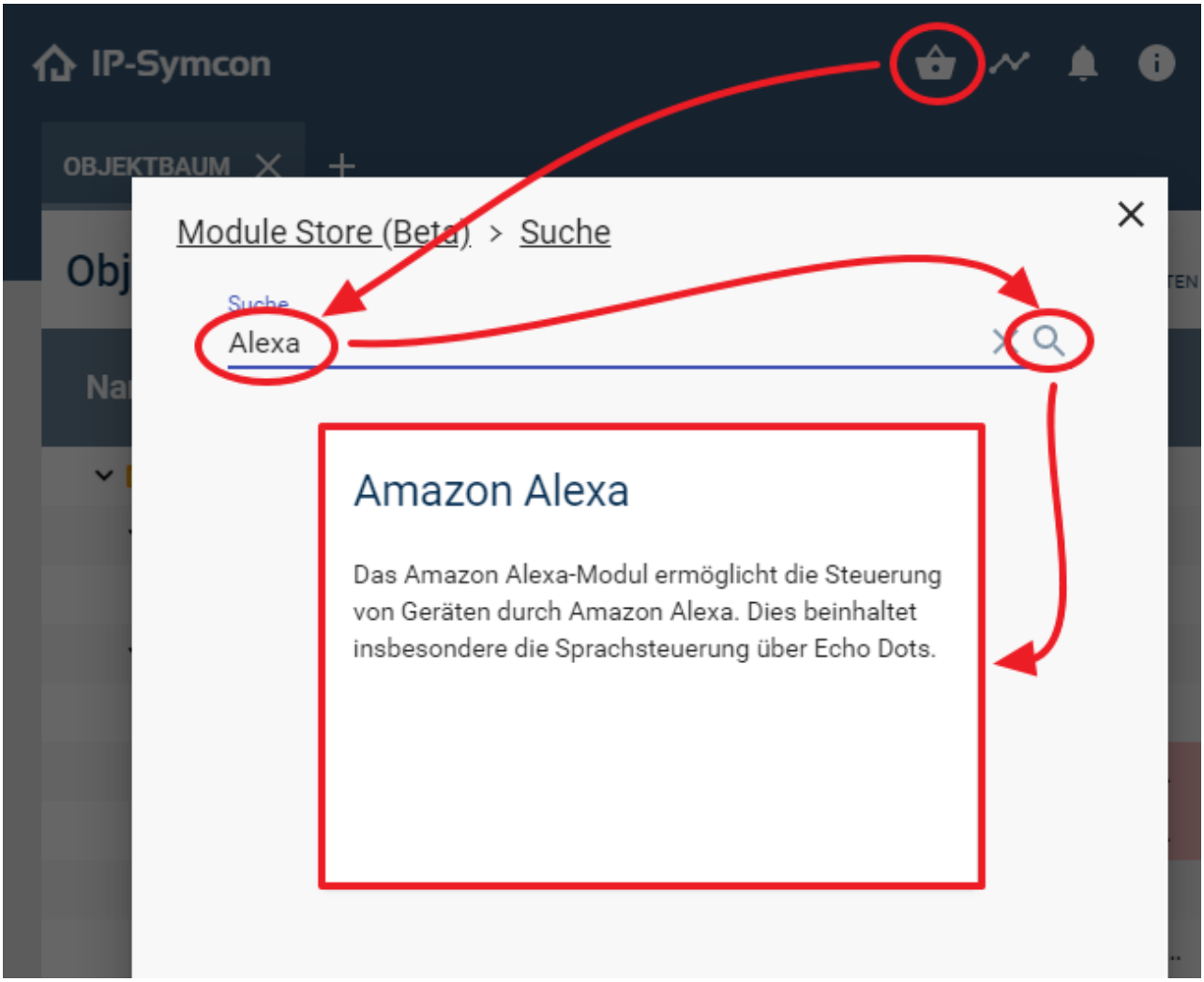
## EINBINDUNG IN IP-SYMCON

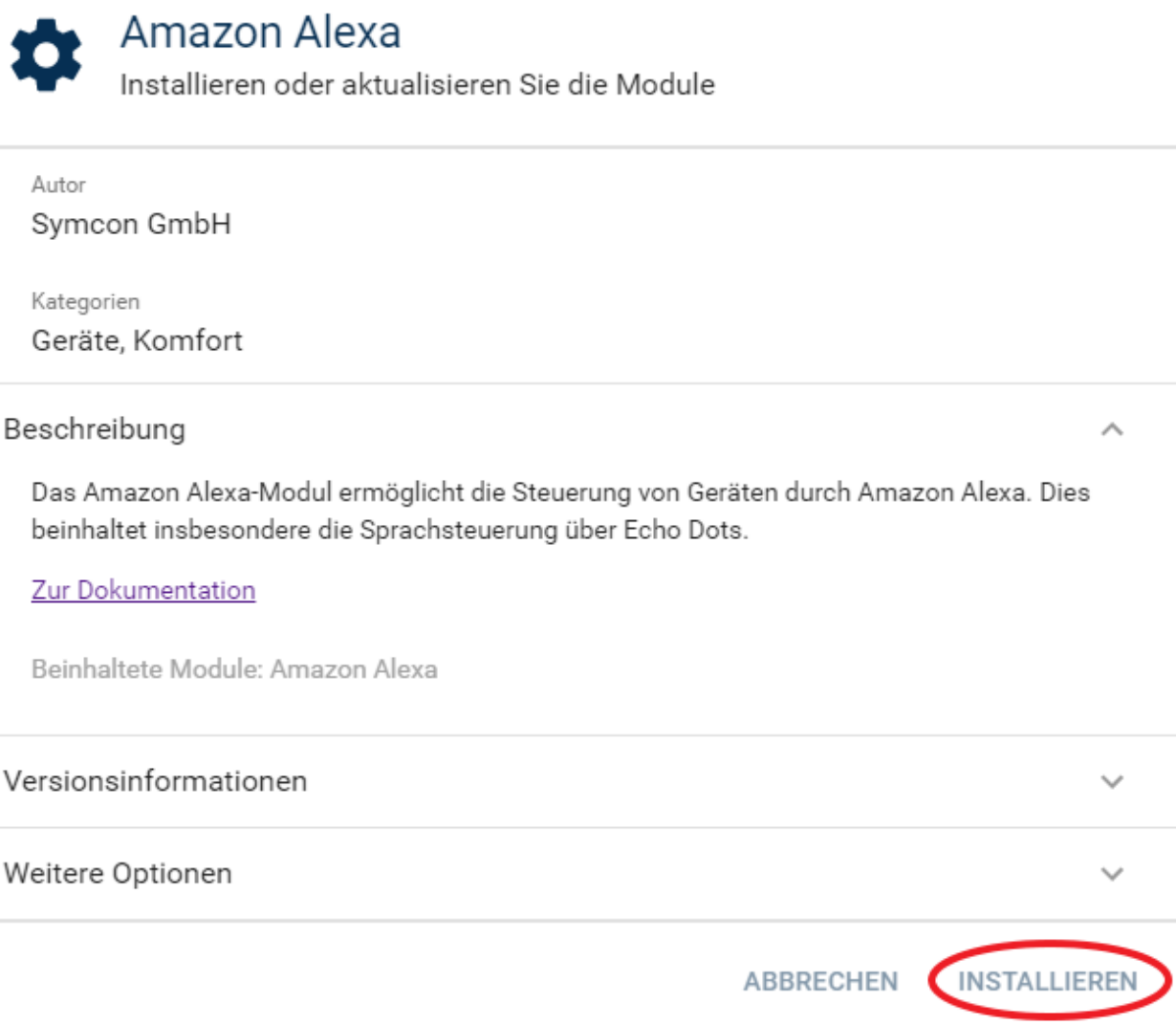


Die Verwendung des Amazon Alexa-Moduls erfordert eine aktivierte Verbindung zum Connect-Dienst, welche wiederum eine laufende Subskription erfordert.

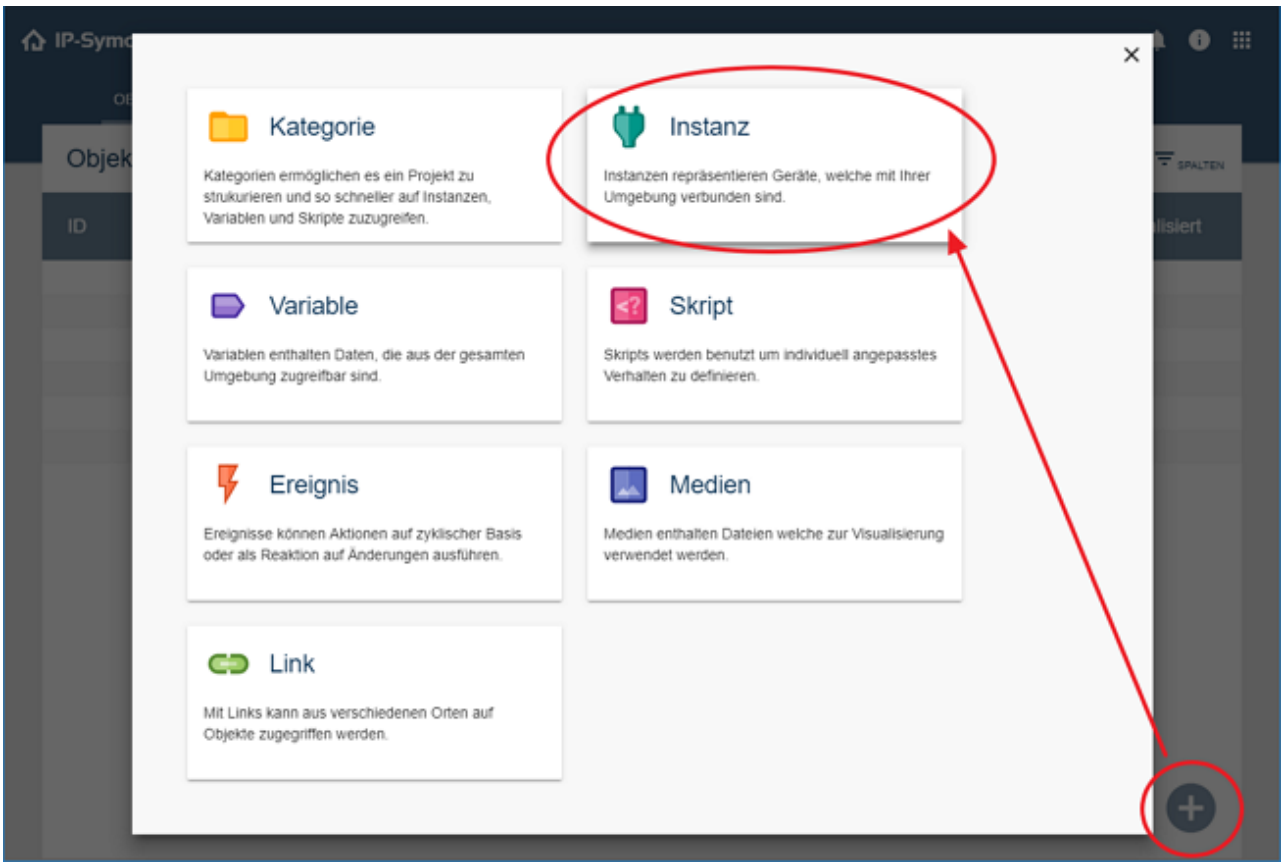
### AMAZON ALEXA-INSTANZ ERSTELLEN

Damit das Amazon Alexa-Modul verwendet werden kann, muss dieses zuerst über den Module Store installiert werden. Hierfür muss der Module Store geöffnet werden. Dieser befindet sich im oberen rechten Bereich. Über das Suchfeld kann via "Alexa" das Modul Amazon Alexa gefunden werden. Beim öffnen des gefundenen Modules, kann im folgenden Dialog die Installation des Moduls via "Installieren"-Knopf angestoßen werden.

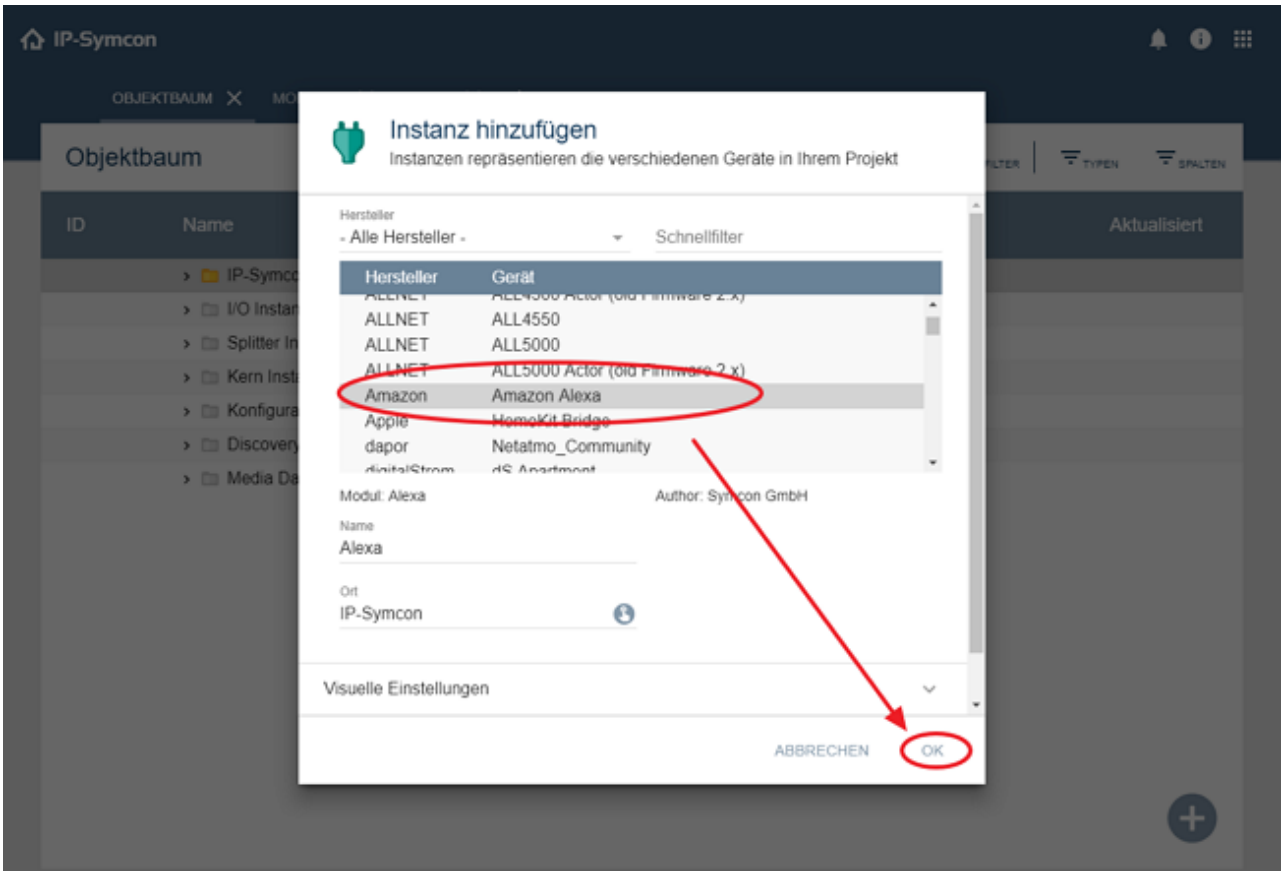

IP-Symcon
OBJEKTBAUM
Module Store (Beta) > Suche
Suche
Alexa
Amazon Alexa
Das Amazon Alexa-Modul ermöglicht die Steuerung von Geräten durch Amazon Alexa. Dies beinhaltet insbesondere die Sprachsteuerung über Echo Dots.




Im Anschluss öffnet sich der Hinzufügendialog zum Erstellen einer Amazon Alexa Instanz. Soll die Instanz manuell hinzugefügt werden muss eine Instanz vom Amazon Alexa-Modul erstellt werden. Da muss zuerst der Objektbaum geöffnet werden. In diesem den Hinzufügen-Button "+" unten rechts betätigen und Instanz auswählen.



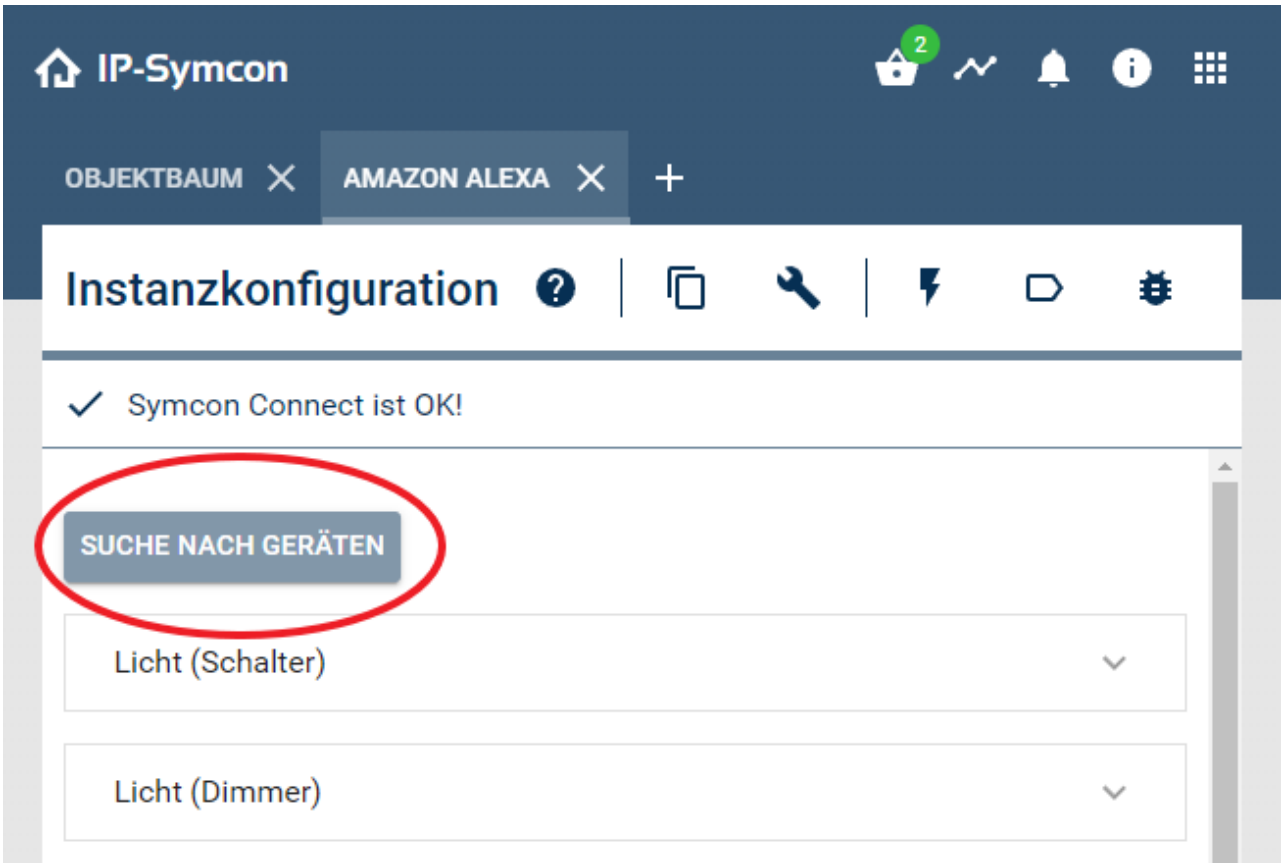

In der Modulliste das Gerät "Alexa" vom Hersteller "Amazon" auswählen. Der Ort sollte nicht verändert werden, der Name kann beliebig gewählt werden. Abschließend mit "OK" bestätigen.



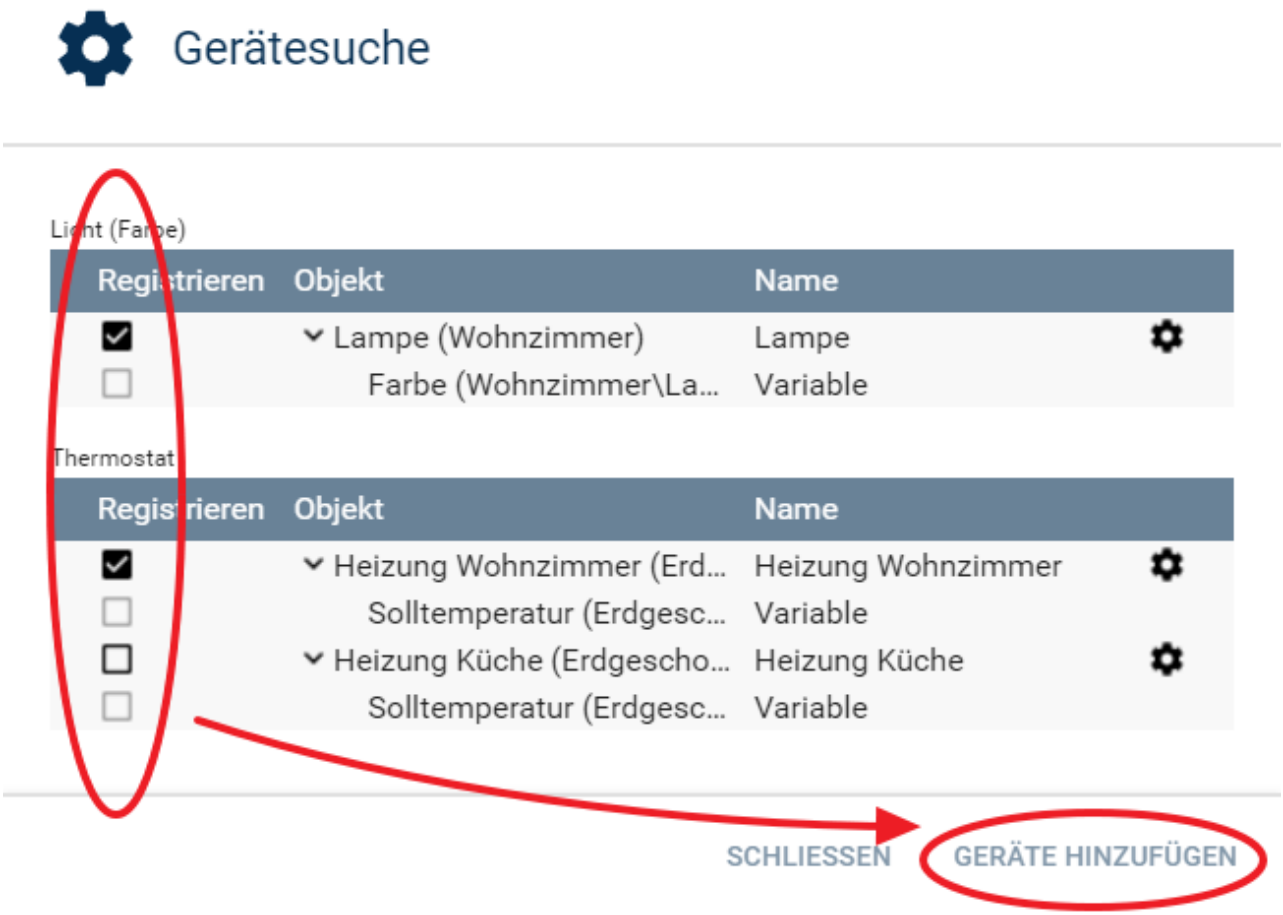

Nach der Erstellung der Amazon Alexa-Instanz öffnet sich diese automatisch und kann eingerichtet werden.

## AMAZON ALEXA-INSTANZ EINRICHTEN

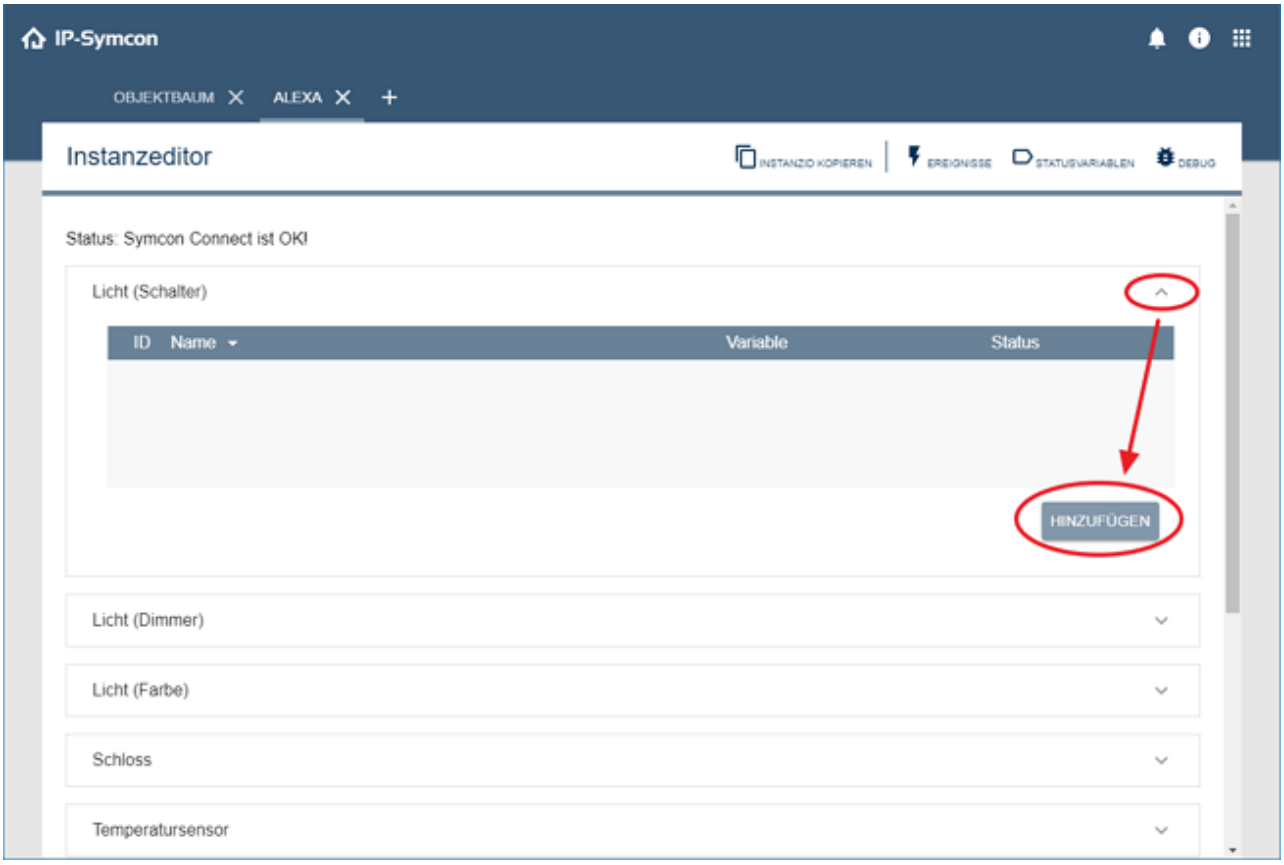
Das Modul bietet verschiedene Gerätetypen an, welche verwendet werden können. Inbesondere um die initiale Einrichtung zu beschleunigen, sollte die Gerätesuche verwendet werden. Später können weitere Geräte unkompliziert manuell hinzugefügt werden.
Um die Gerätesuche zu starten muss auf den Button "Suche nach Geräten" am oberen Ende der Instanzkonfiguration geklickt werden




Daraufhin öffnet sich ein Dialog, welcher nach einer Ladezeit gefundene Geräte darstellt, welche bisher noch nicht eingebunden wurden. Die Instanzen werden nach Gerätetypen aufgeteilt. Unter jeder Instanz werden die Statusvariablen dargestellt, welche bei einer Einbindung vom Modul verwendet würden. Soll eine Instanz eingebunden werden, so muss die Checkbox in dieser Zeile ausgewählt werden. Standardmäßig wird der Name der Instanz verwendet. Soll dieser angepasst werden, kann dies über das "Bearbeiten"-Symbol rechts im Listeneintrag getan werden. Nachdem alle gewünschten Geräte ausgewählt und ggfs. passend benannt wurden, muss die Auswahl mit "Geräte hinzufügen" bestätigt werden. Dann werden die gewählten Geräte in die dazugehörigen Listen in der Instanzkonfiguration hinzugefügt.

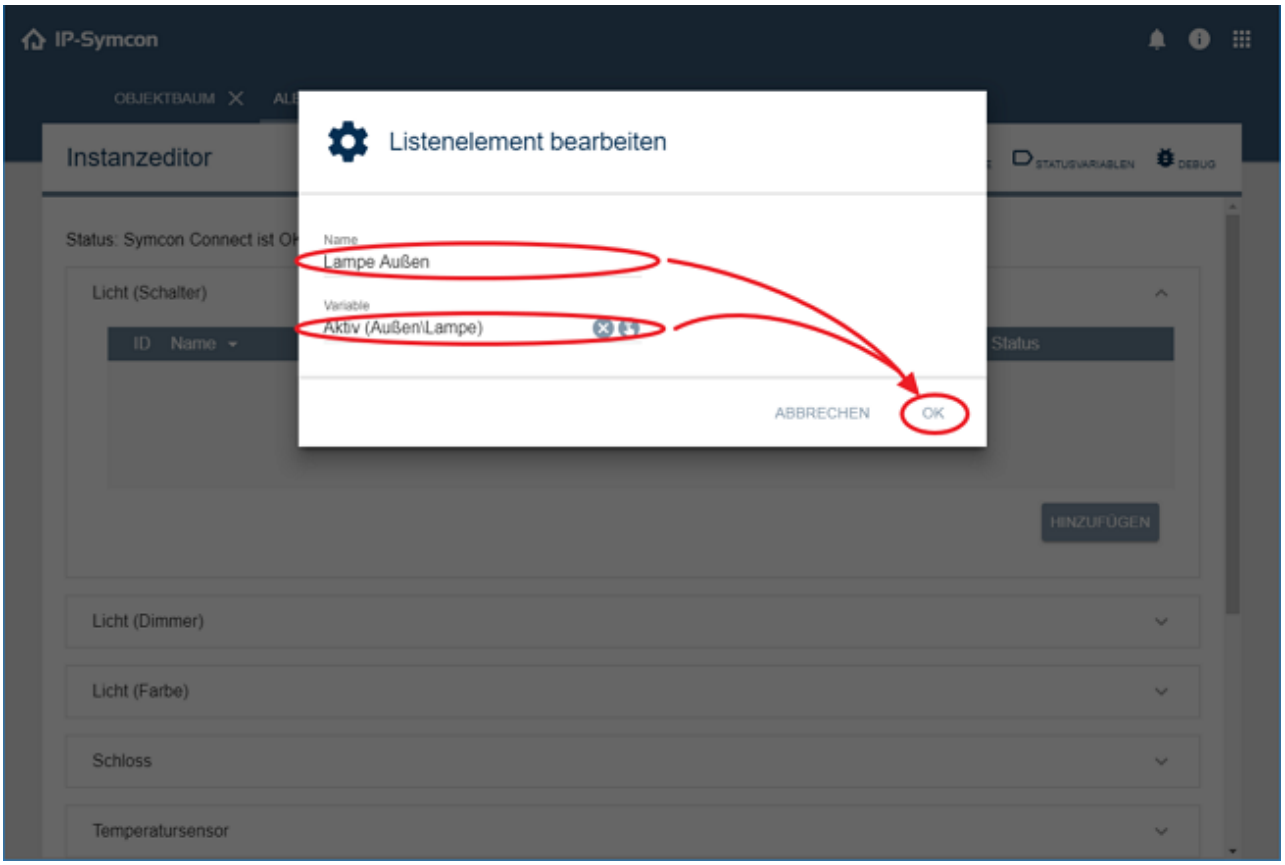



Um ein Gerät manuell einzurichten, wird das dazugehörige Panel ausgeklappt und in der entsprechenden Liste durch einen Klick auf "Hinzufügen" ein neuer Eintrag erstellt.




Jeder Eintrag erfordert einen Namen unter welchem das Gerät bei Amazon Alexa bekannt

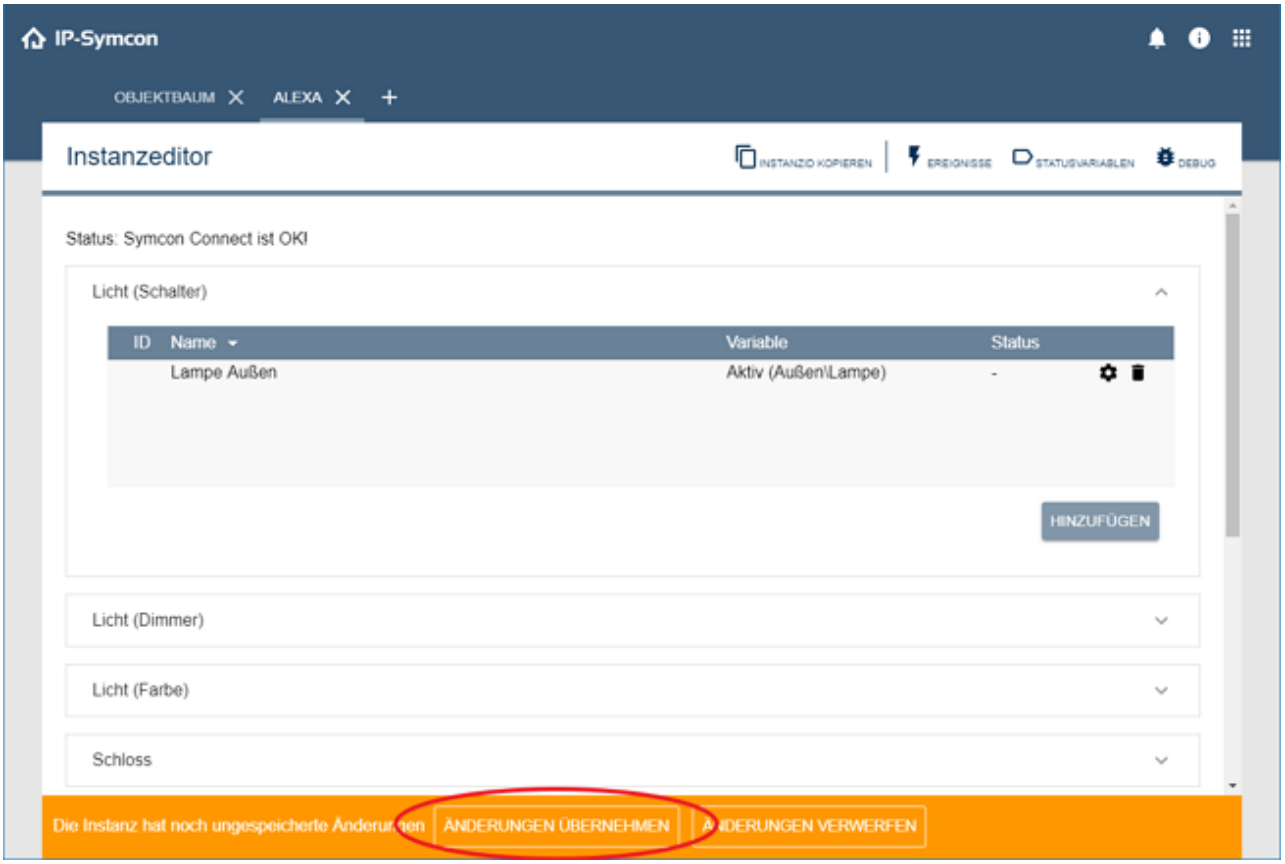
sein wird. Weitere Parameter hängen vom jeweiligen Gerätetyp ab und können den dazugehörigen Seiten der Dokumentation entnommen werden. Nachdem alle Parameter gesetzt sind, wird das Gerät mit "OK" bestätigt.



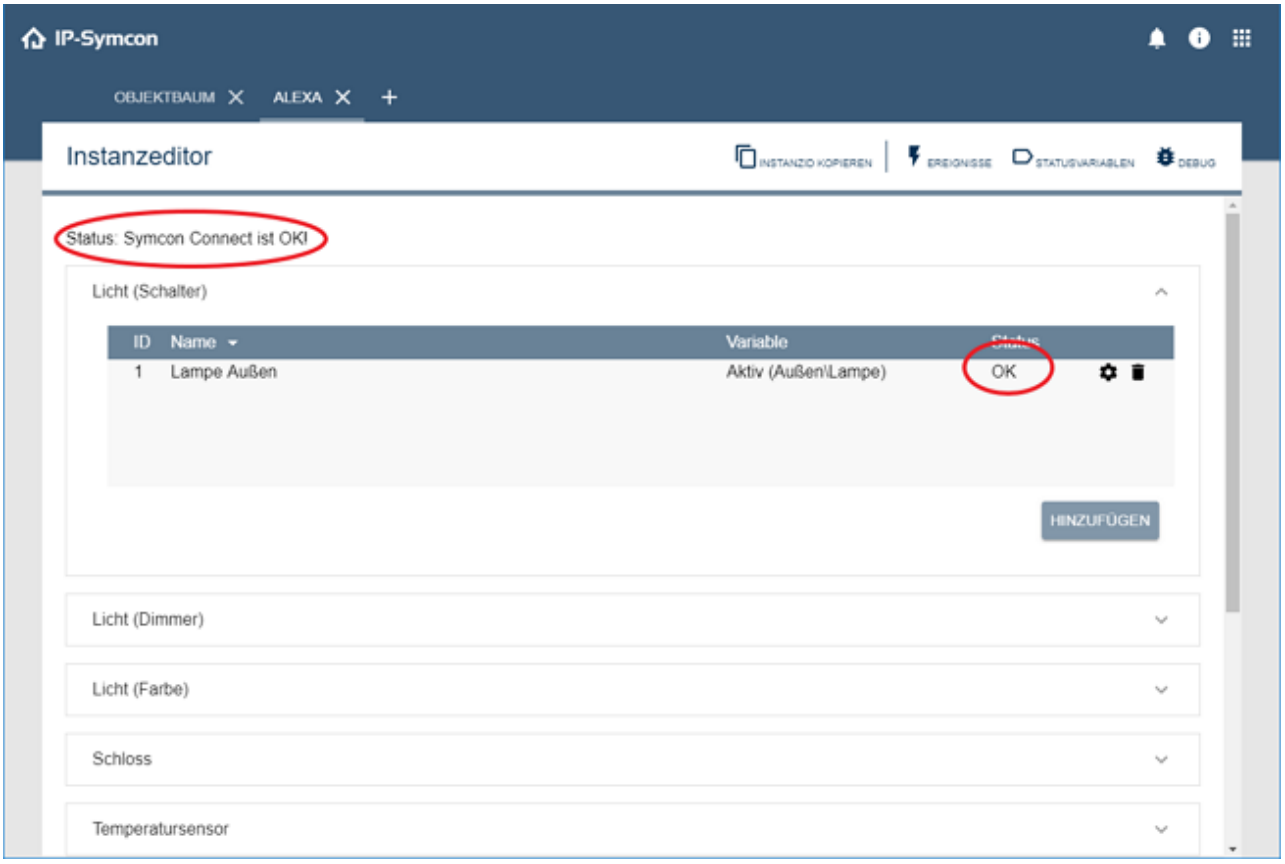

## MÖGLICHE GERÄTETYPEN

- Licht (Schalter)
- Licht (Dimmer)
- Licht (Farbe)
- Licht (Experte)
- Schloss
- Temperatursensor
- Thermostat
- Lautsprecher
- Lautsprecher mit Stummschaltung
- Fernsehgerät
- Medienabspieler
- Rollladen
- Generischer Schalter
- Generischer Slider
- Szenen
- Szenen (deaktivierbar)

Nachdem alle Geräte eingerichtet sind, werden diese mit einem Klick auf "Änderungen übernehmen" bestätigt.

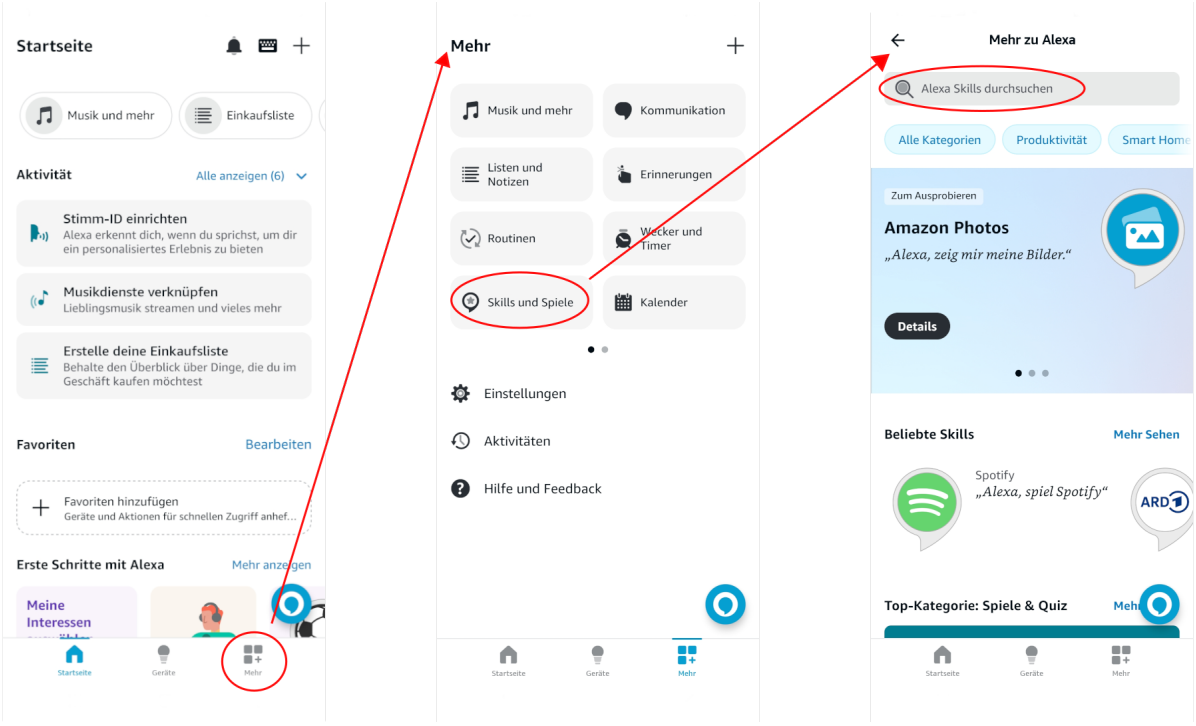



Nach dem Übernehmen sollte sichergestellt werden, dass bei allen Geräten "OK" in der Spalte "Status" steht. Ist dies nicht der Fall, so enthält die Spalte eine Fehlermeldung, welche über die Komplikation informiert.
Ebenfalls sollte geprüft werden, dass oben im Instanzeditor "Status: Symcon Connect ist OK" steht. Sollte dies nicht der Fall sein, muss der Connect-Dienst aktiviert werden.

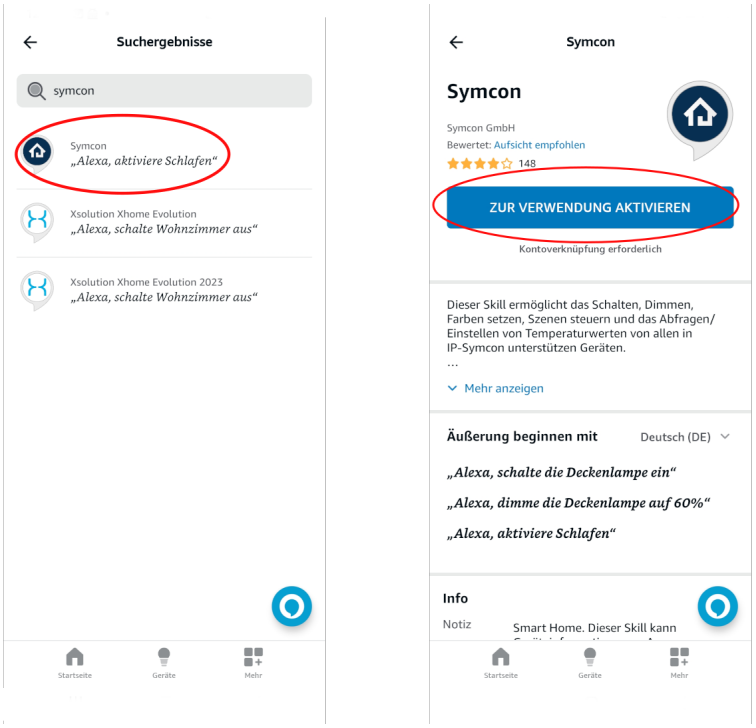


## IP-SYMCON MIT AMAZON ALEXA VERKNÜPFEN

Abschließend muss IP-Symcon noch mit Amazon Alexa verknüpft werden, damit die Amazon Alexa-Instanz die Anfragen von Amazon Alexa empfangen kann. Hierfür muss zuerst der Symcon-Skill für Amazon Alexa installiert werden.
Die Installation erfolgt über die Amazon Alexa App.
Von der Startseite der App wird auf "Mehr" gewechselt und dort der Punkt "Skills und Spiele" angeklickt.

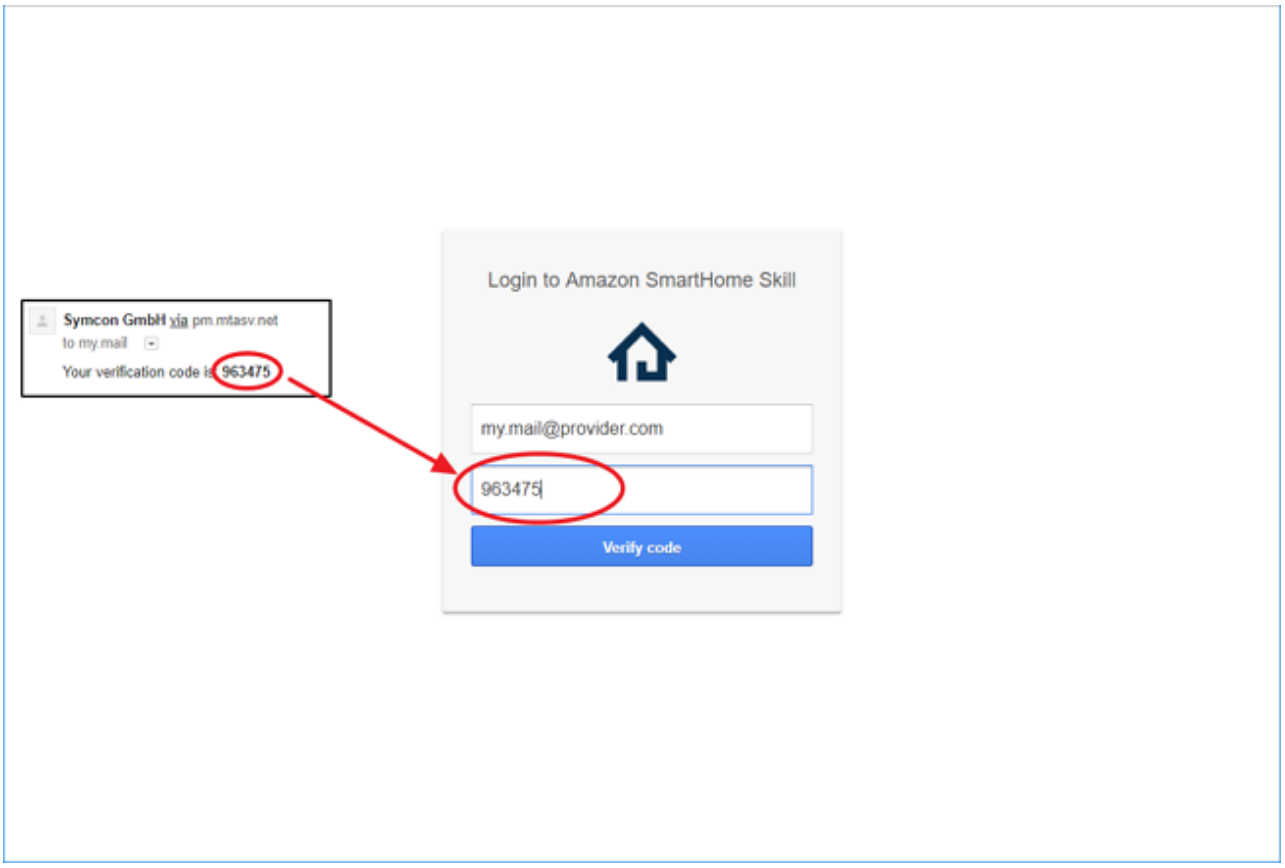



Dort wird in der Suche der Skill "Symcon" gesucht, auf den ersten Eintrag "Symcon" geklickt und der Button "Zur Verwendung aktivieren" gedrückt.




Dann sollte sich der Browser öffnen. Im Feld die E-Mail-Adresse eingeben, auf welche die

verwendete IP-Symcon-Lizenz registriert ist, und mit "Send verification code" bestätigen. Es wird eine E-Mail mit einem Code an die angegebene E-Mail-Adresse geschickt. Den Code, der in dieser E-Mail steht, im Popup eintragen und mit "Verify code" bestätigen.




Eine Meldung "Symcon wurde erfolgreich verknüpft" sollte erscheinen. Abschließend müssen noch die eingerichteten Geräte von Amazon Alexa gesucht werden. Dies kann beispielsweise getan werden, indem "Alexa, suche nach Geräten" gesagt wird.

# Expertenoptionen

Benötigt: IP-Symcon >= 5.0



Die Expertenoptionen müssen üblicherweise nicht angepasst werden. Gehen Sie mit diesen Optionen sorgsam um, damit das Amazon Alexa Modul wie vorgesehen funktioniert.

## OPTIONEN

| NAME | BESCHREIBUNG |
|---|---|
| Status emulieren | Ist Status emulieren aktiv, so wartet das Modul nicht auf die Aktualisierung der geschalteten Variablen und gibt stattdessen einen neuen Zustand auf Basis des geschalteten Wertes zurück. |
| Expertengeräte anzeigen | Ist Expertengeräte anzeigen aktiviert, so werden Expertengeräte angezeigt. Expertengeräte sind Geräte, welche in der Konfiguration komplexer sind und ein tieferes Verständniss für IP-Symcon erfordern. |

# Fernsehgerät

Benötigt: IP-Symcon >= 5.3

Fernsehgerät ist ein Expertengerät und wird nur angezeigt, wenn die entsprechende Expertenoption gesetzt ist.

## BESCHREIBUNG

Bei Geräten vom Typ Fernseher können diverse Parameter gesetzt werden. Das Gerät kann an und aus geschaltet werden, den Kanal wechseln, die Lautstärke einstellen, unterstützt Stummschaltung und kann den Eingang umstellen. Unterstützt ein Fernsehgerät nur manche dieser Eigenschaften, muss einfach nur der entsprechende Teil der Variablen definiert werden.

## PARAMETER

| NAME | BESCHREIBUNG |
|---|---|
| Name | Name mit dem das Gerät über Amazon Alexa angesprochen werden kann |
| Schaltervariable | Eine schaltbare Variable vom Typ Boolean, über welche das Gerät An oder Aus geschaltet wird (optional) |
| Kanalvariable | Eine schaltbare Variable vom Typ Integer mit einem Profil mit Assoziationen und Schrittweite 0. Die Assoziationen müssen durchnummeriert sein und bei 0 oder 1 beginnen (optional) |
| Lautstärkevariable | Eine schaltbare Variable vom Typ Integer oder Float, welche die Lautstärke kontrolliert (optional) |
| Stummvariable | Eine schaltbare Variable vom Typ Boolean, welche die Stummschaltung kontrolliert (optional) |
| Eingangsvariable | Eine schaltbare Variable vom Typ String, welche den verwendeten Eingang kontrolliert (optional) |
| Unterstützte Eingänge | In dieser Liste können die unterstützten Eingänge für die Eingangsvariable definiert werden. Ist die Eingangsvariable definiert, muss mindestens ein Eingang ausgewählt werden |

Mindestens eine der optionalen Variable muss definiert werden

Werden optionale Variablen nicht gesetzt, so erkennt Alexa, dass das Gerät die dazugehörigen Eigenschaften nicht besitzt. Werden optionale Variablen erst später gesetzt, so muss eine erneute Suche nach Geräten durchgeführt werden bevor die neu freigeschaltete Funktion genutzt werden kann.

### MÖGLICHE AKTIONEN

Beim Schalten von Lautstärken versteht Alexa die Anweisung häufig nicht, wenn auf einen prozentualen Wert geschaltet wird obwohl es um Prozentwerte geht. Daher sollte das "Prozent" beim Sprachbefehl weggelassen werden, also "schalte auf 40" statt "schalte auf 40 Prozent".

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
|---|---|---|
| An oder Aus schalten | Schaltet die Schaltervariable auf True oder False | "Alexa, schalte *<Name>* an." |
| Auf Kanal schalten | Schaltet die Kanalvariable auf den Wert, dessen Assoziation dem genannten Kanal entspricht. Hierbei wird keine Groß- und Kleinschreibung betrachtet. Auch Leerzeichen werden ignoriert. | "Alexa, schalte *<Name>* auf ARD." |
| Kanal weiterschalten | Erhöht die Kanalvariable um 1. Steht die Variable aktuell auf dem letzten Kanal, wird stattdessen zum ersten Kanal gesprungen. | "Alexa, schalte *<Name>* einen Kanal weiter." |
| Lautstärke einstellen | Schaltet die Lautstärkevariable auf den angegebenen Wert | "Alexa, schalte die Lautstärke von *<Name>* auf 40." |
| Stumm oder Laut schalten | Schaltet die Stummvariable auf True oder False | "Alexa, schalte *<Name>* stumm." |
| Eingang umschalten | Schalte die Eingangsvariable auf den Namen des Eingangs | "Alexa, *<Name>* auf HDMI 1 umschalten" |

## GERÄTESUCHE

Dieser Gerätetyp wird nicht von der Gerätesuche unterstützt.

# Generischer Schalter

Benötigt: IP-Symcon >= 5.0

## BESCHREIBUNG

Geräte vom Typ generischer Schalter können An und Aus geschaltet werden. Innerhalb von Amazon Alexa werden die Geräte als allgemeine Schalter interpretiert und können für verschiedenste Geräte verwendet werden.

## PARAMETER

| NAME | BESCHREIBUNG |
|---|---|
| Name | Name mit dem das Gerät über Amazon Alexa angesprochen werden kann |
| Schaltervariable | Eine schaltbare Variable vom Typ Boolean |

### MÖGLICHE AKTIONEN

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
|---|---|---|
| An oder Aus schalten | Schaltet die Variable auf True oder False | "Alexa, schalte *<Name>* an." |

### GERÄTESUCHE

Eine Instanz muss Statusvariablen mit den passenden Profilen haben, um in der Gerätesuche erkannt zu werden.

| VARIABLE | MÖGLICHE PROFILE |
|---|---|
| Schaltervariable | ~Switch |

# Generischer Schieberegler

Benötigt: IP-Symcon >= 5.0

## BESCHREIBUNG

Geräte vom Typ generischer Slider können auf einen prozentualen Wert geschaltet werden. Innerhalb von Amazon Alexa werden die Geräte als allgemeine Schalter interpretiert und können für verschiedenste Geräte verwendet werden.

## PARAMETER

| NAME | BESCHREIBUNG |
|---|---|
| Name | Name mit dem das Gerät über Amazon Alexa angesprochen werden kann |
| Reglervariable | Eine schaltbare Variable vom Typ Integer oder Float |

### MÖGLICHE AKTIONEN

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
|---|---|---|
| An oder Aus schalten | Schaltet die Variable auf 100% oder 0% | "Alexa, schalte *<Name>* an." |
| Wert einstellen | Schaltet die Variable auf den angegebenen Wert | "Alexa, schalte *<Name>* auf 40%." |

### GERÄTESUCHE

Eine Instanz muss Statusvariablen mit den passenden Profilen haben, um in der Gerätesuche erkannt zu werden.

| VARIABLE | MÖGLICHE PROFILE |
|---|---|
| Reglervariable | ~Intensity.100, ~Intensity.255, ~Intensity.1 |

# Lautsprecher

Benötigt: IP-Symcon >= 5.0

## BESCHREIBUNG

Bei Geräten vom Typ Lautsprecher kann die Lautstärke eingestellt werden.

## PARAMETER

| NAME | BESCHREIBUNG |
| --- | --- |
| Name | Name mit dem das Gerät über Amazon Alexa angesprochen werden kann |
| Lautstärkevariable | Eine schaltbare Variable vom Typ Integer oder Float, welche die Lautstärke kontrolliert |

### MÖGLICHE AKTIONEN



Beim Schalten von Lautstärken versteht Alexa die Anweisung häufig nicht, wenn auf einen prozentualen Wert geschaltet wird obwohl es um Prozentwerte geht. Daher sollte das "Prozent" beim Sprachbefehl weggelassen werden, also "schalte auf 40" statt "schalte auf 40 Prozent".

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
| --- | --- | --- |
| Lautstärke einstellen | Schaltet die Variable auf den angegebenen Wert | "Alexa, schalte die Lautstärke von *<Name>* auf 40." |

### GERÄTESUCHE

Dieser Gerätetyp wird nicht von der Gerätesuche unterstützt.

# Lautsprecher mit Stummschaltung

Benötigt: IP-Symcon >= 5.3

## BESCHREIBUNG

Bei Geräten vom Typ Lautsprecher mit Stummschaltung kann die Lautstärke eingestellt werden. Zusätzlich kann das Gerät stumm und wieder laut geschaltet werden ohne die eingestellte Lautstärke zu verlieren.

## PARAMETER

| NAME | BESCHREIBUNG |
|---|---|
| Name | Name mit dem das Gerät über Amazon Alexa angesprochen werden kann |
| Lautstärkevariable | Eine schaltbare Variable vom Typ Integer oder Float, welche die Lautstärke kontrolliert (optional) |
| Stummvariable | Eine schaltbare Variable vom Typ Boolean, welche die Stummschaltung kontrolliert (optional) |

Entweder Lautstärkevariable oder Stummvariable muss definiert sein



Werden optionale Variablen nicht gesetzt, so erkennt Alexa, dass das Gerät die dazugehörigen Eigenschaften nicht besitzt. Werden optionale Variablen erst später gesetzt, so muss eine erneute Suche nach Geräten durchgeführt werden bevor die neu freigeschaltete Funktion genutzt werden kann.

### MÖGLICHE AKTIONEN



Beim Schalten von Lautstärken versteht Alexa die Anweisung häufig nicht, wenn auf einen prozentualen Wert geschaltet wird obwohl es um Prozentwerte geht. Daher sollte das "Prozent" beim Sprachbefehl weggelassen werden, also "schalte auf 40" statt "schalte auf 40 Prozent".

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
|---|---|---|
| Lautstärke einstellen | Schaltet die Lautstärkevariable auf den angegebenen Wert | "Alexa, schalte die Lautstärke von *<Name>* auf 40." |
| Stumm oder Laut schalten | Schaltet die Stummvariable auf True oder False | "Alexa, schalte *<Name>* stumm." |

### GERÄTESUCHE

Dieser Gerätetyp wird nicht von der Gerätesuche unterstützt.

# Licht (Dimmer)

Benötigt: IP-Symcon >= 5.0

## BESCHREIBUNG

Geräte vom Typ Licht (Dimmer) beschreiben Lampen, welche gedimmt werden können.

## PARAMETER

| NAME | BESCHREIBUNG |
| --- | --- |
| Name | Name mit dem das Gerät über Amazon Alexa angesprochen werden kann |
| Helligkeitsvariable | Eine schaltbare Variable vom Typ Integer oder Float, über welche das Licht gedimmt wird |

### MÖGLICHE AKTIONEN

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
| --- | --- | --- |
| An oder Aus schalten | Schaltet die Variable auf 100% oder 0% | "Alexa, schalte *<Name>* an." |
| Dimmen | Schaltet die Variable auf den angegebenen Wert | "Alexa, dimme *<Name>* auf 40%." |

### GERÄTESUCHE

Eine Instanz muss Statusvariablen mit den passenden Profilen haben, um in der Gerätesuche erkannt zu werden.

| VARIABLE | MÖGLICHE PROFILE |
| --- | --- |
| Helligkeitsvariable | ~Intensity.100, ~Intensity.255, ~Intensity.1 |

# Licht (Experte)

Benötigt: IP-Symcon >= 5.0

## BESCHREIBUNG

Geräte vom Typ Licht (Experte) beschreiben Lampen, deren Eigenschaften in verschiedenen Variablen unabhängig voneinander geschaltet werden können.

## PARAMETER

| NAME | BESCHREIBUNG |
|---|---|
| Name | Name mit dem das Gerät über Amazon Alexa angesprochen werden kann |
| Schaltervariable | Eine schaltbare Variable vom Typ Boolean, über welche die Lampe An oder Aus geschaltet wird |
| Helligkeitsvariable | Eine schaltbare Variable vom Typ Integer oder Float, über welche das Licht gedimmt wird (optional) |
| Farbvariable | Eine schaltbare Variable vom Typ Integer mit dem Profil ~HexColor, über welche die Farbe des Lichts geschaltet wird (optional) |
| Farbtemperaturvariable | Eine schaltbare Variable vom Typ Integer oder Float, welche die Farbtemperatur in Kelvin darstellt (optional) |



Werden optionale Variablen nicht gesetzt, so erkennt Alexa, dass die Lampe die dazugehörigen Eigenschaften nicht besitzt. Werden optionale Variablen erst später gesetzt, so muss eine erneute Suche nach Geräten durchgeführt werden bevor die neu freigeschaltete Funktion genutzt werden kann.

### MÖGLICHE AKTIONEN

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
|---|---|---|
| An oder Aus schalten | Schaltet die Schaltervariable auf True oder False | "Alexa, schalte *<Name>* an." |
| Dimmen | Schaltet die Helligkeitsvariable auf den angegebenen Wert | "Alexa, dimme *<Name>* auf 40%." |
| Farbe schalten | Schaltet die Farbvariable auf die angegebene Farbe | "Alexa, schalte *<Name>* auf rot." |
| Farbtemperatur schalten | Schaltet die Farbtemperaturvariable auf den passenden Wert | "Alexa, schalte *<Name>* auf Tageslicht" |

### GERÄTESUCHE

Eine Instanz muss Statusvariablen mit den passenden Profilen haben, um in der Gerätesuche erkannt zu werden. Um als Expertenlicht erkannt zu werden, muss mindestens die Schaltervariable und eine weitere Variable erkannt werden.

| VARIABLE | MÖGLICHE PROFILE |
|---|---|
| Schaltervariable | ~Switch |
| Helligkeitsvariable | ~Intensity.100, ~Intensity.255, ~Intensity.1 |
| Farbvariable | ~HexColor |
| Farbtemperaturvariable | ~TWColor |

# Licht (Farbe)

Benötigt: IP-Symcon >= 5.0

## BESCHREIBUNG

Geräte vom Typ Licht (Farbe) beschreiben Lampen, welche beliebige Farben annehmen können.

## PARAMETER

| NAME | BESCHREIBUNG |
|---|---|
| Name | Name mit dem das Gerät über Amazon Alexa angesprochen werden kann |
| Farbvariable | Eine schaltbare Variable vom Typ Integer mit dem Profil ~HexColor, über welche die Farbe des Lichts geschaltet wird |

### MÖGLICHE AKTIONEN

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
|---|---|---|
| An oder Aus schalten | Schaltet die Variable auf Weiß oder Schwarz | "Alexa, schalte *<Name>* an." |
| Dimmen | Schaltet Helligkeit der aktuellen Farbe der Variable auf den angegebenen Wert | "Alexa, dimme *<Name>* auf 40%." |
| Farbe schalten | Schaltet die Variable auf die angegebene Farbe | "Alexa, schalte *<Name>* auf rot." |

### GERÄTESUCHE

Eine Instanz muss Statusvariablen mit den passenden Profilen haben, um in der Gerätesuche erkannt zu werden.

| VARIABLE | MÖGLICHE PROFILE |
|---|---|
| Farbvariable | ~HexColor |

# Licht (Schalter)

Benötigt: IP-Symcon >= 5.0

## BESCHREIBUNG

Geräte vom Typ Licht (Schalter) beschreiben Lampen, welche an- und ausgeschaltet werden können.

## PARAMETER

| NAME | BESCHREIBUNG |
|---|---|
| Name | Name mit dem das Gerät über Amazon Alexa angesprochen werden kann |
| Schaltervariable | Eine schaltbare Variable vom Typ Boolean, über welche das Licht An oder Aus geschaltet wird |

### MÖGLICHE AKTIONEN

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
|---|---|---|
| An oder Aus schalten | Schaltet die Variable auf True oder False | "Alexa, schalte *<Name>* an." |

### GERÄTESUCHE

Eine Instanz muss Statusvariablen mit den passenden Profilen haben, um in der Gerätesuche erkannt zu werden.

| VARIABLE | MÖGLICHE PROFILE |
|---|---|
| Schaltervariable | ~Switch |

# Medienabspieler

Benötigt: IP-Symcon >= 5.4

## BESCHREIBUNG

Bei Geräten vom Typ Medienabspieler können diverse Parameter gesetzt werden. Das Gerät kann an und aus geschaltet werden, die Lautstärke wechseln, unterstützt Stummschaltung und kann die Wiedergabe steuern. Unterstützt ein Medienabspieler nur manche dieser Eigenschaften, muss einfach nur der entsprechende Teil der Variablen definiert werden.

## PARAMETER

| NAME | BESCHREIBUNG |
|---|---|
| Name | Name mit dem das Gerät über Amazon Alexa angesprochen werden kann |
| Schaltervariable | Eine schaltbare Variable vom Typ Boolean, über welche das Gerät An oder Aus geschaltet wird (optional) |
| Lautstärkevariable | Eine schaltbare Variable vom Typ Integer oder Float, welche die Lautstärke kontrolliert (optional) |
| Stummvariable | Eine schaltbare Variable vom Typ Boolean, welche die Stummschaltung kontrolliert (optional) |
| Abspielvariable | Eine schaltbare Variable vom Typ Integer mit dem Profil ~Playback oder ~PlaybackPreviousNext, welche den Zustand der Wiedergabe darstellt; Je nach Profil bietet Alexa auch nur die dazugehörigen Aktionen an, also Stop, Play und Pause für ~Playback und zusätzlich Zurück und Weiter für ~PlackbackPreviousNext (optional) |

Mindestens eine der optionalen Variablen muss definiert werden



Werden optionale Variablen nicht gesetzt, so erkennt Alexa, dass das Gerät die dazugehörigen Eigenschaften nicht besitzt. Werden optionale Variablen erst später gesetzt, so muss eine erneute Suche nach Geräten durchgeführt werden bevor die neu freigeschaltete Funktion genutzt werden kann.

### MÖGLICHE AKTIONEN



Beim Schalten von Lautstärken versteht Alexa die Anweisung häufig nicht, wenn auf einen prozentualen Wert geschaltet wird obwohl es um Prozentwerte geht. Daher sollte das "Prozent" beim Sprachbefehl weggelassen werden, also "schalte auf 40" statt "schalte auf 40 Prozent".

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
|---|---|---|
| An oder Aus schalten | Schaltet die Schaltervariable auf True oder False | "Alexa, schalte *<Name>* an." |
| Lautstärke einstellen | Schaltet die Lautstärkevariable auf den angegebenen Wert | "Alexa, schalte die Lautstärke von *<Name>* auf 40." |
| Stumm oder Laut schalten | Schaltet die Stummvariable auf True oder False | "Alexa, schalte *<Name>* stumm." |
| Wiedergabe einstellen | Schaltet die Abspielvariable auf den angegebenen Wert | "Alexa, *<Name>* abspielen." |

## GERÄTESUCHE

Eine Instanz muss Statusvariablen mit den passenden Profilen haben, um in der Gerätesuche erkannt zu werden. Um als Medienabspieler erkannt zu werden, muss mindestens die Abspielvariable erkannt werden.

| VARIABLE | MÖGLICHE PROFILE |
|---|---|
| Schaltervariable | ~Switch |
| Lautstärkevariable | Diese Variable wird bei der Gerätesuche nicht erkannt |
| Stummvariable | Diese Variable wird bei der Gerätesuche nicht erkannt |
| Abspielvariable | ~Playback, ~PlaybackPreviousNext |

# Rollladen

Benötigt: IP-Symcon >= 5.3

## BESCHREIBUNG

Geräte vom Typ Rollladen können geöffnet und geschlossen werden. Unterstützt das Gerät eine prozentuale Öffnung, so ist diese ebenfalls steuerbar.

## PARAMETER

| NAME | BESCHREIBUNG |
|---|---|
| Name | Name mit dem das Gerät über Amazon Alexa angesprochen werden kann |
| Rollladenvariable | Eine schaltbare Variable vom Typ Integer oder Float, welche das Profil ~ShutterMoveStop oder ~ShutterMoveStep unterstützt. Alternativ kann eine Variable verwendet werden, welcher durch ihr Profil ein Minimal- und Maximalwert zugewiesen wurde |

### MÖGLICHE AKTIONEN

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
|---|---|---|
| Öffnen | Schaltet die Variable bei einem ~ShutterMove-Profil auf "Öffnen". Ansonsten wird die Variable auf den Minimalwert des Profils geschaltet. | "Alexa, öffne *<Name>*" |
| Schließen | Schaltet die Variable bei einem ~ShutterMove-Profil auf "Schliessen". Ansonsten wird die Variable auf den Maximalwert des Profils geschaltet. | "Alexa, schließe *<Name>*" |
| Prozentual öffnen | Schaltet die Variable bei einem ~ShutterMove-Profil auf "Öffnen" falls der Prozentsatz größer als 50% ist, sonst auf "Schliessen". Ansonsten wird die Variable entsprechend des Prozentsatzes geschaltet. Es ist zu beachten, dass bei einer vollen Öffnung die Variable auf 0% geschaltet wird und dies bei anderen Prozentwerten auch entsprechend angewandt wird. Wird also beispielsweise zu 30% geöffnet, so wird die Variable auf 70% geschaltet. | "Alexa, öffne *<Name>* zu 30%" |

### GERÄTESUCHE

Eine Instanz muss Statusvariablen mit den passenden Profilen haben, um in der Gerätesuche erkannt zu werden.

| VARIABLE | MÖGLICHE PROFILE |
| --- | --- |
| Rollladenvariable | ~ShutterMoveStop, ~ShutterMoveStep, ~Intensity.100, ~Intensity.255, ~Intensity.1 |

# Schloss

Benötigt: IP-Symcon >= 5.0

## BESCHREIBUNG

Geräte vom Typ Schloss können verschlossen werden.

## PARAMETER

| NAME | BESCHREIBUNG |
|---|---|
| Name | Name mit dem das Gerät über Amazon Alexa angesprochen werden kann |
| Schlossvariable | Eine schaltbare Variable vom Typ Boolean, über welche das Schloss geschaltet wird |

### MÖGLICHE AKTIONEN



Bevor ein Schloss entriegelt werden kann, muss dies in der Alexa App aktiviert werden.

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
|---|---|---|
| Verriegeln | Schaltet die Variable auf True | "Alexa, verschließe *<Name>*." |
| Entriegeln | Schaltet die Variable auf False | "Alexa, entriegle *<Name>*." |

### GERÄTESUCHE

Eine Instanz muss Statusvariablen mit den passenden Profilen haben, um in der Gerätesuche erkannt zu werden.

| VARIABLE | MÖGLICHE PROFILE |
|---|---|
| Schlossvariable | ~Lock, ~Lock.Reversed, ~Door, ~Door.Reversed |

# Szenen

Benötigt: IP-Symcon >= 5.0

## BESCHREIBUNG

Geräte vom Typ Szene können aktiviert werden und beliebige Aktionen auslösen.

## PARAMETER

| NAME | BESCHREIBUNG |
| --- | --- |
| Name | Name mit dem das Gerät über Amazon Alexa angesprochen werden kann |
| Aktion beim Aktivieren | Eine Aktion, welche beim Aktivieren der Szene ausgeführt wird |

### MÖGLICHE AKTIONEN

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
| --- | --- | --- |
| Szene aktivieren | Führt die Aktion aus | "Alexa, aktiviere *<Name>*." |

### GERÄTESUCHE

Dieser Gerätetyp wird nicht von der Gerätesuche unterstützt.

# Szenen (deaktivierbar)

Benötigt: IP-Symcon >= 5.0

## BESCHREIBUNG

Geräte vom Typ Szene können aktiviert und deaktiviert werden und dabei beliebige Aktionen auslösen.

## PARAMETER

| NAME | BESCHREIBUNG |
|---|---|
| Name | Name mit dem das Gerät über Amazon Alexa angesprochen werden kann |
| Aktion beim Aktivieren | Eine Aktion, welche beim Aktivieren der Szene ausgeführt wird |
| Aktion beim Deaktivieren | Eine Aktion, welche beim Deaktivieren der Szene ausgeführt wird |

### MÖGLICHE AKTIONEN

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
|---|---|---|
| Szene aktivieren | Führt die Aktion zum Aktivieren aus | "Alexa, aktiviere *<Name>*." |
| Szene deaktivieren | Führt die Aktion zum Deaktivieren aus | "Alexa, deaktiviere *<Name>*." |

### GERÄTESUCHE

Dieser Gerätetyp wird nicht von der Gerätesuche unterstützt.

# Temperatursensor

Benötigt: IP-Symcon >= 5.0

## BESCHREIBUNG

Geräte vom Typ Temperatursensor messen Temperaturen.

## PARAMETER

| NAME | BESCHREIBUNG |
|---|---|
| Name | Name mit dem das Gerät über Amazon Alexa angesprochen werden kann |
| Sensorvariable | Eine Variable vom Typ Float, welche die aktuelle Temperatur in °C enthält |

### MÖGLICHE AKTIONEN



Momentan scheint Alexa keine direkten Anfragen an den Temperatursensor zu unterstützen. Daher muss für eine sprachliche Abfrage der Umweg über eine Gruppe gegangen werden. Sobald der Temperatursensor Teil einer Gruppe ist, kann die Temperatur der Gruppe abgefragt werden.

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
|---|---|---|
| Temperatur abfragen | Fragt die aktuelle Temperatur des Sensors ab | "Alexa, welche Temperatur hat *<Gruppe>*?" |

### GERÄTESUCHE

Dieser Gerätetyp wird nicht von der Gerätesuche unterstützt.

# Thermostat

Benötigt: IP-Symcon >= 5.0

## BESCHREIBUNG

Geräte vom Typ Thermostat können Temperaturen einstellen.

## PARAMETER

| NAME | BESCHREIBUNG |
|---|---|
| Name | Name mit dem das Gerät über Amazon Alexa angesprochen werden kann |
| Variable | Eine schaltbare Variable vom Typ Float, durch welche die Temperatur eingestellt wird |

### MÖGLICHE AKTIONEN

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
|---|---|---|
| Temperatur einstellen | Schalte die Variable auf die angegebene Temperatur. | "Alexa, schalte *<Name>* auf 35 °C." |

### GERÄTESUCHE

Eine Instanz muss Statusvariablen mit den passenden Profilen haben, um in der Gerätesuche erkannt zu werden.

| VARIABLE | MÖGLICHE PROFILE |
|---|---|
| Variable | ~Temperature, ~Temperature.Room |

# Google Assistant

Benötigt: IP-Symcon >= 5.0

## BESCHREIBUNG

Das Google Assistant-Modul ermöglicht die Steuerung von Geräten durch Google Assistant. Dies beinhaltet insbesondere die Sprachsteuerung über Google Home.

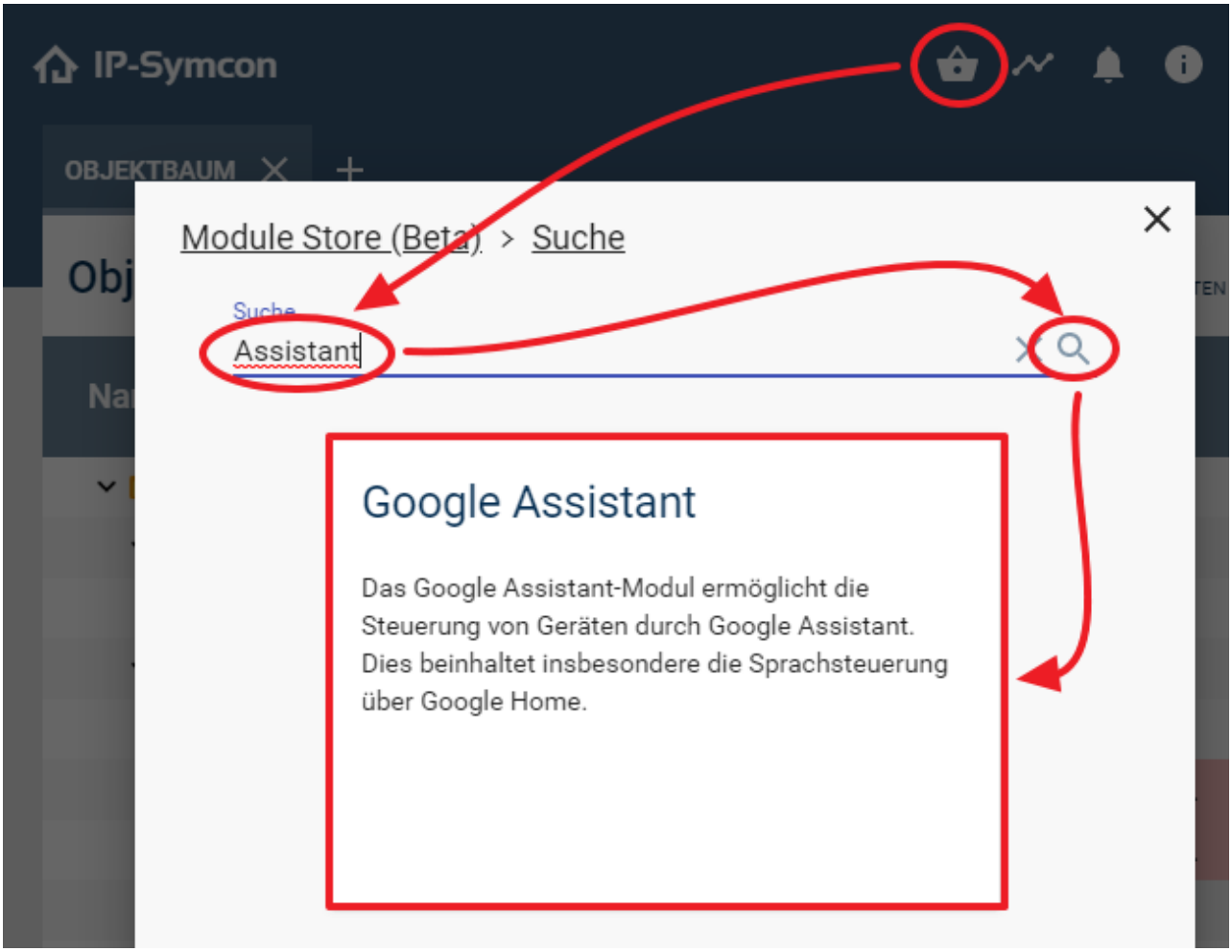
## EINBINDUNG IN IP-SYMCON

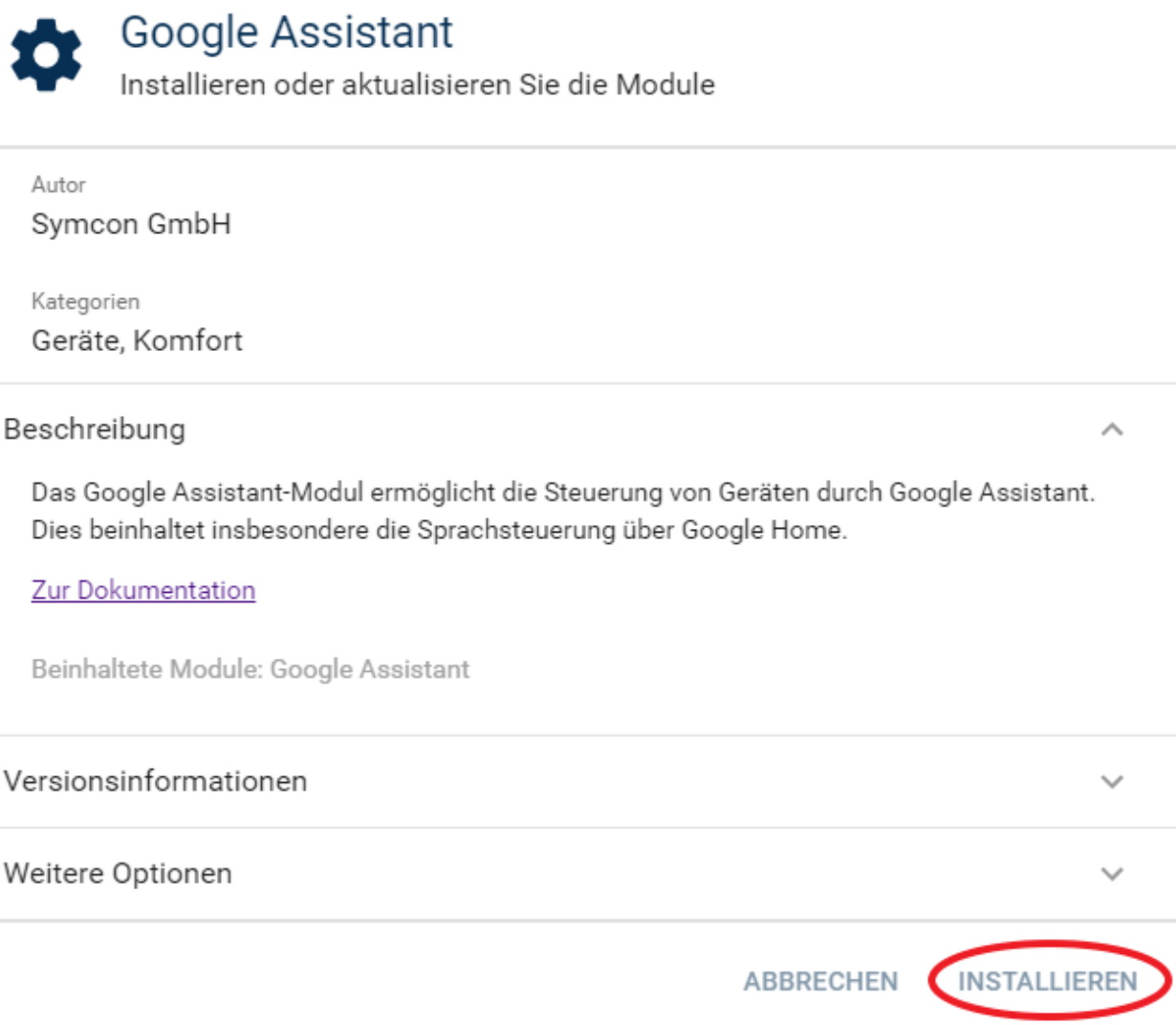


Die Verwendung des Google Assistant-Moduls erfordert eine aktivierte Verbindung zum Connect-Dienst, welche wiederum eine laufende Subskription erfordert.

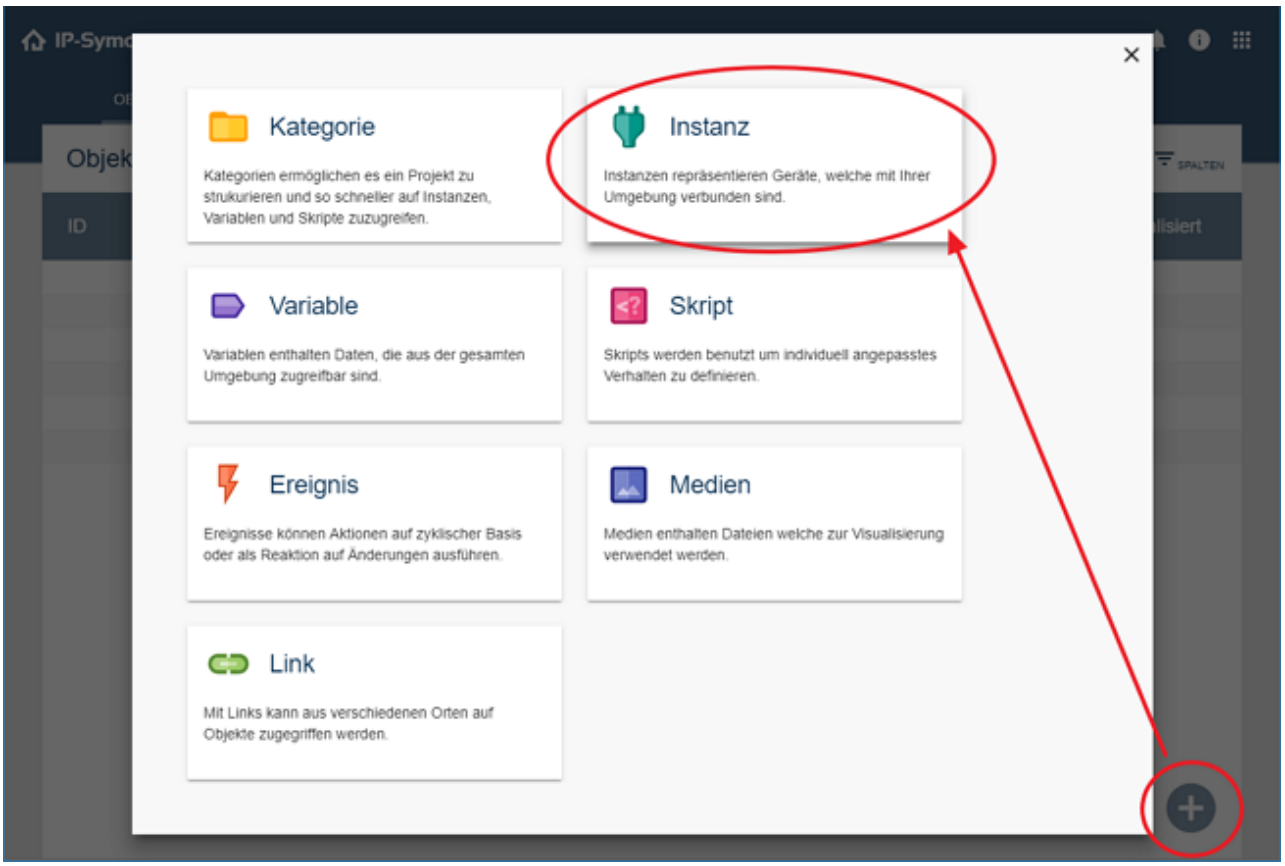
### GOOGLE ASSISTANT-INSTANZ ERSTELLEN

Damit das Google Assistant-Modul verwendet werden kann, muss dieses zuerst über den Module Store installiert werden. Hierfür muss der Module Store geöffnet werden. Dieser befindet sich im oberen rechten Bereich. Über das Suchfeld kann via "Assistant" das Modul Google Assistant gefunden werden. Beim öffnen des gefundenen Modules, kann im folgenden Dialog die Installation des Moduls via "Installieren"-Knopf angestoßen werden.

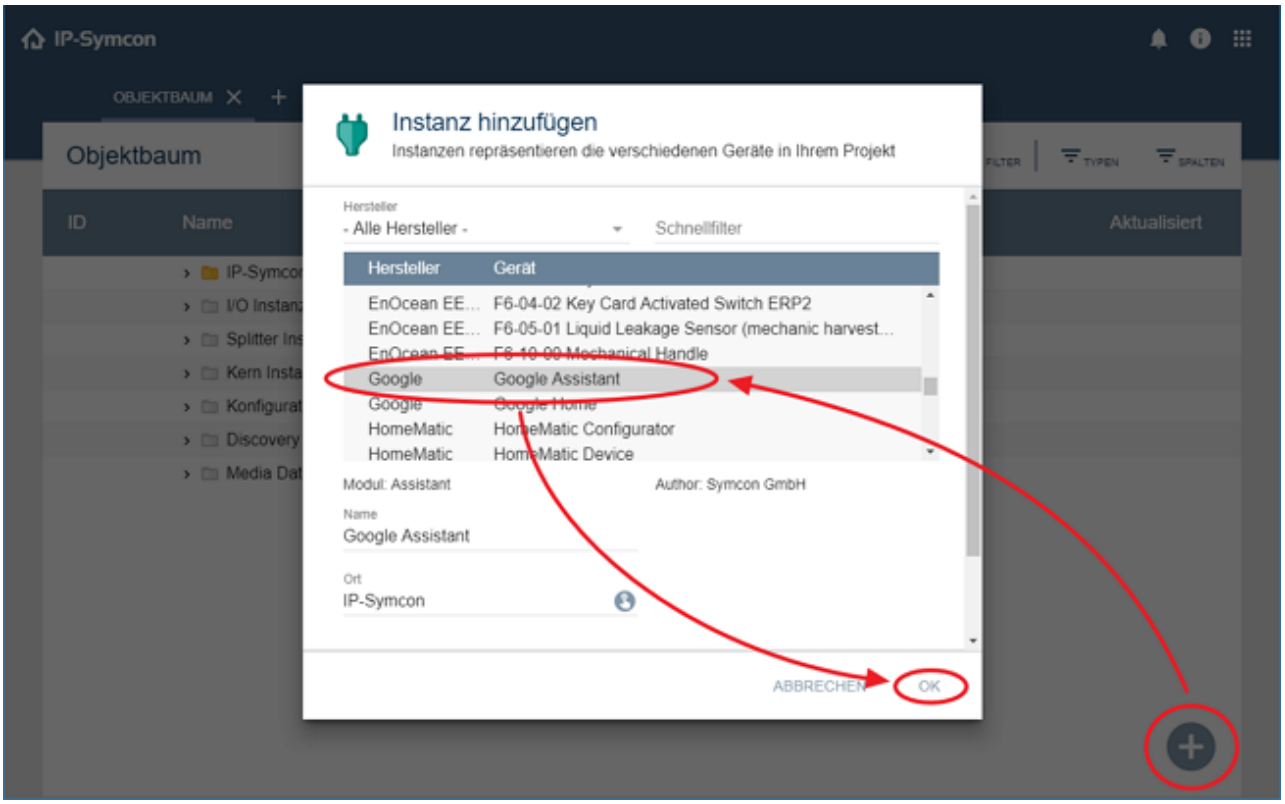

IP-Symcon
OBJEKTBAUM
Module Store (Beta) > Suche
Suche
Assistant
Google Assistant
Das Google Assistant-Modul ermöglicht die Steuerung von Geräten durch Google Assistant. Dies beinhaltet insbesondere die Sprachsteuerung über Google Home.




Im Anschluss öffnet sich der Hinzufügendialog zum Erstellen einer Google Assistant Instanz.
Soll die Instanz manuell hinzugefügt werden muss eine Instanz vom Google Assistant-Modul erstellt werden. Da muss zuerst der Objektbaum geöffnet werden. In diesem den Hinzufügen-Button "+" unten rechts betätigen und Instanz auswählen.



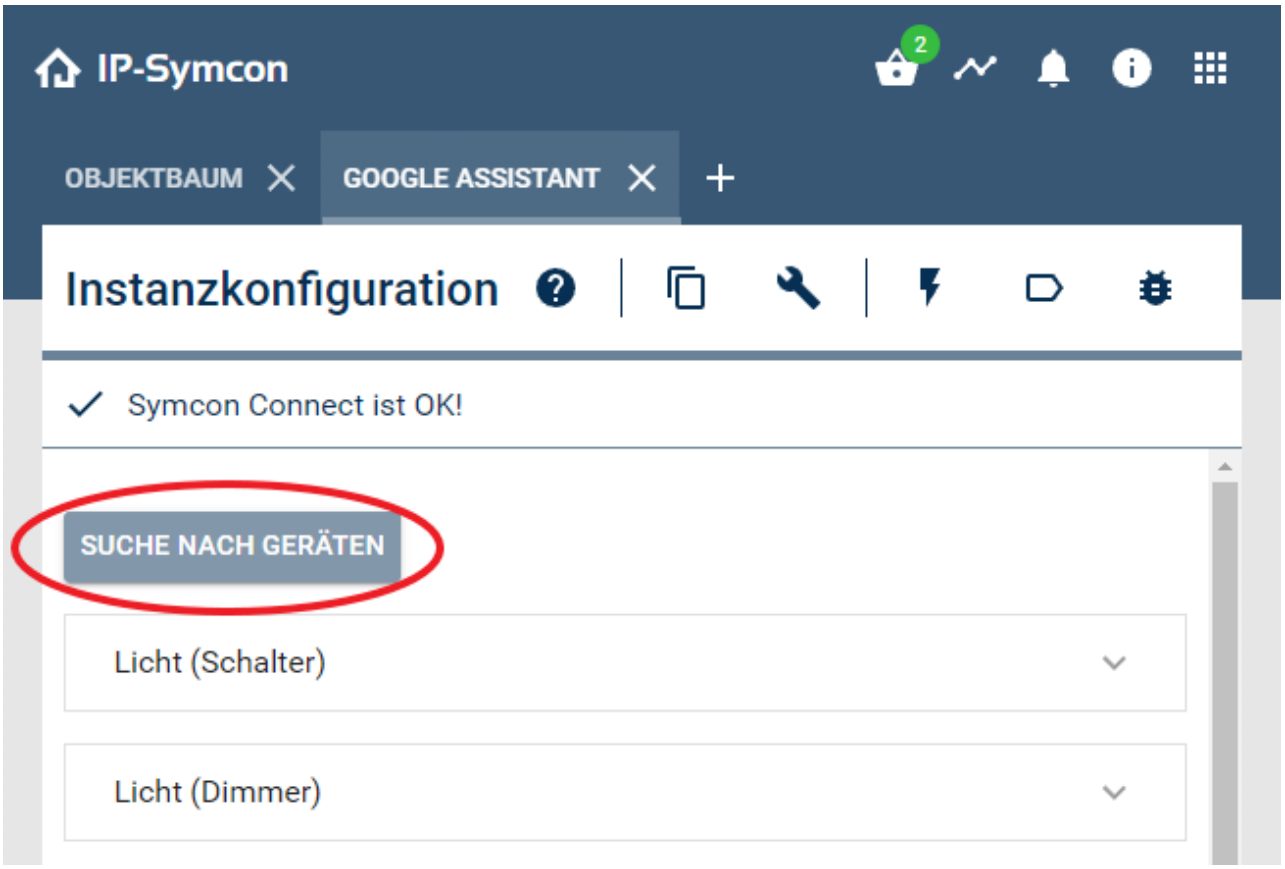

In der Modulliste das Gerät "Google Assistant" oder "Google Home" vom Hersteller "Google" auswählen. Der Ort sollte nicht verändert werden, der Name kann beliebig gewählt werden. Abschließend mit "OK" bestätigen.




Nach der Erstellung der Google Assistant-Instanz öffnet sich diese automatisch und kann eingerichtet werden.

## GOOGLE ASSISTANT-INSTANZ EINRICHTEN

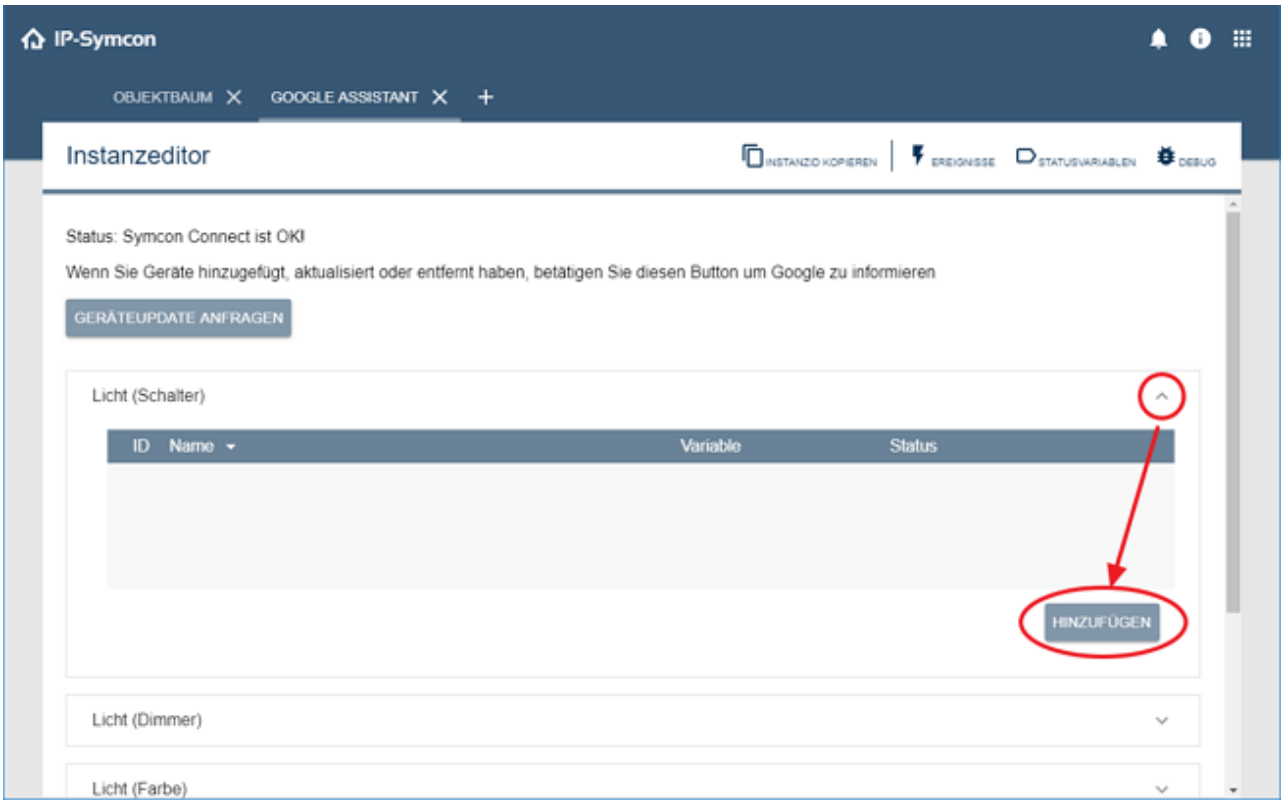
Das Modul bietet verschiedene Gerätetypen an, welche verwendet werden können. Inbesondere um die initiale Einrichtung zu beschleunigen, sollte die Gerätesuche verwendet werden. Später können weitere Geräte unkompliziert manuell hinzugefügt werden.
Um die Gerätesuche zu starten muss auf den Button "Suche nach Geräten" am oberen Ende der Instanzkonfiguration geklickt werden



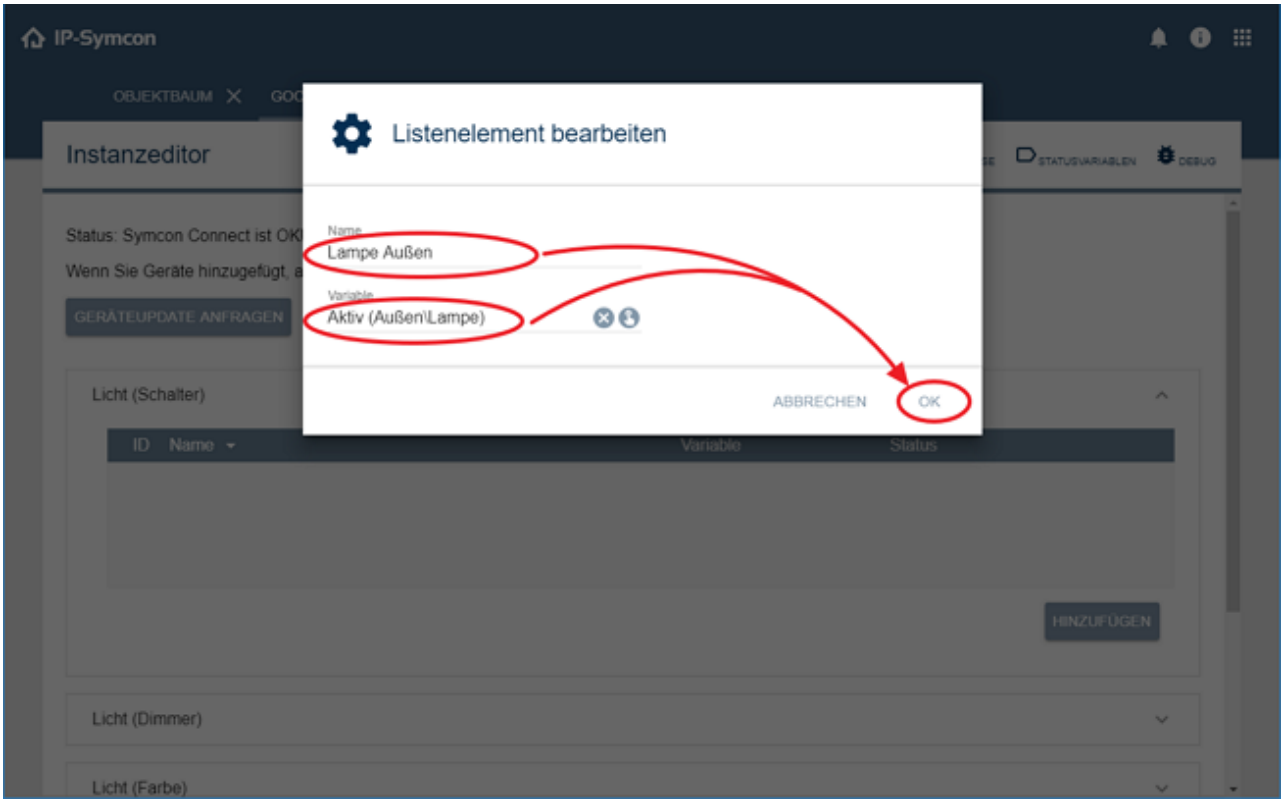

Daraufhin öffnet sich ein Dialog, welcher nach einer Ladezeit gefundene Geräte darstellt, welche bisher noch nicht eingebunden wurden. Die Instanzen werden nach Gerätetypen aufgeteilt. Unter jeder Instanz werden die Statusvariablen dargestellt, welche bei einer Einbindung vom Modul verwendet würden. Soll eine Instanz eingebunden werden, so muss die Checkbox in dieser Zeile ausgewählt werden. Standardmäßig wird der Name der Instanz verwendet. Soll dieser angepasst werden, kann dies über das "Bearbeiten"-Symbol rechts im Listeneintrag getan werden. Nachdem alle gewünschten Geräte ausgewählt und ggfs. passend benannt wurden, muss die Auswahl mit "Geräte hinzufügen" bestätigt werden. Dann werden die gewählten Geräte in die dazugehörigen Listen in der Instanzkonfiguration hinzugefügt.




Um ein Gerät manuell einzurichten, wird das dazugehörige Panel ausgeklappt und in der entsprechenden Liste durch einen Klick auf "Hinzufügen" ein neuer Eintrag erstellt.



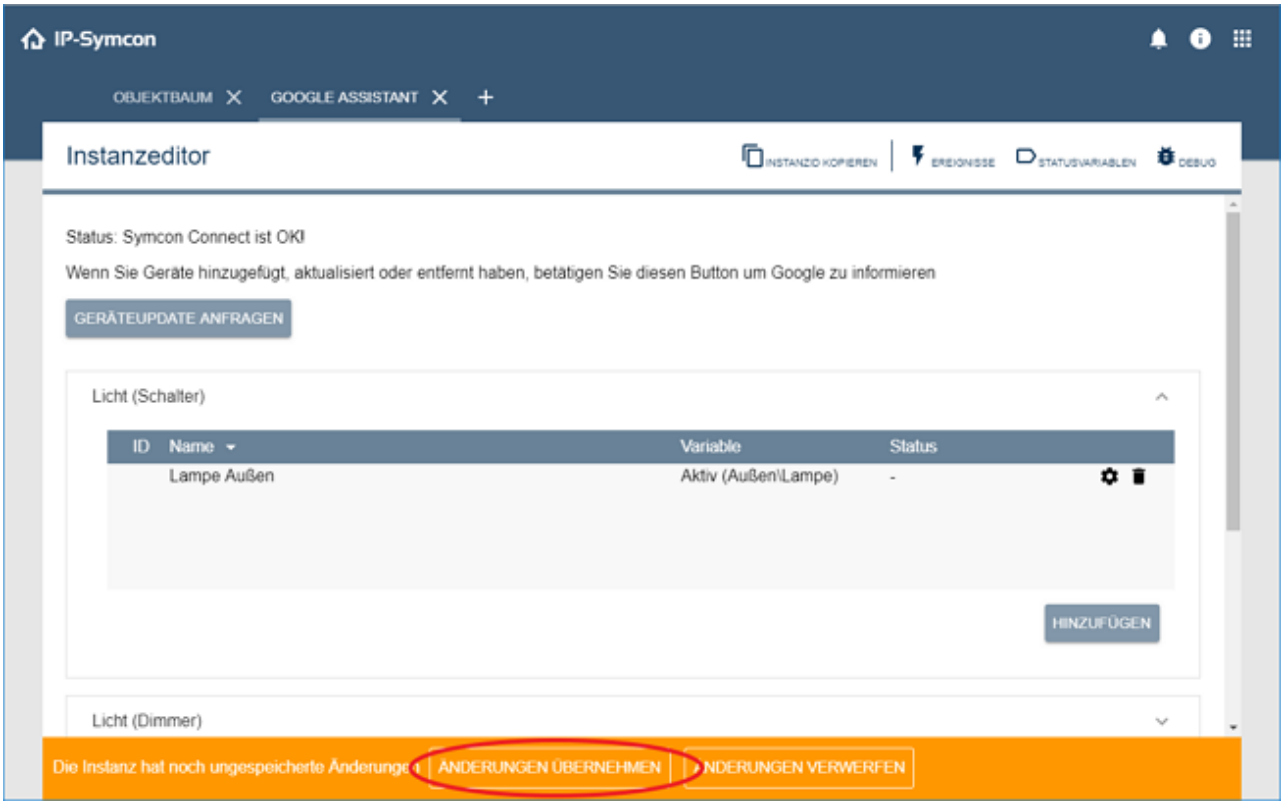

Jeder Eintrag erfordert einen Namen unter welchem das Gerät bei Google Assistant bekannt sein wird. Weitere Parameter hängen vom jeweiligen Gerätetyp ab und können den dazugehörigen Seiten der Dokumentation entnommen werden. Nachdem alle Parameter gesetzt sind, wird das Gerät mit "OK" bestätigt.



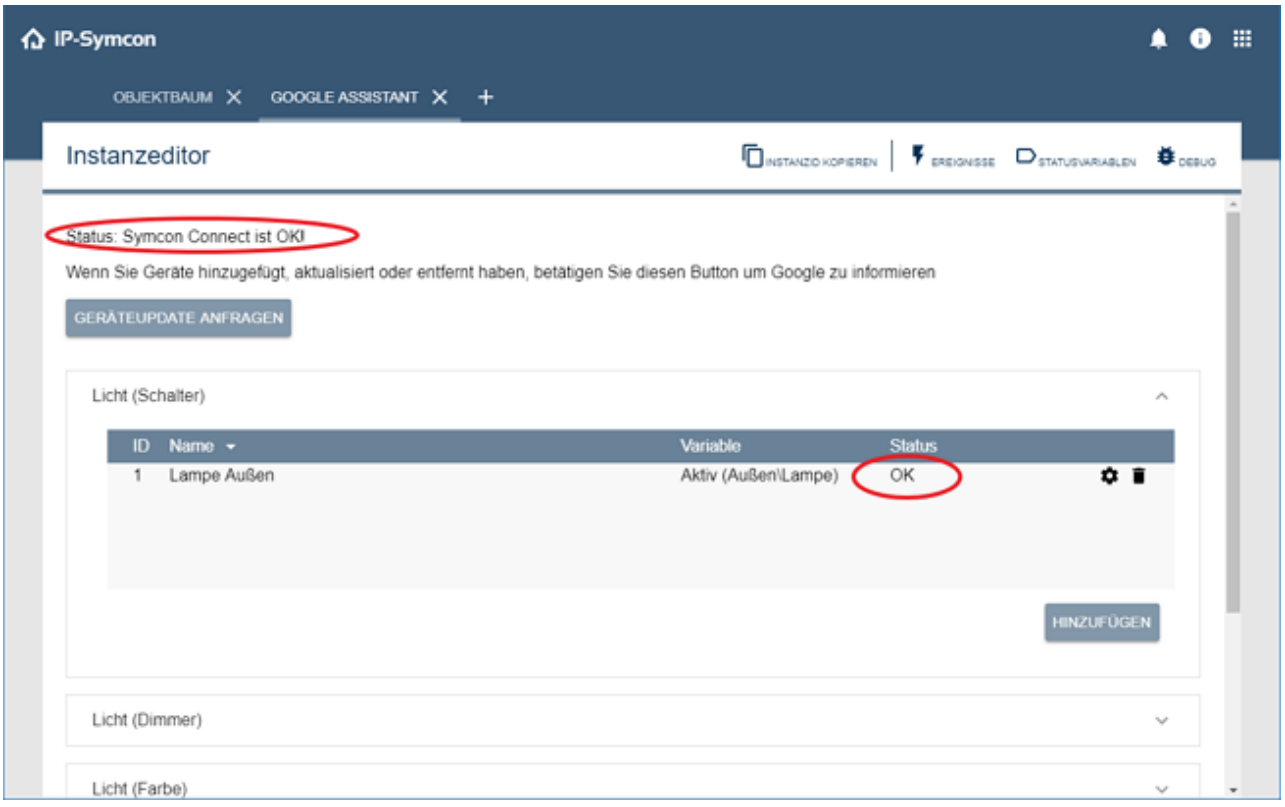

## MÖGLICHE GERÄTETYPEN

- Licht (Schalter)
- Licht (Dimmer)
- Licht (Farbe)
- Licht (Experte)
- Thermostat
- Rollladen
- Generischer Schalter
- Szenen
- Szenen (deaktivierbar)

Nachdem alle Geräte eingerichtet sind, werden diese mit einem Klick auf "Änderungen übernehmen" bestätigt.



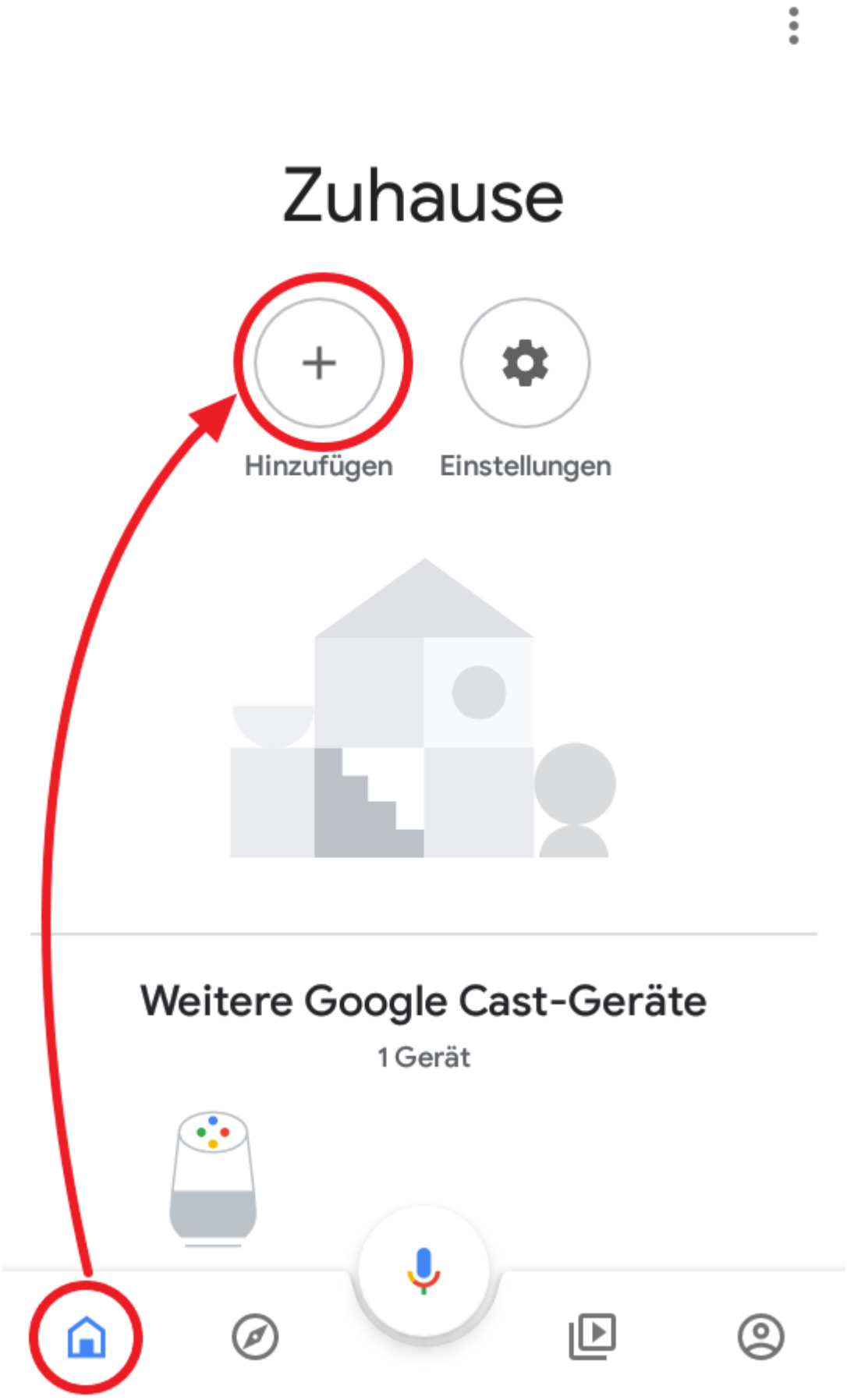

Nach dem Übernehmen sollte sichergestellt werden, dass bei allen Geräten "OK" in der Spalte "Status" steht. Ist dies nicht der Fall, so enthält die Spalte eine Fehlermeldung, welche über die Komplikation informiert.
Ebenfalls sollte geprüft werden, dass oben im Instanzeditor "Status: Symcon Connect ist OK" steht. Sollte dies nicht der Fall sein, muss der Connect-Dienst aktiviert werden.



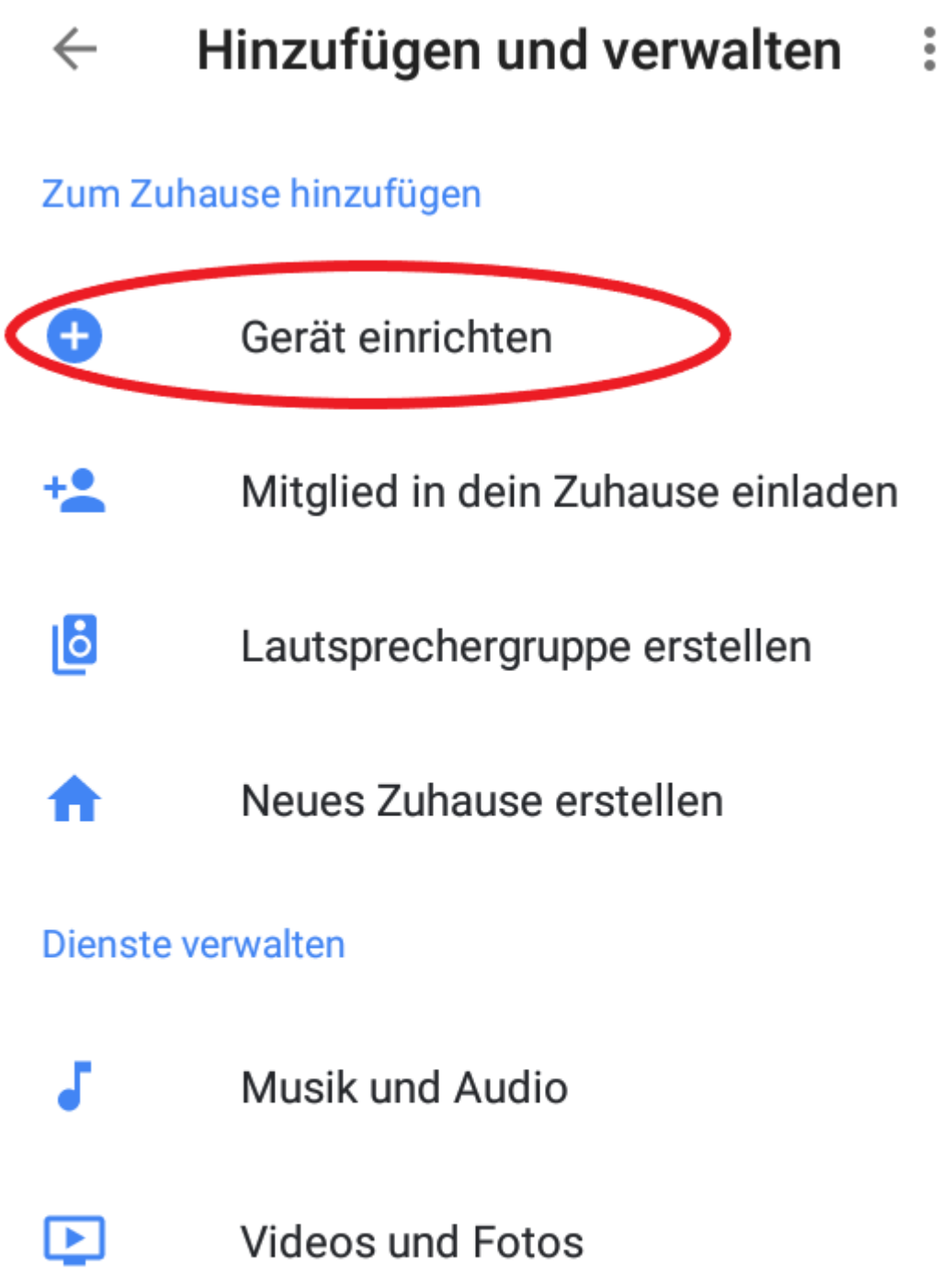

## IP-SYMCON MIT GOOGLE ASSISTANT VERKNÜPFEN

Abschließend muss IP-Symcon noch mit Google Assistant verknüpft werden, damit die

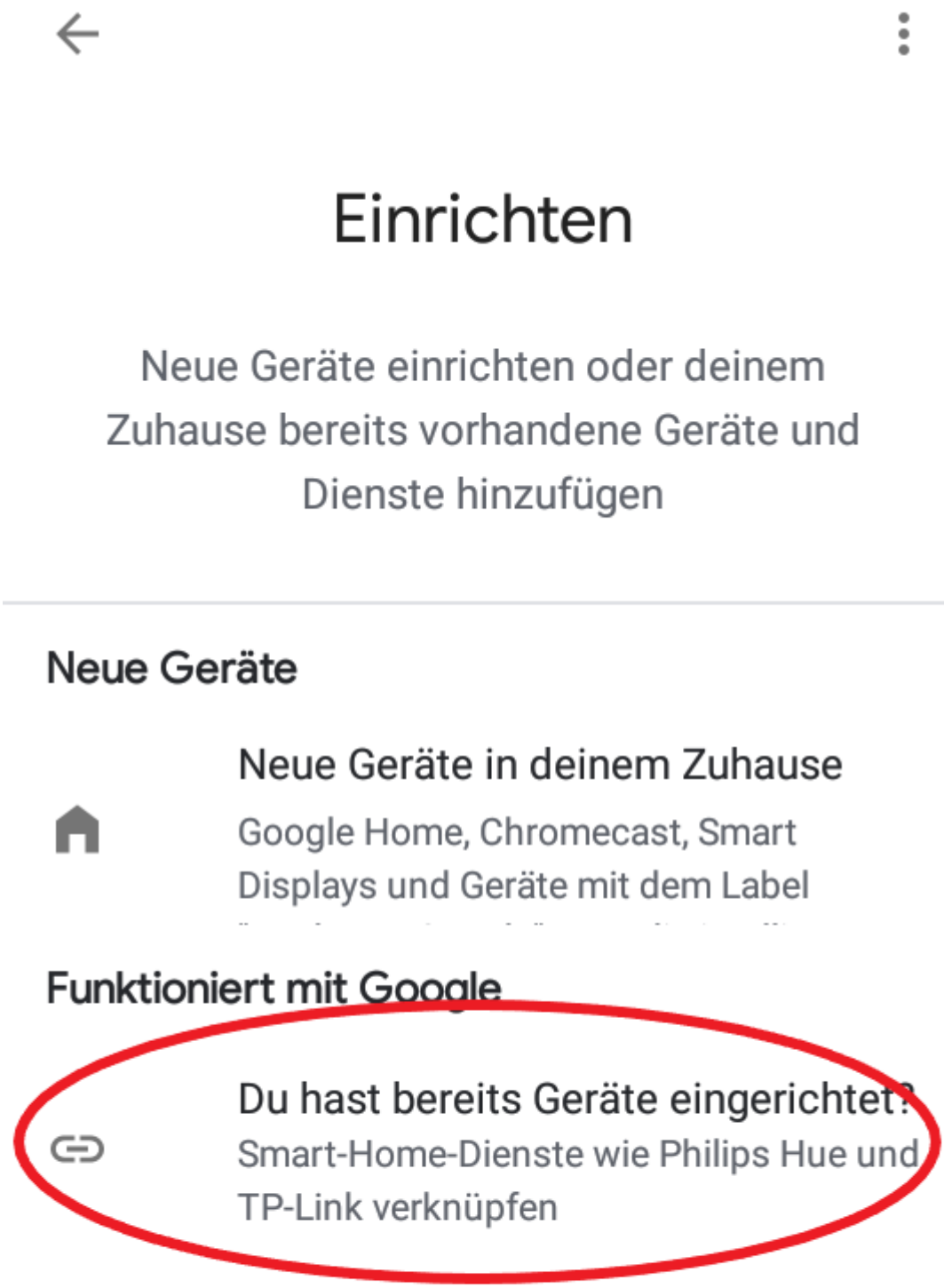
Google Assistant-Instanz die Anfragen von Google empfangen kann.
Die Installation erfolgt über die Google Home-App.
In der App den Tab zur SmartHome-Steuerung unten links auswählen und dort "Hinzufügen" betätigen.



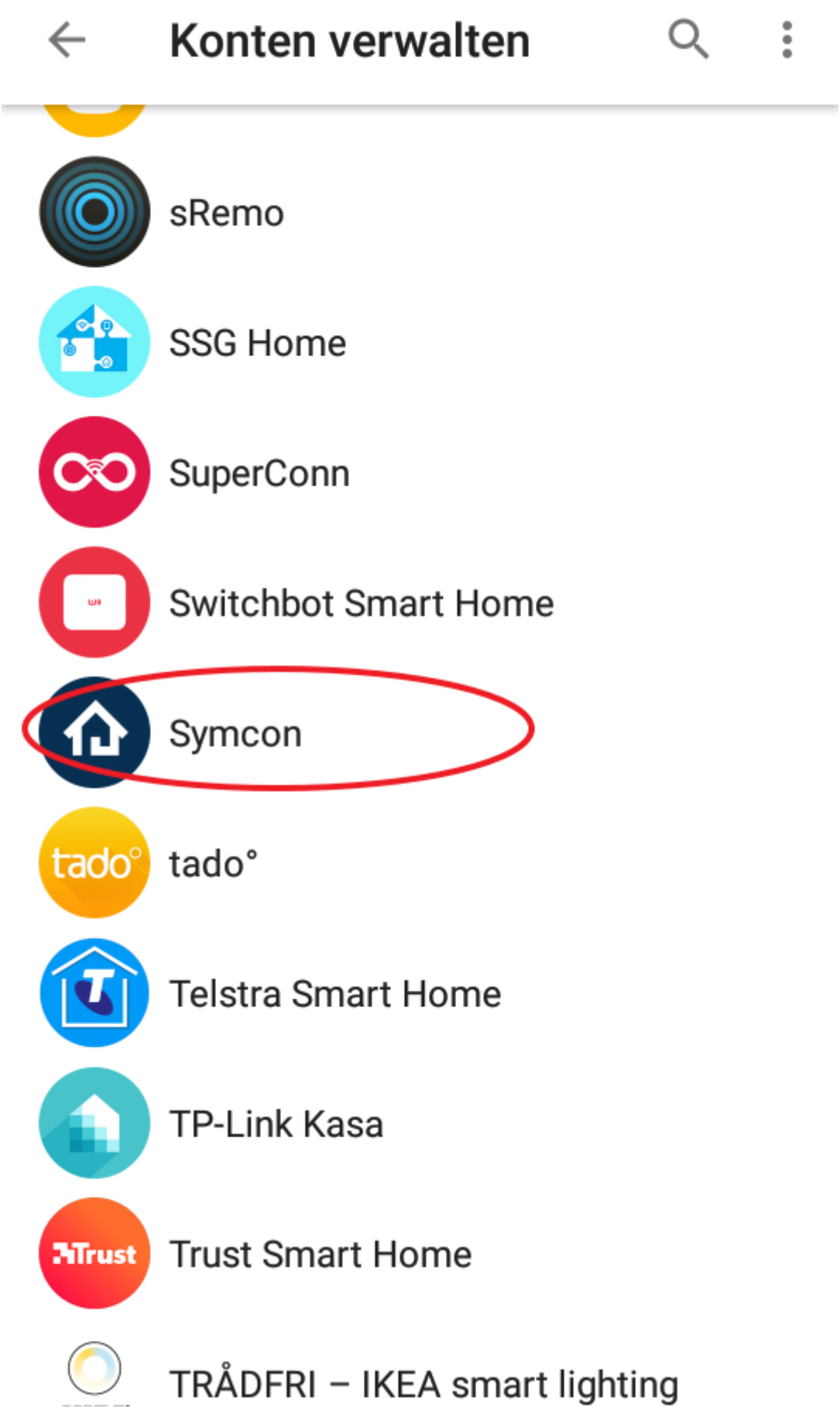
Bei der Auswahl "Gerät einrichten" auswählen.



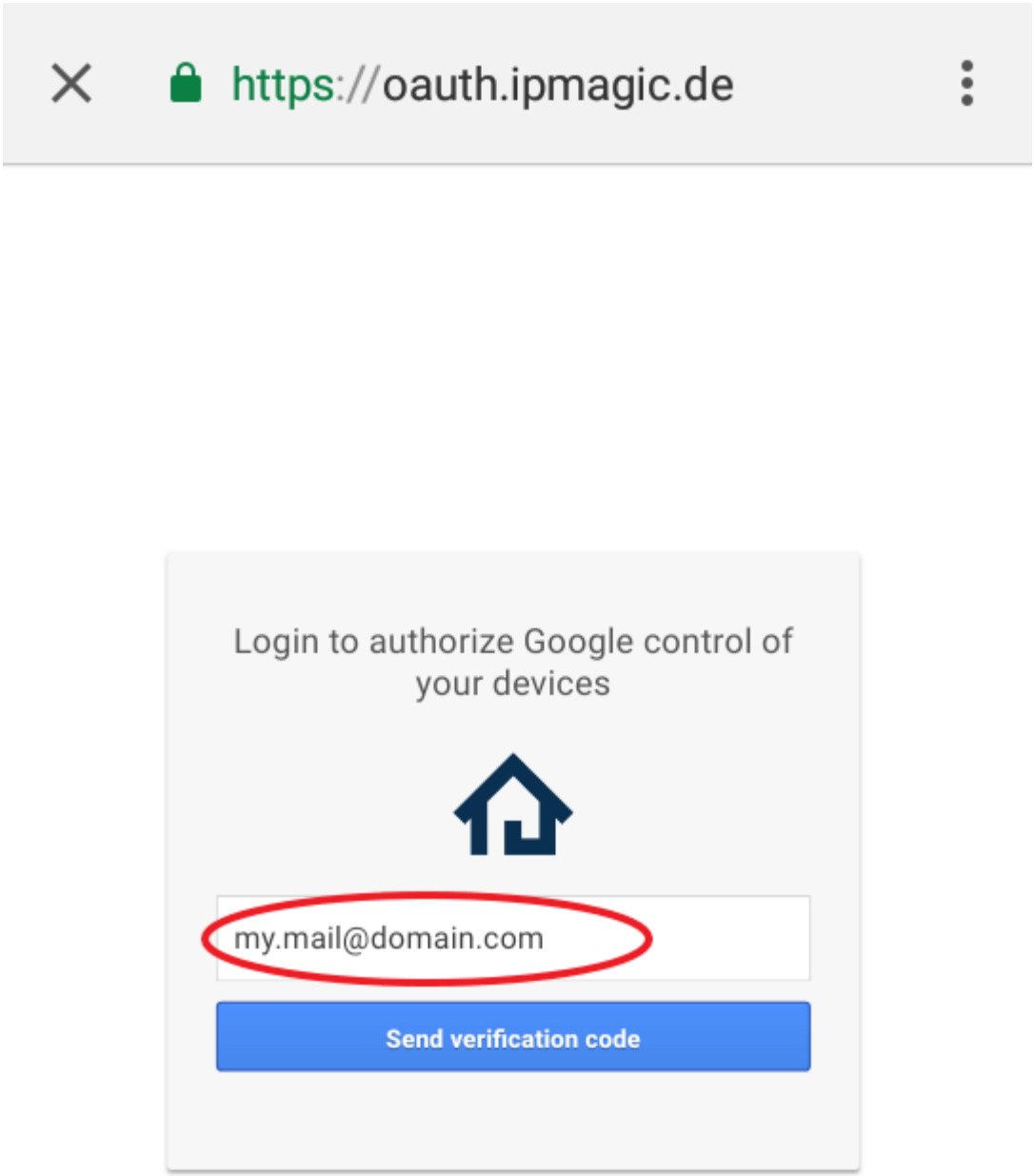

Im Einrichten-Dialog "Du hast bereits Geräte eingerichtet?" betätigen.



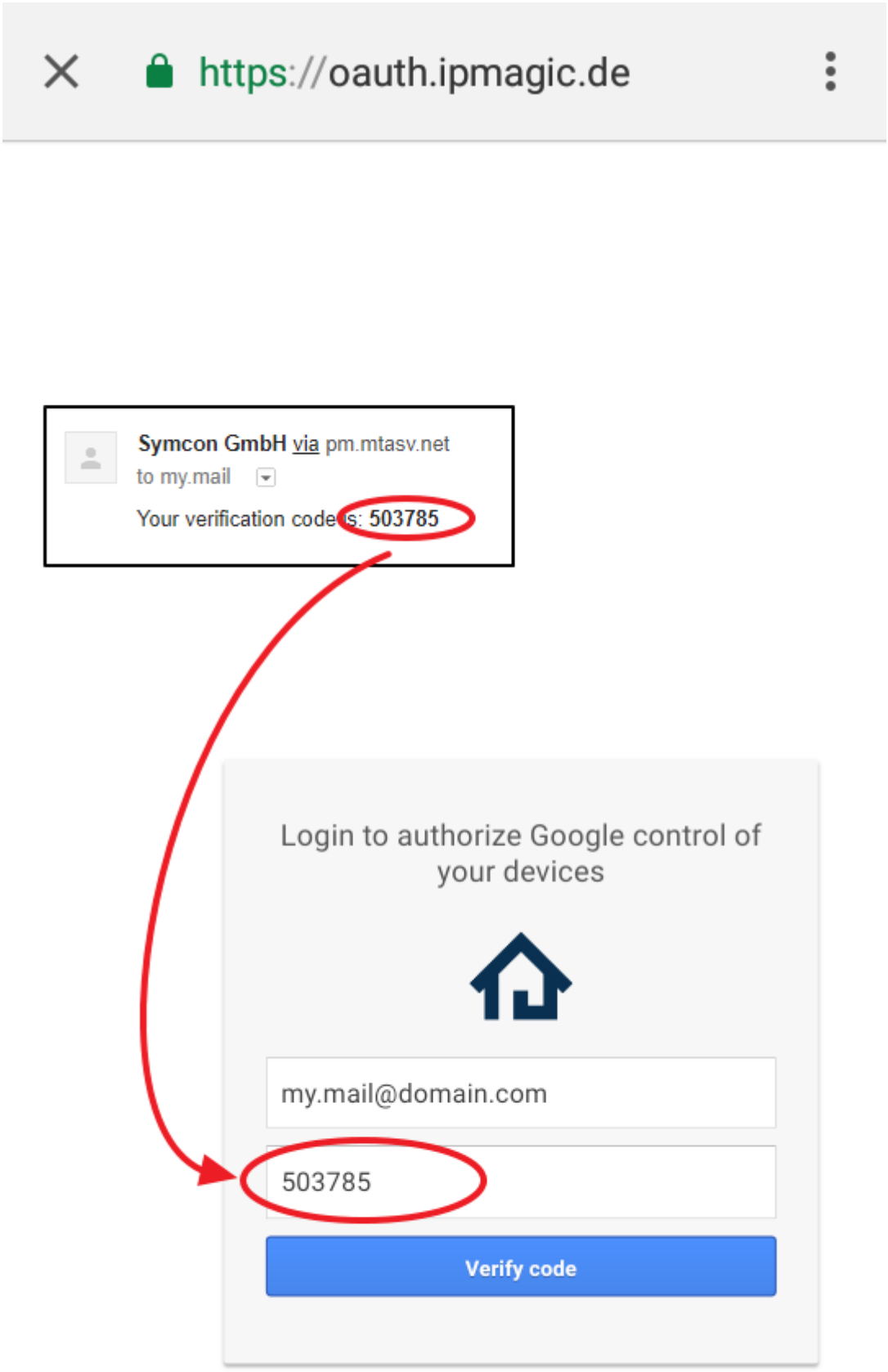

In der angezeigten Liste auf "Symcon" tippen.



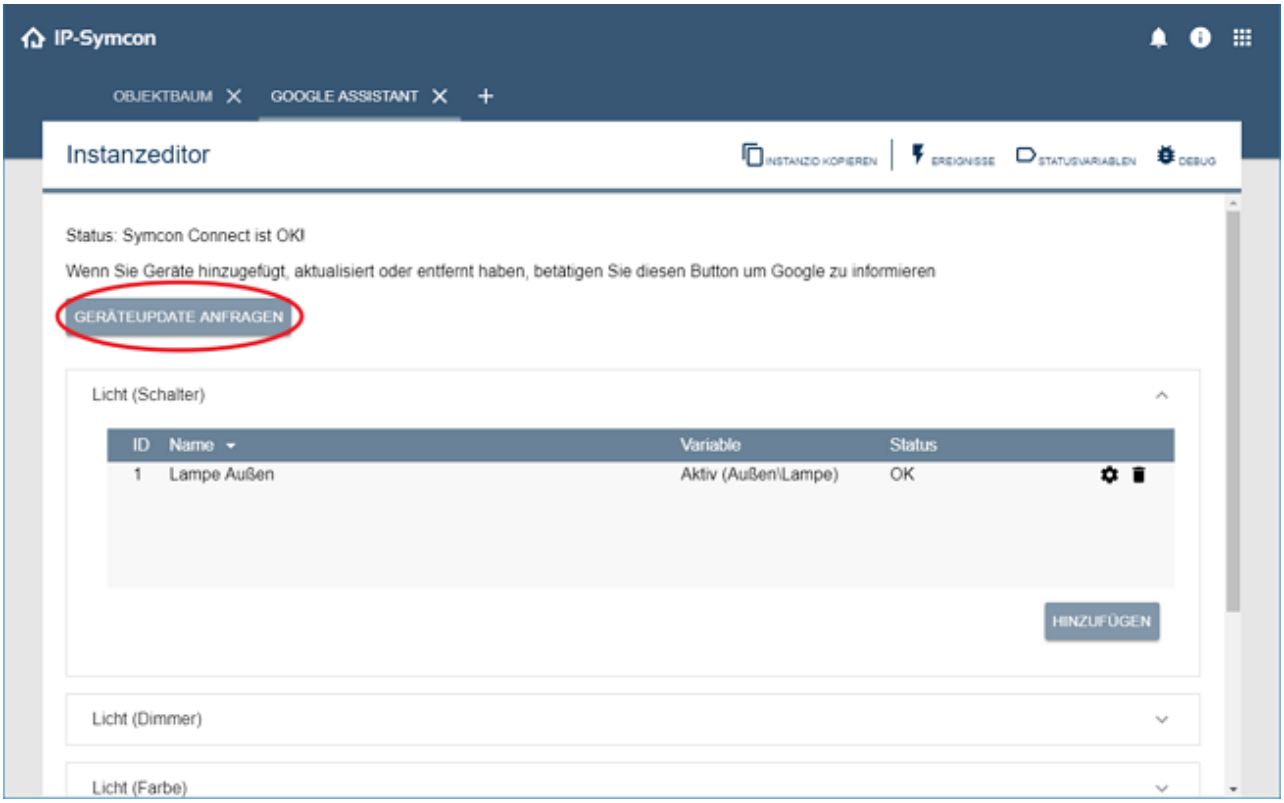

Im Popup die E-Mail-Adresse eingeben, auf welche die verwendete IP-Symcon-Lizenz registriert ist, und mit "Send verification code" bestätigen.




Es wird eine E-Mail mit einem Code an die angegebene E-Mail-Adresse geschickt. Den Code, der in dieser E-Mail steht, im Popup eintragen und mit "Verify code" bestätigen.




Nach erfolgreicher Bestätigung erscheinen die konfigurierten Geräte in der Google Home-App und können verwendet werden.


IP-Symcon
OBJEKTBAUM
GOOGLE ASSISTANT
Instanzeditor
INSTANZ KOPIEREN
EREIGNISSE
STATUSVARIABLEN
DEBUG
Status: Symcon Connect ist OK!
Wenn Sie Geräte hinzugefügt, aktualisiert oder entfernt haben, betätigen Sie diesen Button um Google zu informieren
GERÄTEUPDATE ANFRAGEN
Licht (Schalter)
ID
Name
Variable
Status
1
Lampe Außen
Aktiv (Außen\Lampe)
OK
HINZUFÜGEN
Licht (Dimmer)
Licht (Farbe)

# Expertenoptionen

Benötigt: IP-Symcon >= 5.0



Die Expertenoptionen müssen üblicherweise nicht angepasst werden. Gehen Sie mit diesen Optionen sorgsam um, damit das Google Assistant Modul wie vorgesehen funktioniert.

## OPTIONEN

| NAME | BESCHREIBUNG |
| --- | --- |
| Status emulieren | Ist Status emulieren aktiv, so wartet das Modul nicht auf die Aktualisierung der geschalteten Variablen und gibt stattdessen einen neuen Zustand auf Basis des geschalteten Wertes zurück. |
| Geräteupdate anfragen | Mit diesem Button kann ein Geräteupdate bei Google angefragt werden, wodurch die eingestellten Geräte erneut mit dem Google Assistant synchronisiert werden. Dies geschieht üblicherweise automatisch beim Übernehmen der Änderungen. |

# Generischer Schalter

Benötigt: IP-Symcon >= 5.0

## BESCHREIBUNG

Geräte vom Typ generischer Schalter können An und Aus geschaltet werden. Innerhalb von Google Assistant werden die Geräte als allgemeine Schalter interpretiert und können für verschiedenste Geräte verwendet werden.

## PARAMETER

| NAME | BESCHREIBUNG |
|---|---|
| Name | Name mit dem das Gerät über Google Assistant angesprochen werden kann |
| Schaltervariable | Eine schaltbare Variable vom Typ Boolean |

### MÖGLICHE AKTIONEN

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
|---|---|---|
| An oder Aus schalten | Schaltet die Variable auf True oder False | "Ok Google, schalte *<Name>* an." |

### GERÄTESUCHE

Eine Instanz muss Statusvariablen mit den passenden Profilen haben, um in der Gerätesuche erkannt zu werden.

| VARIABLE | MÖGLICHE PROFILE |
|---|---|
| Schaltervariable | ~Switch |

# Licht (Dimmer)

Benötigt: IP-Symcon >= 5.0

## BESCHREIBUNG

Geräte vom Typ Licht (Dimmer) beschreiben Lampen, welche gedimmt werden können.

## PARAMETER

| NAME | BESCHREIBUNG |
|---|---|
| Name | Name mit dem das Gerät über Google Assistant angesprochen werden kann |
| Variable | Eine schaltbare Variable vom Typ Integer oder Float, über welche das Licht gedimmt wird |

### MÖGLICHE AKTIONEN

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
|---|---|---|
| An oder Aus schalten | Schaltet die Variable auf 100% oder 0% | "Ok Google, schalte *<Name>* an." |
| Dimmen | Schaltet die Variable auf den angegebenen Wert | "Ok Google, dimme *<Name>* auf 40%." |

### GERÄTESUCHE

Eine Instanz muss Statusvariablen mit den passenden Profilen haben, um in der Gerätesuche erkannt zu werden.

| VARIABLE | MÖGLICHE PROFILE |
|---|---|
| Variable | ~Intensity.100, ~Intensity.255, ~Intensity.1 |

# Licht (Experte)

Benötigt: IP-Symcon >= 5.0

## BESCHREIBUNG

Geräte vom Typ Licht (Experte) beschreiben Lampen, deren Eigenschaften in verschiedenen Variablen unabhängig voneinander geschaltet werden können.

## PARAMETER

| NAME | BESCHREIBUNG |
|---|---|
| Name | Name mit dem das Gerät über Amazon Alexa angesprochen werden kann |
| Schaltervariable | Eine schaltbare Variable vom Typ Boolean, über welche die Lampe An oder Aus geschaltet wird |
| Helligkeitsvariable | Eine schaltbare Variable vom Typ Integer oder Float, über welche das Licht gedimmt wird (optional) |
| Farbvariable | Eine schaltbare Variable vom Typ Integer mit dem Profil ~HexColor, über welche die Farbe des Lichts geschaltet wird (optional) |



Werden optionale Variablen nicht gesetzt, so erkennt Google, dass die Lampe die dazugehörigen Eigenschaften nicht besitzt. Werden optionale Variablen erst später gesetzt, so muss ein Geräteupdate durchgeführt werden bevor die neu freigeschaltete Funktion genutzt werden kann.

### MÖGLICHE AKTIONEN

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
|---|---|---|
| An oder Aus schalten | Schaltet die Schaltervariable auf True oder False | "Ok Google, schalte *<Name>* an." |
| Dimmen | Schaltet die Helligkeitsvariable auf den angegebenen Wert | "Ok Google, dimme *<Name>* auf 40%." |
| Farbe schalten | Schaltet die Farbvariable auf die angegebene Farbe | "Ok Google, schalte *<Name>* auf rot." |

### GERÄTESUCHE

Eine Instanz muss Statusvariablen mit den passenden Profilen haben, um in der Gerätesuche erkannt zu werden. Um als Expertenlicht erkannt zu werden, muss mindestens die Schaltervariable und eine weitere Variable erkannt werden.

| VARIABLE | MÖGLICHE PROFILE |
| --- | --- |
| Schaltervariable | ~Switch |
| Helligkeitsvariable | ~Intensity.100, ~Intensity.255, ~Intensity.1 |
| Farbvariable | ~HexColor |

# Licht (Farbe)

Benötigt: IP-Symcon >= 5.0

## BESCHREIBUNG

Geräte vom Typ Licht (Farbe) beschreiben Lampen, welche beliebige Farben annehmen können.

## PARAMETER

| NAME | BESCHREIBUNG |
|---|---|
| Name | Name mit dem das Gerät über Google Assistant angesprochen werden kann |
| Variable | Eine schaltbare Variable vom Typ Integer mit dem Profil ~HexColor, über welche die Farbe des Lichts geschaltet wird |

### MÖGLICHE AKTIONEN

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
|---|---|---|
| An oder Aus schalten | Schaltet die Variable auf Weiß oder Schwarz | "Ok Google, schalte *<Name>* an." |
| Dimmen | Schaltet Helligkeit der aktuellen Farbe der Variable auf den angegebenen Wert | "Ok Google, dimme *<Name>* auf 40%." |
| Farbe schalten | Schaltet die Variable auf die angegebene Farbe | "Ok Google, schalte *<Name>* auf rot." |

### GERÄTESUCHE

Eine Instanz muss Statusvariablen mit den passenden Profilen haben, um in der Gerätesuche erkannt zu werden.

| VARIABLE | MÖGLICHE PROFILE |
|---|---|
| Variable | ~HexColor |

# Licht (Schalter)

Benötigt: IP-Symcon >= 5.0

## BESCHREIBUNG

Geräte vom Typ Licht (Schalter) beschreiben Lampen, welche an- und ausgeschaltet werden können.

## PARAMETER

| NAME | BESCHREIBUNG |
|---|---|
| Name | Name mit dem das Gerät über Google Assistant angesprochen werden kann |
| Variable | Eine schaltbare Variable vom Typ Boolean, über welche das Licht An oder Aus geschaltet wird |

### MÖGLICHE AKTIONEN

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
|---|---|---|
| An oder Aus schalten | Schaltet die Variable auf True oder False | "Ok Google, schalte *<Name>* an." |

### GERÄTESUCHE

Eine Instanz muss Statusvariablen mit den passenden Profilen haben, um in der Gerätesuche erkannt zu werden.

| VARIABLE | MÖGLICHE PROFILE |
|---|---|
| Schaltervariable | ~Switch |

# Rollladen

Benötigt: IP-Symcon >= 5.0

## BESCHREIBUNG

Geräte vom Typ Rollladen können geöffnet und geschlossen werden. Unterstützt das Gerät eine prozentuale Öffnung, so ist diese ebenfalls steuerbar.

## PARAMETER

| NAME | BESCHREIBUNG |
|---|---|
| Name | Name mit dem das Gerät über Amazon Alexa angesprochen werden kann |
| Rollladenvariable | Eine schaltbare Variable vom Typ Integer oder Float, welche das Profil ~ShutterMoveStop oder ~ShutterMoveStep unterstützt. Alternativ kann eine Variable verwendet werden, welcher durch ihr Profil ein Minimal- und Maximalwert zugewiesen wurde |

### MÖGLICHE AKTIONEN

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
|---|---|---|
| Öffnen | Schaltet die Variable bei einem ~ShutterMove-Profil auf "Öffnen". Ansonsten wird die Variable auf den Minimalwert des Profils geschaltet. | "Ok Google, öffne *<Name>*" |
| Schließen | Schaltet die Variable bei einem ~ShutterMove-Profil auf "Schliessen". Ansonsten wird die Variable auf den Maximalwert des Profils geschaltet. | "Ok Google, schließe *<Name>*" |
| Prozentual öffnen | Schaltet die Variable bei einem ~ShutterMove-Profil auf "Öffnen" falls der Prozentsatz größer als 50% ist, sonst auf "Schliessen". Ansonsten wird die Variable entsprechend des Prozentsatzes geschaltet. Es ist zu beachten, dass bei einer vollen Öffnung die Variable auf 0% geschaltet wird und dies bei anderen Prozentwerten auch entsprechend angewandt wird. Wird also beispielsweise zu 30% geöffnet, so wird die Variable auf 70% geschaltet. | "Ok Google, öffne *<Name>* zu 30%" |

### GERÄTESUCHE

Eine Instanz muss Statusvariablen mit den passenden Profilen haben, um in der Gerätesuche erkannt zu werden.

| VARIABLE | MÖGLICHE PROFILE |
| --- | --- |
| Rollladenvariable | ~ShutterMoveStop, ~ShutterMoveStep, ~Intensity.100, ~Intensity.255, ~Intensity.1 |

# Szenen

Benötigt: IP-Symcon >= 5.0

## BESCHREIBUNG

Geräte vom Typ Szene können aktiviert werden und beliebige Aktionen auslösen.

## PARAMETER

| NAME | BESCHREIBUNG |
|---|---|
| Name | Name mit dem das Gerät über Google Assistant angesprochen werden kann |
| Aktion | Eine Aktion, welche beim Aktivieren der Szene ausgeführt wird |

### MÖGLICHE AKTIONEN

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
|---|---|---|
| Szene aktivieren | Führt die Aktion aus | "Ok Google, aktiviere *<Name>*." |

### GERÄTESUCHE

Dieser Gerätetyp wird nicht von der Gerätesuche unterstützt.

# Szenen (deaktivierbar)

Benötigt: IP-Symcon >= 5.0

## BESCHREIBUNG

Geräte vom Typ Szene können aktiviert und deaktiviert werden und dabei beliebige Aktionen auslösen.

## PARAMETER

| NAME | BESCHREIBUNG |
|---|---|
| Name | Name mit dem das Gerät über Google Assistant angesprochen werden kann |
| Aktion beim Aktivieren | Eine Aktion, welche beim Aktivieren der Szene ausgeführt wird |
| Aktion beim Deaktivieren | Eine Aktion, welche beim Deaktivieren der Szene ausgeführt wird |

### MÖGLICHE AKTIONEN

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
|---|---|---|
| Szene aktivieren | Führt die Aktion zum Aktivieren aus | "Ok Google, aktiviere *<Name>*." |
| Szene deaktivieren | Führt die Aktion zum Deaktivieren aus | "Ok Google, deaktiviere *<Name>*." |

### GERÄTESUCHE

Dieser Gerätetyp wird nicht von der Gerätesuche unterstützt.

# Thermostat

Benötigt: IP-Symcon >= 5.0

## BESCHREIBUNG

Geräte vom Typ Thermostat können Temperaturen einstellen und Umgebungstemperaturen lesen.

## PARAMETER

| NAME | BESCHREIBUNG |
|---|---|
| Name | Name mit dem das Gerät über Amazon Alexa angesprochen werden kann |
| Sollwert | Eine schaltbare Variable vom Typ Float, durch welche die Solltemperatur in °C eingestellt wird |
| Umgebungstemperatur | Eine Variable vom Typ Float, welche die Umgebungstemperatur in °C beinhaltet |

### MÖGLICHE AKTIONEN

| AKTION | BESCHREIBUNG | MÖGLICHER SATZ ZUM AKTIVIEREN |
|---|---|---|
| Temperatur einstellen | Schalte die Variable auf die angegebene Temperatur. | "Ok Google, setze *<Name>* auf 35 °C." |

### GERÄTESUCHE

Dieser Gerätetyp wird nicht von der Gerätesuche unterstützt.

# Benachrichtigungen

Benachrichtigungen bezeichnen die verschiedenen Wege, wie Symcon sich bei einem Nutzer melden und diesen informieren kann. Dies bezeichnet den Umfang von Audiovisuell bis hin zu monatlichen Reports via E-Mail.

## ALARMIERUNG

Löst einen Alarm aus, wenn eine der Sensorenvariablen aktiv wird.

| | |
|---|---|
| ARM_GetLastAlertID | Gibt die ID der Variable zurück, die als letztes einen Alarm ausgelöst hat. |
| ARM_SetActive | Schaltet das Alarmierungsmodul mit der InstanzID __InstanzID__ auf den Wert __Value__ (true = An; false = Aus). |
| ARM_SetAlert | Schaltet den Alarm mit der InstanzID __InstanzID__ auf den Wert __Value__ (true = An; false = Aus). |

## BENACHRICHTIGUNG

Ermöglicht eine mehrstufige Benachrichtigung, wobei die Stufe sich nach einer definierten Zeit erhöht und bei einer Quittierung zurückgesetzt wird.

| | |
|---|---|
| BN_IncreaseLevel | Erhöht die Stufe und führt die Aktion aus |
| BN_Reset | Setzt die Benachrichtigungsstufe zurück und deaktiviert die Kette |
| BN_SetNotifyLevel | Setzt die Benachichtigung auf eine vorgegebene Stufe |

## DURCHSAGE

Das Durchsage Modul bietet die Möglichkeit von AWS Polly erzeugte Audiodaten über Sonos oder den Media Player wiederzugeben.

| | |
|---|---|
| DS_Play | Spielt den Text als Durchsage ab. |

## DYNAMISCHE E-MAIL

Versendet eine E-Mail mit einem dynamischen Text.

| | |
|---|---|
| DM_SendMail | Nutzt den Betreff und Text um eine E-Mail zu versenden. |

## FERTIGMELDER

Meldet nach einer Totzeit, nachdem ein Gerät einen gewissen Verbrauch unterschritten hat.

## IMAP

Funktionen zum IMAP-Modul zur Abfrage von E-Mails

| | |
|---|---|
| IMAP_DeleteMail | löscht eine Mail mit einer bestimmten E-Mail(UID) |
| IMAP_GetCachedMails | gibt ein Array zu den gecachten E-Mails zurück |
| IMAP_GetMailEx | liefert ein Array mit Informationen zu einer bestimmten E-Mail(UID) wieder |

## MEDIAPLAYER

Funktionen zur Ansteuerung von Mediaplayer-Instanzen

| | |
|---|---|
| WAC_AddFile | fügt Musikdatei zur Playlist hinzu |
| WAC_ClearPlaylist | löscht die Playlist |
| WAC_GetPlaylistLength | löscht die Playlist |
| WAC_GetPlaylistPosition | löscht die Playlist |
| WAC_Next | spielt den nächsten Titel der Playlist ab |
| WAC_Pause | pausiert die Wiedergabe |
| WAC_Play | spielt die Playlist ab |
| WAC_PlayFile | spielt eine Musikdatei direkt ab |
| WAC_Prev | spielt den vorherigen Titel der Playlist ab |
| WAC_SetPlaylistPosition | löscht die Playlist |
| WAC_SetPosition | setzt den Positionsmarker auf eine bestimmte Sekunde |
| WAC_SetRepeat | schaltet die Wiederholung der Playlist ein/aus |
| WAC_SetShuffle | schaltet die Zufallswiedergabe ein/aus |
| WAC_SetVolume | setzt die Lautstärke |
| WAC_Stop | stoppt die Wiedergabe |

## POP3

Funktionen zum POP3-Modul zur Abfrage von E-Mails

| | |
|---|---|
| POP3_DeleteMail | löscht eine Mail mit einer bestimmten E-Mail(UID) |
| POP3_GetCachedMails | gibt ein Array zu den gecachten E-Mails zurück |
| POP3_GetMailEx | liefert ein Array mit Informationen zu einer bestimmten E-Mail(UID) wieder |

## POPUP MODUL

Stilistisches Modul zur Darstellung

## SMS

Funktionen zum Versenden von SMS via SMS-Modul

| | |
|---|---|
| SMS_Send | versendet eine E-Mail an die Standardadresse |

## SMTP

Funktionen zum SMTP-Modul zur Versendung von E-Mails

| | |
|---|---|
| SMTP_SendMail | versendet eine E-Mail an die Standardadresse |
| SMTP_SendMailAttachment | versendet eine E-Mail mit Anhang an die Standardadresse |
| SMTP_SendMailAttachmentEx | versendet eine E-Mail mit Anhang an eine beliebige Adresse |
| SMTP_SendMailEx | versendet eine E-Mail an eine beliebige Adresse |
| SMTP_SendMailMedia | versendet eine E-Mail mit Anhang eines Medienobjekts vom Typ "Bild/Ton" an die Standardadresse |
| SMTP_SendMailMediaEx | versendet eine E-Mail mit Anhang eines Medienobjekts vom Typ "Bild/Ton" an eine beliebige Adresse |

## SPOTIFY

Steuert die Wiedergabe von Spotify

| | |
|---|---|
| SPO_MakeAPIRequest | Stellt eine Anfrage an die Spotify Web-API mit den angegebenen Parametern |
| SPO_NextTrack | Existiert eine aktuelle Wiedergabe, so wird der nächste Song abgespielt. |
| SPO_Pause | Existiert eine aktuelle Wiedergabe, so wird diese pausiert. |
| SPO_Play | Existiert eine aktuelle, möglicherweise pausierte, Wiedergabe, so wird diese fortgesetzt. |
| SPO_PlayURI | Spielt die Spotify-Resource mit der URI __URI__ auf dem aktuell ausgewähltem Gerät ab. |
| SPO_PreviousTrack | Existiert eine aktuelle Wiedergabe, so wird der vorherige Song abgespielt. |
| SPO_ResetToken | Setzt die OAuth-Token und somit die Verknüpfung mit dem Spotify-Konto zurück. |
| SPO_SetRepeat | Setzt die Wiederholung auf den Wert Repeat. |
| SPO_SetShuffle | Ist Shuffle true, so wird die zufällige Wiedergabe aktiviert, ansonsten deaktiviert. |

## STÖRUNGSMANAGER

Zeigt an, wenn ausgewählte Variablen aktiv sind und kann diese je nach Einstellung quittieren.

## SYMCONREPORT

Erstellt je nach Module ein Report als CSV oder PDF

| | |
|---|---|
| MailReport | Erstellt einen CSV-Report von Archivwerten und sendet diesen per E-Mail(SMTP) |
| MR_SendInfo | Verschickt eine Info-Mail über das letzte abgeschlossene Zeitintervall |
| MR_SetActive | Aktiviert oder deaktiviert das Modul |
| PDFReport (Energie) | Erstellt ein Report von Archivwerten zweier Variablen einer Zeitspanne als PDF |
| RAC_GenerateEnergyReport | Generiert ein PDF mit den im Modul mit der InstanzID |
| PDFReport (Multi) | Erstellt einen Report von Archivwerten mehrerer Variablen einer Zeitspanne als PDF |
| RAC_GenerateReport | Generiert ein PDF mit den im Modul mit der InstanzID angegebenen Werten |
| PDFReport (Multi Energie) | |
| PDFReport (Single) | Erstellt einen Report von Archivwerten einer Variable mit Min/Max/Avg einer Zeitspanne als PDF |
| RAC_GenerateReport | Generiert ein PDF mit den im Modul mit der InstanzID angegebenen Werten |

## TELEFONANSAGE

Das Telefonansage-Modul ermöglicht die komfortable Verknüpfung einer VoIP-Instanz und einer Text-to-Speech-Instanz (AWS Polly) um eine Telefonnummer anzurufen und bei Annahme des Anrufs einen Text auszugeben. Darüber hinaus, kann das Modul auf DTMF-Töne reagieren und weitere Texte ausgeben.

| | |
|---|---|
| TA_StartCall | Startet einen Anruf auf Basis der aktuellen Einstellung. |
| TA_StartCallEx | Startet einen Anruf an eine Telefonnummer und gibt den Text aus |

## TELEFONKETTE

Das Telefonkette-Modul ermöglicht es eine Liste von Telefonnummern nacheinander anzurufen.

## TELEGRAMBOT

Erlaubt das Versenden von Nachrichten an angegebene Personen und Ausführen von Aktionen

## TEXT TO SPEECH

Funktionen zum Ansteuern eines TTS-Moduls

| | |
|---|---|
| TTS_GenerateFile | generiert eine WAV Datei mit dem gewünschten Text |
| TTS_Speak | spricht einen beliebigen Text auf der gewählten Soundkarte |

## TTSAWSPOLLY

Kann Sounddaten/-dateien in verschiedenen Formaten erzeugen

| | |
|---|---|
| TTSAWSPOLLY_GenerateData | Fragt über AWS den Text an und liefert die Sprachdaten in der Rückgabe zurück. |
| TTSAWSPOLLY_GenerateFile | Fragt über AWS den Text an und liefert den Dateinamen zu den Sprachdaten zurück. |

## VERBRAUCHSALARM

Überwacht eine Zähler-Variable auf unnatürlichen Verbrauch.

| | |
|---|---|
| VBA_CheckAlert | Kontrolliert ob ein Grenzwert überschritten wird |

## WASSER-ALARM

Überwacht eine Zähler-Variable auf unnatürlichen Wasserverbrauch (Leck/Bruch).

| | |
|---|---|
| WAA_CheckAlert | Kontrolliert ob die Grenzwerte überschritten wurden |

## WATCHDOG

Überwacht Variablen und löst einen Alarm aus, wenn diese zu lange nicht aktualisiert oder geändert wurden.

| | |
|---|---|
| WD_GetAlertTargets | gibt ein Array mit den überfälligen Variablen zurück |
| WD_SetActive | De-/aktiviert die Instanz |

# Alarmierung

Benötigt: IP-Symcon >= 5.3
Das Modul löst einen Alarm aus, wenn eine der Sensorenvariablen aktiv wird. Dabei werden Zielvariablen bei einem Alarm auf den maximalen Wert bzw. An (True) gesetzt. Ein einmal geschalteter Alarm wird nicht automatisch deaktiviert, dieser muss manuell zurückgesetzt werden.

## FUNKTIONSUMFANG

Konfiguration von Sensor- und Zielvariablen via Listenauswahl, welche den Alarm auslösen oder bei einem Alarm geschaltet werden
Einstellbare Einschaltverzögerung
Ein-/Ausschaltbarkeit via Visualisierungsbutton oder Skript-Funktion
Konvertierungsfunktion für alte Versionen des Alarmierungsmoduls

## SOFTWARE-INSTALLATION

Über den Module Store das Modul 'Alarmierung' installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Alarmierung'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen in der Dokumentation der Instanzen.

### KONFIGURATIONSSEITE

| NAME | BESCHREIBUNG |
|---|---|
| Button "Konvertierung" | (Wird nur angezeigt, wenn die Listen leer und alte Links vorhanden sind) Wenn eine alte Version des Moduls erkannt wurde, können die alten Links in die neuen Listen via Knopfdruck eingepflegt werden. Ist dies Erfolgreich erscheint ein Meldungsfenster. |
| Sensorvariablen | Diese Liste beinhaltet die Variablen, welche bei Aktualisierung auf einen aktiven Wert einen Alarm auslösen. Als aktiv gelten hierbei Variablen mit dem Wert true oder einen Wert ungleich 0. Sollte die Variable ein .Reversed Profil haben gelten die Werte false und 0 als aktiv. |
| Zielvariablen | Diese Liste beinhaltet die Variablen, welche bei Alarm geschaltet werden. Diese müssen eine Standardaktion oder Aktionsskript beinhalten. |
| Einschaltverzögerung | Wenn größer 0 wird die Alarmierung erst nach der eingestellten Zeit aktiv |

## STATUSVARIABLEN UND PROFILE

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Aktiv | Boolean | De-/Aktiviert die Alarmierung. Wird die Alarmierung deaktiviert, so wird auch der ggf. vorhandene Alarm deaktiviert sowie alle ausgewählten Zielvariablen. |
| Zeit bis Aktivierung | String | Zeigt während des Aktivierungs-Vorgangs die noch verbleibende Zeit an. |
| Alarm | Boolean | De-/Aktiviert den Alarm und alle ausgewählten Zielvariablen. |
| Aktive Sensoren | String | Listet alle aktiven Sensoren auf und wird ausgeblenet, wenn keine Aktiv sind. |

## VISUALISIERUNG

Über die Visualisierung kann die Alarmierung de-/aktiviert werden.
Es wird zusätzlich angezeigt, ob ein Alarm vorliegt oder nicht. Es wird eine Liste aller noch aktiven Sensoren angezeigt. Der Alarm kann auch manuell de-/aktiviert werden.

# ARM_GetLastAlertID

Benötigt: IP-Symcon >= 5.3

```
int ARM_GetLastAlertID (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

ID der Variable, die als letztes einen Alarm ausgelöst hat.

## BESCHREIBUNG

Gibt die ID der Variable zurück, die als letztes einen Alarm ausgelöst hat.

## BEISPIEL

```
ARM_GetLastAlertID(12345); //Variablen ID abfragen
```

# ARM_SetActive

Benötigt: IP-Symcon >= 5.3

```
boolean ARM_SetActive (int $InstanzID, boolean $Value)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Value | Wert auf den das Alarmierungsmodul geschaltet werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schaltet das Alarmierungsmodul mit der InstanzID **InstanzID** auf den Wert **Value** (true = An; false = Aus).

## BEISPIEL

```
ARM_SetActive(12345, true); //Alarmierungsmodul aktivieren
```

# ARM_SetAlert

Benötigt: IP-Symcon >= 5.3

```
boolean ARM_SetAlert (int $InstanzID, boolean $Value)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Value | Wert auf den der Alarm geschaltet werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schaltet den Alarm mit der InstanzID **InstanzID** auf den Wert **Value** (true = An; false = Aus).

## BEISPIEL

```
ARM_SetAlert(12345, false); //Alarm ausschalten
```

# Benachrichtigung

Benötigt: IP-Symcon >= 5.1
Dieses Modul ermöglicht eine mehrstufige Benachrichtigung, wobei die Stufe sich nach einer definierten Zeit erhöht und bei einer Quittierung zurückgesetzt wird. Beim Erreichen einer neuen Stufe können verschiedene Aktionen ausgeführt werden.

## FUNKTIONSUMFANG

Starten einer mehrstufigen Benachrichtigungskette durch eine Variable
Auf jeder Stufe können individuell Aktionen festgesetzt werden:
- Skripte ausführen
- Push-Nachrichten, E-Mails oder SMS verschicken
- Telefonanruf mit Ansage durchfügen (sofern "Telefonansage" Modul installiert)
- Durchsage über Lautsprecher durchführen (sofern "Durchsage" Modul installiert)
- Nachricht über Telegram versenden (sofern "TelegramBot" Modul installiert)
- Nach bestimmter Zeit auf nächste Stufe erhöhen

Quittierung über beigefügtes Skript oder Push-Nachricht beendet Benachrichtigungen
Einzelne Stufen können bei Bedarf deaktiviert werden

## SOFTWARE-INSTALLATION

Über den Modul Store das Modul Benachrichtigung installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Benachrichtigung'-Modul mithilfe des Schnellfilters gefunden werden.
- Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

Bei 'Auslöser' eine Variable auswählen
- Nimmt diese Variable einen Wert an, der nicht neutral ist (false bei Boolean, 0 bei Integer und Float, '' bei String), wird die Benachrichtigungskette auf Stufe 1 gestartet

Der Liste Benachrichtigungsstufen beliebig viele Stufen hinzufügen und konfigurieren
- Dauer definiert die Zeit bis die nächste Stufe aktiviert wird
- Aktionen werden beim Erreichen einer Stufe ausgeführt
- Aktiv definiert, ob die Stufe ausgeführt werden soll. Nicht aktive Stufen werden übersprungen
- 'Status' beinhaltet Fehlermeldungen, falls bei der Konfiguration der Stufe etwas nicht korrekt ist, ansonsten "OK"

### AKTIONEN EINRICHTEN

Jede Aktion besitzt Parameter für den Aktionstyp, ein Empfängerobjekt, eine Empfängeradresse, einen Titel, eine Nachricht und eine Nachrichtvariable. Die "Aktion" definiert was aufgerufen werden soll. (z.B. Ein Skript oder eine Email versendet)
Dabei werden weitere Informationen wie "Titel" und "Nachricht" an das eingestellte

"Empfängerobjekt" geschickt. Der Inhalt der Nachricht wird definiert durch den Text im Konfigurationsfeld "Nachricht". Dieser wird mit dem Text in der "Nachrichtenvariable" verkettet.
Für generische Nachrichten steht der Inhalt der Nachrichtenvariable als '{variable}' zur Verfügung. Ebenso können durch '\n' Zeilenumbrüche eingefügt werden. Die Empfängeradresse hat je nach Aktion eine andere Funktionalität.

### SKRIPT

Das als Empfängerobjekt ausgewählte Skript wird ausgeführt. Während dieses Aufrufs können folgende Systemvariablen verwendet werden:

| SYSTEMVARIABLE | BESCHREIBUNG |
|---|---|
| $_IPS['RECIPIENT'] | Der Inhalt des Tabellenfeldes Empfängeradresse |
| $_IPS['TITLE'] | Der Inhalt des Tabellenfeldes Titel |
| $_IPS['MESSAGE'] | Der Inhalt des Tabellenfeldes Nachricht |
| $_IPS['MESSAGE_VARIABLE'] | Die ID der Nachrichtenvariablen |

### PUSH

Eine Pushnachricht wird an alle Geräte des gewählten Webfronts geschickt. Diese Nachricht verlinkt das Quittierungsskript. Durch Tippen auf die Pushnachricht kann also die Benachrichtigung quittiert werden. Die Empfängeradresse hat bei diesem Aktionstyp keinen Effekt. Die Nachricht hat eine Maximallänge von 256 Zeichen.

### E-MAIL (SMTP)

Eine E-Mail wird über die gewählte SMTP-Instanz verschickt. Ist eine Empfängeradresse angegeben, so wird die E-Mail an diese Adresse verschickt. Ist keine Empfängeradresse angegeben, so wird die E-Mail an den angegebenen Empfänger der SMTP-Instanz geschickt. Wenn die Option 'Erweiterte Antwort' aktiviert ist kann mit dem Stichwort '{actions}' ein Block mit Links eingefügt werden, über welche die verfügbaren Aktionen ausgeführt werden können.

### SMS

Eine SMS wird über die gewählte SMS-Instanz an die in der Empfängeradresse angegebene Telefonnummer geschickt. Wenn die Option 'Erweiterte Antwort' aktiviert ist kann mit dem Stichwort '{actions}' ein Link eingefügt werden, über den die verfügbaren Aktionen ausgeführt werden können. Ist eine Nachricht länger als 160 Zeichen (Begrenzung durch SMS), wird diese auf bis zu 2 weitere SMS aufgeteilt (maximal 459 Zeichen).

### TELEFONANSAGE (NUR VERFÜGBAR, WENN DAS MODUL TELEFONANSAGE INSTALLIERT IST)

Die in der Empfängeradresse angegebene Telefonnummer wird angerufen und der Titel sowie die Nachricht vorgelesen. Wenn die Option 'Erweiterte Antwort' aktiviert ist kann

mit den Tasten 0-9 die Dazugehörige Aktion ausgeführt werden. Wenn gewünscht können mit dem Stichwort '{actions}' die verfügbaren Aktionen in die Nachricht eingebunden werden.

### DURCHSAGE (NUR VERFÜGBAR, WENN DAS MODUL DURCHSAGE INSTALLIERT IST)

Der Titel und die Nachricht werden mithilfe der ausgewählten Instanz vorgelesen. Die Empfängeradresse hat bei diesem Aktionstyp keinen Effekt.

### TELEGRAM (NUR VERFÜGBAR, WENN DAS MODUL TELEGRAMBOT INSTALLIERT IST)

Der Titel und die Nachricht werden mithilfe der ausgewählten Instanz an den Empfänger gesendet. Die Empfängeradresse kann entweder der Name oder die UserID vom Telegram Empfänger sein. Sofern die Empfängeradresse leer gelassen wird, werden alle Empfänger, die im Telegram Bot hinterlegt sind, benachrichtigt.

### ERWEITERTE ANTWORT

Wenn erweiterte Antwort aktiviert ist, können verschiedene Aktionen in der entsprechenden Liste definiert werden. Um festzulegen was bei einer Aktion zusätzlich ausgeführt wird kann ein ausgelöstes Ereignis erstellt werden. Als auslösende Variable wird die Variable 'Antwortaktion' und als Auslöser 'Bei bestimmtem Wert' gewählt. Als Wert kann nun die gewünschte Aktion ausgewählt werden.

## STATUSVARIABLEN

| STATUSVARIABLE | BESCHREIBUNG |
|---|---|
| Antwortaktion | Kann über die Visualisierung angeklickt oder per Befehl aufgerufen werden um die Benachrichtigung zurückzusetzen |
| Benachrichtigung aktiv | Steuert ob das Benachrichtigungsmodul aktiv ist oder nicht |
| Benachrichtigungsstufe | Beinhaltet die aktuelle Benachrichtigungsstufe |

## VISUALISIERUNG

Über die Visualisierung kann "Antwortaktion" ausgeführt werden um die Benachrichtigung zu reseten.
Über die Visualisierung kann über "Benachrichtigung aktiv" die Benachrichtigung An/Aus geschaltet werden.
Die aktuelle Benachrichtigungsstufe wird angezeigt.

# BN_IncreaseLevel

Benötigt: IP-Symcon >= 5.1

```
boolean BN_IncreaseLevel (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Falls noch eine weitere Stufe existiert, wird die aktuelle Stufe erhöht. Hierbei werden die Aktionen der neuen Stufe ausgeführt.

## BEISPIEL

```
BN_IncreaseLevel(12345);
```

# BN_Reset

Benötigt: IP-Symcon >= 5.1

```
boolean BN_Reset (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE

## BESCHREIBUNG

Setzt die Benachrichtigungsstufe zurück und deaktiviert die Benachrichtigungskette

## BEISPIEL

```
BN_Reset(12345);
```

# BN_SetNotifyLevel

Benötigt: IP-Symcon >= 5.1

```
boolean BN_SetNotifyLevel (int $InstanzID, int $Level)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| Level | Stufe des Benachrichtigungsmoduls |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Schaltet das Benachrichtigungsmodul auf die vorgegebene Stufe. Die Aktionen der neuen Stufe werden direkt ausgeführt. Aktionen von übersprungenen Stufen werden NICHT ausgeführt.

## BEISPIEL

```
BN_SetNotifyLevel(12345, 2);
```

# Durchsage

Benötigt: IP-Symcon >= 5.4
Das Durchsage Modul bietet die Möglichkeit von AWS Polly erzeugte Audio Daten über Sonos oder unter Windows den Media Player wiederzugeben.

## FUNKTIONSUMFANG

Von AWS Polly erzeugte Audiodaten über ein Sonos Player oder den "Symcon" Media Player wiedergeben
Lautstärke der Durchsage ist einstellbar
Durchsage bei Änderung der Text Variable. Alternativ über angebotene Funktion

## SOFTWARE-INSTALLATION

Über den Module Store das 'Durchsage'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Durchsage'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Text-to-Speeach Instanz | AWS Polly Instanz, welche zum erstellen der Durchsage genutzt werden soll. Als Ausgabeformat muss mp3 ausgewählt sein. Sonos kann nur mit den Abtastraten 16000 Hz, 22050 Hz, 24000 Hz, 32000 Hz, 44100 Hz, 48000 Hz genutzt werden |
| Ausgabegerät | Typ der Ausgabeinstanz |
| Symcon IP | Die IP-Addresse, unter welcher der Sonos Player IP-Symcon erreicht. |
| Sonos/Media Player | Instanz, über welche die Durchsage widergegeben wird |
| Lautstärke | Sonos: Die Lautstärkenänderung der Durchsage (0 → keine Änderung , 50 → 50, +10 → um 10 lauter) Media Player: die Lautstärke der Durchsage in Prozent (wird nicht zurückgesetzt) |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

## STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
| --- | --- | --- |
| Text | String | Text welcher für die Durchsage genutzt wird. Bei Aktualisierung der Variable wird der Inhalt wiedergegeben. |

# DS_Play

Benötigt: IP-Symcon >= 5.4

```
boolean DS_Play (int $InstanzID, string $Text)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| Text | Abzuspielender Text |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Spielt den Text als Durchsage ab.

## BEISPIEL

```
DS_Play(12345, "Dies ist ein Text");
```

# Dynamische E-Mail

Benötigt: IP-Symcon >= 7.1
Versendet eine E-Mail mit einem dynamischen Text.

## FUNKTIONSUMFANG

Versenden einer E-Mail mit einem festen/dynamischen Text
Dynamik durch das Ersetzen von Platzhaltern durch die Variablenwerte

## SOFTWARE-INSTALLATION

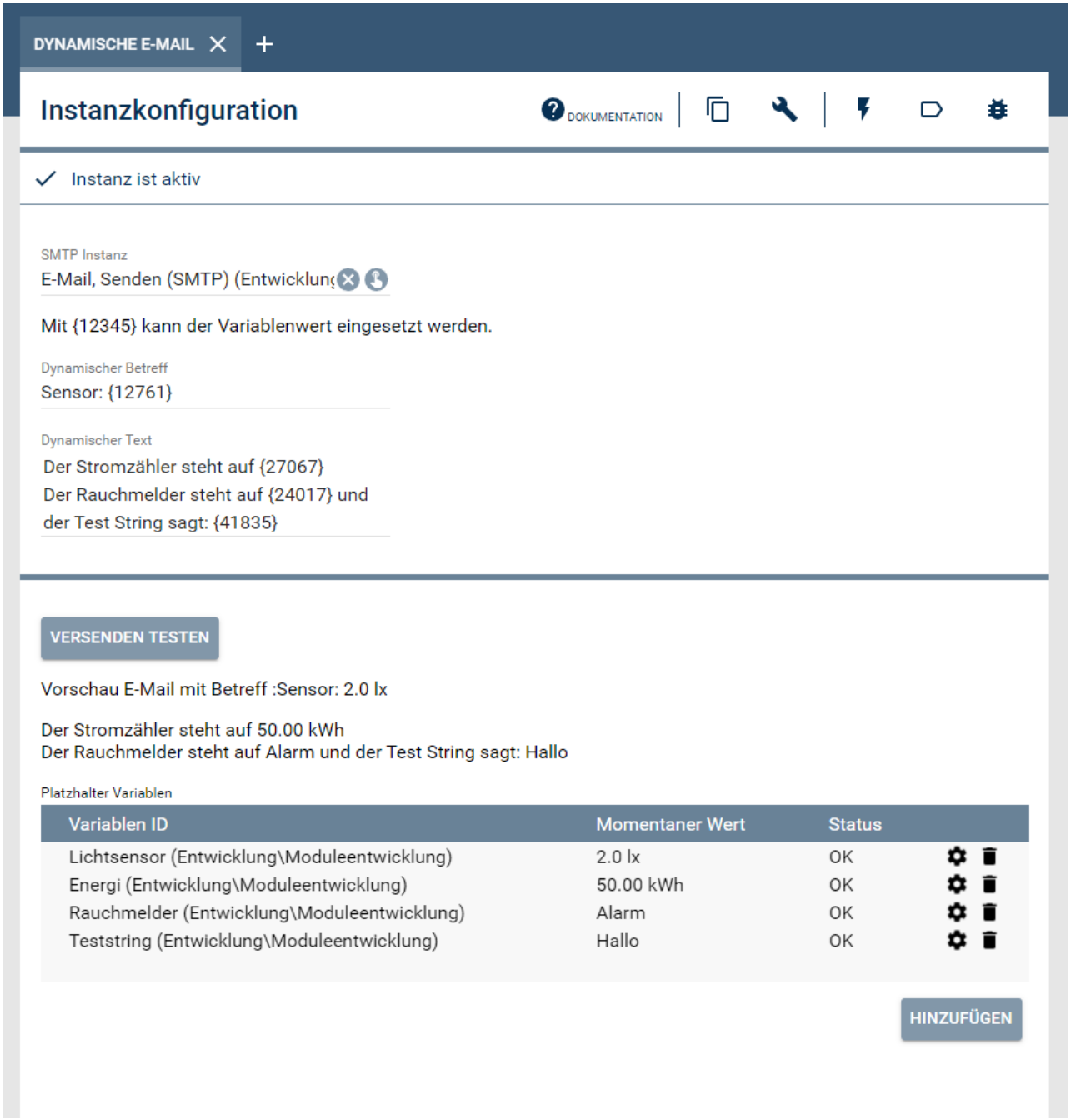
Über den Module Store das 'Dynamische E-Mail'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Dynamische E-Mail'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen
Konfigurationsseite:

| NAME | BESCHREIBUNG |
|---|---|
| SMTP Instanz | SMTP Instanz welche vollständig eingerichtet ist |
| Dynamischer Betreff | Texteingabe welche gesetzt werden kann. |
| Dynamischer Text | Mehrzeilige Texteingabe. |

Der Betreff wie auch der Text bekommt eine Dynamik durchs Einsetzen von Variablenwerte. Diese können innerhalb des Textes durch die in geschweifte Klammern gesetzte VariablenID gesetzt werden. Ist die ID zu kurz oder keine Variable, so bleiben die Klammern mit der Nummer bestehen.




Aktionsbereich:

| NAME | BESCHREIBUNG |
|---|---|
| Versenden Testen | Verschickt eine E-Mail mit dem gesetzten Betreff und Text |
| Vorschau E-Mail mit Betreff | Eine Vorschau mit gesetzten Variablewerte |
| Platzhalter Variablen | Tabelle mit den gegebenen Platzhaltern |

Tabelle:
In der Tabelle sind die gefundenen Platzhalter mit ihren aktuellen Wert und ihrem Status aufgelistet. Ist ein Platzhalter nicht verfügbar ist die eingegebene Nummer keine ID. Ist ein Platzhalter ungültig, so ist die gefundene ID keine Variable.

## STATUSVARIABLEN UND PROFILE

Es werden keine Variablen oder Profile angelegt.

## VISUALISIERUNG

Die Instanz hat keine Funktionalität in der Visualisierung.

# DM_SendMail

Benötigt: IP-Symcon >= 7.1

```
boolean DM_SendMail (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanze |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Nutzt den Betreff und Text um eine E-Mail zu versenden.

## BEISPIEL

```
DM_SendMail(12345);
```

# FertigMelder

Benötigt: IP-Symcon >= 4.2
Der FertigMelder meldet ob ein Gerät fertig ist. Dazu wird die Variable der Leistungsaufnahme des Geräts ausgewählt und ein Grenzwert festgelegt. Sobald dieser Grenzwert erstmalig überschritten wird, wird der FertigMelder gestartet. Wird dieser Grenzwert unterschritten und in einer einstellbaren Zeitspanne nicht wieder überschritten, wird die Statusvariable auf "Fertig" gesetzt.

## FUNKTIONSUMFANG

Ein/Ausschaltbarkeit des gesamten Moduls
Auswahl einer Quellvariable
Setzen des Grenzwerts
Zeitspanne bis Fertigmeldung nach Grenzwertunterschreitung gesetzt wird

## SOFTWARE-INSTALLATION

Unter "Instanz hinzufügen" ist kann 'Fertigmelder'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Quelle | Quellvariable, welche zum Vergleich mit dem Grenzwert genutzt wird. |
| Grenzwert | Wert bei desssen Unterschreitung eine Fertigmeldung gesetzt wird. |
| Intervall | Zeitraum bis eine Fertigmeldung gesetzt wird. Wird erst aktiv nachdem der Grenzwert unterschritten wird. |

## STATUSVARIABLEN UND PROFILE

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Aktiv | boolean | Schaltet das Modul Ein/Aus |
| Status | integer | Zeigt den Status an (Aus/Läuft/Fertig) |

### PROFILE:

| NAME | TYP |
| --- | --- |
| FM.Status | Integer |

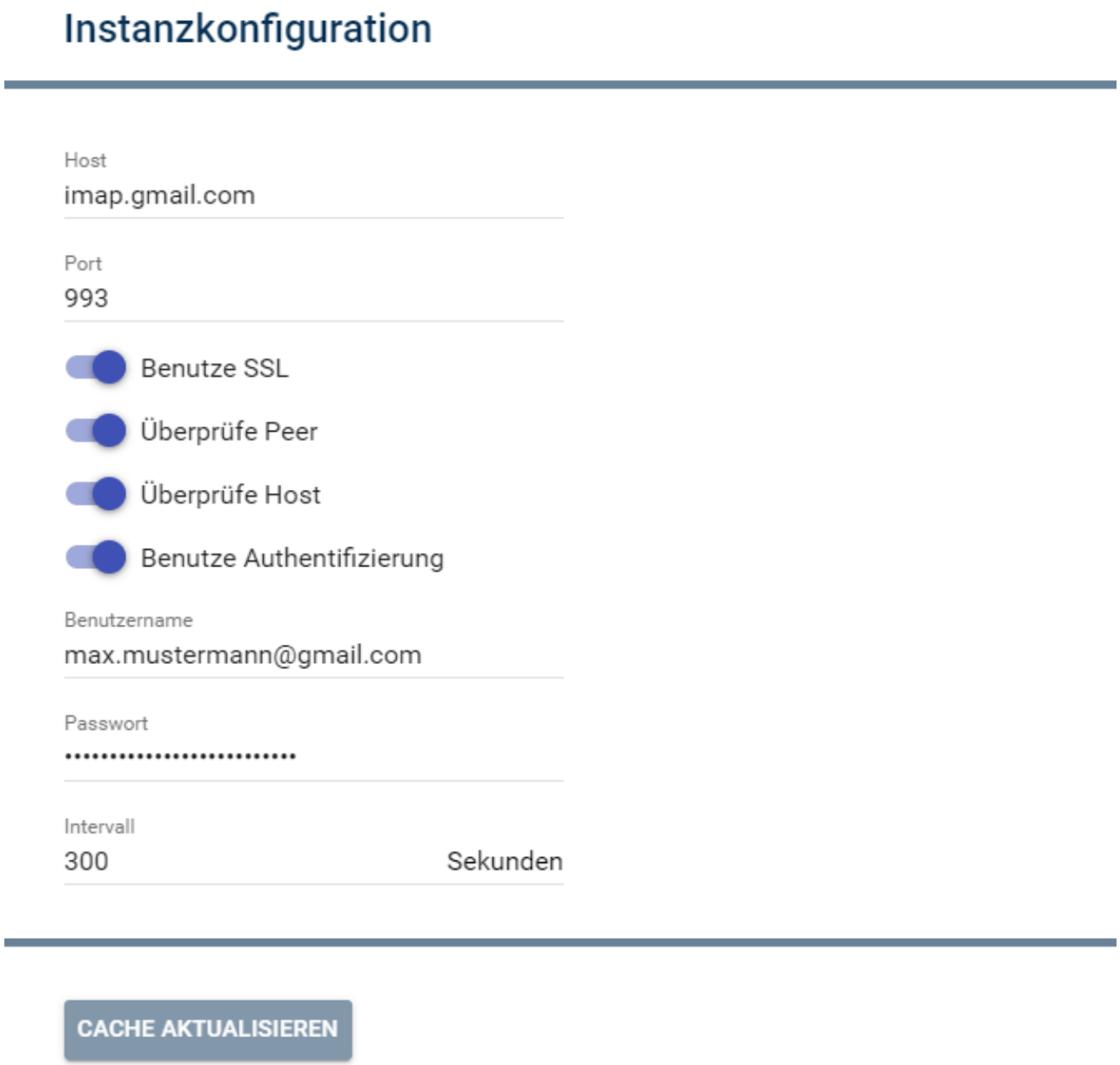
# IMAP

Benötigt: IP-Symcon >= 2.2
Das IMAP-Modul kann über die Konfigurationsseite innerhalb des Objektbaums der Verwaltungskonsole konfiguriert werden. Die Parameter für Host und Port können der Anleitung des jeweiligen Mail-Providers entnommen werden. Der Benutzername und das Passwort sind die Logindaten des E-Mail Accounts.



Seit Oktober 2022 wird die Verbindung zu Microsoft 365 (outlook.office365.com) nicht mehr unterstützt, da Microsoft nur noch OAuth2 als Authentifizierung erlaubt.

Damit die E-Mails in der Visualisierung angezeigt werden, muss das Intervall eingestellt werden. Dieses bestimmt die zyklische Abfrage des Posteingangs des Postfachs. Dabei werden nur die Betreffzeilen abgefragt. Die komplette E-Mail wird erst geladen, wenn die Visualisierung diese anfordert.
Über die Variable "Letzte Nachricht" kann ermittelt werden, ob und wann neue Mails eingetroffen sind. Falls keine E-Mails im Postfach sind, hat die Variable den Wert 0. Die Variable "Ungelesene Nachrichten" gibt die Anzahl der ungelesen Nachrichten zurück.




| EIGENSCHAFT | BEDEUTUNG |
|---|---|
| Host | Servername oder IP zum Mail-Provider |
| Port | Port, auf dem der Server arbeitet |
| Benutze SSL | Gibt an, ob SSL genutzt werden soll |
| Überprüfe Peer | Verifiziere, dass das SSL Zertifikat des Hosts korrekt ist (nur sichtbar falls Benutze SSL aktiv ist) |
| Überprüfe Host | Verifiziere, dass die Host URL aus dem SSL Zertifikat mit dem Host aus der Eigenschaft übereinstimmt (nur sichtbar falls Benutze SSL aktiv ist) |
| Benutze Authentifizierung | Gibt an, ob ein Benutzername/Passwort erforderlich ist |
| Benutzername | Benutzername |
| Passwort | Passwort |
| Intervall | Anzahl Sekunden, in der das Postfach abgefragt wird |



Eine Übersicht zu Host, Port und Authentifizierung für viele Provider kann hier gefunden werden: Tabelle

# IMAP_DeleteMail

Benötigt: IP-Symcon >= 5.0

```
boolean IMAP_DeleteMail (int $InstanzID, string $UID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der IMAP-Instanz |
| UID | UID der zu löschenden E-Mail |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Der Befehl löscht eine E-Mail mit der UID **UID** aus der IMAP-Instanz mit der ID **InstanzID**. Sollte die Instanz oder eine E-Mail mit der übergebenen UID nicht existieren, wird eine Warnung generiert.

## BEISPIEL

```
// Löscht die Mail mit der E-Mail(UID) 1234
IMAP_DeleteMail(12345, "1234"));
```

# IMAP_GetCachedMails

Benötigt: IP-Symcon >= 2.2

```
array IMAP_GetCachedMails (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der IMAP-Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis ein Array mit Daten zu den gecacheten E-Mails, andernfalls ein Boolean mit dem Wert **FALSE**.

## BESCHREIBUNG

Der Befehl gibt ein Array mit Informationen über die gecacheten E-Mails einer IMAP-Instanz mit der ID **InstanzID** zurück. Sollte die Instanz nicht existieren, wird eine Warnung generiert.

## BEISPIEL

```
print_r(IMAP_GetCachedMails(12345));

/* liefert z.B.:
Array
(
[0] => Array
(
[Date] => 1295756412
[Flags] => SEEN
[Recipient] => recipient@test.test
[SenderAddress] => sender@test.test
[SenderName] => Test Sender
[Subject] => 3 2 1 Test
[UID] => 1234
)
)
*/
```

# IMAP_GetMailEx

Benötigt: IP-Symcon >= 2.2

```
array IMAP_GetMailEx (int $InstanzID, string $UID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der IMAP-Instanz |
| UID | UID der zu ladenden E-Mail |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis ein Array mit Daten zu den gecacheten E-Mails, andernfalls ein Boolean mit dem Wert **FALSE**.

## BESCHREIBUNG

Der Befehl lädt eine E-Mail mit der UID **UID** über die IMAP-Instanz mit der ID **InstanzID** und gibt ein Array mit den Daten der E-Mail zurück. Sollte die Instanz oder eine E-Mail mit der übergebenen UID nicht existieren, wird eine Warnung generiert.

## BEISPIEL

```
print_r(IMAP_GetMailEx(12345, "1234"));

/* liefert z.B.:
Array
(
[ContentType] => text/plain
[Date] => 1295756412
[Flags] => SEEN
[Recipient] => ips@test.test
[SenderAddress] => sender@test.test
[SenderName] => Test Sender
[Subject] => 3 2 1 Test
[Text] => This is a test!
[UID] => 1234
)
*/
```

# MediaPlayer



Der MediaPlayer steht momentan nur bei der Windows Installation zur Verfügung.

Mit dem Media PlayerModul können verschiedene Sound-Dateien wie Wave, Midi oder MP3 abgespielt werden. Das Modul erlaubt die Verwendung mehrer Soundkarten. Damit können Sprachausgaben (Hinweise, Warnungen) oder die Lieblingsmusik raumbezogen ausgegeben werden.



Bei erstmaligem Anlegen der MediaPlayer-Instanz muss der Dienst neu gestartet werden, damit die Soundausgabe korrekt funktioniert!

## TIPPS & TRICKS

Es können beliebig viele Media Player Instanzen erstellt und sogar auf dieselbe Soundkarte konfiguriert werden. Dadurch kann über eine Instanz die Musik im Hintergrund (leise) weiterlaufen, während mit einer zweiten Instanz eine Ansage getätigt wird.

# WAC_AddFile

```
boolean WAC_AddFile (int $InstanzID, string $Dateiname)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu steuernden Media Players |
| Dateiname | Pfad zur abzuspielenden Datei |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Fügt eine Datei der internen Playlist vom Media Player mit der ID **InstanzID** zu.
Es wird nicht geprüft, ob die Datei schon vorhanden ist. Dieselbe Datei kann also mehrfach in der Playlist vorhanden sein.
Um die Wiedergabe zu starten, müssen Sie WAC_Play aufrufen.

## BEISPIEL

```
//Bitte beachten Sie die Slashes im Pfad.
//Die genaue Bedeutung können Sie im PHP Handbuch nachlesen:
//http://de.php.net/manual/de/language.types.string.php
WAC_AddFile(12345, "D:/MP3s/Lieblingssong.mp3");
```

# WAC_ClearPlaylist

```
boolean WAC_ClearPlaylist (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu steuernden Media Players |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Löscht die interne Playlist vom Media Player mit der ID **InstanzID**.

## BEISPIEL

```
WAC_ClearPlaylist(12345);
```

# WAC_GetPlaylistLength

```
int WAC_GetPlaylistLength (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu steuernden Media Players |

## RÜCKGABEWERT

Anzahl der Tracks, die in der Playlist eingetragen sind.

## BESCHREIBUNG

Gibt die Länge der internen Playlist vom Media Player mit der ID **InstanzID** zurück.

## BEISPIEL

```
$len = WAC_GetPlaylistLength(12345);
if($len > 0)
WAC_Play(12345);
```

# WAC_GetPlaylistPosition

```
int WAC_GetPlaylistPosition (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Wert zwischen 1 und WAC_GetPlayListLength; Falls die Playlist leer sein sollte, wird -1 zurückgegeben.

## BESCHREIBUNG

Gibt die aktuelle Position der internen Playlist vom Media Player mit der ID **InstanzID** zurück.

## BEISPIEL

```
$pos = WAC_GetPlaylistPosition(12345);
```

# WAC_Next

```
boolean WAC_Next (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu steuernden Media Players |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Geht in der internen Playlist vom Media Player mit der ID **InstanzID** einen Titel vorwärts oder wählt zufällig einen Titel aus, wenn Shuffle aktiviert ist. Sollte die Wiedergabe gestartet sein, wird nach dem Zurückgehen der dann neue, aktuelle Titel von Anfang an abgespielt.
Sollte die Playliste leer sein, wird eine Fehlermeldung generiert.
Sollte die Playlist am Ende angelangt sein, wird wieder entweder eine Fehlermeldung generiert oder, falls Repeat aktiviert ist, am Anfang der Playlist neu gestartet.

## BEISPIEL

```
WAC_Next(12345);
```

# WAC_Pause

```
boolean WAC_Pause (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu steuernden Media Players |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**. Es wird auch **TRUE** zurückgegeben, falls keine aktive Wiedergabe vorhanden ist.

## BESCHREIBUNG

Pausiert die Wiedergabe der internen Playlist vom Media Player mit der ID **InstanzID**. Durch ein erneutes aufrufen von WAC_Pause oder einem Aufruf von WAC_Play kann die Wiedergabe fortgesetzt werden.

## BEISPIEL

```
WAC_Pause(12345);
```

# WAC_Play

```
boolean WAC_Play (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu steuernden Media Players |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Startet die Wiedergabe der internen Playlist vom Media Player mit der ID **InstanzID**. Falls Sie noch keine Dateien hinzugefügt haben, können Sie dies über WAC_AddFile tun. Eine direkte Funktion zum Abspielen einer einzelnen Datei ist WAC_PlayFile. Bitte beachten Sie auch die unterstützten Dateiformate, die sich auf WAC_PlayFile finden können.
Sollte eine Wiedergabe bereits gestartet worden sein, wird durch Aufrufen dieses Befehls die Wiedergabe vom Anfang der Datei gestartet.
Sollte die Playliste leer sein, wird eine Fehlermeldung generiert.

## BEISPIEL

```
WAC_Play(12345); //Abspielen
```

# WAC_PlayFile

```
boolean WAC_PlayFile (int $InstanzID, string $Dateiname)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu steuernden Media Players |
| Dateiname | Pfad zur abzuspielenden Datei |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Spielt eine Datei mit Dateiname direkt über den Media Player mit der ID **InstanzID** ab. Der Befehl ist nur zur Vereinfachung vorhanden. Sie können ihn auch durch eine Abfolge von WAC_ClearPlaylist, WAC_AddFile, WAC_Play ersetzen.
Bitte beachten Sie die unterstützten Dateiformate.
Zur Zeit unterstützte Formate: ***.wav,*** **.mp3, *.wma** (jeweils ohne DRM)

## BEISPIEL

```
//Bitte beachten Sie die Slashes im Pfad.
//Die genau Bedeutung können Sie im PHP Handbuch nachlesen: http://
de.php.net/manual/de/language.types.string.php
WAC_PlayFile(12345, "D:/MP3s/Lieblingssong.mp3");
```

# WAC_Prev

```
boolean WAC_Prev (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu steuernden Media Players |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Geht in der internen Playlist vom Media Player mit der ID **InstanzID** einen Titel zurück.
Sollte die Wiedergabe gestartet sein, wird nach dem Zurückgehen der dann neue aktuelle Titel von Anfang an abgespielt.
Sollte die Playliste leer sein, wird eine Fehlermeldung generiert.
Sollte die Playlist am Anfang angelangt sein, wird der aktuelle Titel von Vorne abgespielt.

## BEISPIEL

```
WAC_Prev(12345);
```

# WAC_SetPlaylistPosition

```
boolean WAC_SetPlaylistPosition (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt die aktuelle Position der internen Playlist vom Media Player mit der ID **InstanzID** auf die neue **Position**. Nach dem Ausführen muss WAC_Play ausgeführt werden, wenn die Wiedergabe an der neuen Position starten soll. Andernfalls wird der nächste Titel, nach dem Ende des aktuellen Titels abgespielt.

## BEISPIEL

```
WAC_SetPlaylistPosition(12345, 0); //An Anfang setzen
```

# WAC_SetPosition

```
boolean WAC_SetPosition (int $InstanzID, int $Sekunden)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu steuernden Media Players |
| Sekunden | Sprungmarke in Sekunden |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den Positionsmarker vom Media Player mit der ID **InstanzID** auf die angegeben **Sekunden**. Der Titel muss zum Ausführungszeitpunkt bereits abgespielt werden.

## BEISPIEL

```
WAC_PlayFile(12345, "my.mp3"); //Titel abspielen
WAC_SetPosition(12345, 60); //auf die 1 Minutenmarke im Lied
springen
```

# WAC_SetRepeat

```
boolean WAC_SetRepeat (int $InstanzID, boolean $Wiederholen)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu steuernden Media Players |
| Wiederholen | TRUE für An, FALSE für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Aktiviert/ deaktiviert die Wiederholung der Playlist vom Media Player mit der ID InstanzID. Der Befehl wird sofort übernommen, macht sich aber erst beim nächsten Track der aktuellen Playlist bemerkbar. Die aktuelle Wiedergabe wird nicht beeinflusst.

## BEISPIEL

```
WAC_SetRepeat(12345, true); //Playlist wiederholung aktivieren
```

# WAC_SetShuffle

```
boolean WAC_SetShuffle (int $InstanzID, boolean $Zufallswiedergabe)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu steuernden Media Players |
| Zufallswiedergabe | TRUE für An, FALSE für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Aktiviert/ deaktiviert die Zufallswiedergabe vom Media Player mit der ID InstanzID. Der Befehl wird sofort übernommen, macht sich aber erst beim nächsten Track der aktuellen Playlist bemerkbar. Die aktuelle Wiedergabe wird nicht beeinflusst.

## BEISPIEL

```
WAC_SetShuffle(12345, true); //Zufallswiedergabe einschalten
```

# WAC_SetVolume

```
boolean WAC_SetVolume (int $InstanzID, int $Lautstärke)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu steuernden Media Players |
| Lautstärke | 0-100 (%) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Fadet die Lautstärke vom Media Player mit der ID **InstanzID** innerhalb von 500ms auf die gewünschte Lautstärke.

## BEISPIEL

```
WAC_SetVolume(12345, 85);
```

# WAC_Stop

```
boolean WAC_Stop (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu steuernden Media Players |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE. Es wird auch TRUE zurückgegeben, falls keine aktive Wiedergabe vorhanden ist.

## BESCHREIBUNG

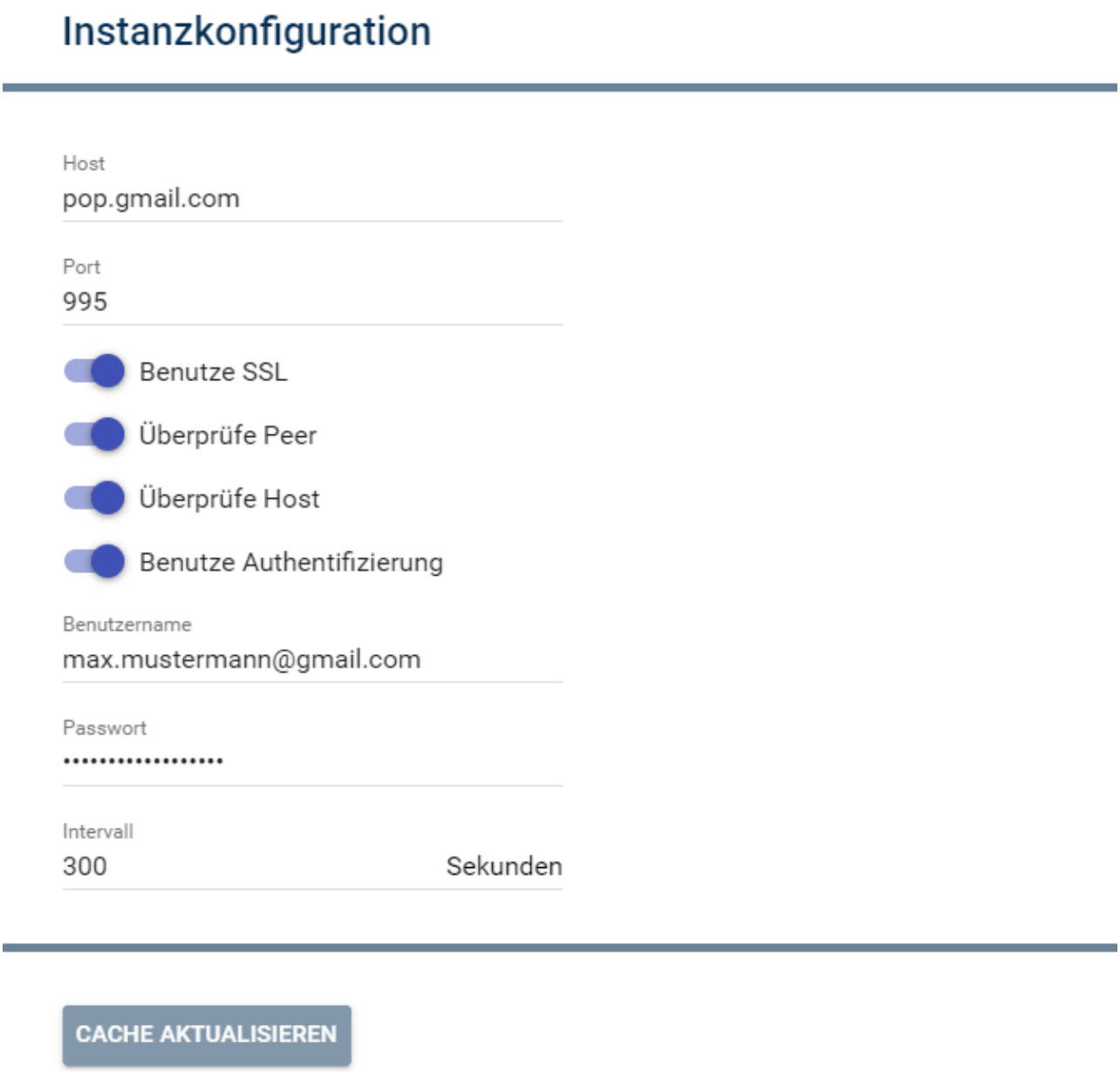
Stoppt die Wiedergabe der internen Playlist vom Media Player mit der ID InstanzID.

## BEISPIEL

```
WAC_Stop(12345);
```

# POP3

Benötigt: IP-Symcon >= 2.2
Das POP3-Modul kann über die Konfigurationsseite innerhalb des Objektbaums der Verwaltungskonsole konfiguriert werden. Die Parameter für Host und Port können der Anleitung des jeweiligen Mail-Providers entnommen werden. Der Benutzername und das Passwort sind die Logindaten des E-Mail Accounts.
Damit die E-Mails in der Visualisierung angezeigt werden, muss das Intervall eingestellt werden. Dieses bestimmt die zyklische Abfrage des Posteingangs des Postfachs. Dabei werden nur die Betreffzeilen abgefragt. Die komplette E-Mail wird erst geladen, wenn die Visualisierung diese anfordert.
Über die Variable "Letzte Nachricht" kann ermittelt werden, ob und wann neue Mails eingetroffen sind. Falls keine E-Mails im Postfach sind, hat die Variable den Wert 0. Im Gegensatz zu IMAP ist die Anzeige der Anzahl der ungelesenen E-Mails für POP3 technisch nicht realisierbar.



| EIGENSCHAFT | BEDEUTUNG |
| --- | --- |
| Host | Servername oder IP zum Mail-Provider |
| Port | Port, auf dem der Server arbeitet |
| Benutze SSL | Gibt an, ob SSL genutzt werden soll |
| Überprüfe Peer | Verifiziere, dass das SSL Zertifikat des Hosts korrekt ist (nur sichtbar falls Benutze SSL aktiv ist) |
| Überprüfe Host | Verifiziere, dass die Host URL aus dem SSL Zertifikat mit dem Host aus der Eigenschaft übereinstimmt (nur sichtbar falls Benutze SSL aktiv ist) |
| Benutze Authentifizierung | Gibt an, ob ein Benutzername/Passwort erforderlich ist |
| Benutzername | Benutzername |
| Passwort | Passwort |
| Intervall | Anzahl Sekunden, in der das Postfach abgefragt wird |



Eine Übersicht zu Host, Port und Authentifizierung für viele Anbieter können Sie hier finden: Tabelle

# POP3_DeleteMail

Benötigt: IP-Symcon >= 5.0

```
boolean POP3_DeleteMail (int $InstanzID, string $UID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der POP3-Instanz |
| UID | UID der zu löschenden E-Mail |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Der Befehl löscht eine E-Mail mit der UID **UID** aus der POP3-Instanz mit der ID **InstanzID**. Sollte die Instanz oder eine E-Mail mit der übergebenen UID nicht existieren, wird eine Warnung generiert.

## BEISPIEL

```
// Löscht die Mail mit der E-Mail(UID) 1234
POP3_DeleteMail(12345, "1234");
```

# POP3_GetCachedMails

Benötigt: IP-Symcon >= 2.2

```
array POP3_GetCachedMails (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der POP3-Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis ein Array mit Daten zu den gecacheten E-Mails, andernfalls ein Boolean mit dem Wert **FALSE**.

## BESCHREIBUNG

Der Befehl gibt ein Array mit Informationen über die gecacheten E-Mails einer POP3-Instanz mit der ID **InstanzID** zurück. Sollte die Instanz nicht existieren, wird eine Warnung generiert.

## BEISPIEL

```
print_r(POP3_GetCachedMails(12345));

/* liefert z.B.:
Array
(
[0] => Array
(
[Date] => 1295756412
[Flags] =>
[Recipient] => recipient@test.test
[SenderAddress] => sender@test.test
[SenderName] => Test Sender
[Subject] => 3 2 1 Test
[UID] => 1234
)
)
*/
```

# POP3_GetMailEx

Benötigt: IP-Symcon >= 2.2

```
array POP3_GetMailEx (int $InstanzID, string $UID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der POP3-Instanz |
| UID | UID der zu ladenden E-Mail |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis ein Array mit den Daten der E-Mail, andernfalls ein Boolean mit **FALSE**.

## BESCHREIBUNG

Der Befehl lädt eine E-Mail mit der UID **UID** über die POP3-Instanz mit der ID **InstanzID** und gibt ein Array mit den Daten der E-Mail zurück. Sollte die Instanz oder eine E-Mail mit der übergebenen UID nicht existieren, wird eine Warnung generiert.

## BEISPIEL

```
print_r(POP3_GetMailEx(12345, "1234"));

/* liefert z.B.:
Array
(
[ContentType] => text/plain
[Date] => 1295756412
[Flags] =>
[Recipient] => ips@test.test
[SenderAddress] => sender@test.test
[SenderName] => Test Sender
[Subject] => 3 2 1 Test
[Text] => This is a test!
[UID] => 1234
)
*/
```

# Popup Modul

Das Popup Modul ist eine Instanz, die keine eigene Funktion oder Status Variablen beinhaltet. Das Modul kann als stilistisches Mittel verwendet werden, um Objekte innerhalb eines Popups darzustellen.

## ANWENDUNG

Mehrere Objekte gehören zu einer Themengruppe, welche aber nur bei Bedarf schnell eingesehen werden soll. Um die Objekte im Popup anzeigen zu lassen, müssen diese selbst oder via Link in die Popup Instanz kopiert werden.

## DARSTELLUNG

Wie das Modul dargestellt wird kann man hier nachsehen:
Popup Darstellung



In den mobilen Apps hat die Popup Instanz die gleiche Darstellung wie ein Dummy Modul, da diese keine Verschachtelung ermöglichen.

# SMS

Benötigt: IP-Symcon >= 4.1
SMS ist ein Modul von IP-Symcon mit dem sich SMS via Clickatell verschicken lassen. Eine Verbindung mit IP-Symcon wird über die REST API hergestellt.



Seit November 2016 bietet Clickatell nur noch die REST-API an, welche IP-Symcon mit dem Modul SMS (Clickatell) integriert hat.

## INSTALLATION (REST API)

Um das SMS Modul verwenden zu können, wird ein Account bei clickatell.com benötigt.
Die nachfolgende Step-By-Step Anleitung erklärt den Anmeldeprozess:

Die folgende URL öffnet die Anmeldeseite zur Registrierung am SMS Gateway:
clickatell.com/sign-up
Nach dem Bestätigen der Verifizierungs-Mail muss sich unter portal.clickatell.com eingeloggt werden. Es muss "Platform" ausgewählt werden.
Über das Dashboard auf der Seite portal.clickatell.com müssen sowohl Telefonnummern verifiziert, als auch eine "Integration" eingerichtet werden.
Eine Integration konfiguriert die Art der Kommunikation.
Guthaben wird über portal.clickatell.com verwaltet.

Damit IP-Symcon mit dem Gateway kommunizieren kann, muss zusätzlich noch der APIKey der Integration in IP-Symcon eingepflegt werden.

Folgende URL muss geöffnet werden:
portal.clickatell.com
Es muss die Option "**API key**" kopiert werden.
Der erhaltene APIKey, muss in IP-Symcon auf der Konfigurationsseite des SMS-Modul angegeben werden.



Die Absendernummer muss im Clickatell Account verfiziert worden sein, bevor über diese SMS versendet werden können.

## INSTALLATION (VERALTET, HTTPS API)

Diese API wird seit November 2016 nicht mehr von Clickatell unterstützt.
Aus dem Grund wurde das Modul überarbeitet.

## EINBINDUNG IN IP-SYMCON

Die Konfiguration besteht ausschließlich aus dem API-Key, der im Account eingerichtet wurde.



Es gibt mit der neuen REST-API bisher keine Möglichkeit das Guthaben abzufragen. SMS_RequestBalance ist nur für die alte API noch funktionial.

## LÄNDERCODE VORWAHL NUMMER

Es müssen keine führenden Nullen angegeben werden.
Beispiel: +49171123456789
Es gibt auch auf der Clickatell-Homepage auf der Settings-Seite die Möglichkeit eine automatische Ländervorwahl zu integrieren.
"Convert mobile numbers into international format"
Somit entfällt das +49 für die Ländervorwahl.

# SMS_Send

```
boolean SMS_Send (int $InstanzID, string $Nummer, string $Text)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Nummer | Mobilfunknummer des Empfängers |
| Text | Text der SMS (max. 160 Zeichen) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet über die Instanz mit der ID **InstanzID** eine SMS mit dem Text an Nummer.



Clickatel (veraltet): Es wird die Absendernummer genutzt, die in der Eigenschaftenseite angegeben wurde.
Clickatel (REST): Es wird die Absendernummer genutzt, welche in der API angegeben wurde.

## BEISPIEL

```
SMS_Send(12345, "0171123456789", "Alarm!");
```

# SMTP

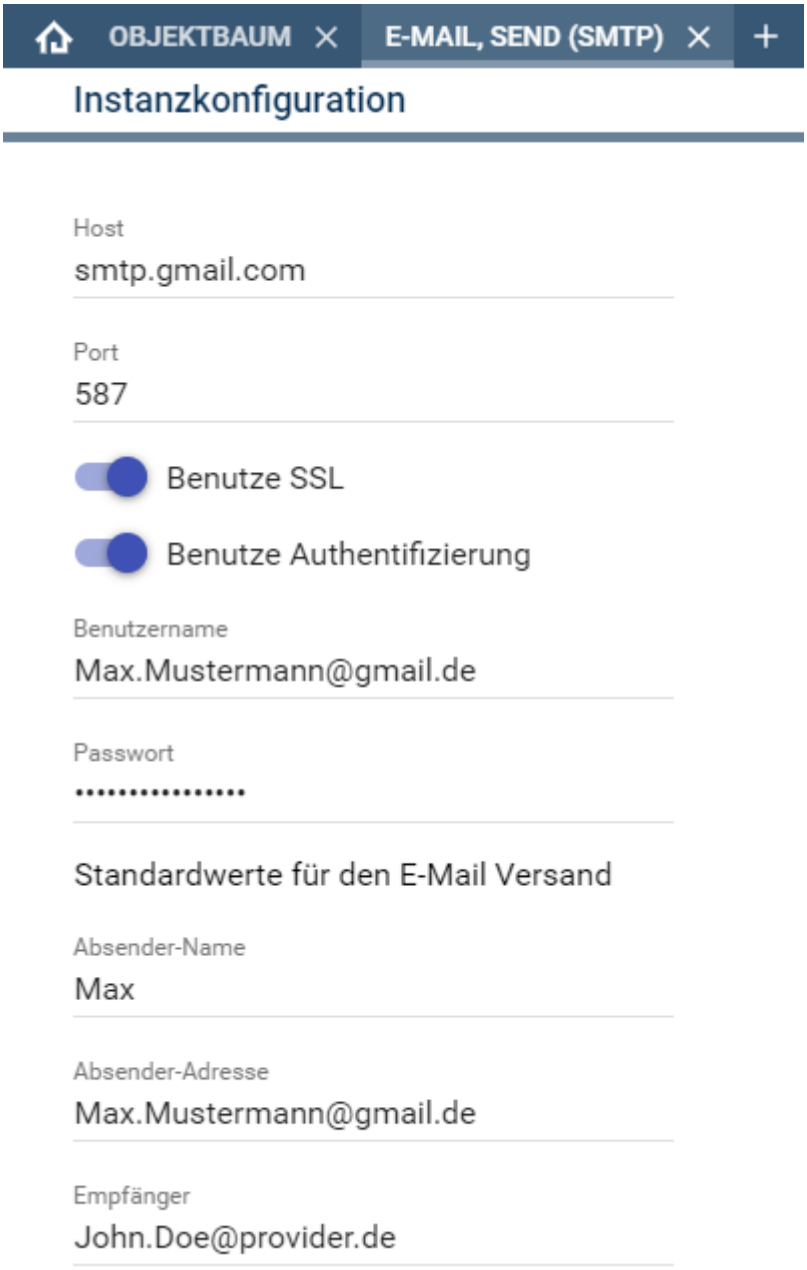
Benötigt: IP-Symcon >= 2.2
SMTP ist ein Modul von IP-Symcon, welches E-Mails verschicken kann. Das Modul wird als Instanz in IP-Symcon eingerichtet. Es ist unter Hersteller "None" und "E-Mail senden (SMTP)" zu finden. Der Tabelle können die einzelnen Eigenschaften entnommen werden.



Um an eine andere oder mehrere E-Mail Adressen eine Nachricht zu senden muss die Funktion SMTP_SendMailEx genutzt werden.



Die Emails können nicht speziell formatiert werden. /n /r sind funktional aber jegliche Formatierung wie z.B. bold, kursiv etc funktionieren nicht. Dazu muss z.B. dann ein PHPMailer genutzt werden.




| EIGENSCHAFT | BEDEUTUNG |
|---|---|
| Host | Servername oder IP zum Mail-Provider |
| Port | Port, auf dem der Server arbeitet |
| SSL | Gibt an, ob SSL genutzt werden soll |
| Authentifizierung | Gibt an, ob ein Benutzername/Passwort erforderlich ist |
| Benutzername | Benutzername (Details gibt es beim Mail-Provider) |
| Passwort | Passwort |
| Absender-Name | Der Name, der beim Empfänger sichtbar ist |
| Absender-Adresse | Die Adresse, die beim Empfänger sichtbar ist. Je nach Provider muss die Absender-Adresse der eigenen Adresse entsprechen. |
| Empfänger | Nur erforderlich für die SMTP_SendMail Funktion, die automatisch an den Empfänger dieses Feldes sendet. |

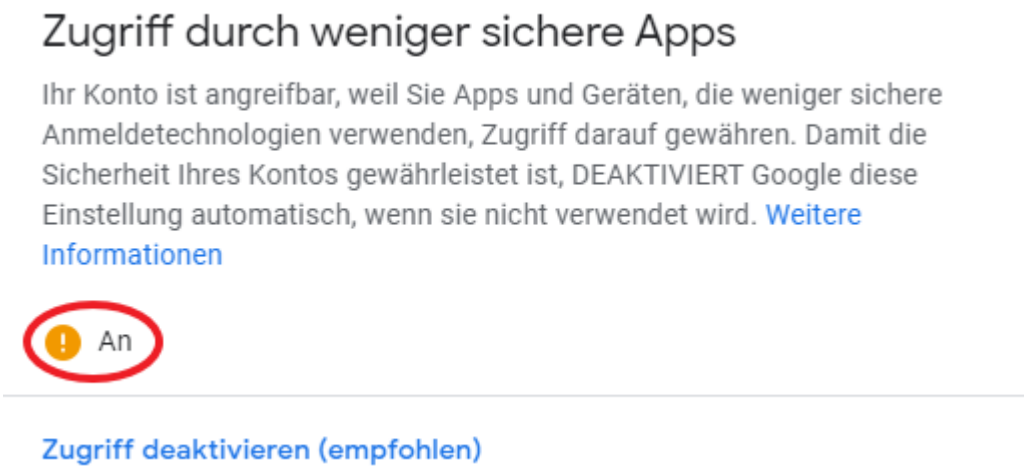
Eine Übersicht zu Host, Port und Authentifizierung für viele Provider können hier eingesehen werden: Tabelle

## BEISPIEL MIT GMAIL

GoogleMail lässt einen direkten Zugriff mit dem Benutzernamen und Passwort nicht ohne weitere Einstellung zu. Bei diesen Einstellungen kommt es darauf an, ob man eine Zwei-Faktor Authentifizierung aktiviert hat.

### OHNE ZWEI-FAKTOR AUTHENTIFIZIERUNG

Für die Nutzung ohne Zwei-Faktor Authentifizierung muss in der Verwaltung des Google Kontos unter "Sicherheit" der "Zugriff durch weniger sichere Apps" aktiviert werden. Ein Direktlink wäre myaccount.google.com/lesssecureapps .
Wenn der Zugriff aktiviert ist, sieht es wie folgt aus. Es kann sich nun mit Benutzername und Passwort eingeloggt werden.

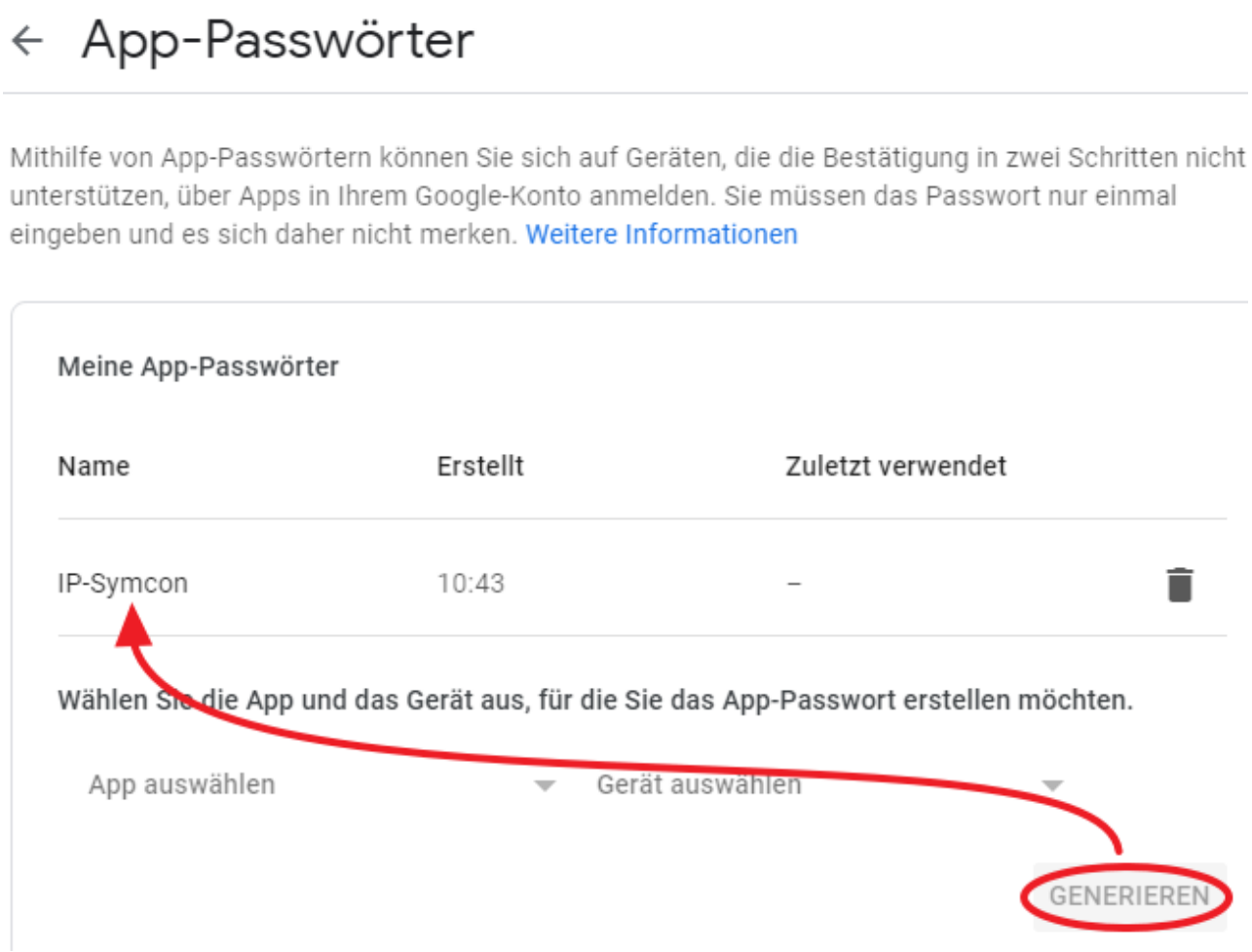



### MIT ZWEI-FAKTOR AUTHENTIFIZIERUNG

Bei der Nutzung mit Zwei-Faktor Authentifizierung muss in der Verwaltung des Google Kontos unter "Sicherheit" im Abschnitt "Bei Google anmelden" ein App-Passwort vergeben werden.
Hier auf die Zeile "App-Passwörter" klicken.

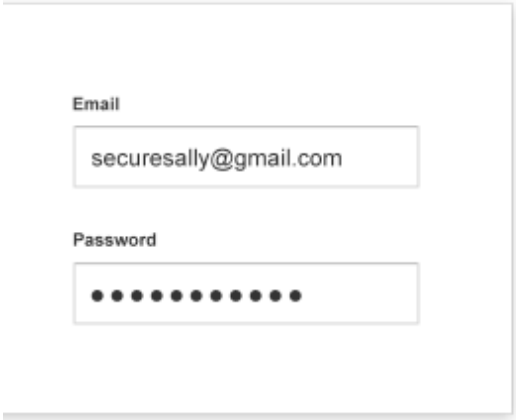



In der Überischt kann ein neues Passwort mit einem individuellen Namen generiert werden.




Das generierte Passwort muss dann in IP-Symcon eingetragen werden.

## Generiertes App-Passwort




Ihr App-Passwort für Ihr Gerät

**eymx reme ojye pauj**

Verwendung

Öffnen Sie in der App oder auf dem Gerät, die bzw. das Sie einrichten möchten, die Einstellungen Ihres Google-Kontos. Ersetzen Sie Ihr Passwort durch das oben angezeigte 16-stellige Passwort.
Wie mit Ihrem normalen Passwort haben Sie auch mit diesem App-Passwort vollen Zugriff auf Ihr Google-Konto. Sie müssen es sich jedoch nicht merken und sollten es weder aufschreiben noch an Dritte weitergeben.

FERTIG

# SMTP_SendMail

Benötigt: IP-Symcon >= 2.2

```
boolean SMTP_SendMail (int $InstanzID, string $Betreff, string
$Inhalt)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der SMTP Instanz |
| Betreff | Betreff der E-Mail |
| Inhalt | Inhalt der E-Mail. Wird als HTML versendet (seit 7.0), sofern <html> und </html> gefunden wird, ansonsten als Text |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet über die Instanz mit der ID **InstanzID** eine E-Mail mit dem Betreff und Text.



Es wird der Empfänger genutzt, der in der Eigenschaftenseite angegeben wurde. Wenn unterschiedliche Empfänger addressiert werden sollen, nutzen Sie bitte die SMTP_SendMailEx Funktion.

## BEISPIEL

```
SMTP_SendMail(12345, "Alarm!", "Die Heizung ist ausgefallen!");
```

# SMTP_SendMailAttachment

Benötigt: IP-Symcon >= 2.2

```
boolean SMTP_SendMailAttachment (int $InstanzID, string $Betreff,
string $Inhalt, string $Dateipfad)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der SMTP Instanz |
| Betreff | Betreff der E-Mail |
| Inhalt | Inhalt der E-Mail. Wird als HTML versendet (seit 7.0), sofern <html> und </html> gefunden wird, ansonsten als Text |
| Dateipfad | Pfad zur Datei |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet über die Instanz mit der ID **InstanzID** eine E-Mail mit dem Betreff, Anhang und Text.



Es wird der Empfänger genutzt, der in der Eigenschaftenseite angegeben wurde. Wenn unterschiedliche Empfänger addressiert werden sollen, nutzen Sie bitte die SMTP_SendMailAttachmentEx Funktion.

## BEISPIEL

```
SMTP_SendMailAttachment(12345, "Alarm!", "Die Heizung ist
ausgefallen!", "media/webcam.jpg");
```

# SMTP_SendMailAttachmentEx

Benötigt: IP-Symcon >= 2.2

```
boolean SMTP_SendMailAttachmentEx (int $InstanzID, string
$Empfänger, string $Betreff, string $Inhalt, string $Dateipfad)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der SMTP Instanz |
| Empfänger | E-Mail Adresse des Empfängers |
| Betreff | Betreff der E-Mail |
| Inhalt | Inhalt der E-Mail. Wird als HTML versendet (seit 7.0), sofern <html> und </html> gefunden wird, ansonsten als Text |
| Dateipfad | Pfad zur Datei |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet über die Instanz mit der ID **InstanzID** eine E-Mail mit dem Betreff, Anhang und Text an Empfänger. Im Gegensatz zur SMTP_SendMail Funktion kann hier der Empfänger explizit im Funktionsaufruf angegeben werden.

## BEISPIEL

```
SMTP_SendMailAttachmentEx(12345, "me@example.com", "Alarm!", "Die
Heizung ist ausgefallen!", "media/webcam.jpg");
```

# SMTP_SendMailEx

Benötigt: IP-Symcon >= 2.2

```
boolean SMTP_SendMailEx (int $InstanzID, string $Empfänger, string
$Betreff, string $Inhalt)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der SMTP Instanz |
| Empfänger | E-Mail Adresse des Empfängers |
| Betreff | Betreff der E-Mail |
| Inhalt | Inhalt der E-Mail. Wird als HTML versendet (seit 7.0), sofern <html> und </html> gefunden wird, ansonsten als Text |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet über die Instanz mit der ID **InstanzID** eine E-Mail mit dem Betreff und Text an Empfänger. Im Gegensatz zur SMTP_SendMail Funktion kann hier der Empfänger explizit im Funktionsaufruf angegeben werden.

## BEISPIEL

```
//Eine Nachricht an einen Empfänger versenden, welche anders sein
kann als die Eingestellte.
SMTP_SendMailEx(12345, "ich@provider.com", "Alarm!", "Die Heizung
ist ausgefallen!");

//Eine Nachricht an mehrere Empfängern versenden.
$empfaengerListe = Array("ich@provider.com",
"meinpartner@provider.com", "meinhausmeister@provider.com");
foreach($empfaengerListe as $empfaenger) {
SMTP_SendMailEx(12345, $empfaenger, "Alarm!", "Die Heizung ist
ausgefallen!");
}
```

# SMTP_SendMailMedia

Benötigt: IP-Symcon >= 4.0

```
boolean SMTP_SendMailMedia (int $InstanzID, string $Betreff, string
$Inhalt, int $MedienID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der SMTP Instanz |
| Betreff | Betreff der E-Mail |
| Inhalt | Inhalt der E-Mail. Wird als HTML versendet (seit 7.0), sofern <html> und </html> gefunden wird, ansonsten als Text |
| MedienID | ID des Medienobjekts (nur Bild/Ton und Dokument) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet über die Instanz mit der ID **InstanzID** eine E-Mail mit dem Betreff, Medienobjekt und Text.



Es wird der Empfänger genutzt, der in der Eigenschaftenseite angegeben wurde. Wenn unterschiedliche Empfänger addressiert werden sollen, nutzen Sie bitte die SMTP_SendMailMediaEx Funktion.

## BEISPIEL

```
SMTP_SendMailMedia(12345, "Alarm!", "Da hat jemand an der Tür
geklingelt, aber du bist leider nicht da!", 23456);
```

# SMTP_SendMailMediaEx

Benötigt: IP-Symcon >= 4.0

```
boolean SMTP_SendMailMediaEx (int $InstanzID, string $Empfänger,
string $Betreff, string $Inhalt, int $MedienID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der SMTP Instanz |
| Empfänger | E-Mail Adresse des Empfängers |
| Betreff | Betreff der E-Mail |
| Inhalt | Inhalt der E-Mail. Wird als HTML versendet (seit 7.0), sofern <html> und </html> gefunden wird, ansonsten als Text |
| MedienID | ID des Medienobjekts (nur Bild/Ton und Dokument) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet über die Instanz mit der ID **InstanzID** eine E-Mail mit dem Betreff, Medienobjekt und Text an Empfänger. Im Gegensatz zur SMTP_SendMailMedia Funktion kann hier der Empfänger explizit im Funktionsaufruf angegeben werden.

## BEISPIEL

```
SMTP_SendMailMediaEx(12345, "me@example.com", "Alarm!", "Da hat
jemand an der Tür geklingelt, aber du bist leider nicht da!",
23456);
```

# Spotify

Benötigt: IP-Symcon >= 5.3
Das Spotify-Modul ermöglicht die Verknüpfung mit einem Spotify-Premium-Konto. Darüber kann die Wiedergabe über das Konto gesteuert werden. So ist es möglich, die eine Wiedergabe zu starten oder die aktuelle Wiedergabe zu ändern bzw. zu beenden. Auch Optionen wie Wiederholung oder Zufallswiedergabe können gesetzt werden. Innerhalb des Moduls werden Favoriten gespeichert, welche dann komfortabel aus der Visualisierung abgerufen werden können.

## FUNKTIONSUMFANG

Verknüpfung mit einem Spotify-Premium-Konto
Wiedergabe abspielen oder pausieren
Zu nächstem oder vorherigen Song wechseln
Gerät für Wiedergabe auswählen oder wechseln
Favoriten abspeichern und wieder aufrufen
Wiederholung oder Zufallswiedergabe aktivieren

## VORAUSSETZUNGEN

Spotify-Premium-Konto

## SOFTWARE-INSTALLATION

Über den Module Store das 'Spotify'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Spotify'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

Initial muss das Modul durch einen Klick auf den Button "Registrieren" mit einem Spotify-Premium-Konto verknüpft werden. Durch den Klick öffnet sich ein Anmeldungsdialog von Spotify. Nach Eingabe der Benutzerdaten muss die Verknüpfung bestätigt werden. Danach stehen alle Funktionen des Moduls zur Verfügung.

#### *ERWEITERTE EINSTELLUNGEN*

| NAME | BESCHREIBUNG |
|---|---|
| Aktualisierungsintervall | In diesem Intervall werden die Werte der Statusvariablen mit der aktuellen Wiedergabe von Spotify abgeglichen. Dies beinhaltet das Gerät, Wiedergabe oder Pause, Zufallswiedergabe und Wiederholung |
| Cover: Maximale Breite | Maximale Breite des dargestellten Covers |
| Cover: Maximale Höhe | Maximale Höhe des dargestellten Covers |

### *SUCHE*

| NAME | BESCHREIBUNG |
|---|---|
| Suche | In diesem Feld wird ein Suchbegriff für eine Suche eingegeben |
| Suche nach Alben | Gibt an ob die nächste Suche Alben beinhalten soll |
| Suche nach Künstlern | Gibt an ob die nächste Suche Künstler beinhalten soll |
| Suche nach Playlists | Gibt an ob die nächste Suche Playlists beinhalten soll |
| Suche nach Songs | Gibt an ob die nächste Suche Songs beinhalten soll |
| Suche starten | Startet eine Suche |
| Suchergebnisse | Diese Liste beinhaltet die Ergebnisse der aktuellen Suche - Durch ein Aktivieren des Feldes "Favorit" kann ein Ergbnis den Favoriten hinzugefügt werden |

Die Liste "Ihre Playlists" beinhaltet die Playlists des verknüpften Spotify-Kontos. Diese können über das Feld "Favorit" den Favoriten hinzugefügt werden. Die Liste "Favoriten" beinhaltet die eingestellten Favoriten. Durch ein Klick auf das Mülleimer-Symbol können Favoriten entfernt werden.

## STATUSVARIABLEN UND PROFILE

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Aktuelles Cover | HTMLBox | Hier wird das Cover des aktuell abgespielten Albums dargestellt, sofern vorhanden |
| Aktueller Song | string | Hier wird der Name des aktuell abgespielten Songs angezeigt |
| Aktueller Künstler | string | Hier wird der Künstler des aktuell abgespielten Songs angezeigt |
| Aktuelles Album | string | Hier wird das Album des aktuell abgespielten Songs angezeigt |
| Aktion | integer | Hierüber kann eine Wiedergabe gestartet oder pausiert werden. Außerdem ist es möglich auf den vorherigen oder nächsten Song zu wechseln |
| Gerät | integer | Mit dieser Variable kann ein Wiedergabegerät ausgewählt werden. Ist aktuell eine Wiedergabe aktiv, so wird diese auf das neue Gerät gewechselt |
| Favorit | integer | Diese Variable stellt die im Modul konfigurierten Favoriten dar. Durch die Auswahl eines Favoriten wird dieser auf dem aktuell gewählten Gerät abgespielt |
| Wiederholen | integer | Über diese Variable kann die Wiederholung aktiviert werden - "Aus": Keine Wiederholung, "Kontext": Der aktuelle Kontext, also das Album, die Playlist, ... wird wiederholt, "Song": Der aktuelle Song wird wiederholt |
| Zufällige Wiedergabe | boolean | Aktiviert bzw. deaktiviert die zufällige Wiedergabe |

### PROFILE:

| NAME | TYP |
|---|---|
| Spotify.Favorites. | Integer |
| Spotify.Devices | Integer |
| Spotify.Repeat | Integer |

## VISUALISIERUNG

Über die Visualisierung oder in den mobilen Apps werden Statusvariablen angezeigt und können geschaltet werden.

# SPO_MakeAPIRequest

Benötigt: IP-Symcon >= 5.3

```
string SPO_MakeAPIRequest (int $InstanzID, string $Method, string
$Url, string $Body)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| Method | Art der HTML-Anfrage |
| Url | Pfad, welcher nach `https://api.spotify.com/v1` steht |
| Body | Nachricht, welche an den Endpunkt gesendet werden soll |

## RÜCKGABEWERT

Die Funktion gibt die Rückgabe der API als String zurück oder false bei einem Fehler.

## BESCHREIBUNG

Stellt eine Anfrage an die Spotify Web-API mit den angegebenen Parametern. Dabei muss nur der endpunktspezifische Pfad angegeben werden, der allgemeine Teil, also https://api.spotify.com/v1 wird automatisch hinzugefügt

## BEISPIEL

```
SPO_MakeAPIRequest(12345, 'GET', '/me', '');
```

# SPO_NextTrack

Benötigt: IP-Symcon >= 5.3

```
boolean SPO_NextTrack (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Existiert eine aktuelle Wiedergabe, so wird der nächste Song abgespielt.

## BEISPIEL

```
SPO_NextTrack(12345);
```

# SPO_Pause

Benötigt: IP-Symcon >= 5.3

```
boolean SPO_Pause (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Existiert eine aktuelle Wiedergabe, so wird diese pausiert.

## BEISPIEL

```
SPO_Pause(12345);
```

# SPO_Play

Benötigt: IP-Symcon >= 5.3

```
boolean SPO_Play (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Existiert eine aktuelle, möglicherweise pausierte, Wiedergabe, so wird diese fortgesetzt. Ansonsten wird der aktuell ausgewählte Favorit auf dem aktuell ausgewähltem Gerät abgespielt.

## BEISPIEL

```
SPO_Play(12345);
```

# SPO_PlayURI

Benötigt: IP-Symcon >= 5.3

```
boolean SPO_PlayURI (int $InstanzID, string $URI)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| URI | URI der Spotify-Resource |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Spielt die Spotify-Resource mit der URI **URI** auf dem aktuell ausgewähltem Gerät ab.

## BEISPIEL

```
SPO_PlayURI(12345, 'spotify:artist:1Lw1vZvhgNZk7hVSvdY4OA');
```

# SPO_PreviousTrack

Benötigt: IP-Symcon >= 5.3

```
boolean SPO_PreviousTrack (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Existiert eine aktuelle Wiedergabe, so wird der vorherige Song abgespielt.

## BEISPIEL

```
SPO_PreviousTrack(12345);
```

# SPO_ResetToken

Benötigt: IP-Symcon >= 5.3

```
boolean SPO_ResetToken (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt die OAuth-Token und somit die Verknüpfung mit dem Spotify-Konto zurück.

## BEISPIEL

```
SPO_ResetToken(12345);
```

# SPO_SetRepeat

Benötigt: IP-Symcon >= 5.3

```
boolean SPO_SetRepeat (int $InstanzID, string $Repeat)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| Repeat | |

| WERT | BEDEUTUNG |
|---|---|
| 0 | Wiederholung deaktiviert |
| 1 | Wiederholung des Kontext |
| 2 | Wiederholung des Songs |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt die Wiederholung auf den Wert **Repeat**.

## BEISPIEL

```
SPO_SetRepeat(12345, 2);
```

# SPO_SetShuffle

Benötigt: IP-Symcon >= 5.3

```
boolean SPO_SetShuffle (int $InstanzID, boolean $Shuffle)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| Shuffle | De-/Aktiviert die zufällige Wiedergabe |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Ist **Shuffle** true, so wird die zufällige Wiedergabe aktiviert, ansonsten deaktiviert.

## BEISPIEL

```
SPO_SetShuffle(12345, true);
```

# Störungsmanager

Benötigt: IP-Symcon >= 6.0
Das Modul dient dazu aktive Variablen im Webfront anzuzeigen und je nach Einstellung nach Eingabe oder Deaktivierung auszublenden.
Wenn 'Automatisches Ausblenden', aber nicht 'Bestätigung' aktiviert wurde, wird ein Link sichtbar, sobald die überwachte Variable aktiv ist. Dieser verschwindet sobald die überwachte Variabel deaktiviert ist.
Wenn 'Bestätigung' und 'Automatisches Ausblenden' aktiviert wurde, wird eine Variable erstellt, sobald die überwachte Variable aktiv ist. Diese verschwindet sobald bei der erstellten Variable der Wert auf 1 bzw. 'In Bearbeitung' geändert wurde und die überwachte Variable inaktiv ist.
Wenn 'Bestätigung' aber nicht 'Automatisches Ausblenden' aktiviert wurde, wird eine Variable erstellt, sobald die überwachte Variable aktiv ist. Diese verschwindet sobald bei der erstellten Variable der Wert auf über 2 bzw. 'Alles in Ordnung' geändert wurde.

## FUNKTIONSUMFANG

Anzeige von aktiven Variablen
Möglichkeit diese je nach Auswahl zu quittieren

## SOFTWARE-INSTALLATION

Über den Module Store das Modul Störungsmanger installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Störungsmanager'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

Konfigurationsseite:

| NAME | BESCHREIBUNG |
|---|---|
| Variable | Variable, welche überwacht werden sollen. |
| Bestätigung | Checkbox, damit die Meldung bestätigt oder nicht bestätigt werden soll. |
| Automatisches Ausblenden | Checkbox, damit die Meldung bei deaktivierten Zustand wieder automatisch ausgeblendet wird |
| Umgang mit Nachrichten | Auswahl wie mit den Variablen umgegangen werden soll, wenn diese aktiv sind. |

## STATUSVARIABLEN UND PROFILE

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

Statusvariablen

| NAME | TYP | BESCHREIBUNG |
| --- | --- | --- |
| Link: Variablennamen | Link | Link, welcher sichtbar oder unsichtbar ist. Wird erstellt, wenn 'Meldung verschwindet automatisch' ausgewählt wird. |
| Variablennamen - Status | Integer | Variable, welche erstellt wird, wenn 'Quittiere Meldung' oder 'Quittiere Meldung, verschwindet, wenn die Störung behoben wurde' ausgewählt wurde. |

Profile:

| BEZEICHNUNG | BESCHREIBUNG |
| --- | --- |
| STA.Confirm | Profil für erstellte Variablen, mit 3 Stufen |
| STA.ConfirmHide | Profil für erstellte Variablen, mit 2 Stufen |

## VISUALISIERUNG

Über die Visualisierung werden die erstellten Variablen angezeigt und können verändert werden, wenn diese quittiert werden müssen.

# SymconReport

Erstellt je nach Module ein Report als CSV oder PDF

| | |
|---|---|
| MailReport | Erstellt einen CSV-Report von Archivwerten und sendet diesen per E-Mail(SMTP) |
| MR_SendInfo | Verschickt eine Info-Mail über das letzte abgeschlossene Zeitintervall |
| MR_SetActive | Aktiviert oder deaktiviert das Modul |
| PDFReport (Energie) | Erstellt ein Report von Archivwerten zweier Variablen einer Zeitspanne als PDF |
| RAC_GenerateEnergyReport | Generiert ein PDF mit den im Modul mit der InstanzID |
| PDFReport (Multi) | Erstellt einen Report von Archivwerten mehrerer Variablen einer Zeitspanne als PDF |
| RAC_GenerateReport | Generiert ein PDF mit den im Modul mit der InstanzID angegebenen Werten |
| PDFReport (Multi Energie) | |
| PDFReport (Single) | Erstellt einen Report von Archivwerten einer Variable mit Min/Max/Avg einer Zeitspanne als PDF |
| RAC_GenerateReport | Generiert ein PDF mit den im Modul mit der InstanzID angegebenen Werten |

# MailReport

Benötigt: IP-Symcon >= 5.5
Das Modul verschickt regelmäßig die geloggten aggregierten Daten einer Variablen via E-Mail.

## FUNKTIONSUMFANG

Verschicke E-Mails mit aggregierten Daten einer Variablen
- E-Mails werden nach jedem Abschluß des gewählten Zeitintervalls (Täglich, Wöchentlich oder Monatlich)
- E-Mails beinhalten aggregierte Daten, welche auch in einem entsprechenden Graphen dargestellt würden (Stündlich bei täglichem Zeitintervall, Täglich bei wöchentlichem oder monatlichem Zeitintervall)
- Die E-Mail enthält eine CSV-Datei, welche analog zum CSV-Export von Graphen formatiert ist.

## VORAUSSETZUNGEN

Eingerichtete SMTP-Instanz zum Versenden von E-Mails

## SOFTWARE-INSTALLATION

Über den Module Store kann unter der Kategorie "Informationen"=>"Aufbereitung" oder direkt über die Suche nach "Report Modul" das Modul gefunden werden. Durch den Knopf "Installieren" wird das Modul IP-Symcon zur Verfügung gestellt.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Über "Instanz hinzufügen" kann das 'Report(Mail)'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| E-Mail SMTP | Auswahl der SMTP-Instanz über welche die E-Mails verschickt werden |
| Aggregierte Variable | Auswahl der Variablen wessen aggregierte Daten verschickt werden sollen |
| Dezimaltrennzeichen | Auswahl vom Dezimaltrennzeichen Komma oder Punkt. |
| Intervall der Nachrichten | Auswahl des Zeitintervalls in welchen die Nachrichten verschickt werden sollen |
| Sende Daten | Betätigen um eine Info-Mail über das letzte abgeschlossene Zeitintervall zu verschicken |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
| --- | --- | --- |
| Mail Report aktiv | Variable | Aktiviert bzw. deaktiviert das Modul |

## VISUALISIERUNG

Über die Visualisierung kann das Modul aktiviert oder deaktiviert werden.

# MR_SendInfo

Benötigt: IP-Symcon >= 5.5

```
boolean MR_SendInfo (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Verschickt eine Info-Mail über das letzte abgeschlossene Zeitintervall

## BEISPIEL

```
MR_SendInfo(12345);
```

# MR_SetActive

Benötigt: IP-Symcon >= 5.5

```
boolean MR_SetActive (int $InstanzID, boolean $Active)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| Active | De-/Aktivieren des Moduls |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Aktiviert oder deaktiviert das Modul in Abhängigkeit von **Active**

## BEISPIEL

```
// Aktiviere das Modul MR_SetActive(12345, true);
```

# PDFReport (Energie)

Benötigt: IP-Symcon >= 5.5
Dieses Modul bietet die Funktion die Archivwerte zweier Variablen als Graphen dazustellen, und ausgewählte Angabe als PDF für den vergangenen Monat bereitzustellen.

## FUNKTIONSUMFANG

Ermöglicht das Erstellen und Download von erstellter PDF.
Einstellung via Instanzkonfiguration
Logoauswahl für Header ist möglich

## SOFTWARE-INSTALLATION

Über den Module Store kann unter der Kategorie "Informationen"=>"Aufbereitung" oder direkt über die Suche nach "Report Modul" das Modul gefunden werden. Durch den Knopf "Installieren" wird das Modul IP-Symcon zur Verfügung gestellt.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Report (PDF, Energie)'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

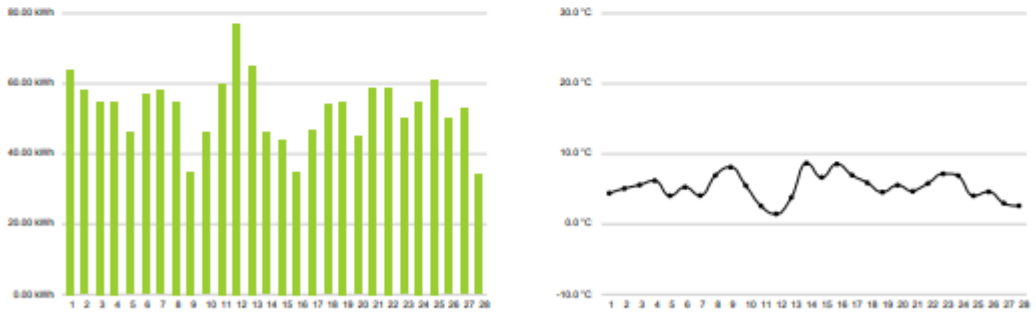
| NAME | BESCHREIBUNG |
|---|---|
| Logo | (optional) Auswählbare Grafik als PNG |
| Verbrauchsart | Freitext welcher unterhalb des Datums steht |
| Dezimaltrennzeichen | Auswahl vom Dezimaltrennzeichen Komma oder Punkt. |
| Verbrauchszähler | Als Zähler geloggte Variable |
| Temperatur | (optional) geloggte Variable |
| Vorhersage | (optional) Variable, welche den Verbrauch aufgrund des Verhaltens vorhersagen kann, zum Beispiel aus dem Modul Verbrauchsverhalten |
| CO2 Typ | (optional) CO2 Equivalent für die Verbrauchsart |

### BEISPIEL DOKUMENT

Jaetzel

FEBRUAR 2022
IHR GASVERBRAUCH

## VERBRAUCH



Im Februar haben Sie 1478.00 kWh verbraucht.

## VERHALTEN

### Februar

Die Durchschnittstemperatur betrug: 4.6 °C
Erwarteter Verbrauch aufgrund Ihres Verhaltens: 1537.60 kWh
Tatsächlicher Verbrauch: 1478.00 kWh

Sie haben in diesen Zeitraum Ihren Verbrauch um 3.88% gesenkt.

### CO2 Emmisionen

353.242 kg

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

## STATUSVARIABLEN

Die PDF wird als "Report (PDF)" generiert.

| NAME | TYP | BESCHREIBUNG |
| --- | --- | --- |
| Report (PDF) | Medien | Erstellte PDF, welche über die Visualisierung heruntergeladen oder per E-Mail verschickt werden kann. |

## VISUALISIERUNG

Über die Visualisierung kann die generierte PDF heruntergeladen werden.

# RAC_GenerateEnergyReport

Benötigt: IP-Symcon >= 5.5

```
boolean RAC_GenerateEnergyReport (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Generiert ein PDF mit den im Modul mit der InstanzID **InstanzID** angegebenen Werten.
Die PDF steht dann als Mediendatei zur Verfügung.
Wenn bereits eine Datei existiert, wird diese aktualisiert.

## BEISPIEL

```
RAC_GenerateEnergyReport(12345);
```

# PDFReport (Multi)

Benötigt: IP-Symcon >= 5.5
Dieses Modul bietet die Funktion Archivwerte als Bericht in einer PDF zusammenzufassen.

## FUNKTIONSUMFANG

Ermöglicht das Erstellen und Download von erstellter PDF.
Einstellung via Instanzkonfiguration
Logoauswahl für Header ist möglich
Einstellbare Anzahl Datensätze, Aggregationsstärke, Min- und Max-Werte der ausgewählten Variablen

## SOFTWARE-INSTALLATION

Über den Module Store kann unter der Kategorie "Informationen"=>"Aufbereitung" oder direkt über die Suche nach "Report Modul" das Modul gefunden werden. Durch den Knopf "Installieren" wird das Modul IP-Symcon zur Verfügung gestellt.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Report (PDF, Multi)'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Logo | Auswählbare Grafik als PNG |
| Firma | Firmenname |
| Title | Titel der PDF |
| Fußzeile | Fußzeile |
| Datenquelle | Mehrere Variablen aus der die Datensätze pro Spalte erstellt werden |
| Dezimaltrennzeichen | Auswahl vom Dezimaltrennzeichen Komma oder Punkt. |
| Aggregation | Definiert Aggregationsstufe der aufgelisteten Datensätze (Stunde - Jahr) |
| Anzahl | Anzahl der aufgelisteten Datensätze |
| Datensatz überspringen | Ist diese Option aktiviert, wird der aktuellste unvollständige Datensatz verworfen |

### BEISPIEL DOKUMENT

Symcon GmbH
29.10.2021

**Titel**

| Datum | Variable 1 | Variable 2 | Variable 3 |
|---|---|---|---|
| 29.10.2021 23:26 | 100 % | 100 % | 100 % |

Fußzeile

# STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

## STATUSVARIABLEN

Die PDF wird als "Report (PDF)" generiert.

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Report (PDF) | Medien | Erstellte PDF, welche über die Visualisierung heruntergeladen oder per E-Mail verschickt werden kann. |

# VISUALISIERUNG

Über die Visualisierung kann die generierte PDF heruntergeladen werden.

# RAC_GenerateReport

Benötigt: IP-Symcon >= 5.5

```
boolean RAC_GenerateReport (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Generiert ein PDF mit den im Modul mit der InstanzID **InstanzID** angegebenen Werten.
Die PDF steht dann als Mediendatei zur Verfügung.
Wenn bereits eine Datei existiert, wird diese aktualisiert.

## BEISPIEL

```
RAC_GenerateReport(12345);
```

# PDFReport (Multi Energie)

Dieses Modul bietet die Funktion den Gesamtverbrauch der ausgewählten Variablen als auch den Einzelverbrauch in der Zeitspanne anzuzeigen.

## FUNKTIONSUMFANG

Ermöglicht das Erstellen und Download von erstellter PDF.
Einstellung via Instanzkonfiguration

## SOFTWARE-INSTALLATION

Über den Module Store kann unter der Kategorie "Informationen"=>"Aufbereitung" oder direkt über die Suche nach "Report Modul" das Modul gefunden werden. Durch den Knopf "Installieren" wird das Modul IP-Symcon zur Verfügung gestellt.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Report (PDF, Energie)'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Aggregation Level | Auswahl des Detailgrades der verwendeten Daten |
| Dezimal Separator | Auswahl, wie der Dezimal Separator dargestellt werden soll |
| Energiezähler | Liste der Energiezähler, welche verwendet werden sollen |

### BEISPIEL DOKUMENT

# VERBRAUCH

Von 01.03.2024 bis 23.10.2024

**Gesamtverbrauch: 250407.7 kwh**

| Zähler | Verbrauch | Prozentsatz |
|---|---|---|
| Stromzähler | 63132.9 kWh | 25.21 % |
| Stromzähler (Haus) | 187274.8 kWh | 74.79 % |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

## STATUSVARIABLEN

Die PDF wird als "Report (PDF)" generiert.

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Start | Integer | Gibt das Startdatum an, ab wann die Daten genutzt werden sollen |
| End | Integer | Gibt das Enddatum an, bis wann die Daten genutzt werden sollen |
| Generate | Skript | Ausführbar, um die PDF zu aktualisieren |
| Report (PDF) | Medien | Erstellte PDF, welche über die Visualisierung heruntergeladen oder per E-Mail verschickt werden kann. |

## VISUALISIERUNG

Über die Visualisierung kann Start- und Enddatum eingegeben, die PDF generiert und heruntergeladen werden.

# PDFReport (Single)

Benötigt: IP-Symcon >= 5.5
Dieses Modul bietet die Funktion Archivwerte als Bericht in einer PDF zusammenzufassen. Dabei wird eine einzige Zielvariable dargestellt mit Ihren Min/Max/Avg Werten.

## FUNKTIONSUMFANG

Ermöglicht das Erstellen und Download von erstellter PDF.
Einstellung via Instanzkonfiguration
Logoauswahl für Header ist möglich
Einstellbare Anzahl Datensätze, Aggregationsstärke, Min- und Max-Werte der ausgewählten Variablen

## SOFTWARE-INSTALLATION

Über den Module Store kann unter der Kategorie "Informationen"=>"Aufbereitung" oder direkt über die Suche nach "Report Modul" das Modul gefunden werden. Durch den Knopf "Installieren" wird das Modul IP-Symcon zur Verfügung gestellt.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Report (PDF, Single)'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Logo | Auswählbare Grafik als PNG |
| Firma | Firmenname |
| Title | Titel der PDF |
| Fußzeile | Fußzeile |
| Datenquelle | Variable aus der die Datensätze erstellt werden |
| Aggregation | Definiert Aggregationsstufe der aufgelisteten Datensätze (Stunde - Jahr) |
| Dezimaltrennzeichen | Auswahl vom Dezimaltrennzeichen Komma oder Punkt. |
| Anzahl | Anzahl der aufgelisteten Datensätze |
| Datensatz überspringen | Ist diese Option aktiviert, wird der aktuellste unvollständige Datensatz verworfen |
| Toleranz (Min) | Akzeptierter Minimalwert der aggregierten Datensätze. Werte außerhalb der Toleranz werden ignoriert. |
| Toleranz (Max) | Akzeptierter Maximalwert der aggregierten Datensätze. Werte außerhalb der Toleranz werden ignoriert. |

### BEISPIEL DOKUMENT




Firma
16.08.2019

**Titel**

| Datum | Min | Max | Mittel | Toleranz |
|---|---|---|---|---|

Fußzeile

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

Die PDF wird als "Report (PDF)" generiert.

| NAME | TYP | BESCHREIBUNG |
| --- | --- | --- |
| Report (PDF) | Medien | Erstellte PDF, welche über die Visualisierung heruntergeladen oder per E-Mail verschickt werden kann. |

## VISUALISIERUNG

Über die Visualisierung kann die generierte PDF heruntergeladen werden.

# RAC_GenerateReport

Benötigt: IP-Symcon >= 5.5

```
boolean RAC_GenerateReport (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Generiert ein PDF mit den im Modul mit der InstanzID **InstanzID** angegebenen Werten.
Die PDF steht dann als Mediendatei zur Verfügung.
Wenn bereits eine Datei existiert, wird diese aktualisiert.

## BEISPIEL

```
RAC_GenerateReport(12345);
```

# Telefonansage

Benötigt: IP-Symcon >= 5.5
Das Telefonansage-Modul ermöglicht die komfortable Verknüpfung einer VoIP-Instanz und einer Text-to-Speech-Instanz (Polly) um eine Telefonnummer anzurufen und bei Annahme des Anrufs einen Text auszugeben. Darüber hinaus, kann das Modul auf DTMF-Töne reagieren und weitere Texte ausgeben.

## FUNKTIONSUMFANG

Anrufen einer Telefonnummer und Ausgabe eines Textes
Auswerten von DTMF-Tönen
Ausgabe von weiteren Texten auf einer bestehenden Verbindung

## VORAUSSETZUNGEN

Eingerichtete VoIP-Instanz
Eingerichtete Text-to-Speech-Instanz (AWS Polly)

## SOFTWARE-INSTALLATION

Über den Module Store das 'Telefonansage'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Telefonansage'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| VoIP-Instanz | Die VoIP-Instanz, welche die Anrufe verwaltet |
| Text-to-Speech-Instanz (Polly) | Die Text-to-Speech-Instanz, welche die Texte in Tondaten umwandelt |
| Dauer bis die Verbindung getrennt wird | Die maximale Zeit, die gewartet wird, dass ein Anruf angenommen wird |

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

## STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Telefonnummer | String | Die Telefonnummer, welche bei der Betätigung von "Anruf starten" angerufen wird |
| Text | String | Der Text, welcher bei der Betätigung von "Anruf starten" initial ausgegeben wird. Wird diese Variable geschaltet während eine Verbindung besteht, wird der neue Text direkt ausgegeben |
| DTMF-Ton | String | Wird während des Anrufs ein DTMF-Ton verwendet, wird er in dieser Variable gespeichert |
| Anruf starten | Skript | Startet einen Anruf mit der aktuell eingestellten Telefonnummer und dem aktuellen Text |

# TA_StartCall

Benötigt: IP-Symcon >= 5.5

```
void TA_StartCall (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

## BESCHREIBUNG

Startet einen Anruf auf Basis der aktuellen Einstellung.

## BEISPIEL

```
TA_StartCall(12345);
```

# TA_StartCallEx

Benötigt: IP-Symcon >= 5.5

```
void TA_StartCallEx (int $InstanzID, string $Telefonnummer, string
$Text)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| Telefonnummer | Anzurufende Telefonnummer |
| Text | Auszugebender Text |

## RÜCKGABEWERT

## BESCHREIBUNG

Startet einen Anruf an die Telefonnummer **Telefonnummer** und gibt den Text **Text** aus.

## BEISPIEL

```
TA_StartCallEx(12345, "+4945130500511", "Ich liebe IP-Symcon!");
```

# Telefonkette

Benötigt: IP-Symcon >= 5.5
Das Telefonkette-Modul ermöglicht es eine Liste von Telefonnummern nacheinander anzurufen.

## FUNKTIONSUMFANG

Eine Liste von Telefonnummern kann absteigend angerufen werden
Bei Annahme kann eine TTS-Datei abgespielt werden
Weitere Anrufe werden beendet, wenn ein Angerufener durch ein DTMF-Zeichen bestätigt
DTMF-Zeichen frei wählbar
Dauer bis ein Anruf beendet wird frei wählbar
Bool Variable als Auslöser

## VORAUSSETZUNGEN

Eigenes VoIP Abarbeitungsskript für die Audio-Ansage oder eine eingerichtete TTSAWSPOLLY-Instanz

## SOFTWARE-INSTALLATION

Über den Module Store das 'Telefonkette'-Modul installieren

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Telefonkette'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
| --- | --- |
| Auslöser | Bool Variable die als Auslöser dient |
| VoIP Instanz | VoIP Instanz, welche für die Anrufe genutzt werden soll |
| TTS Instanz | TTSAWSPOLLY Instanz, welche für die Ansage genutzt werden soll |
| TTS Type | **Statisch**, Statischer Text für eine Ansage oder dynamische Ansage über eine String-Variable |
| Telefonnummern in Visu schaltbar | Option um die Telefonnummern in der Visualisierung an und ausschaltbar zu machen |
| Telefonnummern | Liste und Reihenfolge der Nummern mit Beschreibung, welche angerufen werden sollen |
| Anzahl gleichzeitiger Anrufe | Anzahl der Anrufe, welche parallel gemacht werden |
| Anrufdauer | Zeit in Sekunden, welche auf ein Annehmen des Anrufes gewartet wird |
| DTMF Bestätigungstaste | Taste, mit der ein Anruf bestätigt wird |

Anmerkung zu der Option 'Telefonnummern in Visu schalbar': Diese sorgt dafür das die Telefonnummern in der Liste als Variable erstellt werden. Die Telefonnummern werden nur dann angerufen, wenn die dazugehörige Variable den Wert 'True' und in der Visualisierung aktiviert ist.

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
| --- | --- | --- |
| Anruf bestätigt von | String | Beinhaltet die Nummer, welche den Anruf bestätigt hat |
| Status | Integer | Zeigt an welcher stelle die Kette momentan ist |
| Status zurücksetzen | Skript | Setzt den Status zurück |

# TelegramBot

Benötigt: IP-Symcon >= 6.0
Erlaubt das Versenden von Nachrichten an angegebene Personen und Ausführen von Aktionen innerhalb von IP-Symcon.

## FUNKTIONSUMFANG

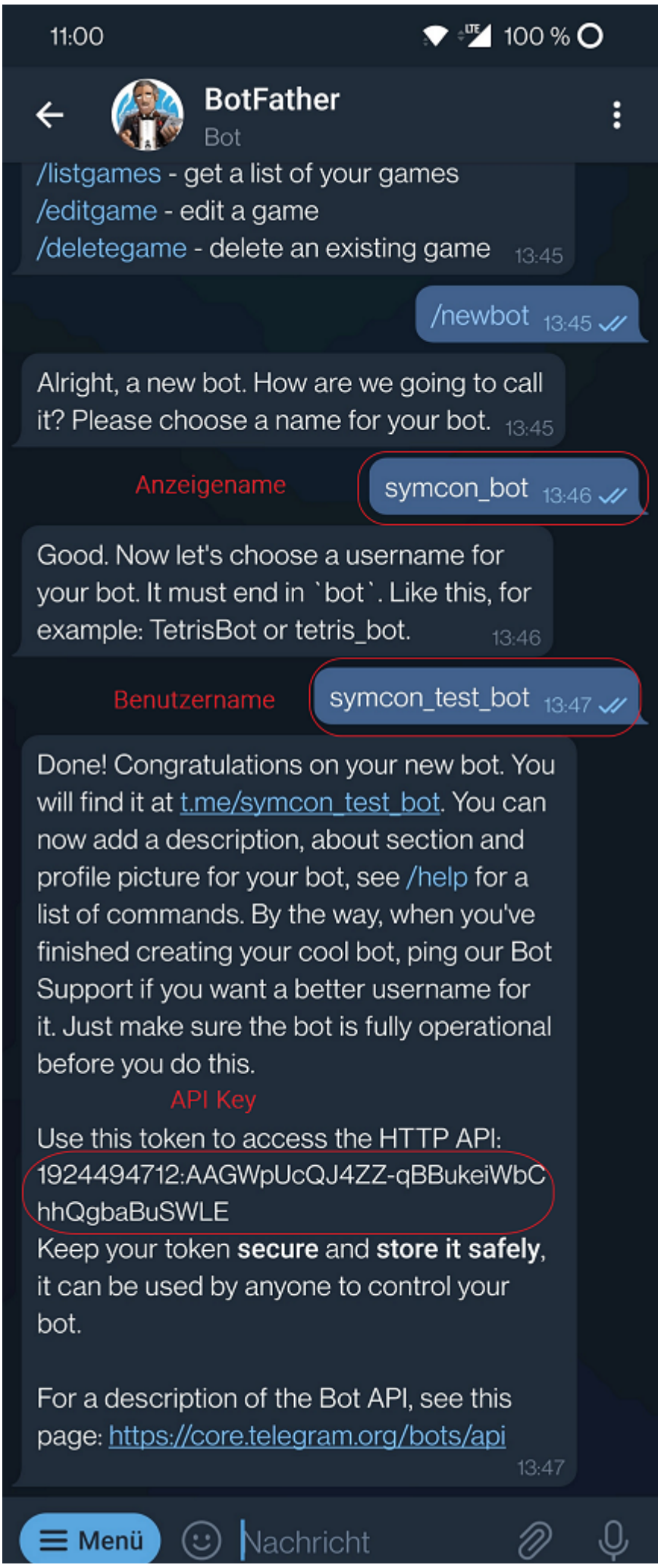
Nachrichten versenden (an alle oder einzelne Personen)
Bilder (aus Medien) versenden (an alle oder einzelne Personen)
Auf Chat-Kommandos mit Aktionen reagieren

## VORAUSSETZUNGEN

Ein Telegram Bot

## TELEGRAM BOT ERSTELLEN

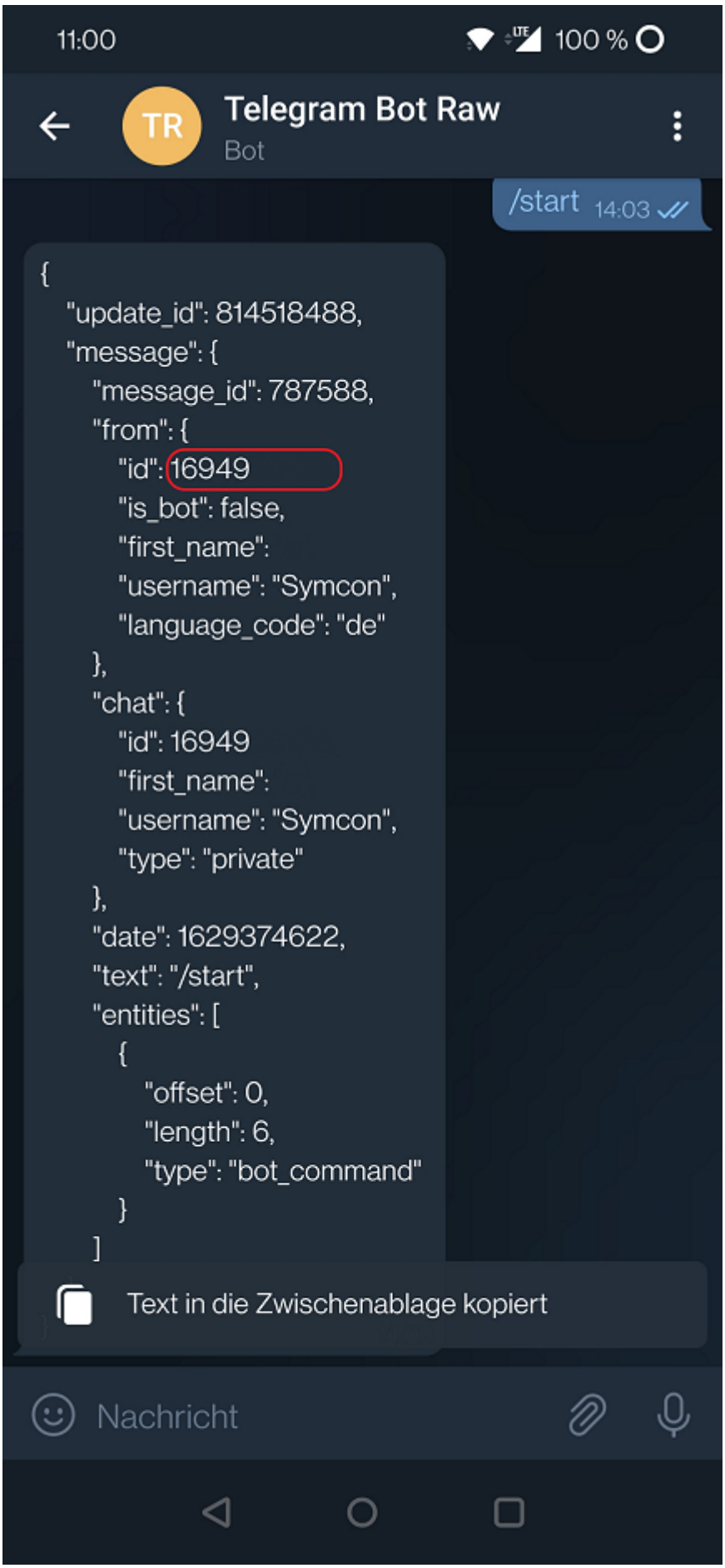
Im Telegram-Client nach BotFather suchen
Mit */newbot* wird ein neuer Bot erstellt
Unter dem Benutzernamen kann der Bot gefunden werden und sollte daher mit beispielsweise zufälligen Zahlen bestückt werden


11:00
100 %
BotFather
Bot
/listgames - get a list of your games
/editgame - edit a game
/deletegame - delete an existing game 13:45
/newbot 13:45
Alright, a new bot. How are we going to call it? Please choose a name for your bot. 13:45
Anzeigename
symcon_bot 13:46
Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot. 13:46
Benutzername
symcon_test_bot 13:47
Done! Congratulations on your new bot. You will find it at t.me/symcon_test_bot. You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.
API Key
Use this token to access the HTTP API:
1924494712:AAGWpUcQJ4ZZ-qBBukeiWbChhQgbaBuSWLE
Keep your token secure and store it safely, it can be used by anyone to control your bot.
For a description of the Bot API, see this page: https://core.telegram.org/bots/api
13:47
Menü
Nachricht



Im Telegram-Client nach RawDataBot suchen
Wenn der Bot angeschrieben wird gibt dieser einige Daten zurück, aus denen die eigene BenutzerID entnommen werden kann

Dieser Vorgang muss für jeden Benutzer wiederholt werden, der von dem Bot erreicht werden soll




Den Bot mit */start* anschreiben um ihn zu starten

## SOFTWARE-INSTALLATION

Über den Module Store das 'TelegramBot'-Modul installieren.
Alternativ über das Module Control folgende URL hinzufügen

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'TelegramBot'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Bot API Key | Der API Key des erstellten Bots |
| Bot Benutzername | Der Benutzername des erstellten Bots |
| Benutzer | Eine Liste von Benutzern, die vom Bot angeschrieben werden können |
| Aktionen | List mit Befehlen und den dazugehörigen Aktionen |

## AKTIONEN

### VERSCHICKE BILD

Mit dieser Aktion kann ein Bild an den ausgewählten Empfänger gesendet werden

#### *PARAMETER:*

Bild: Das Medienobjekt, welches versendet werden soll
Empfänger: Ein bestimmter oder alle in der Benutzer-Liste definierten Personen

### VERSCHICKE NACHRICHT

Mit dieser Aktion kann ein Nachricht an den ausgewählten Empfänger gesendet werden

#### *PARAMETER:*

Text: Der Inhalt der Nachricht
Empfänger: Ein bestimmter oder alle in der Benutzer-Liste definierten Personen
Mehr Informationen zur Nutzung von Aktionen

# Text to Speech

Das TTS-Modul kann Text in Sprache umwandeln. Dazu wird die systeminterne Speechsoftware oder eine Nachinstallierte genutzt.



Das TTS-Modul kann momentan nur in der Windows Installation genutzt werden. Alternativ kann das "Text to Speech (AWS Polly)" - Modul aus dem Module Store genutzt werden. Dies funktioniert auf allen Betriebsystemen.

## EINRICHTUNG IN IP-SYMCON

Im Konfigurationsreiter kann die Sprach-Engine und das Audio-Ausgabegerät eingestellt werden.
Um eine Sprachausgabe zu tätigen, muss der PHP Befehl TTS_Speak oder TTS_GenerateFile aus einem Script aufgerufen werden.



Der TTS_Speak Befehl ist veraltet und wird ab Windows Vista oder höher nicht mehr unterstützt. TTS_GenerateFile funktioniert in allen Versionen.

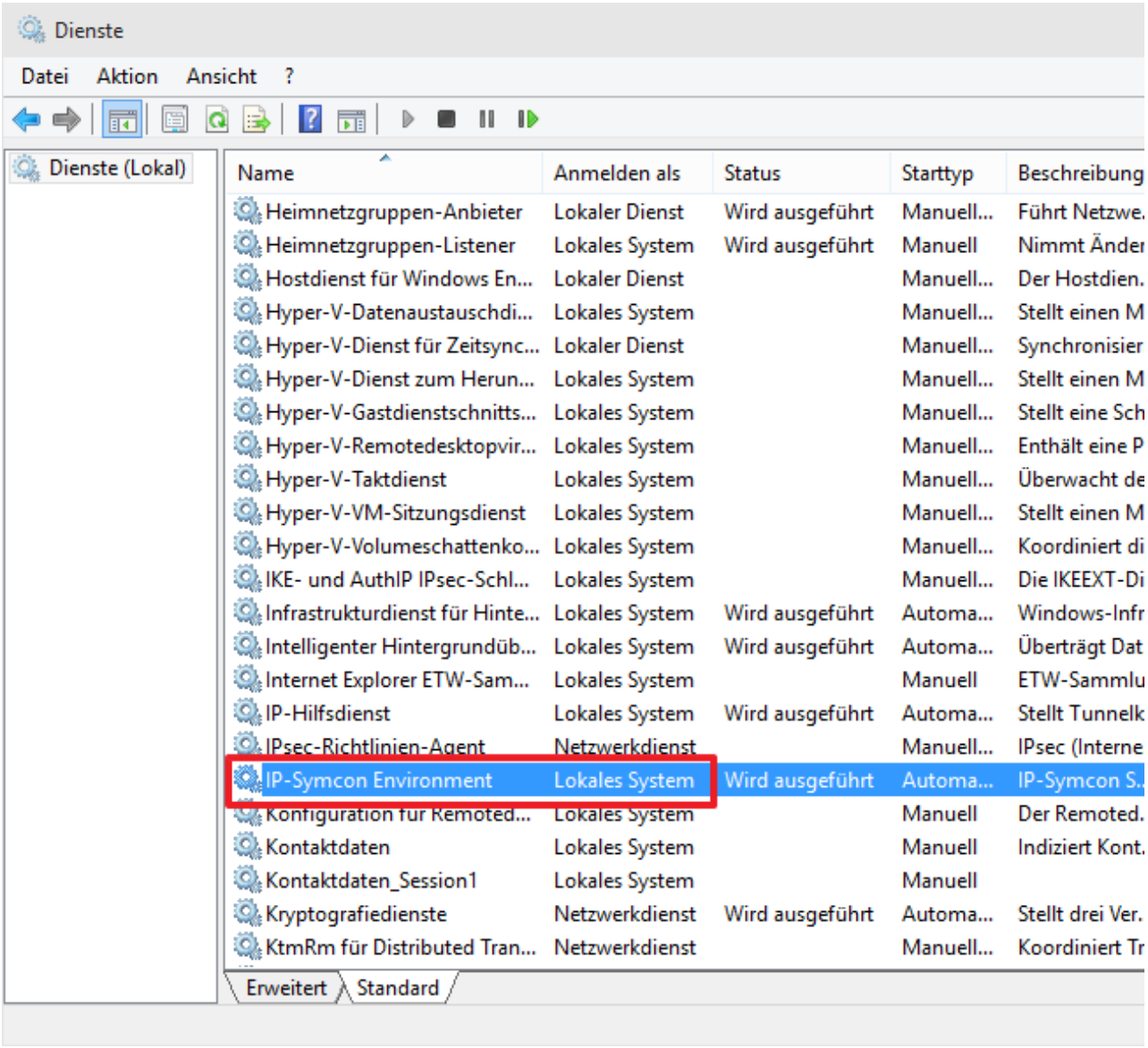
Bei der Nutzung von TTS_GenerateFile ist die Nutzung vom MediaPlayer nötig.
Ein Beispielcode ist unter TTS_GenerateFile zu finden.

## INSTALLATION

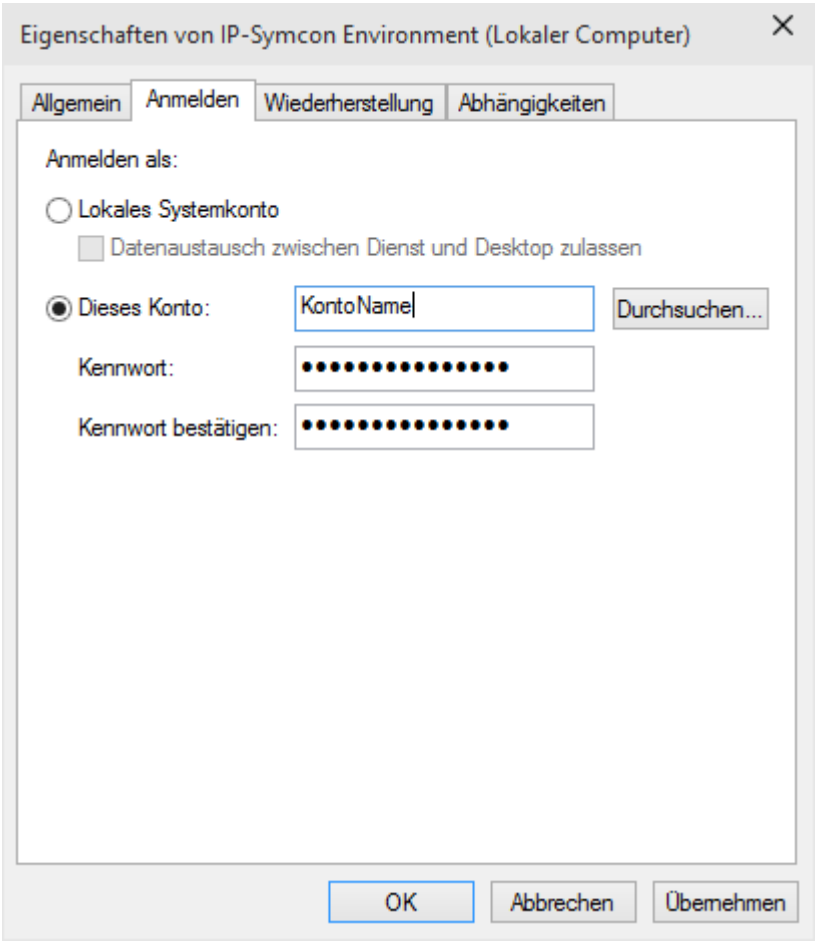
Neben möglichen Standardstimmen können hier weitere deutsche und internationale Ausgaben gefunden werden.

### BESONDERHEIT WINDOWS VISTA/7

Um die Sprachausgabe unter Windows Vista/7 nutzen zu können, muss der IP-Symcon Dienst als Lokaler Benutzer ausgeführt werden. Es ist zu beachten, dass dadurch die IPS_ExecuteEx Funktion nicht mehr zur Verfügung steht.
Über Start > Ausführen die Dienstverwaltung aufrufen
*%SystemRoot%\system32\services.msc /s*




Doppelklick auf den Eintrag IP-Symcon Enviroment. Es sollte sich folgender Dialog öffnen und im Reiter "Anmelden" muss das Passwort für den aktuellen Benutzer angegeben werden.




Nach Bestätigen mit Übernehmen/OK muss der IP-Symcon Dienst neu gestartet werden. Dieser ist am einfachsten über den Tray zu stoppen und starten.

# TTS_GenerateFile

```
boolean TTS_GenerateFile (int $InstanzID, string $Text, string
$Dateiname, int $Format)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Text-To-Speech Moduls |
| Text | Zu sagender Text |
| Dateiname | Dateiname/-pfad im System |

Format

| FORMAT | WERT |
|---|---|
| Default | 0 |
| 8kHz, 8Bit, Mono | 4 |
| 8kHz, 8Bit, Stereo | 5 |
| 8kHz, 16Bit, Mono | 6 |
| 8kHz, 16Bit, Stereo | 7 |
| 11kHz, 8Bit, Mono | 8 |
| 11kHz, 8Bit, Stereo | 9 |
| 11kHz, 16Bit, Mono | 10 |
| 11kHz, 16Bit, Stereo | 11 |
| 12kHz, 8Bit, Mono | 12 |
| 12kHz, 8Bit, Stereo | 13 |
| 12kHz, 16Bit, Mono | 14 |
| 12kHz, 16Bit, Stereo | 15 |
| 16kHz, 8Bit, Mono | 16 |
| 16kHz, 8Bit, Stereo | 17 |
| 16kHz, 16Bit, Mono | 18 |
| 16kHz, 16Bit, Stereo | 19 |
| 22kHz, 8Bit, Mono | 20 |
| 22kHz, 8Bit, Stereo | 21 |
| 22kHz, 16Bit, Mono | 22 |
| 22kHz, 16Bit, Stereo | 23 |
| 24kHz, 8Bit, Mono | 24 |
| 24kHz, 8Bit, Stereo | 25 |
| 24kHz, 16Bit, Mono | 26 |
| 24kHz, 16Bit, Stereo | 27 |
| 32kHz, 8Bit, Mono | 28 |
| 32kHz, 8Bit, Stereo | 29 |
| 32kHz, 8Bit, Stereo | 30 |
| 32kHz, 16Bit, Stereo | 31 |
| 44kHz, 8Bit, Mono | 32 |
| 44kHz, 8Bit, Stereo | 33 |
| 44kHz, 16Bit, Mono | 34 |
| 44kHz, 16Bit, Stereo | 35 |
| 48kHz, 8Bit, Mono | 36 |
| 48kHz, 8Bit, Stereo | 37 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Generiert über das Text-To-Speech Modul mit der ID **InstanzID** eine Sounddatei im Wav-Format in die Datei **Dateiname**. Der Format Parameter bestimmt die Qualität der Wav Datei.
*Warning: % wurde nicht gefunden in [...] on line …*
Sollte o.g. Fehlermeldung erscheinen, deutet es auf ein Zugriffsproblem hin. Wahrscheinlich wird die zu beschreibende Datei von einem anderen Prozess blockiert. Desweiteren ist es nicht möglich auf **Netzwerkpfade** zu schreiben.

## BEISPIEL

```
//Erstellt die Audiodatei
TTS_GenerateFile(16991 /*[Text To Speech]*/ , "Hallo Welt", "C:/ip-symcon-tts.wav", 39);
//Spielt die erstellte Audio-Datei ab
WAC_PlayFile(53825 /*[Media Player]*/ ,"C:/ip-symcon-tts.wav");
```

# TTS_Speak

```
boolean TTS_Speak (int $InstanzID, string $Text, boolean $Warten)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des Text-To-Speech Moduls |
| Text | Zu sagender Text |
| Warten | **TRUE** für An, **FALSE** für Aus |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Spricht über das Text-To-Speech Modul mit der ID **InstanzID** den in **Text** angegeben Text. Der **Warten** Parameter gibt an, ob gewartet werden soll, bis die Sprachausgabe beendet wurde.



Diese Funktion funktioniert nicht unter Windows **Vista/2003** oder neuer. Das Problem macht sich mit der Fehlermeldung: **OLE error 80045063**
Auszug MSDN : SPERR_NOT_ACTIVE_SESSION bemerkbar.

Als *Ersatz* sollte die Funktion TTS_GenerateFile im Zusammenhang mit dem Media Player genutzt werden.

## BEISPIEL

```
TTS_Speak(12345, "Hallo Welt!", true); //Warten bis zu Ende
gesprochen wurde
```

# TTSAWSPolly

Benötigt: IP-Symcon >= 5.1
Das Text to Speech (AWS Polly) Modul dient dazu Sounddaten-/dateien zu generieren, welche z.B. für Audio Notifications oder VoIP Ansagen genutzt werden können.

## FUNKTIONSUMFANG

Erstellen von Sounddaten, die z.B. in einer Medien Datei gespeichert werden können für eine Ausgabe über Audio Notifications
Erstellen von Sounddateien, die für eine Ausgabe im VoIP Modul verwendet werden können

## VORAUSSETZUNGEN

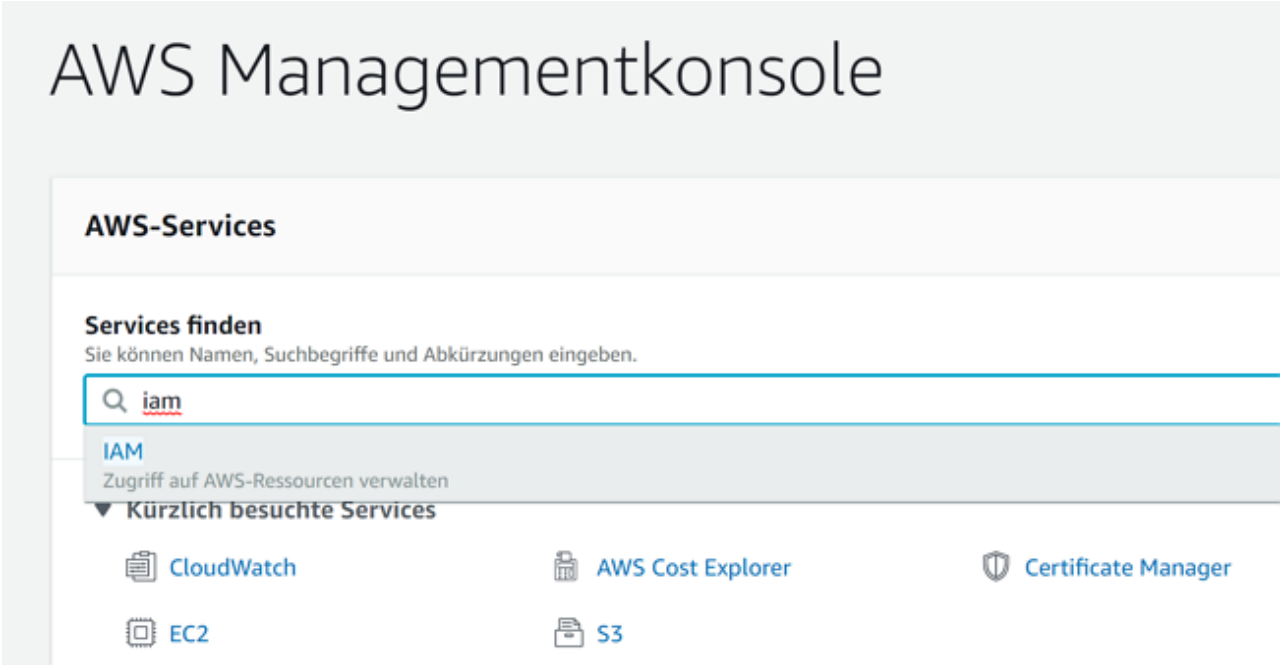
Konto bei Amazon Webservices
Benutzer mit passendem Access Key/Secret Key und Zugriffsrechten zu Polly (z.B. AmazonPollyFullAccess)

## SOFTWARE-INSTALLATION

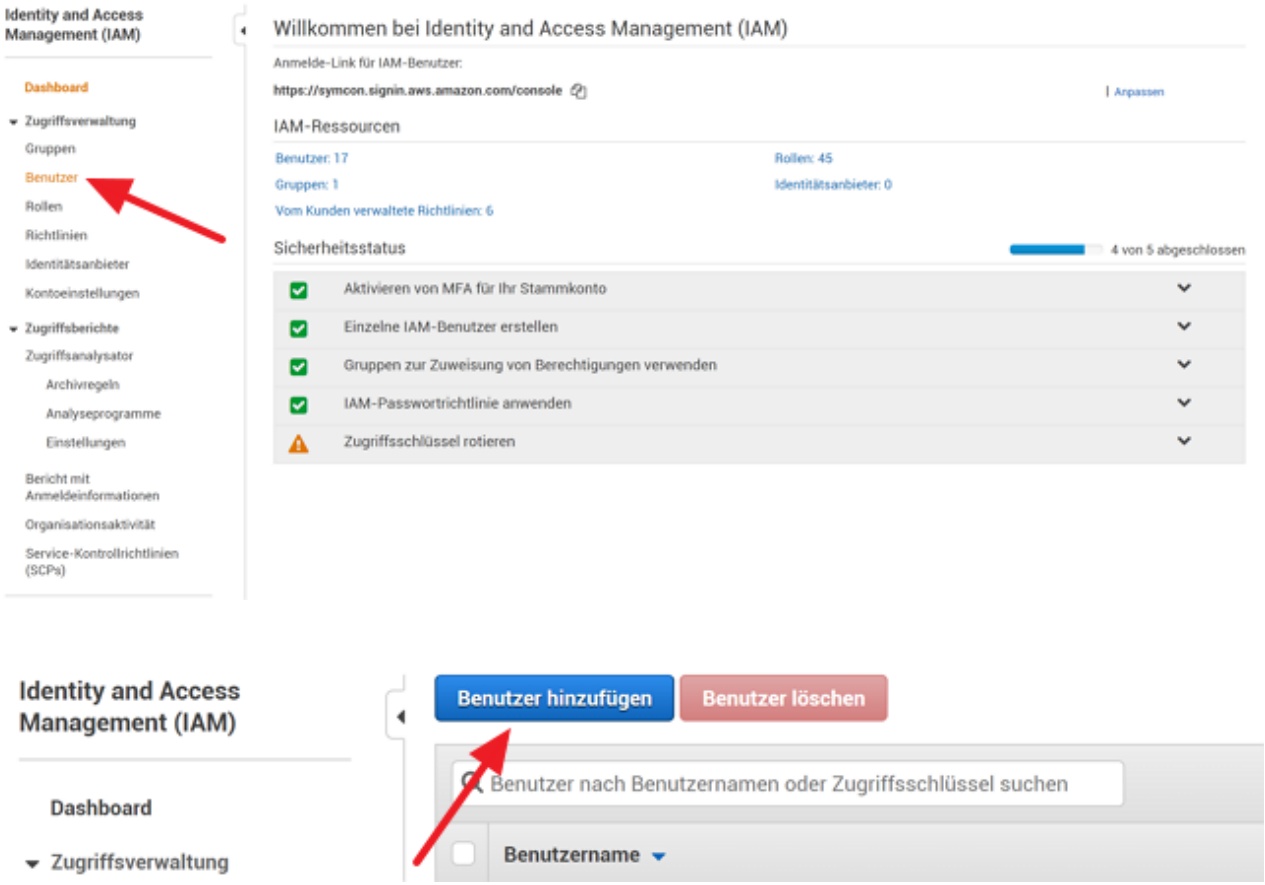
Das "Text to Speech (AWS Polly)" Modul kann direkt über den Module Store installiert werden.

## EINRICHTEN DES BENUTZERS IM AWS IAM

Auf der Amazon AWS Homepage muss bei den AWS Services nach dem IAM gesucht werden.



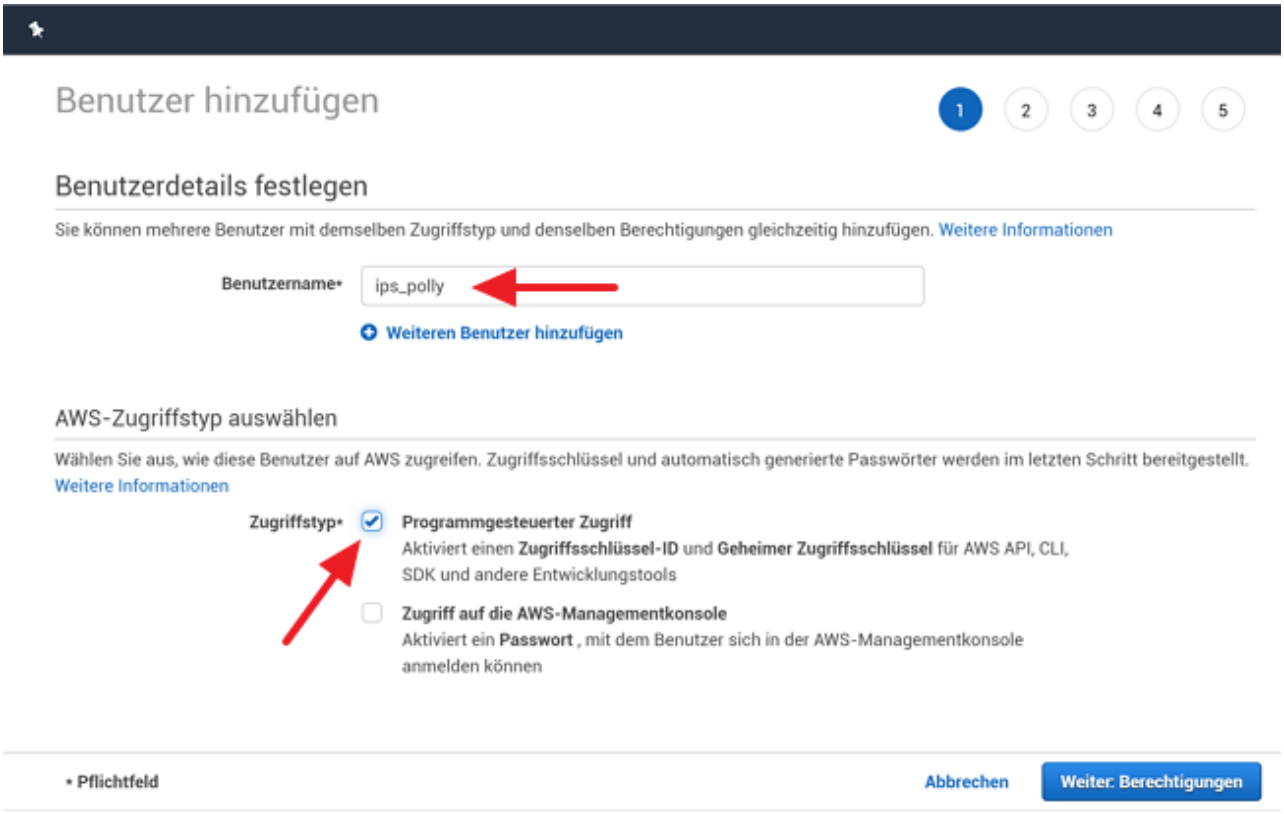

Innerhalb des IAM muss ein neuer Benutzer hinzugefügt werden



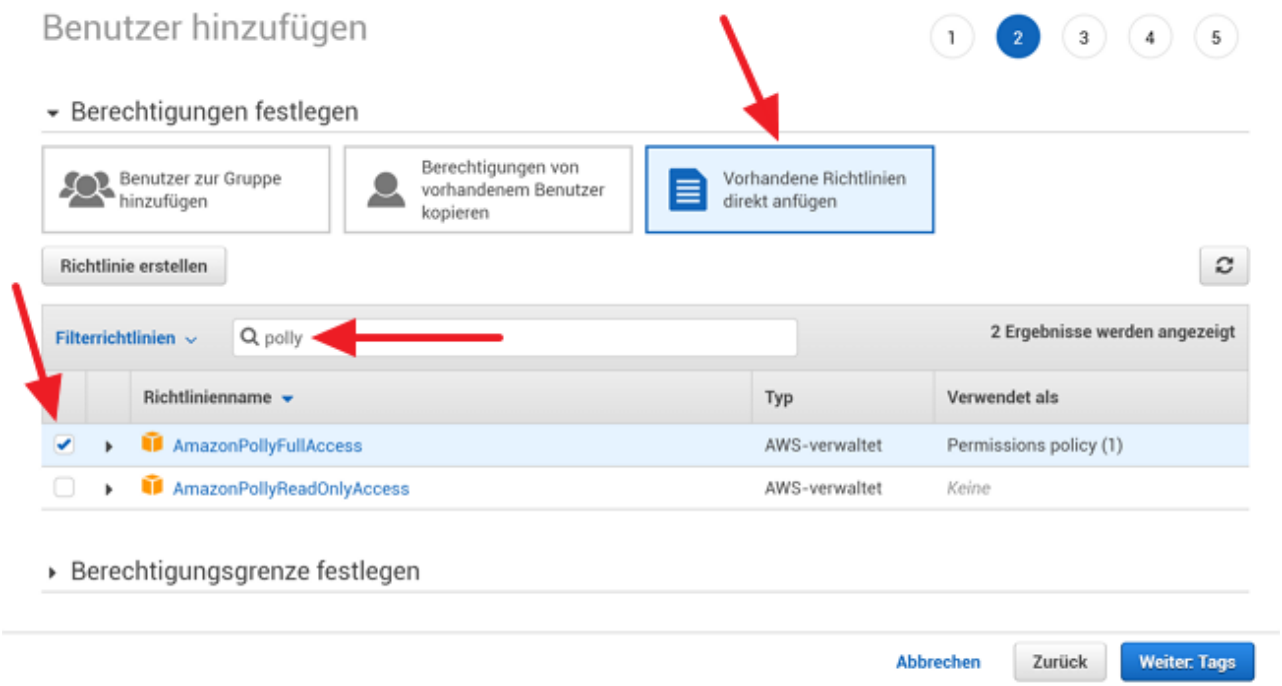




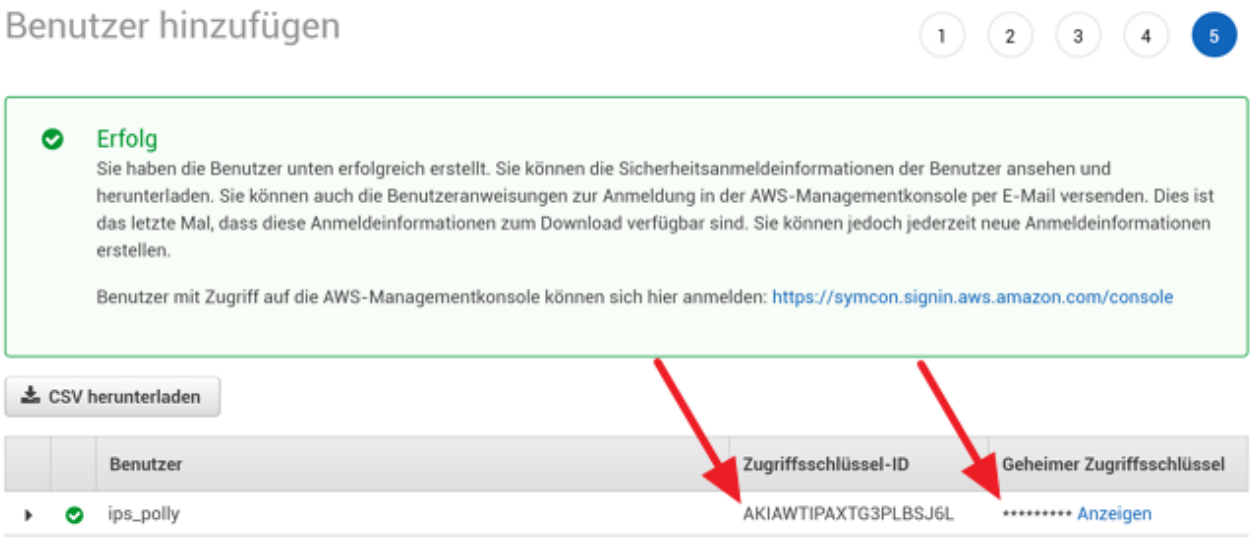
Anschließend sollte ein sprechender Benutzername vergeben und "Programmgesteuerter Zugriff" aktiviert werden.




Als Richtlinie muss "AmazonPollyFullAccess" aktiviert werden.




Den nächsten Dialog mit "Weiter: Prüfen" überspringen.
Den Benutzer legt man über "Benutzer erstellen" fertig an.
Nach dem Anlegen des Benutzers wird diesem eine ZugriffsschlüsselID und geheimer Zugriffsschlüssel zugewiesen.




## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Text to Speech (AWS Polly)'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
| --- | --- |
| Access Key | Access Key vom AWS Benutzer der Zugriff auf Polly hat |
| Secret Key | Secret Key vom AWS Benutzer der Zugriff auf Polly hat |
| Region | Region in der Polly genutzt werden soll |
| Sprache | Sprache in der die Ausgabe stattfinden soll |
| Ausgabeformat | Format (MP3/WAV/OGG) der Ausgabe |
| Abtastrate | Abtastrate (Standard, 8000 Hz, 16000 Hz, 22050 Hz) der Ausgabe |
| Texttyp | Typ vom Text. Bei SSML können die speziellen SSML Tags verwendet werden |

Nach Eintragung des Access Key/Secret Key muss die Konfiguration gespeichert werden, um die verfügbaren Sprachen zu Laden.

# TTSAWSPOLLY_GenerateData

Benötigt: IP-Symcon >= 5.1

```
string TTSAWSPOLLY_GenerateData (int $InstanzID, string $Text)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| Text | Sprachdaten |

## RÜCKGABEWERT

Base64-kodierte Sprachdaten

## BESCHREIBUNG

Fragt über AWS den Text an und liefert die Sprachdaten Base64-kodiert in der Rückgabe zurück.

## BEISPIEL

```
echo TTSAWSPOLLY_GenerateData(12345, "Dies ist ein Test");
```

# TTSAWSPOLLY_GenerateFile

Benötigt: IP-Symcon >= 5.1

```
string TTSAWSPOLLY_GenerateFile (int $InstanzID, string $Text)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| Text | Sprachdaten |

## RÜCKGABEWERT

Sprachdaten Dateiname

## BESCHREIBUNG

Fragt über AWS den Text an und liefert den Dateinamen zu den Sprachdaten zurück.

## BEISPIEL

```
echo TTSAWSPOLLY_GenerateFile(12345, "Dies ist ein Test");
```

# VerbrauchsAlarm

Benötigt: IP-Symcon >= 6.0
Das Modul dient dazu einen unnatürlich Verbrauch festzustellen. Es reagiert auf eine Zählervariable und schaltet unter bestimmten Bedingungen einen Alarm. Es gibt zwei Zustandsvariablen.
Einen Großverbrauch-Zustand, welcher schaltet wenn ein einstellbarer Grenzwert überschritten wird. Einen Kleinverbrauch-Zustand, welcher in 7 Stufen hochtickt wenn über längeren Zeitraum einen eingestellter Grenzwert (z.B. tropfender Wasserhahn) überschritten wird. Das Intervall für beide Kontrollen kann über die Konfiguration eingestellt werden. Ein Alarm, welche schaltet wenn der Großverbrauch-Zustand auf Alarm geschaltet ist, oder der Kleinverbrauch-Alarm einen eingestellten Wert überschreitet.

## FUNKTIONSUMFANG

Auswahl der Zählervariable
Groß-/Kleinverbrauch Timer in Minuten einstellbar
Groß-/Kleinverbrauch Grenzwert einstellbar
7 Stufen Anzeige für Kleinverbrauch
Alarmanzeige bei Großverbrauch

## SOFTWARE-INSTALLATION

Über den Module Store das Modul Verbraucher-Alarm installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'VerbrauchsAlarm'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Zählervariable | Variable, welche den Zählerwert wiedergibt. |
| Alarmsauslöser | Standard: 6 Kleinverbrauchsstufe bei dem der Alarm ausgelöst wird. |
| Kleinverbrauch Intervall | Standard: 1min Zeitintervall in dem kontrolliert wird, ob der Verbrauch zu hoch ist ist. |
| Kleinverbrauch Grenzwert | Standart: 0 Grenzwert bei dem der Zustand geändert wird. Ist dieser nicht gesetzt wird nicht darauf geprüft. |
| Großverbrauch Intervall | Standard: 5min Zeitintervall in dem kontrolliert wird, ob der Verbrauch zu hoch ist. |
| Großverbrauch Grenzwert | Standart: 0 Grenzwert bei dem der Zustand geändert wird.Ist dieser nicht gesetzt wird nicht darauf geprüft. |

## STATUSVARIABLEN UND PROFILE

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN:

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Alarmbenachrichtigung | Boolean | Alarm, wenn Großverbrauch geschaltet oder Kleinverbrauch den Wert überschreitet. |
| Kleinverbrauch Zustand | Integer | 7 Stufenanzeige für den Stand den Alarmlevels. |
| Großverbrauch Zustand | Boolean | Alarm ob der Durchfluss zu hoch ist. |

### PROFILE:

| BEZEICHNUNG | BESCHREIBUNG |
|---|---|
| VBA.LeakLevel | Profil für Kleinverbrauch - 7 Alarmstufen mit verschiedenen Symbolen und Farbanzeigen |
| VBA.ThresholdValue | Profil für Klein-/Großverbrauch Grenzwert |

Aufschlüsselung VBA.LeakLevel

| STUFE | WERT |
|---|---|
| Keine Aktivität | 0 |
| Alles im grünen Bereich | 1 |
| Normale Aktivität | 2 |
| Erhöte Aktivität | 3 |
| Unnormale Aktivität | 4 |
| Vor-Alarm | 5 |
| Alarm | 6 |

## VISUALISIERUNG

Über die Visualisierung können die Grenzwerte eingestellt werden.
Es wird zusätzlich angezeigt, ob ein Alarm vorliegt oder nicht.

# VBA_CheckAlert

Benötigt: IP-Symcon >= 6.0

```
void VBA_CheckAlert (int $InstanzID, string $BorderValue, string
$OldValue)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| BorderValue | Grenzwert |
| OldValue | Alter Wert |

## RÜCKGABEWERT

Die Funktion liefert keinerlei Rückgabewert.

## BESCHREIBUNG

Kontrolliert innerhalb des VerbraucherAlarms mit der InstanzID **InstanzID** ob ein Grenzwert überschritten wurde und setzt die Alarmvariablen

## BEISPIEL

```
VBA_CheckAlert(12345, "SmallUserThreashold", "SmallUserBuffer");
```

# Wasser-Alarm

Benötigt: IP-Symcon >= 4.2
Das Modul dient dazu einen unnatürlich hohen Wasserverbrauch festzustellen. Es reagiert auf eine Zählervariable und schaltet unter bestimmten Bedingungen einen Alarm. Es gibt zwei Alarmvariablen.
Einen Rohrbruch-Alarm, welcher schaltet wenn große Mengen auf einmal fließen. Einen Leckage-Alarm, welcher in 7 Stufen hochtickt wenn über längeren Zeitraum eine kleine Menge fließt (z.B. tropfender Wasserhahn). Das Intervall für beide Kontrollen kann über die Konfiguration eingestellt werden.

## FUNKTIONSUMFANG

Auswahl der Wasserzählervariable
Rohrbruch Timer in Minuten einstellbar
Leckage Timer in Minuten einstellbar
7 Stufen Anzeige für Leckage
Alarmanzeige bei Rohrbruch

## SOFTWARE-INSTALLATION

Über den Module Store das Modul Wasser-Alarm installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'WasserAlarm'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Zählervariable | Variable, welche den Zählerwert wiedergibt. |
| Leckage Intervall | Standard: 1min Zeitintervall in dem kontrolliert wird, ob zuviel Wasser durchgeflossen ist. |
| Rohrbruch Intervall | Standard: 15min Zeitintervall in dem kontrolliert wird, ob zuviel Wasser durchgeflossen ist. |

## STATUSVARIABLEN UND PROFILE

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### STATUSVARIABLEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Leckage Grenzwert | Float | Differenzwert, welcher im eingestellten "Leckage Intervall" überprüft wird und bei Überschreitung wird die "Leckage" um 1 erhöht. Wenn die Differenz unterhalb oder gleich der Grenze ist, wird das Alarmlevel resetet. |
| Leckage | Integer | 7 Stufenanzeige für den Stand den Alarmlevels. |
| Rohrbruch Grenzwert | Float | Grenzwert, welcher geprüft wird und gegebenfalls der Auslöser für einen Alarm ist. |
| Rohrbruch | Boolean | Alarm ob der Durchfluss zu hoch ist. |

### PROFILE:

| BEZEICHNUNG | BESCHREIBUNG |
|---|---|
| WAA.LeakLevel | Profil für Leckage - 7 Alarmstufen mit verschiedenen Symbolen und Farbanzeigen |
| WAA.ThresholdValue | Profil für Leckage/Rohrbruch Grenzwert |

## VISUALISIERUNG

Über die Visualisierung können die Grenzwerte eingestellt werden.
Es wird zusätzlich angezeigt, ob ein Alarm vorliegt oder nicht.

# WAA_CheckAlert

Benötigt: IP-Symcon >= 4.2

```
boolean WAA_CheckAlert (int $InstanzID, string $BorderValue, string
$OldValue)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| BorderValue | Grenzwert |
| OldValue | Alter Wert |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Kontrolliert innerhalb des WasserAlarms mit der InstanzID InstanzID ob ein Grenzwert überschritten wurde und setzt die Alarmvariablen

## BEISPIEL

```
WAA_CheckAlert(12345, "LeakThreashold", "LeakBuffer");
```

# Watchdog

Benötigt: IP-Symcon >= 5.1
Überprüft ob in einer Liste definierte Variablen überfällig sind. Sind Variablen überfällig, wird ein Alarm gesetzt und eine Liste dieser in der Visualisierung angezeigt.

## FUNKTIONSUMFANG

Überwachen von gelisteten Variablen.
Einstellbarkeit ob die Variablen auf Änderung oder Aktuallisierung geprüft werden sollen.
Einstellbarkeit wie lange die gelisteten Variablen überfällig sein dürfen.
Ein-/Ausschaltbarkeit via Visualisierungsbutton oder Skript-Funktion.
Anzeige wann die gelisteten Variablen zuletzt überprüft wurden.
Darstellung des Originalpfades oder eines inndividuellen Namens.

## SOFTWARE-INSTALLATION

Über den Module Store das Modul Watchdog installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Watchdog'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

Alle zu schaltenden Variablen müssen der Liste "Variablen" in der Instanzkonfiguration hinzugefügt werden.

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Variablen | Dieser Liste werden die zu beobachtenden Variablen hinzugefügt. |
| Zeit | Dauer der Inaktivität bis die gelisteten Variablen den Alarm auslösen. |
| Einheit | Einheit der Zeit. |

Sollten Variablen unterschiedlich lange erlaubt Inaktivität haben, ist es zu empfehlen mehrere Instanzen des Watchdog-Moduls zu verwenden.

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

| NAME | TYP | BESCHREIBUNG |
| --- | --- | --- |
| Aktive Alarme | String | Beinhaltet die Tabelle für die Darstellung in der Visualisierung. |
| Alarm | Boolean | Die Variable zeigt an ob ein Alarm vorhanden ist. True = Alarm; False = OK; |
| Letzte Überprüfung | Integer | UnixTimestamp der den Zeitpunkt angibt zu dem zuletzt überprüft wurde. |
| Watchdog aktiv | Boolean | Zeigt an ob der Watchdog aktiviert ist oder nicht. True = Aktiviert; False = Deaktiviert; |

## VISUALISIERUNG

Über die Visualisierung kann der Watchdog de-/aktiviert werden.
Es wird zusätzlich die Information angezeigt, zu welchem Zeitpunkt zuletzt überprüft wurde.

# WD_GetAlertTargets

Benötigt: IP-Symcon >= 5.1

```
array WD_GetAlertTargets (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Array der aktiven Alarme der Watchdoginstanz mit der InstanzID **InstanzID**

## BESCHREIBUNG

Die Funktion liefert ein Array mit den aktiven Alarmen der Watchdoginstanz mit der InstanzID **InstanzID**.
Die Funktion liefert ein Array mit überfälligen Objekten. Es beinhaltet den optionalen Namen, VariablenID und den letzten Zeitpunkt (UnixTimestamp) des Updates oder der Änderung.

## BEISPIEL

```
WD_GetAlertTargets(12345);
```

# WD_SetActive

Benötigt: IP-Symcon >= 5.1

```
boolean WD_SetActive (int $InstanzID, boolean $SetActive)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| SetActive | Aktiviert/ Deaktiviert den Watchdog |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

SetActive aktiviert (true) oder deaktiviert (false) den Watchdog mit der InstanzID InstanzID.

## BEISPIEL

```
WD_SetActive(12345, true);
```

# Kern Instanzen

Auch im Kern von Symcon gibt es Module, welche grundsätzliche Funktionalitäten bieten.

| | |
|---|---|
| Archive Control | Funktionen zur Verwaltung des Datenloggings |
| AC_AddLoggedValues | fügt einer geloggten Variable weitere Datensätze hinzu |
| AC_ChangeVariableID | migriert Daten einer Variable in eine bisher nicht geloggte Variable |
| AC_DeleteVariableData | löscht in einem bestimmten Zeitraum alle Datensätze einer geloggten Variable |
| AC_GetAggregatedValues | holt aggregierte Daten aus dem Archiv |
| AC_GetAggregationType | gibt den Aggregationstyp einer Variable wieder |
| AC_GetAggregationVariables | gibt ein Array aller geloggten Variablen zurück |
| AC_GetCounterIgnoreZeros | holt den Status ob Nullen und negative Werte ignoriert werden |
| AC_GetGraphStatus | fragt ab ob eine Variable visualisiert wird |
| AC_GetLoggedValues | holt Rohdaten aus dem Archiv Control |
| AC_GetLoggingStatus | fragt ab ob eine Variable geloggt wird |
| AC_ReAggregateVariable | startet die Reaggregation einer Variable |
| AC_SetAggregationType | setzt den Aggregationstyp einer Variable |
| AC_SetCompaction | Konfiguriert die Verdichtung der Variable |
| AC_SetCounterIgnoreZeros | setzt den Status ob Nullen und negative Werte ignoriert werden |
| AC_SetGraphStatus | setzt die Eigenschaft zur Visualisierung einer Variable |
| AC_SetLoggingStatus | setzt die Eigenschaft zum Loggen einer Variable |
| Datenformat | |
| Berechtigungssteuerung | Benutzerverwalung und Synchronisierung mit LDAP Server |
| Calendar Control | Kalender Modul |
| Connect Control | Verbindungsmodul um eine gesicherte Verbindung von außerhalb zu ermöglichen |
| Cutter | |
| Cutter_ClearBuffer | löscht den Buffer der Cutter Instanz |
| DNS-SD Control | verwaltet eingerichtete Bonjour Services |
| ZC_QueryService | fragt Informationen eines einzelnen Services ab |
| ZC_QueryServiceType | fragt Informationen von Services eines bestimmten Types ab |
| ZC_QueryServiceTypes | Fragt alle eingerichteten Services ab |
| Event Control | Modul um Skripte bei festgesetzten Ereignissen zu starten |
| Location Control | berechnet Dämmerungszeitpunkte anhand von Längen- und Breitengrad |
| Module Control | verwaltet externe und hinzugeladene PHP-Module |
| Notification Control | ein Modul zum verschicken von Push-Notifications |
| Presence Control | Funktionen zur Detektion von Anwesenheit in Bereichen |
| PC_Enter | setzt die Anwesenheit auf true |
| PC_Leave | setzt die Anwesenheit auf false |

# Archive Control

Das Archive Control Modul kümmert sich um das Speichern der Variablenwerte in einer Datenbank.

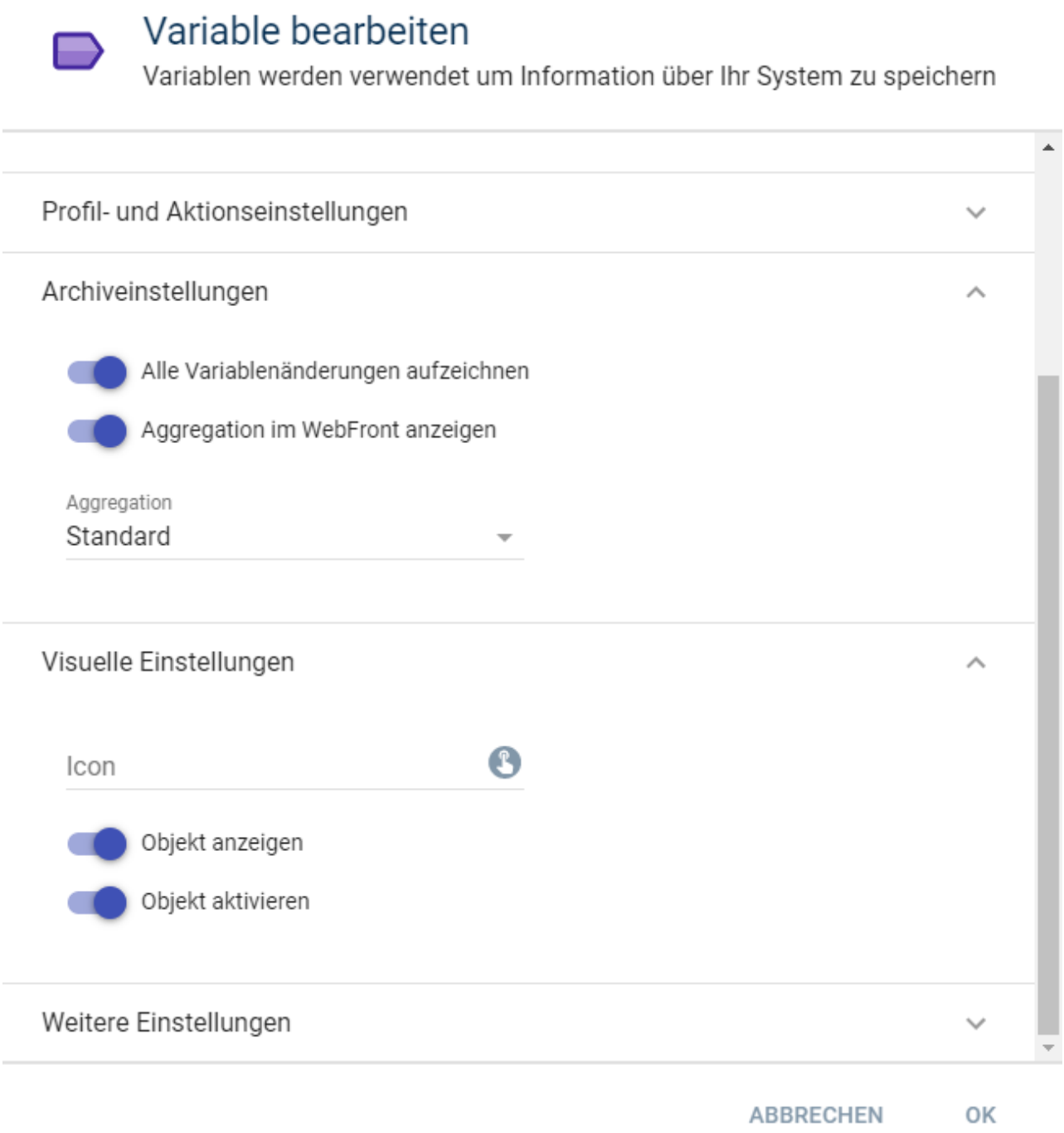
## SPEICHERART

Zur leichteren Handhabe und besseren Verwaltbarkeit wurde ab Version 4.0 die SQLite Datenbank entfernt und stattdesssen CSV eingeführt.

| IPS VERSION | SPEICHERART |
|---|---|
| bis Version 3.4 | SQLite Datenbank |
| ab Version 4.0 | CSV Dateien |

## LOGGING AKTIVIEREN

Um eine Variable in der Datenbank zu loggen, muss per Doppelklick das Fenster "Variable editieren" geöffnet und die Checkbox "Alle Variablenänderungen aufzeichnen" ausgewählt werden. Sofern kein Button für den Graphen dieser Variable in der Visualisierung angezeigt werden soll, kann im Dropdown-Menü "Darstellung" die Option "In Visualisierung verstecken" ausgewählt werden.




| AGGREGATIONSTYP | BESCHREIBUNG |
|---|---|
| Standard | Für alle Aggregationsstufen (Stunde, Tag, Woche, Monat, Jahr) wird Min / Max / Durchschnitt berechnet. |
| Zähler | Für alle Aggregationsstufen (Stunde, Tag, Woche, Monat, Jahr) wird das Delta (der Unterschied) der Werte als Min / Max / Summe berechnet. |

Sofern die Zähleraggregation ausgewählt wurde, wird der erste im Archiv geloggte Wert als Referenzwert genutzt und erst die darauf folgenden Werte zur Summierung der positiven Deltas genutzt. Das heißt eine Variable mit den geloggten Werten 500, 505, 520, 521 wird mit den Werten 0, 5, 15, 1 angezeigt.

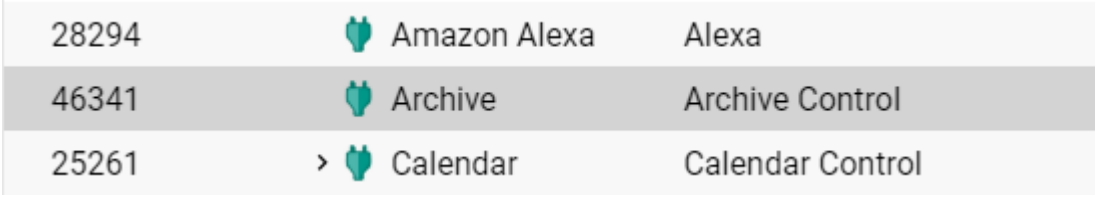
| ARCHIVEINSTELLUNGEN | BESCHREIBUNG |
|---|---|
| Alle Variablenveränderungen aufzeichnen | Aktiviert das Logging der Variable. |
| Aggregation in Visualisierung anzeigen | Deaktiviert/Versteckt den Button zur Visualisierung der geloggten Variable. Das Aufzeichnen der Datenpunkte der Variable ist immernoch aktiv. |

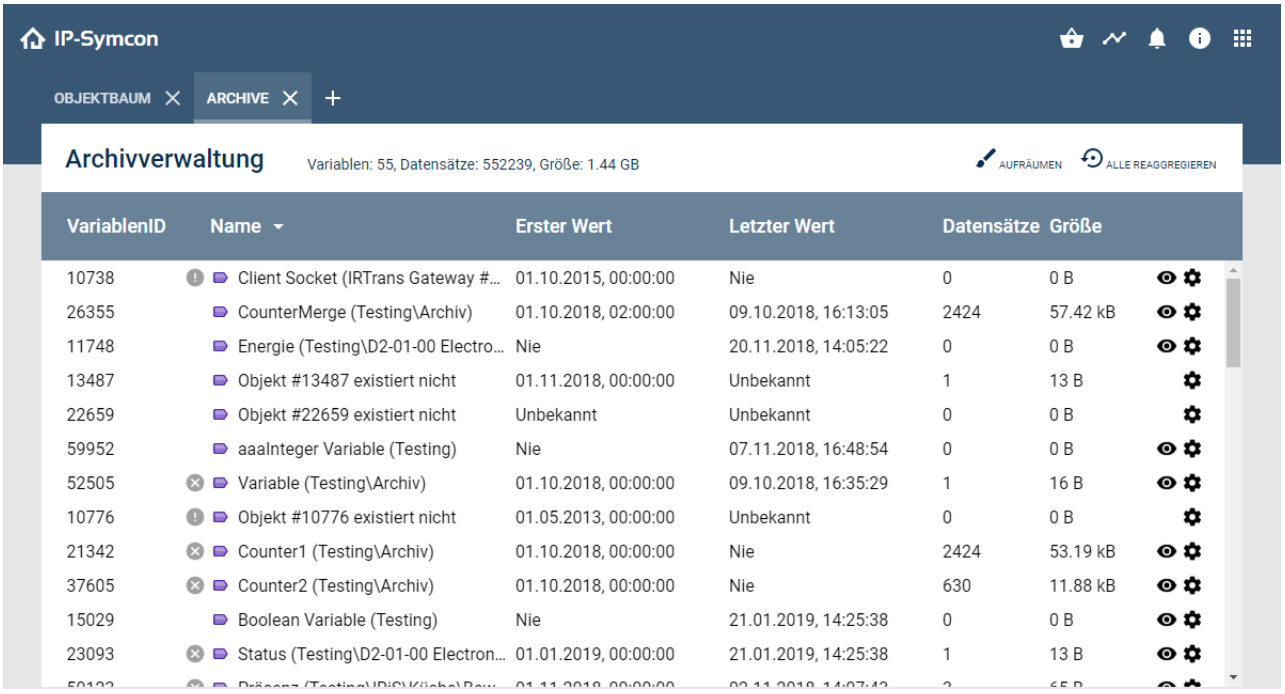
Wenn der Aggregationstyp einer Variable verändert wird, wird diese automatisch neu aggregiert. Dies kann einige Zeit in Anspruch nehmen.

Sobald das Logging ausgewählt ist, werden alle Daten in der Datenbank abgespeichert. Dabei werden die Daten zweifach gespeichert. Alle Rohdaten werden in der Datenbank abgelegt und diese Daten werden wiederum auf die einzelnen Intervalle (Tag, Woche, Monat, Jahr) zusammengefasst abgespeichert, um ein schnelleres erstellen der Graphen zu ermöglichen. Die Graphen können, wenn für die Variable aktiviert, über die Visualisierung angesehen werden.

## ARCHIVE HANDLER

Der Archive Handler ist eine Funktion zur Verwaltung aller geloggten Variablen. Es können alle Werte betrachtet und bearbeitet werden.

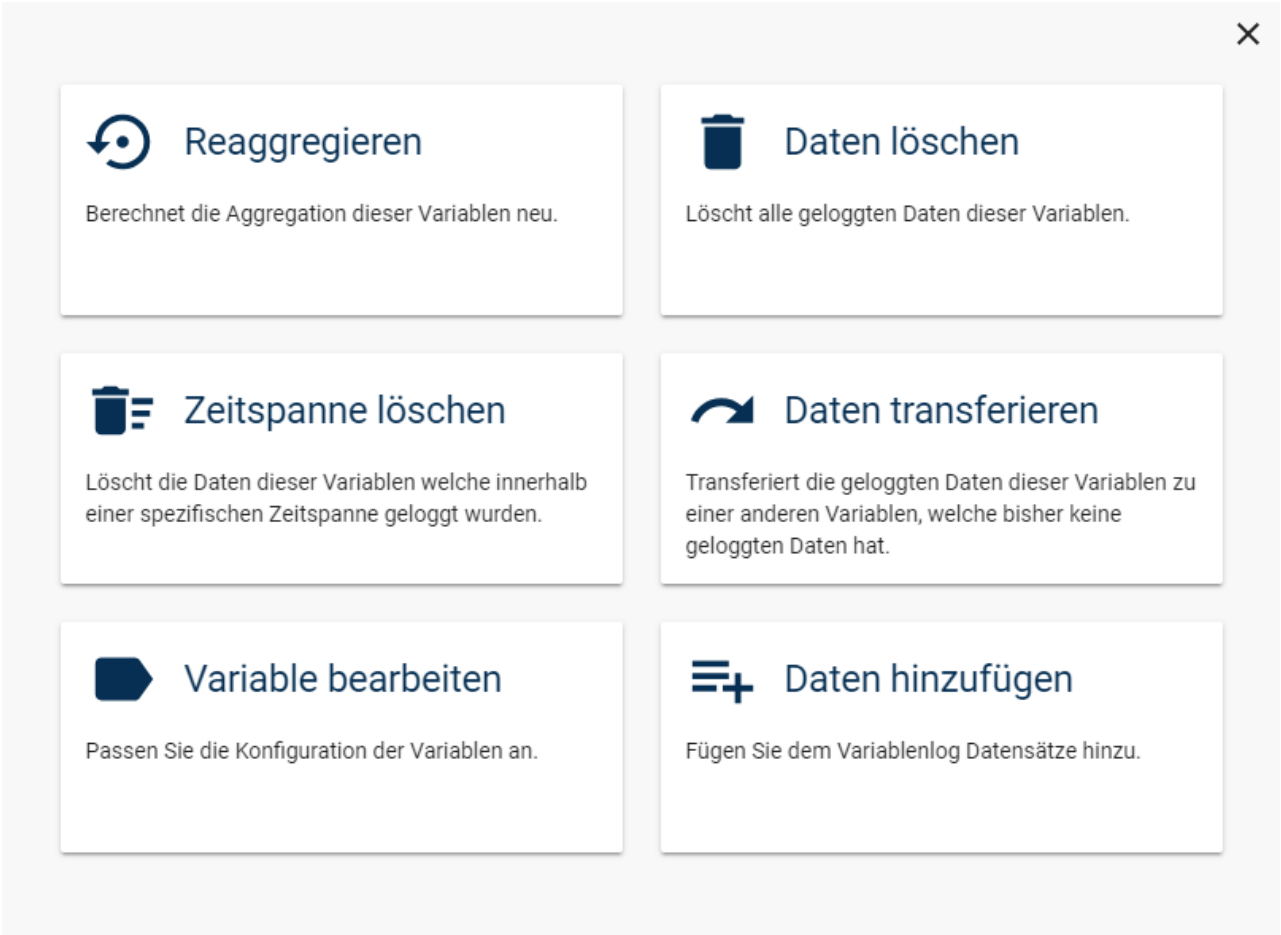



Die Titelzeile bietet folgende Funktionen:

| FUNKTION | BESCHREIBUNG |
| --- | --- |
| Aufräumen | Entfernt alle Archivdaten zu nicht mehr existierenden Variablen. Variablen deren Logging nur deaktiviert ist, werden nicht gelöscht. |
| Alle reaggregieren | Aggregiert alle geloggten Variablen neu. Dies kann je nach Umfang sehr lange dauern. Regelmäßige Status-, sowie Abschlussmeldungen, werden im Reiter "Meldungen" angezeigt. |

Ein Klick auf das Auge-Icon öffnet die Ansicht mit gespeicherten Rohdaten der gewählten Variable an und erlaubt das Löschen von einzelnen Datensätzen. **Nach dem Löschen ist es erforderlich, die Variablendaten neu zu aggregieren.** Diese Funktion steht nicht für Variablen zur Verfügung, welche nicht mehr existieren. Ein Klick auf das Zahnrad öffnet einen Dialog mit weiterführenden Funktionen zur ausgewählten Variablen



| FUNKTION | BESCHREIBUNG |
|---|---|
| Reaggregieren | Aggregiert die ausgewählte Variable neu. Dies kann je nach Umfang sehr lange dauern. Regelmäßige Status-, sowie Abschlussmeldungen, werden im Reiter "Meldungen" angezeigt. |
| Daten löschen | Löscht alle Rohdaten der gewählten Variable aus dem Archiv. |
| Zeitspanne löschen | Hier kann eine Zeitspanne ausgewählt werden, in der alle Datensätze der gewählten Variable gelöscht werden sollen. **Nach dem Löschen ist es erforderlich, die Variablendaten neu zu aggregieren.** |
| Daten transferieren | Die Rohdaten einer Variable werden auf eine andere Variable übertragen. Dies kann Sinnvoll sein wenn eine VariablenID nicht mehr existiert oder ein Sensor ausgetauscht wurde. Hierfür muss eine Variable gewählt werden, welche bisher nicht geloggt wurde und vom Variablentyp her zu den geloggten Werten passt. |
| Variable bearbeiten | Öffnet den Variable bearbeiten Dialog im Objektbaum |
| Daten hinzufügen | Hier können der Variablen weitere Datensätze hinzugefügt werden. Datensätze können entweder manuell oder aus Dateien hinzugefügt werden. Bei manuellem Hinzufügen öffnet sich ein weiter Dialog, in welchem in einer Liste Zeiten und Werte eingetragen werden können, welche bei einem Klick auf "Werte hinzufügen" der Variablen hinzugefügt werden. Sollen Datensätze aus Dateien hinzugefügt werden, so sind hierfür vorbereitete csv-Dateien erforderlich. In einem weiteren Schritt können die Werte noch manuell geprüft und schließlich der Variablen hinzugefügt werden. |

## MÖGLICHE DATEIFORMATE BEI "DATEN HINZUFÜGEN"

In .csv-Dateien für das Hinzufügen von Daten muss je Zeile ein Zeitpunkt zusammen mit einem Datenwert notiert sein. Hier sind zwei grundlegende Notationen möglich:

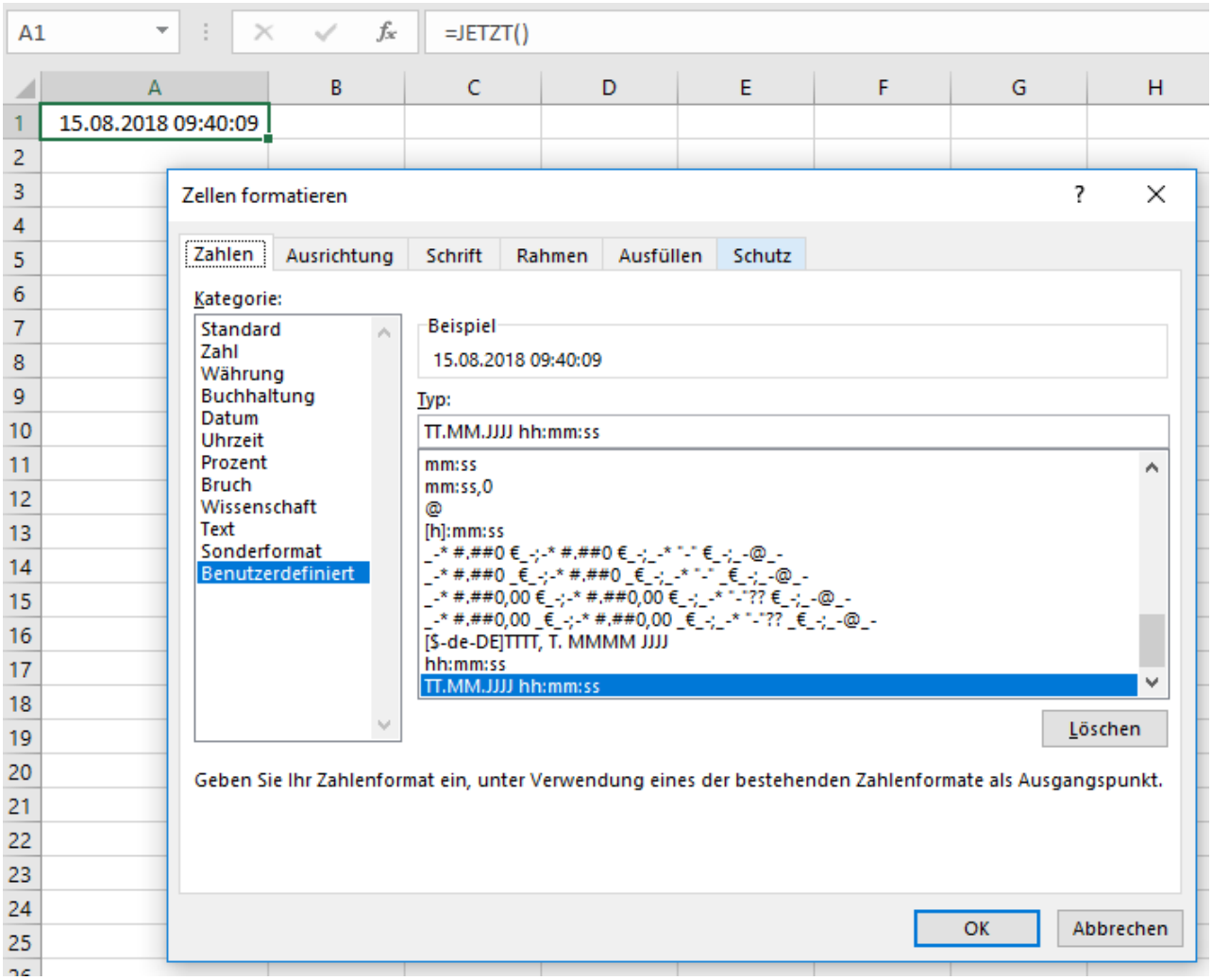
Komma als Trennzeichen und Punkt als Dezimaltrennzeichen (Datum,1.5)
Semikolon als Trennzeichen und Komma als Dezimaltrennzeichen (Datum;1,5)

Für das Datum werden mehrere Formate unterstützt:

Unix-Timestamp, z.B. 1522527010
Deutsches Format, z.B. 31.03.2018 22:10:10
Amerikanisches Format, z.B. 03/31/2018 10:10:10 PM
RFC2822, z.B. Sat, 31 Mar 2018 22:10:10 +0000
ISO8601, z.B. 2018-03-31 22:10:10

## DATEIEN AUS EXCEL-DATEI HINZUFÜGEN

Es ist möglich die .csv-Datei für das Hinzufügen von Daten mit Microsoft Excel vorzubereiten. Hier werden die Zeitpunkte in der ersten Spalte und die Werte in der zweiten Spalte eingetragen. Hierbei ist zu beachten, dass die Zeitpunkte in einem passenden Format formatiert sind. Beispielsweise kann für das deutsche Format das benutzerdefinierte Zahlenformat TT.MM.JJJJ hh:mm:ss verwendet werden.




Abschließend kann die Liste als .csv-Datei abgespeichert werden, indem bei "Speichern unter" der Typ "CSV (Trennzeichen-getrennt)" ausgewählt und danach mit "Speichern" bestätigt wird. Die generierte .csv-Datei kann dann über "Daten hinzufügen" einer Variable bei IP-Symcon hinzugefügt werden.

# AC_AddLoggedValues

Benötigt: IP-Symcon >= 5.1

```
boolean AC_AddLoggedValues (int $InstanzID, int $VariableID, array
$Datensätze)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID zum Archiv |
| VariableID | ID der Variable, welcher die Datensätze hinzugefügt werden sollen |
| Datensätze | Ein Array mit Datensätzen, wobei jeder Datensatz ein Array mit folgenden **key => value** Paaren ist: |

| PARAMETER | TYP | BESCHREIBUNG |
|---|---|---|
| TimeStamp | Integer | Zeitpunkt des neuen Datensatzes als Unix Timestamp |
| Value | nach Variable | Der Wert des neuen Datensatzes - Der Typ des Wertes muss hierbei zum Typen der Variable passen |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion fügt den geloggten Rohdaten der Variable mit der ID **VariableID** die Datensätze **Datensaetze** hinzu.



Nach dieser Funktion sind die aggregierten Daten der Variable ungültig und müssen neu aggregiert werden.

## BEISPIEL

```
// Füge einer Integer Variable drei zusätzliche Datensätze hinzu
AC_AddLoggedValues(12345, 34567, [
[
'TimeStamp' => 1128255120,
```

```
'Value' => 30
],
[
'TimeStamp' => 1128257851,
'Value' => 50
],
[
'TimeStamp' => 1128278514,
'Value' => 130
]
]);
```

# AC_ChangeVariableID

Benötigt: IP-Symcon >= 3.0

```
boolean AC_ChangeVariableID (int $InstanzID, int $OldVariableID, int
$NewVariableID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID zum Archiv |
| OldVariableID | ID der zu migrierenden Variable |
| NewVariableID | ID der neuen Variable |

## RÜCKGABEWERT

**True** wenn erfolgreich, sonst **False**

## BESCHREIBUNG

Diese Funktion migriert die Daten der Variable mit der ID **OldVariableID** in die Variable mit der ID **NewVariableID**.



Die Variable mit der ID **NewVariableID** darf nicht geloggt sein. Das heißt auch, dass keine alten Datensätze durch Logging vorhanden sein dürfen.

## BEISPIEL

```
// Die Daten der Variable 34567 werden migriert in die Variable
56789
AC_ChangeVariableID(12345, 34567, 56789);
```

# AC_DeleteVariableData

Benötigt: IP-Symcon >= 3.0

```
int AC_DeleteVariableData (int $InstanzID, int $VariablenID, int
$Startzeit, int $Endzeit)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID zum Archiv |
| VariablenID | ID der Variable in der die Datensätze gelöscht werden sollen |
| Startzeit | Datum/Zeit als Unix Zeitstempel (0 = von Anfang an) |
| Endzeit | Datum/Zeit als Unix Zeitstempel (0 = bis jetzt) |

## RÜCKGABEWERT

gibt die Anzahl der gelöschten Datensätze zurück

## BESCHREIBUNG

Diese Funktion löscht alle geloggten Datensätze der Variable mit der ID **VariablenID** im Zeitraum von **Startzeit** bis inklusive **Endzeit**. Sollte Die Startzeit als auch die Endzeit = 0 sein, wird zusätzlich das Logging der Variable deaktiviert.



Falls Datenssätze zu einem einzelnen Zeitpunkt gelöscht werden sollen, muss die Endzeit gleich der Startzeit gesetzt werden



Es ist zu bedenken, dass der Unix Zeitstempel auf UTC+0 basiert. Viele europäische Länder haben aber Sommerzeit (CEST/UTC+2) und Winterzeit (CET/UTC+1).



Nach Ausführung ist die Aggregation der Variable ungültig und muss erst reaggregiert werden. Dazu kann AC_ReaggregateVariable genutzt werden.



Werden die Parameter Start- und Endzeit = 0 gesetzt, werden die Datensätze der Variable restlos gelöscht.

## BEISPIEL

```
// Löscht alle Daten der Variable "TestVariable" im Zeitraum
07.10.2015 16:00 UTC+2(CEST) bis 07.10.2015 17:00 UTC+2(CEST)
AC_DeleteVariableData(12345 /*[Archive]*/, 45678 /*[Variable]*/,
1444226400, 1444237200);

// Löscht alle Daten der Variable "TestVariable" im Zeitraum
07.10.2015 16:00 UTC+2(CEST) bis jetzt
AC_DeleteVariableData(12345 /*[Archive]*/, 45678 /*[Variable]*/,
1444226400, 0);

// Restlos löschen und aus dem Archiv Control entfernen
// Löscht alle Daten der Variable und deaktiviert die Variable im
Archive Control
AC_DeleteVariableData(12345 /*[Archive]*/, 45678 /*[TestVariable]*/,
0, 0);
```

# AC_GetAggregatedValues

Benötigt: IP-Symcon >= 3.0

```
array AC_GetAggregatedValues (int $InstanzID, int $VariablenID, int
$Aggregationsstufe, int $Startzeit, int $Endzeit, int $Limit)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID zum Archiv |
| VariablenID | ID der abzufragenden Variablen |
| Aggregationsstufe | <table><tr><th>AGGREGATIONSSTUFE</th><th>BESCHREIBUNG</th></tr><tr><td>0</td><td>Stündliche Aggregation (00:00 - 59:59)</td></tr><tr><td>1</td><td>Tägliche Aggregation (00:00:00 - 23:59:59)</td></tr><tr><td>2</td><td>Wöchentliche Aggregation (Montag 00:00:00 - Sonntag 23:59:59)</td></tr><tr><td>3</td><td>Monatliche Aggregation (Erster Monatstag 00:00:00 - Letzter Monatstag 23:59:59)</td></tr><tr><td>4</td><td>Jährliche Aggregation (01.01. 00:00:00 - 31.12. 23:59:59)</td></tr><tr><td>5</td><td>5-Minütige Aggregation (Aus Rohdaten berechnet)</td></tr><tr><td>6</td><td>1-Minütige Aggregation (Aus Rohdaten berechnet)</td></tr><tr><td>8</td><td>15-Minütige Aggregation (Aus Rohdaten berechnet) (ab Version 8.1)</td></tr></table> |
| Startzeit | Datum/Zeit als Unix Zeitstempel (0 = von Anfang an) |
| Endzeit | Datum/Zeit als Unix Zeitstempel (0 = bis jetzt) |
| Limit | Maximale Anzahl an Datensätzen. (0 = Kein Limit, 10000 ist das harte Limit, welches immer greift) |

## RÜCKGABEWERT

Ein Array mit folgenden **key => value** Paaren.



Die Ausgabe startet mit dem neusten Datensatz und dann absteigend mit den älteren Datensätzen.

Bedeutung der Felder beim Aggregationstyp **Standard**

| INDEX | TYP | BESCHREIBUNG |
|---|---|---|
| Avg | variant | Durchschnittswert innerhalb dieses Aggregationszeitraums |
| Duration | integer | Dauer des Aggregationszeitraums in Sekunden |
| Max | variant | Größter Wert innerhalb dieses Aggregationszeitraums |
| MaxTime | variant | Datum/Zeit von **Max** als Unix Timestamp |
| Min | variant | Kleinster Wert innerhalb dieses Aggregationszeitraums |
| MinTime | variant | Datum/Zeit von **Min** als Unix Timestamp |
| TimeStamp | integer | Datum/Zeit vom Begin des Aggregationszeitraums als Unix Timestamp |

Bedeutung der Felder beim Aggregationstyp **Zähler**

| INDEX | TYP | BESCHREIBUNG |
|---|---|---|
| Avg | variant | Summe der positiven Delta innerhalb dieses Aggregationszeitraums |
| Duration | integer | Dauer des Aggregationszeitraums in Sekunden |
| Max | variant | Größter positives Delta innerhalb dieses Aggregationszeitraums |
| MaxTime | variant | Datum/Zeit von **Max** als Unix Timestamp |
| Min | variant | Kleinstes positives Delta innerhalb dieses Aggregationszeitraums |
| MinTime | variant | Datum/Zeit von **Min** als Unix Timestamp |
| TimeStamp | integer | Datum/Zeit vom Begin des Aggregationszeitraums als Unix Timestamp |

## BESCHREIBUNG

Dieser Befehlt gibt die aggregierten Daten zurück, die über das Archiv für eine Variable mit der **VariablenID** mitprotokolliert und berechnet wurden. Dabei können die **Start-** und **Endzeit** als Parameter angegeben werden. Der Parameter **Limit** limitiert die maximale Anzahl der zurückgegebenen Datensätze, wobei das Limit von 10000 Datensätzen pro Abfrage nie überschritten werden kann.



Diese Funktion garantiert, dass alle Datensätze zwischen Startzeit und Endzeit auch vorhanden sind und ausgegeben werden. Das bedeutet, dass es es keine zeitlichen Lücken zwischen den Datensätzen gibt. Datensätze werden maximal bis zum jetzigen Zeitpunkt ausgegeben.



Es werden immer ganze aggregierte Zeiträume abgefragt. Dafür muss der Startzeitpunkt (abhängig von der **Aggregationsstufe**) der/des Stunde/Tages/Woche/Monats/Jahr im Zeitraum von **Startzeit** und **Endzeit** umfasst sein.



Die Aggregationsstufen 15-/5-/1-Minütige Aggregation werden aus den Rohdaten berechnet, sodass diese bei vielen Rohdaten ggf. mehr Systemleistung erfordern, als die anderen Aggregationsstufen

## BEISPIEL

```
//Alle Datensätze vom 01.01.2013 bis zum 31.12.2013 abfragen
(Tägliche Aggregationsstufe)
//z.B. um den Verbrauch am jeweiligen Tag zu ermitteln oder die
Durchschnittstemperatur am jeweiligen Tag
$werte = AC_GetAggregatedValues(12345, 55554, 1 /* Täglich */,
mktime(0, 0, 0, 1, 1, 2013), mktime(23, 59, 59, 12, 31, 2013), 0);
//55554 ist die ID der Variable, 12345 vom Archiv

//Alle heutigen Datensätze abfragen (Tägliche Aggregationsstufe)
//z.B. um den heutigen Verbrauch er ermitteln oder die heutige
Durchschnittstemperatur
$werte = AC_GetAggregatedValues(12345, 55554, 1 /* Täglich */,
strtotime("today 00:00"), time(), 0); //55554 ist die ID der
Variable, 12345 vom Archiv

//Alle gestrigen Datensätze abfragen (Stündlichen Aggregationsstufe)
//z.B. um den gesterigen Verbrauch oder die durchschittliche
Windgeschwindigkeit jeder Stunde zu begutachten
$werte = AC_GetAggregatedValues(12345, 55554, 0 /* Stündlich */,
strtotime("yesterday 00:00"), strtotime("today 00:00")-1, 0);
//55554 ist die ID der Variable, 12345 vom Archiv

//Dieser Teil erstellt eine Ausgabe im Skriptfenster mit den
abgefragten Werten
foreach($werte as $wert) {
echo date("d.m.Y H:i:s", $wert['TimeStamp']) . " -> " . $wert['Avg']
. PHP_EOL;
}
```

# AC_GetAggregationType

Benötigt: IP-Symcon >= 3.0

```
int AC_GetAggregationType (int $InstanzID, int $VariablenID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID zum Archiv |
| VariablenID | ID der abzufragenden Variablen |

## RÜCKGABEWERT

Ein Integer der den Typ der Aggregation wiedergibt.

| RÜCKGABEWERT | BESCHREIBUNG |
|---|---|
| 0 | Standard |
| 1 | Zähler |

## BESCHREIBUNG

Diese Funktion fragt den Aggregationstyp der Variable mit der ID **VariablenID** ab und gibt diesen als Integer zurück.

## BEISPIEL

```
// Die Variable wird als Zähler aggregiert
echo AC_GetAggregationType(39147 /*[Archive]*/, 53716 /
*[TestVariableZähler]*/);

// Ausgabe: 1
```

# AC_GetAggregationVariables

Benötigt: IP-Symcon >= 3.0

```
array AC_GetAggregationVariables (int $InstanzID, boolean
$DatenbankAbfrage)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID zum Archiv |
| DatenbankAbfrage | True, wenn zusätzliche Informationen angezeigt werden sollen, sonst False (ab Version 4.0 unwichtig, Angabe aber nötig) |

## RÜCKGABEWERT

Ein Array mit folgenden **key => value** Paaren.

| INDEX | TYP | BESCHREIBUNG |
|---|---|---|
| FirstTime | integer | Datum/Zeit vom Beginn des Aggregationszeitraums als Unix Zeitstempel |
| LastTime | integer | Datum/Zeit vom letzten Eintrag des Aggregationszeitraums als Unix Zeitstempel |
| RecordCount | integer | Anzahl der Datensätze |
| RecordSize | integer | Größe aller Datensätze in Bytes |
| VariableID | integer | ID der Variable |
| AggregationType | integer | Aggregationstyp als Integer. Siehe auch AC_GetAggregationType |
| AggregationVisible | boolean | Gibt an ob die Variable in der Visualisierung angezeigt wird. Siehe auch AC_GetGraphStatus |
| AggregationActive | boolean | Gibt an ob das Logging für diese Variable Aktiv ist. Siehe auch AC_GetLoggingStatus |

## BESCHREIBUNG



Ab Version 4.0 wird der angegebene Parameter **DatenbankAbfrage** nicht mehr genutzt. Für Abwärtskompatibilität muss dieser aber mit angegeben werden.

Diese Funktion gibt ein Array aller geloggten Variablen des Archiv Handlers mit der ID **InstanzID** wieder. Zusätzliche Informationen zu jeder Variable können mit **Datenbankabfrage** aktiviert werden.(Funktional bis Version 3.4)

## BEISPIEL

```
// Ausgabe mit zusätzlichen Informationen
var_dump(AC_GetAggregationVariables(39147 /*[Archive]*/, true));

/* Beispielausgabe
array(1) {
[0]=>
array(8) {
["FirstTime"]=>
int(1444221643)
["LastTime"]=>
int(1444221705)
["RecordCount"]=>
int(6)
["RecordSize"]=>
int(78)
["VariableID"]=>
int(53716)
["AggregationType"]=>
int(0)
["AggregationVisible"]=>
bool(true)
["AggregationActive"]=>
bool(true)
}
}
*/
```

# AC_GetCounterIgnoreZeros

Benötigt: IP-Symcon >= 5.5

```
boolean AC_GetCounterIgnoreZeros (int $InstanzID, int $VariablenID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID zum Archiv |
| VariablenID | ID der abzufragenden Variable |

## RÜCKGABEWERT

True wenn Ignorieren von Nullen und negative Werte für die Variable aktiv ist, sonst False

## BESCHREIBUNG

Diese Funktion fragt ab, ob die Variable mit der ID **VariablenID** im Archiv Control mit der ID **InstanzID** Nullen und negative Werte ignoriert.



Die Funktion funktioniert nur mit Variablen, die als Zähler geloggt sind. Ist das Ignorieren von Nullen und negativen Werte aktiv, werden diese dennoch weiterhin geloggt, fließen allerdings nicht mit in die Aggregation ein.

## BEISPIEL

```
// Abfrage der Variable "54321" aus dem ArchivControl "12345"
AC_GetCounterIgnoreZeros(12345, 54321);
```

# AC_GetGraphStatus

Benötigt: IP-Symcon >= 3.0

```
boolean AC_GetGraphStatus (int $InstanzID, int $VariablenID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID zum Archiv |
| VariablenID | ID der abzufragenden Variablen |

## RÜCKGABEWERT

**True** wenn die Visualisierung der Variable aktiv ist, sonst **False**

## BESCHREIBUNG

Diese Funktion fragt ab, ob die Variable mit der ID **VariablenID** in der Visualisierung angezeigt wird.

## BEISPIEL

```
// Abfrage der Variable "TestVariable"
var_dump(AC_GetGraphStatus(39147 /*[Archive]*/, 53716 /
*[TestVariable]*/));
echo AC_GetGraphStatus(39147 /*[Archive]*/, 53716 /
*[TestVariable]*/);

/* Beispielausgabe
var_dump:
bool(true)
echo:
1
*/
```

# AC_GetLoggedValues

Benötigt: IP-Symcon >= 3.0

```
array AC_GetLoggedValues (int $InstanzID, int $VariablenID, int
$Startzeit, int $Endzeit, int $Limit)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID zum Archiv |
| VariablenID | ID der abzufragenden Variablen |
| Startzeit | Datum/Zeit als Unix Zeitstempel. (0 = von Anfang an) |
| Endzeit | Datum/Zeit als Unix Zeitstempel. (0 = bis jetzt) |
| Limit | Maximale Anzahl an Datensätzen. (0 = Kein Limit, 10000 ist das harte Limit, welches immer greift) |

## RÜCKGABEWERT

Ein Array mit folgenden **key => value** Paaren.



Die Ausgabe startet mit dem neusten Datensatz und dann absteigend mit den älteren Datensätzen.

| INDEX | TYP | BESCHREIBUNG |
|---|---|---|
| **Duration** | integer | Dauer in Sekunden, die der Datensatz gesetzt war |
| **TimeStamp** | integer | Datum/Zeit beim Erstellen des Datensatzes als Unix Timestamp |
| **Value** | variant | Wert |

## BESCHREIBUNG

Dieser Befehlt gibt die Rohdaten zurück, die über das Archiv Control für eine Variable mit der **VariablenID** mitprotokolliert wurden. Dabei können die **Start-** und **Endzeit** als Parameter angegeben werden. Der Parameter **Limit** limitiert die maximale Anzahl der zurückgegebenen Datensätze, wobei das Limit von 10000 Datensätzen pro Abfrage nie überschritten werden kann.



Das Archiv Control erstellt nur neue Datensätze für geänderte Werte! Aktualisierungen werden nicht berücksichtigt.

Diese Funktion verursacht eine erhebliche Belastung für das System, da in den meisten Fällen sehr viele Datensätze verarbeitet werden müssen. Bitte weichen Sie in jedem möglichen Falle auf bereits voraggregierte Werte über die AC_GetAggregatedValues Funktion aus. Diese Funktion bietet Ihnen direkten Zugriff auf Mittelwerte oder Summen für verschiedene Zeiträume wie z.B. Jahre, Monate, Wochen und Tage.

In IP-Symcon 2.x war diese Funktion fehlerbehaftet. Es wurde immer ein Wert mehr ausgegeben, welcher vor der Startzeit lag. Das war insofern nützlich, dass die für z.B. die Graphengenerierung der vorherige Wert bekannt war, und für die vollständige Zeitspanne ein Graph erstellt werden konnte. Anderfalls wäre eine Lücke am Anfang des Graphen sichtbar. Für eine Funktion mit dem Parameter Startzeit ist diese Funktionsweise jedoch nicht korrekt. Wer für seine Skripte genau diese besondere Funktionsweise benötigt, kann die Hilfsfunktion nutzen, die den vorherigen Datensatz, der vor der Startzeit liegt, entsprechend abfragt und zum Array hinzufügt.

## BEISPIEL

```
//Den letzten Wert, der in der Datenbank gespeichert wurde, holen
$letzter_Wert = AC_GetLoggedValues(12345, 55554, 0, 0, 1)[0]
['Value'];

//Alle Datensätze vom 01.01.2013 bis zum 07.01.2013 abfragen
$werte = AC_GetLoggedValues(12345, 55554, mktime(0, 0, 0, 1, 1,
2013), mktime(23, 59, 59, 1, 7, 2013), 0); //55554 ist die ID der
Variable, 12345 vom Archiv Control

//Alle heutigen Datensätze abfragen
$werte = AC_GetLoggedValues(12345, 55554, strtotime("today 00:00"),
time(), 0); //55554 ist die ID der Variable, 12345 vom Archiv
Control

//Alle gestrigen Datensätze abfragen
$werte = AC_GetLoggedValues(12345, 55554, strtotime("yesterday
00:00"), strtotime("today 00:00")-1, 0); //55554 ist die ID der
Variable, 12345 vom Archiv Control

//Dieser Teil erstellt eine Ausgabe im Skriptfenster mit den
abgefragten Werten
foreach($werte as $wert) {
echo date("d.m.Y H:i:s", $wert['TimeStamp']) . " -> " .
$wert['Value'] . PHP_EOL;
}
```

```
//Hilfsfunktion, die die Funktionsweise von IP-Symcon 2.x nachbildet
function AC_GetLoggedValuesCompatibility($instanceID, $variableID,
$startTime, $endTime, $limit) {
$values = AC_GetLoggedValues($instanceID, $variableID, $startTime,
$endTime, $limit );
if((sizeof($values) == 0) || (end($values)['TimeStamp'] >
$startTime)) {
$previousRow = AC_GetLoggedValues($instanceID, $variableID, 0,
$startTime - 1, 1 );
$values = array_merge($values, $previousRow);
}
return $values;
}
```

# AC_GetLoggingStatus

Benötigt: IP-Symcon >= 3.0

```
boolean AC_GetLoggingStatus (int $InstanzID, int $VariablenID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID zum Archiv |
| VariablenID | ID der abzufragenden Variablen |

## RÜCKGABEWERT

**True** wenn Logging für die Variable aktiv ist, sonst **False**

## BESCHREIBUNG

Diese Funktion fragt ab, ob die Variable mit der ID **VariablenID** geloggt wird.

## BEISPIEL

```
// Abfrage der Variable "TestVariable"
var_dump(AC_GetLoggingStatus(39147 /*[Archive]*/, 53716 /
*[TestVariable]*/));
echo AC_GetLoggingStatus(39147 /*[Archive]*/, 53716 /
*[TestVariable]*/);

/* Beispielausgabe
var_dump:
bool(true)
echo:
1
*/
```

# AC_ReAggregateVariable

Benötigt: IP-Symcon >= 3.0

```
boolean AC_ReAggregateVariable (int $InstanzID, int $VariablenID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID zum Archiv |
| VariablenID | ID der zu reaggregierenden Variable |

## RÜCKGABEWERT

**True** wenn erfolgreich, sonst **False**

## BESCHREIBUNG

Diese Funktion startet die Reaggregation der Variable mit der ID **VariablenID**.



Das Reaggregieren kann sehr lange dauern. Der momentane Fortschritt der Reaggregation kann über regelmässige Einträge im Meldungsfenster kontrolliert werden.

## BEISPIEL

```
// Dies startet die Reaggregation der Variable "Testvariable"
AC_ReAggregateVariable(39147 /*[Archive]*/, 53716 /
*[TestVariable]*/);

/* Beispielausgabe im Meldungsfenster
07.10.2015 15:50:05 | Archive Control | Reaggregation for VariableID
#53716 is in progress... 10/2015! (6 rows)
07.10.2015 15:50:05 | Archive Control | Reaggregation for VariableID
#53716 is complete! (6 rows)
*/
```

# AC_SetAggregationType

Benötigt: IP-Symcon >= 3.0

```
boolean AC_SetAggregationType (int $InstanzID, int $VariablenID, int
$Aggregationstyp)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID zum Archiv |
| VariablenID | ID der Variable bei der der Aggregationstyp gesetzt werden soll |
| Aggregationstyp | |

| AGGREGATIONSTYP | DESCRIPTION |
|---|---|
| 0 | Standard |
| 1 | Counter |

## RÜCKGABEWERT

**True** wenn erfolgreich, sonst **False**

## BESCHREIBUNG

Diese Funktion setzt den Aggregationstyp **Aggregationstyp** bei der Variable mit der ID **VariablenID**



Im Anschluss ist eine Reaggregation der Variable nötig.

## BEISPIEL

```
// Setzt den Aggregationtyp der Variable "TestVariable" auf 1
(Zähler)
AC_SetAggregationType(39147 /*[Archive]*/, 53716 /*[TestVariable]*/,
1);
```

# AC_SetCompaction

Benötigt: IP-Symcon >= 6.3

```
boolean AC_SetCompaction (int $InstanzID, int $VariablenID, int
$MonatsVersatz, int $Verdichtungstyp)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID zum Archiv |
| VariablenID | ID der Variablen bei der die Verdichtung angepasst werden soll |
| MonatsVersatz | Nach wie vielen Monaten die Verdichtung angewandt werden soll; 0 = Nach Abschluß das aktuellen Monats, -1 = Werte direkt verdichten |
| Verdichtungstyp | |

| VERDICHTUNGSTYP | BESCHREIBUNG |
|---|---|
| -1 | Verdichtung deaktivieren |
| 0 | Auf einen Wert pro Minute verdichten |
| 1 | Auf einen Wert pro 5 Minuten verdichten |
| 2 | Auf einen Wert pro Stunde verdichten |
| 3 | Auf einen Wert pro Tag verdichten |
| 4 | Auf einen Wert pro Woche verdichten |
| 5 | Auf einen Wert pro Monat verdichten |
| 6 | Auf einen Wert pro Jahr verdichten |
| 7 | Werte löschen |

## RÜCKGABEWERT

**True** wenn erfolgreich, sonst **False**

## BESCHREIBUNG

Diese Funktion passt den Verdichtungseintrag der Variable mit der ID **VariablenID** für den Monatsversatz **MonatsVersatz** an und setzt ihn auf den Verdichtungstyp **Verdichtungstyp**. Dabei bleiben Verdichtungseinträge mit anderen Monatsversätzen erhalten.



Im Anschluss ist eine Reaggregation der Variable nötig.

## BEISPIEL

```
// Setzt das direkte Verdichten der Variable "TestVariable" auf 1
(Ein Wert pro 5 Minuten)
AC_SetCompaction(39147 /*[Archive]*/, 53716 /*[TestVariable]*/, -1,
1);
```

# AC_SetCounterIgnoreZeros

Benötigt: IP-Symcon >= 5.5

```
boolean AC_SetCounterIgnoreZeros (int $InstanzID, int $VariablenID,
boolean $NullenIgnorieren)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID zum Archiv |
| VariablenID | ID der abzufragenden Variable |
| NullenIgnorieren | True, wenn Nullen und negative Werte ignoriert werden sollen, sonst False. |

## RÜCKGABEWERT

## BESCHREIBUNG

Diese Funktion ändert die Eigenschaft zum Ignorieren von Nullen und negative Werte der Variable mit der ID **VariablenID** im Archiv Control mit der ID **InstanzID** auf Aktiv.



Die Funktion funktioniert nur mit Variablen, die als Zähler geloggt sind. Nullen und negative Werte werden weiterhin geloggt, fließen allerdings nicht in die Aggregation mit ein.

// Setzt die Eigenschaft der Variable "54321" im Archiv Control "12345" zum Ignorieren von Nullen und negativen Werte auf true
AC_SetCounterIgnoreZeros(12345, 54321, true);

## BEISPIEL

# AC_SetGraphStatus

Benötigt: IP-Symcon >= 3.0

```
boolean AC_SetGraphStatus (int $InstanzID, int $VariablenID, boolean
$Aktiv)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID zum Archiv |
| VariablenID | ID der abzufragenden Variablen |
| Aktiv | True, wenn das Anzeigen der Variable in der Visualisierung aktiviert werden soll, sonst False. |

## RÜCKGABEWERT

**True** wenn erfolgreich, sonst **False**

## BESCHREIBUNG

Diese Funktion ändert die Eigenschaft zur Visualisierung der Variable mit der ID **VariablenID** auf **Aktiv**.

## BEISPIEL

```
// Setzt die Eigenschaft der "Testvariable" zur Visualisierung auf
true
AC_SetGraphStatus(39147 /*[Archive]*/, 53716 /*[TestVariable]*/,
true);
```

# AC_SetLoggingStatus

Benötigt: IP-Symcon >= 3.0

```
boolean AC_SetLoggingStatus (int $InstanzID, int $VariablenID,
boolean $Aktiv)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID zum Archiv |
| VariablenID | ID der abzufragenden Variablen |
| Aktiv | True, wenn das Logging der Variable aktiviert werden soll, sonst False. |

## RÜCKGABEWERT

**True** wenn erfolgreich, sonst **False**

## BESCHREIBUNG

Diese Funktion ändert die Eigenschaft zum Loggen der Variable mit der ID **VariablenID** auf **Aktiv**.



Das Logging der Variablen wird sofort aktiviert, da allerdings der aktuelle Wert und Änderungen der Variablen asynchron geloggt werden, kann es sein, dass diese bei Ausführung eines Skriptes nicht direkt zur Verfügung stehen. Ein möglicher Workaround hierfür ist die Verwendung des Sleep-Befehls

## BEISPIEL

```
// Setzt die Eigenschaft der "Testvariable" zum Loggen auf true
AC_SetLoggingStatus(39147 /*[Archive]*/, 53716 /*[TestVariable]*/,
true);
```

# Datenformat



Ab Version 4.0 von IP-Symcon werden CSV-Dateien zur Datenspeicherung genutzt. Zuvor wurde dies von einer MySQL Datenbank erledigt.

Das Archiv speichert zwei unterschiedliche Arten der Daten.
Primär werden die Rohdaten gespeichert.
Sekündar werden die nach Zeitraum (Jahr, Monat, Woche, Tag, Stunde) aggregierten Daten berechnet und geschrieben.

## CSV STRUKTUR

Alle Dateien haben die VariablenID im Namen und .csv als Dateiendung
Rohdaten werden in eigenen Ordnern pro Jahr und pro Monat geschrieben
Aggregationsdaten werden im DB Hauptordner gespeichert
Dateien werden erst erstellt, sobald Daten vorhanden sind
Das Dezimaltrennzeichen ist immer ein Punkt (.)

### ROHDATEN INHALT

Ein Datensatz pro Zeile.
Daten sind durch Komma (,) voneinander getrennt.
Unix-Zeitstempel
Rohwert



Die geloggten Rohwerte werden, unabhängig von der eingestellten Serverzeit, immer in UTC gespeichert.



String-Werte werden Base64 kodiert gespeichert.

### ROHDATEN VERZEICHNISSTRUKTUR

db\2010\01\12345.csv
db\2010\01\23456.csv
db\2010\02\12345.csv
db\2010\02\23456.csv

### INHALT AGGREGATIONSDATEN

Ein Datensatz pro Zeile.

Daten sind durch Komma (,) voneinander getrennt.

Unix-Zeitstempel für Avg-Wert
Avg-Wert
Delta zum Unix-Zeitstempel für Min-Wert
Min-Wert
Delta zum Unix-Zeitstempel für Max-Wert
Max-Wert

## AGGREGATIONSDATEN VERZEICHNISSTRUKTUR

db\12345.year.csv
db\12345.month.csv
db\12345.week.csv
db\12345.day.csv
db\12345.hour.csv

# Berechtigungssteuerung

Benötigt: IP-Symcon >= 8.0

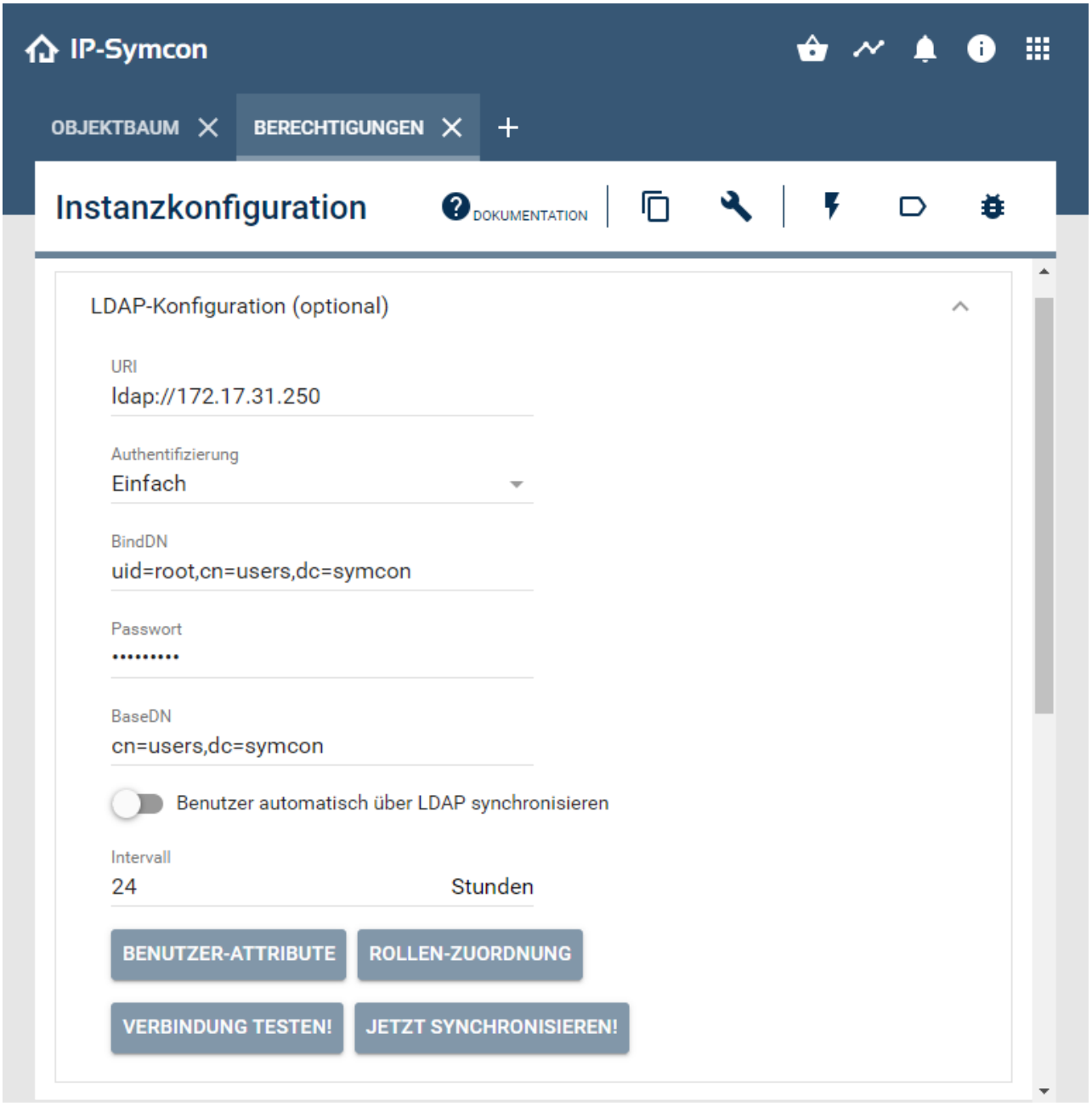


Die Berechtigungssteuerung ist eine kostenpflichtige Erweiterung, welche zu jeder bestehenden Symcon Lizenz erworben werden kann. Für eine passende Demo-Version wenden Sie sich bitte an unseren Support. Die Erweiterung kann direkt im Shop erworben werden.

Die Berechtigungssteuerung ermöglicht es Benutzer und Rollen zu erstellen und ihnen bestimmte Berechtigungen zuzuweisen. Außerdem können Benutzer mit einem bestehenden LDAP Authentifizierungsserver synchronisiert werden.

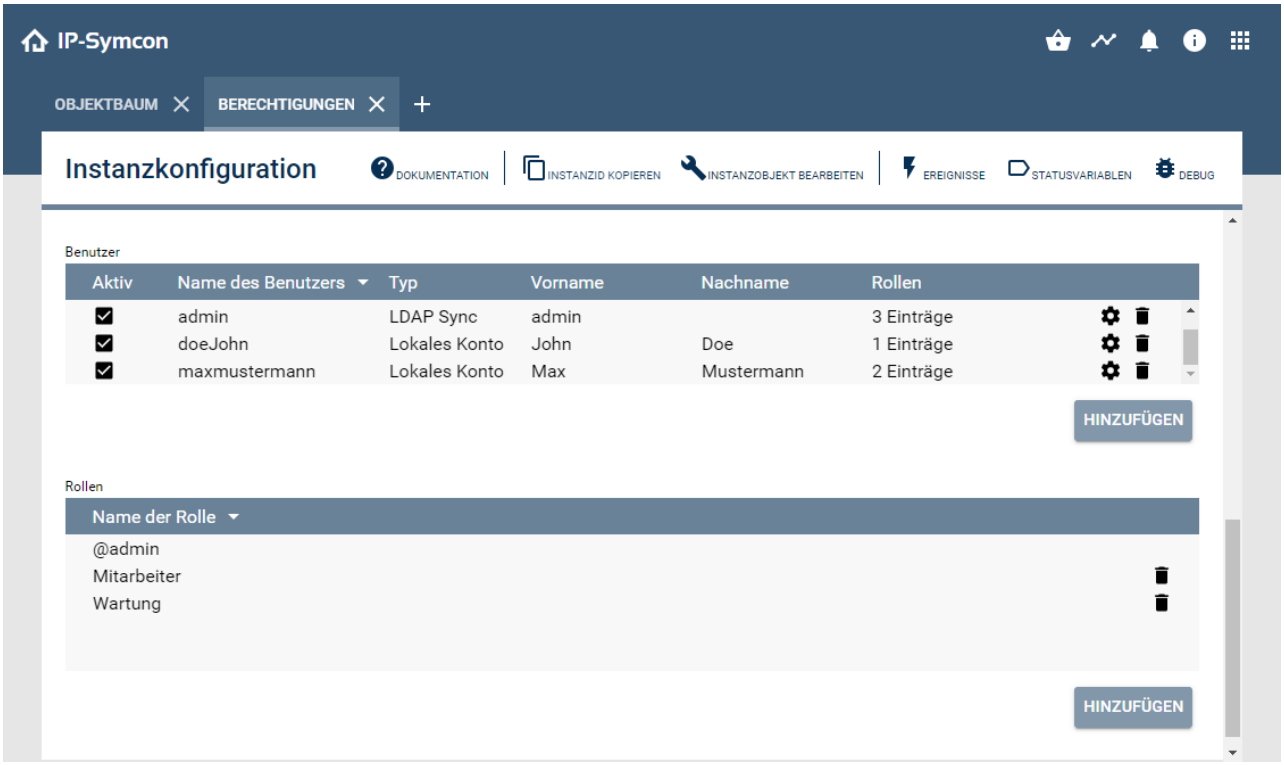
## LDAP-KONFIGURATION (OPTIONAL)

Nachdem alle Daten zur Verbindung angegeben wurden, können die Nutzer- und Rollenlisten mit einem Klick auf "Jetzt synchronisieren" mit den mit LDAP eingerichteten Äquivalenten gefüllt werden. Wenn die automatische Synchronisation aktiviert ist, werden Benutzer und Rollen in dem angegebenen Intervall synchronisiert.




# BENUTZER

Standardmäßig gibt es den @admin Benutzer. Dieser kann nicht editiert werden und hat alle Berechtigungen. Um sich mit diesem anzumelden wird das Kennwort für den Fernzugriff verwendet. Die Berechtigungen für etwa den Zugriff auf eine Visualisierung kann in der entsprechenden Instanz festgelegt werden. Bei Bedarf kann ein Nutzer deaktiviert werden, ohne ihn direkt zu löschen.

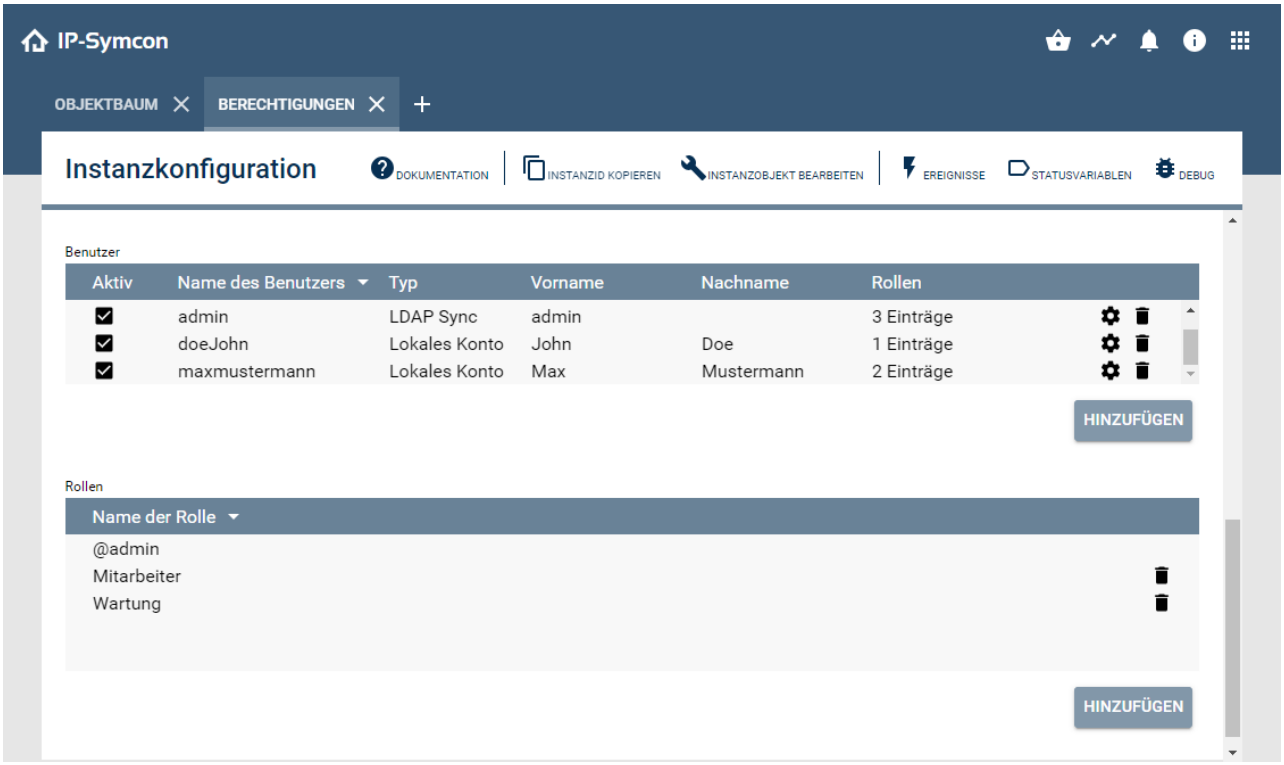



## BENUTZER HINZUFÜGEN

Beim Hinzufügen eines Nutzers können neben dem Passwort auch ein Vor- und Nachname angegeben werden. Diese werden unter anderem in der Visualisierung angezeigt. Beim Erstellen kann zwischen den Kontotypen "Lokales Konto" und "LDAP Sync" gewählt werden. Ist "LDAP Sync" gewählt, muss anstelle eines Passworts die UserDN angegeben werden. In der Liste können alle gewünschten Rollen ausgewählt werden.

# ROLLEN

Standardmäßig gibt es die @admin Rolle. Diese kann nicht editiert werden und hat alle Berechtigungen. Erstellte Rollen können den erstellten Benutzern hinzugefügt werden. Die Berechtigungen für etwa den Zugriff auf eine Visualisierung kann in der entsprechenden Instanz festgelegt werden.




## ROLLEN HINZUFÜGEN

Jeder Rolle kann beim Erstellen ein Name gegeben werden.

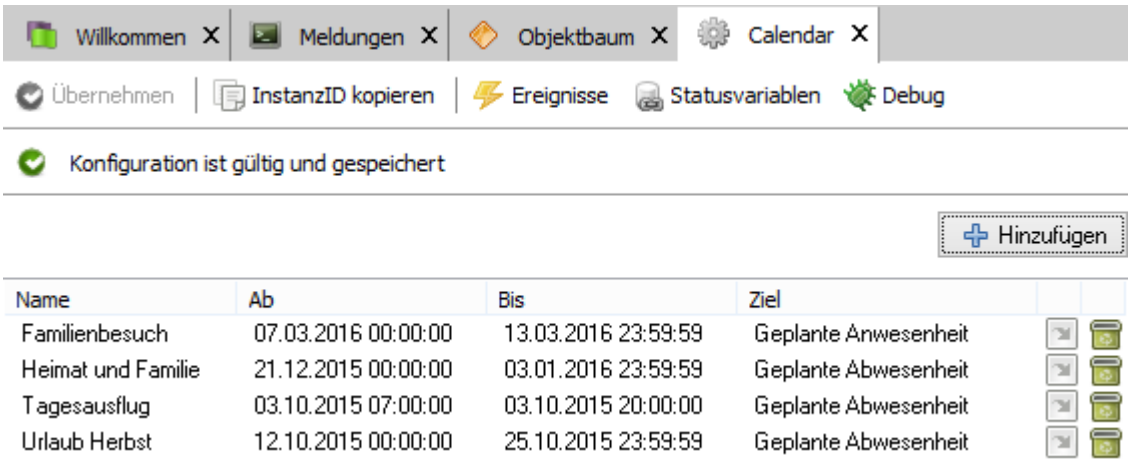
# Calendar Control

Benötigt: IP-Symcon >= 4.0

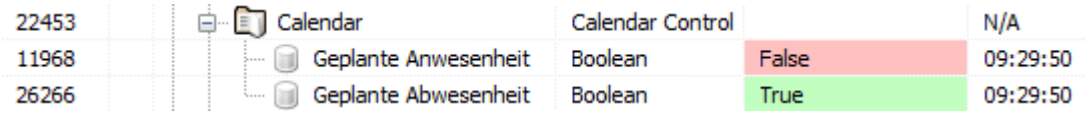
## BESCHREIBUNG

Das "Calendar Control" Modul bietet die Möglichkeit Zeiträume zu definieren in denen man voraussichtlich an- oder abwesend ist.

## EINBINDUNG IN IP-SYMCON




In der Instanz "Kalender" ("Logische Baumansicht"->"Kern Instanzen") können über die Konfigurationsseite neue Zeiträume hinzugefügt werden.
In diesen Zeiträumen wird die jeweils passende Variable auf "True" geschaltet. Wenn beide Variablen auf False geschaltet sind, ist für die jetzige Uhrzeit kein Zeitraum definiert.




## BEISPIELE

Geplante Anwesenheit
Ein Familienbesuch wird als geplante Anwesenheit eingetragen. Zu Zeiten, in denen normalerweise niemand in der Wohnung ist, wird automatisch die Heizung gedrosselt um Energie zu sparen. Durch Berücksichtigung der Variable "geplante Anwesenheit" kann dies verhindert werden, ohne eingerichteten Skripte oder Wochenpläne abzuändern oder manuell auszuschalten.
Geplante Abwesenheit
Der Familienurlaub an der Ostsee wird als geplante Abwesenheit eingetragen. Durch die geplante Abwesenheit kann eine eingerichtete Energiesparautomation dauerhaft für den Zeitraum aktiviert werden, welche im Alltagsleben normalerweise nur zu bestimmten Zeiten aktiv wäre.

# Connect Control

Benötigt: IP-Symcon >= 4.0

## BESCHREIBUNG

Das Connect Control Modul bietet die Möglichkeit einen gesicherten Zugriff von Außerhalb zu gewährleisten. Diese Verbindung kann dazu genutzt werden von überall via Verwaltungskonsole das System zu verwalten oder über die Visualisierungen und mobilen Apps darauf zuzugreifen.
Es müssen immer zusätzlich die unter Sicherheit angesprochen Maßnahmen ergriffen werden um auch nach Aktivierung des Connect Moduls einen ungewollten Zugriff von Außerhalb zu unterbinden.
Das Modul kommt komplett ohne Portfreigabe oder Einrichtung eines DynDNS aus. Es wird automatisch eine URL über unseren Server bereit gestellt. Durch das gesicherte Tunneling sind keine besonderen Einstellung in der Firewall nötig und auch der Zugriff innerhalb eines Firmennetzwerk kann möglich sein.

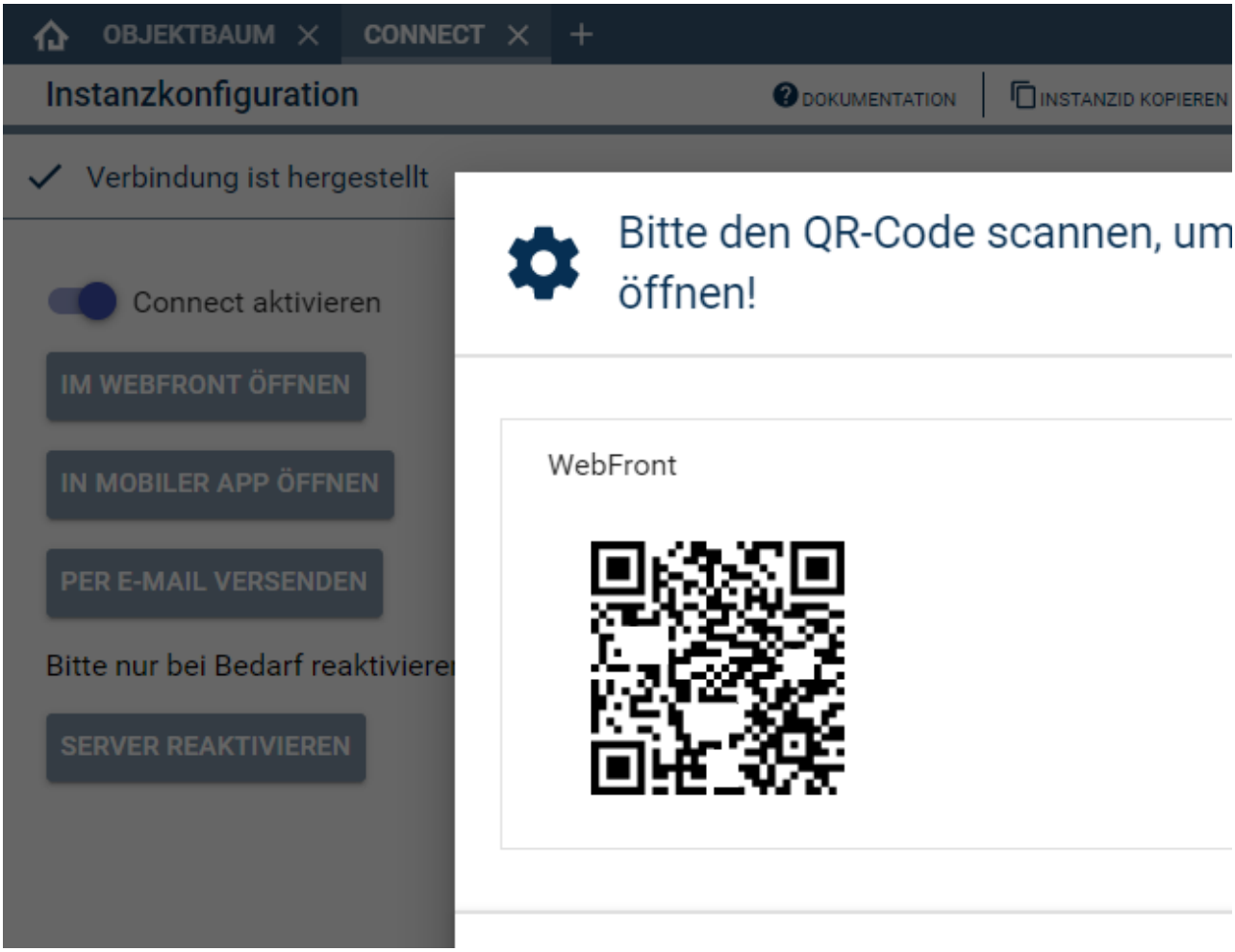


Im Fehlerfall sind im Meldungfenster weitere Informationen einsehbar. Weitere Informationen sind unter Lösungen im Fehlerfall nachzulesen




## EINBINDUNG IN IP-SYMCON

Voraussetzung um den Symcon Connect Dienst nutzen zu können, ist eine aktive Subskription. Pro Lizenz ist maximal eine aktive Server-Verbindung möglich




In der Instanz "Connect" ("Logische Baumansicht"->"Kern Instanzen") kann über "Symcon Connect aktivieren" das Modul aktiviert werden.
Die folgenden 3 Zugriffsmethoden sind vorhanden:
"WebFront öffnen"
Dies öffnet im Webbrowser die persönliche Internetadresse, welche das folgende Format (http://32-stellige-zeichenfolge.ipmagic.de) aufweist. Über die in der Adressleiste angezeigte Adresse, kann ab sofort von überall auf das WebFront zugegriffen werden. Dazu kann diese in einen beliebigen Browser oder Smartphone eingegeben werden. Es wird folgend eine verschlüsselte Verbindung aufgebaut.

Es ist empfohlen das WebFront mit einem Passwort zu sichern.

"Auf Smartphone öffnen"
Auf mobilen Geräten kann die Adresse via QR Code eingelesen werden. (Hierzu wird eine beliebige QR-Code Lesesoftware benötigt. Ab iOS11 kann direkt über die integrierte Kamera App eingescannt werden.) Im QR-Code wird der Link in Form eines WebFront Freigabelinks formatiert und öffnet somit automatisch die benötigte IP-Symcon App, inklusive dem gewählten WebFront.

Manche QR-Code Lesesoftware kommt mit den erstellten Links nicht zurecht. In diesem Fall ist eine andere QR-Code Software zu nutzen.

"Via E-Mail senden"
Das als Standard eingerichtete E-Mail Programm wird automatisch geöffnet und es muss nur noch die Adresse des Empfängers eingetragen werden.
Auf diesem Wege kann der Link und das damit verknüpfte IP-Symcon, Dritten, zur Verfügung gestellt werden.



Um über das Connect Modul auf die Verwaltungskonsole zugreifen zu können, muss der Fernzugriff aktiviert und ein Passwort gesetzt sein. Nun kann über eine installierte Verwaltungskonsole und der Angabe der URL, eine verschlüsselte Verbindung aufgebaut und genutzt werden.

## LIMITATIONEN

### DATENMENGE

Die Gesamtdatenmenge, die vom Connect Control täglich versendet werden kann, ist 1GB.

### TÄGLICHE RECONNECTS

Als Standard sind pro Tag 25 Reconnects zum Connect Dienst möglich. Per Spezialschalter (ConnectLimit) sind diese auf 100 erhöhbar.

## LÖSUNGEN IM FEHLERFALL

Im Zusammenhang mit dem Connect Control sind einige Einstellungen zu beachten die zu einem Fehler führen können. Im Fehlerfall werden im Meldungsfenster Hinweise gegeben, an welcher Stelle ein Fehler aufgetreten ist.

### STATUS

Der momentane Serverstatus vom Connect Control ist auf dieser Statusseite einsehbar.

### NEUSTART DES DIENSTES

Ist ein unvorhersehbarer Verbindungsfehler vorgefallen, kann ein Neustart des Dienstes via Tray-Icon oder eine Reaktivierung des Servers Abhilfe schaffen. Informationen dazu können unter Server reaktivieren gefunden werden.

### EINRICHTUNG IN DER FIREWALL

Um IP-Symcon den Zugriff auf bestimmte Funktionen (z.B. Connect Control) außerhalb zu gewähren ist es gegebenenfalls nötig diese in der Firewall einzurichten. Weitere Informationen können unter Firewall gefunden werden.

## SERVER REAKTIVIEREN

Zur zusätzlichen Sicherheit authentifiziert sich der sendene Server mithilfe eines Tokens, sodass dieser eindeutig als Versender identifiziert werden kann. Wenn der Server geändert, ausgetauscht oder neu installiert wird, ist ggf. eine Reaktivierung erforderlich. Diese Reaktivierung kann alle 24h vorgenommen und über den Button "Server reaktivieren" gestartet werden. Dabei wird zusätzlich per E-Mail über die Änderung benachrichtigt. Dies geschieht um einen Missbrauch einer Lizenz frühzeitig erkennen zu können.



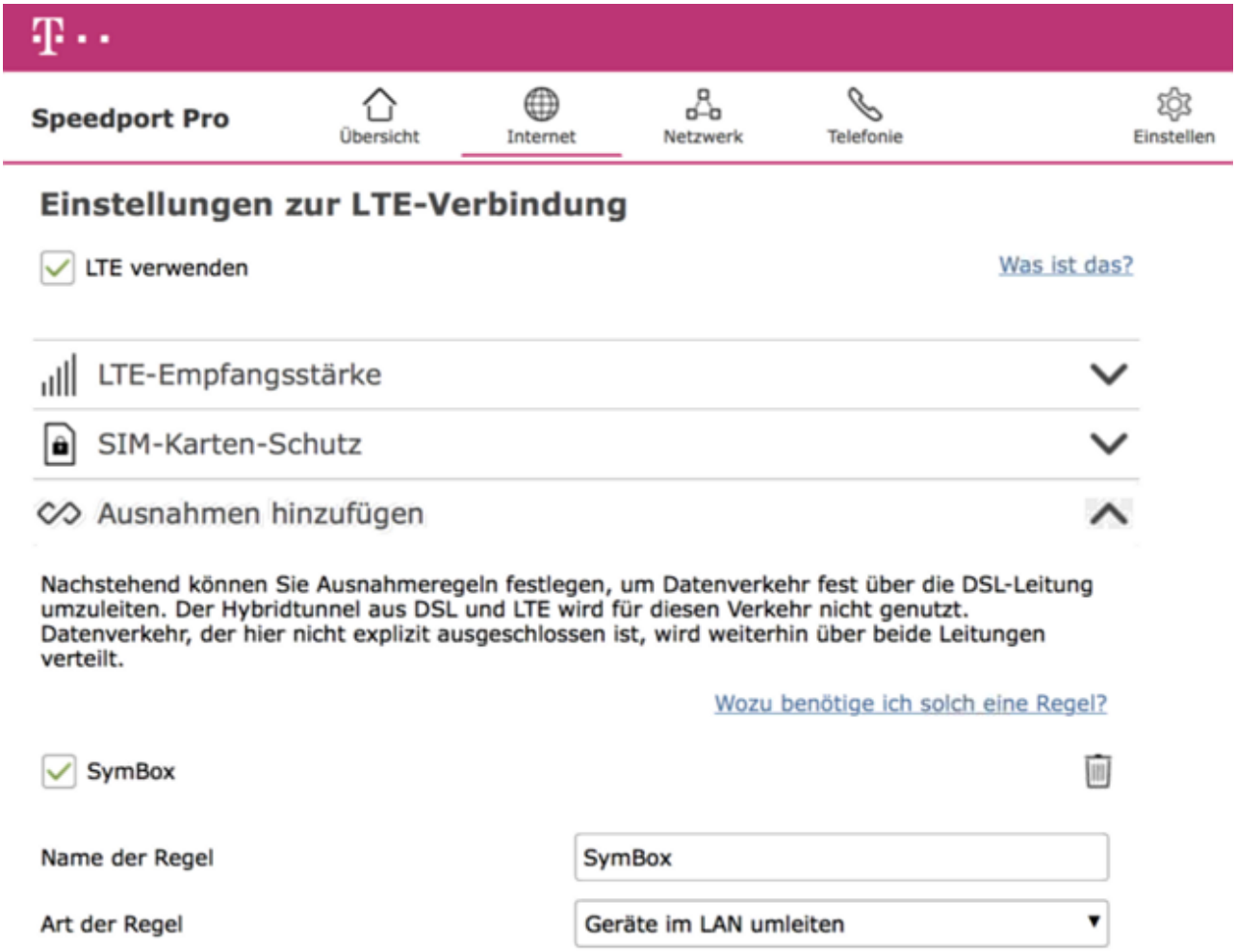
Eine Reaktivierung ist erforderlich, wenn die folgende Fehlermeldung erscheint: Ihr Server-Authentifizierungs-Token hat sich geändert. Bitte reaktivieren Sie Ihren Server.

## FIREWALL EINSTELLEN

Es müssen die Zugriffe auf live.symcon.de (443/TCP) und ipmagic.de (50000/TCP, 60000/TCP) freigeschaltet werden. Die TTL (Time To Live) für DNS Anfragen muss gleich oder kleiner als 60 Sekunden sein, da AWS (Amazon Web Services) diese jede Minute ändert.

## TELEKOM MIT DSL UND LTE

Bei Telekom Kunden, welche Hybride Lösung mit DSL+LTE nutzen, sollte eine Ausnahme für die SymBox bzw. den IP-Symcon Dienst erstellt werden, sodass dieser nur über DSL Kommuniziert.


Speedport Pro
Übersicht
Internet
Netzwerk
Telefonie
Einstellen
Einstellungen zur LTE-Verbindung
LTE verwenden
Was ist das?
LTE-Empfangsstärke
SIM-Karten-Schutz
Ausnahmen hinzufügen
Nachstehend können Sie Ausnahmeregeln festlegen, um Datenverkehr fest über die DSL-Leitung umzuleiten. Der Hybridtunnel aus DSL und LTE wird für diesen Verkehr nicht genutzt. Datenverkehr, der hier nicht explizit ausgeschlossen ist, wird weiterhin über beide Leitungen verteilt.
Wozu benötige ich solch eine Regel?
SymBox
Name der Regel
SymBox
Art der Regel
Geräte im LAN umleiten

# Cutter

Der Cutter dient dazu aus einem einkommenden Datenstrom, anhand von Trennzeichen oder einer festen Schrittanzahl, Abschnitte herauszunehmen und leitet diese weiter.

## EINBINDUNG IN IP-SYMCON

Der Cutter fungiert als Splitter Instanz und steht somit oftmals zwsichen I/O- und Geräte-Instanz.
Somit nimmt der Cutter den eingehenden Datenstrom auseinander und gibt diesen weiter.

### BENUTZE ZEICHEN FÜR SCHNITTE

Diese Einstellung setzt Trennzeichen von Links (Anfang des Datenstroms) nach Rechts (Ende des Datenstroms). Alles was sich zwischen den Trennzeichen befindet wird weitergeleitet. Optional können Hexzeichen eingeschaltet werden und somit können die Trennzeichen in Hexadezimal eingegeben werden. Wenn nur linkes oder rechts Trennzeichen eingetragen werden, wird der gesamte Datenstrom davor (nur rechts gesetzt) bzw. danach (nur links gesetzt) weitergeleitet.
Im Dropdown-Menü befinden sich Standard-Steuerbefehle, welche oftmal genutzt werden und das 1:1 weiterleiten.



Die Trennzeichen selbst werden nicht mit weitergeleitet. Diese werden ebenfalls herausgeschnitten, sodass nur die reinen Nutzdaten weitergeleitet werden.

### BENUTZE FESTE SCHNITTE

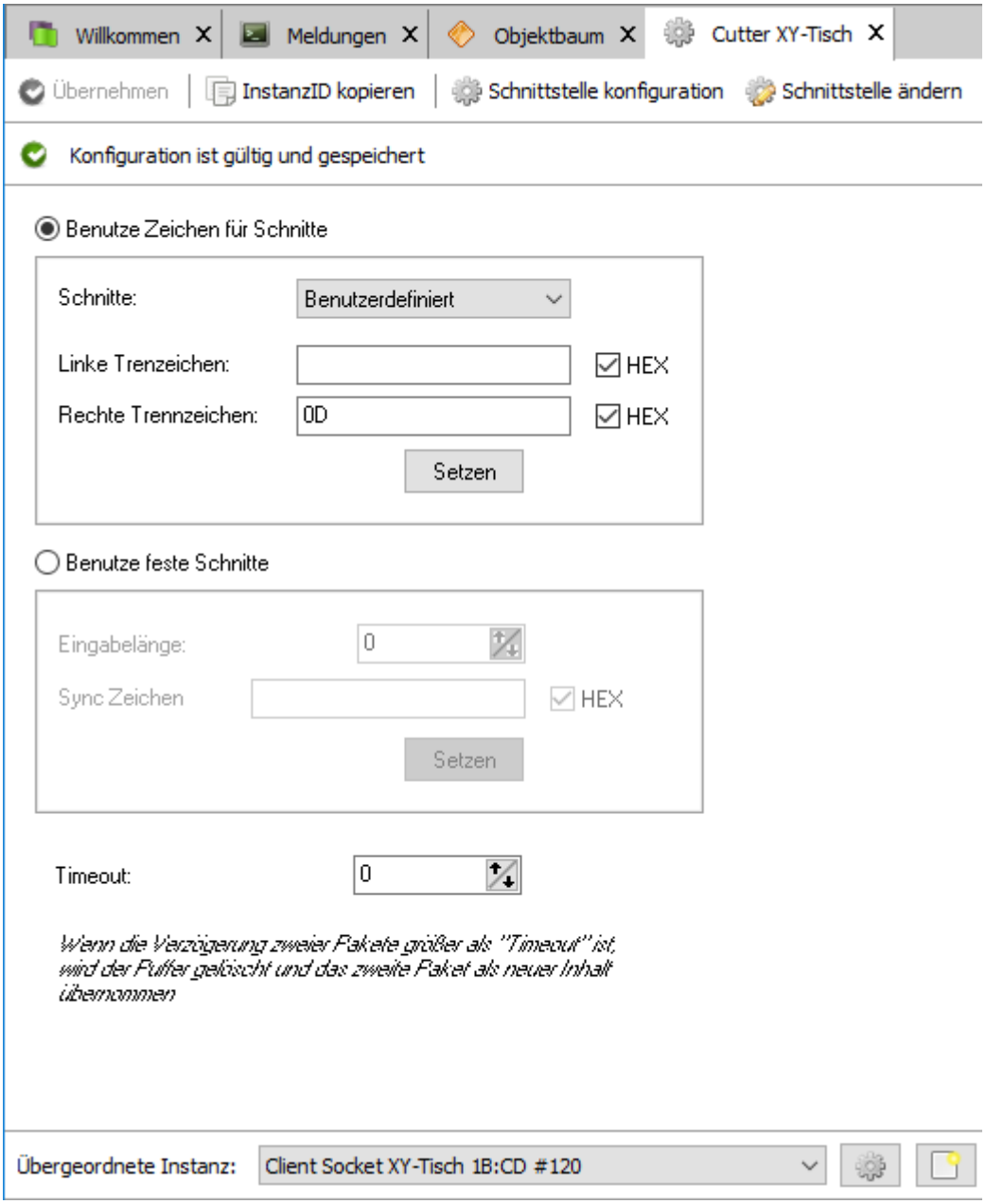
Es wird im Datenstrom nach dem Sync Zeichen gesucht. Der anschließende Datenstrom (inklusive Sync Zeichen) wird nach der eingestellten Eingabelänge abgeschnitten und weitergeleitet. Ist Eingabelänge = 0, wird der gesamte Datenstrom mitsamt Sync Zeichen weitergeleitet.

### IN VARIABLE ÜBERTRAGEN

Der Datenstrom kann über eine RegisterVariable (mithilfe eines Skripts) in eine Variable übertragen werden.

### SONDEROPTION TIMEOUT

Wenn die Verzögerung zweier Datenstrompakete größer als die eingestellte "Timeout"-Zeit ist, dann werden die beiden Pakete nicht als zusammenhängender Datenstrom, sondern als 2 serperate Datenpakete behandelt.


Willkommen
Meldungen
Objektbaum
Cutter XY-Tisch
Übernehmen
InstanzID kopieren
Schnittstelle konfiguration
Schnittstelle ändern
Konfiguration ist gültig und gespeichert
Benutze Zeichen für Schnitte
Schnitte:
Benutzerdefiniert
Linke Trenzeichen:
HEX
Rechte Trennzeichen:
0D
HEX
Setzen
Benutze feste Schnitte
Eingabelänge:
0
Sync Zeichen
HEX
Setzen
Timeout:
0
Wenn die Verzögerung zweier Pakete größer als "Timeout" ist, wird der Puffer gelöscht und das zweite Paket als neuer Inhalt übernommen
Übergeordnete Instanz:
Client Socket XY-Tisch 1B:CD #120

# Cutter_ClearBuffer

```
boolean Cutter_ClearBuffer (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID zum Cutter |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Diese Funktion löscht den Zwischenspeicher (Buffer) der Cutter Instanz mit der ID InstanzID.

## BEISPIEL

```
// Säubert den Buffer von der Clutter Instanz mit der ID 12345
Cutter_ClearBuffer(12345);
```

# DNS-SD Control

Benötigt: IP-Symcon >= 5.0
Das DNS-SD Control Modul verwaltet eingerichtete Bonjour Services.

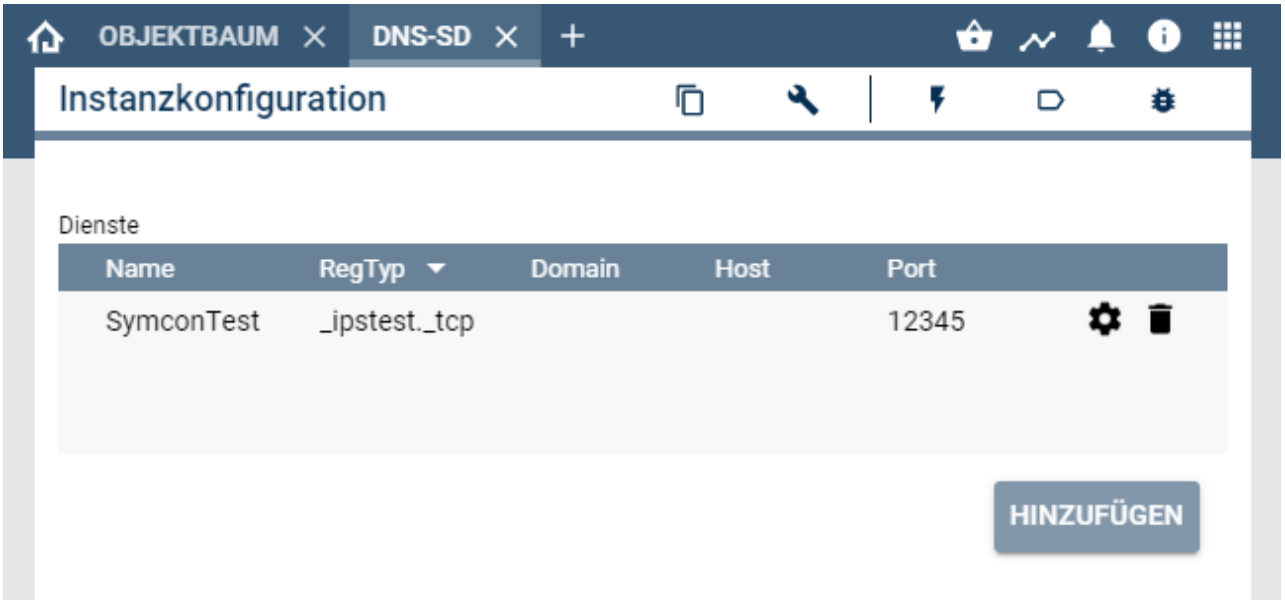
Für weiterführende Informationen kann nach Zeroconf oder mDNS gesucht werden.

## INSTALLATION

Das DNS-SD Control wird automatisch als Kern-Instanz installiert. Dieses kann im Objektbaum unter Kern Instanzen gefunden werden.

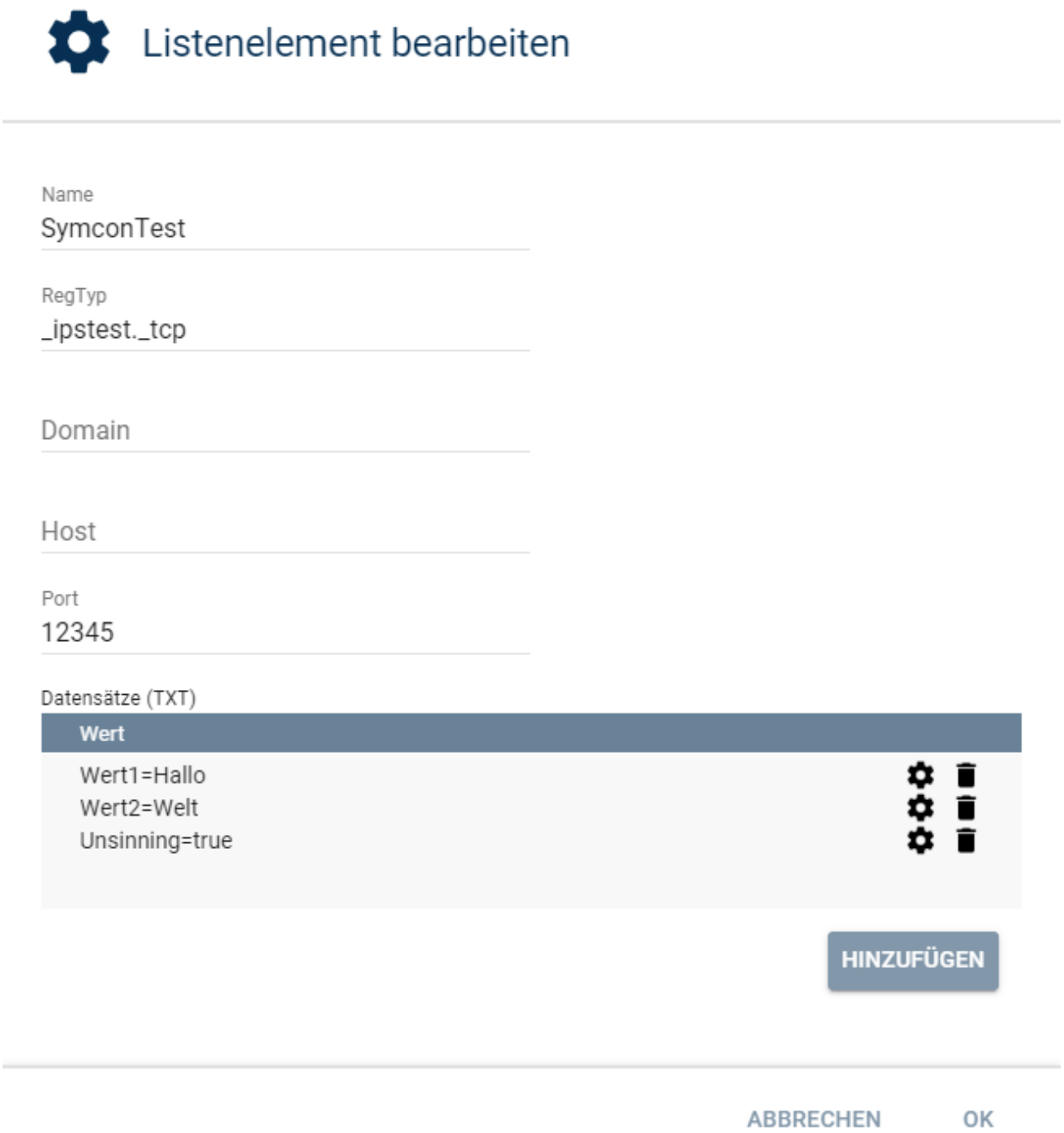
## KONFIGURATION

Über die Konfigurationseite des DNS-SD Control können neue Services (Listenelement) hinzugefügt werden.




Diese benötigen einen Namen, RegType und Port. Domain und Host sind optional, genauso wie Werte. Diese werden als Key -> Value Paare gespeichert.

Es dürfen bei verschiedenen Listenelementen nicht der gleiche Name und RegType wiederverwendet werden.




Nach erfolgreichem hinzufügen, kann über einen Bonjour Browser direkt der neue RegType und Name gefunden und genutzt werden.

Available Bonjour services:

| Type | Name |
| --- | --- |
| _http._tcp. | RackStation |
| _http._tcp. | AXIS M1054 - 00408CE2D34F |
| _http._tcp. | shellyplug-s-7A16F8 |
| _ipp._tcp. | Brother MFC-J6710DW |
| _ipstest._tcp. | SymconTest |
| _pdl-datastream._tcp. | Brother MFC-J6710DW |
| _printer._tcp. | Brother MFC-J6710DW |
| _raop._tcp. | b63a5feab687@SymconPresentation |
| _rtsp._tcp. | AXIS M1054 - 00408CE2D34F H.264 |
| _sftp-ssh._tcp. | ALL3073 |
| _sftp-ssh._tcp. | SymBox-2 |

Service information:

3 TXT Records:

| Key | Value |
| --- | --- |
| Wert1 | Hallo |
| Unsinning | true |
| Wert2 | Welt |

IP Addresses

172.17.31.180:12345

# ZC_QueryService

Benötigt: IP-Symcon >= 5.3

```
array ZC_QueryService (int $InstanzID, string $Name, string
$RegType, string $Domain)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des anzufragenden DNS-SD Control |
| Name | Name des Services |
| RegType | Der RegType des Services |
| Domain | Domain des Services |

## RÜCKGABEWERT

Gibt alle Informationen zum gefundenen Service als Array zurück.

## BESCHREIBUNG

Fragt den Service des DNS-SD Controls mit der ID **InstanzID** und dem passenden Namen **Name**, RegType **RegType** und der Domain **Domain** ab.



Die Domain muss einen gültigen Wert haben und darf nicht leer sein

## BEISPIEL

```
ZC_QueryService(12345, "NAS", "_http._tcp", "local.");

// Beispielausgabe
/*
Array
(
[0] => Array
(
[Name] => NAS._http._tcp.local.
[Host] => NAS.local.
[Port] => 5000
[TXTRecords] => Array
(
```

```
[0] => vendor=Synology
[1] => model=RS818+
[2] => serial=xxxxx
[3] => version_major=6
[4] => version_minor=2
[5] => version_build=24922
[6] => admin_port=5000
[7] => secure_admin_port=5001
[8] => mac_address=xxxxxx
)

[IPv4] => Array
(
[0] => 192.168.1.105
)

[IPv6] => Array
(
[0] => fe60::cc4b:5ccf:fac7:5d82
)
)
)
*/
```

# ZC_QueryServiceType

Benötigt: IP-Symcon >= 5.3

```
array ZC_QueryServiceType (int $InstanzID, string $RegType, string
$Domain (optional))
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des anzufragenden DNS-SD Control |
| RegType | Der RegType des Services |
| Domain (optional) | Domain des Services |

## RÜCKGABEWERT

Gibt alle gefundenen Services als Array zurück.

## BESCHREIBUNG

Fragt die Services des DNS-SD Controls mit der ID **InstanzID** und dem passenden RegType **RegType** und der Domain **Domain** ab.



Die Domain ist optional und kann einen Leerstring enthalten

## BEISPIEL

```
ZC_QueryServiceType(12345, "_http._tcp", "");

// Beispielausgabe
/*
Array
(
[0] => Array
(
[Name] => NAS
[Type] => _http._tcp.
[Domain] => local.
)
)
*/
```

# ZC_QueryServiceTypes

Benötigt: IP-Symcon >= 5.3

```
array ZC_QueryServiceTypes (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des anzufragenden DNS-SD Control |

## RÜCKGABEWERT

Gibt alle Services und Informationen als Array zurück.

## BESCHREIBUNG

Fragt die Services des DNS-SD Controls mit der ID **InstanzID** ab und gibt diese als Array zurück.

## BEISPIEL

```
// Anfrage der registrierten Services des DNS-SD Control mit der ID
12345
ZC_QueryServiceTypes(12345);

// Beispielausgabe
/*
Array
(
[0] => Array
(
[Name] => _afpovertcp
[Type] => _tcp.local.
[Domain] => .
)


[1] => Array
(
[Name] => _airplay
[Type] => _tcp.local.
[Domain] => .
)
```

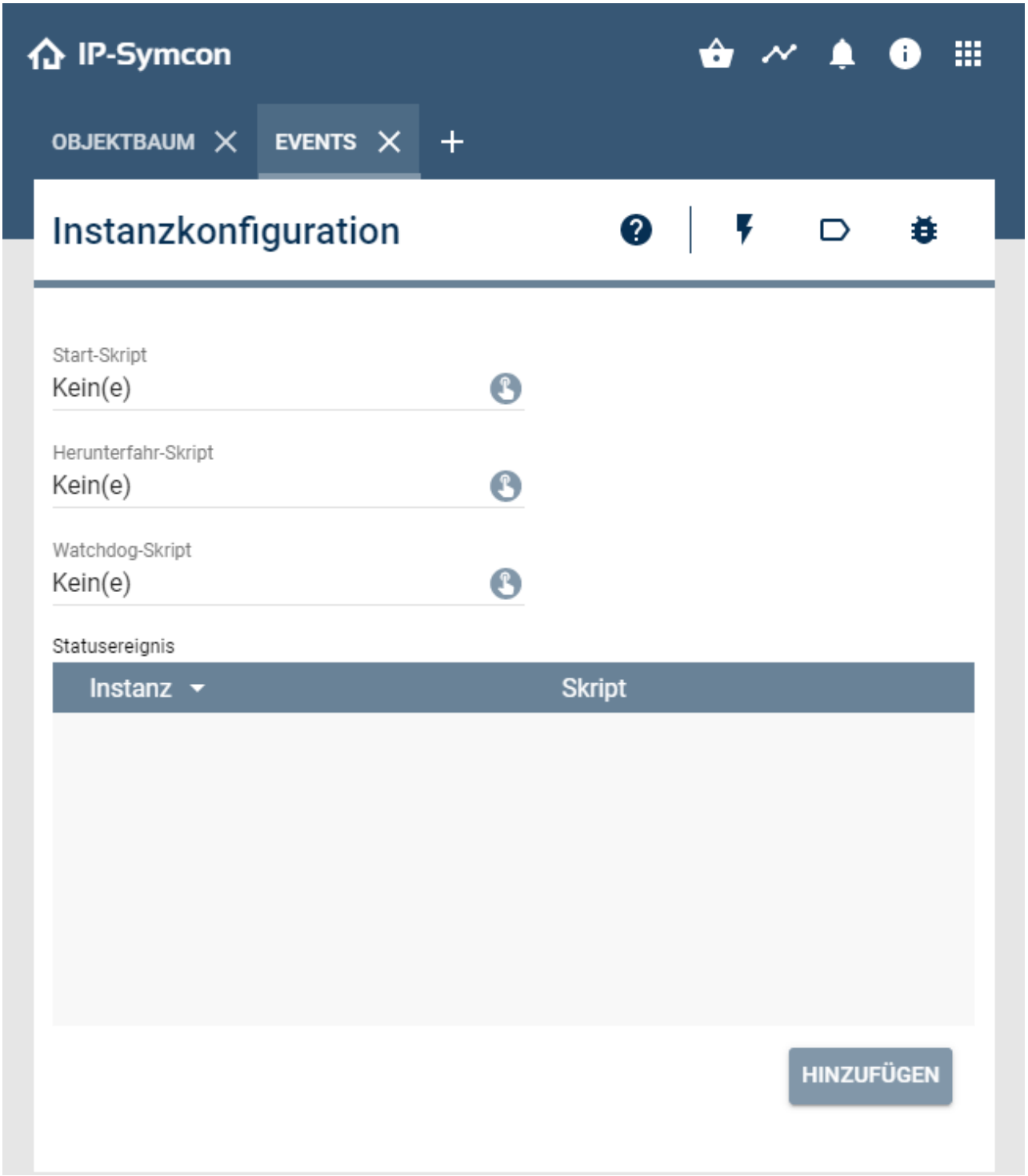
```
[2] => Array
(
[Name] => _airtame
[Type] => _tcp.local.
[Domain] => .
)


[3] => Array
(
[Name] => _apple-mobdev2
[Type] => _tcp.local.
[Domain] => .
)


[4] => Array
(
[Name] => _axis-video
[Type] => _tcp.local.
[Domain] => .
)
)
*/
```

# Event Control

Dieses Modul erlaubt dem User auf bestimmte Ereignisse in IP-Symcon zu reagieren und ein Skript zu starten. Es wird automatisch erstellt und steht dem Benutzer jederzeit zur Verfügung. Es kann im Objektbaum an folgender Stelle gefunden werden:
Kern Instanzen -> Events




Folgende Ereignisse stehen zur Verfügung:

| EREIGNIS | BESCHREIBUNG |
| --- | --- |
| Beim Start<br>(Start-Skript) | Nachdem IP-Symcon gestartet ist, wird dieses Ereignis aufgerufen. Zu diesem Zeitpunkt sollten bereits alle Schnittstellen verfügbar sein und alle Geräte geschaltet werden können. Es stehen folgende Systemvariablen innerhalb des Start-Skripts zur Verfügung. |
| Beim Herunterfahren<br>(Herunterfahr-Skript) | Kurz bevor IP-Symcon herunterfährt wird dieses Ereignis aufgerufen. An diesem Zeitpunkt sollten noch alle Geräte schaltbar sein. Es stehen folgende Systemvariablen innerhalb des Herunterfahr-Skripts zur Verfügung. |
| Profilgrenzüberschreitung<br>(Watchdog-Skript) | Sobald eine Variable (nur Integer/Float) außerhalb des Grenzbereiches vom dazugehörigen Variablenprofil kommt, wird dieses Ereignis aufgerufen. Eine Variable muss ein Profil haben um überwacht werden zu können. Es stehen folgende Systemvariablen innerhalb des Watchdog-Skripts zur Verfügung. |
| Bei Statusaktualisierung<br>(Statusereignis) | Wenn sich der Status einer Instanz aktualisiert, kann in einem angegebenen Skript auf diese Aktualisierung reagiert werden. Es stehen folgende Systemvariablen innerhalb des Statusergeinis-Skripts zur Verfügung. |



Dieses Modul sollte nicht gelöscht werden.
Falls es gelöscht wird, wird es beim nächsten Start wieder neu erstellt.

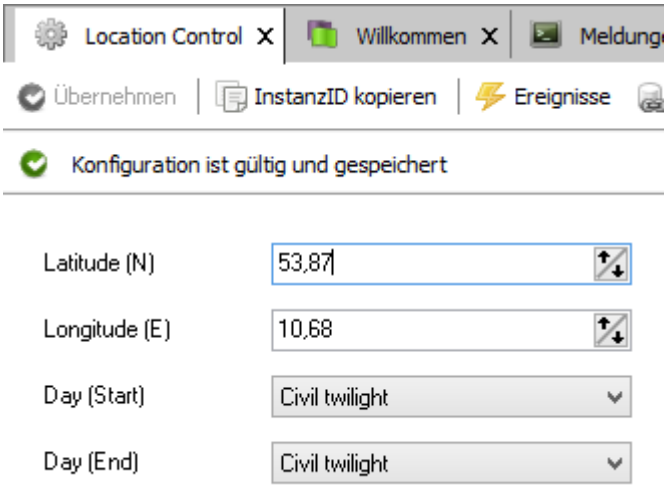
# Location Control

Benötigt: IP-Symcon >= 4.0

## BESCHREIBUNG

Das Location Control Modul bietet die Möglichkeit die Dämmerungszeitpunkte berechnen zu lassen. Dazu sind einfach Längen- und Breitengrad von dem zu berechnenden Ort notwendig.

## EINBINDUNG IN IP-SYMCON




In der Instanz "Location" ("Logische Baumansicht"->"Kern Instanzen") kann der Längengrad N und Breitengrad E, auf 2 Nachkommastellen genau, eingetragen werden. Eine Methode zur Bestimmung des Längen- und Breitengrades ist unter Tipps & Tricks zu finden.
Darüber werden dann die Dämmerungswerte für Sonnenaufgang, Sonnenuntergang, bürgerlich, nautisch und astronomisch berechnet.
Die beiden Auswahlmenüs Tagesbeginn und Tagesende bilden die Grenze, zwischen denen die boolsche Variable "Ist es Tag" auf True gesetzt wird.

| | | | | |
|---|---|---|---|---|
| 53993 | Location Control | Location Control | | N/A |
| 18456 | Ist es Tag | Boolean | True | 15:15:42 |
| 30060 | Sonnenaufgang | Integer | 28.08.2015 06:18:15 | 15:15:42 |
| 58785 | Sonnenuntergang | Integer | 27.08.2015 20:18:58 | 15:15:42 |
| 45869 | Ziviler Dämmerungsbeginn | Integer | 28.08.2015 05:38:39 | 15:15:42 |
| 52072 | Ziviles Dämmerungsende | Integer | 27.08.2015 20:59:09 | 15:15:42 |
| 10583 | Nautischer Dämmerungsbeginn | Integer | 28.08.2015 04:50:56 | 15:15:42 |
| 50634 | Nautisches Dämmerungsende | Integer | 27.08.2015 21:46:51 | 15:15:42 |
| 37357 | Astronomischer Dämmerungsstart | Integer | 28.08.2015 03:55:22 | 15:15:42 |
| 14038 | Astronomisches Dämmerungsende | Integer | 27.08.2015 22:42:25 | 15:15:42 |

Die Dämmerungswerte bestimmen auch den Zeitpunkt der Neuberechnung. Das heißt, dass beim Erreichen des Zeitpunkts für den Sonnenaufgang, der neue Wert für den nächsten Sonnenaufgang berechnet wird.

## BEISPIEL

Ein Beispiel wäre ein Skript zum Herunterfahren der Rolläden. Dieses wird automatisch gestartet, wenn es durch ein Ereignis ausgeführt wird, welches auf die Variablenänderung von "Sonnenuntergang" reagiert.



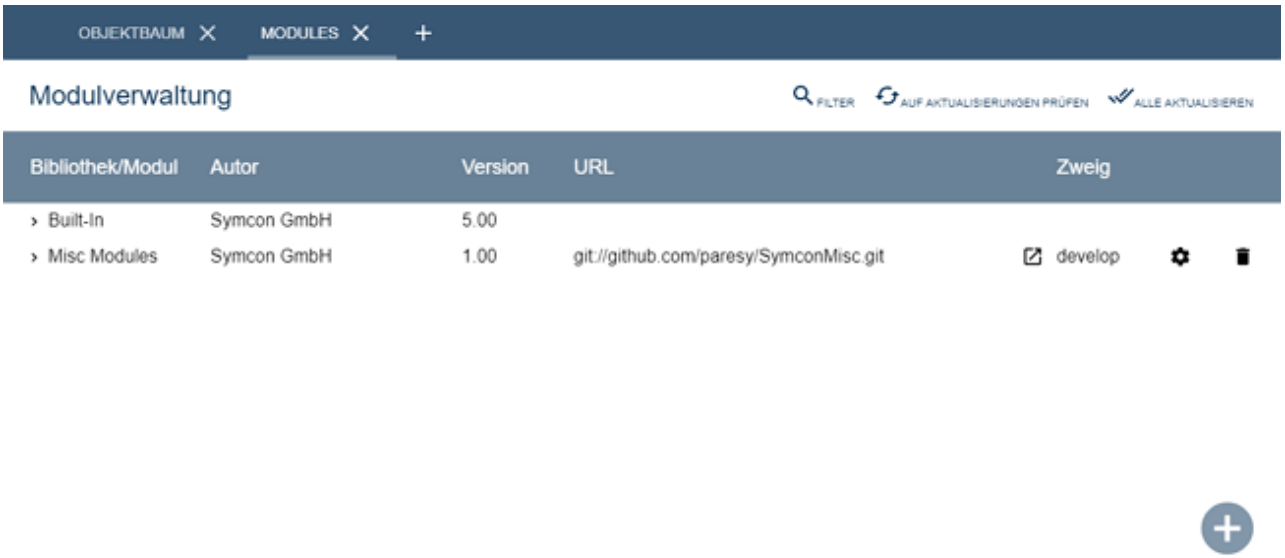
# Module Control

Benötigt: IP-Symcon >= 4.0

## BESCHREIBUNG

Das "Module Control" Modul bietet die Möglichkeit selbst erstellte oder von anderen Nutzern bereitgestellte PHP-Module einzubinden.

## EINBINDUNG IN IP-SYMCON



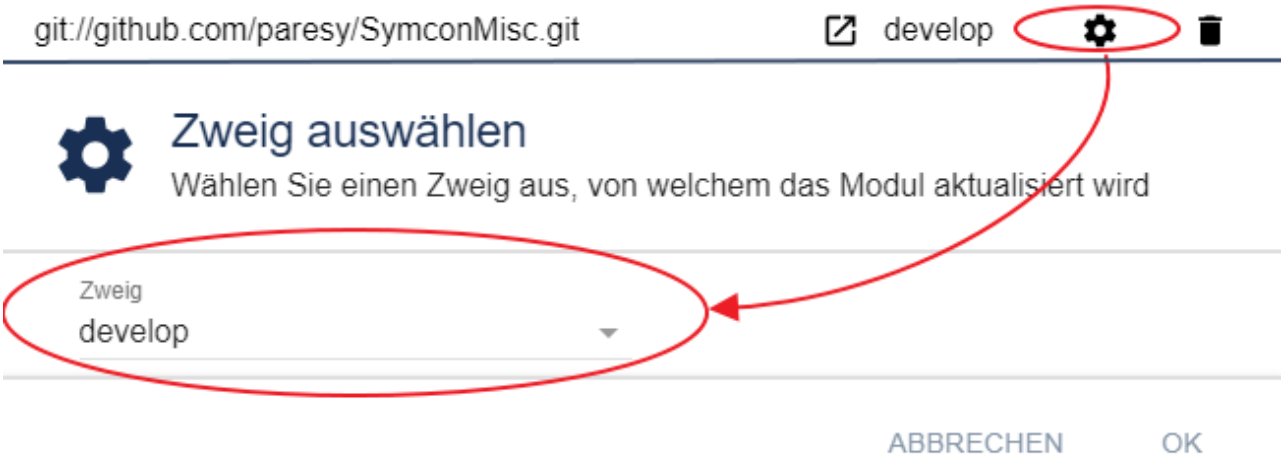

In der Instanz "Modules" ("Logische Baumansicht"->"Kern Instanzen") kann über "+" ein/-e Git-Repository Link/URL eingetragen werden. Über diesen wird das Modul dann installiert und steht sofort im "Instanz hinzufügen"-Dialog bereit. Welcher Link für welches Modul gebraucht wird, kann im Regelfall in der Dokumentation des Moduls oder Foreneintrag eingesehen werden. Sind die Daten unvollständig kann der jeweilige Modulentwickler über das Forum mithilfe einer privaten Nachricht angeschrieben werden.

## AKTUALISIERUNG

Der Button "Auf Aktualisierung prüfen" überprüft ob eine Aktualisierung für bereits hinzugefügte Module vorhanden ist. Wenn eine Aktualisierung möglich ist, kann über den erscheinenden Update Button das jeweilge Modul auf den aktuellsten Stand gebracht werden. Ist ein Modul auf dem aktuellsten Stand, wird stattdessen ein Haken angezeigt.

## ZWEIG WECHSELN

Das Zahnrad bei dem jeweiligen Modul ermöglicht es den Zweig des Repositories zu wechseln. Wie diese heißen liegt an der Benennung durch den Repository-Ersteller.




## EIGENE MODULE ENTWICKELN

Im Entwicklerbereich sind alle Informationen vorhanden um eigene Module zu entwickeln. Weitere Informationen sind direkt im SDK für PHP zu finden.

## BEISPIELE

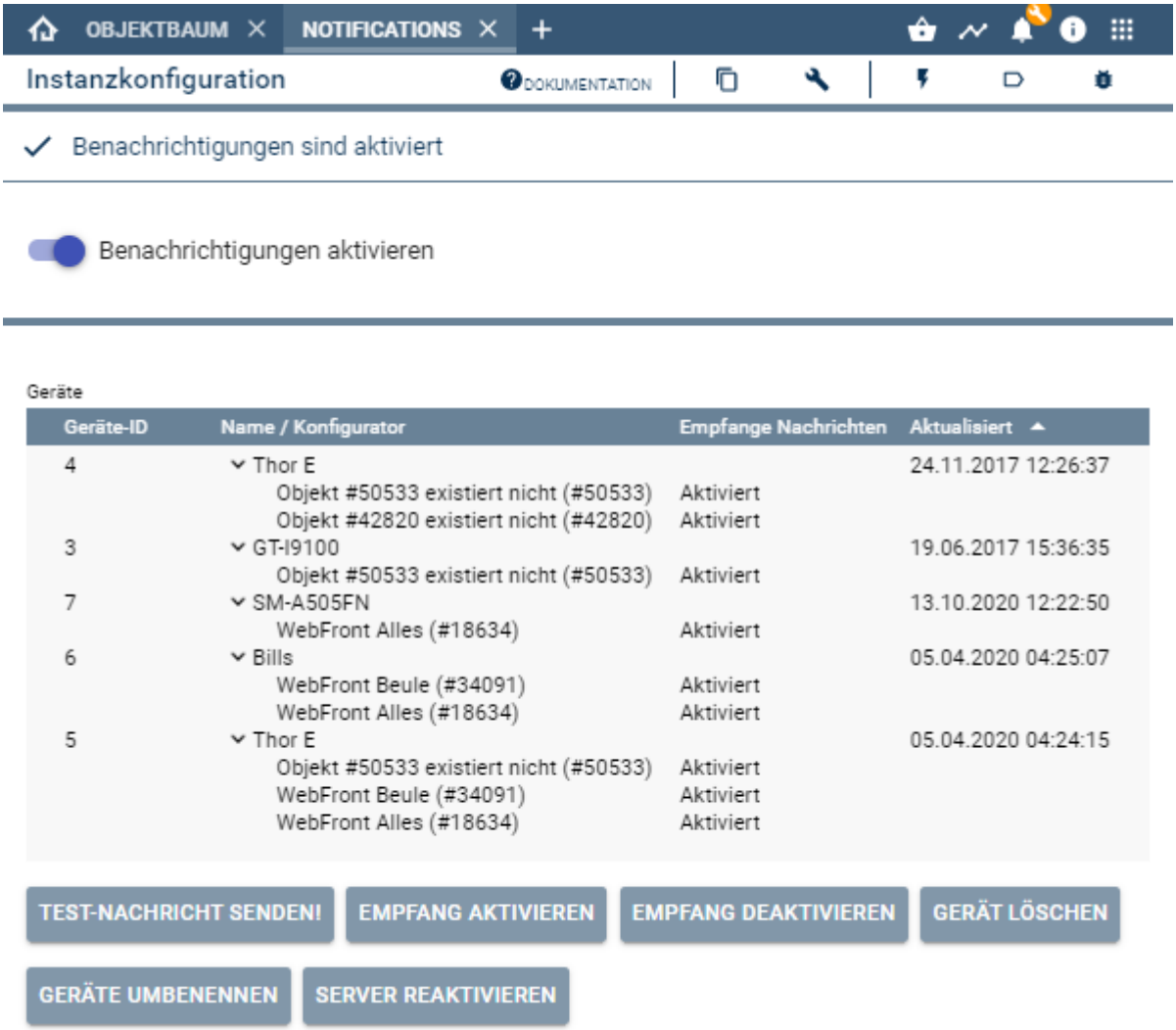
Im Forum ist eine Linksammlung vorhanden, welche von Nutzern entwickelte PHP-Module auflistet.

# Notification Control

Das Notification Control Modul gibt eine schnelle Übersicht über die zur Zeit für die Push-Benachrichtigung registrierten Geräte. In der Liste werden alle Geräte mit einer eindeutigen ID, dem änderbaren Namen und den zugeordneten Konfiguratoren angezeigt. Durch Auswählen eines Geräts und klicken der "Test-Nachricht senden" Schaltfläche kann eine Test-Nachricht an das gewählte Gerät verschickt werden. Push-Nachrichten können über WFC_PushNotification versendet werden.

Es ist pro Lizenz nur einem IP-Symcon Server möglich Push-Notifications zu senden. Weiterführende Informationen können unter dem Abschnitt Nutzungsrechte nachgeschlagen werden.

Um IP-Symcon den Zugriff auf bestimmte Funktionen (z.B. Push Notifications) außerhalb zu gewähren ist es gegebenenfalls nötig diese in der Firewall einzurichten. Weitere Informationen können unter Firewall gefunden werden.




## GERÄTE HINZUFÜGEN

Neue Geräte können nur über die mobilen Apps hinzugefügt werden. Dazu muss innerhalb der jeweiligen App die Konfiguration des Server erneut ausgeführt werden. Im letzten Schritt besteht die Möglichkeit bei der Auswahl des Konfigurators die Benachrichtigungen jeweils pro Konfigurator zu aktivieren/deaktivieren. Sobald das Gerät dann am Server registriert wurde, erscheint dieses im jeweils registrierten Konfigurator und in der Notification Control Übersicht.

## GERÄTE UMBENENNEN

Der Name eines Gerätes kann jederzeit geändert werden. Dazu muss in der Liste auf das kleine Pfeil-Icon geklickt werden.

## GERÄTE ENTFERNEN

Im Notification Control können Geräte vollständig von der Benachrichtigung entfernt werden. Sofern Geräte aus einem einzelnen Konfigurator entfernt werden sollen, kann dies im jeweiligen Konfigurator unter dem Reiter "Benachrichtigungen" eingestellt werden.

## SERVER REAKTIVIEREN

Die Push-Benachrichtigungen werden über den IP-Symcon versendet. Als zusätzliche Sicherheit authentifiziert sich der sendene Server, sodass er eindeutig als Versender identifiziert werden kann. Wenn der Server geändert, ausgetauscht oder neu installiert wird, ist ggf. eine Reaktivierung erforderlich. Diese Reaktivierung kann alle 24h vorgenommen und über den Button "Server reaktivieren" gestartet werden. Dabei wird zusätzlich per E-Mail über die Änderung benachrichtigt. Dies geschieht um einen Missbrauch einer Lizenz frühzeitig erkennen zu können.

### TÄGLICHE PUSH NACHRICHTEN

Als Standard sind pro Tag 250 Push Nachrichten vom Notification Control möglich. Per Spezialschalter (NotificationLimit) sind diese auf 1000 erhöhbar.



Eine Reaktivierung ist erforderlich, wenn die folgende Fehlermeldung erscheint:
"Ihr Server-Authentifizierungs-Token hat sich geändert. Bitte reaktivieren Sie Ihren Server."

# Presence Control

Benötigt: IP-Symcon >= 5.1
Das "Presence Control" bietet die Möglichkeit die Anwesenheit in einem Bereich auszuwerten. Hierfür werden Sensoren in zwei Gruppen eingeordnet. Impulsvariablen geben bei Aktivierung ein Signal ab, welches für eine gewisse Zeit die Anwesenheit bestätigt. Die andere Gruppe sind Zustandsvariablen, welche durch ihren Zustand die aktuelle Anwesenheit im Bereich auswerten.
In der Statusvariablen "Präsenz" wird die aktuelle Anwesenheit wiedergegeben. Diese ist gesetzt, solange mindestens eine Variable die Anwesenheit im Bereich bestätigt.
Alternativ kann die Anwesenheit manuell gesetzt werden. In diesem Fall wird die Anwesenheit für eine Totzeit auf dem gesetzten Wert gehalten. Während dieser Zeit werden alle Aktualisierungen von Impuls- und Zustandsvariablen ignoriert. Die Statusvariable "Detektion" informiert, ob die "Präsenz"-Variable aktuell durch die Totzeit nach einer manuellen Schaltung gesperrt ist oder aktiv ist, also durch die Impuls- und Zustandsvariablen berechnet wird.

## IMPULSVARIABLEN

| OPTION | BESCHREIBUNG |
|---|---|
| Variable | Boolean-Variable, die den Impuls enthält |
| Gültigkeit | Gültigkeit in Sekunden, für welche ein Impuls auf der Variablen die Anwesenheit im Bereich bestätigt |
| Invertieren? | Standardmäßig reagiert das Modul auf Variablenaktualisierungen auf true. Ist "Invertieren?" gesetzt, so reagiert das Modul stattdessen auf false |

Wird eine Impulsvariable auf true (oder false, falls "Invertieren?" gesetzt ist) aktualisiert, so wird die Anwesenheit für die Gültigkeitszeit bestätigt.



Wird die Gültigkeit verändert oder eine Impulsvariable entfernt, wirkt sich dies nicht auf laufende Impulse aus. Für laufende Impulse wird die Einstellung zum Zeitpunkt des Impulses verwendet.

## ZUSTANDSVARIABLEN

| OPTION | BESCHREIBUNG |
|---|---|
| Variable | Boolean-Variable, die den Zustand enthält |
| Invertieren? | Standardmäßig wird Anwesenheit bestätigt, sofern die Variable auf true steht. Ist "Invertieren?" gesetzt, so reagiert das Modul stattdessen auf false |

Steht eine Zustandsvariable auf true (oder false, falls "Invertieren?" gesetzt ist), so wird die Anwesenheit bestätigt.

## TOTZEIT

Hier kann die Totzeit in Sekunden nach einer manuellen Anpassung der Präsenz eingestellt werden. Während der Totzeit bleibt Präsent auf dem eingesetellten Wert

stehen. Impulse werden während dieser Zeit ignoriert. Nach der Totzeit wird die Präsenz wieder wie gewohnt bestimmt.



Wird die Totzeit verändert, wird diese erst beim nächsten manuellen Anpassen angewandt. Für laufende Totzeiten wird die Einstellung zum Zeitpunkt der manuellen Anpassung verwendet.

# PC_Enter

Benötigt: IP-Symcon >= 5.1

```
boolean PC_Enter (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Presence-Control-Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Der Befehl setzt die Anwesenheitvariable der Instanz mit der ID **InstanzID** auf true. Darüber hinaus wird die Anwesenheit für die im Modul eingestellte Totzeit festgesetzt und ändert sich nicht durch Status- oder Impulsvariablen. Während der Totzeit werden Impulse auf Impulsvariablen und Änderungen von Zustandsvariablen komplett ignoriert. Nach der Totzeit wird die Anweseheit initial auf Basis der Zustandsvariablen gesetzt.

## BEISPIEL

```
// Setzt die Anwesenheit der Presence Control Instanz 12345 auf
"true"
PC_Enter(12345);
```

# PC_Leave

Benötigt: IP-Symcon >= 5.1

```
boolean PC_Leave (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Presence-Control-Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Der Befehl setzt die Anwesenheitvariable der Instanz mit der ID **InstanzID** auf false. Darüber hinaus wird die Anwesenheit für die im Modul eingestellte Totzeit festgesetzt und ändert sich nicht durch Status- oder Impulsvariablen. Während der Totzeit werden Impulse auf Impulsvariablen und Änderungen von Zustandsvariablen komplett ignoriert. Nach der Totzeit wird die Anweseheit initial auf Basis der Zustandsvariablen gesetzt.

## BEISPIEL

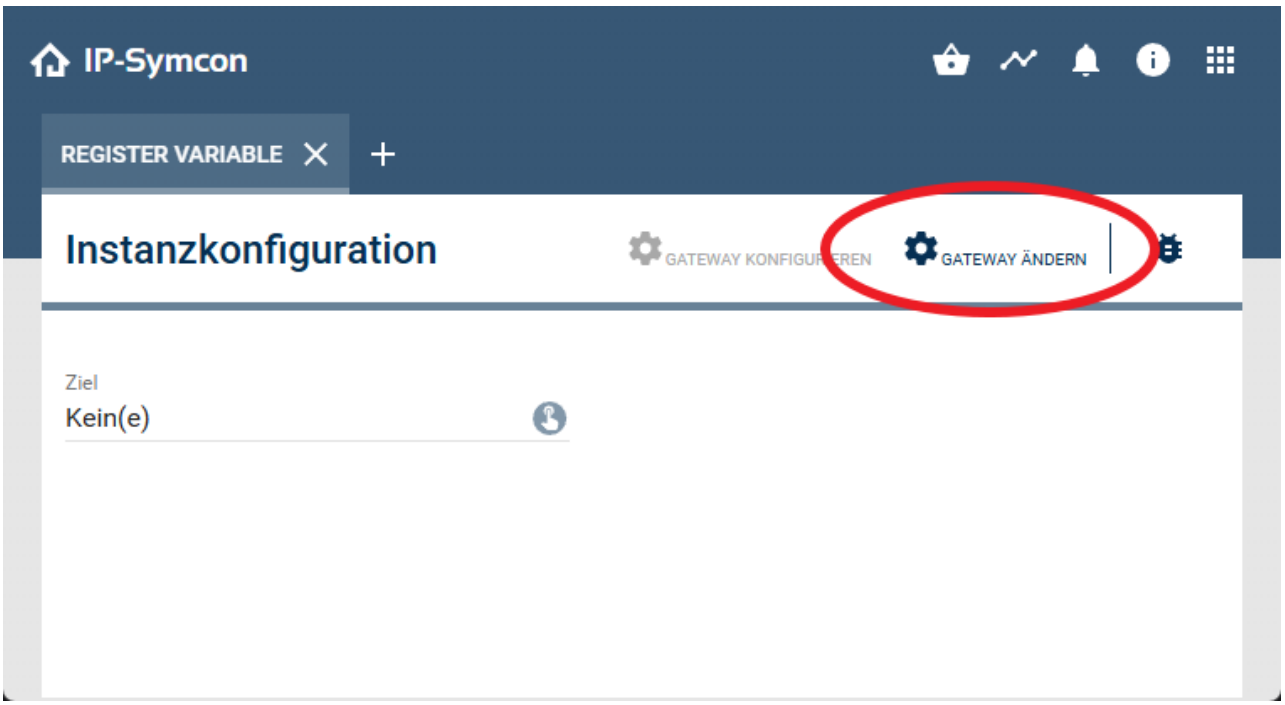
```
// Setzt die Anwesenheit der Presence Control Instanz 12345 auf
"false"
PC_Leave(12345);
```

# RegisterVariable

RegisterVariable stellt eine Datenweiterleitungs- und -verarbeitungsschnittstelle zur Verfügung.

## KONFIGURATION

Das RegisterVariable Modul verbindet sich nicht automatisch mit einer übergeordneten Instanz. Es muss dafür eine übergeordnete Instanz ausgewählt oder eine Neue erstellt werden. Es können auch weitere Splitter (z.B. Cutter) hinzugefügt werden, um die ankommenden Daten vor der Auswertung direkt synchronisieren zu lassen.
Die übergeordnete Instanz kann unkompliziert über die Funktion "Gateway ändern" oben in der Konfiguration angepasst werden.




Die Ausführung der Skripte erfolgt **seriell**. Das Bedeutet, dass die Skripte nicht mit Semaphoren geschützt werden müssen.

## DATENVERARBEITUNG

Werden von der übergeordneten Instanz (Kommunikationsschnittstelle) neue Daten empfangen, wird das Ziel-Skript der RegisterVariable-Instanz ausgeführt. In diesem stehen die empfangenen Daten über $_IPS['VALUE'] zur Verfügung. Sollte die übergeordnete Instanz ein ServerSocket sein, stehen zusätzlich noch die Daten $_IPS['CLIENTIP'] und $_IPS['CLIENTPORT'] zur Verfügung.

> Skripte, die durch die RegisterVariable ausgeführt werden, beinhalten diese Systemvariablen

Es gibt Datenquellen, bei denen erst mehrere empfangene Datensendungen eine

sinnvolle Auswertung ermöglichen. Daher müssen die empfangenen Daten zwischengespeichert werden. Da Probleme auftreten können, wenn man binäre Daten in einer normalen String-Variable in IP-Symcon speichert (defekte IP-Symcon-Konfiguration), gibt es die Funktion RegVar_SetBuffer(integer $InstanzID, string $Puffer), die Daten im einen zur RegisterVariable-Instanz gehörigen Puffer speichert. Die im Puffer gespeicherten Daten lassen sich mit der Funktion RegVar_GetBuffer(integer $InstanzID) auslesen. Mit der Funktion RegVar_SendText(integer $InstanzID, string $Text) lassen sich Datenstrings über die Kommunikationsschnittstelle senden. Intern wird hierbei die passende Sendefunktion, wie z.B. COMPort_SendText, ausgeführt.

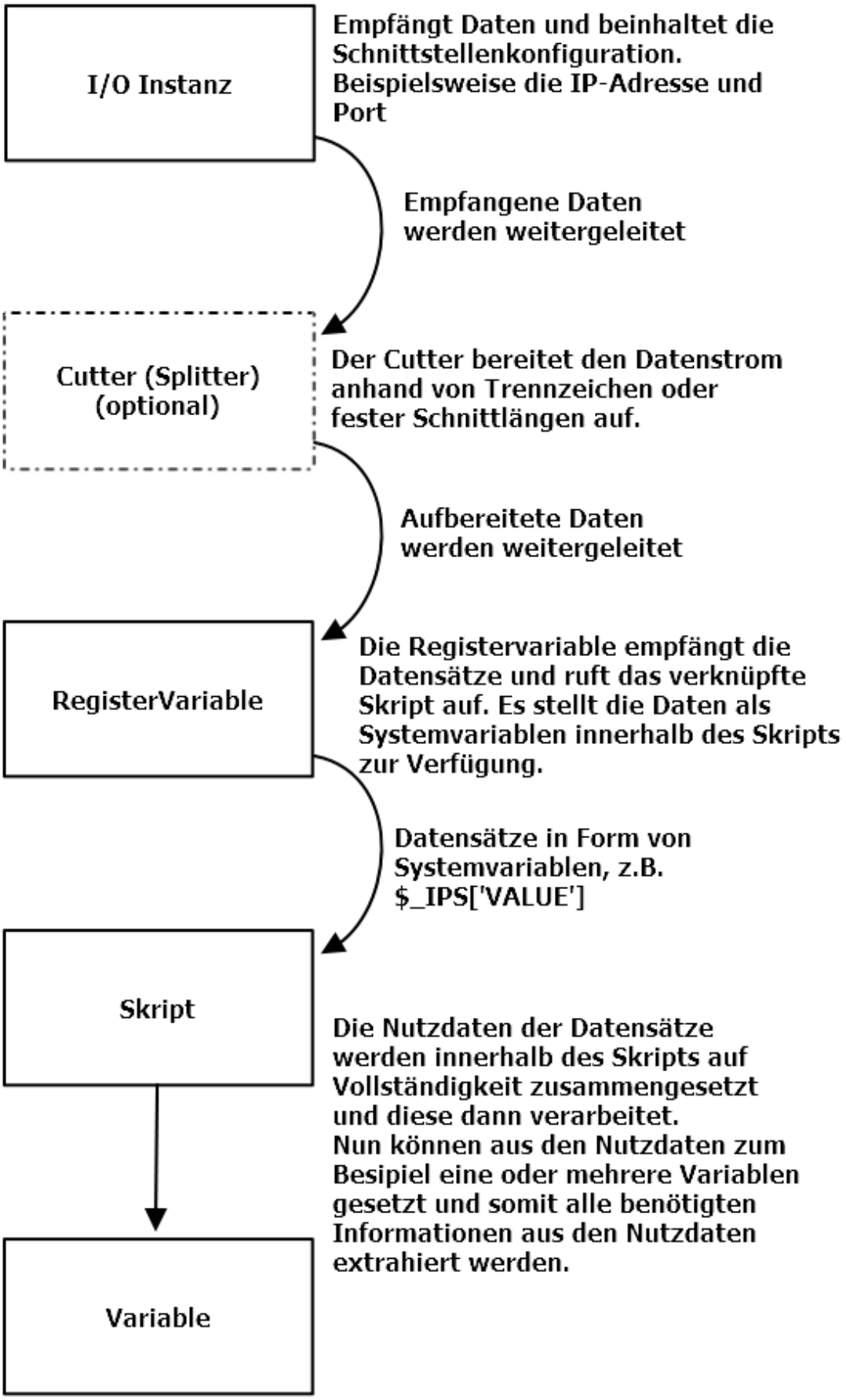
Der Puffer wird nach einem Neustart von IP-Symcon gelöscht!

Es soll KEINE VARIABLE als Puffer genutzt werden. Andernfalls kann es zu Instabilitäten innerhalb von IP-Symcon kommen und im schlimmsten Fall eine komplett zerstörte Konfiguration verursachen. Da sich dieses Problem auch inkrementell auf den Backup Ordner auswirken kann, ist dann unter Umständen nicht einmal eine Backup Konfiguration vorhanden. Es sollen die internen Puffer der Instanz genutzt werden, um Daten zwischenzuspeichern.
RegVar_SetBuffer / RegVar_GetBuffer

## BEISPIELE

### AUFBAU

Das folgende Bild zeigt den typischen Aufbau und die Schritte für eine Registervariable. So kann die I/O Instanz in Form eines WWW-Readers zum Beispiel eine Webseite einlesen, der Cutter den Datenstrom aufbereiten und und die Registervariable im Skript einzelne Werte einer Homepage nutzbar machen und abspeichern.


I/O Instanz
Empfängt Daten und beinhaltet die Schnittstellenkonfiguration. Beispielsweise die IP-Adresse und Port
Empfangene Daten werden weitergeleitet
Cutter (Splitter) (optional)
Der Cutter bereitet den Datenstrom anhand von Trennzeichen oder fester Schnittlängen auf.
Aufbereitete Daten werden weitergeleitet
RegisterVariable
Die Registervariable empfängt die Datensätze und ruft das verknüpfte Skript auf. Es stellt die Daten als Systemvariablen innerhalb des Skripts zur Verfügung.
Datensätze in Form von Systemvariablen, z.B. $_IPS['VALUE']
Skript
Die Nutzdaten der Datensätze werden innerhalb des Skripts auf Vollständigkeit zusammengesetzt und diese dann verarbeitet.
Nun können aus den Nutzdaten zum Besipiel eine oder mehrere Variablen gesetzt und somit alle benötigten Informationen aus den Nutzdaten extrahiert werden.
Variable

## SKRIPT

Folgendes Beispiel verkettet empfangene Daten und gibt durch ; separierte Datensätze bei Vervollständigung aus:

```
// wenn das Skript von einer RegisterVariable-Instanz aus aufgerufen
worden ist
if ($_IPS['SENDER'] == "RegisterVariable") {
// bereits im Puffer der Instanz vorhandene Daten in $data kopieren
$data = RegVar_GetBuffer($_IPS['INSTANCE']);
// neu empfangene Daten an $data anhängen
$data .= $_IPS['VALUE'];

// wenn das Trennzeichen ; in $data gefunden worden ist
if (strpos($data, ';')) {
// $data in durch ; separierte Datensätze zerlegen
$datasets = explode(';', $data);

// alle nicht durch ; terminierten Datensätze ausgeben
for ($i = 0; $i < count($datasets) - 1; $i++) {
echo "empfangener Datensatz: ".$datasets[$i]."\n";
}

// $data auf den Inhalt des letzten (unvollständigen) Datensatzes
setzen
$data = $datasets[count($datasets) - 1];
}

// Inhalt von $data im Puffer der RegisterVariable-Instanz speichern
RegVar_SetBuffer($_IPS['INSTANCE'], $data);
}
```

Folgendes Beispiel verkettet empfangene Daten und gibt Blöcke von einer Länge von genau 16 Zeichen aus:

```
// wenn das Skript von einer RegisterVariable-Instanz aus aufgerufen
worden ist
if ($_IPS['SENDER'] == "RegisterVariable") {
// bereits im Puffer der Instanz vorhandene Daten in $data kopieren
$data = RegVar_GetBuffer($_IPS['INSTANCE']);
// neu empfangene Daten an $data anhängen
$data .= $_IPS['VALUE'];

// wenn $data mindestens 16 Zeichen lang ist
if (strlen($data) >= 16) {
// $data in Blöcke von bis zu 16 Zeichen zerlegen
$datasets = str_split($data, 16);
// $data leeren
$data = "";

// alle Datensätze durcharbeiten
for ($i = 0; $i < count($datasets); $i++) {
// vollständige Datensätze (genau 16 Zeichen lang) ausgeben
if (strlen($datasets[$i]) == 16) {
```

```
echo "empfangener Datensatz: ".$datasets[$i]."\n";
} else {
// Unvollständige Datensätze in $data schreiben
$data = $datasets[$i];
}
}
}

// Inhalt von $data im Puffer der RegisterVariable-Instanz speichern
RegVar_SetBuffer($_IPS['INSTANCE'], $data);
}
```

## TIPPS & TRICKS

### SICH ALLE ÜBERGEBEN VARIABLE ANZEIGEN LASSEN

```
unset($_SERVER['argv']);
unset($_SERVER['argc']);

foreach ($_SERVER as $key => $value) {
IPS_LogMessage(IPS_GetName($IPS_INSTANCE), "Key: $key; Value:
$value");
}
```

# RegVar_GetBuffer

```
boolean RegVar_GetBuffer (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |

## RÜCKGABEWERT

String des aktuellen Puffers

## BESCHREIBUNG

Liest den internen Puffer der Instanz mit ID **InstanzID**.

## BEISPIEL

```
$buf = RegVar_GetBuffer(12345);
$buf .= $_IPS['VALUE']; //konkatenieren

//verarbeiten ...
RegVar_SetBuffer(12345, $buf); //Restpuffer zurückschreiben
```

# RegVar_SendEvent

Benötigt: IP-Symcon >= 4.1

```
boolean RegVar_SendEvent (int $InstanzID, int $ReportID, string
$Text)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der RegisterVariable-Instanz |
| ReportID | Der Datensatztyp als ReportID |
| Text | Zu sendender Puffer |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet einen String **Text** über die Instanz mit der ID **InstanzID** als Datensatztyp **ReportID**.



RegVar_SendEvent kann nur in Verbindung zu einer HID-Instanz verwendet werden

## BEISPIEL

```
// Sendet ein Event "Hallo Welt" an die übergeordnete Instanz mit
der ReportID 0
RegVar_SendEvent(12345, 0, "Hallo Welt");
```

# RegVar_SendPacket

Benötigt: IP-Symcon >= 4.1

```
boolean RegVar_SendPacket (int $InstanzID, string $Text, string
$ClientIP, string $ClientPort)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der RegisterVariable-Instanz |
| Text | Zu sendender Puffer |
| ClientIP | IP-Adresse an die gesendet werden soll |
| ClientPort | Port an den gesendet werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet ein Datenpaket **Text** über die Instanz mit der ID **InstanzID**. Zusätzlich werden noch die IP-Adresse **ClientIP** und der Port **ClientPort** angegeben.



RegVar_SendPacket kann nur in Verbindung zu einer ServerSocket-Instanz verwendet werden

## BEISPIEL

```
// Sendet via ServerSocket an die IP:Port 192.168.1.2:1234 das Paket
"Hallo Welt"
RegVar_SendPacket(12345, "Hallo Welt", "192.168.1.2", 1234);
```

# RegVar_SendText

```
boolean RegVar_SendText (int $InstanzID, string $Text)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Text | Zu sendender Puffer |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Sendet einen String **Text** über die Instanz mit der ID **InstanzID**

## BEISPIEL

```
RegVar_SendText(12345, "Hallo Welt");
```

# RegVar_SetBuffer

```
boolean RegVar_SetBuffer (int $InstanzID, string $Puffer)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Puffer | Daten, die in den Puffer geschrieben werden sollen |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den internen Puffer der Instanz mit ID **InstanzID** auf Wert **Puffer**

## BEISPIEL

```
$buf = RegVar_GetBuffer(12345);
$buf .= $_IPS['VALUE']; //konkatenieren

//verarbeiten ...
RegVar_SetBuffer(12345, $buf); //Restpuffer zurückschreiben
```

# Skin Control

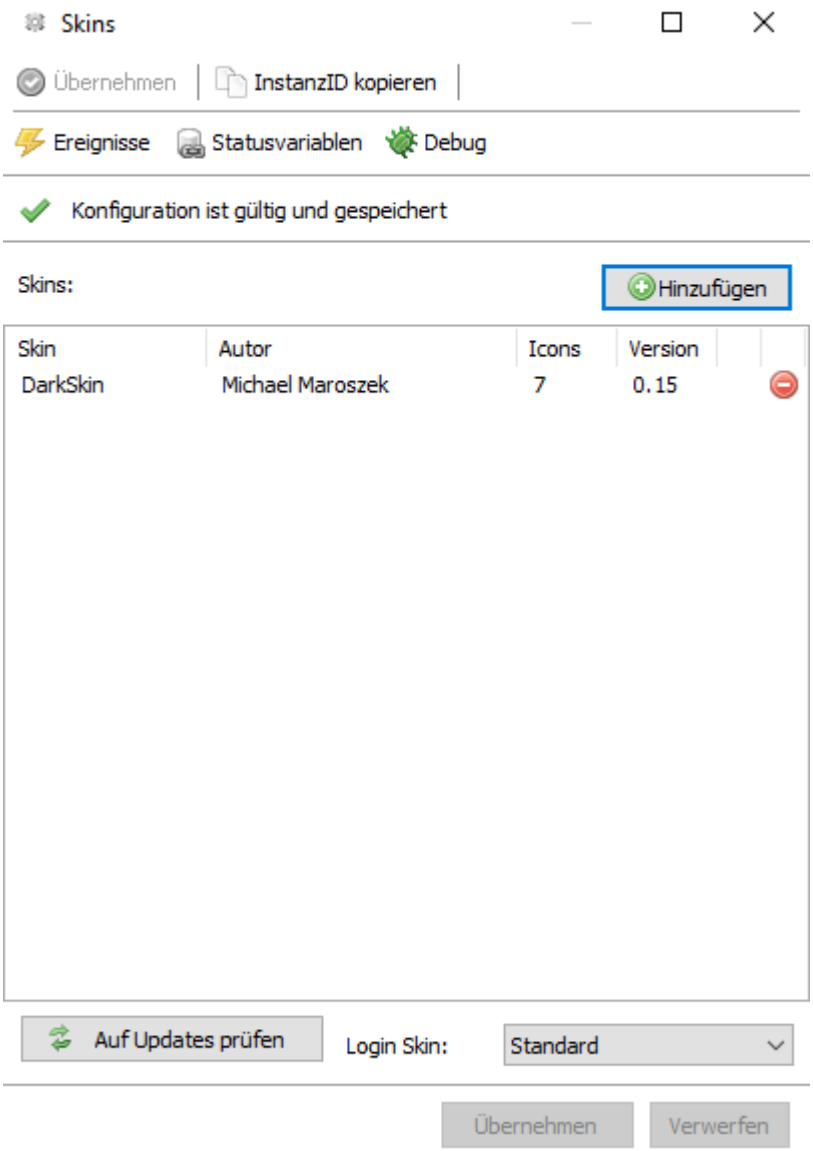
Benötigt: IP-Symcon >= 4.0

## BESCHREIBUNG

Das "Skin Control" Modul bietet die Möglichkeit selbst erstellte oder von anderen Nutzern bereitgestellte Skins einzubinden.

## EINBINDUNG IN IP-SYMCON

In der Instanz "Modules" ("Logische Baumansicht"->"Kern Instanzen") kann über "Hinzufügen" ein Git-Repository Link eingetragen werden. Über diesen wird der Skin dann installiert und steht sofort zur Verfügung.




## AKTUALISIERUNG

Der Button "Auf Updates prüfen" überprüft ob eine Aktualisierung für bereits

hinzugefügte Skins vorhanden ist. Wenn eine Aktualisierung möglich ist, kann über den erscheinenden Update Button der jeweilge Skin auf den aktuellsten Stand gebracht werden. Ist ein Skin auf dem aktuellsten Stand, wird stattdessen ein grüner Haken angezeigt.

## LOGIN SKIN AUSWÄHLEN

Seperat zu den jeweiligen WebFront-Skins, welche im jeweiligen WebFront-Konfigurator ausgewählt werden, kann auch der Skin des Login Bereichs des WebFronts geändert werden. Dieser kann via Auswahl im Skin-Control gesetzt werden.

## EIGENE SKINS ENTWICKELN

Informationen um eigene Skins zu entwickeln befinden sich hier:
SDK für Skins

## BEISPIELE

Folgende Skins werden von uns gepflegt und bereitgestellt:
DarkSkin: github.com/symcon/SkinDark
LightSkin: github.com/symcon/SkinLight

# SSDP Control

Das SSDP (Simple Service Discovery Protocol) Control Modul ermöglicht das Suchen nach UPnP Geräten im Netzwerk.
SSDP ist ein auf UDP basierendes Protokoll mit dem sich UPnP Geräte im Netzwerk identifizieren und nur die wichtigesten Informationen mitteilen. Dieses beinhalten zum Beispiel Gerätenamen, Gerätetyp oder eine URL zur Beschreibung des Geräts.



Weiterführende Informationen sind beispielsweise bei Wikipedia zu finden.

## INSTALLATION

Das SSDP Control wird automatisch als Kern-Instanz installiert. Dieses kann im Objektbaum unter Kern Instanzen gefunden werden.

## KONFIGURATION

Das Modul benötigt keine weitere Konfiguration. Es nutzt standardmäßig den Port 1900 zur Suche von Geräten.

# YC_SearchDevices

Benötigt: IP-Symcon >= 5.3

```
array YC_SearchDevices (int $InstanzID, string $SuchZiel)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des anzufragenden SSDP Control |
| SuchZiel | . |

## RÜCKGABEWERT

Gibt alle Informationen zu gefundenen Geräten als Array zurück.

## BESCHREIBUNG

Schickt eine Anfrage über das SSDP Control mit der ID **InstanzID** und dem passenden Suchziel **SuchZiel**.
---param2---

| SUCHZIEL | BESCHREIBUNG |
|---|---|
| "ssdp:all" | Alle Geräte welche auf das SSDP antworten |
| "upnp:rootdevice" | Alle Geräte mit dem SubType "rootdevice" |
| "uuid:<UID>" | Alle Geräte mit der eindeutigen ID **UID** |
| "urn:<URN>" | Alle Geräte mit der eindeutigen URN **URN** |

## BEISPIEL

```
YC_SearchDevices(27253, "ssdp:all");

// Beispielausgabe
var_dump(YC_SearchDevices(27253, "ssdp:all"));
/*
array(97) {
[...]
[11]=>
array(9) {
["CacheControl"]=>
string(14) "max-age = 1800"
["Date"]=>
string(0) ""
```

```
["Ext"]=>
string(0) ""
["Location"]=>
string(52) "http://172.17.31.142:1400/xml/device_description.xml"
["Server"]=>
string(38) "Linux UPnP/1.0 Sonos/53.2-70210 (ZPS5)"
["ST"]=>
string(47) "urn:schemas-upnp-org:service:DeviceProperties:1"
["USN"]=>
string(78) "uuid:RINCON_000E58507DEA01400::urn:schemas-upnp-
org:service:DeviceProperties:1"
["Fields"]=>
array(5) {
[0]=>
string(52) "X-RINCON-HOUSEHOLD: HHID_tZkieK2lkPYES3tycishsbwfXIz"
[1]=>
string(21) "X-RINCON-BOOTSEQ: 636"
[2]=>
string(20) "X-RINCON-WIFIMODE: 0"
[3]=>
string(19) "X-RINCON-VARIANT: 0"
[4]=>
string(83) "HOUSEHOLD.SMARTSPEAKER.AUDIO:
HHID_tZkieK2lkPYES3tycishsbwfXIz.hPqvaa7W35eMY2P3XidJ"
}
["IPv4"]=>
string(13) "172.17.31.142"
)
[...]
}
*/
```

# System Informationen

Über die Systemfunktionen können Informationen über das System und die vorhandene Hardware ausgelesen werden. Die gesammelten Informationen können zur Visualisierung oder anderweitig in Skripten verwendet werden.

# Sys_GetBattery

```
array Sys_GetBattery ()
```

## RÜCKGABEWERT

Array mit Informationen zur Batterie

## BESCHREIBUNG

Gibt Informationen über das Vorhandensein einer Batterie, ob sie in Gebrauch ist und wieviel Prozent der Batterie noch vorhanden sind. Die Angaben der Restlaufzeit und maximalen Batterielaufzeit werden in Sekunden angegeben.

## BEISPIEL

```
print_r(Sys_GetBattery());
/*
// Beispielausgabe auf einem Desktop PC ohne Batterie
// HasBattery ist verfügbar mit IP-Symcon 7.0+
Array
(
[HasBattery] =>
[OnBattery] =>
[IsCharging] =>
[BatteryLevel] => -1
[BatteryRemainingTime] => -1
[BatteryMaxTime] => -1
)
*/
```

# Sys_GetCPUInfo

```
array Sys_GetCPUInfo ()
```

## RÜCKGABEWERT

Array mit Informationen zur CPU-Auslastung

## BESCHREIBUNG

Ab Version 4.0
Liefert für jede CPU die aktuelle Last (CPU_*) und die momentane Durchschnittlast für alle CPU Kerne (CPU_AVG).
Bis Version 3.4
Liefert für jede CPU die aktuelle Last (CPU_*) und die Durchschnittlast der letzten 60 Sekunden für alle CPU Kerne (CPU_AVG). Die durchschnittliche Last wird berechnet, indem jeder Sekunde ein Messwert der aktuellen Last genommen und abgespeichert wird. Die durchschnittliche Last wird berechnet, indem die letzten 60 Messwerte addiert und durch 60 geteilt werden.

## BEISPIEL

```
print_r(Sys_GetCPUInfo());

/*
Array
(
[CPU_0] => 3
[CPU_AVG] => 3
)
*/
```

# Sys_GetHardDiskInfo

```
array Sys_GetHardDiskInfo ()
```

## RÜCKGABEWERT

Array mit Informationen zu allen Festplatten

## BESCHREIBUNG

Liefert Informationen zu den vorhandenen Festplatten im System inklusive der Größe und dem bereits verbrauchten Speicher.

## BEISPIEL

```
//Ab Version 4.0
print_r(Sys_GetHarddiskInfo());
/* Beispielausgabe:
Array
(
[0] => Array
(
[LETTER] => c:\
[LABEL] =>
[TOTAL] => 53684989952
[FREE] => 23275171840
)

[NUMDRIVES] => 1
)
*/


// Bis Version 3.4
print_r(Sys_GetHarddiskInfo());
/* Beispielausgabe:
Array
(
[HDD0] => Array
(
[LETTER] => c:\
[LABEL] =>
[TOTAL] => 53684989952
[FREE] => 23275171840
)
```

```
[NUMDRIVES] => 1
)
*/
```

# Sys_GetMemoryInfo

```
array Sys_GetMemoryInfo ()
```

## RÜCKGABEWERT

Array mit Informationen zum Gesamtspeicherverbrauch und virtuellen Speicherverbrauch

## BESCHREIBUNG

Gibt Informationen zum Speicherverbrauch des Betriebssystems.

## BEISPIEL

```
print_r(Sys_GetMemoryInfo());

/*
Array
(
[TOTALPHYSICAL] => 1072467968
[AVAILPHYSICAL] => 526647296
[TOTALPAGEFILE] => 2420019200
[AVAILPAGEFILE] => 1386422272
[TOTALVIRTUAL] => 2147352576
[AVAILVIRTUAL] => 1906978816
)
*/
```

# Sys_GetNetworkInfo

```
array Sys_GetNetworkInfo ()
```

## RÜCKGABEWERT

Array mit Informationen zu allen Netzwerkadaptern

## BESCHREIBUNG

Liefert Informationen über alle Netzwerkadapter

## BEISPIEL

```
print_r(Sys_GetNetworkInfo());
/*
// Beispielausgabe

Array
(
[0] => Array
(
[InterfaceIndex] => 10
[IP] => 192.168.1.2
[MAC] => 00:A0:03:AD:14:BD
[Description] => Siemens I BT USB Remote NDIS Network Device
[Speed] => 9728000
[InTotal] => 40236
[OutTotal] => 247248
)

[1] => Array
(
[InterfaceIndex] => 13
[IP] => 172.12.1.200
[MAC] => 00:A0:03:AD:14:BD
[Description] => Qualcomm Atheros AR8151 PCI-E Gigabit Ethernet
Controller (NDIS 6.30)
[Speed] => 1000000000
[InTotal] => 169987950
[OutTotal] => 86029648
)
)
*/
```

# Sys_GetProcessInfo

```
array Sys_GetProcessInfo ()
```

## RÜCKGABEWERT

Array mit Informationen, wie z.B. Speicherverbrauch und Anzahl der Threads

## BESCHREIBUNG

Gibt Informationen zum Prozess von IP-Symcon.

## BEISPIEL

```
print_r(Sys_GetProcessInfo());
/*
Array
(
[IPS_HANDLECOUNT] => 635
[IPS_NUMTHREADS] => 53
[IPS_VIRTUALSIZE] => 240373760
[IPS_WORKINGSETSIZE] => 32706560
[IPS_PAGEFILE] => 52719616
[PROCESSCOUNT] => 53
)
*/
```

# Sys_GetSpooler

```
array Sys_GetSpooler ()
```

## RÜCKGABEWERT

Array mit Informationen zu allen Druckerwarteschlagen und den im Druck befindlichen Dokumenten

## BESCHREIBUNG

Gibt Informationen über alle Druckerwarteschlangen aus



Diese Funktion funktioniert nur unter Windows

## BEISPIEL

```
print_r(Sys_GetSpooler());
```

# Sys_GetURLContent

```
string Sys_GetURLContent (string $URL)
```

## PARAMETERLISTE

| | |
|---|---|
| URL | Vollständige URL |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis den Inhalt der Webseite (auch Binärcode), andernfalls **FALSE**.

## BESCHREIBUNG

Liest den Inhalt der **URL** aus und gibt ihn als Rückgabewert zurück.



Die Funktion hat einen Standard-Timeout von 10 Sekunden. Soll dieser geändert werden, kann die Funktion Sys_GetURLContentEx genutzt werden

## BEISPIEL

```
echo Sys_GetURLContent("http://www.google.de");
```

# Sys_GetURLContentEx

```
string Sys_GetURLContentEx (string $URL, array $Parameter)
```

## PARAMETERLISTE

| | |
|---|---|
| URL | Vollständige URL |
| Parameter | Array mit Index => Value Paaren |

| INDEX | TYP | BESCHREIBUNG |
|---|---|---|
| Timeout | Integer | **Default = 10000**. Timeout in Millisekunden |
| AuthUser | String | **Default = ""**. Benutzername für Basis-Authentifizierung |
| AuthPass | String | **Default = ""**. Passwort für Basis-Authentifizierung |
| VerifyPeer | Boolean | **Default = true**. Verifiziere, dass das SSL Zertifikat des Hosts korrekt ist (nur sichtbar falls Benutze SSL aktiv ist) |
| VerifyHost | Boolean | **Default = true**. Verifiziere, dass die Host URL aus dem SSL Zertifikat mit dem Host aus der Eigenschaft übereinstimmt (nur sichtbar falls Benutze SSL aktiv ist) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis den Inhalt der Webseite (auch Binärcode), andernfalls **FALSE**.

## BESCHREIBUNG

Liest den Inhalt der **URL** aus und gibt ihn als Rückgabewert zurück. Dabei können einige **Parameter** für Timeouts, Authentifizierung und Proxy vorgenommen werden.

## BEISPIEL

```
// Rückgabe von http://www.google.de mit Basis-Authentifizierung
"test", Passwort "test123" und einen Timeout von 20 Sekunden
echo Sys_GetURLContentEx("http://www.google.de", Array("Timeout"=>
20000, "AuthUser"=> "test", "AuthPass"=> "test123"));
```

# Sys_Ping

```
boolean Sys_Ping (string $Host, int $Timeout)
```

## PARAMETERLISTE

| | |
|---|---|
| Host | Rechnername oder IP-Adresse |
| Timeout | Zeit in Millisekunden |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet einen Ping an **Host** und wartet **Timeout** Millisekunden.

## BEISPIEL

```
Sys_Ping("meinrechner", 1000); //Max. 1 Sekunde warten
```

# Tailscale VPN

Benötigt: IP-Symcon >= 7.1
Dieses Modul bietet die Möglichkeit, Tailscale VPN auf der SymBox zu verwenden.



Dieses Modul ist ausschließlich für die SymBox verfügbar!

## FUNKTIONSUMFANG

Herunterladen und Aktualisieren von Tailscale VPN
Authentifizieren über Auth Keys
Zugriff per VPN auf die SymBox (inkl. SSH)
Zugriff auf Geräte hinter der SymBox (Subnets)
VPN aktivieren/deaktivieren per Variable

## SOFTWARE-INSTALLATION

Über den Module Store das 'Tailscale VPN'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Tailscale VPN'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

## EINRICHTEN VON TAILSCALE

Auth Key erstellen: https://login.tailscale.com/admin/settings/keys
Tailscale Instanz öffnen
Buttons drücken, um Tailscale herunterzuladen, zu installieren und zu starten.
Danach kann die Authentifizierung durchgeführt werden mit Hilfe des am Anfang erstellen Auth Keys. Der Auth Key wird im Anschluss nicht mehr benötigt und muss nirgends gespeichert werden.

## NETZWERK FREIGEBEN

Das VPN in IP-Symcon einmal aus- und wieder einschalten.
Um Geräte hinter der SymBox freizugeben, muss die "Advertise Routes" Funktion eingerichtet werden
Dazu in die Liste das Subnet z.B. 192.168.178.0/24 in der CIDR Notation eintragen und die Konfiguration übernehmen.
Die Freigabe muss noch im Tailscale Dashboard aktiviert werden. Dazu folgende Seite

öffnen: https://login.tailscale.com/admin/machines
Dort die korrekte Maschine auswählen, Review klicken und die Freigabe des Subnets bestätigen.

## KONFIGURATIONSSEITE

| NAME | BESCHREIBUNG |
| --- | --- |
| Advertise Routes | Subnet, welches freigegeben werden soll in der CIDR Notation z.B. 192.168.178.0/24 |

# TextParser

Der TextParser ist ein Modul, welches Textabschnitte aus Webseiten oder Dateien zur Verfügung stellt.

## EINBINDUNG IN IP-SYMCON

Der Textparser kann als Instanz über das "+" im Objektbaum gesucht und hinzugefügt werden. Innerhalb der Konfiguration des Textparsers kann über "Gateway ändern" ein Gateway, welches den zu bearbeitenden Text zur Verfügung stellt, ausgewählt werden. Dies kann z.B. ein Http Client sein.

## BESCHREIBUNG

Der TextParser in IP-Symcon ermöglicht es bestimmte Teile aus z.B. einer Webseite oder eine Datei herauszuschneiden, indem mehrere Regeln hintereinander ausführt werden. Es stehen dabei folgende Regeln zur Verfügung:

| FUNKTION | BESCHREIBUNG |
|---|---|
| **Text abscheiden vor** | Schneidet allen Text vor Tag1 ab |
| **Text abschneiden hinter** | Schneidet allen Text hinter Tag1 ab |
| **Hole Text** | Holt den Text bis zum Tag1 in die Variable |
| **Text ausschneiden** | Schneidet den Text zwischen Tag1 und Tag2 aus und schreibt ihn in die Variable |

Es können mehrere Regeln hintereinander angewendet und sogar mehrere “Hole Text”- und “Text ausschneiden”-Operationen verwendet werden.

## BEISPIEL

Ein kleines Beispiel soll zeigen, wie man den Titel von heise.de auslesen kann.
Zuerst muss der Text Parser erstellt werden.
Danach muss eine übergeordnete Instanz erstellt werden. In diesem Beispiel ist der HTTP Client erforderlich.
Im HTTP Client muss die URL heise.de angeben werden und ggf. der Timer aktiviert werden, der die Seite zyklisch abfragt.
Zurück im Text Parser muss der HTTP Client als Gateway eingerichtet werden. Dieser kann über "Gateway ändern" ausgewählt werden.
Im Text Parser müssen folgende Regeln angegeben werden, um den Titel auslesen zu können.
Der Weg diese Regeln herauszufinden ist, sich den HTML Code, der durch den HTTP Client eingelesenen wurde, anzusehen. In diesem Code sucht man nach markanten Stellen, welche abgeschnitten werden können. Dies solange wiederholen bis am Ende nur noch der gewünschte Text übrig bleibt. Diese Regeln können dann in IP-Symcon eingetragen werden.
Mit "Hole Text" gibt man zusätzlich eine Variable an, in der der fertige ausgeschnitte Text gespeichert werden soll.

## REGELN

| BESCHREIBUNG | REGEL |
|---|---|
| Text abschneiden vor | <title> |
| Hole Text | </title> |

## SCREENSHOT



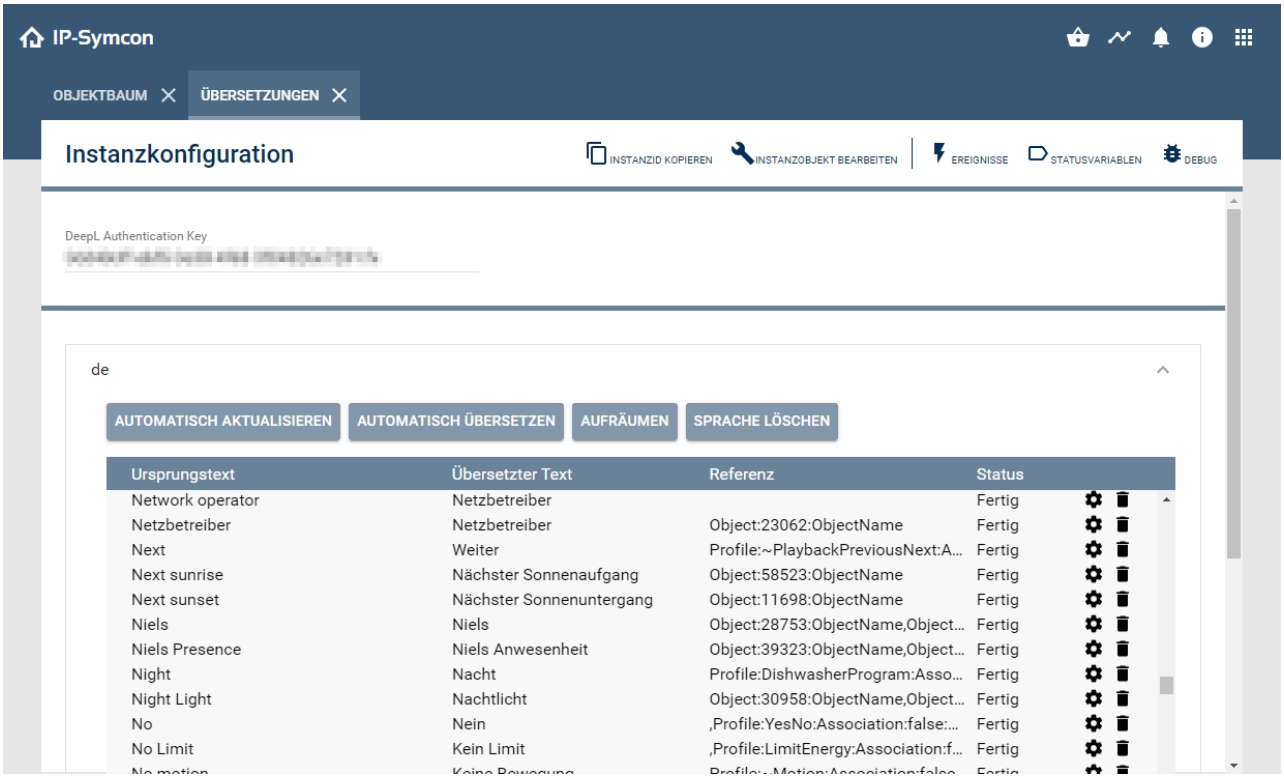
# Translation Control

Benötigt: IP-Symcon >= 7.1

Das Translation Modul ist eine kostenpflichtige Erweiterung, welche zu jeder bestehenden Symcon Lizenz erworben werden kann. Bei jeder Symcon Lizenz können bis zu 10 Strings je Sprache kostenlos übersetzt werden werden. Die Erweiterung kann direkt im Shop erworben werden.

Es wird empfohlen von Englisch in verschiedene Sprachen zu übersetzen. Daher sollte das Symcon System auf Englisch geführt werden. Hierfür kann die Sprache des Betriebssystems geändert werden. Wenn das Gerät auf dem Symcon installiert ist nicht beeinflusst werden soll, kann der Spezialschalter Locale auf en gesetzt werden

Das Translation Control ermöglicht es jegliche in Symcon vorhandene Text in verschiedene Sprachen zu übersetzen.
Die verschiedenen Übersetzungen können hier jederzeit angepasst werden. Mithilfe des Dienstes DeepL kann das gesamte System automatisch übersetzt werden. Hierfür wird ein API-Key von DeepL benötigt. Die Übersetzungen gelten sowohl für die Konsole als auch für die Visualisierung.




## SPRACHE HINZUFÜGEN

Eine neue Sprache kann mit dem Entsprechenden ISO-Sprachcode hinzugefügt werden.

## AKTIONEN

| AKTION | BESCHREIBUNG |
|---|---|
| Automatisch Aktualisieren | Fügt alle Texte, aus dem System in die Übersetzungstabelle hinzu. Diese können manuell oder automatisch übersetzt werden |
| Automatisch Übersetzen | Alle Werte der 'Ursprungstext' Spalter, welche nicht den Status 'Unbenutzt' haben werden mithilfe von DeepL übersetzt und in die Spalte 'Übersetzter Text' übertragen. |
| Aufräumen | Alle Einträge, die im System nicht mehr im System sind, werden entfernt |
| Sprache Löschen | Löscht die Übersetzungstabelle |

## ÜBERSETZUNGSTABELLE

| SPALTE | BESCHREIBUNG |
|---|---|
| Ursprungstext | Der Text, der im System gefunden wurde und für die Übersetzung mit DeepL genutzt wird |
| Übersetzter Text | Initial leer. Kann manuell oder durch die automatische Übersetzung gefüllt werden. Ist das Feld leer, wird der Ursprungstext angezeigt. |
| Referenz | Zeigt an ob der Text von einem Profil oder Objekt stammt und ob es sich beispielsweise um einen Objektnamen oder das Suffix eine Profils Handels. Dieser Kontext kann zum Verbessern der Übersetzung genutzt werden. |
| Status (Neu, Unbenutzt, Fertig) | Der aktuell Status der einzelnen Übersetzungen. Neu wird gesetzt, wenn die Übersetzung initial erkannt wird. Wurde die Übersetzung erfolgreich durchgeführt wird der Status auf Fertig gesetzt. Einträge mit dem Status Unbenutzt werden nicht für die Übersetzung übermittelt |

# Util Control

Dieses Modul ist eine Schnittstelle zwischen dem Dienst und der IP-Symcon Konsole und erlaubt erweiterte Funktionen, wie z.B. Suchen und Ersetzen in Dateien.
Auf der Modulkonfigurationsseite ist die Möglichkeit die Umlaute des Projekts zu überprüfen und zu korrigieren.



Dieses Modul sollte nicht gelöscht werden.
Falls es gelöscht wird, wird es beim nächsten Start wieder neu erstellt.

# WebHook Control

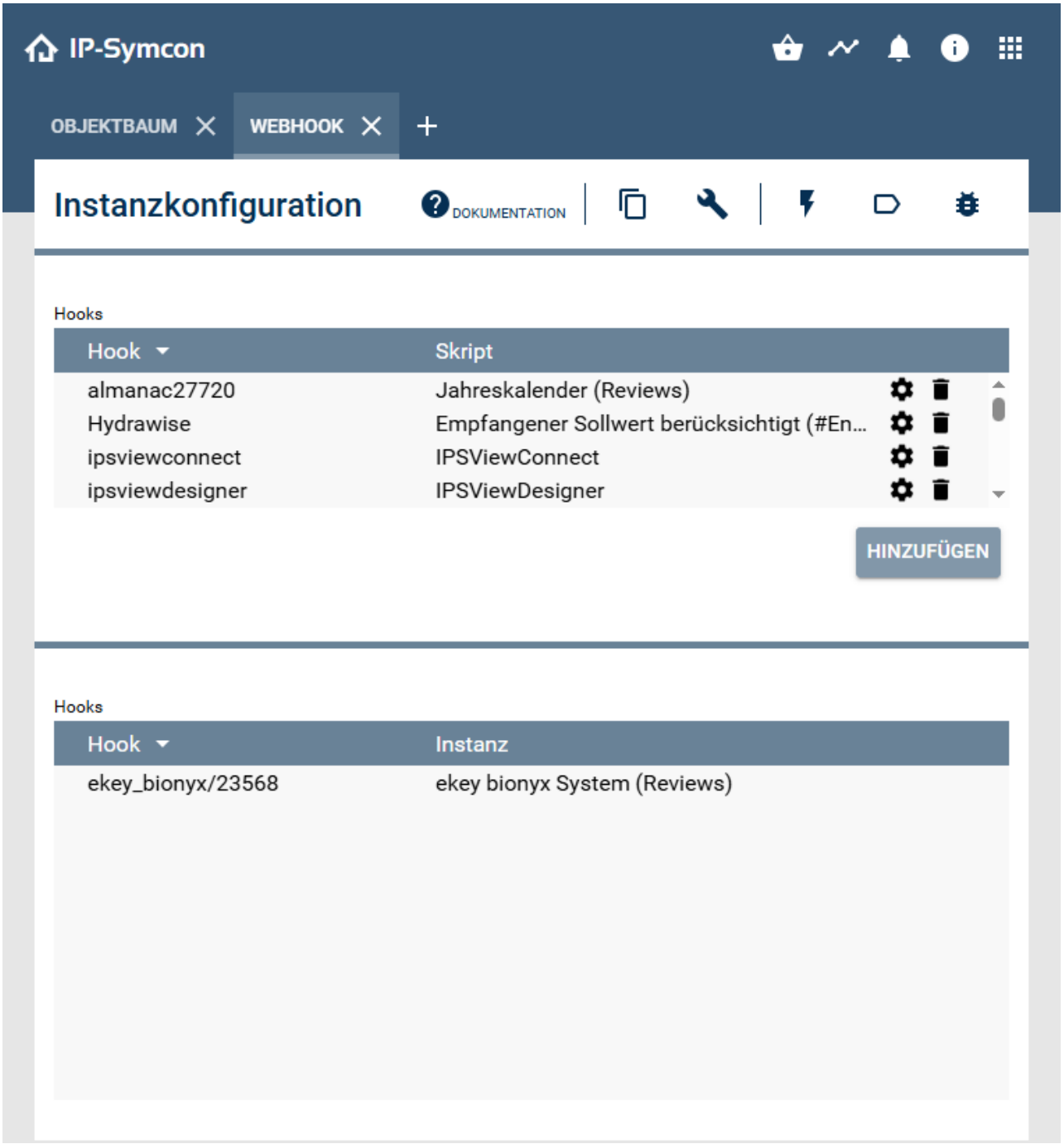
Benötigt: IP-Symcon >= 4.0

## BESCHREIBUNG

Das WebHook Control Modul bietet die Möglichkeit Skripte über einen Browser aufzurufen. Über den Symcon Connect Dienst sind auch Aufrufe außerhalb des lokalen Netzwerks möglich.
Es bietet weitergehend die Möglichkeit dem Skript Daten mitzuliefern.



Skripte, die durch den Webhook ausgeführt werden, beinhalten diese Systemvariablen




### WEBSOCKET SUPPORT

Ab der Symcon Version 5.2 nutzt das WebHook Control WebSockets und ermöglicht somit Echtzeitübermittlung.
Empfangene Daten können wie folgt erfasst und ausgegeben werden:

```
// Beim Empfangen landen die Daten im php://input Stream.
IPS_LogMessage("WebSocket", file_get_contents("php://input"));
```

## EINBINDUNG IN SYMCON

Das WebHook Control gehört zu den Kern Instanzen und ist somit sofort konfigurierbar.

Innerhalb eines Browsers kann das jeweilig verknüpfte Skript mit "IP:PORT/hook/HOOKNAME" aufgerufen werden.
Um einen Hook einzurichten muss in der "WebHook Control" Instanz bei der oberen Liste auf "Hinzufügen" geklickt werden. Folgend muss zuerst der Aufrufname des Hooks eingegeben werden und anschließend das Skript gewählt werden, welches durch den Hook ausgeführt werden soll.



Auch Aufrufe via Connect Control sind möglich.

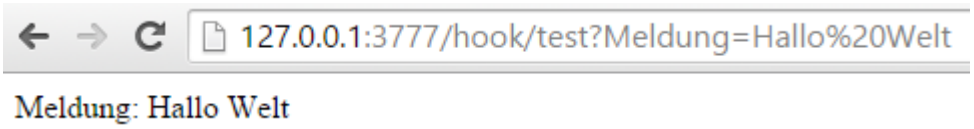
## BEISPIEL

### SKRIPT AUFRUF

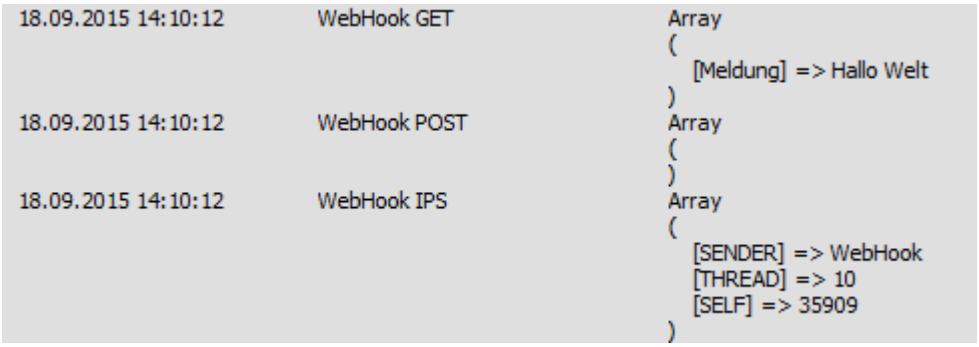
Das Beispielskript, welches aufgerufen werden soll, ist "WebHook TestSkript" mit der ObjektID "35909".

```
IPS_LogMessage("WebHook GET", print_r($_GET, true));
IPS_LogMessage("WebHook POST", print_r($_POST, true));
IPS_LogMessage("WebHook IPS", print_r($_IPS, true));
IPS_LogMessage("WebHook RAW", file_get_contents("php://input"));
echo "Meldung: " . $_GET['Meldung'];
```

Der Aufruf und das Ergebnis wären wiefolgt:




An Symcon werden die Daten in Arrays dem Skript übergeben.




### AUTHENTIFIZIERUNG

Ein Beispiel wie eine automatische Authentifizierung über das Modul Control verwirklicht wurde.

Als Nutzername wurde "Symcon" und als Passwort wurde "passwort" gewählt.



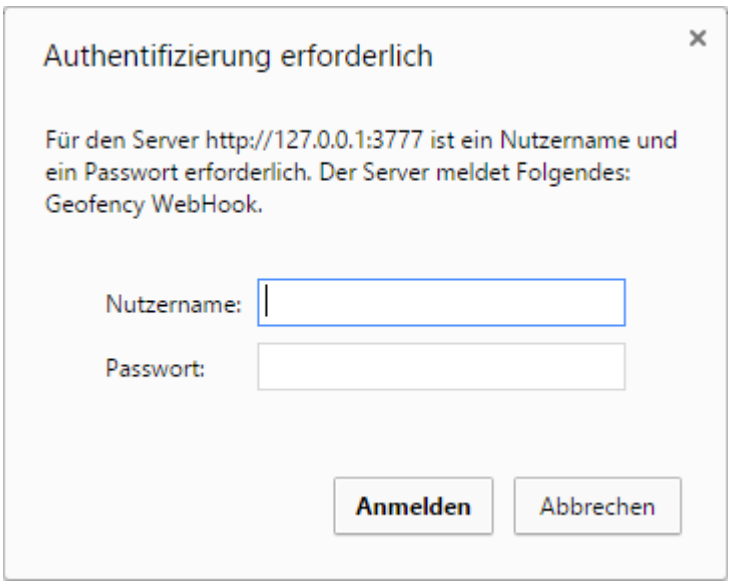
Eine Authentifizierung wird immer empfohlen, damit ein unerlaubter Zugriff von Außen auf Symcon verhindert wird.

*Quellcode*

```
if(!isset($_SERVER['PHP_AUTH_USER']))
$_SERVER['PHP_AUTH_USER'] = "";
if(!isset($_SERVER['PHP_AUTH_PW']))
$_SERVER['PHP_AUTH_PW'] = "";

if(($_SERVER['PHP_AUTH_USER'] != "Symcon") ||
($_SERVER['PHP_AUTH_PW'] != "passwort")) {
header('WWW-Authenticate: Basic Realm="Geofency WebHook"');
header('HTTP/1.0 401 Unauthorized');
echo "Authorization required";
return;
}
echo "Willkommen im geschützten Bereich";
```

Wird der Hook aufgerufen, erscheint die Abfrage nach Nutzername und Passwort. Und die Daten werden via $Post-Array an das Skript verschickt.




## EINBINDUNG VIA PHP-MODUL

Neben dem Aufruf via Skript, ist es auch möglich über die Funktion "ProcessHookData()" innerhalb eines PHP-Moduls auf Hookaufrufe zu reagieren.
Dazu wird ein Hook auf die eigene InstanceID registriert. Anschließend wird die Funktion "ProcessHookData()" automatisch durch das Hook Control aufgerufen.

## BEISPIEL

Dem Beispiel HookServe kann der Quellcode zur gesamten Handhabe eines WebHooks via PHP-Modul entnommen werden.

# WC_PushMessage

Benötigt: IP-Symcon >= 5.2

```
boolean WC_PushMessage (int $InstanzID, string $Hookpfad, string
$Nachricht)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des WebHook Control |
| Hookpfad | Pfad des Hooks |
| Nachricht | Zu sendene Nachricht |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet die Nachricht **Nachricht** an alle verbundene Clients auf den Hook **Hookpfad** des WebHook Controls mit der ID **InstanzID**.

## BEISPIEL

```
// Sendet "Hallo Welt" auf den hook "test" des WebHook Controls mit
der ID 12345
WC_PushMessage(12345, "/hook/test", "Hallo Welt");
```

# WC_PushMessageEx

Benötigt: IP-Symcon >= 5.3

```
boolean WC_PushMessageEx (int $InstanzID, string $Hookpfad, string
$Text, string $Zieladresse, int $Zielport)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des WebHook Control |
| Hookpfad | Pfad des Hooks |
| Text | Zu sendene Nachricht |
| Zieladresse | Adresse an die gesendet werden soll |
| Zielport | Port auf den gesendet werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet die Nachricht **Nachricht** an einen bestimmten Client auf den Hook **Hookpfad** des WebHook Controls mit der ID **InstanzID**. Der Client kann über die Paramter der IP-Adresse **Zieladresse** und des Zielports **Port** bestimmt werden.

## BEISPIEL

```
// Sendet "Hallo Welt zurück" auf den hook "test" des WebHook
Controls mit der ID 12345. Zusätzlich wird der Client mit der
Adresse $_SERVER["REMOTE_ADDR"] und Port $_SERVER["REMOTE_PORT"] als
Empfänger bestimmt.
WC_PushMessageEx(12345, "/hook/test", "Hallo Welt zurück",
$_SERVER["REMOTE_ADDR"], $_SERVER["REMOTE_PORT"]);
```

# WebServer

Das WebServer Modul in IP-Symcon wird verwendet, um die Visualisierung über einen weiteren Port verfügbar zu machen. Standardmäßig ist die Visualisierung über Port 3777 verfügbar. Insbesondere ist ein weiterer WebServer von Vorteil, wenn eine SSL-Verschlüsselung genutzt werden soll.
Optional ist es möglich die Basis-Authentifizierung zu aktiveren, welche einen zusätzlichen Schutz für den "user"-Ordner gewährt.



Skripte, die durch den WebServer ausgeführt werden, beinhalten diese Systemvariablen .



Die JSON-RPC API (/api/) ist auf jedem WebServer verfügbar und wird durch den Fernzugriff abgesichert. Ab IP-Symcon 4.0 sind zusätzlich die WebHooks (/hook/) verfügbar, welche jeweils durch die Authentifizierung des jeweiligen WebHooks abgesichert werden.



Ab Version 4.0 wird der WebServer nicht mehr automatisch hinzugefügt. Um diesen nutzen zu können, muss er via "Objekt hinzufügen -> Instanz hinzufügen -> WebServer" hinzugefügt werden.

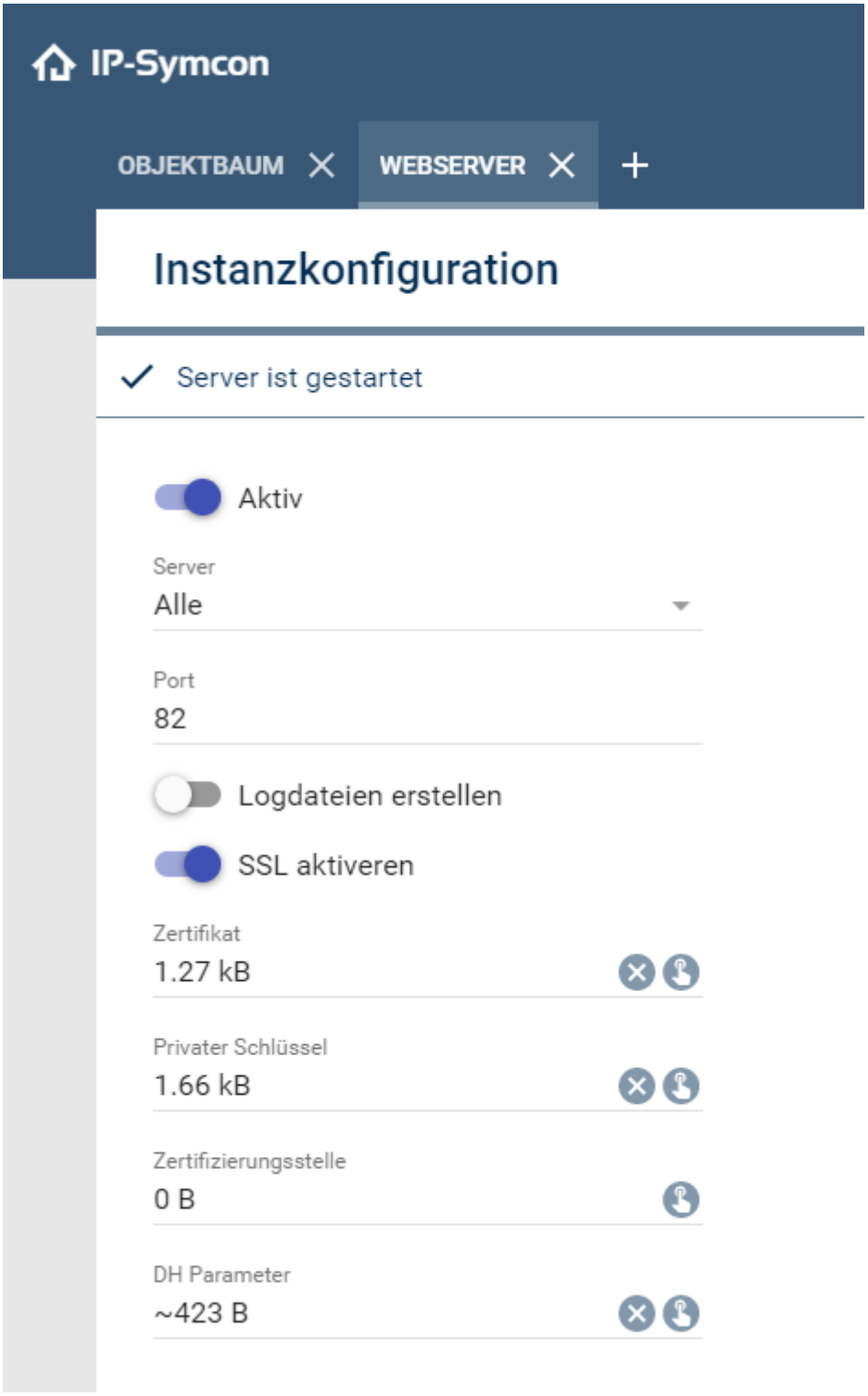


Die Basis-Authentifizierung gilt ausschließlich für den "user"-Ordner. Die Startseite mit der Auflistung der verfügbaren Konfiguratoren ist immer ohne Kennwort erreichbar. Falls je Visualisierungskonfigurator ein Visualsierungskennwort vergeben werden soll, kann dies in der Konfiguration vom jeweiligen Visualisierungskonfigurator eingerichtet werden.

## SSL-VERSCHLÜSSELUNG

Um SSL nutzen zu können, werden mind. das Zertifikat und der private Schlüssel benötigt.

| OPTION | BESCHREIBUNG |
|---|---|
| Zertifikat | Auswahl der *.pem Datei, welche das Zertifikat beinhaltet |
| Privater Schlüssel | Auswahl der *.pem Datei, welche den privaten Schlüssel beinhaltet |
| Zertifizierungsstelle (optional) | Auswahl der *.pem Datei, welche die Zertifizierungsstelle für einen vertrauenswürdigen Zugriff beinhaltet |
| DH Parameter (optional) | Auswahl der dhparam.pem Datei für den Diffie-Hellman-Parameter, welcher einen Logjam verhindert. Dies bringt eine erhöhte Sicherheit |




## ZERTIFIKAT ERSTELLEN

Mithilfe von OpenSSL können Zertifikat und Private Key erstellt werden.

```
// Generate a private key
openssl genrsa -out pk.pem 1024

// Create a certificate signing request
openssl req -new -key pk.pem -out req.csr

// Self-sign the csr
```

```
openssl x509 -req -days 3650 -in req.csr -signkey pk.pem -out
cert.pem
```

Damit Browser ein Zertifikat akzeptiert muss IP-Symcon über eine öffentliche Domain zugreifbar und ein von einer anerkannten Zertifizierungsstelle ausgestelltes Zertifikat vorhanden sein.

## "USER" ORDNER

Der "user"-Ordner beinhaltet benutzerdefinierte Inhalte von IP-Symcon (z.B. weitere Darstellungskripte).
Der Pfad zu diesem Ordner ist abhängig vom genutzten Betriebssystem.

SymBox: /var/lib/symcon/user/
Windows: C:\ProgramData\Symcon\user\
MacOS: /Library/Application Support/Symcon/user/
Linux: /var/lib/symcon/user/
Raspberry Pi: /var/lib/symcon/user/

Werden Dateien unterhalb eines anderen Pfads gespeichert, so werden diese beim nächsten Update von IP-Symcon automatisch gelöscht. Nur so ist sicherzustellen, dass die Installation von IP-Symcon stets frei von Dateiversionskonflikten bleibt.

Bis inklusive IP-Symcon 6.4 waren die Pfade wie folgt:
SymBox: /var/lib/symcon/user/
Windows: C:\ProgramData\Symcon\user\
MacOS: /Library/Application Support/Symcon/user/
Linux: /var/lib/symcon/user/
Raspberry Pi: /var/lib/symcon/user/

## "USER"-ORDNER ABSICHERN

Sofern Inhalte im speziellen "user"-Ordner abgelegt wurden, ist es ratsam die Basis-Authentifizierung zu aktivieren. Dadurch werden diese Inhalte zusätzlich geschützt und bei Abruf innerhalb der Visualisierung oder der mobilen Apps nach der jeweiligen Benutzername/Passwort-Kombination gefragt.
Die Einrichtung der Authentifizierung befindet sich bei den Spezialschalter.

Sollte in einer älteren IP-Symcon Version (bis inkl. 5.0) in einem WebServer die Basis-Authentifizierung aktiviert worden sein, so gilt diese bis zu dessen Deaktivierung.

## LOGDATEIEN

Die Option "Logdateien erstellen" erstellt im "logs"-Verzeichnis von IP-Symcon je WebServer eine Datei namens access_12345.log, wobei die Zahl 12345 für die InstanzID des WebServers steht.
Die erstellten Logdateien sind Webalizer-Kompatibel und können somit grafisch ausgewertet werden.
Der Pfad zu diesem Ordner ist abhängig vom genutzten Betriebssystem.

SymBox: /var/log/symcon/
Windows: C:\ProgramData\Symcon\logs\
MacOS: /Library/Logs/Symcon/
Linux: /var/log/symcon/
Raspberry Pi: /var/log/symcon/

## TIPPS UND TRICKS

Forum: "SSL-Zertifikat bestellen und in IP-Symcon einrichten"

# I/O Instanzen

I/O Instanzen liefern Daten von Geräten oder Quellen außerhalb von IP-Symcon und stellen diese zur Verfügung. Oftmals bilden diese die erste Schnittstelle zum physikalischen Gerät.

## EINBINDUNG IN IP-SYMCON

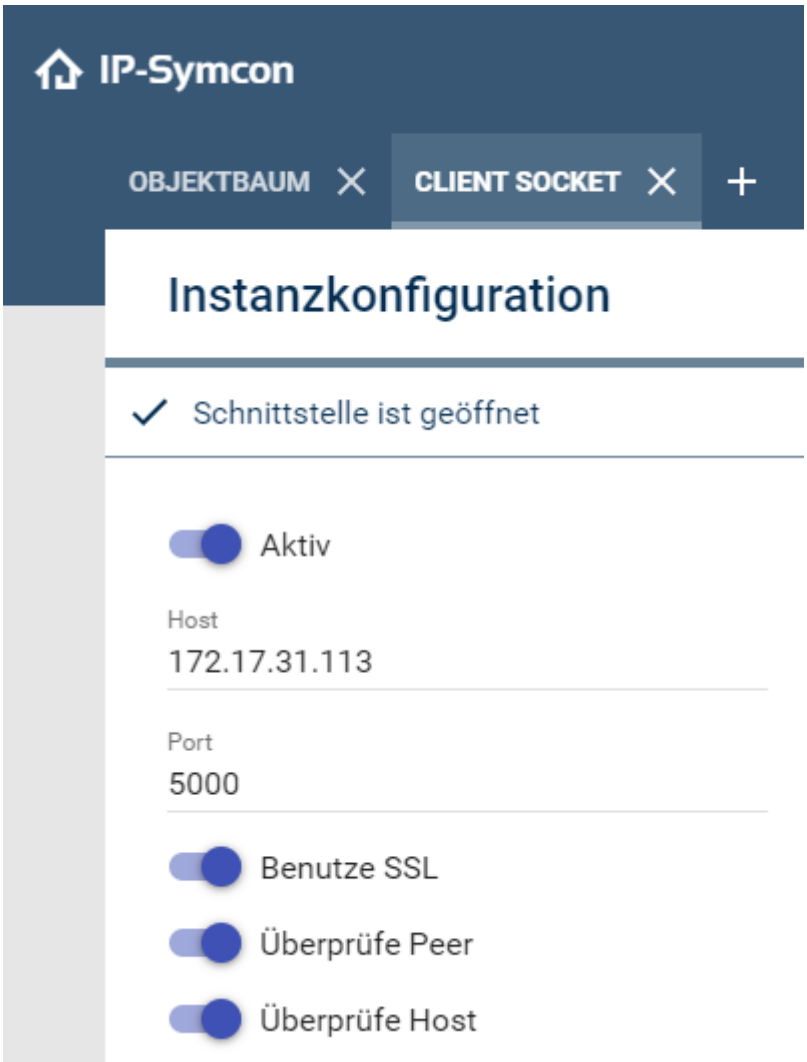
Beim Erstellen einer Geräte-Instanz, welche als übergordnete Instanz eine I/O Instanz benötigt, wird oftmal die passende Instanz automatisch erstellt und eingerichtet.
Es müssen nur die Einstellungen kontrolliert werden.
Eine I/O Instanz kann bei Bedarf auch eingefügt werden. Dies wird oftmal in Kombination mit einer RegisterVariable genutzt, welche via Systemvariablen weitere Werte zur Verfügung hat.
Auch werden I/O Instanzen in der PHP Modul Entwicklung genutzt. Dort werden, ähnlich zu nativen Modulen, I/O Instanzen als Schnittstelle eingefügt.

# Client Socket

Benötigt: IP-Symcon >= 2.0
Der Client Socket öffnet eine Schnittstelle, welche oftmals für die Kommunikation zwischen Gateway/Splitter Instanz und dem angeschlossenen Gerät genutzt wird. Es verwendet das TCP Protokoll.

## EINBINDUNG IN IP-SYMCON

Auf der Konfigurationsseite muss unter "Host" die IP/Hostname und unter "Port" der Port des anzusteuernden Geräts angegeben werden.
Bei aktiviertem SSL kann noch ausgewählt werden ob Host oder Peer überprüft werden sollen.
Erst ein Klick auf "Übernehmen" speichert die Einstellungen und anschließend wird versucht die Verbindung aufzubauen.



# CSCK_SendText

Benötigt: IP-Symcon >= 2.0

```
boolean CSCK_SendText (int $InstanzID, string $Text)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu aktualisierenden Client Sockets |
| Text | der zusendene String |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet den String **Text** auf dem Client Socket mit der ID **InstanzID**.

## BEISPIEL

```
CSCK_SendText(12345, "Beliebiger Datensatz"); //Sendet den Text
"Beliebiger Datensatz" auf dem Client Socket mit der ID 12345
```

# HID

Benötigt: IP-Symcon >= 2.0
Das HID (Human Interface Device) öffnet eine USB-Schnittstelle, welche oftmals für die Kommunikation zwischen Gateway/Splitter Instanz und dem angeschlossenen Gerät genutzt wird.

## EINBINDUNG IN IP-SYMCON

Auf der Konfigurationsseite muss unter "Gerät" das anzusteuernden Gerät angegeben werden.
Das Anhaken von "HID Gerät öffnen" setzt die Verbindung nach "Übernehmen" auf aktiv.
Alle Einstellungen/Änderungen werden erst nach "Übernehmen" gespeichert



# HID_SendEvent

Benötigt: IP-Symcon >= 2.0

```
boolean HID_SendEvent (int $InstanzID, int $ReportID, string $Text)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu aktualisierenden HIDs |
| ReportID | ID der zu sendenen ReportID |
| Text | der zusendene String |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet den String **Text** und dem Datensatztyp **ReportID** auf dem HID mit der ID **InstanzID**.

## BEISPIEL

```
HID_SendEvent(12345, 0, "Beliebiger Datensatz"); //Sendet den Text
"Beliebiger Datensatz" auf dem HID mit der ID 12345
```

# HTTP Client

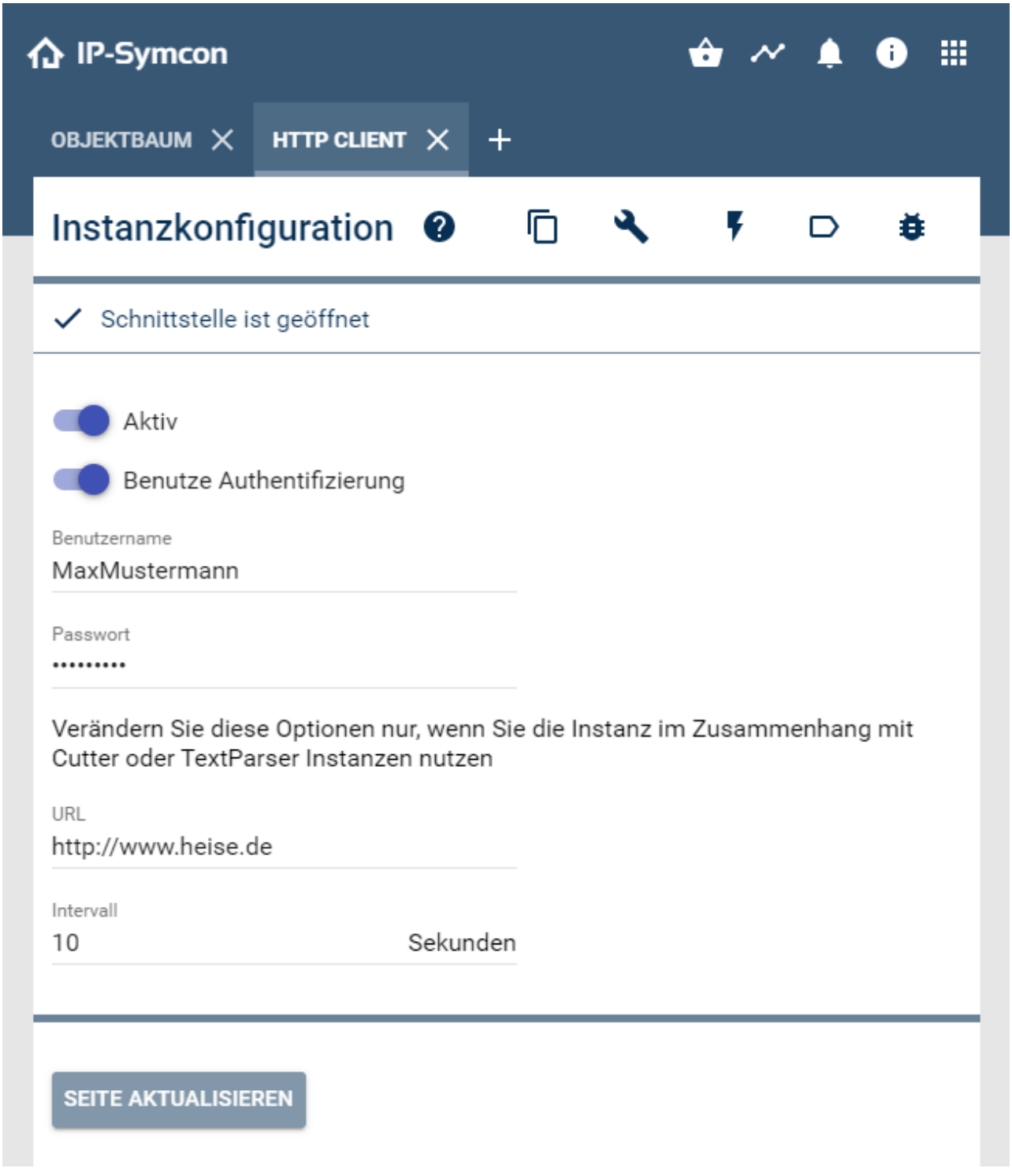
Benötigt: IP-Symcon >= 2.3
Der HTTP Client liest Daten aus einer URL ein und stellt diese zur Verfügung.

## EINBINDUNG IN IP-SYMCON

Auf der Konfigurationsseite kann optional eine Authentifizierung über Benutzername und Passwort eingerichtet werden.

### WEBSEITE EINLESEN

Wenn eine Webseite eingelesen werden soll, muss die URL eingetragen werden.
Das Intervall bestimmt, wie oft diese Abfrage durchgeführt wird.


IP-Symcon
OBJEKTBAUM
HTTP CLIENT
Instanzkonfiguration
Schnittstelle ist geöffnet
Aktiv
Benutze Authentifizierung
Benutzername
MaxMustermann
Passwort
Verändern Sie diese Optionen nur, wenn Sie die Instanz im Zusammenhang mit Cutter oder TextParser Instanzen nutzen
URL
http://www.heise.de
Intervall
10
Sekunden
SEITE AKTUALISIEREN

# WWW_UpdatePage

Benötigt: IP-Symcon >= 2.3

```
boolean WWW_UpdatePage (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu aktualisierenden HTTP Clients |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Lässt die URL des HTTP Clients mit der ID **InstanzID** abfragen.



Dieser Befehl hat nur eine Funktion sofern die URL-Maske auf der Konfigurationsseite genutzt wird.

## BEISPIEL

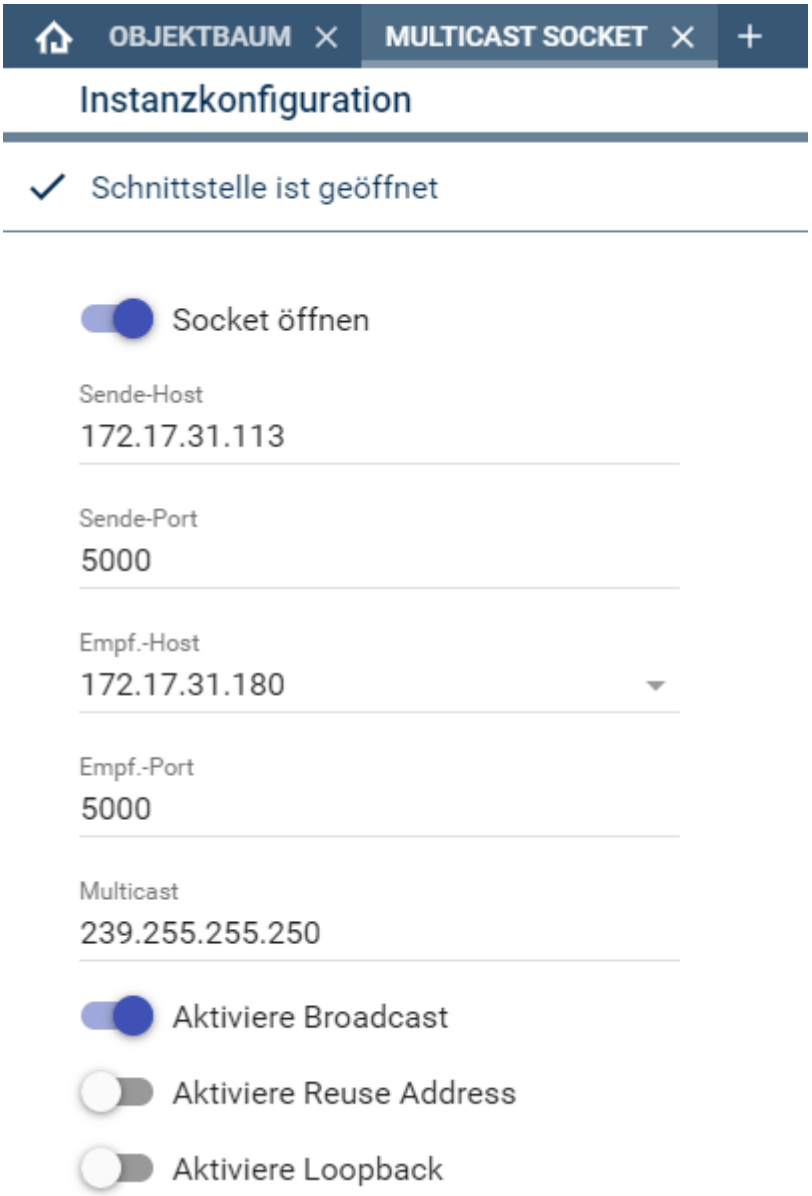
```
//Manuelle Abfrage der konfigurierten URL des HTTP Clients mit der
InstanzID 12345
WWW_UpdatePage(12345);
```

# Multicast Socket

Benötigt: IP-Symcon >= 4.1
Der Multicast Socket öffnet eine Schnittstelle, welche oftmals für die Kommunikation zwischen Gateway/Splitter Instanz und dem angeschlossenen Gerät genutzt wird. Es verwendet das UDP Protokoll.

## EINBINDUNG IN IP-SYMCON

Auf der Konfigurationsseite kann der Multicast-Socket eingerichtet und aktiviert werden. Erst ein Klick auf "Übernehmen" speichert die Einstellungen und anschließend wird versucht die Verbindung, bei aktivierten "Socket öffnen", aufzubauen.




### SENDE-HOST/PORT

IP und Port des anzusteuernden Geräts.

### EMPFANGS HOST/PORT

Bestimmt auf welchem Netzwerk und über welche IP und Port die Daten entgegengenommen werden.

### MULTICAST

Multicast Gruppen Adresse bei der sich angemeldet und auf die gehorcht wird.

### AKTIVIERBARE OPTIONEN

| OPTION | BESCHREIBUNG |
|---|---|
| Broadcast | Erlaubt das Senden von Broadcast Nachrichten |
| Reuse Address | Erlaubt das Anbinden des Sockets an eine Adresse, welche bereits in Benutzung ist. |
| Loopback | Aktiviert das Empfangen von Multicast-Paketen auf dem Sendesocket, sofern dieser ein Mitglied der Multicast Gruppe ist. |

# MSCK_SendText

Benötigt: IP-Symcon >= 4.1

```
boolean MSCK_SendText (int $InstanzID, string $Text)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu aktualisierenden Multicast Sockets |
| Text | der zusendene String |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Sendet den String Text auf dem Multicast Socket mit der ID InstanzID.

## BEISPIEL

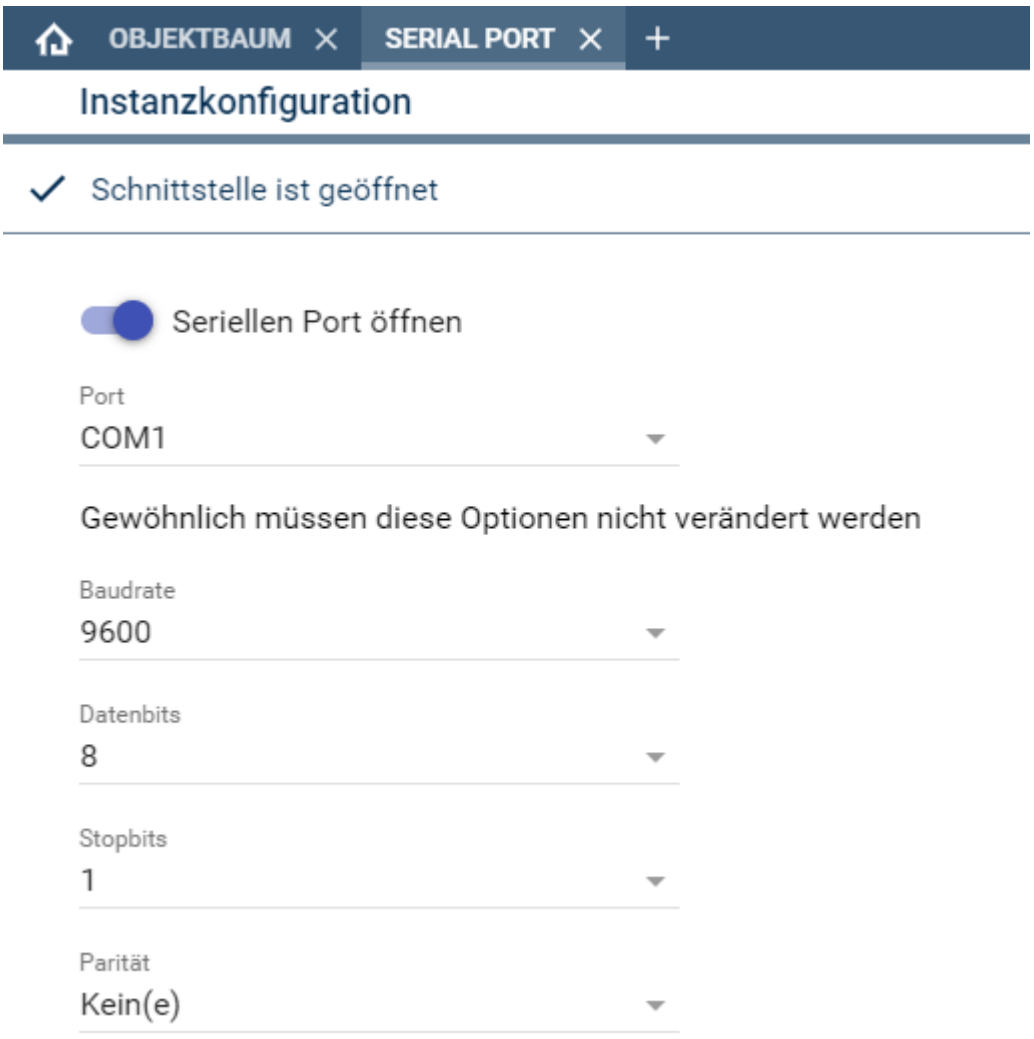
```
//Sendet den String "Beliebiger Datensatz" auf dem Multicast Socket
mit der ID 12345
MSCK_SendText(12345, "Beliebiger Datensatz");
```

# Serial Port

Benötigt: IP-Symcon >= 2.0
Der Serial Port öffnet eine Schnittstelle, welche oftmals für die Kommunikation zwischen Gateway/Splitter Instanz und dem angeschlossenen Gerät genutzt wird.

## EINBINDUNG IN IP-SYMCON

Auf der Konfigurationsseite muss der serielle Anschluss, an dem das anzusteuernde Gerät angeschlossen ist, angegeben werden.
Gegebenfalls können Baudrate, Datenbits, Stopbits und Parität eignestellt werden.
Erst ein Klick auf "Übernehmen" speichert die Einstellungen und anschließend wird versucht die Verbindung aufzubauen.



# SPRT_SendText

Benötigt: IP-Symcon >= 2.0

```
boolean SPRT_SendText (int $InstanzID, string $Text)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu aktualisierenden Serial Ports |
| Text | der zusendene String |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Sendet den String Text auf dem Serial Port mit der ID InstanzID.

## BEISPIEL

```
//Sendet einen den Text "Beliebiger Datensatz" auf dem Serial Port
mit der ID 12345
SPRT_SendText(12345, "Beliebiger Datensatz");
```

# SPRT_SetBreak

Benötigt: IP-Symcon >= 2.6

```
boolean SPRT_SetBreak (int $InstanzID, boolean $AnAus)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu aktualisierenden Serial Ports |
| AnAus | schaltet den Break An **True** oder Aus **False** |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schaltet den Break auf **AnAus** (True = An; False = Aus) auf dem Serial Port mit der ID **InstanzID**.

## BEISPIEL

```
// schaltet den Break der Serial Port Instanz mit der ID 12345 auf
"An"
SPRT_SetBreak(12345, true);
```

# SPRT_SetDTR

Benötigt: IP-Symcon >= 2.6

```
boolean SPRT_SetDTR (int $InstanzID, boolean $AnAus)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu aktualisierenden Serial Ports |
| AnAus | schaltet den DTR An **True** oder Aus **False** |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schaltet den DTR auf **AnAus** (True = An; False = Aus) auf dem Serial Port mit der ID **InstanzID**.

## BEISPIEL

```
// schaltet den DTR der Serial Port Instanz mit der ID 12345 auf
"An"
SPRT_SetDTR(12345, true);
```

# SPRT_SetRTS

Benötigt: IP-Symcon >= 2.6

```
boolean SPRT_SetRTS (int $InstanzID, boolean $AnAus)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu aktualisierenden Serial Ports |
| AnAus | schaltet den RTS An **True** oder Aus **False** |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Schaltet den RTS auf **AnAus** (True = An; False = Aus) auf dem Serial Port mit der ID **InstanzID**.

## BEISPIEL

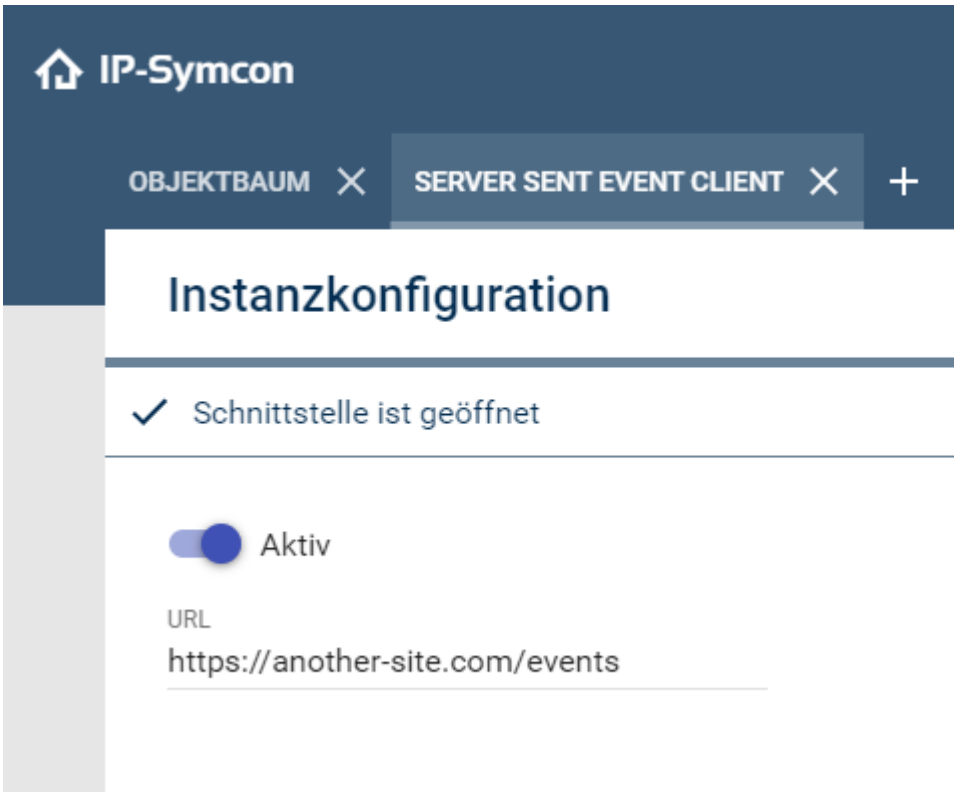
```
// schaltet den RTS der Serial Port Instanz mit der ID 12345 auf
"An"
SPRT_SetRTS(12345, true);
```

# Server Sent Event Client

Der Server Sent Event Client öffnet eine Schnittstelle, welche automatisch auf Server Event reagiert. Es verwendet eine HTTP Verbindung.

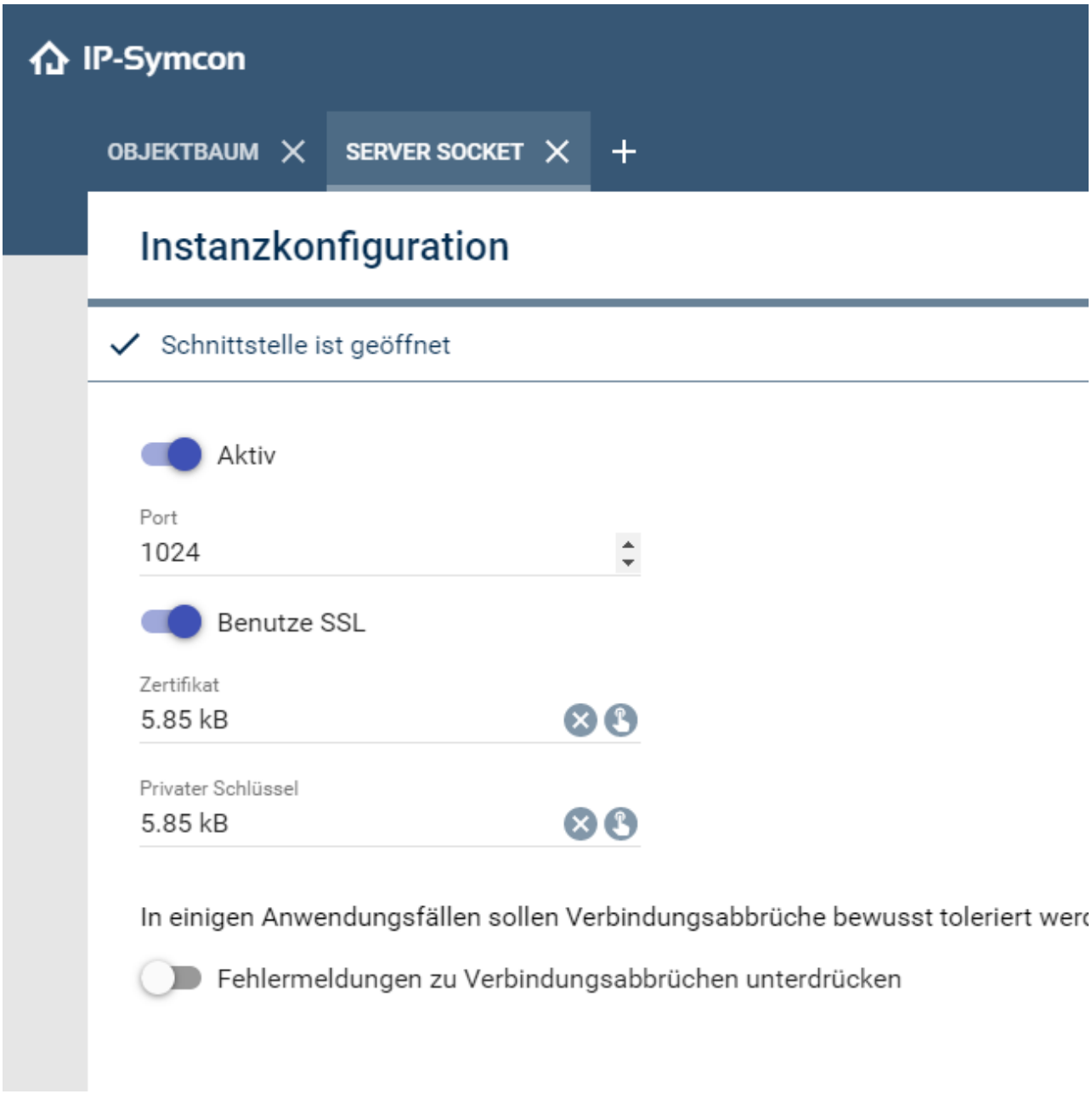
## EINBINDUNG IN IP-SYMCON

Auf der Konfigurationsseite muss unter "URL" die Adresse des Servers angegeben werden auf den man reagiert.
Erst ein Klick auf "Übernehmen" speichert die Einstellungen und anschließend wird versucht die Verbindung aufzubauen.



# Server Socket

Benötigt: IP-Symcon >= 4.1
Der Server Socket öffnet auf dem Server einen Port für alle Netzwerke. Es kann dann auf diesem Port gehorcht und Daten empfangen werden.
Erst ein Klick auf "Übernehmen" speichert die Einstellungen und anschließend ist der Port geöffnet.
Bei aktiviertem SSL müssen Zertifikat und der private Schlüssel angegegeben werden.




## ZERTIFIKAT ERSTELLEN

Mithilfe von OpenSSL können Zertifikat und Private Key erstellt werden.

```
// Generate a private key
openssl genrsa -out pk.pem 1024
```

```
// Create a certificate signing request
openssl req -new -key pk.pem -out req.csr

// Self-sign the csr
openssl x509 -req -days 3650 -in req.csr -signkey pk.pem -out
cert.pem
```



Das Zertifikat für die Zertifizierungsstelle ist optional.

# SSCK_SendPacket

Benötigt: IP-Symcon >= 4.1

```
boolean SSCK_SendPacket (int $InstanzID, string $Text, string
$ClientIP, int $ClientPort)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu aktualisierenden Server Sockets |
| Text | der zusendene String |
| ClientIP | IP des Empfängers |
| ClientPort | Port des Empfängers |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet den String **Text** auf dem Server Socket mit der ID **InstanzID**. Als Empfängeradressierung wird die IP **ClientIP** und Port **ClientPort** genutzt.



Sollte keine Verbindung von der IP-Adresse auf dem entsprechenden Port vorhanden sein, so wird der zu sendende String verworfen

## BEISPIEL

```
//Sendet das Packet mit dem Datensatz "Beliebiger Datensatz" auf dem
Server Socket mit der ID 12345 an die IP:Port 192.168.1.123:1234
SSCK_SendPacket(12345, "Beliebiger Datensatz", "192.168.1.123",
1234);
```

# SSCK_SendText

Benötigt: IP-Symcon >= 4.1

```
boolean SSCK_SendText (int $InstanzID, string $Text)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu aktualisierenden Server Sockets |
| Text | der zusendene String |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Sendet den String Text auf dem Server Socket mit der ID InstanzID.

## BEISPIEL

```
SSCK_SendText(12345, "Beliebiger Datensatz"); //Sendet einen den
Text "Beliebiger Datensatz" auf dem Server Socket mit der ID 12345
```

# UDP Socket

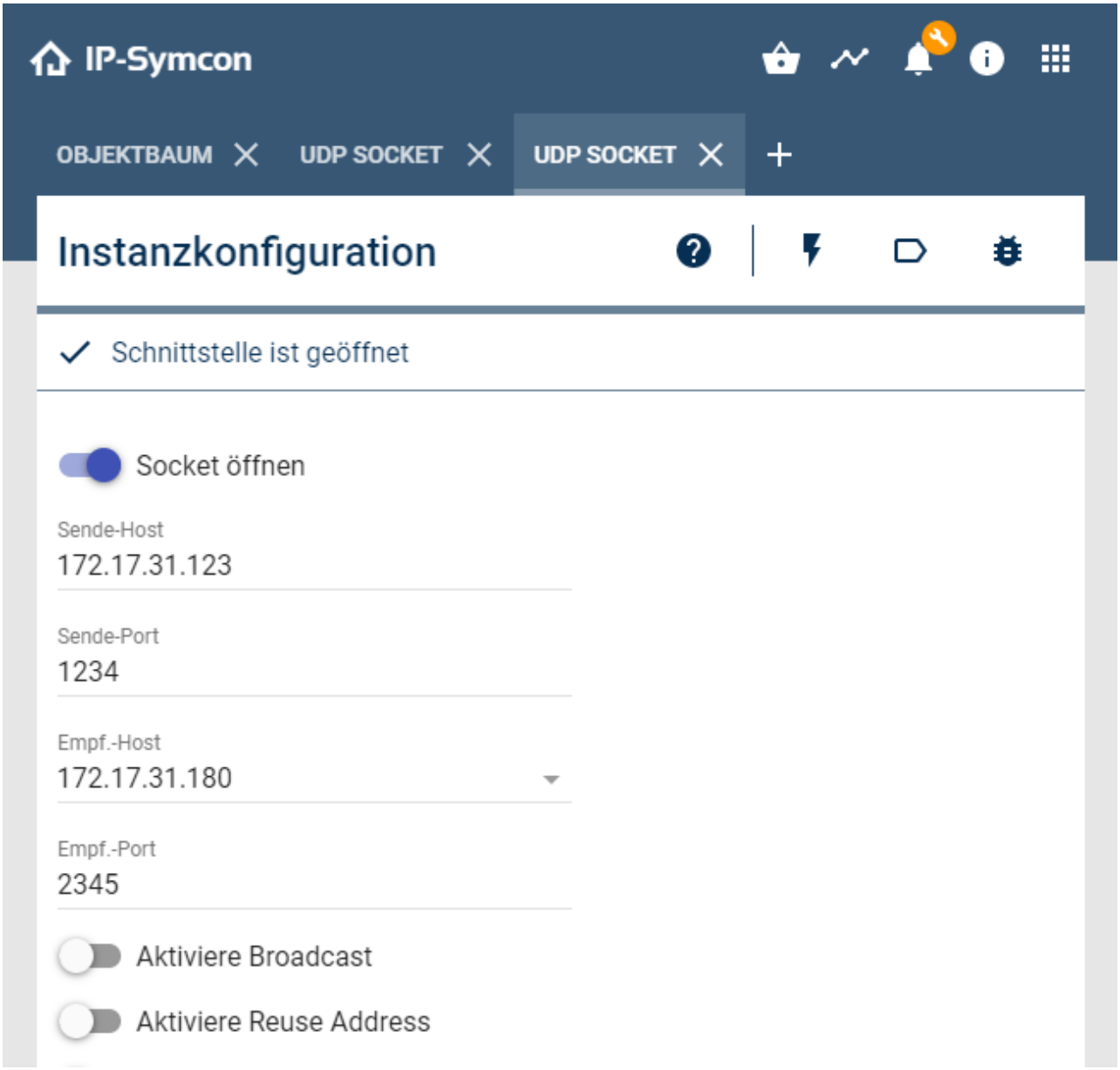
Benötigt: IP-Symcon >= 3.3
Der UDP Socket öffnet eine Schnittstelle, welche oftmals für die Kommunikation zwischen Gateway/Splitter Instanz und dem angeschlossenen Gerät genutzt wird. Es verwendet das UDP Protokoll.

## EINBINDUNG IN IP-SYMCON

Über "+" im Objektbaum kann eine neue UDP Socket Instanz erstellt werden.
Auf der Konfigurationsseite können folgende Einstellungen vorgenommen werden.

| OPTION | BESCHREIBUNG |
|---|---|
| Socket öffnen | Ob der Socket geöffnet oder geschlossen sein soll. |
| Sende-Host | Auf welche IP-Adresse die Daten gesendet werden sollen. |
| Sende-Port | Auf welchen Port die Daten gesendet werden sollen. |
| Empf.-Host | Auf welchem TCP-Interface auf die Daten gelauscht werden soll. |
| Empf.-Port | Auf welchem Port auf die Daten gelauscht werden soll. |
| Aktiviere Broadcast | Bei Aktivierung schickt der UDP Socket an die Broadcast-Adresse des ausgewählten Empf.-Host Interfaces. Die eingetragene Sende-Host Adresse ist dabei unerheblich. |
| Aktiviere Reuse Address | Erlaubt es anderen Programmen bei ebenfalls aktivierten "Reuse address" auf dem Empf.-Port zu lauschen. |

Erst ein Klick auf "Übernehmen" speichert die Einstellungen und anschließend wird versucht die Verbindung aufzubauen.


IP-Symcon
OBJEKTBAUM
UDP SOCKET
UDP SOCKET
Instanzkonfiguration
Schnittstelle ist geöffnet
Socket öffnen
Sende-Host
172.17.31.123
Sende-Port
1234
Empf.-Host
172.17.31.180
Empf.-Port
2345
Aktiviere Broadcast
Aktiviere Reuse Address

# USCK_SendPacket

Benötigt: IP-Symcon >= 5.0

```
boolean USCK_SendPacket (int $InstanzID, string $Text, string
$ClientIP, int $ClientPort)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu aktualisierenden UDP Sockets |
| Text | der zusendene String |
| ClientIP | IP des Empfängers |
| ClientPort | Port des Empfängers |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet den String **Text** auf dem UDP Socket mit der ID **InstanzID**. Als Empfängeradressierung wird die IP **ClientIP** und Port **ClientPort** genutzt.



Wenn ClientIP leer gelassen wird, wird an den im UDP Socket eingestellten Host gesendet. Wenn Broadcast aktiv ist, wird der ClientIP Parameter ignoriert und der Inhalt von $Text an die Broadcast Adresse gesendet. Wenn ClientPort = 0 ist, dann wird der im UDP Socket eingestellte Port verwendet.

## BEISPIEL

```
//Sendet das Packet mit dem Datensatz "Beliebiger Datensatz" auf dem
Server Socket mit der ID 12345 an die IP:Port 192.168.1.123:1234
USCK_SendPacket(12345, "Beliebiger Datensatz", "192.168.1.123",
1234);
```

# USCK_SendText

Benötigt: IP-Symcon >= 3.3

```
boolean USCK_SendText (int $InstanzID, string $Text)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu aktualisierenden UDP Sockets |
| Text | der zusendene String |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet den String **Text** auf dem UDP Socket mit der ID **InstanzID**.

## BEISPIEL

```
USCK_SendText(12345, "Beliebiger Datensatz"); //Sendet einen den
Text "Beliebiger Datensatz" auf dem UDP Socket mit der ID 12345
```

# Virtual I/O

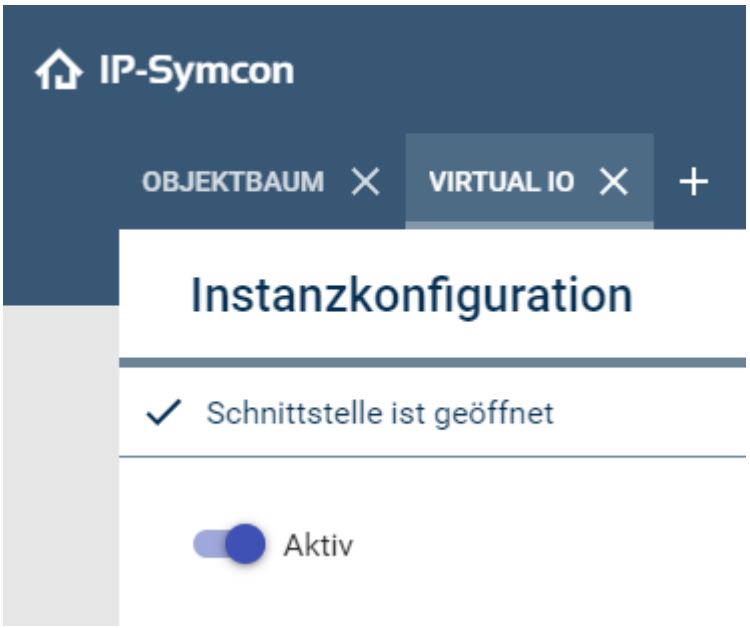
Benötigt: IP-Symcon >= 3.3
Der Virtual I/O öffnet eine Schnittstelle, welche für die Kommunikation zwischen Gateway/Splitter Instanz und dem angeschlossenen Gerät genutzt wird. Es verwendet kein bestimmtes Protokoll, sondern dient für Testzwecke um ein korrektes Verhalten der verbundenen Instanzen sicherzustellen.
Das Virtual I/O ist in der Lage Datenpakete vom Typ Simpel und Erweitert (Socket) zu versenden. Es gilt, dass Funktionen mit "Text" solche simplen Datenpakete nutzen. Connect, Disconnect und Funktionen mit "Packet" nutzen das Datenpaket Erweitert (Socket).

## EINBINDUNG IN IP-SYMCON

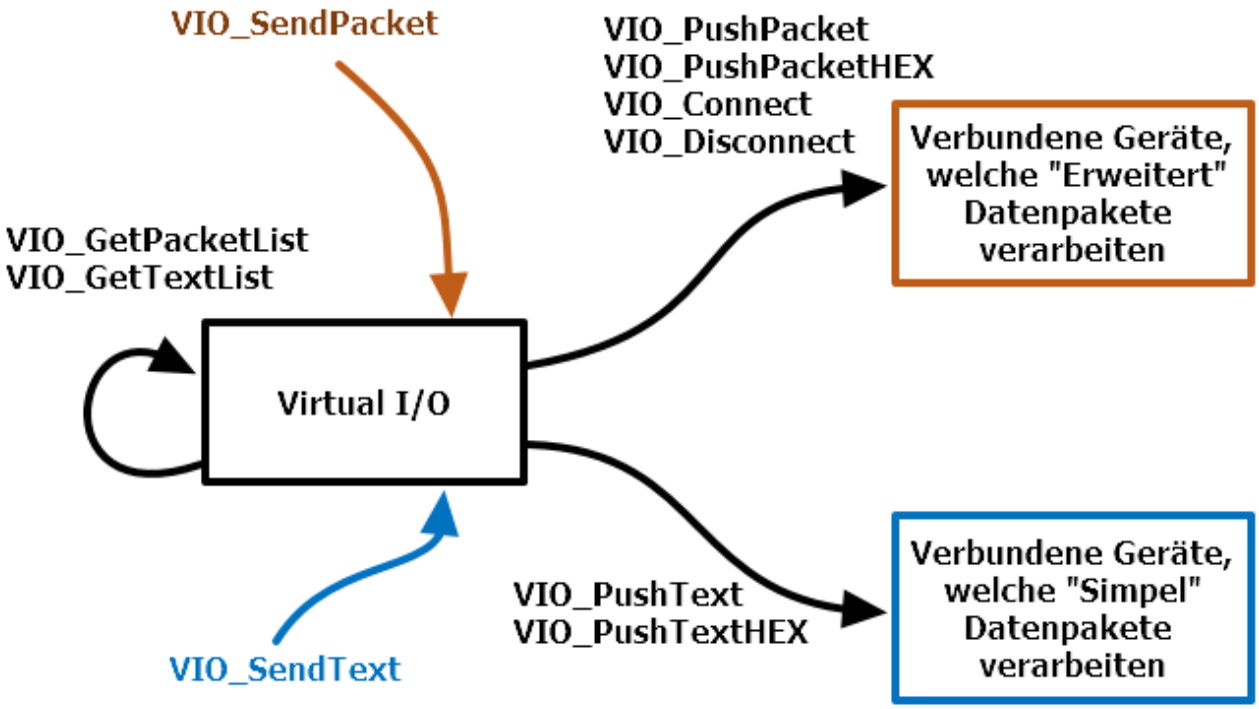
Über "+" im Objektbaum kann eine neue Virtual I/O Instanz erstellt werden.
Auf der Konfigurationsseite können folgende Einstellungen vorgenommen werden.

| OPTION | BESCHREIBUNG |
|---|---|
| Aktiv | Ob die Schnittstelle geöffnet oder geschlossen sein soll. |

Erst ein Klick auf "Übernehmen" speichert die Einstellungen und anschließend wird versucht die Verbindung aufzubauen.




Zum weiteren Verständnis zeigt die Grafik auf, welche Richtung und Datenpakettypen die einzelnen Befehle nutzen.


VIO_SendPacket
VIO_PushPacket
VIO_PushPacketHEX
VIO_Connect
VIO_Disconnect
Verbundene Geräte, welche "Erweitert" Datenpakete verarbeiten
VIO_GetPacketList
VIO_GetTextList
Virtual I/O
VIO_PushText
VIO_PushTextHEX
Verbundene Geräte, welche "Simpel" Datenpakete verarbeiten
VIO_SendText

# VIO_Connect

Benötigt: IP-Symcon >= 5.1

```
boolean VIO_Connect (int $InstanzID, string $ClientIP, int
$ClientPort)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der zu aktualisierenden Virtual I/O |
| ClientIP | die zu verbindene ClientIP |
| ClientPort | der zu verbindene Port |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet an alle mit der Virtual I/O mit der ID **InstanzID** verbunden Instanzen ein Datenpaket vom Typ Erweitert (Socket). Dieses beinhaltet: Type = 1 (Connect), Buffer = "" (Buffer leer), ClientIP = **ClientIP** und ClientPort = **ClientPort**.

## BEISPIEL

```
// Schickt ein Datenpaket an die Virtual I/O Instanz mit der ID
12345 mit der IP-Adresse 192.168.0.8 und dem Port 502
VIO_Connect(12345, "192.168.0.8", 502);
```

# VIO_Disconnect

Benötigt: IP-Symcon >= 5.1

```
boolean VIO_Disconnect (int $InstanzID, string $ClientIP, int
$ClientPort)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der zu aktualisierenden Virtual I/O |
| ClientIP | die zu verbindene ClientIP |
| ClientPort | der zu verbindene Port |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet an alle mit der Virtual I/O mit der ID **InstanzID** verbunden Instanzen ein Datenpaket vom Typ Erweitert (Socket). Dieses beinhaltet: Type = 2 (Disconnect), Buffer = "" (Buffer leer), ClientIP = **ClientIP** und ClientPort = **ClientPort**.

## BEISPIEL

```
// Schickt ein Datenpaket an die Virtual I/O Instanz mit der ID
12345 mit der IP-Adresse 192.168.0.8 und dem Port 502
VIO_Disconnect(12345, "192.168.0.8", 502);
```

# VIO_GetPacketList

Benötigt: IP-Symcon >= 5.1

```
array VIO_GetPacketList (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der zu aktualisierenden Virtual I/O |

## RÜCKGABEWERT

liefert ein Array von erweiterten Datenpaketen

## BESCHREIBUNG

Liefert ein Array aller empfangen Datenpakete der Virtual I/O Instanz mit der ID **InstanzID** vom Typ Erweitert (Socket).

## BEISPIEL

```
// Liefert eine Array aller erweiterten Datenpakete der Virtual I/O
Instanz mit der ID 12345
$allePakete = VIO_GetPacketList(12345);
print_r($allePakete);

/* liefert z.B.:
Array
(
[0] => Array
(
[Type] => 0
[Buffer] => Hallo Welt
[ClientIP] => 192.168.0.8
[ClientPort] => 502
)

[1] => Array
(
[Type] => 0
[Buffer] => IP-Symcon
[ClientIP] => 192.168.0.4
[ClientPort] => 3777
)
```

```
...
)
*/
```

# VIO_GetTextList

Benötigt: IP-Symcon >= 5.1

```
array VIO_GetTextList (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der zu aktualisierenden Virtual I/O |

## RÜCKGABEWERT

liefert ein Array von simplen Datenpaketen

## BESCHREIBUNG

Liefert ein Array aller empfangen Datenpakete der Virtual I/O Instanz mit der ID **InstanzID** vom Typ Simpel.

## BEISPIEL

```
// Liefert eine Array aller simplen Datenpakete der Virtual I/O
Instanz mit der ID 12345
$alleTexts = VIO_GetTextList(12345);
print_r($alleTexts);

/* liefert z.B.:
Array
(
[0] => Hallo Welt
[1] => IP-Symcon
...
)
*/
```

# VIO_PushPacket

Benötigt: IP-Symcon >= 5.1

```
boolean VIO_PushPacket (int $InstanzID, string $Text, string
$ClientIP, int $ClientPort)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der zu aktualisierenden Virtual I/O |
| Text | der zusendene String |
| ClientIP | auf die zusendende ClientIP |
| ClientPort | auf den zusendenden Port |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet an alle mit der Virtual I/O mit der ID **InstanzID** verbunden Instanzen ein Datenpaket vom Typ Erweitert (Socket). Dieses beinhaltet: Type = 0 (Data), Buffer = **Text**, ClientIP = **ClientIP** und ClientPort = **ClientPort**.

## BEISPIEL

```
// Schickt auf die Virtual I/O Instanz mit der ID 12345 den Text
"Hallo Welt" mit der IP-Adresse 192.168.0.8 und dem Port 502
VIO_PushPacket(12345, "Hallo Welt", "192.168.0.8", 502);
```

# VIO_PushPacketHEX

Benötigt: IP-Symcon >= 5.1

```
boolean VIO_PushPacketHEX (int $InstanzID, string $Text, string
$ClientIP, int $ClientPort)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der zu aktualisierenden Virtual I/O |
| Text | der zusendene String in HEX |
| ClientIP | auf die zusendende ClientIP |
| ClientPort | auf den zusendenden Port |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet an alle mit der Virtual I/O mit der ID **InstanzID** verbunden Instanzen ein Datenpaket vom Typ Erweitert (Socket). Dieses beinhaltet: Type = 0 (Data), Buffer = **Text**, ClientIP = **ClientIP** und ClientPort = **ClientPort**.



Beim Parameter **Text** werden die Groß-/Kleinschreibung und Leerzeichen ignoriert. Des Weiteren sind nur die Zeichen 0..9, Aa..Ff erlaubt.

## BEISPIEL

```
// Schickt auf die Virtual I/O Instanz mit der ID 12345 den
Hexstring "1a2fee" mit der IP-Adresse 192.168.0.8 und dem Port 502
VIO_PushPacketHEX(12345, "1A 2F EE", "192.168.0.8", 502);
```

# VIO_PushText

Benötigt: IP-Symcon >= 3.3

```
boolean VIO_PushText (int $InstanzID, string $Text)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der zu aktualisierenden Virtual I/O |
| Text | der zusendene String |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet an alle mit der Virtual I/O mit der ID **InstanzID** verbunden Instanzen ein Datenpaket vom Typ Simpel. Dieses beinhaltet: Buffer = **Text**.

## BEISPIEL

```
// Schickt auf die Virtual I/O Instanz mit der ID 12345 den Text
"Hallo Welt"
VIO_PushText(12345, "Hallo Welt");
```

# VIO_PushTextHEX

Benötigt: IP-Symcon >= 3.3

```
boolean VIO_PushTextHEX (int $InstanzID, string $Text)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der zu aktualisierenden Virtual I/O |
| Text | der zusendene String in HEX |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet an alle mit der Virtual I/O mit der ID **InstanzID** verbunden Instanzen ein Datenpaket vom Typ Simpel. Dieses beinhaltet: Buffer = **Text**.



Beim Parameter **Text** werden die Groß-/Kleinschreibung und Leerzeichen ignoriert. Des Weiteren sind nur die Zeichen 0..9, Aa..Ff erlaubt.

## BEISPIEL

```
// Schickt auf die Virtual I/O Instanz mit der ID 12345 den
Hexstring "1a2fee"
VIO_PushTextHEX(12345, "1A 2F EE");
```

# VIO_SendPacket

Benötigt: IP-Symcon >= 5.1

```
boolean VIO_SendPacket (int $InstanzID, string $Text, string
$ClientIP, int $ClientPort)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der zu aktualisierenden Virtual I/O |
| Text | der zusendene String |
| ClientIP | auf die zusendende ClientIP |
| ClientPort | auf den zusendenden Port |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet an die Virtual I/O mit der ID **InstanzID** selbst ein Datenpaket vom Typ Erweitert (Socket). Dieses beinhaltet: Type = 0 (Data), Buffer = **Text**, ClientIP = **ClientIP** und ClientPort = **ClientPort**.

## BEISPIEL

```
// Schickt an die Virtual I/O Instanz mit der ID 12345 den Text
"Hallo Welt" mit der IP-Adresse 192.168.0.8 und dem Port 502
VIO_PushPacket(12345, "Hallo Welt", "192.168.0.8", 502);
```

# VIO_SendText

Benötigt: IP-Symcon >= 3.3

```
boolean VIO_SendText (int $InstanzID, string $Text)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der zu aktualisierenden Virtual I/O |
| Text | der zusendene String |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet an die Virtual I/O mit der ID **InstanzID** selbst ein Datenpaket vom Typ Simpel. Dieses beinhaltet: Buffer = **Text**.

## BEISPIEL

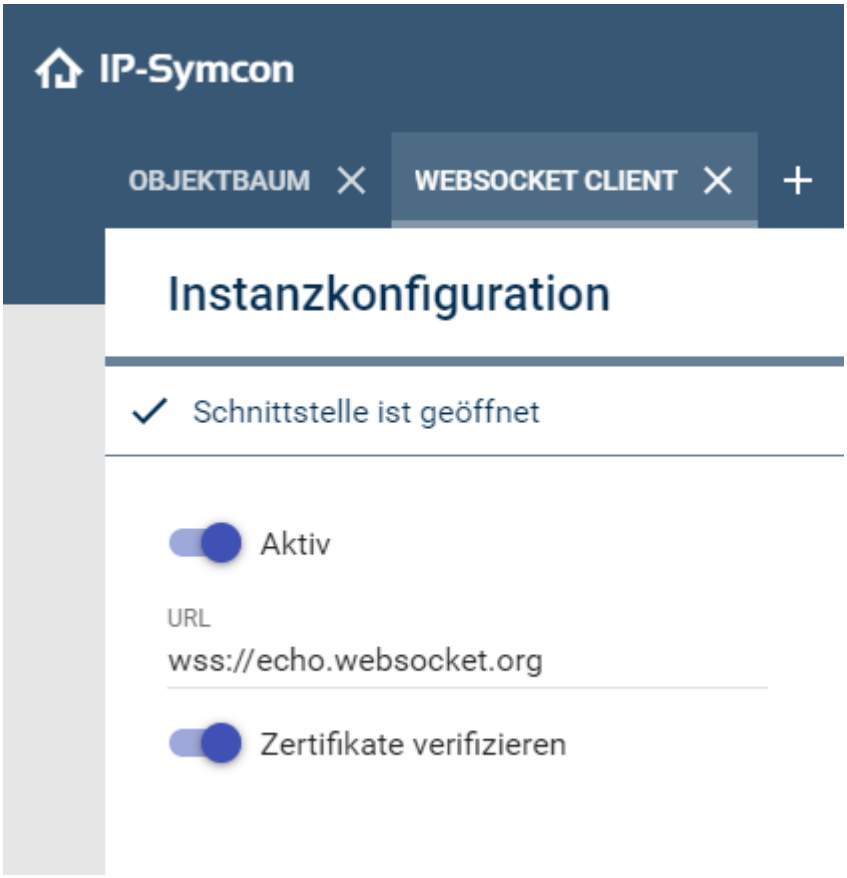
```
// Schickt an die Virtual I/O Instanz mit der ID 12345 den Text
"Hallo Welt"
VIO_SendText(12345, "Hallo Welt");
```

# WebSocket Client

Benötigt: IP-Symcon >= 5.5
Der WebSocket Client öffnet eine Schnittstelle, welche oftmals für die bidirektionale Kommunikation mit Webanwendungen/WebServer genutzt wird. Es verwendet das TCP Protokoll.

## EINBINDUNG IN IP-SYMCON

Auf der Konfigurationsseite muss unter "URL" die Adresse der/des Webanwendung/ WebServers angegeben werden. Bei aktiviertem "Zertifikate verifizieren" werden die SSL Zertifikate geprüft.
Erst ein Klick auf "Übernehmen" speichert die Einstellungen und anschließend wird versucht die Verbindung aufzubauen.



# WSC_SendMessage

Benötigt: IP-Symcon >= 5.5

```
boolean WSC_SendMessage (int $InstanzID, string $Nachricht)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu aktualisierenden WebSocket Clients |
| Nachricht | der zu sendene String |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Sendet an den WebSocket Client mit der ID **InstanzID** die Nachricht **Nachricht**.

## BEISPIEL

```
// Schickt an die WebSocket Client Instanz mit der ID 12345 die
Nachricht "Hallo Welt"
WSC_SendMessage(12345, "Hallo Welt");
```

# Datensicherung

Zur Sicherung von Systemdaten können Backup Module genutzt werden. Dies wird in regelmäßigen Abständen oder nach größeren Änderungen empfohlen.

| | |
|---|---|
| Backup | Erstellt ein Backup über SFTP, FTP oder FTPS |
| SB_CreateBackup | Erstellt oder updatet ein Backup |

# Backup

Benötigt: IP-Symcon >= 6.3
Erstellt ein Backup über SFTP, FTP oder FTPS.

## FUNKTIONSUMFANG

Erstellt ein Backup über SFTP, FTP oder FTPS auf einem Server
Aktualisiert ein Backup über SFTP, FTP oder FTPS auf einem Server

## SOFTWARE-INSTALLATION

Über den Module Store das 'Backup (SFTP/FTP/FTPS)'-Modul installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter 'Instanz hinzufügen' kann das 'Backup (SFTP/FTP/FTPS)'-Modul mithilfe des Schnellfilters gefunden werden.
Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

Konfigurationsseite:

| NAME | BESCHREIBUNG |
|---|---|
| Verbindungstyp | Auswahl der Verbindungsart |
| Host | IP Adresse des Servers |
| Port | Port auf dem der Client sich auf den Server verbindet. Bei FTPS und FTP ist d normalerweise 21. Bei SFTP ist dies der Port 22 |
| Benutzername | Benutzernamen für die SFTP Verbindung |
| Passwort | Passwort für die SFTP Verbindung |
| Modus | Modus was für ein Backup erstellt werden soll |
| Wechsel Ordner nach | Zum Anfang der Periode wird ein neuer Ordner erstellt |
| Zielordner | Ordner auf dem Server, in dem das Backup erstellt werden soll |
| Suche Zielordner | Durch die Ordner browsen und ein valider Pfad kann erstellt werden |
| Automatische Backups aktivieren | Aktiviert ein tägliches Update |
| Täglich um | Zeit, wann das Update täglich startet |
| Expertenoptionen | ---------- |
| Größen Limit | Ist eine Datei Größer als dieses Limit wird sie ignoriert |
| Gefilterte Ordner | Filter um bestimmte Ordner nicht zu übertragen |
| Backup erstellen | Button, welcher sofort ein Update startet. |
| Verbindung testen | Testet, ob eine Verbindung hergestellt werden kann |

Modus:
Vollständiges Backup: Erstellt eine vollständige Kopie in einem separaten Ordner nach

dem Pattern: symcon-backup-{Jahr}-{Monat}-{Tag}-{Stunde}-{Minute}-{Sekunde}
Inkrementelles Backup: Updatet ein bestehendes Backup.
Steht die Option 'Wechsel nach Ordner' nicht auf 'Niemals', so wird der Ordner nach der eingestellten Zeit gewechselt.
Die Ordner, welche beim inkrementellen Backup erstellt werden, folgen folgenden Pattern:

Niemals: symcon-backup
Woche: symcon-backup-{Jahr}-0{Kalenderwoche}
Monat: symcon-backup-{Jahr}-{Monat}
Jahr: symcon-backup-{Jahr}

## STATUSVARIABLEN

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Zuletzt abgeschlossenes Backup | Integer | Zeitpunkt, zudem das letzte Backup abgeschlossen wurde |
| Übertragene Megabytes | Float | Megabytes, welche beim letzten Update übertragen wurde |

# SB_CreateBackup

```
boolean SB_CreateBackup (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE

## BESCHREIBUNG

Erstellt oder updatet ein Backup nach den Einstellungen der Instanz.

## BEISPIEL

```
SB_CreateBackup(12345);
```

# Legacy

Legacy Module werden in der aktuellsten Version nicht mehr unterstützt. Dies kann z.B. daran liegen, dass diese in aktuellen Versionen nicht mehr benötigt werden.

| | |
|---|---|
| RRDTool | Direkteinbindung des RRDTool |
| RRD_Execute | gibt einen Befehl an die RRDTool Library weiter |
| Shutter Control (legacy) | Funktionen zur Ansteuerung von Shutter(Rollladen) Komponenten (legacy) |
| SC_Move | bewegt den Rollladen an eine bestimmte Position |
| SC_MoveDown | bewegt den Rollladen bis zu Endposition runter |
| SC_MoveUp | bewegt den Rollladen bis zu Endposition hoch |
| SC_Stop | stoppt einen Bewegungsvorgang |
| USBMapper | Setzt automatisch den korrekten USB-Port zu den eingetragenen Serial-Ports. |
| USBM_FixPorts | Kontroliert ob die USB-Ports noch richtig konfiguriert sind |
| Web Graph | Beliebige Graphen können über den Web Hook eigenständig angezeigt werden |
| Wunderground Weather | Holt Wetterdaten (Aktuell/stündlich/12-stündlich/Unwetterwarnung) via API von www.wunderground.com |
| WGW_UpdateStormWarningData | Aktualisiert die Unwetterwarnungen des Weathergroundmoduls |
| WGW_UpdateWeatherData | Aktualisiert die Wetterdaten (aktuell/stündlich/täglich) des Weathergroundmoduls |

# RRDTool



Diese Funktion wurde mit IP-Symcon 2.4 entfernt.

Das RRDTool Modul ist eine direkte Einbindung von RRDTool als PHP Funktion in IP-Symcon.
Es werden keine weiteren Funktionen angeboten.
Somit ist das RRDTool Handbuch die erste Anlaufstelle, um Informationen zur Handhabung dieses Moduls zu finden.
Die aktuelle Versionsnummer können sie folgendermaßen herausfinden:

```
echo RRD_Execute("version -");
```

# RRD_Execute

```
boolean RRD_Execute (int $InstanzID, string $Befehl)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID des zu schaltenden Geräts |
| Befehl | RRD Befehl. Siehe |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG



Diese Funktion wurde mit IP-Symcon 2.4 entfernt.

Gibt einen Befehl an die RRDTool Library weiter und führt ihn aus. Sobald der Befehl ausgeführt wurde, wird mit der Ausführung fortgefahren.

## BEISPIEL

```
//Kein Beispiel vorhanden
```

# Shutter Control (legacy)

Dies ist eine veraltete Version. Es ist empfohlen ab Version 5.0 die neue Variante Shutter Control zu nutzen.

Um das IP-Symcon Shutter-Modul hardwareunabhängig zu machen, braucht das Modul ein Skript, welches sich um die Weitergabe der Befehle an Ihre Hardware kümmert.
Alle, die folgende Hardware besitzen, können das Skript einfach kopieren und verwenden. Andere Hardware lässt sich leicht hinzufügen. (Mehr dazu unterhalb des Skripts).

Skripte, die durch das Shutter-Control ausgeführt werden, beinhalten diese Systemvariablen

Eaton Xcomfort Shutter
1-Wire Shutter
Homematic (Dank an hengesb )
LCN (Dank an philipp )
FS20MS
Dieses Skript muss im Normalfall nur einmal in IP-Symcon vorhanden sein und es können beliebig viele Instanzen darauf zugreifen. Es muss Nichts editiert werden.

## ROLLLADEN EINMESSEN

Beim zweiten Schritt werden die Zeiten gemessen, die der Rollladen braucht, um auf die einzelnen Positionen zu fahren. Die Messung muss zweimal durchgeführt werden. Für Auf und Ab. Zum Starten wird der entsprechende Button gedrückt, woraufhin der Rollladen sich in Bewegung setzen sollte. Der Button muss erneut gedrückt werden, wenn die auf ihm stehende Position erreicht wird.

Nach jedem Einlernen müssen die Werte mit dem “Setzen” Button übernommen werden.

## TIPPS & TRICKS

Bei einer Fahrt auf 0% und 100% werden fünf Sekunden zur Fahrzeit addiert, um eine definierte Position zu erreichen.

Sollte die Laufzeit des Rollladens oberhalb des PHP-Skript-Limits liegen (180sek) muss dieses in der php.ini erhöht werden. (siehe PHP)

## HINWEISE FÜR LCN NUTZER

Das Shuttermodul kann mit dem Script universell für die Steuerung des Rollladen/Jalousie über die Ausgänge wie auch über die Relais genutzt werden.
Soll das Shuttermodul einen Rollladen/Jalousie steuern, welche/r an den zwei Ausgängen eines UPP/SH/HU hängt, dann muss unter “Instance1” die Instanz des Ausgangs 1 des Moduls gewählt werden und für “Instance2” die Instanz des Ausgangs 2 des Moduls. Diese Belegung entspricht dann der Standardrichtung, wie sie aus der LCN-Pro mit Hoch- bzw. Runterfahren angesteuert wird. Sollte der Anschluss umgekehrt sein, muss nur die Zuordung von Ausgang 1 und Ausgang 2 zu den Einstellungen “Instance1” und “Instance2” geändert werden.
Beim Betrieb mit Relais muss für “Instance1” immer das erste Relais der beiden angegeben werden. Das erste Relais ist das, welches den Strom AN/AUS schaltet. “Instance2” ist immer das Relais, welches die Richtung steuert.
Sollte die Fahrtrichtung verdreht sein, muss dies manuell im Script geändert werden. Im “case Block für LCN” des Shutter Scriptes findet sich an den beiden Stellen ein entsprechender Kommentar.
Für den Fall, dass man unterschiedlich verdrahtete Rollladen/Jalousien an den Relais nutzt, müssen also für die jeweiligen Konfigurationen zwei Shutterscripte verwendet werden.

## SHUTTER SKRIPT

```
//Variables provided by ShutterControl Module
//IPS_LogMessage("InstanceID", $_IPS['INSTANCE']); /* InstanceID */
//IPS_LogMessage("Direction", $_IPS['DIRECTION']); /* {0..2} Stop,
Up, Down */
//IPS_LogMessage("Duration", $_IPS['DURATION']); /* ms */

if($_IPS['SENDER'] != "ShutterControl") {
die("This script can only be started by the ShutterControl Module");
}

define("SC_DIRECTION_STOP", 0);
define("SC_DIRECTION_UP", 1);
define("SC_DIRECTION_DOWN", 2);

$instance = IPS_GetInstance($_IPS['INSTANCE']);
switch($instance['ModuleInfo']['ModuleID']) {

//Siemens Device S7/LOGO
case "{932076B1-B18E-4AB6-AB6D-275ED30B62DB}":
switch($_IPS['DIRECTION']) {
case SC_DIRECTION_STOP:
S7_WriteBit($_IPS['INSTANCE'], false);
S7_WriteBit($_IPS['INSTANCE2'], false);
break;
```

```
case SC_DIRECTION_UP:
S7_WriteBit($_IPS['INSTANCE'], true);
if($_IPS['DURATION'] > 0) {
IPS_Sleep($_IPS['DURATION']);
S7_WriteBit($_IPS['INSTANCE'], false);
}
break;

case SC_DIRECTION_DOWN:
S7_WriteBit($_IPS['INSTANCE2'], true);
if($_IPS['DURATION'] > 0) {
IPS_Sleep($_IPS['DURATION']);
S7_WriteBit($_IPS['INSTANCE2'], false);
}
break;
}
break;

//FS20
case "{48FCFDC1-11A5-4309-BB0B-A0DB8042A969}":
$running = CreateVariableByName($_IPS['INSTANCE'], "Moving", 0);
$value = GetValue(IPS_GetObjectIDByIdent("StatusVariable",
$_IPS['INSTANCE']));
switch($_IPS['DIRECTION']) {
case SC_DIRECTION_STOP:
if(GetValue($running)) {
FS20_SwitchMode($_IPS['INSTANCE'], $value);
SetValue($running, false);
}
break;

case SC_DIRECTION_UP:
if(!GetValue($running)) {
FS20_SwitchMode($_IPS['INSTANCE'], true);
SetValue($running, true);
}
if($_IPS['DURATION'] > 0) {
IPS_Sleep($_IPS['DURATION']);
FS20_SwitchMode($_IPS['INSTANCE'], true);
SetValue($running, false);
}
break;

case SC_DIRECTION_DOWN:
if(!GetValue($running)) {
FS20_SwitchMode($_IPS['INSTANCE'], false);
SetValue($running, true);
}
if($_IPS['DURATION'] > 0) {
IPS_Sleep($_IPS['DURATION']);
FS20_SwitchMode($_IPS['INSTANCE'], false);
SetValue($running, false);
}
break;
}
break;

//EnOcean
```

```
case "{1463CAE7-C7D5-4623-8539-DD7ADA6E92A9}":
switch($_IPS['DIRECTION']) {
case SC_DIRECTION_STOP:
ENO_ShutterStop($_IPS['INSTANCE']);
break;

case SC_DIRECTION_UP:
ENO_ShutterMoveUp($_IPS['INSTANCE']);
if($_IPS['DURATION'] > 0) {
IPS_Sleep($_IPS['DURATION']);
ENO_ShutterStop($_IPS['INSTANCE']);
}
break;

case SC_DIRECTION_DOWN:
ENO_ShutterMoveDown($_IPS['INSTANCE']);
if($_IPS['DURATION'] > 0) {
IPS_Sleep($_IPS['DURATION']);
ENO_ShutterStop($_IPS['INSTANCE']);
}
break;
}
break;

//digitalStrom
case "{3DDA1E2B-B807-4680-AB6D-E7E8FBD6093A}":
switch($_IPS['DIRECTION']) {
case SC_DIRECTION_STOP:
DS_ShutterStop($_IPS['INSTANCE']);
break;

case SC_DIRECTION_UP:
DS_ShutterMoveUp($_IPS['INSTANCE']);
if($_IPS['DURATION'] > 0) {
IPS_Sleep($_IPS['DURATION']);
DS_ShutterStop($_IPS['INSTANCE']);
}
break;

case SC_DIRECTION_DOWN:
DS_ShutterMoveDown($_IPS['INSTANCE']);
if($_IPS['DURATION'] > 0) {
IPS_Sleep($_IPS['DURATION']);
DS_ShutterStop($_IPS['INSTANCE']);
}
break;
}
break;

//xComfort
case "{1B7B5B7D-CAA9-4AB5-B9D8-EC805EC955AD}":
switch($_IPS['DIRECTION']) {
case SC_DIRECTION_STOP:
MXC_ShutterStop($_IPS['INSTANCE']);
break;

case SC_DIRECTION_UP:
MXC_ShutterMoveUp($_IPS['INSTANCE']);
if($_IPS['DURATION'] > 0) {
```

```
IPS_Sleep($_IPS['DURATION']);
MXC_ShutterStop($_IPS['INSTANCE']);
}
break;

case SC_DIRECTION_DOWN:
MXC_ShutterMoveDown($_IPS['INSTANCE']);
if($_IPS['DURATION'] > 0) {
IPS_Sleep($_IPS['DURATION']);
MXC_ShutterStop($_IPS['INSTANCE']);
}
break;
}
break;

//LCN Unit
case "{2D871359-14D8-493F-9B01-26432E3A710F}":
$type=IPS_GetProperty($_IPS['INSTANCE'],'Unit');
switch($type) {
//Ausgänge
case 0:
switch($_IPS['DIRECTION'])
{
case SC_DIRECTION_STOP:
LCN_SetIntensity($_IPS['INSTANCE'],0,0);
LCN_SetIntensity($_IPS['INSTANCE2'],0,0);
break;

case SC_DIRECTION_UP:
LCN_SetIntensity($_IPS['INSTANCE'],100,4);
if($_IPS['DURATION'] > 0) {
IPS_Sleep($_IPS['DURATION']);
LCN_SetIntensity($_IPS['INSTANCE'],0,0);
}
break;

case SC_DIRECTION_DOWN:
LCN_SetIntensity($_IPS['INSTANCE2'],100,4);
if($_IPS['DURATION'] > 0) {
IPS_Sleep($_IPS['DURATION']);
LCN_SetIntensity($_IPS['INSTANCE2'],0,0);
}
break;
}
break;

//Relais
case 2:
switch($_IPS['DIRECTION']) {
case SC_DIRECTION_STOP:
LCN_SwitchRelay($_IPS['INSTANCE'],false);
break;

case SC_DIRECTION_UP:
LCN_SwitchRelay($_IPS['INSTANCE2'],false); // Um Relaisrichtung zu
ändern bitte auf true
LCN_SwitchRelay($_IPS['INSTANCE'],true);
if($_IPS['DURATION'] > 0) {
IPS_Sleep($_IPS['DURATION']);
```

```
LCN_SwitchRelay($_IPS['INSTANCE'],false);
}
break;

case SC_DIRECTION_DOWN:
LCN_SwitchRelay($_IPS['INSTANCE2'],true);// Um Relaisrichtung zu
ändern bitte auf false
LCN_SwitchRelay($_IPS['INSTANCE'],true);
if($_IPS['DURATION'] > 0) {
IPS_Sleep($_IPS['DURATION']);
LCN_SwitchRelay($_IPS['INSTANCE'],false);
}
break;
}
break;
}
break;

//LCN Shutter
case "{C81E019F-6341-4748-8644-1C29D99B813E}":
switch($_IPS['DIRECTION']) {
case SC_DIRECTION_STOP:
LCN_ShutterStop($_IPS['INSTANCE']);
break;

case SC_DIRECTION_UP:
LCN_ShutterMoveUp($_IPS['INSTANCE']);
if($_IPS['DURATION'] > 0) {
IPS_Sleep($_IPS['DURATION']);
LCN_ShutterStop($_IPS['INSTANCE']);
}
break;

case SC_DIRECTION_DOWN:
LCN_ShutterMoveDown($_IPS['INSTANCE']);
if($_IPS['DURATION'] > 0) {
IPS_Sleep($_IPS['DURATION']);
LCN_ShutterStop($_IPS['INSTANCE']);
}
break;
}
break;

//1-Wire Shutter Modul (e-service Online)
case "{BD0F2622-F67C-4248-9A04-316DF13914C3}":
switch($_IPS['DIRECTION']) {
case SC_DIRECTION_STOP:
OW_SetPin($_IPS['INSTANCE'], 0, true);
OW_SetPin($_IPS['INSTANCE'], 1, true);
IPS_Sleep(100);
OW_SetPin($_IPS['INSTANCE'], 0, false);
OW_SetPin($_IPS['INSTANCE'], 1, false);
break;

case SC_DIRECTION_UP:
OW_SetPin($_IPS['INSTANCE'], 0, true);
IPS_Sleep(100);
OW_SetPin($_IPS['INSTANCE'], 0, false);
if($_IPS['DURATION'] > 0) {
```

```
IPS_Sleep($_IPS['DURATION']);
OW_SetPin($_IPS['INSTANCE'], 0, true);
OW_SetPin($_IPS['INSTANCE'], 1, true);
IPS_Sleep(100);
OW_SetPin($_IPS['INSTANCE'], 0, false);
OW_SetPin($_IPS['INSTANCE'], 1, false);
}
break;

case SC_DIRECTION_DOWN:
OW_SetPin($_IPS['INSTANCE'], 1, true);
IPS_Sleep(100);
OW_SetPin($_IPS['INSTANCE'], 1, false);
if($_IPS['DURATION'] > 0) {
IPS_Sleep($_IPS['DURATION']);
OW_SetPin($_IPS['INSTANCE'], 0, true);
OW_SetPin($_IPS['INSTANCE'], 1, true);
IPS_Sleep(100);
OW_SetPin($_IPS['INSTANCE'], 0, false);
OW_SetPin($_IPS['INSTANCE'], 1, false);
}
break;
}
break;

//1-Wire Shutter (1-wire.de)
case "{6A75828A-25CD-4CF3-83EA-DAAB914030A7}":
switch($_IPS['DIRECTION']) {
case SC_DIRECTION_STOP:
OneWireShutter($_IPS['INSTANCE'], 0, $_IPS['DURATION']);
break;

case SC_DIRECTION_UP:
if($_IPS['DURATION'] == 0) {
$_IPS['DURATION'] = 120000;
}
OneWireShutter($_IPS['INSTANCE'], 0, $_IPS['DURATION']);
break;

case SC_DIRECTION_DOWN:
if($_IPS['DURATION'] == 0) {
$_IPS['DURATION'] = 120000;
}
OneWireShutter($_IPS['INSTANCE'], 1, $_IPS['DURATION']);
break;
}
break;

//Homematic Shutter
case "{EE4A81C6-5C90-4DB7-AD2F-F6BBD521412E}":
switch($_IPS['DIRECTION']) {
case SC_DIRECTION_STOP:
HM_WriteValueBoolean($_IPS['INSTANCE'], "STOP", true);
break;

case SC_DIRECTION_UP:
HM_WriteValueFloat($_IPS['INSTANCE'], "LEVEL", 1.0);
if($_IPS['DURATION'] > 0) {
IPS_Sleep($_IPS['DURATION']);
```

```
HM_WriteValueBoolean($_IPS['INSTANCE'], "STOP", true);
}
break;

case SC_DIRECTION_DOWN:
HM_WriteValueFloat($_IPS['INSTANCE'], "LEVEL", 0.0);
if($_IPS['DURATION'] > 0) {
IPS_Sleep($_IPS['DURATION']);
HM_WriteValueBoolean($_IPS['INSTANCE'], "STOP", true);
}
break;
}
break;

//DMXOUT
case "{E19C2E41-7347-4A3B-B7D9-A9A88E0D133E}":
switch($_IPS['DIRECTION']) {
case SC_DIRECTION_STOP:
DMX_SetValue($_IPS['INSTANCE'],1,0); //Relais Hoch
DMX_SetValue($_IPS['INSTANCE2'],1,0); //Relais Runter
break;

case SC_DIRECTION_UP:
DMX_SetValue($_IPS['INSTANCE2'],1,0); //Relais Runter
DMX_SetValue($_IPS['INSTANCE'],1,255); //Relais Hoch
if($_IPS['DURATION'] > 0) {
IPS_Sleep($_IPS['DURATION']);
DMX_SetValue($_IPS['INSTANCE'],1,0); //Relais Hoch
}
break;

case SC_DIRECTION_DOWN:
DMX_SetValue($_IPS['INSTANCE'],1,0); //Relais Hoch
DMX_SetValue($_IPS['INSTANCE2'],1,255); //Relais Runter
if($_IPS['DURATION'] > 0) {
IPS_Sleep($_IPS['DURATION']);
DMX_SetValue($_IPS['INSTANCE2'],1,0); //Relais Runter
}
break;
}
break;

default:
die("No Handler for Module ".$instance['ModuleInfo']['ModuleName']."
found");
}

function OneWireShutter($ins, $dir, $sec) {
@OW_SetStrobe($ins, True);
$res = ($dir * 128) + ($sec / 1000);
@OW_SetPort((integer)$ins, (integer)$res);
}

function CreateVariableByName($id, $name, $type) {
$vid = @IPS_GetVariableIDByName($name, $id);
if($vid===false) {
$vid = IPS_CreateVariable($type);
IPS_SetParent($vid, $id);
IPS_SetName($vid, $name);
```

```
IPS_SetInfo($vid, "This Variable was created by Script");
}
return $vid;
}
```

## INFORMATIONEN ZUM ERWEITERN DES SKRIPTES

Es werden bei einem Aufruf immer drei Parameter übergeben.
Die InstanzID des in dem Shutter Control Instanz eingestellten “Transmit Device”
Die Richtung, in die gefahren werden soll.
Die Zeit, die gefahren werden soll. (Dieser Wert ist 0 bei STOP und bei den TEST Buttons in der Eigenschaften-Seite)
In den “Case Statements” müssen die GUIDs für das jeweilige Modul eingefügt werden.
Alle IP-Symcon Funktionen können verwendet werden. Es darf kein Output generiert werden (echo), andernfalls gilt dies als Fehler und wird im Message-Log angezeigt und der Status der “Position” Variable wird nicht aktualisiert.

# SC_Move

```
boolean SC_Move (int $InstanzID, int $Position)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Shutter Control Instanz |
| Position | 0%-100%, 99% = Auf Spalt fahren |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Bewegt den Rollladen für das Gerät mit der ID **InstanzID** auf die **Position**.



Da einige Funk-Systeme unidirektonal sind, kann nicht festgestellt werden, ob ein Befehl erfolgreich war. Dadurch kann die genaue Positionierung nicht garantiert werden.

## BEISPIEL

```
SC_Move(12345, 50); //Auf 50% fahren
```

# SC_MoveDown

```
boolean SC_MoveDown (int $InstanzID, int $Dauer)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Shutter Control Instanz |
| Dauer | 0 = Endlagenschalter, >1 = Dauer in ms |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Bewegt den Rollladen für das Gerät mit der ID **InstanzID** für **Dauer** Millisekunden nach unten.



Die Statusvariable wird unabhängig von der gefahrenen Zeit auf 100 zurückgesetzt.

## BEISPIEL

```
SC_MoveDown(12345, 0); //Nach unten fahren
```

# SC_MoveUp

```
boolean SC_MoveUp (int $InstanzID, int $Dauer)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Shutter Control Instanz |
| Dauer | 0 = Endlagenschalter, >1 = Dauer in ms |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Bewegt den Rollladen für das Gerät mit der ID **InstanzID** für **Dauer** Millisekunden nach oben.



Die Statusvariable wird unabhängig von der gefahrenen Zeit auf 0 zurückgesetzt.

## BEISPIEL

```
SC_MoveUp(12345, 0); //Nach oben fahren
```

# SC_Stop

```
boolean SC_Stop (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Shutter Control Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Stoppt einen Bewegungsvorgang für das Gerät mit der ID InstanzID.
Die Statusvariable wird auf -1 zurückgesetzt.

## BEISPIEL

```
SC_Stop(12345); //Anhalten
```

# USBMapper

Benötigt: IP-Symcon >= 5.5



Dieses Modul ist veraltet und sollte nicht mehr verwendet werden.

Das Modul setzt automatisch den korrekten USB-Port zu den eingetragenen Serial-Ports.

## FUNKTIONSUMFANG

Das Modul mappt automatisch eingetragene SerialPorts auf den jeweiligen korrekten USB-Port.
Es können mehere Serial-Ports in der Liste eingetragen werden.
Es wird bei Start von IP-Symcon und jede Minute kontrolliert ob die SerialPorts korrekt konfiguriert sind.

## VORAUSSETZUNGEN

Linux oder Raspberry Pi

## SOFTWARE-INSTALLATION

Über den Module Store das Modul USB-Mapper installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'USBMapper'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
| --- | --- |
| Geräte | Liste, welche die zu mappenden SerialPorts beinhaltet |

# USBM_FixPorts

Benötigt: IP-Symcon >= 5.5

```
boolean USBM_FixPorts (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Kontrolliert alle, in der Liste des USBMappers mit der InstanzID $InstanzID, eingetragenen SerialPorts, ob diese noch die richten USB-Ports konfiguriert haben.

## BEISPIEL

```
USBM_FixPorts(12345);
```

# Web Graph

Benötigt: IP-Symcon >= 4.3
Das Modul dient dazu, bereits vorhandene Diagramme via WebHook zur Verfügung zu stellen. Der WebHook ist dann sowohl lokal als auch via Connect Service aufrufbar. Auch stilistische Konfigurationen sind über Parameter in der URL möglich.

## FUNKTIONSUMFANG

Erstellt einen eigenen Webhook
Hinzufügen der erlaubten ObjektIDs für Diagramme via Liste
Möglichkeit zur Sicherung durch Benutzername und Passwort
Bereitstellung via WebHook sowohl lokal oder via Connect Service
Konfiguration der Darstellung via URL Parameter

## VORAUSSETZUNGEN

Aktive Subskription bei Nutzung des Connect Services

## SOFTWARE-INSTALLATION

Über den Module Store das Modul Web Graph installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'Web Graph'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Access List | Die Liste beinhaltet die ObjektIDs von Diagrammmedien, welche via WebHook bereitgestellt werden dürfen |
| Benutzername/Passwort (Experte) | Begrenzt den Zugang mithilfe eines Benutzernamens und Passwort |
| Testumgebung | Hier können verschiedene Einstellung ausprobiert werden und die passende URL ausgelesen werden |

Nach Eintragung in die "Access List" stehen die Graphen mit ihrer ID zur Verfügung.

### BEISPIEL

Diagrammmedia mit der ID 12345 wird in die "Access List" eingetragen. Daraufhin ist diese sichtbar unter:

Lokal:
http://127.0.0.1:3777/hook/webgraph/?id=12345
Symcon Connect Service: https://xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.ipmagic.de/hook/webgraph/?id=12345
Weitere Einstellungen können an die URL angehangen werden.
/hook/webgraph/?
id=12345&startTime=&timeSpan=3&isHighDensity=1&isExtrema=&isDynamic=1&isContinuou

## STATUSVARIABLEN UND PROFILE

Es werden keine zusätzlichen Statusvariablen oder Profile erstellt. Es wird lediglich der WebHook "/hook/webgraph/" eingetragen.

## VISUALISIERUNG

Über die Visualisierung ist keine weitere Konfiguration ode Anzeige möglich.

# Wunderground Weather

Benötigt: IP-Symcon >= 4.2



Dieses Modul ist veraltet und sollte nicht mehr verwendet werden.

Das Modul fragt über die Wunderground API Wetterdaten ab.
Dafür ist eine Registrierung auf Wunderground nötig, um einen API-Key zu erhalten.
Es können aktuelle Daten, Unwetterwarnungen, sowie stündliche als auch täglichen Vorhersagen abgefragt werden.

## FUNKTIONSUMFANG

De-/Aktivierbare Abfrage von gewünschten Wetterdaten.
Einstellbarkeit der Menge der Unwetter, stündlichen und täglichen Daten.
Timer für automatische Aktualisierung der Daten.

## SOFTWARE-INSTALLATION

Über den Module Store das Modul WundergroundWeather installieren.

## EINRICHTEN DER INSTANZEN IN IP-SYMCON

Unter "Instanz hinzufügen" kann das 'WundergroundWeather'-Modul mithilfe des Schnellfilters gefunden werden.

Weitere Informationen zum Hinzufügen von Instanzen in der Dokumentation der Instanzen

### KONFIGURATIONSSEITE:

| NAME | BESCHREIBUNG |
|---|---|
| Standort | Standort, von dem die Daten entnommen werden sollen. Ob ein Standort verfügbar ist, kann auf der wunderground.com Seite ausprobiert werden. Sollte ein Standort nicht vorhanden sein, sollte ein nächstgelegener größerer Ort gewählt werden. |
| Land | Hier muss das Land eingetragen werden. |
| PWS-ID | ID der 'Personal Weatherstation' Hiermit kann man eine spezifische Wetterstation abfragen |
| API Key | Wunderground API-Key. Kann auf der Wunderground Homepage nach Registrierung angefordert werden. "More"->"Weather API for Developers". |
| Aktuelle Daten abfragen | Aktiviert die Abfrage der aktuellen Wetterdaten. |
| Stündliche Vorhersage | Aktiviert die Abfrage der stündlichen Vorhersage. |
| 12stündliche Vorhersage | Aktiviert die Abfrage der 12-stündlichen Vorhersage. |
| tägliche Vorhersage | Aktiviert die Abfrage der täglichen Vorhersage. |
| Unwetterwarnung abfragen | Aktiviert die Abfrage der Unwetter Vorhersage. |
| Anzahl Vorhersagen (12-stündlich) | Die Anzahl der 12-stündlichen Vorhersagen. Maximalwert: 8 |
| Anzahl Vorhersagen (täglich) | Die Anzahl der täglichen Vorhersagen. Maximalwert: 4 |
| Anzahl Vorhersagen (stündlich) | Die Anzahl der stündlichen Vorhersagen. Maximalwert: 24 |
| Anzahl Unwetterwarnung | Die Anzahl der Unwetter Vorhersagen. Maximalwert: 6 |
| Update Wetterdaten | Setzt den Timer in Minuten, wie oft die Wetterdaten aktualisiert werden sollen. (aktuell/stündlich/12-stündlich) |
| Update Unwetterwarnungen | Setzt den Timer in Minuten, wie oft die Unwetterwarnungen aktualisiert werden sollen. |
| Button Update Wetter | Aktualisiert die Wetterdaten (aktuell/stündlich/12-stündlich/ täglich). Sofern alle drei Abfragen deaktiviert sind oder der Timer auf 0 gesetzt ist => Timer deaktiviert |
| Button Update Unwetterwarnungen | Aktualisiert die Unwetterwarnungen. Sofern die Unwetterwarnungsabfrage deaktiviert oder der Timer auf 0 gesetzt ist => Timer deaktiviert |

## STATUSVARIABLEN UND PROFILE

Die Statusvariablen/Kategorien werden automatisch angelegt. Das Löschen einzelner kann zu Fehlfunktionen führen.

### AKTUELLE WETTERDATEN

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Luftdruck | Float | Angabe in hPa |
| Luftfeuchtigkeit | Float | Angabe in % |
| Niederschlag/h | Float | Angabe in Liter/m² |
| Niederschlag Tag | Float | Angabe in Liter/m² |
| Sichtweite | Float | Angabe in km |
| Sonnenstrahlung | Float | Angabe in W/m² |
| Temperatur | Float | Angabe in °C |
| Temperatur gefühlt | Float | Angabe in °C |
| Temperatur Taupunkt | Float | Angabe in °C |
| UV Strahlung | Integer | Informationen: UVIndex Erklärung |
| Windböe | Float | Angabe in km/h |
| Windgeschwindigkeit | Float | Angabe in km/h |
| Windrichtung | Float | Angabe in Himmelsrichtungen |

## STÜNDLICHE VORHERSAGE

Die Variablen werden mit 1..24h gekennzeichnet. (1 = Vorhersage nächste volle Stunde; 24 = Vorhersage der 24ten vollen Stunde)

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Gegebenheit | String | Beschreibt das Wetter z.B. "Bedeckt", "Regen möglich" |
| Luftfeuchtigkeit | Float | Angabe in % |
| Luftdruck | Float | Angabe in hPa |
| Regenmenge | Float | Angabe in Liter/m² |
| Regenwahrscheinlichkeit | Integer | Angabe in % |
| Temperatur | Float | Angabe in °C |
| Wolkendecke | Integer | Angabe in % |
| Windgeschwindigkeit | Float | Angabe in km/h |
| Windrichtung | Float | Angabe in Himmelsrichtungen |

## 12-STÜNDLICHE VORHERSAGE

Die Variablen werden mit 12, 24..96h gekennzeichnet (12 = Vorhersage in 12 Stunden; 96 = Vorhersage in 96 Stunden)
**Hinweis: die 12h-stündliche Vorhersage wird von Wunderground nicht mehr geliefert. Die Variablen bleiben aus Kompatibilitätsgründen erhalten, enthalten aber die Werte der täglichen Vorhersage.**

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Höchsttemperatur | Float | Angabe in °C |
| Tiefsttemperatur | Float | Angabe in °C |

### TÄGLICHE VORHERSAGE

Die Variablen werden mit 1..4d gekennzeichnet (1 = morgen; 2 = übermorgen ... )

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Gegebenheit | String | Beschreibt das Wetter z.B. "Bedeckt", "Regen möglich" |
| Höchsttemperatur | Float | Angabe in °C |
| Tiefsttemperatur | Float | Angabe in °C |
| Luftfeuchtigkeit | Float | Angabe in % |
| Regenmenge | Float | Angabe in Liter/m² |
| Regenwahrscheinlichkeit | Integer | Angabe in % |
| Windgeschwindigkeit | Float | Angabe in km/h |
| Windrichtung | Float | Angabe in Himmelsrichtungen |

### UNWETTERWARNUNG

| NAME | TYP | BESCHREIBUNG |
|---|---|---|
| Beschreibung | String | Beschreibt die Warnung mit möglichen weiteren Informationen wie z. B. Windgeschwindigkeiten oder Regenmengen. |
| Datum | Integer | Angabe in UnixTimeStamp. Datum wann die Warnung ausgesprochen wurde. |
| Name | String | Ausgeschriebener Typ. z.B. Gewitter |
| Typ | String | 3-Buchstabenkürzel für die Warnung Übersicht |

### PROFILE:

| NAME | TYP |
|---|---|
| WGW.Rainfall | Float |
| WGW.Sunray | Float |
| WGW.Visibility | Float |
| WGW.WindSpeedkmh | Float |
| WGW.UVIndex | Integer |
| WGW.ProbabilityOfRain | Integer |

# WGW_UpdateStormWarningData

Benötigt: IP-Symcon >= 4.2

```
boolean WGW_UpdateStormWarningData (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Aktualisiert die Unwetterwarnungen des Weathergroundmoduls mit der InstanzID **InstanzID**.

## BEISPIEL

```
WGW_UpdateStormWarningData(12345);
```

# WGW_UpdateWeatherData

Benötigt: IP-Symcon >= 4.2

```
boolean WGW_UpdateWeatherData (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Aktualisiert die Wetterdaten (aktuell/stündlich/täglich) des Weathergroundmoduls mit der InstanzID **InstanzID**.

## BEISPIEL

```
WGW_UpdateWeatherData(12345);
```

# Befehlsreferenz

Die Befehlsreferenz beinhaltet die Befehle, welche IP-Symcon zur Verfügung stellt und zur eigenen Verwaltung nutzt.
Hier befindet sich eine Auflistung der IP-Symcon spezifischen Befehle.



Eine Auflistung der modulspezifischen Befehle kann unter Modulreferenz gefunden werden.

## ABLAUFSTEUERUNG

Funktionen in dieser Gruppe erlauben es, auf den Ablauf eines Skripts Einfluss zu nehmen.

| | |
|---|---|
| IPS_Execute | startet ein externes Programm |
| IPS_ExecuteEx | startet ein externes Programm im Userkontext |
| IPS_RunAction | führt eine einzelne Aktion aus |
| IPS_RunActionWait | führt eine einzelne Aktion aus und wartet auf das Resultat |
| IPS_RunScript | startet ein anderes IP-Symcon Skript |
| IPS_RunScriptEx | startet ein anderes IP-Symcon Skript und übergibt Variablen |
| IPS_RunScriptText | startet einen Text als IP-Symcon Skript |
| IPS_RunScriptTextEx | startet einen Text als IP-Symcon Skript und übergibt Variablen |
| IPS_RunScriptTextWait | startet einen Text als IP-Symcon Skript |
| IPS_RunScriptTextWaitEx | startet einen Text als IP-Symcon Skript und übergibt Variablen |
| IPS_RunScriptWait | startet ein anderes IP-Symcon Skript und wartet auf das Resultat |
| IPS_RunScriptWaitEx | startet ein anderes IP-Symcon Skript und übergibt Variablen |
| IPS_SemaphoreEnter | erlaubt die Synchronisation mit anderen, gleichzeitig laufenden Skripts |
| IPS_SemaphoreLeave | erlaubt die Synchronisation mit anderen, gleichzeitig laufenden Skripts |
| IPS_Sleep | verzögert ein Skript um ein bestimme zeit |

## EREIGNISVERWALTUNG

Funktionen in dieser Gruppe dienen der Verwaltung von Kategorien.

| | |
|---|---|
| IPS_CreateEvent | legt ein neues Ereignis an |
| IPS_DeleteEvent | löscht eine bestehendes Ereignis |
| IPS_EventExists | prüft, ob ein Ereignis bereits existiert |
| IPS_GetEvent | liefert umfassende Informationen über ein Ereignis |
| IPS_GetEventIDByName | ermittelt die ID eines Ereignisses anhand des Namens |
| IPS_GetEventList | liefert eine Liste aller existierenden Ereignisse |
| IPS_GetEventListByType | liefert eine Liste aller Ereignisse eines bestimmten Typs |
| IPS_IsConditionPassing | prüft ob eine Bedingung erfüllt ist |
| IPS_SetEventAction | setzt die auszuführende Aktion |
| IPS_SetEventActive | aktiviert/deaktiviert ein Ereignis |
| IPS_SetEventCondition | modifiziert die Bedingung eines Ereignisses |
| IPS_SetEventConditionDateRule | modifiziert die Regel einer Bedingung eines Ereignisses für Datumsvergleiche |
| IPS_SetEventConditionDayOfTheWeekRule | modifiziert die Regel einer Bedingung eines Ereignisses für Wochentagsvergleiche |
| IPS_SetEventConditionTimeRule | modifiziert die Regel einer Bedingung eines Ereignisses für Uhrzeitvergleiche |
| IPS_SetEventConditionVariableDynamicRule | modifiziert die Regel einer Bedingung eines Ereignisses für Variablenvergleiche |
| IPS_SetEventConditionVariableRule | modifiziert die Regel einer Bedingung eines Ereignisses für Variablenvergleiche |
| IPS_SetEventConditionVariableStaticRule | modifiziert die Regel einer Bedingung eines Ereignisses für Variablenvergleiche |
| IPS_SetEventCyclic | konfiguriert ein zyklisches Ereignis |
| IPS_SetEventCyclicDateBounds | konfiguriert die Zeitspanne (Datum) |
| IPS_SetEventCyclicDateFrom | konfiguriert das Startdatum |
| IPS_SetEventCyclicDateTo | konfiguriert das Enddatum |
| IPS_SetEventCyclicTimeBounds | konfiguriert die Zeitspanne (Zeit) |
| IPS_SetEventCyclicTimeFrom | konfiguriert die Startzeit |
| IPS_SetEventCyclicTimeTo | konfiguriert die Endzeit |
| IPS_SetEventLimit | limitiert die Anzahl der Ereignissaufrufe |
| IPS_SetEventScheduleAction | modifiziert die Aktionsstabelle eines Wochenplans und setzt die Aktion dabei auf "Führe PHP Code aus" |
| IPS_SetEventScheduleActionEx | modifiziert die Aktionsstabelle eines Wochenplans |
| IPS_SetEventScheduleGroup | modifiziert die Gruppierung der Tage eines Wochenplans |
| IPS_SetEventScheduleGroupPoint | modifiziert einen Schaltpunkt der Gruppe eines Wochenplans |
| IPS_SetEventScript | setzt die Aktion auf "Führe PHP Code aus" und definiert den auszuführenden Code |

# INSTANZENVERWALTUNG

Funktionen in dieser Gruppe dienen der Verwaltung von Instanzen.

Debug

- IPS_EnableDebugFile — aktiviert das Schreiben von Debug-Logs in eine Datei

IPS_CreateInstance — legt eine Instanz an

IPS_DeleteInstance — löscht eine Instanz

IPS_GetInstance — liefert umfassende Informationen zu einer bestimmten Instanz

IPS_GetInstanceIDByName — liefert die ID einer Instanz anhand des Namens

IPS_GetInstanceList — liefert eine Aufstellung aller existierenden Instanzen

IPS_GetInstanceListByModuleID — liefert alle Instanzen die von einer bestimmten ModulID sind

IPS_GetInstanceListByModuleType — liefert eine Liste aller Instanzen eines bestimmten Typs

IPS_InstanceExists — prüft, ob eine bestimmte Instanz existiert

Konfiguration

- IPS_ApplyChanges — übernimmt eine veränderte Konfiguration
- IPS_GetConfiguration — liest eine Konfiguration aus
- IPS_GetConfigurationForm — liest eine Konfigurationseite aus
- IPS_GetProperty — liest den aktuellen Wert einer Eigenschaft
- IPS_HasChanges — prüft, ob eine Konfiguration verändert wurde
- IPS_ResetChanges — setzt eine veränderte Konfiguration auf die aktuelle zurück
- IPS_SetConfiguration — setzt eine neue Konfiguration
- IPS_SetProperty — setzt einen neuen Wert für eine Eigenschaft

Referenzen

- IPS_GetReferenceList — gibt die IDs aller Referenzen wieder

StatusVariablen

Verbindungen

- IPS_ConnectInstance — erstellt eine (Daten-)Verbindung zwischen zwei Instanzen
- IPS_DisconnectInstance — löst eine (Daten-)Verbindung zwischen zwei Instanzen
- IPS_GetCompatibleInstances — liefert kompatible Instanzen zu einer Instanz
- IPS_IsInstanceCompatible — prüft, ob eine Verbindung zwischen zwei Instanzen möglich ist

## KATEGORIEVERWALTUNG

Funktionen in dieser Gruppe dienen der Verwaltung von Kategorien.

| | |
|---|---|
| IPS_CategoryExists | prüft, ob eine Kategorie bereits existiert |
| IPS_CreateCategory | legt eine neue Kategorie an |
| IPS_DeleteCategory | löscht eine bestehende Kategorie |
| IPS_GetCategoryIDByName | sucht und liefert die ID einer bestehenden Kategorie über dessen Namen |
| IPS_GetCategoryList | liefert eine Liste aller existierenden Kategorien |

## LINKVERWALTUNG

Funktionen in dieser Gruppe dienen der Verwaltung von Links.

| | |
|---|---|
| IPS_CreateLink | legt einen neuen Link an |
| IPS_DeleteLink | löscht einen bestehenden Link |
| IPS_GetLink | liefert umfassende Informationen über einen Link |
| IPS_GetLinkIDByName | ermittelt die ID eines Links anhand des Namens |
| IPS_GetLinkList | liefert eine Liste aller existierenden Links |
| IPS_LinkExists | prüft, ob ein Link bereits existiert |
| IPS_SetLinkTargetID | setzt die Verknüpfung für den Link |

## MEDIENVERWALTUNG

Funktionen in dieser Gruppe dienen der Medienverwaltung.

| | |
|---|---|
| IPS_CreateMedia | legt ein neues Medienobjekt im MedienPool an |
| IPS_DeleteMedia | löscht ein Medienobjekt aus dem MedienPool |
| IPS_GetMedia | liefert umfassende Information über ein bestimmtes Medienobjekt |
| IPS_GetMediaContent | liefert den Inhalt eines Medienobjekts |
| IPS_GetMediaID | ermittelt die ID eines Medienobjekts |
| IPS_GetMediaIDByFile | ermittelt die ID eines Medienobjekts anhand der Datei |
| IPS_GetMediaIDByName | ermittelt die ID eines Medienobjekts anhand des Namens |
| IPS_GetMediaList | liefert eine Liste aller Medienobjekte im MedienPool |
| IPS_GetMediaListByType | liefert eine Liste aller Medienobjekte eines bestimmten Typs |
| IPS_MediaExists | prüft, ob sich ein bestimmtes Medienobjekt bereits im MedienPool befindet |
| IPS_SendMediaEvent | sendet eine Mitteilung, dass ein Medienobjekt sich geändert hat |
| IPS_SetMediaCached | aktiviert, dass ein Medienobjekt nur im Arbeitsspeicher bearbeitet wird |
| IPS_SetMediaContent | setzt den Inhalt eines Medienobjekts |
| IPS_SetMediaFile | bindet eine Mediendatei an ein Medienobjekt |

## MODULVERWALTUNG

Funktionen in dieser Gruppe dienen der Verwaltung von Modulen und Bibliotheken.

| | |
|---|---|
| IPS_GetCompatibleModules | liefert alle Kompatiblen Module zu einem bestimmten Modul |
| IPS_GetLibrary | gibt Details zu einer Bibliothek |
| IPS_GetLibraryList | liefert eine Liste der Bibliotheken |
| IPS_GetLibraryModules | gibt eine Liste der Module einer Bibliothek |
| IPS_GetModule | gibt Details zu einem Modul |
| IPS_GetModuleList | gibt eine Liste aller Module |
| IPS_GetModuleListByType | gibt eine Liste aller Module des typs |
| IPS_IsModuleCompatible | überprüft, ob zwei Modul miteinander Kompatibel sind |
| IPS_LibraryExists | überprüft, ob Bibliothek existiert |
| IPS_ModuleExists | überprüft, ob Modul existiert |

## OBJEKTVERWALTUNG

Funktionen in dieser Gruppe erlauben es, Objekte zu manipulieren.

| | |
|---|---|
| IPS_GetChildrenIDs | liefert eine Aufstellung aller untergeordneten Objekte |
| IPS_GetLocation | liefert den kompletten Pfad des Objekts |
| IPS_GetName | liefert den Namen des Objekts |
| IPS_GetObject | liefert umfassende Informationen über ein Objekt |
| IPS_GetObjectIDByIdent | liefert die ID eines Objekts anhand des Identifikators |
| IPS_GetObjectIDByName | liefert die ID eines Objekts anhand des Namens |
| IPS_GetObjectList | liefert eine Aufstellung aller existierenden Objekte |
| IPS_GetParent | liefert die ParentID des Objekts |
| IPS_HasChildren | prüft, ob Objekt untergeordnete Objekte hat |
| IPS_IsChild | prüft, ob Objekt einem anderen Objekt untergeordnet ist |
| IPS_ObjectExists | prüft, ob ein bestimmtes Objekt existiert |
| IPS_SetDisabled | Setzt ein Objekt für das WebFront inaktiv |
| IPS_SetHidden | setzt die Sichtbarkeit eines beliebigen Objekts |
| IPS_SetIcon | ordnet einem beliebigen Objekt ein Icon zu |
| IPS_SetIdent | ordnet einem beliebigen Objekt einen Identifikator zu |
| IPS_SetInfo | ordnet einem beliebigen Objekt erweiterte Informationen zu |
| IPS_SetName | ordnet einem beliebigen Objekt einen Namen zu |
| IPS_SetParent | verschiebt ein Objekt im logischen Strukturbaum |
| IPS_SetPosition | ordnet einem beliebigen Objekt eine Position zu |

## PROGRAMMINFORMATIONEN

Funktionen in dieser Gruppe liefern Informationen zu IP-Symcon selbst.

| | |
|---|---|
| IPS_FunctionExists | prüft, ob eine bestimmte Funktion existiert |
| IPS_GetFunction | liefert umfassende Informationen über eine Funktion |
| IPS_GetFunctionList | liefert eine Liste mit allen Verfügbaren Funktionsnamen |
| IPS_GetFunctionListByModuleID | liefert eine Liste mit allen verfügbaren Funktionen eines Moduls |
| IPS_GetKernelArchitecture | liefert die Architektur für die IP-Symcon erstellt wurde |
| IPS_GetKernelDate | liefert das Datum der Erstellung des Kernels |
| IPS_GetKernelDir | liefert den Pfad zum IP-Symcon Programmverzeichnis |
| IPS_GetKernelDirEx | liefert den Pfad zum IP-Symcon Installationsverzeichnis |
| IPS_GetKernelDirSpace | Liefert freien/verfügbaren/gesamt Speicherplatz auf dem Laufwerk, auf dem alle Daten von IP-Symcon gespeichert werden |
| IPS_GetKernelPlatform | liefert das Betriebssystem auf dem IP-Symcon läuft |
| IPS_GetKernelRevision | liefert eine eindeutige Revisionskennung |
| IPS_GetKernelRunlevel | liefert den Runlevel des Kernels |
| IPS_GetKernelStartTime | liefert den Zeitpunkt, an dem IP-Symcon gestartet wurde |
| IPS_GetKernelVersion | liefert die Versionsnummer des verwendeten Kernels |
| IPS_GetLogDir | liefert den Pfad zum IP-Symcon Logverzeichnis |
| IPS_GetSystemLanguage | liefert die Sytemsprache des Betriebssystems auf dem IP-Symcon läuft |
| IPS_GetUpdateChannel | liefert den aktuellen konfigurierten Update-Kanal zurück |
| IPS_LogMessage | schreibt eine benutzerdefinierte Meldung in die Log-Datei |

## SKRIPTVERWALTUNG

Funktionen in dieser Gruppe dienen der Skriptsteuerung und -verwaltung.

| | |
|---|---|
| IPS_CreateScript | legt ein neues Skript an |
| IPS_DeleteScript | entfernt eine bestehendes Skript aus der Skriptliste |
| IPS_GetScript | liefert umfassende Informationen über ein Skript |
| IPS_GetScriptContent | liefert den Inhalt eines Skripts |
| IPS_GetScriptEventList | liefert alle Ereignisse die einem Skript zugeordnet sind |
| IPS_GetScriptFile | ermittelt den Dateinamen zur SkriptID |
| IPS_GetScriptIDByFile | liefert die ID eines Skripts andhand der Datei |
| IPS_GetScriptIDByName | liefert die ID eines Skripts andhand des Namens |
| IPS_GetScriptList | liefert eine Aufstellung aller existierenden Skripte |
| IPS_GetScriptTimer | ermittelt den Startwert des ScriptTimers |
| IPS_ScriptExists | prüft, ob ein bestimmtes Skript existiert |
| IPS_SetScriptContent | setzt den Inhalt eines Skripts |
| IPS_SetScriptFile | ordnet einem Skript einen Dateinamen zu |
| IPS_SetScriptTimer | setzt den Startwert des ScriptTimers |

# VARIABLENVERWALTUNG

Funktionen in dieser Gruppe dienen der Variablenvarwaltung und Variablenwertänderung.

| | |
|---|---|
| IPS_CreateVariable | legt eine Variable an |
| IPS_DeleteVariable | löscht eine Variable |
| IPS_GetVariable | liefert umfassende Informationen zu einer bestimmten Variablen |
| IPS_GetVariableEventList | liefert alle Ereignisse die einer Variable zugeordnet sind |
| IPS_GetVariableIDByName | liefert die ID einer Variablen anhand des Namens |
| IPS_GetVariableList | liefert eine Liste aller existierenden Variablen |
| IPS_GetVariablePresentation | Gibt alle Parameter der aktuell genutzten Darstellung zurück |
| IPS_SetVariableCustomAction | ordnet der Variable ein benutzerdefiniertes Aktionsskript zu |
| IPS_SetVariableCustomPresentation | ordnet der Variable eine benutzerdefinierte Darstellung zu |
| IPS_SetVariableCustomProfile | ordnet der Variable ein benutzerdefiniertes Profil zu |
| IPS_VariableExists | prüft, ob eine bestimmte Variable existiert |
| Variablendarstellung | Funktionen in dieser Gruppe dienen der Verwaltung von Variablendarstellungen |
| IPS_GetPresentation | liefert umfassende Informationen über eine bestimmte Darstellung |
| IPS_GetPresentationForm | liefert das Konfigurationsformular einer Darstellung für einen Variablentypen |
| IPS_GetPresentations | liefert alle Darstellungen des Systems |
| IPS_PresentationExists | prüft ob eine bestimmte Darstellung existiert |
| Vorlagenverwaltung | Funktionen in dieser Gruppe dienen der Verwaltung von Vorlagen |

Variablenprofile

| | |
|---|---|
| IPS_CreateVariableProfile | legt ein Variablenprofil an |
| IPS_DeleteVariableProfile | löscht ein Variablenprofil |
| IPS_GetVariableProfile | liefert umfassende Informationen zu einem bestimmten Variablenprofil |
| IPS_GetVariableProfileList | liefert eine Liste aller existierenden Variablenprofile |
| IPS_GetVariableProfileListByType | liefert eine Liste aller existierenden Variablenprofile eines bestimmten Typs |
| IPS_SetVariableProfileAssociation | modifiziert die Assoziationstabelle eines Variablenprofils |
| IPS_SetVariableProfileDigits | setzt die Anzahl der Nachkommastellen eines Variablenprofils |
| IPS_SetVariableProfileIcon | setzt das Icon eines Variablenprofils |
| IPS_SetVariableProfileText | setzt das Prefix und Suffix eines Variablenprofils |
| IPS_SetVariableProfileValues | setzt den Minimalwert, Maximalwert und die Schrittweite eines Variablenprofils |
| IPS_VariableProfileExists | prüft, ob ein bestimmtes Variablenprofil existiert |

# VARIABLENZUGRIFF

Funktionen in dieser Gruppe sind für den lesenden und schreibenden Zugriff auf IP-Symcon Variable

| | |
|---|---|
| GetValue | holt den Wert einer IP-Symcon Variable unabhängig des Typs |
| GetValueBoolean | holt den Wert einer IP-Symcon Variable vom Typ Boolean |
| GetValueFloat | holt den Wert einer IP-Symcon Variable vom Typ Float |
| GetValueFormatted | holt den Wert einer IP-Symcon Variable und formatiert ihn anhand des Profils |
| GetValueFormattedEx | gibt einen Wert formatiert anhand des Profils einer Variable zurück |
| GetValueInteger | holt den Wert einer IP-Symcon Variable vom Typ Integer |
| GetValueString | holt den Wert einer IP-Symcon Variable vom Typ String |
| HasAction | prüft, ob eine Aktion definiert ist |
| RequestAction | führe die Aktion einer IP-Symcon Variable aus |
| RequestActionEx | führe die Aktion einer IP-Symcon Variable mit individuellen Sender aus |
| SetValue | setzt den Wert einer IP-Symcon Variable unabhängig des Typs |
| SetValueBoolean | setzt den Wert einer IP-Symcon Variable vom Typ Boolean |
| SetValueFloat | setzt den Wert einer IP-Symcon Variable vom Typ Float |
| SetValueInteger | setzt den Wert einer IP-Symcon Variable vom Typ Integer |
| SetValueString | setzt den Wert einer IP-Symcon Variable vom Typ String |

# Ablaufsteuerung

| | |
|---|---|
| IPS_Execute | startet ein externes Programm |
| IPS_ExecuteEx | startet ein externes Programm im Userkontext |
| IPS_RunAction | führt eine einzelne Aktion aus |
| IPS_RunActionWait | führt eine einzelne Aktion aus und wartet auf das Resultat |
| IPS_RunScript | startet ein anderes IP-Symcon Skript |
| IPS_RunScriptEx | startet ein anderes IP-Symcon Skript und übergibt Variablen |
| IPS_RunScriptText | startet einen Text als IP-Symcon Skript |
| IPS_RunScriptTextEx | startet einen Text als IP-Symcon Skript und übergibt Variablen |
| IPS_RunScriptTextWait | startet einen Text als IP-Symcon Skript |
| IPS_RunScriptTextWaitEx | startet einen Text als IP-Symcon Skript und übergibt Variablen |
| IPS_RunScriptWait | startet ein anderes IP-Symcon Skript und wartet auf das Resultat |
| IPS_RunScriptWaitEx | startet ein anderes IP-Symcon Skript und übergibt Variablen |
| IPS_SemaphoreEnter | erlaubt die Synchronisation mit anderen, gleichzeitig laufenden Skripts |
| IPS_SemaphoreLeave | erlaubt die Synchronisation mit anderen, gleichzeitig laufenden Skripts |
| IPS_Sleep | verzögert ein Skript um ein bestimme zeit |

# IPS_Execute

```
boolean IPS_Execute (string $Programmpfad, string $Parameter,
boolean $Dummy, boolean $Warten)
```

## PARAMETERLISTE

| | |
|---|---|
| Programmpfad | Vollständiger Pfad zum Programm |
| Parameter | An das Programm zu übergebende Parameter (optional) |
| Dummy | Immer true angeben, Parameter wird nicht ausgewertet, dieser wurde nur in Version 1.x genutzt. |
| Warten | Gibt an, ob auf das Programmende gewartet werden soll. |

## RÜCKGABEWERT

Den Rückgabewert von stderr/stdout, wenn der **Warten** Parameter auf True ist, sonst ist die Rückgabe ein leerer String.

## BESCHREIBUNG

Die Funktion veranlasst den Start des externen Programms mit dem Namen **ProgrammName**. Dem Programm wird der **String**-Parameter als Aufrufparameter übergeben. Der Boolean-Wert **Warten** legt fest, ob das Programmende abgewartet oder das aufrufende Skript sofort fortgesetzt werden soll. Wird auf das Ergebnis gewartet, erscheint dieses, falls vorhanden, als String im Rückgabewert.
Als Quelle für den Rückgabewert kommt eine der folgenden Programmausgaben in Frage:

Ausgabe nach **stderror**
Ausgabe nach **stdout**

Werden vom Programm beide Möglichkeiten genutzt, erhält **stderror** den Vorrang, um z.B. Fehlermeldungen nicht zu verlieren.



**Vorsicht:** Beim Warten auf das Programmende verlängert sich die Laufzeit des Skripts entsprechend. Dabei ist darauf zu achten, dass die in der Datei "**php.ini**" definierte maximale Laufzeit nicht überschritten wird. Andernfalls wird das Skript mit einer Fehlermeldung abgebrochen. Dies ist vor allem dann wichtig, wenn das Programm Benutzereingaben erwartet.



**Achtung:** Wenn das zu startende Programm interaktiv ist (z.B. ein Fenster beinhaltet), wird dieses nicht angezeigt/ sichtbar. Es wird im Hintergrund als System-Prozess gestartet und kann dann nur mit Hilfe des Task-Managers gestoppt werden. Soll das Programm auf dem Desktop angezeigt werden, muss die IPS_ExecuteEx Funktion verwendet werden.

## BEISPIEL

```
//Eine Batchdatei starten
IPS_Execute("C:/autoexec.bat", "", false, false);
```

# IPS_ExecuteEx

```
string IPS_ExecuteEx (string $Programmpfad, string $Parameter,
boolean $FensterZeigen, boolean $Warten, int $SessionID)
```

## PARAMETERLISTE

| | |
|---|---|
| Programmpfad | Vollständiger Pfad zum Programm |
| Parameter | An das Programm zu übergebende Parameter (optional) |
| FensterZeigen | **True**, wenn das Fenster angezeigt werden soll; **False**, wenn das Fenster unsichbar bleiben soll |
| Warten | Gibt an, ob aus das Programmende gewartet werden soll. |
| SessionID | Die User Session ID, die genutzt werden soll (Unter XP ab 0, 2003/Vista ab 1), < 0 für automatische Auswahl |

## RÜCKGABEWERT

Leerstring

## BESCHREIBUNG

Die Funktion veranlasst den Start des externen Programms mit dem Namen **ProgrammName**. Dem Programm wird der String-**Parameter** als Aufrufparameter übergeben. Der Boolean-Wert **Warten** legt fest, ob das Programmende abgewartet oder das aufrufende Skript sofort fortgesetzt werden soll.
Der Parameter SessionID gibt an, in welcher User-Session das Programm gestartet werden soll. (Unter Windows 2003 können z.B. mehrere Terminal Sessions parallel gestartet sein.) Wird der Parameter kleiner 0 gewählt, so ermittelt IP-Symcon die erste verfügbare Session automatisch und startet dort das gewünschte Programm.

**Vorsicht:** Beim Warten auf das Programmende verlängert sich die Laufzeit des Skripts entsprechend. Dabei ist darauf zu achten, dass die in der Datei "**php.ini**" definierte maximale Laufzeit nicht überschritten wird. Andernfalls wird das Skript mit einer Fehlermeldung abgebrochen. Dies ist vor allem dann wichtig, wenn das Programm Benutzereingaben erwartet.

## BEISPIEL

```
// Notepad starten
```

```
IPS_ExecuteEx("C:\\Windows\\notepad.exe", "", true, false, -1);
```

# IPS_RunAction

```
boolean IPS_RunAction (string $AktionsID, array $AktionsParameter)
```

## PARAMETERLISTE

| | |
|---|---|
| AktionsID | GUID der auszuführenden Aktion |
| AktionsParameter | Liste der Parameter der auszuführenden Aktion |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion veranlasst die Ausführung der Aktion mit der GUID **AktionsID** und den Parametern **AktionsParameter**. Die Ausführung läuft parallel (gleichzeitig) zum aufrufenden Skript.
Hierbei müssen die Aktionsparameter folgende Einträge beinhalten:

| PARAMETER | BESCHREIBUNG |
|---|---|
| TARGET | ID des Zielobjektes der Aktion (-1 bei allgemeinen Aktionen) |
| ENVIRONMENT | Die Ausführumgebung der Aktion, siehe auch Umgebungen |
| PARENT | ID des ausführenden Objekts der Aktion, beispielsweise das Skript oder der Ablaufplan |

## BEISPIEL

```
// Führe Skript mit der ID 12345 aus
IPS_RunAction("{7938A5A2-0981-5FE0-BE6C-8AA610D654EB}", ["TARGET" =>
12345, "ENVIRONMENT" => "Default", "PARENT" => $_IPS['SELF']]);

// Zeige alle Aktionen und dazugehörigen GUIDs an
foreach(json_decode(IPS_GetActions(), true) as $action) {
echo $action['id'] . " -> " . $action['caption'] . PHP_EOL;
// var_dump($action); <-- Gibt alle Details wie z.B. restrictions
aus
}
```

# IPS_RunActionWait

```
string IPS_RunActionWait (string $AktionsID, array
$AktionsParameter)
```

## PARAMETERLISTE

| | |
|---|---|
| AktionsID | ID der auszuführenden Aktion |
| AktionsParameter | Liste der Parameter der auszuführenden Aktion |

## RÜCKGABEWERT

Fehlermeldung der Ausführung

## BESCHREIBUNG

Die Funktion veranlasst die Ausführung der Aktion mit der ID **AktionsID** und den Parametern **AktionsParameter**. Die Aktion wird ausgeführt und wartet auf das Ende um ggfs. eine Fehlermeldung zurückzugeben. Lief die Ausführung fehlerfrei, so gibt die Funktion einen leeren String zurück.
Hierbei müssen die Aktionsparameter folgende Einträge beinhalten:

| PARAMETER | BESCHREIBUNG |
|---|---|
| TARGET | ID des Zielobjektes der Aktion (-1 bei allgemeinen Aktionen) |
| ENVIRONMENT | Die Ausführumgebung der Aktion, siehe auch Umgebungen |
| PARENT | ID des ausführenden Objekts der Aktion, beispielsweise das Skript oder der Ablaufplan |

## BEISPIEL

```
// Führe Skript mit der ID 12345 aus
$error = IPS_RunActionWait("{7938A5A2-0981-5FE0-BE6C-8AA610D654EB}",
["TARGET" => 12345, "ENVIRONMENT" => "Default", "PARENT" =>
$_IPS['SELF']]);
if ($error === "") {
echo "Aktion wurde fehlerfrei ausgeführt";
}
else {
echo "Bei der Ausführung gab es einen Fehler: " . $error;
}
```

# IPS_RunScript

```
boolean IPS_RunScript (int $SkriptID)
```

## PARAMETERLISTE

| | |
|---|---|
| SkriptID | Eindeutige ID des Skriptes |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion veranlasst den Start des Skripts mit der ID **SkriptID**. Das aufgerufene Skript läuft parallel (gleichzeitig) zum aufrufenden Skript.



Skripte, die durch die IPS_RunScript ausgeführt werden, beinhalten diese Systemvariablen

**Vergleich zwischen IPS_RunScript und der PHP-Anweisung include:**
**include** ist eine PHP-Anweisung, keine IPS-Funktion. Das durch **include** eingebundene Skript wird so abgearbeitet, als stünde es anstelle der **include**-Anweisung. Es wirkt sich so aus, als wäre es ein Bestandteil des aufrufenden Skripts. Das bedeutet, dass alle Variablen, die dem aufrufenden Skript bekannt sind auch dem eingebundenen Skript zur Verfügung stehen. Die Laufzeit des Skripts verlängert sich um die Laufzeit des eingebundenen Skripts.
Bei Verwendung von **IPS_RunScript** wird das aufgerufene Skript in einem eigenen Kontext gestartet. Es weiß also folglich nichts von den Variablen des aufrufenden Skripts. Daten können hier nur über externe Variablen ausgetauscht werden. Die Funktion ist geeignet für Situationen, bei denen ein Skript zeitgerecht ausgeführt werden muss, aber kein direkter Bezug zum aufrufenden Skript erforderlich ist. Die Laufzeit des Skripts ändert sich nicht, da beide Skripte (quasi) gleichzeitig ausgeführt werden.

## BEISPIEL

```
IPS_RunScript(12345 /*[Gartenbeleuchtung An]*/);
```

# IPS_RunScriptEx

```
boolean IPS_RunScriptEx (int $SkriptID, array $Parameter)
```

## PARAMETERLISTE

| | |
|---|---|
| SkriptID | Eindeutige ID des Skriptes |
| Parameter | Key (string) => Value (variant) Paar auf die im neu ausgeführten Skript zu gegriffen werden kann. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion veranlasst den Start des Skripts mit der ID **SkriptID** und übergibt dabei das Array Parameter als Argument. Das aufgerufene Skript läuft parallel (gleichzeitig) zum aufrufenden Skript. Auch hier gelten die bei IPS_RunScript beschriebenen Hinweise.



Skripte, die durch die IPS_RunScriptEx ausgeführt werden, beinhalten diese Systemvariablen

Das Array **Parameter** kann beliebig viele **"Key => Value"** Paare enthalten (siehe PHP Arrays). Jedes Paar erscheint im aufgerufenen Skript als Variable **Key** mit ihrem Inhalt **Value**.

## BEISPIEL

```
//Skript, dass ein anderes Skript mit Parameterübergabe starten will
IPS_RunScriptEx(12345 /*[Temp]*/, Array("Titel" => "Temp.", "Tmin"
=> 10.0));

//Skript, das aufgerufen wurde. Parameter sind als einzelne
Variablen in der
//globalen Variablen $_IPS verfügbar. Der Variablenname entspricht
dem im Array-Index
//übergebenen Namen.
$Ueberschrift = $_IPS['Titel']. "verlauf"; //Ergibt: Temp.verlauf
$MaxTemp = $_IPS['Tmin'] + 30.0;
```

# IPS_RunScriptText

```
boolean IPS_RunScriptText (string $SkriptText)
```

## PARAMETERLISTE

| | |
|---|---|
| SkriptText | String welcher als Skript aufgerufen werden soll. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Die Funktion startet ein Skript mit dem Inhalt SkriptText. Es ist nicht nötig PHP-Tags mit anzugeben.
Bei Verwendung von IPS_RunScriptText wird der als Skript aufgerufene Text in einem eigenen Kontext gestartet. Es weiß also folglich nichts von den Variablen des aufrufenden Skripts. Daten können hier nur über externe Variablen ausgetauscht werden. Die Funktion ist geeignet für Situationen, bei denen ein Skript zeitgerecht ausgeführt werden muss, aber kein direkter Bezug zum aufrufenden Skript erforderlich ist. Die Laufzeit des Skripts ändert sich nicht, da beide Skripte (quasi) gleichzeitig ausgeführt werden.

## BEISPIEL

```
// Gibt "Hallo Welt" im Meldungsfenster zurück
IPS_RunScriptText("echo 'Hallo Welt';");
```

# IPS_RunScriptTextEx

```
boolean IPS_RunScriptTextEx (string $SkriptText, array $Parameter)
```

## PARAMETERLISTE

| | |
|---|---|
| SkriptText | String welcher als Skript aufgerufen werden soll. |
| Parameter | Key (string) => Value (variant) Paar auf die im neu ausgeführten Skript zugegriffen werden kann. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion startet ein Skript mit dem Inhalt **SkriptText**. Es ist nicht nötig PHP-Tags mit anzugeben.
Bei Verwendung von **IPS_RunScriptTextEx** wird der als Skript aufgerufene Text in einem eigenen Kontext gestartet. Es weiß also folglich nichts von den Variablen des aufrufenden Skripts. Daten können hier nur über externe Variablen ausgetauscht werden. Die Funktion ist geeignet für Situationen, bei denen ein Skript zeitgerecht ausgeführt werden muss, aber kein direkter Bezug zum aufrufenden Skript erforderlich ist. Die Laufzeit des Skripts ändert sich nicht, da beide Skripte (quasi) gleichzeitig ausgeführt werden.
Das Array **Parameter** kann beliebig viele **"Key => Value"** Paare enthalten (siehe PHP Arrays). Jedes Paar erscheint im aufgerufenen Skript als Variable **Key** mit ihrem Inhalt **Value**.

## BEISPIEL

```
// Gibt "Hallo Welt" im Meldungsfenster zurück
IPS_RunScriptTextEx('echo $_IPS["Anfang"] . " ". $_IPS["Ende"];',
Array("Anfang" => "Hallo", "Ende" => "Welt"));
```

# IPS_RunScriptTextWait

```
string IPS_RunScriptTextWait (string $SkriptText)
```

## PARAMETERLISTE

| | |
|---|---|
| SkriptText | String welcher als Skript aufgerufen werden soll. |

## RÜCKGABEWERT

Resultat des ausgeführten Skripts

## BESCHREIBUNG

Die Funktion startet ein Skript mit dem Inhalt **SkriptText** und wartet auf das Ende der Skriptausführung, um das Resultat als Rückgabewert zu übergeben. Es ist nicht nötig PHP-Tags mit anzugeben.

## BEISPIEL

```
// Gibt "Hallo Welt" zurück
echo IPS_RunScriptTextWait("echo 'Hallo Welt';");
```

# IPS_RunScriptTextWaitEx

```
string IPS_RunScriptTextWaitEx (string $SkriptText, array
$Parameter)
```

## PARAMETERLISTE

| | |
|---|---|
| SkriptText | String welcher als Skript aufgerufen werden soll. |
| Parameter | Key (string) => Value (variant) Paar auf die im neu ausgeführten Skript zugegriffen werden kann. |

## RÜCKGABEWERT

Resultat des ausgeführten Skripts

## BESCHREIBUNG

Die Funktion startet ein Skript mit dem Inhalt **SkriptText**, übergibt dabei das Array Parameter als Argument und wartet auf das Ende der Skriptausführung, um das Resultat als Rückgabewert zu übergeben.. Es ist nicht nötig PHP-Tags mit anzugeben.
Das Array **Parameter** kann beliebig viele **"Key => Value"** Paare enthalten (siehe PHP Arrays). Jedes Paar erscheint im aufgerufenen Skript als Variable **Key** mit ihrem Inhalt **Value**.

## BEISPIEL

```
//Skript, dass einen Text als Skript mit Parameterübergabe starten
will
echo IPS_RunScriptTextWaitEx('echo $_IPS["Titel"]. "verlauf" .
PHP_EOL; echo $_IPS["Tmin"] + 30.0;', Array("Titel" => "Temp.",
"Tmin" => 10.0));
```

# IPS_RunScriptWait

```
string IPS_RunScriptWait (int $SkriptID)
```

## PARAMETERLISTE

| | |
|---|---|
| SkriptID | Eindeutige ID des Skriptes |

## RÜCKGABEWERT

Resultat des ausgeführten Skripts

## BESCHREIBUNG

Die Funktion veranlasst den Start des Skripts mit der ID **SkriptID**. Das Skript wird aufgerufen und wartet auf das Ende, um das Resultat als Rückgabewert zu übergeben. Auch hier gelten die bei IPS_RunScript beschriebenen Hinweise.

## BEISPIEL

```
echo IPS_RunScriptWait(12345 /*[Skript A]*/);
```

# IPS_RunScriptWaitEx

```
string IPS_RunScriptWaitEx (int $SkriptID, array $Parameter)
```

## PARAMETERLISTE

| | |
|---|---|
| SkriptID | Eindeutige ID des Skriptes |
| Parameter | Key (string) => Value (variant) Paar auf die im neu ausgeführten Skript zugegriffen werden kann. |

## RÜCKGABEWERT

Resultat des ausgeführten Skripts

## BESCHREIBUNG

Die Funktion veranlasst den Start des Skripts mit der ID **SkriptID**, übergibt dabei das Array Parameter als Argument und wartet auf das Ende der Skriptausführung, um das Resultat als Rückgabewert zu übergeben. Auch hier gelten die bei IPS_RunScript beschriebenen Hinweise.
Das Array **Parameter** kann beliebig viele **"Key => Value"** Paare enthalten (siehe PHP Arrays). Jedes Paar erscheint im aufgerufenen Skript als Variable **Key** mit ihrem Inhalt **Value**.

## BEISPIEL

```
//Skript, dass ein anderes Skript mit Parameterübergabe starten will
echo IPS_RunScriptWaitEx(12345 /*[Temp]*/, Array("Titel" => "Temp.",
"Tmin" => 10.0));

//Skript, das aufgerufen wurde. Parameter sind als einzelne
Variablen in der
//globalen Variablen $_IPS verfügbar. Der Variablenname entspricht
dem im Array-Index
//übergebenen Namen.
$Ueberschrift = $_IPS['Titel']. "verlauf"; //Ergibt: Temp.verlauf
echo $_IPS['Tmin'] + 30.0;
```

# IPS_SemaphoreEnter

```
boolean IPS_SemaphoreEnter (string $Name, int $Wartezeit)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name, der die Semaphore beschreibt |
| Wartezeit | Millisekunden, die gewartet wird, bis die Funktion abbricht |

## RÜCKGABEWERT

TRUE, wenn die Semaphore betreten wurde. FALSE, wenn die Semaphore vor Ablauf der Wartezeit nicht betreten werden konnte

## BESCHREIBUNG

Diese Funktion setzt ein **Semaphor** (ein Zeichen mit Signalwirkung), das dazu verwendet werden kann, einen Teil des Skriptes exklusiv auszuführen und diesen vor konkurrierenden Skripten zu schützen.
In IP-Symcon können grundsätzlich mehrere Skripte gleichzeitig laufen. Solange ein Skript keinen exklusiven Zugriff auf Variablen oder System-Ressourcen benötigt, stellt dieser Fall kein Problem dar. Falls aber ein exklusiver Zugriff unumgänglich ist, kann mit der Funktion ein Semaphor gesetzt werden. Die Funktion prüft zuerst, ob das Semaphor mit dem angegebenen Namen bereits gesetzt ist. Existiert es noch nicht, wird es durch die Funktion gesetzt und liefert **TRUE** als Rückmeldung. Alle anderen Skripte, die danach versuchen das selbe Semaphor zu setzen, versuchen für die Dauer **Wartezeit** weiterhin das Semaphor zu setzen. Ist es dabei erfolgreich, wird **TRUE** zurückgemeldet, andernfalls, nach Ablauf der Wartezeit, **FALSE**.
Wenn man gewährleisten will, dass während der Laufzeit eines Skripts die von ihm verwendeten Daten nicht von anderen, gleichzeitig laufenden Skripten verfälscht werden, kann man diese Funktion verwenden, um sich exklusiven Zugriff zu sichern.
Bei der Verwendung dieser Funktion ist darauf zu achten, dass das Semaphor nach Abarbeiten des kritischen Teils wieder mit IPS_SemaphoreLeave zurückgesetzt wird.

## BEISPIEL

```
if (IPS_SemaphoreEnter("KritischerPunkt", 1000))
{
// ...Kritischer Codeabschnitt

//Semaphore wieder freigeben!
IPS_SemaphoreLeave("KritischerPunkt");
}
else
```

```
{
// ...Keine ausführung Möglich. Ein anderes Skript nutzt den
"KritischenPunkt"
// für länger als 1 Sekunde, sodass unsere Wartezeit überschritten
wird.
}
```

# IPS_SemaphoreLeave

```
boolean IPS_SemaphoreLeave (string $Name)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name, der die Semaphore beschreibt. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Diese Funktion löscht das zuvor mit IPS_SemaphoreEnter gesetzte Semaphor.

## BEISPIEL

```
//Siehe IPS_SemaphoreEnter()
```

# IPS_Sleep

```
boolean IPS_Sleep (int $Wartezeit)
```

## PARAMETERLISTE

| | |
|---|---|
| Wartezeit | Gibt die Wartezeit in Millisekunden an |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion setzt die Abarbeitung des Skripts für **Wartezeit** Millisekunden aus. Nach Ablauf dieser Zeit wird das Skript fortgesetzt. Mit Hilfe dieser Funktion lassen sich **kurze** (!) Wartezeiten realisieren. Längere Wartezeiten sollten nicht durch die Sleep Funktion realisiert, sondern über geeignete **Ereignisse** gesteuert werden.
Da sich die Laufzeit des Skripts um **Wartezeit** verlängert, ist darauf zu achten, dass die maximale Laufzeit (30 Sekunden) nicht überschritten wird. Andernfalls wird das Skript mit einer Fehlermeldung abgebrochen. Längere Wartezeiten im Bereich von mehreren Sekunden und länger sollten mit anderen Methoden realisiert werden.



IPS_Sleep hat keinen Einfluss auf andere gleichzeitig laufende Skripte.



Es können auch alternativ die PHP Befehle sleep(int sekunden) oder usleep(int Mikrosekunden) genutzt werden.

## BEISPIEL

```
// Daten zeitverzögert an den COM Port ausgeben
COMPort_SendText($id, chr(0x1b)); // ESC an LC-Display senden
IPS_Sleep(200); // 200ms warten
COMPort_SendText($id, "0"); // LC-Display löschen
IPS_Sleep(200); // 200ms warten
COMPort_SendText($id, "Guten Tag!"); // Text ausgeben
```

```
// 2 Sekunden warten mit php Alternativen
sleep(2);
IPS_Sleep(2000);
usleep(2000000);
```

# Ereignisverwaltung

| | |
|---|---|
| IPS_CreateEvent | legt ein neues Ereignis an |
| IPS_DeleteEvent | löscht eine bestehendes Ereignis |
| IPS_EventExists | prüft, ob ein Ereignis bereits existiert |
| IPS_GetEvent | liefert umfassende Informationen über ein Ereignis |
| IPS_GetEventIDByName | ermittelt die ID eines Ereignisses anhand des Namens |
| IPS_GetEventList | liefert eine Liste aller existierenden Ereignisse |
| IPS_GetEventListByType | liefert eine Liste aller Ereignisse eines bestimmten Typs |
| IPS_IsConditionPassing | prüft ob eine Bedingung erfüllt ist |
| IPS_SetEventAction | setzt die auszuführende Aktion |
| IPS_SetEventActive | aktiviert/deaktiviert ein Ereignis |
| IPS_SetEventCondition | modifiziert die Bedingung eines Ereignisses |
| IPS_SetEventConditionDateRule | modifiziert die Regel einer Bedingung eines Ereignisses für Datumsvergleiche |
| IPS_SetEventConditionDayOfTheWeekRule | modifiziert die Regel einer Bedingung eines Ereignisses für Wochentagsvergleiche |
| IPS_SetEventConditionTimeRule | modifiziert die Regel einer Bedingung eines Ereignisses für Uhrzeitvergleiche |
| IPS_SetEventConditionVariableDynamicRule | modifiziert die Regel einer Bedingung eines Ereignisses für Variablenvergleiche |
| IPS_SetEventConditionVariableRule | modifiziert die Regel einer Bedingung eines Ereignisses für Variablenvergleiche |
| IPS_SetEventConditionVariableStaticRule | modifiziert die Regel einer Bedingung eines Ereignisses für Variablenvergleiche |
| IPS_SetEventCyclic | konfiguriert ein zyklisches Ereignis |
| IPS_SetEventCyclicDateBounds | konfiguriert die Zeitspanne (Datum) |
| IPS_SetEventCyclicDateFrom | konfiguriert das Startdatum |
| IPS_SetEventCyclicDateTo | konfiguriert das Enddatum |
| IPS_SetEventCyclicTimeBounds | konfiguriert die Zeitspanne (Zeit) |
| IPS_SetEventCyclicTimeFrom | konfiguriert die Startzeit |
| IPS_SetEventCyclicTimeTo | konfiguriert die Endzeit |
| IPS_SetEventLimit | limitiert die Anzahl der Ereignissaufrufe |
| IPS_SetEventScheduleAction | modifiziert die Aktionsstabelle eines Wochenplans und setzt die Aktion dabei auf "Führe PHP Code aus" |
| IPS_SetEventScheduleActionEx | modifiziert die Aktionsstabelle eines Wochenplans |
| IPS_SetEventScheduleGroup | modifiziert die Gruppierung der Tage eines Wochenplans |
| IPS_SetEventScheduleGroupPoint | modifiziert einen Schaltpunkt der Gruppe eines Wochenplans |
| IPS_SetEventScript | setzt die Aktion auf "Führe PHP Code aus" und definiert den auszuführenden Code |

# IPS_CreateEvent

```
int IPS_CreateEvent (int $EreignisTyp)
```

## PARAMETERLISTE

| WERT | BESCHREIBUNG |
|---|---|
| EreignisTyp | |
| 0 | Legt ein **“ausgelöstes”** Ereignis an |
| 1 | Legt ein **“zyklisches”** Ereignis an |
| 2 | Legt ein **“Wochenplan”** Ereignis an |

## RÜCKGABEWERT

ID des neu erstellten Ereignisses

## BESCHREIBUNG

Die Funktion legt ein Ereignisobjekt vom Typ **EreignisTyp** an.
Die Funktion liefert eine ID, mit deren Hilfe das erzeugte Ereignisobjekt eindeutig identifiziert werden kann. Das erzeugte Objekt ist nach dem Anlegen noch umkonfiguriert und muss mit den Funktionen IPS_SetEventActive, IPS_SetEventScript und den entsprechenden IPS_SetEventCyclic / IPS_SetEventTrigger konfiguriert werden.

## BEISPIEL

```
$eid = IPS_CreateEvent(0); //Ausgelöstes Ereignis
IPS_SetEventTrigger($eid, 1, 15754); //Bei Änderung von Variable mit
ID 15754
IPS_SetParent($eid, $_IPS['SELF']); //Ereignis zuordnen
IPS_SetEventActive($eid, true); //Ereignis aktivieren

$eid = IPS_CreateEvent(1); //Zyklisches Ereignis
IPS_SetEventCyclic($eid, 2, 1, 0, 0, 3, 6); //Jeden Tag alle 6
Stunden
IPS_SetEventCyclicDateFrom($eid, 1, 12, date("Y")); //Vom 1.12
diesen Jahres
IPS_SetEventCyclicDateTo($eid, 31, 12, date("Y")); //Bis 31.12
```

```
diesen Jahres

IPS_SetEventCyclicTimeFrom($eid, 15, 0, 0); //Von 15:00
IPS_SetEventCyclicTimeTo($eid, 23, 30, 0); //Bis 23:30

IPS_SetParent($eid, $_IPS['SELF']); //Eregnis zuordnen
IPS_SetEventActive($eid, true); //Ereignis aktivieren

// Seit IP-Symcon 6.0 erforderlich, sofern das Ereignis eine
Automation ausführen soll (z.B. ein PHP-Skript)
IPS_SetEventAction($eid, '{7938A5A2-0981-5FE0-BE6C-8AA610D654EB}',
[]);
```

# IPS_DeleteEvent

```
boolean IPS_DeleteEvent (int $EreignisID)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu löschenden Ereignisses |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion löscht ein vorhandenes Ereignis mit der ID **EreignisID**.



Das Ereignis darf nur gelöscht werden, wenn keine Unterobjekte mehr vorhanden sind.

## BEISPIEL

```
IPS_DeleteEvent($EreignisID);
```

# IPS_EventExists

```
boolean IPS_EventExists (int $EreignisID)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu prüfenden Ereignisses |

## RÜCKGABEWERT

Falls die EreignisID im System existiert, wird **TRUE** zurückgemeldet, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion überprüft ob ein Ereignis mit der ID **EreignisID** existiert.

## BEISPIEL

```
if (IPS_EventExists(34881))
echo "Ein Ereignis mit dieser ID existiert!";
```

# IPS_GetEvent

Benötigt: IP-Symcon >= 3.1

```
array IPS_GetEvent (int $EreignisID)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | Die zu zeigende EreignisID |

## RÜCKGABEWERT

Folgende Informationen stehen als **key => value** Paare zur Verfügung:

| INDEX | TYPE | BESCHREIBUNG |
|---|---|---|
| CyclicDateDay | integer | Datumstage. Siehe IPS_SetEventCyclic |
| CyclicDateDayValue | integer | Datumstagesintervall. Siehe IPS_SetEventCyclic |
| CyclicDateFrom | array | Startdatum als Array von Tag, Monat, Jahr; 0/0/0 = Immer. Siehe IPS_SetEventCyclicDateFrom |
| CyclicDateTo | array | Enddatum als Array von Tag, Monat, Jahr; 0/0/0 = Nie. Siehe IPS_SetEventCyclicDateTo |
| CyclicDateType | integer | Datumstyp. Siehe IPS_SetEventCyclic |
| CyclicDateValue | integer | Datumsintervall. Siehe IPS_SetEventCyclic |
| CyclicTimeFrom | array | Startzeit als Array von Stunde, Minute, Sekunde; 0/0/0 = Mitternacht. Siehe IPS_SetEventCyclicTimeFrom |
| CyclicTimeTo | array | Endzeit als Array von Stunde, Minute, Sekunde; 0/0/0 = Nie. Siehe IPS_SetEventCyclicTimeTo |
| CyclicTimeType | integer | Zeittyp. Siehe IPS_SetEventCyclic |
| CyclicTimeValue | integer | Zeitintervall. Siehe IPS_SetEventCyclic |
| EventActionID | string | ID der auszuführenden Aktion |
| EventActionParameters | array | Parameter der auszuführenden Aktion |
| EventActive | boolean | TRUE, wenn das Ereignis aktiv ist |
| EventConditions | array | Array von Bedingungen. Siehe IPS_SetEventCondition, IPS_SetEventConditionDateRule, IPS_SetEventConditionDayOfTheWeekRule, IPS_SetEventConditionTimeRule und IPS_SetEventConditionVariableRule |
| EventID | integer | ID des Ereignisses |
| EventLimit | integer | Restliche Anzahl der Ausführungen. 0 = Kein Limit |
| EventScript | string | Immer ein leerer String, da dieses Feld nur noch aus Kompatibilitätsgründen vorhanden ist. Die Ausführung wird immer über die Aktion, welche über IPS_SetEventScheduleAction konfiguriert wird, gesteuert. |
| EventType | integer | Ereignis-Typ: (0: Auslöser, 1: Zyklisch, 2: Wochenplan) |
| LastRun | float | Unix Timestamp der letzten Ausführung, 0 = Nie |
| NextRun | float | Unix Timestamp der nächsten Ausführung, 0 = Nie |
| LastActionID | integer | Aktion ID vom Wochenplan der letzten Ausführung, 0 = Keine Aktion wurde ausgeführt |
| NextActionID | integer | Aktion ID vom Wochenplan der nächsten Ausführung, 0 = Keine Aktion wird nächstes mal ausgeführt |
| ScheduleActions | array | Array von Aktionen, welche dem Wochenplan zugehörig sind. Siehe IPS_SetEventScheduleActionEx |

## BESCHREIBUNG

Die Funktion liefert ein Array mit Informationen über Ereignis mit der ID **EreignisID**.

Seit IP-Symcon 3.1 sind die Felder CyclicDateFrom, CyclicDateTo, CyclicTimeFrom und CyclicTimeTo jeweils Arrays. Wer Kompatibilität zur Version 3.0 und älter benötigt, kann auf die seit IP-Symcon 3.1 verfügbare Funktion IPS_GetEventCompatibility zurückgreifen, welche die alte Funktionsweise wiederherstellt.

LastActionID/NextActionID sind seit Version 8.1 verfügbar

## BEISPIEL

```
$EreignisID = 46413;
$EreignisInfo = IPS_GetEvent($EreignisID);
print_r($EreignisInfo);

/* liefert z.B.:
Array
(
[EventID] => 41227
[EventType] => 2
[EventActive] => 1
[EventLimit] => 0
[EventConditions] => Array
(
[0] => Array
(
[ID] => 0
[ParentID] => 0
[VariableRules] => Array
(
[0] => Array
```

```
(
[ID] => 1
[VariableID] => 29025
[Comparison] => 4
[Value] => 500
)

)

[DateRules] => Array
(
[0] => Array
(
[ID] => 0
[Comparison] => 0
[Value] => Array
(
[Day] => 8
[Month] => 8
[Year] => 2001
)
)

[1] => Array(3)
(
[ID] => 1
[Comparison] => 4
[Value] => Array
(
[Day] => 1
[Month] => 1
[Year] => 2021
)
)
)

[TimeRules] => Array
(
[0] => Array
(
[ID] => 1
[Comparison] => 3
[Value] => Array
(
[Hour] => 9
[Minute] => 0
[Second] => 0
)

)

[1] => Array
(
[ID] => 2
[Comparison] => 5
```

```
[Value] => Array
(
[Hour] => 17
[Minute] => 0
[Second] => 0
)

)

)
[DayOfTheWeekRules] => Array
(
[0] => Array
(
[ID] => 0
[Comparison] => 0
[Value] => 3
)

)

[Operation] => 1
)
)

[TriggerType] => 0
[TriggerVariableID] => 0
[TriggerValue] =>
[TriggerSubsequentExecution] =>
[CyclicDateType] => 0
[CyclicDateValue] => 0
[CyclicDateDay] => 0
[CyclicDateDayValue] => 0
[CyclicDateFrom] => Array
(
[Day] => 0
[Month] => 0
[Year] => 0
)

[CyclicDateTo] => Array
(
[Day] => 0
[Month] => 0
[Year] => 0
)

[CyclicTimeType] => 0
[CyclicTimeValue] => 0
[CyclicTimeFrom] => Array
(
[Hour] => 0
[Minute] => 0
[Second] => 0
)
```

```
[CyclicTimeTo] => Array
(
[Hour] => 0
[Minute] => 0
[Second] => 0
)

[ScheduleActions] => Array
(
[0] => Array
(
[ID] => 0
[Name] => Kalt
[Color] => 255
[ScriptText] =>
[ActionID] => {3644F802-C152-464A-868A-242C2A3DEC5C}
[ActionParameters] => Array
(
[VALUE] => 2
)
)

[1] => Array
(
[ID] => 1
[Name] => Warm
[Color] => 16711680
[ScriptText] =>
[ActionID] => {3644F802-C152-464A-868A-242C2A3DEC5C}
[ActionParameters] => Array
(
[VALUE] => 0
)
)

)

[ScheduleGroups] => Array
(
[0] => Array
(
[ID] => 0
[Days] => 127
[Points] => Array
(
[0] => Array
(
[ID] => 13
[Start] => Array
(
[Hour] => 0
[Minute] => 0
[Second] => 0
)
```

```
[ActionID] => 1
)

[1] => Array
(
[ID] => 14
[Start] => Array
(
[Hour] => 12
[Minute] => 0
[Second] => 0
)

[ActionID] => 0
)

)

)

)

[EventScript] =>
[LastRun] => 0
[NextRun] => 0
[LastActionID] => 0
[NextActionID] => 0
)
*/
```

# IPS_GetEventIDByName

```
int IPS_GetEventIDByName (string $EreignisName, int
$ÜbergeordneteID)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisName | Zu suchender EreignisName |
| ÜbergeordneteID | Objekt, in dessen direkten Unterobjekten gesucht werden soll |

## RÜCKGABEWERT

ID des gefundenen Ereignisses, sonst FALSE

## BESCHREIBUNG

Die Funktion versucht die ID des Ereignisses mit dem Namen **EreignisName**, dessen übergeordnetes Objekt die ID **ÜbergeordneteID** besitzt, zu ermitteln. Es wird die ID des ersten gefundenen Ereignisses verwendet, dessen Name mit **EreignisName** übereinstimmt. Falls ein solches Ereignis gefunden wurde, wird dessen ID zurückgemeldet, andernfalls FALSE.

**Vorsicht:** Da Namen in IP-Symcon beliebig vergeben werden können, und damit **nicht eindeutig** sind, kann die Funktion eine falsche ID zurückmelden.

Diese Funktion generiert eine **Warnung**, sollte der Name nicht gefunden werden. Benutzen Sie den @ Operator, falls Sie diese Meldung unterdrücken wollen. Die genaue Funktionsweise dieses Operators kann dem PHP Handbuch entnommen werden.

## BEISPIEL

```
$EreignisID = @IPS_GetEventIDByName("TimerABC", $ParentID);
if ($EreignisID === false)
echo "Ereignis nicht gefunden!";
else
echo "Die Ereignis-ID lautet: ". $EreignisID;
```

# IPS_GetEventList

```
array IPS_GetEventList ()
```

## RÜCKGABEWERT

Ein Array von Integer-Werten aller IDs der Ereignisse in IP-Symcon

## BESCHREIBUNG

Die Funktion ermittelt die IDs aller in IP-Symcon registrierten Ereignisse. Die IDs werden in einem Array aufgelistet. Falls kein Ereignis existiert, ist das Array leer.

## BEISPIEL

```
$alleEreignisse = IPS_GetEventList();
print_r($alleEreignisse);

/* liefert z.B.:
Array
(
[0] => 37659
[1] => 18326
usw. ...
)
*/
```

# IPS_GetEventListByType

```
array IPS_GetEventListByType (int $EreignisTyp)
```

## PARAMETERLISTE

| WERT | BESCHREIBUNG |
|---|---|
| EreignisTyp | |
| 0 | Ausgelöstes Ereignis |
| 1 | Zyklisches Ereignis |
| 2 | Wochenplan Ereignis |

## RÜCKGABEWERT

Ein Array von Integer-Werten aller IDs der Ereignisse vom Typ EreignisTyp in IP-Symcon

## BESCHREIBUNG

Die Funktion ermittelt die IDs aller Ereignisse eines bestimmten **EreignisTyps**. Die IDs werden in einem Array aufgelistet. Falls kein Ereignis des angeforderten Typs existiert, ist das Array leer.
Es werden nur Objekte des Ereignistyps **EreignisTyp** aufgelistet. Sollen alle Ereignisse aufgelistet werden, so kann hierfür die Funktion IPS_GetEventList verwendet werden.

## BEISPIEL

```
$alleEreignisse = IPS_GetEventListByType(1); // nur zyklische
Ereignisse auflisten
print_r($alleEreignisse);

/* liefert z.B.:
Array
(
[0] => 37659
[1] => 18326
usw. …
)
*/
```

# IPS_IsConditionPassing

Benötigt: IP-Symcon >= 6.1

```
boolean IPS_IsConditionPassing (string $Bedingung)
```

## PARAMETERLISTE

| | |
|---|---|
| Bedingung | Eine Liste von JSON-codierten Bedingungen im Format von SelectCondition |

## RÜCKGABEWERT

TRUE, falls die Bedingung erfüllt ist, andernfalls FALSE

## BESCHREIBUNG

Diese Funktion prüft, ob die JSON-codierte Bedingung **Bedingung** erfüllt ist. Typischerweise wird dies im Kontext eines Moduls mit dem Formularfeld SelectCondition verwendet.

## BEISPIEL

```
if (IPS_IsConditionPassing($Bedingung)) {
// Tue etwas nur falls die Bedingung erfüllt ist
};
```

# IPS_SetEventAction

Benötigt: IP-Symcon >= 6.0

```
boolean IPS_SetEventAction (int $EreignisID, string $AktionsID,
array $AktionsParameter)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu verändernden Ereignisses |
| AktionsID | ID der auszuführenden Aktion |
| AktionsParameter | Liste der Parameter der auszuführenden Aktion |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Aktion des Ereignisses mit der ID **EreignisID** wird auf die Aktion mit der ID **AktionsID** gesetzt. Die Parameter dieser Aktion werden als **AktionsParameter** definiert.



Die Spezialparameter TARGET, ENVIRONMENT und PARENT müssen hier nicht gesetzt werden, da sich diese durch das Ereignis ergeben

## MIGRATION VON IP-SYMCON 5.5 ODER ÄLTER

Soll das Ereignis, wie standardmäßig in IP-Symcon 5.5 oder älter, das übergeordnete Skript starten, so muss ab IP-Symcon 6.0 diese Verknüpfung entsprechend mitgeteilt werden. Dazu ist folgende Zeile notwendig, welche die "Führe Automation aus" Aktion dem Ereignis zuordnet.
IPS_SetEventAction($EreignisID, "{7938A5A2-0981-5FE0-BE6C-8AA610D654EB}", []);

## BEISPIEL

```
// Multipliziere Zielvariable des Ereignisses mit 3
IPS_SetEventAction($EreignisID, "{A3153696-013A-41B1-
A001-5E8085D95465}", ["FACTOR" => 3]);
```

```
// Führe Zielautomation des Ereignisses aus
IPS_SetEventAction($EreignisID, "{7938A5A2-0981-5FE0-
BE6C-8AA610D654EB}", []);
```

# IPS_SetEventActive

```
boolean IPS_SetEventActive (int $EreignisID, boolean $Aktiv)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu verändernden Ereignisses |
| Aktiv | Gibt an, ob das Ereignis aktiviert (TRUE) / deaktiviert (FALSE) werden soll. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Nach dem Aktiveren/Deaktiveren kann über IPS_GetEvent der neue Status ausgelesen werden. Bei aktiven, zyklischen Ereignissen ist es möglich, den nächsten Ausführungszeitpunkt zu bestimmen. Bei einem deaktivierten Ereignis ist der NextRun Wert gleich null.

## BEISPIEL

```
IPS_SetEventActive($EreignisID, true); // Aktivert Ereignis
```

# IPS_SetEventCondition

Benötigt: IP-Symcon >= 4.4

```
boolean IPS_SetEventCondition (int $EreignisID, int $BedingungsID,
int $ÜbergeordneteID, int $Operation)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu verändernden Ereignisses |
| BedingungsID | Eindeutige ID für diese Bedingung. Die Wurzelbedingung hat die ID 0. IDs müssen nur für dieses Ereignis eindeutig sein. |
| ÜbergeordneteID | ID der übergeordneten Bedingung. Bei der Erstellung der Wurzelbedingung muss dieser Parameter 0 sein. |
| Operation | |

| WERT | BESCHREIBUNG |
|---|---|
| 0 | AND |
| 1 | OR |
| 2 | NAND (noch nicht implementiert) |
| 3 | NOR (noch nicht implementiert) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion modifiziert die Bedingung mit der ID **BedingungsID** des Ereignisses mit der ID **EreignisID**. Die Bedingung wird der Bedingung mit der ID **ÜbergeordneteID** untergeordnet und verwendet die Operation **Operation**. Wenn keine Bedingung mit der ID **BedingungsID** existiert, wird eine erstellt.
Falls **ÜbergeordneteID** negativ ist, wird die Bedingung mit der ID **BedingungsID** gelöscht. Eine Bedingung kann nicht gelöscht werden falls sie untergeordnete Bedingungen hat. Wird eine Bedingung gelöscht, werden alle Regeln der Bedingung ebenfalls gelöscht.



Eine Bedingung wird ausgewertet, indem ihre Operation auf die Menge all ihrer Regeln und untergeordneten Bedingungen angewandt wird. Auf diese Weise ist es möglich verschachtelte Bedingungen zu erstellen.

## BEISPIEL

```
// Erstelle die Wurzelbedingung mit der Operation OR für das
Ereignis mit der ID 12345
IPS_SetEventCondition(12345, 0, 0, 1);

// Erstelle eine untergeordnete Bedingung mit der Operation AND
IPS_SetEventCondition(12345, 1, 0, 0);

// Lösche untergeordnete Bedingung
IPS_SetEventCondition(12345, 1, -1, 0);
```

# IPS_SetEventConditionDateRule

Benötigt: IP-Symcon >= 4.4

```
boolean IPS_SetEventConditionDateRule (int $EreignisID, int
$BedingungsID, int $RegelID, int $Vergleich, int $Tag, int $Monat,
int $Jahr)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu verändernden Ereignisses |
| BedingungsID | ID der Bedingung, welche die Regel beinhaltet |
| RegelID | Eindeutige ID für diese Regel. IDs müssen nur für dieses Bedingung eindeutig sein. |
| Vergleich | <table><tr><th>WERT</th><th>BESCHREIBUNG</th></tr><tr><td>0</td><td>==</td></tr><tr><td>1</td><td>!=</td></tr><tr><td>2</td><td>></td></tr><tr><td>3</td><td>>=</td></tr><tr><td>4</td><td><</td></tr><tr><td>5</td><td><=</td></tr></table> |
| Tag | Tag mit dem das aktuelle Datum verglichen wird |
| Monat | Monat mit dem das aktuelle Datum verglichen wird |
| Jahr | Jahr mit dem das aktuelle Datum verglichen wird |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion modifiziert die Regel mit der ID **RegelID** der Bedingung mit der ID **BedingungsID** des Ereignisses mit der ID **EreignisID**. Die Regel vergleicht das aktuelle Datum mit den **Tag**, **Monat** und **Jahr**, entsprechend der Vergleichsoperation **Vergleich**.
Falls keine Regel mit der ID **RegelID** existiert, wird eine erstellt.
Ist einer der Werte **Tag**, **Monat** oder **Jahr** < 0 , wird die Regel mit der ID **RegelID** gelöscht.



Tag, Monat und Jahr kann auf 0 gesetzt werden und so als Wildcard genutzt werden



Welche RegelID welche Regel beinhaltet kann via IPS_GetEvent nachgesehen werden.

## BEISPIEL

```
//Eine AND Bedingung hinzufügen
IPS_SetEventCondition(12345, 0, 0, 0)

// Fügt hinzu/Modifiziert die Regel mit der ID 2 beim Ereignis mit
der ObjektID 12345
// Es ist die erste Jahreshälfte -> aktuelles Datum < 01.07.2018
IPS_SetEventConditionDateRule(12345, 0, 2, 4, 1, 7, 2018);

// Lösche die Regel mit der ID 1
IPS_SetEventConditionDateRule(12345, 0, 1, 0, -1, 0, 0);
```

# IPS_SetEventConditionDayOfTheWeekRule

Benötigt: IP-Symcon >= 5.2

```
boolean IPS_SetEventConditionDayOfTheWeekRule (int $EreignisID, int $BedingungsID, int $RegelID, int $Vergleich, int $Wochentag)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu verändernden Ereignisses |
| BedingungsID | ID der Bedingung, welche die Regel beinhaltet |
| RegelID | Eindeutige ID für diese Regel. IDs müssen nur für dieses Bedingung eindeutig sein. |
| Vergleich | |

| WERT | BESCHREIBUNG |
|---|---|
| 0 | == |
| 1 | != |
| 2 | > |
| 3 | >= |
| 4 | < |
| 5 | <= |

Wochentag

| WERT | BESCHREIBUNG |
|---|---|
| 0 | Löschen |
| 1 | Montag |
| 2 | Dienstag |
| 3 | Mittwoch |
| 4 | Donnerstag |
| 5 | Freitag |
| 6 | Samstag |
| 7 | Sonntag |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion modifiziert die Regel mit der ID **RegelID** der Bedingung mit der ID

**BedingungsID** des Ereignisses mit der ID **EreignisID**. Die Regel vergleicht den aktuellen Wochentag mit dem **Wochentag**, entsprechend der Vergleichoperation **Vergleich**. Falls keine Regel mit der ID **RegelID** existiert, wird eine erstellt.
Ist der Wert **Wochentag** gleich 0, wird die Regel mit der ID **RegelID** gelöscht.



Welche RegelID welche Regel beinhaltet kann via IPS_GetEvent nachgesehen werden.

## BEISPIEL

```
//Eine AND Bedingung hinzufügen
IPS_SetEventCondition(12345, 0, 0, 0)

// Fügt hinzu/Modifiziert die Regel mit der ID 2 beim Ereignis mit
der ObjektID 12345
// Es ist größer gleich Donnerstag d.h.: Donnerstag bis Sonntag
inklusive
IPS_SetEventConditionDayOfTheWeekRule(12345, 0, 2, 3, 4);

// Fügt hinzu/Modifiziert die Regel mit der ID 3 beim Ereignis mit
der ObjektID 12345
// Es ist ungleich Sonntag
IPS_SetEventConditionDayOfTheWeekRule(12345, 0, 3, 1, 7);

// Lösche die Regel mit der ID 2
IPS_SetEventConditionDayOfTheWeekRule(12345, 0, 2, 0, 0);
```

# IPS_SetEventConditionTimeRule

Benötigt: IP-Symcon >= 4.4

```
boolean IPS_SetEventConditionTimeRule (int $EreignisID, int
$BedingungsID, int $RegelID, int $Vergleich, int $Stunde, int
$Minute, int $Sekunde)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu verändernden Ereignisses |
| BedingungsID | ID der Bedingung, welche die Regel beinhaltet |
| RegelID | Eindeutige ID für diese Regel. IDs müssen nur für dieses Bedingung eindeutig sein. |
| Vergleich | <table><tr><th>WERT</th><th>BESCHREIBUNG</th></tr><tr><td>0</td><td>==</td></tr><tr><td>1</td><td>!=</td></tr><tr><td>2</td><td>></td></tr><tr><td>3</td><td>>=</td></tr><tr><td>4</td><td><</td></tr><tr><td>5</td><td><=</td></tr></table> |
| Stunde | Stundenwert (0..23) mit dem die aktuelle Zeit verglichen wird |
| Minute | Minutenwert (0..59) mit dem die aktuelle Zeit verglichen wird |
| Sekunde | Sekundenwert (0..59) mit dem die aktuelle Zeit verglichen wird |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion modifiziert die Regel mit der ID **RegelID** der Bedingung mit der ID **BedingungsID** des Ereignisses mit der ID **EreignisID**. Die Regel vergleicht die aktuelle Uhrzeit mit den **Stunden**, **Minuten** und **Sekunden**, entsprechend der Vergleichsoperation **Vergleich**. Falls keine Regel mit der ID **RegelID** existiert, wird eine erstellt.
Ist einer der Werte **Stunde**, **Minute** oder **Sekunde** < 0 , wird die Regel mit der ID **RegelID** gelöscht.

Welche RegelID welche Regel beinhaltet kann via IPS_GetEvent nachgesehen werden.

## BEISPIEL

```
// Erstelle eine Wurzelbedingung
IPS_SetEventCondition(12345, 0, 0, 0);

// Fügt hinzu/Modifiziert die Regel mit der ID 2 beim Ereignis mit
der ObjektID 12345
// Es muss genau 22:30 sein damit diese Regel erfüllt ist.
IPS_SetEventConditionTimeRule(12345, 0, 2, 1, 22, 30, 0);

// Lösche die Regel mit der ID 1
IPS_SetEventConditionTimeRule(12345, 0, 1, 0, -1, 0, 0);
```

# IPS_SetEventConditionVariableDynamicRule

Benötigt: IP-Symcon >= 6.1

```
boolean IPS_SetEventConditionVariableDynamicRule (int $EreignisID,
int $BedingungsID, int $RegelID, int $VariablenID, int $Vergleich,
int $VergleichVariablenID)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu verändernden Ereignisses |
| BedingungsID | ID der Bedingung welche die Regel beinhaltet |
| RegelID | Eindeutige ID für diese Regel. IDs müssen nur für dieses Bedingung eindeutig sein. |
| VariablenID | ID der geprüften Variable |
| Vergleich | <table><tr><th>WERT</th><th>BESCHREIBUNG</th></tr><tr><td>0</td><td>==</td></tr><tr><td>1</td><td>!=</td></tr><tr><td>2</td><td>></td></tr><tr><td>3</td><td>>=</td></tr><tr><td>4</td><td><</td></tr><tr><td>5</td><td><=</td></tr></table> |
| VergleichVariablenID | ID der Variablen, mit der der Wert verglichen werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion modifiziert die Regel mit der ID **RegelID** der Bedingung mit der ID **BedingungsID** des Ereignisses mit der ID **EreignisID**. Die Regel vergleicht die Variable mit der ID **VariablenID** mit dem Wert der Variable mit der ID **VergleichVariableID** entsprechend der Vergleichsoperation **Vergleich**. Falls keine Regel mit der ID **RegelID** existiert, wird eine erstellt.
Ist **VariablenID** 0, wird die Regel mit der ID **RegelID** gelöscht.



Für die Variablentypen Boolean und String stehen nur die ersten beiden Vergleichsoperatoren zur Verfügung



Der Wert in **Wert** muss dem Typ der Variablen mit der ID **VariablenID** entsprechen.

## BEISPIEL

```
// Erstelle eine Wurzelbedingung für das Ereignis mit ID 12345
IPS_SetEventCondition(12345, 0, 0, 0);

// Füge eine Regel mit der ID 2 hinzu: Variable 23456 < Variable
34567
IPS_SetEventConditionVariableDynamicRule(12345, 0, 2, 23456, 4,
34567);

// Lösche die Regel mit der ID 2
IPS_SetEventConditionVariableDynamicRule(12345, 0, 2, 0, 0, 0);
```

# IPS_SetEventConditionVariableRule

Benötigt: IP-Symcon >= 4.4

```
boolean IPS_SetEventConditionVariableRule (int $EreignisID, int
$BedingungsID, int $RegelID, int $VariablenID, int $Vergleich,
variant $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu verändernden Ereignisses |
| BedingungsID | ID der Bedingung welche die Regel beinhaltet |
| RegelID | Eindeutige ID für diese Regel. IDs müssen nur für dieses Bedingung eindeutig sein. |
| VariablenID | ID der geprüften Variable |
| Vergleich | <table><tr><th>WERT</th><th>BESCHREIBUNG</th></tr><tr><td>0</td><td>==</td></tr><tr><td>1</td><td>!=</td></tr><tr><td>2</td><td>></td></tr><tr><td>3</td><td>>=</td></tr><tr><td>4</td><td><</td></tr><tr><td>5</td><td><=</td></tr></table> |
| Wert | Wert mit dem die geprüfte Variable verglichen wird |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG



Diese Funktion ist veraltet. Bitte verwenden Sie die neuere IPS_SetEventConditionVariableStaticRule Funktion, welche die selben Parameter akzeptiert und einen verständlicheren Namen besitzt

Diese Funktion modifiziert die Regel mit der ID **RegelID** der Bedingung mit der ID **BedingungsID** des Ereignisses mit der ID **EreignisID**. Die Regel vergleicht die Variable mit der ID **VariablenID** mit dem Wert **Wert** entsprechend der Vergleichsoperation **Vergleich**.

Falls keine Regel mit der ID **RegelID** existiert, wird eine erstellt.
Ist **VariablenID** 0, wird die Regel mit der ID **RegelID** gelöscht.



Für die Variablentypen Boolean und String stehen nur die ersten beiden Vergleichsoperatoren zur Verfügung



Der Wert in **Wert** muss dem Typ der Variablen mit der ID **VariablenID** entsprechen.

## BEISPIEL

```
// Erstelle eine Wurzelbedingung für das Ereignis mit ID 12345
IPS_SetEventCondition(12345, 0, 0, 0);

// Füge eine Regel mit der ID 2 hinzu: Variable 23456 < 75
IPS_SetEventConditionVariableRule(12345, 0, 2, 23456, 4, 75);

// Lösche die Regel mit der ID 2
IPS_SetEventConditionVariableRule(12345, 0, 2, 0, 0, 0);
```

# IPS_SetEventConditionVariableStaticRule

Benötigt: IP-Symcon >= 6.1

```
boolean IPS_SetEventConditionVariableStaticRule (int $EreignisID,
int $BedingungsID, int $RegelID, int $VariablenID, int $Vergleich,
variant $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu verändernden Ereignisses |
| BedingungsID | ID der Bedingung welche die Regel beinhaltet |
| RegelID | Eindeutige ID für diese Regel. IDs müssen nur für dieses Bedingung eindeutig sein. |
| VariablenID | ID der geprüften Variable |
| Vergleich | <table><tr><th>WERT</th><th>BESCHREIBUNG</th></tr><tr><td>0</td><td>==</td></tr><tr><td>1</td><td>!=</td></tr><tr><td>2</td><td>></td></tr><tr><td>3</td><td>>=</td></tr><tr><td>4</td><td><</td></tr><tr><td>5</td><td><=</td></tr></table> |
| Wert | Wert mit dem die geprüfte Variable verglichen wird |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion modifiziert die Regel mit der ID **RegelID** der Bedingung mit der ID **BedingungsID** des Ereignisses mit der ID **EreignisID**. Die Regel vergleicht die Variable mit der ID **VariablenID** mit dem Wert **Wert** entsprechend der Vergleichsoperation **Vergleich**.
Falls keine Regel mit der ID **RegelID** existiert, wird eine erstellt.
Ist **VariablenID** 0, wird die Regel mit der ID **RegelID** gelöscht.



Für die Variablentypen Boolean und String stehen nur die ersten beiden Vergleichsoperatoren zur Verfügung



Der Wert in **Wert** muss dem Typ der Variablen mit der ID **VariablenID** entsprechen.

## BEISPIEL

```
// Erstelle eine Wurzelbedingung für das Ereignis mit ID 12345
IPS_SetEventCondition(12345, 0, 0, 0);

// Füge eine Regel mit der ID 2 hinzu: Variable 23456 < 75
IPS_SetEventConditionVariableStaticRule(12345, 0, 2, 23456, 4, 75);

// Lösche die Regel mit der ID 2
IPS_SetEventConditionVariableStaticRule(12345, 0, 2, 0, 0, 0);
```

# IPS_SetEventCyclic

```
boolean IPS_SetEventCyclic (int $EreignisID, int $Datumstyp, int
$Datumsintervall, int $Datumstage, int $Datumstagesintervall, int
$Zeittyp, int $Zeitintervall)
```

## PARAMETERLISTE

EreignisID

ID des zu verändernden Ereignisses

Datumstyp

| WERT | BESCHREIBUNG |
|---|---|
| 0 | Kein Datumstyp. Tägliche Ausführung. |
| 1 | Einmalig. IPS_SetEventCyclicDateFrom für Zieldatum |
| 2 | Täglich |
| 3 | Wöchentlich |
| 4 | Monatlich |
| 5 | Jährlich |

Datumsintervall

| WERT | BESCHREIBUNG |
|---|---|
| 0 | 0 (Keine Auswertung) |
| 1 | 0 (Keine Auswertung) |
| 2 | Alle **X** Tage |
| 3 | Alle **X** Wochen |
| 4 | Alle **X** Monate |
| 5 | 1 = Januar … 12 = Dezember |

Datumstage

| WERT | BESCHREIBUNG |
|---|---|
| 0 | 0 (Keine Auswertung) |
| 1 | 0 (Keine Auswertung) |
| 2 | 0 (Keine Auswertung) |
| 3 | siehe Tabelle: Tageswerte (gewünschte Tageswerte müssen addiert werden) |
| 4 | siehe Tabelle: Tageswerte oder 0 für jeden **X** Tag im Monat |
| 5 | 0 (Keine Auswertung) |

*Tabelle: Tageswerte*

| TAG | WERT |
|---|---|
| Montag | 1 |
| Dienstag | 2 |
| Mittwoch | 4 |
| Donnerstag | 8 |
| Freitag | 16 |
| Samstag | 32 |
| Sonntag | 64 |

Datumstagesintervall

| WERT | BESCHREIBUNG |
|---|---|
| 0 | 0 (Keine Auswertung) |
| 1 | 0 (Keine Auswertung) |
| 2 | 0 (Keine Auswertung) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Um ein zyklisches Ereignis zu konfigurieren, muss festgelegt werden, in welchen Intervallen dieses ausgeführt werden soll. Dazu gehört ein **Datumsintervall**, welches das Intervall in einem bestimmten Tages-, Wochen-, Monats- oder Jahresrythmus festlegt und ein Zeitintervall, welches an dem jeweiligen Tag ein sekunden-, minuten- oder stundenbasierendes Intervall festlegt.
Wird das **Datumsintervall** als “Woche” gewählt, so besteht die Möglichkeit die Tage der Woche zu bestimmten an denen das Ereignis gestartet werden soll. Bei Monatlichem oder Jährlichem **Datumsintervall** kann der Tag im Monat/Jahr im Parameter **Datumswert** angegeben werden. Bei allen anderen Typen kann der Parameter mit dem Platzhalter 0 angegeben werden.



Spitzfindigkeiten:

(Monatlich) Bei der Variante “Am X. Tag alle Y Monate” bedeutet X > 31, dass der letzte Tag im Monat gewählt wird. Wenn z.B. für X = 31 ausgewählt wird, so werden nur Monate berücksichtigt, die 31 Tage haben.
(Monatlich) Bei der Variante “Am X. Wochentag alle Y Monate” bedeutet X >5, dass die letzte Woche gewählt wird. Wenn z.B. für X = 5 ausgewählt wird, so werden nur Monate berücksichtigt, die einen jeweiligen 5. Wochentag haben.
(Jährlich) Bei der Variante “Am X. Monat” wird ein Timer nur ausgeführt, wenn X ein gültiger Tag in diesem Monat ist, andernfalls wird der Timer nicht ausgeführt.

## BEISPIEL

```
//Zum aktuellen Zeitpunkt alle 8 Wochen etwas ausführen
$eid = IPS_CreateEvent(1);
IPS_SetParent($eid, $_IPS['SELF']);
IPS_SetEventCyclicTimeFrom($eid, (int)date("H"), (int)date("i"),
(int)date("s"));
IPS_SetEventCyclicDateFrom($eid, (int)date("d"), (int)date("m"),
(int)date("Y"));
IPS_SetEventCyclic($eid, 2 /* Täglich */ , 8 * 7 /* Alle 8 Wochen
*/,0,0,0,0);
IPS_SetEventActive($eid, true);

//Jeden Tag alle 6 Stunden ab 0 Uhr
$eid = IPS_CreateEvent(1);
IPS_SetParent($eid, $_IPS['SELF']);
IPS_SetEventCyclic($eid, 2 /* Täglich */, 1 /* Jeden Tag */, 0, 0, 3
/* Stündlich */, 6 /* Alle 6 Stunden */);

//Alle 2 Minuten ab 0 Uhr bis 12 Uhr
$eid = IPS_CreateEvent(1);
IPS_SetParent($eid, $_IPS['SELF']);
IPS_SetEventCyclic($eid, 0 /* Keine Datumsüberprüfung */, 0, 0, 2, 2
/* Minütlich */ , 2 /* Alle 2 Minuten */);
IPS_SetEventCyclicTimeTo($eid, 12, 0, 0);

//Alle 2 Minuten zwischen 8 Uhr und 10 Uhr
$eid = IPS_CreateEvent(1);
IPS_SetParent($eid, $_IPS['SELF']);
IPS_SetEventCyclic($eid, 0 /* Keine Datumsüberprüfung */, 0, 0, 2, 2
/* Minütlich */ , 2 /* Alle 2 Minuten */);
IPS_SetEventCyclicTimeFrom($eid, 8, 0, 0);
IPS_SetEventCyclicTimeTo($eid, 10, 0, 0);

//Alle 2 Wochen am Montag+Mittwoch einmalig um genau 0:00 Uhr
$eid = IPS_CreateEvent(1);
IPS_SetParent($eid, $_IPS['SELF']);
IPS_SetEventCyclic($eid, 3 /* Wöchentlich */, 2 /* Alle 2 Wochen */,
1+4 /* Montag (1) + Dienstag (4) */, 0, 0, 0);

//Täglich um 15 Uhr
$eid = IPS_CreateEvent(1);
IPS_SetParent($eid, $_IPS['SELF']);
IPS_SetEventCyclicTimeFrom($eid, 15, 0, 0);

//Am ersten Tag im Monat um 0 Uhr
$eid = IPS_CreateEvent(1);
IPS_SetParent($eid, $_IPS['SELF']);
IPS_SetEventCyclic($eid, 4 /* Monatlich */, 1 /* Jeden Monat */, 0 /
* Egal welcher Tag */, 1 /* Erster Tag im Monat */, 0, 0);

//Am ersten Freitag im Monat um 0 Uhr
$eid = IPS_CreateEvent(1);
IPS_SetParent($eid, $_IPS['SELF']);
IPS_SetEventCyclic($eid, 4 /* Monatlich */, 1 /* Jeden Monat */, 16
/* Am Freitag */, 1 /* Erster Freitag im Monat */, 0, 0);
```

# IPS_SetEventCyclicDateBounds

```
boolean IPS_SetEventCyclicDateBounds (int $EreignisID, int
$Startdatum, int $Enddatum)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu verändernden Ereignisses |
| Startdatum | Datum als Unix Timestamp |
| Enddatum | Datum als Unix Timestamp |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG



Ab IP-Symcon 3.1 ist diese Funktion veraltet und wird nur noch zur Abwärtskompatibilität emuliert. Es ist empfohlen die neuen Funktionen IPS_SetEventCyclicDateFrom und IPS_SetEventCyclicDateTo zu nutzen.

Die Funktion definiert das Start-/Enddatum für ein Ereignis.
Das Datum muss als Unix Timestamp übergeben werden. (Wikipedia:Unixzeit)
Wird in einem Parameter 0 angegeben, so gibt es kein Start- und/oder Enddatum.

## BEISPIEL

```
//Kein Start/Enddatum dem Ereignis zuordnen
IPS_SetEventCyclicDateBounds($EreignisID, 0, 0);
```

# IPS_SetEventCyclicDateFrom

Benötigt: IP-Symcon >= 3.1

```
boolean IPS_SetEventCyclicDateFrom (int $EreignisID, int $Tag, int
$Monat, int $Jahr)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu verändernden Ereignisses |
| Tag | Start-Tag (0, 1..31) |
| Monat | Start-Monat (0, 1..12) |
| Jahr | Start-Jahr (0, 1970..2038) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Die Funktion definiert das Startdatum für ein Ereignis.
Wird für Tag, Monat und Jahr jeweils 0 angegeben, so gibt es kein spezifisches Startdatum und es wird nicht berücksichtigt.



Das Start-Datum ist besonders relevant, wenn es ein Ereignis gibt, welches nicht Täglich ausgeführt wird. Bei z.B. einer Ausführung alle 2 Wochen am Montag kann über das Start-Datum der Referenzpunkt bestimmt werden, an dem das erste Ereignis auftritt. Dadurch kann gesteuert werden, ob das Ereignis in einer geraden oder ungeraden Kalenderwoche ausgeführt wird.

## BEISPIEL

```
//Kein Startdatum dem Ereignis zuordnen
IPS_SetEventCyclicDateFrom($EreignisID, 0, 0, 0);
```

# IPS_SetEventCyclicDateTo

Benötigt: IP-Symcon >= 3.1

```
boolean IPS_SetEventCyclicDateTo (int $EreignisID, int $Tag, int
$Monat, int $Jahr)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu verändernden Ereignisses |
| Tag | End-Tag (0, 1..31) |
| Monat | End-Monat (0, 1..12) |
| Jahr | End-Jahr (0, 1970..2038) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Die Funktion definiert das Enddatum für ein Ereignis.
Wird für Tag, Monat und Jahr jeweils 0 angegeben, so gibt es kein spezifisches Enddatum und es wird nicht berücksichtigt.



Das Ereignis wird inklusive des Enddatums ausgeführt

## BEISPIEL

```
//Den 1.2.2020 als Enddatum dem Ereignis zuordnen
IPS_SetEventCyclicDateTo($EreignisID, 1, 2, 2020);
```

# IPS_SetEventCyclicTimeBounds

```
boolean IPS_SetEventCyclicTimeBounds (int $EreignisID, int
$Startuhrzeit, int $Enduhrzeit)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu verändernden Ereignisses |
| Startuhrzeit | Zeit als Unix Timestamp |
| Enduhrzeit | Zeit als Unix Timestamp |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG



Ab IP-Symcon 3.1 ist diese Funktion veraltet und wird nur noch zur Abwärtskompatibilität emuliert. Es ist empfohlen die neuen Funktionen IPS_SetEventCyclicTimeFrom und IPS_SetEventCyclicTimeTo zu nutzen.

Die Funktion definiert die Start- und Enduhrzeit für ein Ereignis. Desweiteren kann dadurch definiert werden, zu welchem Sekundenanteil das Skript gestartet werden soll (z.B. 00:00:30, wenn zu jeder halben Minute gestartet werden soll).
Die Zeit muss als Unix Timestamp übergeben werden. (Wikipedia:Unixzeit)
Wird in einem Parameter 0 angegeben, so gibt es keine Start- und/oder Endzeit.

## BEISPIEL

```
//Keine Start/Endzeiten dem Ereignis zuordnen
IPS_SetEventCyclicTimeBounds($EreignisID, 0, 0);
//Ereignis um 7:30 starten
IPS_SetEventCyclicTimeBounds($EreginisID, mktime(7, 30, 0), 0);
```

# IPS_SetEventCyclicTimeFrom

Benötigt: IP-Symcon >= 3.1

```
boolean IPS_SetEventCyclicTimeFrom (int $EreignisID, int $Stunde,
int $Minute, int $Sekunde)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu verändernden Ereignisses |
| Stunde | Start-Stunde (0..23) |
| Minute | Start-Minute (0..59) |
| Sekunde | Start-Sekunde (0..59) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Die Funktion definiert die Startzeit für ein Ereignis. Die Startzeit definiert den ersten Zeitpunkt der Ausführung.



Die Start-Zeit ist besonders relevant, wenn ein Ereignis leicht versetzt gestartet werden soll, um z.B. nicht alle Ereignisse gleichzeitig zur vollen Minute zu starten. Wenn als Startzeit 00:00:05 angegeben wird, so wird das Skript bei einer minütlichen Ausführung immer genau 5 Sekunden nach der vollen Minute ausgeführt.

## BEISPIEL

```
//Als Startzeit 7:15 dem Ereignis zuordnen
IPS_SetEventCyclicTimeFrom($EreignisID, 7, 15, 0);
```

# IPS_SetEventCyclicTimeTo

Benötigt: IP-Symcon >= 3.1

```
boolean IPS_SetEventCyclicTimeTo (int $EreignisID, int $Stunde, int
$Minute, int $Sekunde)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu verändernden Ereignisses |
| Stunde | End-Stunde (0..23) |
| Minute | End-Minute (0..59) |
| Sekunde | End-Sekunde (0..59) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Die Funktion definiert die Endzeit für ein Ereignis. Die Endzeit definiert den letzten Zeitpunkt der Ausführung.
Wird für Stunde, Minute und Sekunde jeweils 0 angegeben, so gibt es keine Endzeit die letzte Ausführung ist um spätestens 23:59:59.

## BEISPIEL

```
//Keine Endzeit dem Ereignis zuordnen
IPS_SetEventCyclicTimeTo($EreignisID, 0, 0, 0);
```

# IPS_SetEventLimit

```
boolean IPS_SetEventLimit (int $EreignisID, int $Anzahl)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu verändernden Ereignisses |
| Anzahl | Die Anzahl der Ausführung, bevor das Ereignis deaktiviert wird. 0 = Keine Limitierung |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion limitiert die Anzahl der Ereignisaufrufe des Ereignisses mit der ID **EreignisID**. Die Limitierung auf **Anzahl** bewirkt, dass das Ereignis nur **Anzahl** Male ausgeführt wird. Danach ist die Anzahl gleich null und das Ereignis ist deaktiviert. Die Einstellung Anzahl = 0 hebt die Limitierung auf.

## BEISPIEL

```
//Keine Limitierung
IPS_SetEventLimit($EreignisID, 0);
```

# IPS_SetEventScheduleAction

Benötigt: IP-Symcon >= 3.2

```
boolean IPS_SetEventScheduleAction (int $EreignisID, int
$WochenplanAktionsID, string $Name, int $Farbe, string
$Skriptinhalt)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu verändernden Ereignisses |
| WochenplanAktionsID | Eindeutige ID für diese Wochenplanaktion. Die Sortierung wird anhand der ID vorgenommen. IDs müssen nur für dieses Ereignis eindeutig sein. |
| Name | Name zur angegebenen Wochenplanaktion |
| Farbe | Farbwert im HTML Farbcode (z.b. 0x0000FF für Blau) |
| Skriptinhalt | PHP Skript ohne PHP Tags (<? … ?>) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion setzt den **Namen**, die **Farbe** und auszuführende Aktion auf "Führe PHP Code aus" mit dem Code **Skriptinhalt** für die Wochenplanaktion mit der ID **WochenplanAktionsID** eines Wochenplan Ereignisses mit der ID **EreignisID**. Die Systemvariable $_IPS['ACTION'] wird definiert. Es stehen für die Skriptinhalte die Systemvariablen zur Verfügung.
Wenn der **Name** leer ist, wird die Wochenplanaktion mit der angegebenen **WochenplanAktionsID** gelöscht. Dies ist jedoch nur möglich, wenn kein Schaltpunkt mehr auf diese **WochenplanAktionsID** verweist.



Der Wert –2147483648 ist keine gültige WochenplanAktionsID.



Es können maximal 32 Aktionen erstellt werden.

## BEISPIEL

```
//Anlegen von Aktionen
IPS_SetEventScheduleAction($EreignisID, 0, "Warm", 0xFF0000,
"FHT_SetTemperature(\$_IPS['TARGET'], 22.5);");
IPS_SetEventScheduleAction($EreignisID, 1, "Kalt", 0x0000FF,
"FHT_SetTemperature(\$_IPS['TARGET'], 17);");
IPS_SetEventScheduleAction($EreignisID, 2, "Öko", 0x00FF00,
"FHT_SetTemperature(\$_IPS['TARGET'], 20);");

//Aktion mit ID 2 wieder löschen
IPS_SetEventScheduleAction($EreignisID, 2, "", 0, "");
```

# IPS_SetEventScheduleActionEx

Benötigt: IP-Symcon >= 6.0

```
boolean IPS_SetEventScheduleActionEx (int $EreignisID, int
$WochenplanAktionsID, string $Name, int $Farbe, string $AktionsID,
array $AktionsParameter)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu verändernden Ereignisses |
| WochenplanAktionsID | Eindeutige ID für diese Wochenplanaktion. Die Sortierung wird anhand der ID vorgenommen. IDs müssen nur für dieses Ereignis eindeutig sein. |
| Name | Name zur angegebenen Wochenplanaktion |
| Farbe | Farbwert im HTML Farbcode (z.b. 0x0000FF für Blau) |
| AktionsID | ID der auszuführenden Aktion |
| AktionsParameter | Liste der Parameter der auszuführenden Aktion |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion setzt den **Namen**, die **Farbe** und auszuführende Aktion mit der ID **AktionsID** mit Parametern **AktionsParameter** für die Wochenplanaktion mit der ID **WochenplanAktionsID** eines Wochenplan Ereignisses mit der ID **EreignisID**. Falls über die Aktion ein PHP-Skript ausgeführt wird, ist die Systemvariable $_IPS['ACTION'] definiert. Es stehen für die Skriptinhalte die Systemvariablen zur Verfügung.
Wenn der **Name** leer ist, wird die Wochenplanaktion mit der angegebenen **WochenplanAktionsID** gelöscht. Dies ist jedoch nur Möglich, wenn kein Schaltpunkt mehr auf diese **WochenplanAktionsID** verweist.



Die Spezialparameter TARGET, ENVIRONMENT und PARENT müssen hier nicht gesetzt werden, da sich diese durch das Ereignis ergeben



Der Wert –2147483648 ist keine gültige WochenplanAktionsID.



Es können maximal 32 Wochenplanaktionen erstellt werden.

## BEISPIEL

```
//Anlegen von Wochenplanaktionen
IPS_SetEventScheduleActionEx($EreignisID, 0, "Warm", 0xFF0000,
"{3644F802-C152-464A-868A-242C2A3DEC5C}", ["VALUE" => 22.5]);
IPS_SetEventScheduleActionEx($EreignisID, 1, "Kalt", 0xFF0000,
"{3644F802-C152-464A-868A-242C2A3DEC5C}", ["VALUE" => 17]);
IPS_SetEventScheduleActionEx($EreignisID, 2, "Öko", 0xFF0000,
"{3644F802-C152-464A-868A-242C2A3DEC5C}", ["VALUE" => 20]);

//Wochenplanaktion mit ID 2 wieder löschen
IPS_SetEventScheduleAction($EreignisID, 2, "", 0, "", []);
```

# IPS_SetEventScheduleGroup

Benötigt: IP-Symcon >= 3.2

```
boolean IPS_SetEventScheduleGroup (int $EreignisID, int $GruppenID,
int $Tage)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu verändernden Ereignisses |
| GruppenID | Eindeutige ID für diese Gruppe. Die Sortierung wird anhand der ID vorgenommen. IDs müssen nur für dieses Ereignis eindeutig sein. |
| Tage | Addition der Tageswerte, die zur Gruppe gehören. Die Tageswerte können der Tabelle bei IPS_SetEventCyclic entnommen werden. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion setzt die Bitmaske **Tage** für die **GruppenID** eines Wochenplan Ereignisses mit der ID **EreignisID**. Es können in einer Gruppe beliebige Tage kombiniert werden. Dabei ist zu beachten, dass ein bestimmter Tag nur zu einer Gruppe zugehörig sein kann. In einer Gruppe können auch alle Tage (Wert = 127) oder nur ein einzelner Tag definiert sein. (z.B. 1 für Montag) Wenn für ein Ereignis nicht alle Tage in Gruppen aufgeteilt wurden, so können für diese Tage keine Schaltpunkte definiert werden und der letzte ausgeführte Schaltpunkt bleibt bestehen.
Wenn der Wert von **Tage** Null (0) ist, wird die Gruppe mit der angegebenen **GruppenID** gelöscht. Dabei werden auch alle der Gruppe zugehörigen Schaltpunkte entfernt.

## BEISPIEL

```
//Wochenplan Ereignis erstellen
$EreignisID = IPS_CreateEvent(2);

//Anlegen von Gruppen
IPS_SetEventScheduleGroup($EreignisID, 0, 31); //Mo - Fr (1 + 2 + 4
+ 8 + 16)
IPS_SetEventScheduleGroup($EreignisID, 1, 96); //Sa + So (32 + 64)
```

```
//Gruppe mit ID 1 wieder löschen
IPS_SetEventScheduleGroup($EreignisID, 1, 0);
```

# IPS_SetEventScheduleGroupPoint

Benötigt: IP-Symcon >= 3.2

```
boolean IPS_SetEventScheduleGroupPoint (int $EreignisID, int
$GruppenID, int $SchaltpunktID, int $StartStunde, int $StartMinute,
int $StartSekunde, int $AktionsID)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu verändernden Ereignisses |
| GruppenID | Eindeutige ID für diese Gruppe. Die Sortierung wird anhand der ID vorgenommen. IDs müssen nur für dieses Ereignis eindeutig sein. |
| SchaltpunktID | Eindeutige ID für diesen Schaltpunkt. IDs müssen nur für diese Gruppe eindeutig sein. |
| StartStunde | Start-Stunde (0..23) |
| StartMinute | Start-Minute (0..59) |
| StartSekunde | Start-Sekunde (0..59) |
| AktionsID | ID der Aktion, welche zum Zeitpunkt ausgeführt werden soll. Die ID muss gültig sein. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion setzt den Schaltpunkt mit der Kombination aus **StartStunde**, **StartMinute** und **StartSekunde** für die **SchaltpunktID** der Gruppe mit **GruppenID** eines Wochenplan Ereignisses mit der ID **EreignisID**.
Wenn **StartStunde**, **StartMinute** oder **StartSekunde** außerhalb des gültigen Bereichs ist (z.B. -1), wird der Schaltpunkt mit der angegebenen **SchaltpunktID** gelöscht.



Es darf maximal ein Startpunkt die selbe Kombination aus **Stunde**, **Minute** und **Sekunde** besitzen

## BEISPIEL

```
//Anlegen von Schaltpunkten für Gruppe mit ID = 0 (=Mo-Fr)
IPS_SetEventScheduleGroupPoint($id, 0, 0, 8, 0, 0, 0); //Um 8:00
Aktion mit ID 0 (=Warm) aufrufen
IPS_SetEventScheduleGroupPoint($id, 0, 1, 22, 30, 0, 2); //Um 22:30
Aktion mit ID 2 (=Öko) aufrufen

//Schaltpunkten mit ID 1 für Gruppe mit ID = 0 (=Mo-Fr) wieder
löschen
IPS_SetEventScheduleGroupPoint($EreignisID, 0, 1, -1, -1, -1, 0);
```

# IPS_SetEventScript

```
boolean IPS_SetEventScript (int $EreignisID, string $Skriptinhalt)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu verändernden Ereignisses |
| Skriptinhalt | PHP Code ohne PHP Tags (<? … ?>) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die auszuführende Aktion des Ereignisses wird auf "Führe PHP Code aus" gesetzt und der Code auf **Skriptinhalt** gesetzt. Der PHP Code wird direkt ohne die PHP Tags (<? … ?>) angegeben.
Die Auflistung der Systemvariablen zeigt, welche speziellen Variablen verfügbar sind.

## BEISPIEL

```
//Skriptinhalt dem Ereignis zuordnen
IPS_SetEventScript($EreignisID, "echo 'Verknüpftes Objekt:' . \
$_IPS['TARGET'];");
```

# IPS_SetEventTrigger

```
boolean IPS_SetEventTrigger (int $EreignisID, int $Auslösertyp, int $AuslöserVariablenID)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu verändernden Ereignisses |
| Auslösertyp | <table><tr><th>WERT</th><th>BESCHREIBUNG</th></tr><tr><td>0</td><td>Bei Variablenaktualisierung</td></tr><tr><td>1</td><td>Bei Variablenänderung</td></tr><tr><td>2</td><td>Bei Grenzüberschreitung. Grenzwert wird über IPS_SetEventTriggerValue festgelegt</td></tr><tr><td>3</td><td>Bei Grenzunterschreitung. Grenzwert wird über IPS_SetEventTriggerValue festgelegt</td></tr><tr><td>4</td><td>Bei bestimmtem Wert. Wert wird über IPS_SetEventTriggerValue festgelegt</td></tr></table> |
| AuslöserVariablenID | VariablenID, auf dessen Änderung/Aktualisierung reagiert werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion konfiguriert ein auslösendes Ereignis, auf welchen **Auslösertyp** das Ereignis reagieren und welche **Variable** das Ereignis beobachten soll.

## BEISPIEL

```
IPS_SetEventTrigger($eid, 0, 12345); //OnUpdate für Variable 12345
```

# IPS_SetEventTriggerSubsequentExecution

```
boolean IPS_SetEventTriggerSubsequentExecution (string $EreignisID,
boolean $NachfolgendeAusführungen)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu verändernden Ereignisses |
| NachfolgendeAusführungen | TRUE für Erlaubt, sonst FALSE |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Die Funktion gibt an, ob das Ereignis erneut ausgeführt werden soll, wenn eine Grenzüberschreitung/-unterschreitung zum zweiten Mal ausgelöst wird. Andernfalls wird das Ereignis erst wieder ausgeführt, wenn der Variablenwert innerhalb des Grenzwertbereichs gewesen ist und erneut überschritten wird.

## BEISPIEL

```
IPS_SetEventTriggerSubsequentExecution($EreignisID, true); //
Erlauben
```

# IPS_SetEventTriggerValue

```
boolean IPS_SetEventTriggerValue (int $EreignisID, variant
$Grenzwert)
```

## PARAMETERLISTE

| | |
|---|---|
| EreignisID | ID des zu verändernden Ereignisses |
| Grenzwert | Abhänger Wert/Typ je nach VariablenID vom Auslöser |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Die Funktion konfiguriert einen Wert, welcher beim Vergleich genutzt werden soll.
Der Auslösertyp des Ereignisses kann wie folgt eingestellt sein:
Bei Grenzüberschreitung
Bei Grenzunterschreitung
Bei bestimmem Wert



Der Grenzwert muss als korrekter Typ angegeben werden. Andernfalls kann der Vergleich fehlschlagen oder unerwartet ausgelöst werden.

## BEISPIEL

```
IPS_SetEventTriggerValue($EreignisID, true); //Nur auf TRUE Werte
auslösen
```

# Instanzenverwaltung

## DEBUG

IPS_EnableDebugFile aktiviert das Schreiben von Debug-Logs in eine Datei

## IPS_CREATEINSTANCE

legt eine Instanz an

## IPS_DELETEINSTANCE

löscht eine Instanz

## IPS_GETINSTANCE

liefert umfassende Informationen zu einer bestimmten Instanz

## IPS_GETINSTANCEIDBYNAME

liefert die ID einer Instanz anhand des Namens

## IPS_GETINSTANCELIST

liefert eine Aufstellung aller existierenden Instanzen

## IPS_GETINSTANCELISTBYMODULEID

liefert alle Instanzen die von einer bestimmten ModulID sind

## IPS_GETINSTANCELISTBYMODULETYPE

liefert eine Liste aller Instanzen eines bestimmten Typs

## IPS_INSTANCEEXISTS

prüft, ob eine bestimmte Instanz existiert

## KONFIGURATION

| | |
|---|---|
| IPS_ApplyChanges | übernimmt eine veränderte Konfiguration |
| IPS_GetConfiguration | liest eine Konfiguration aus |
| IPS_GetConfigurationForm | liest eine Konfigurationseite aus |
| IPS_GetProperty | liest den aktuellen Wert einer Eigenschaft |
| IPS_HasChanges | prüft, ob eine Konfiguration verändert wurde |
| IPS_ResetChanges | setzt eine veränderte Konfiguration auf die aktuelle zurück |
| IPS_SetConfiguration | setzt eine neue Konfiguration |
| IPS_SetProperty | setzt einen neuen Wert für eine Eigenschaft |

## REFERENZEN

| | |
|---|---|
| IPS_GetReferenceList | gibt die IDs aller Referenzen wieder |

## STATUSVARIABLEN

## VERBINDUNGEN

| | |
|---|---|
| IPS_ConnectInstance | erstellt eine (Daten-)Verbindung zwischen zwei Instanzen |
| IPS_DisconnectInstance | löst eine (Daten-)Verbindung zwischen zwei Instanzen |
| IPS_GetCompatibleInstances | liefert kompatible Instanzen zu einer Instanz |
| IPS_IsInstanceCompatible | prüft, ob eine Verbindung zwischen zwei Instanzen möglich ist |

# Debug

| | |
|---|---|
| IPS_EnableDebugFile | aktiviert das Schreiben von Debug-Logs in eine Datei |

# IPS_EnableDebugFile

Benötigt: IP-Symcon >= 5.2

```
boolean IPS_EnableDebugFile (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion aktiviert das Schreiben von Debug-Logs in eine Datei von der Instanz mit der ID **InstanzID**.
Die Debugdatei befindet sich im Logs Ordner (z.B. /var/log/symcon/debug_<InstanzID>.log).



Diese Option ist flüchtig. Nach einem Neustart des IP-Symcon Diensts ist das Schreiben der Debug-Logs automatisch wieder deaktiviert.

## BEISPIEL

```
// Aktiviert das Schreiben von Debug-Logs der Instanz 12345
IPS_EnableDebugFile(12345);

// Dateiname
debug_12345.log
```

# IPS_CreateInstance

```
int IPS_CreateInstance (string $ModulID)
```

## PARAMETERLISTE

| | |
|---|---|
| ModulID | ModulID ( GUID ) des zu erstellenden Objekts |

## RÜCKGABEWERT

ID der neu erstellten Instanz

## BESCHREIBUNG

Die Funktion legt eine unkonfigurierte Instanz der ID **ModulID** an. Zulässige Werte für **ModulID** können über die Funktion IPS_GetModuleList ermittelt werden. Die ModulID ist eine GUID.
Die Funktion liefert eine ID, mit deren Hilfe die erzeugte Instanz eindeutig identifiziert werden kann. Das erzeugte Objekt ist nach dem Anlegen noch unkonfiguriert und mit mit den entsprechenden gerätespezifischen Funktionen konfiguriert werden. Anschließend muss die Konfiguration mit der Funktion IPS_ApplyChanges übernommen werden.

## BEISPIEL

```
//FS20 Instanz erstellen mit dem Namen "Stehlampe"
$InsID = IPS_CreateInstance("{48FCFDC1-11A5-4309-BB0B-
A0DB8042A969}");

IPS_SetName($InsID, "Stehlampe"); // Instanz benennen
IPS_SetParent($InsID, 12345); // Instanz einsortieren unter dem
Objekt mit der ObjektID "12345"


//Konfiguration
IPS_SetProperty($InsID, "HomeCode", "12345678"); // Ändere
Eigenschaft "HomeCode"
IPS_ApplyChanges($InsID); // Übernehme Änderungen -> Die Instanz
benutzt den geänderten HomeCode
```

# IPS_DeleteInstance

```
boolean IPS_DeleteInstance (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der zu löschenden Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion löscht eine vorhandene Instanz mit der ID **InstanzID**.



Die Instanz darf nur gelöscht werden, wenn keine Unterobjekte mehr vorhanden sind.

## BEISPIEL

```
// Löschen der Instanz 47788
IPS_DeleteInstance(47788);
```

# IPS_GetInstance

```
array IPS_GetInstance (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | Die zu zeigende InstanzID |

## RÜCKGABEWERT

Folgende Informationen stehen als key => value Paare zur Verfügung:

| INDEX | TYP | BESCHREIBUNG |
|---|---|---|
| InstanceID | integer | InstanzID |
| ConnectionID | integer | ID der verbundenen Instanz |
| InstanceStatus | integer | siehe Tabelle "Status der Instanz" |
| InstanceSupportsSearching | boolean | Unterstützt die Instanz den Suchmodus |
| InstanceIsSearching | boolean | Die Instanz befindet sich gerade im Suchmodus |
| InstanceChanged | integer | Unix Zeitstempel zu dem das letzte mal die Konfiguration übernommen wurde |
| ModuleInfo | array | siehe Tabelle "Modulinformationen" |

*Tabelle: Status der Instanz*

| CODE | STATUS |
|---|---|
| 101 | Instanz wird erstellt |
| 102 | Instanz ist aktiv |
| 103 | Instanz wird gelöscht |
| 104 | Instanz ist inaktiv |
| 105 | Instanz wurde nicht erstellt |
| 106 | Instanz ist im Standby |
| >=200 | Instanz ist fehlerhaft |

*Tabelle: Modulinformationen*

| INDEX | TYP | BESCHREIBUNG |
|---|---|---|
| ModuleID | string | ModulID zur Instanz |
| ModuleName | string | ModulName der Instanz |
| ModuleType | integer | ModulTyp der Instanz (0: Kern, 1: I/O, 2: Splitter, 3: Gerät, 4: Konfigurator, 5: Discovery, 6: Visualisierung) |

## BESCHREIBUNG

Die Funktion liefert ein Array mit Informationen über die Instanz mit der ID **InstanzID**. Sollte die Instanz nicht vorhanden sein, wird eine Warnung generiert.

## BEISPIEL

```
print_r(IPS_GetInstance(19668));

/* liefert z.B.:
Array
(
[InstanceID] => 19668
[InstanceStatus] => 102
[LastChange] => 0
[ModuleInfo] => Array
(
[ModuleID] => {48FCFDC1-11A5-4309-BB0B-A0DB8042A969}
[ModuleName] => FS20
[ModuleType] => 3
)
[ConnectionID] => 29416
)
*/
```

# IPS_GetInstanceIDByName

```
int IPS_GetInstanceIDByName (string $InstanzName, int
$ÜbergeordneteID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzName | Zu suchender InstanzName |
| ÜbergeordneteID | Objekt, in deren direkten Unterobjekten gesucht werden soll |

## RÜCKGABEWERT

ID der gefundenen Instanz, sonst FALSE

## BESCHREIBUNG

Die Funktion versucht die ID der Instanz mit dem Namen **InstanzName**, deren übergeordnetes Objekt die ID **ÜbergeordneteID** besitzt, zu ermitteln. Es wird die ID der ersten gefundenen Instanz verwendet, deren Name mit **InstanzName** übereinstimmt. Falls eine solche Instanz gefunden wurde, wird deren ID zurückgemeldet, andernfalls FALSE.

**Vorsicht:** Da Namen in IP-Symcon beliebig vergeben werden können, und damit **nicht eindeutig** sind, kann die Funktion eine falsche ID zurückmelden.

Diese Funktion generiert eine **Warnung**, sollte der Name nicht gefunden werden. Sollte der Name nicht gefunden werden, muss der @ Operator genutzt, falls diese Meldung unterdrückt werden soll. Die genau Funktionsweise dieses Operators kann dem PHP Handbuch entnommen werden.

## BEISPIEL

```
$InstanzID = @IPS_GetInstanceIDByName("Regenmenge",
$ÜbergeordneteID);
if ($InstanzID === false)
echo "Instanz nicht gefunden!";
else
```

```
echo "Die Instanz-ID lautet: ". $InstanzID;
```

# IPS_GetInstanceList

```
array IPS_GetInstanceList ()
```

## RÜCKGABEWERT

Ein Array von Integer-Werten aller IDs der Instanzen in IP-Symcon

## BESCHREIBUNG

Die Funktion ermittelt die IDs aller in IP-Symcon registrierten IPS-Instanzen. Die IDs werden in einem Array aufgelistet. Falls keine Instanz existiert, ist das Array leer.

## BEISPIEL

```
$alleInstanzen = IPS_GetInstanceList();
print_r($alleInstanzen);

/* liefert z.B.:
Array
(
[0] => 37659
[1] => 18326
usw. ...
*/
```

# IPS_GetInstanceListByModuleID

```
array IPS_GetInstanceListByModuleID (string $ModulID)
```

## PARAMETERLISTE

| | |
|---|---|
| ModulID | GUID |

## RÜCKGABEWERT

Ein Array von Integer-Werten aller gefundenen IDs

## BESCHREIBUNG

Die Funktion gibt alle Instanzen, die der angegebenen ModulID (GUID) entsprechen, in einem Array zurück.



Die GUIDs können bei IPS_GetModuleList nachgeschlagen werden.

## BEISPIEL

```
//FS20 Sender
$guid = "{48FCFDC1-11A5-4309-BB0B-A0DB8042A969}";
//Auflisten
print_r(IPS_GetInstanceListByModuleID($guid));
```

# IPS_GetInstanceListByModuleType

```
array IPS_GetInstanceListByModuleType (int $ModulTyp)
```

## PARAMETERLISTE

| WERT | BESCHREIBUNG |
|---|---|
| 0 | Kern |
| 1 | I/O |
| 2 | Splitter |
| 3 | Geräte |
| 4 | Konfigurator |
| 5 | Discovery |
| 6 | Visualisierung |

ModulTyp

## RÜCKGABEWERT

Ein Array von Integer-Werten aller IDs der Instanzen vom Typ **InstanzTyp** in IP-Symcon

## BESCHREIBUNG

Die Funktion ermittelt die IDs aller Instanzen eines bestimmten **ModulTyps**. Die IDs werden in einem Array aufgelistet. Falls keine Instanz des angeforderten Typs existiert, ist das Array leer.
Es werden nur Objekte des Modultyps **ModulTyp** aufgelistet. Sollen alle Instanzen aufgelistet werden, so kann hierfür die Funktion IPS_GetInstanceList verwendet werden.

## BEISPIEL

```
$alleInstanzen = IPS_GetInstanceListByModuleType(1); // nur I/O
Instanzen auflisten
print_r($alleInstanzen);

/* liefert z.B.:
Array
(
[0] => 37659
[1] => 18326
usw. ...
*/
```

# IPS_InstanceExists

```
boolean IPS_InstanceExists (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der zu prüfenden Instanz |

## RÜCKGABEWERT

Falls die InstanzID im System existiert, wird **TRUE** zurückgemeldet, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion prüft, ob die Instanz mit der ID **InstanzID** existiert.

## BEISPIEL

```
if (IPS_InstanceExists(45724))
echo "Instanz existiert bereits!";
```

# Konfiguration

| | |
|---|---|
| IPS_ApplyChanges | übernimmt eine veränderte Konfiguration |
| IPS_GetConfiguration | liest eine Konfiguration aus |
| IPS_GetConfigurationForm | liest eine Konfigurationseite aus |
| IPS_GetProperty | liest den aktuellen Wert einer Eigenschaft |
| IPS_HasChanges | prüft, ob eine Konfiguration verändert wurde |
| IPS_ResetChanges | setzt eine veränderte Konfiguration auf die aktuelle zurück |
| IPS_SetConfiguration | setzt eine neue Konfiguration |
| IPS_SetProperty | setzt einen neuen Wert für eine Eigenschaft |

# IPS_ApplyChanges

```
boolean IPS_ApplyChanges (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Die Funktion übernimmt die neue Soll-Konfiguration als aktuelle Ist-Konfiguration und initialisiert bei Bedarf entsprechende Modulkomponenten neu. (Z.B. wird bei einem Socket Modul die Netzwerkverbindung getrennt und neu aufgebaut.)

## BEISPIEL

```
if(IPS_HasChanges(12345))
{
IPS_ApplyChanges(12345);
}
```

# IPS_GetConfiguration

Benötigt: IP-Symcon >= 2.7

```
string IPS_GetConfiguration (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte die Funktion erfolgreich ausgeführt werden, liefert er als Ergebnis die Konfiguration der Instanz als String.

## BESCHREIBUNG

Die Funktion gibt die Konfiguration der Instanz mit der ID **InstanzID** als String zurück.

## BEISPIEL

```
// Liest die Konfiguration der Instanz 12345 aus und gibt diese via
"echo" zurück.
$config = IPS_GetConfiguration(12345);
echo $config;

// Beispielausgabe einer ModBus-Instanz
{"DataType":3,"WriteAddress":0,"ReadAddress":0,"Poller":3600000,"ReadO
```

# IPS_GetConfigurationForm

Benötigt: IP-Symcon >= 2.7

```
string IPS_GetConfigurationForm (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte die Funktion erfolgreich ausgeführt werden, liefert er als Ergebnis die Konfigurationseite der Instanz als JSON-kodierter String.

## BESCHREIBUNG

Die Funktion gibt die Konfigurationseite der Instanz mit der ID **InstanzID** als JSON-kodierter String zurück.

## BEISPIEL

```
// Liest die Konfiguration der Instanz 12345 aus und gibt diese via
"echo" zurück.
$configpage = json_decode(IPS_GetConfigurationForm(12345));
var_dump ($configpage->elements);

//Beispielausgabe
array(2) {
[0]=>
object(stdClass)#2 (2) {
["type"]=>
string(5) "Label"
["label"]=>
string(28) "Minimum needed daily changes"
}
[1]=>
object(stdClass)#3 (3) {
["type"]=>
string(13) "NumberSpinner"
["name"]=>
string(19) "RequiredSwitchCount"
["caption"]=>
string(5) "Count"
```

```
  }
  }
```

# IPS_GetProperty

Benötigt: IP-Symcon >= 2.7

```
variant IPS_GetProperty (int $InstanzID, string $Eigenschaft)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| Eigenschaft | Name der Eigenschaft |

## RÜCKGABEWERT

Aktueller Wert der Eigenschaft

## BESCHREIBUNG

Die Funktion liest für die **InstanzID** den Ist-Wert der **Eigenschaft** ein. Eine genaue Auflistung der möglichen Eigenschaften für die einzelnen Instanzen kann in der Beschreibung der Funktion IPS_SetProperty gefunden werden.
Es kann ein neuer Soll-Wert mit IPS_SetProperty übergeben und mit IPS_ApplyChanges übernommen werden.

## BEISPIEL

```
echo IPS_GetProperty($id, "Open"); //Ist die I/O Instanz aktiv?
```

# IPS_HasChanges

```
boolean IPS_HasChanges (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der zu überprüfenden Instanz |

## RÜCKGABEWERT

TRUE, wenn es Änderungen gibt, sonst FALSE

## BESCHREIBUNG

Die Funktion überprüft, ob bei einer Instanz Änderungen an der Konfiguration vorgenommen wurden, die als aktuelle Ist-Konfiguration übernommen werden soll.

## BEISPIEL

```
if(IPS_HasChanges(12345))
{
IPS_ApplyChanges(12345);
}
```

# IPS_ResetChanges

```
boolean IPS_ResetChanges (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der zu ändernden Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Die Funktion überschreibt die neue Soll-Konfiguration durch die aktuelle Ist-Konfiguration.

## BEISPIEL

```
if(IPS_HasChanges(12345))
{
IPS_ResetChanges(12345);
}
```

# IPS_SetConfiguration

Benötigt: IP-Symcon >= 2.7

```
boolean IPS_SetConfiguration (int $InstanzID, string $Konfiguration)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| Konfiguration | Die Konfiguration als String |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion setzt für die **InstanzID** die Konfiguration **Konfiguration** ein.
Um diese zu speichern und wirksam werden zu lassen, muss im Anschluss IPS_ApplyChanges ausgeführt werden.

## BEISPIEL

```
// Hier soll die Konfiguration der ModBus-Instanz mit der ID 12345
gesetzt werden.
IPS_SetConfiguration(12345,
'{"DataType":3,"WriteAddress":1,"ReadAddress":0,"Poller":3600000,"Read(
IPS_ApplyChanges(12345); //Neue Konfiguration übernehmen
```

# IPS_SetProperty

Benötigt: IP-Symcon >= 2.7

```
boolean IPS_SetProperty (int $InstanzID, string $Eigenschaft,
variant $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| Eigenschaft | Name der Eigenschaft |
| Wert | Neuer Wert der Eigenschaft |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion setzt für die **InstanzID** den Soll-Wert der **Eigenschaft** ein.
Es kann ein neuer Soll-Wert mit IPS_SetProperty übergeben und mit IPS_ApplyChanges übernommen werden.



Sollte in der folgenden Auflistung Informationen fehlen, z.B. welche Werte und Assoziationen zu bestimmten Eigenschaften gehören, kann im Ordner "/forms" in der Symcon Installation zu den jeweiligen Instanzen nachgesehen werden.

ALL2x

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceNum | integer | 0 |
| DeviceType | integer | 0 |
| Interval | integer | 0 |
| IPAddress | string | |

ALL3690

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Interval | integer | 0 |
| IPAddress | string | |

ALL3691

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Interval | integer | 0 |
| IPAddress | string | |

ALL4000

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Devices | string | [0,0,0,

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Device | integer | 128 |
| Interval | integer | 0 |
| IPAddress | string | |
| NodeNumber | integer | 0 |
| RequestDate | boolean | |
| RequestDLBus | boolean | |
| RequestElectricalPower | boolean | |
| RequestGeneral | boolean | |
| RequestInputs | boolean | 1 |
| RequestKNX | boolean | |
| RequestLoggingAnalog | boolean | |
| RequestLoggingDigital | boolean | |
| RequestMBus | boolean | |
| RequestModbus | boolean | |
| RequestNetworkAnalog | boolean | |
| RequestNetworkDigital | boolean | |
| RequestOutputs | boolean | 1 |
| RequestSun | boolean | |
| RequestTime | boolean | |

Connect Control

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Active | boolean | 1 |

Cutter

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeleteCutChars | boolean | |
| InputLength | integer | 0 |
| LeftCutChar | string | |
| LeftCutCharAsHex | boolean | |
| ParseType | integer | 0 |
| RightCutChar | string | |
| RightCutCharAsHex | boolean | |
| SyncChar | string | |
| SyncCharAsHex | boolean | |
| Timeout | integer | 0 |

DMX Gateway

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| GatewayMode | integer | 0 |
| Universe | integer | 0 |

DMX Output

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| ChannelBase | integer | 1 |
| ChannelCount | integer | 1 |
| ChannelSize | integer | 1 |

DMX RGB

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| ChannelB | integer | 3 |
| ChannelG | integer | 2 |
| ChannelR | integer | 1 |
| ChannelSize | integer | 1 |

DNSSD Control

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Services | string | [{"Name":"Symcon","RegType":"_hap._tcp","Domain":"","Host":"","Port":<br>[{"Value":"md=Symcon"},{"Value":"pv=1.0"},{"Value":"id=AA:B1:51:EB:0(<br>{"Value":"c#=1"},{"Value":"s#=1"},{"Value":"ff=0"},{"Value":"ci=2"},{"Valu |

dS Apartment

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| GroupID | integer | 0 |

dS Configurator

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

dS Expert

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

dS Joker

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| BinaryInputsInterval | integer | 0 |
| dSID | string | |
| SensorInputsInterval | integer | 0 |

dS Light

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| dSID | string | |

dS Shutter

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| dSID | string | |

dS Splitter

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| IPAddress | string | dss.local |
| Token | string | |

dS Zone

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| GroupID | integer | 0 |
| ZoneID | integer | 0 |

Dummy Module

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |

ekey Fingerprint

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| Protocol | integer | 0 |
| Separator | string | _ |

EnOcean alphaEOS SENSE TF-H

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean Configurator

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |

EnOcean D20500

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean D20501

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ChannelID | integer | 0 |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean D20502

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean Discovery

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |

EnOcean EEP A50201RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50202RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50203RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50204RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50205RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50206RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50207RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50208RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50209RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A5020ARX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A5020BRX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50210RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50211RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50212RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50213RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A50214RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A50215RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A50216RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A50217RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A50218RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A50219RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A5021ARX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A5021BRX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A50220RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A50230RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A50401RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A50402RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50403RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50601RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50602RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50603RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50701RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50702RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50703RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50801RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50802RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50803RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50901RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50902RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A50904RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A50905RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A50906RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A50907RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A50908RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A51001RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A51002RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A51003RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A51004RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A51005RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A51006RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A51007RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A51008RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A51009RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A5100ARX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A5100BRX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A5100CRX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A5100DRX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A51010RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A51011RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A51012RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A51013RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A51014RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A51015RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A51016RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A51017RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A51018RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A51019RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A5101ARX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A5101BRX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A5101CRX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A5101DRX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A5101FRX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A51020RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A51021RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A51022RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A51023RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A51101RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A51102RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A51103RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A51104RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A51200RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A51201RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A51202RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A51203RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A51301RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A51302RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A51303RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A51304RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A51305RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP A51401RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A51402RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A51403RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A51404RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A51405RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A51406RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A51407RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A51408RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A51409RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A5140ARX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A52006

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | integer | 0 |
| ReturnID | string | 00000000 |
| SPMode | integer | 0 |

EnOcean EEP A52012RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A52012TX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | integer | 0 |

EnOcean EEP A53001RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A53002RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A53003RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A53004RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A53701RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A53808RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP A53809RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP D20100

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| ChannelID | integer | 0 |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean EEP D20101

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| ChannelID | integer | 0 |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean EEP D20102

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ChannelID | integer | 0 |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean EEP D20103

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ChannelID | integer | 0 |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean EEP D20104

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ChannelID | integer | 0 |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean EEP D20105

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ChannelID | integer | 0 |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean EEP D20106

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ChannelID | integer | 0 |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean EEP D20107

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ChannelID | integer | 0 |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean EEP D20108

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ChannelID | integer | 0 |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean EEP D20109

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ChannelID | integer | 0 |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean EEP D2010A

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ChannelID | integer | 0 |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean EEP D2010B

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ChannelID | integer | 0 |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean EEP D2010C

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ChannelID | integer | 0 |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean EEP D2010D

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ChannelID | integer | 0 |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean EEP D2010E

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ChannelID | integer | 0 |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean EEP D2010F

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ChannelID | integer | 0 |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean EEP D20110

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ChannelID | integer | 0 |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean EEP D20111

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ChannelID | integer | 0 |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean EEP D20112

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ChannelID | integer | 0 |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean EEP D20113

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ChannelID | integer | 0 |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean EEP D20114

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ChannelID | integer | 0 |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean EEP D20300

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DestinationID | string | |

EnOcean EEP D2030A

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DestinationID | string | |

EnOcean EEP D20310

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DestinationID | string | |

EnOcean EEP D20601

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean EEP D20700

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DestinationID | string | |

EnOcean EEP D21101

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | |

EnOcean EEP D21102

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | |

EnOcean EEP D21103

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | |

EnOcean EEP D21104

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | |

EnOcean EEP D21105

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | |

EnOcean EEP D21106

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | |

EnOcean EEP D21107

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | |

EnOcean EEP D21108

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | |

EnOcean EEP D21430

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DestinationID | string | |

EnOcean EEP D21500

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DestinationID | string | |

EnOcean EEP D23200

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DestinationID | string | |

EnOcean EEP D23201

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DestinationID | string | |

EnOcean EEP D23202

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DestinationID | string | |

EnOcean EEP D2A001

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DestinationID | string | |
| EmulateStatus | boolean | 1 |

EnOcean EEP D50001RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP F60101RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP F60201RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP F60202RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP F60203RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP F60204RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP F60301RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP F60302RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP F60401RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP F60402RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EEP F60500RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP F60501RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP F60502RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP F61000RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EEP F61001RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EltakoDimmer

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ButtonMode | integer | 0 |
| DeviceID | integer | 0 |
| DualStatus | boolean | |
| EmulateStatus | boolean | 1 |
| ReturnID | string | 00000000 |

EnOcean EltakoFABH130

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EltakoFABH65S

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EltakoFAFT60

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EltakoFAH60

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean EltakoFHK14

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | integer | 0 |
| Mode | integer | 0 |
| ReturnID | string | 00000000 |

EnOcean EltakoFKC

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EltakoFRW

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EltakoFSS12

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EltakoFTR65

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EltakoFWS61

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EltakoFZS

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean EltakoRGBW

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | integer | 0 |
| EmulateStatus | boolean | 1 |
| ReturnID | string | 00000000 |

EnOcean EltakoShutter

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| ButtonMode | integer | 0 |
| DeviceID | integer | 0 |
| EmulateStatus | boolean | 1 |
| ReturnID | string | 00000000 |

EnOcean EltakoSwitch

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| ButtonMode | integer | 0 |
| DeviceID | integer | 0 |
| DualStatus | boolean | |
| EmulateStatus | boolean | 1 |
| ReturnID | string | 00000000 |

EnOcean EltakoSwitchEx

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | integer | 0 |
| EmulateStatus | boolean | 1 |
| ReturnID | string | 00000000 |

EnOcean EltakoWWKW

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | integer | 0 |
| EmulateStatus | boolean | 1 |
| ReturnID | string | 00000000 |

EnOcean Gateway

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| GatewayMode | integer | 3 |
| NoiseThreshold | integer | -1 |

EnOcean Hoppe

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00000000 |

EnOcean Opus

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ButtonMode | integer | 0 |
| DeviceID | integer | 0 |

EnOcean PTM200

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | string | 00102FFF |

EnOcean PTM200RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ButtonMode | integer | 0 |
| DeviceID | string | 00000000 |

EnOcean RCM100

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ButtonMode | integer | 0 |
| DeviceID | integer | 0 |

EnOcean Shutter

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| ButtonMode | integer | 0 |
| DeviceID | integer | 0 |

EnOcean SmartDrive MX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | integer | 0 |
| ReturnID | string | 00000000 |
| WUC | integer | 0 |

EnOcean STM100

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean STM250

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |

EnOcean Thermokon SR0x

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |
| TempOffset | double | 0 |
| TempRange | double | 40 |

EnOcean Thermokon Thanos

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| BaseSetPoint1 | integer | 22 |
| DeviceID | integer | 0 |
| LowerSetPoint1 | integer | 17 |
| ReturnID | string | 00000000 |
| StepSetPoint1 | double | 0,5 |
| UpperSetPoint1 | integer | 30 |

EnOcean Valve Actuator

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | integer | 0 |
| ReturnID | string | 00000000 |
| SPMode | integer | 0 |

Event Control

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| ShutdownScript | integer | 0 |
| StartupScript | integer | 0 |
| StatusEvents | string | [] |
| WatchdogScript | integer | 0 |

FHT

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address | string | 0000 |
| EmulateStatus | boolean | |

FHZ

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

FS10

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | integer | 0 |
| DeviceType | integer | 0 |

FS10 Gateway

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

FS20

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address | string | 11 |
| EnableReceive | boolean | 1 |
| EnableTimer | boolean | 1 |
| HomeCode | string | 11111111 |
| Mapping | string | [] |
| SubAddress | string | 11 |

FS20EX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceList | string | [] |
| HomeCode | string | 11111111 |

Heating Control

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Hysteresis | double | 2 |
| Mode | integer | 0 |
| OverrideVariables | string | [] |
| Range | double | 3 |
| ResendInterval | integer | 15 |
| SetBack | double | 2 |
| SetBackVariables | string | [] |
| SourceID | integer | 0 |
| TargetVariables | string | [] |
| Type | integer | 0 |

Heating Control Legacy

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| HandlerScript | integer | 0 |
| Hysteresis | double | 0 |
| OverrideVariables | string | [] |
| ResendInterval | integer | 15 |
| SetBack | double | 0 |
| SetBackVariables | string | [] |
| SourceVariable | integer | 0 |
| TransmitDevices | string | [] |

HID

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceProductID | integer | 0 |
| DeviceSerial | string | |
| DeviceVendorID | integer | 0 |
| Open | boolean | |

HMS

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | integer | 0 |
| DeviceType | integer | 0 |

HomeMatic Configurator

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |

HomeMatic Device

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| Address | string | |
| EmulateStatus | boolean | 1 |
| Protocol | integer | 0 |

HomeMatic Discovery

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |

HomeMatic Socket

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| GRKeepAliveTimeout | integer | 30 |
| GROpen | boolean | |
| GRPort | integer | 9292 |
| GRSSLPort | integer | 49292 |
| Host | string | |
| HSOpen | boolean | 1 |
| HSPort | integer | 8181 |
| HSSSLPort | integer | 48181 |
| IPAddress | string | 172.17.31.180 |
| IPKeepAliveTimeout | integer | 30 |
| IPOpen | boolean | 1 |
| IPPort | integer | 2010 |
| IPSSLPort | integer | 42010 |
| Open | boolean | |
| Password | string | |
| Port | integer | 5544 |
| RFKeepAliveTimeout | integer | 30 |
| RFOpen | boolean | |
| RFPort | integer | 2001 |
| RFSSLPort | integer | 42001 |
| Username | string | Admin |
| UseSSL | boolean | |
| VerifyHost | boolean | 1 |
| VerifyPeer | boolean | 1 |
| WaitTimeConfigurator | integer | 25000 |
| WaitTimeDevice | integer | 7000 |
| WaitTimePing | integer | 5000 |
| WaitTimeServiceMessages | integer | 10000 |
| WRKeepAliveTimeout | integer | 30 |
| WROpen | boolean | |
| WRPort | integer | 2000 |
| WRSSLPort | integer | 42000 |

HTTP Client

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Active | boolean | 1 |
| AuthPass | string | |
| AuthType | integer | 0 |
| AuthUser | string | |
| Interval | integer | 0 |
| URL | string | |
| UseBasicAuth | boolean | |
| VerifyHost | boolean | 1 |
| VerifyPeer | boolean | 1 |

Image Grabber

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| AuthPass | string | |
| AuthType | integer | 0 |
| AuthUser | string | |
| ImageAddress | string | http://localhost/cam.jpg |
| ImageType | integer | 0 |
| Interval | integer | 300 |
| UseBasicAuth | boolean | |
| UseCaching | boolean | 1 |

IMAP

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| CacheInterval | integer | 300 |
| CacheSize | integer | 10 |
| Host | string | imap.gmail.com |
| Password | string | |
| Port | integer | 993 |
| UseAuthentication | boolean | 1 |
| Username | string | mail@gmail.com |
| UseSSL | boolean | 1 |
| VerifyHost | boolean | 1 |
| VerifyPeer | boolean | 1 |

IPS-868 Accelerometer

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | integer | 0 |

IPS-868 AD

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| Channel | integer | 1 |
| DeviceID | integer | 184 |
| Interval | integer | 0 |

IPS-868 AirQuality

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | integer | 176 |
| Interval | integer | 0 |
| LimitRed | integer | 1500 |
| LimitYellow | integer | 1000 |

IPS-868 Configurator

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |

IPS-868 Counter

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| CounterID | integer | 1 |
| DeviceID | integer | 128 |
| Impulses | integer | 1000 |
| Interval | integer | 0 |
| Mode | integer | 1 |

IPS-868 DA

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| Channel | integer | 1 |
| DeviceID | integer | 184 |

IPS-868 Data

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | integer | 184 |
| Request | string | |

IPS-868 Discovery

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |

IPS-868 Display

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | integer | 0 |

IPS-868 Display Input

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | integer | 0 |

IPS-868 Display Output

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | integer | 0 |
| OutputID | integer | 0 |

IPS-868 Gateway

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| GatewayMode | integer | 0 |

IPS-868 Input

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | integer | 0 |
| InputID | integer | 1 |

IPS-868 Level

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | integer | 166 |
| Interval | integer | 0 |

IPS-868 Servo

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | integer | 144 |

IPS-868 Servo Input

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | integer | 1 |

IPS-868 Stripe

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | integer | 144 |
| Mode | integer | 0 |

IPS-868 Stripe Input

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | integer | 1 |

IPS-868 Thermo

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | integer | 166 |
| Interval | integer | 0 |
| Mode | integer | 1 |

IPS-868 Tracker

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | integer | 0 |
| TrackerID | integer | 1 |

IPS-868 WatchDogTimer

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | integer | 165 |

IRTrans Device

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | integer | 0 |

IRTrans Gateway

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

JSON Decoder

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

KNX Configurator

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| EnableStatusDetection | boolean | |
| ExportMode | integer | 1 |
| KeyringExport | string | |
| KeyringPassword | string | |
| KeyringVisible | boolean | |
| OPCExport | string | |
| StatusDetectionKeywords | string | [{"Keyword": " status"},{"Keyword": ".status"}, {"Keyword": "-status"},{"Keyword": "status "}] |
| StatusDetectionMatching | integer | 0 |
| XMLExport | string | |

KNX Discovery

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

KNX DPT 1

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 10

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 11

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 12

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 13

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 14

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 15

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 16

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 17

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 18

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 19

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 2

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 20

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 200

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 201

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 202

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 203

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 204

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 205

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 206

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 207

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 209

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 21

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 210

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 211

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 212

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 213

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 214

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 215

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 216

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 217

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 218

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 219

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 22

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 220

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 221

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 222

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 223

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 224

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 225

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 229

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 23

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 230

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 231

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 232

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 234

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 235

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 236

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 237

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 238

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 239

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 240

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 241

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 242

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 249

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 25

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 251

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 26

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 27

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 28

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 29

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 3

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 30

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 31

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 4

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 5

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 6

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 7

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 8

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX DPT 9

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address1 | integer | 0 |
| Address2 | integer | 0 |
| Address3 | integer | 0 |
| CapabilityRead | boolean | |
| CapabilityReceive | boolean | 1 |
| CapabilityTransmit | boolean | |
| CapabilityWrite | boolean | 1 |
| Dimension | integer | 0 |
| EmulateStatus | boolean | 1 |
| Mapping | string | [] |

KNX EIS Group

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| GroupAddress1 | integer | 0 |
| GroupAddress2 | integer | 0 |
| GroupAddress3 | integer | 0 |
| GroupCapabilityRead | boolean | |
| GroupCapabilityReceive | boolean | 1 |
| GroupCapabilityTransmit | boolean | |
| GroupCapabilityWrite | boolean | 1 |
| GroupEmulateStatus | boolean | 1 |
| GroupFunction | string | Switch |
| GroupInterpretation | string | Standard |
| GroupMapping | string | [] |

KNX Gateway

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| GatewayMode | integer | 1 |
| SendLimit | integer | 6 |

KNX RGB

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| GroupAddress1 | integer | 0 |
| GroupAddress2 | integer | 0 |
| GroupAddress3 | integer | 0 |
| GroupCapabilityRead | boolean | |
| GroupCapabilityReceive | boolean | 1 |
| GroupCapabilityTransmit | boolean | |
| GroupCapabilityWrite | boolean | 1 |
| GroupEmulateStatus | boolean | 1 |
| GroupMapping | string | [] |

KNX RGBW

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| GroupAddress1 | integer | 0 |
| GroupAddress2 | integer | 0 |
| GroupAddress3 | integer | 0 |
| GroupCapabilityRead | boolean | |
| GroupCapabilityReceive | boolean | 1 |
| GroupCapabilityTransmit | boolean | |
| GroupCapabilityWrite | boolean | 1 |
| GroupEmulateStatus | boolean | 1 |
| GroupMapping | string | [] |

KNX Shutter

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| EnableBladePosition | boolean | |
| EnablePosition | boolean | |
| EnableStep | boolean | |
| GroupBladePositionAddress1 | integer | 0 |
| GroupBladePositionAddress2 | integer | 0 |
| GroupBladePositionAddress3 | integer | 0 |
| GroupBladePositionMapping | string | [] |
| GroupMoveAddress1 | integer | 0 |
| GroupMoveAddress2 | integer | 0 |
| GroupMoveAddress3 | integer | 0 |
| GroupMoveMapping | string | [] |
| GroupPositionAddress1 | integer | 0 |
| GroupPositionAddress2 | integer | 0 |
| GroupPositionAddress3 | integer | 0 |
| GroupPositionMapping | string | [] |
| GroupStepAddress1 | integer | 0 |
| GroupStepAddress2 | integer | 0 |
| GroupStepAddress3 | integer | 0 |
| GroupStepMapping | string | [] |
| GroupStopAddress1 | integer | 0 |
| GroupStopAddress2 | integer | 0 |
| GroupStopAddress3 | integer | 0 |
| GroupStopMapping | string | [] |

KS300

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

LCN Configurator

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

LCN Data

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DataType | integer | 0 |
| Threshold11ValueType | integer | 13 |
| Threshold12ValueType | integer | 13 |
| Threshold13ValueType | integer | 13 |
| Threshold14ValueType | integer | 13 |
| Threshold1ValueType | integer | 13 |
| Threshold21ValueType | integer | 13 |
| Threshold22ValueType | integer | 13 |
| Threshold23ValueType | integer | 13 |
| Threshold24ValueType | integer | 13 |
| Threshold2ValueType | integer | 13 |
| Threshold31ValueType | integer | 13 |
| Threshold32ValueType | integer | 13 |
| Threshold33ValueType | integer | 13 |
| Threshold34ValueType | integer | 13 |
| Threshold3ValueType | integer | 13 |
| Threshold41ValueType | integer | 13 |
| Threshold42ValueType | integer | 13 |
| Threshold43ValueType | integer | 13 |
| Threshold44ValueType | integer | 13 |
| Threshold4ValueType | integer | 13 |
| Threshold5ValueType | integer | 13 |
| Threshold6ValueType | integer | 13 |

LCN Display

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

LCN Gateway

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| OutputMode | integer | 2 |
| Password | string | |
| RequireAcknowledge | boolean | 1 |
| Username | string | |

LCN Module

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Segment | integer | 0 |
| Target | integer | 0 |

LCN RGBW

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

LCN Shutter

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DirectionChannel | integer | 1 |
| DirectionInvert | boolean | |
| MoveChannel | integer | 1 |
| Unit | integer | 0 |

LCN ShutterMotor

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Interval | integer | 0 |
| Motor | integer | 1 |

LCN Unit

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Channel | integer | 0 |
| Unit | integer | 0 |

LCN Value

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| A10ValueEnabled | boolean | |
| A10ValueType | integer | 0 |
| A11ValueEnabled | boolean | |
| A11ValueType | integer | 0 |
| A12ValueEnabled | boolean | |
| A12ValueType | integer | 0 |
| A1ValueEnabled | boolean | |
| A1ValueType | integer | 0 |
| A2ValueEnabled | boolean | |
| A2ValueType | integer | 0 |
| A3ValueEnabled | boolean | |
| A3ValueType | integer | 0 |
| A4ValueEnabled | boolean | |
| A4ValueType | integer | 0 |
| A5ValueEnabled | boolean | |
| A5ValueType | integer | 0 |
| A6ValueEnabled | boolean | |
| A6ValueType | integer | 0 |
| A7ValueEnabled | boolean | |
| A7ValueType | integer | 0 |
| A8ValueEnabled | boolean | |
| A8ValueType | integer | 0 |
| A9ValueEnabled | boolean | |
| A9ValueType | integer | 0 |
| C1CounterEnabled | boolean | |
| C2CounterEnabled | boolean | |
| C3CounterEnabled | boolean | |
| C4CounterEnabled | boolean | |
| EmulateStatus | boolean | |
| Interval | integer | 0 |
| R1TargetEnabled | boolean | |
| R1ValueEnabled | boolean | |
| R1ValueType | integer | 0 |
| R2TargetEnabled | boolean | |
| R2ValueEnabled | boolean | |
| R2ValueType | integer | 0 |
| S1TargetEnabled | boolean | |
| S2TargetEnabled | boolean | |

LevelJet

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

Location Control

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| IntervalPosition | integer | 300 |
| IsDayEnd | integer | 1 |
| IsDayStart | integer | 2 |
| Location | string | {"latitude": 0, "longitude": 0} |
| Offsets | string | [] |

M-Bus Configurator

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address | integer | 1 |
| DefaultAddressing | integer | -1 |
| DefaultInterval | integer | 5 |
| DefaultLimit | integer | 0 |

M-Bus Device

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address | integer | 1 |
| Interval | integer | 5 |
| Limit | integer | 0 |
| SecondaryAddress | string | 00000000 |
| UseSecondaryAddress | boolean | |

M-Bus Discovery

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

M-Bus Gateway

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| GatewayMode | integer | 1 |
| ResetMode | integer | 1 |

Media Player

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceDriver | string | |
| DeviceName | string | |
| DeviceNum | integer | 0 |
| UpdateInterval | integer | 0 |

ModBus Address

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DataType | integer | 0 |
| EmulateStatus | boolean | 1 |
| Factor | double | 0 |
| Length | integer | 0 |
| Poller | integer | 5000 |
| ReadAddress | integer | 0 |
| ReadFunctionCode | integer | 1 |
| SwapBytes | boolean | |
| WriteAddress | integer | 0 |
| WriteFunctionCode | integer | 5 |

ModBus Configurator

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

ModBus Gateway

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DataBlocks | string | [] |
| DelayTimeRTU | integer | 20 |
| DelayTimeTCP | integer | 10 |
| DeviceID | integer | 1 |
| GatewayMode | integer | 0 |
| SwapWords | boolean | 1 |

Module Control

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

MQTT Client

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| ClientID | string | 2c892877d8248807bb30 |
| Password | string | |
| Subscriptions | string | [{"Topic":"#", "QoS": 0}] |
| UserName | string | |

MQTT Client Configurator

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

MQTT Client Device

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Retain | boolean | |
| Topic | string | |
| Type | integer | 3 |

MQTT Server

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Password | string | |
| RetainedMessageExpiration | integer | 604800 |
| SessionQueueLimit | integer | 500 |
| SessionTimeout | integer | 3600 |
| UserName | string | |

MQTT Server Configurator

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

MQTT Server Device

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Retain | boolean | |
| Topic | string | |
| Type | integer | 3 |

Multicast Socket

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| BindIP | string | 172.17.31.180 |
| BindPort | integer | 0 |
| EnableBroadcast | boolean | |
| EnableLoopback | boolean | |
| EnableReuseAddress | boolean | |
| Host | string | |
| MulticastIP | string | 239.255.255.250 |
| Open | boolean | |
| Port | integer | 0 |

Notification Control

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Active | boolean | 1 |

OneWire Configurator

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

OneWire Discovery

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

OneWire F05

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | |
| Interval | integer | 60 |
| Invert | boolean | |
| PortSettings | integer | 0 |

OneWire F10

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | |
| Interval | integer | 60 |

OneWire F12

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | |
| Interval | integer | 60 |
| Invert0 | boolean | |
| Invert1 | boolean | |
| PortSettings0 | integer | 0 |
| PortSettings1 | integer | 0 |

OneWire F1D

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | |
| Interval | integer | 60 |

OneWire F20

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | |
| Interval | integer | 60 |
| PortType0 | integer | 0 |
| PortType1 | integer | 0 |
| PortType2 | integer | 0 |
| PortType3 | integer | 0 |
| Resolution0 | integer | 12 |
| Resolution1 | integer | 12 |
| Resolution2 | integer | 12 |
| Resolution3 | integer | 12 |
| Voltage0 | integer | 0 |
| Voltage1 | integer | 0 |
| Voltage2 | integer | 0 |
| Voltage3 | integer | 0 |

OneWire F26

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | |
| Interval | integer | 60 |

OneWire F28

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | |
| Interval | integer | 60 |
| Precision | integer | 0 |

OneWire F29

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | |
| Interval | integer | 60 |
| Invert0 | boolean | |
| Invert1 | boolean | |
| Invert2 | boolean | |
| Invert3 | boolean | |
| Invert4 | boolean | |
| Invert5 | boolean | |
| Invert6 | boolean | |
| Invert7 | boolean | |
| PortSettings0 | integer | 0 |
| PortSettings1 | integer | 0 |
| PortSettings2 | integer | 0 |
| PortSettings3 | integer | 0 |
| PortSettings4 | integer | 0 |
| PortSettings5 | integer | 0 |
| PortSettings6 | integer | 0 |
| PortSettings7 | integer | 0 |

OneWire F2C

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | |
| Interval | integer | 60 |
| UseCPC | boolean | |

OneWire F3A

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | |
| Interval | integer | 60 |
| Invert0 | boolean | |
| Invert1 | boolean | |
| PortSettings0 | integer | 0 |
| PortSettings1 | integer | 0 |

OneWire Gateway

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DelayTime | integer | 10 |
| GatewayMode | integer | 0 |

OZW Configurator

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

OZW DataPoint

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Id | integer | 0 |
| Interval | integer | 60 |
| WriteAccess | boolean | |

OZW Device

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| SerialNumber | string | |

OZW Gateway

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| IPAddress | string | 172.17.31.91 |
| Password | string | Kartoffel123! |
| Username | string | Administrator |
| UseSSL | boolean | 1 |

POP3

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| CacheInterval | integer | 300 |
| CacheSize | integer | 10 |
| Host | string | pop.gmail.com |
| Password | string | |
| Port | integer | 995 |
| UseAuthentication | boolean | 1 |
| Username | string | mail@gmail.com |
| UseSSL | boolean | 1 |
| VerifyHost | boolean | 1 |
| VerifyPeer | boolean | 1 |

Popup Module

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

Presence Control

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| LockTime | integer | 300 |
| TriggerVariables | string | [] |
| ValueVariables | string | [] |

### Register Variable

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| RXObjectID | integer | 0 |

### Serial Port

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| BaudRate | string | 9600 |
| DataBits | string | 8 |
| Open | boolean | |
| Parity | string | None |
| Port | string | |
| StopBits | string | 1 |

### Server Socket

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Certificate | string | |
| CertificateAuthority | string | |
| Limit | integer | 0 |
| Open | boolean | |
| Port | integer | 1024 |
| PrivateKey | string | |
| SilenceErrors | boolean | |
| UseSSL | boolean | |

### Shutter Control

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DriveDownTimeClose | double | 0 |
| DriveDownTimeDown | double | 0 |
| DriveDownTimeHalf | double | 0 |
| DriveUpTimeDown | double | 0 |
| DriveUpTimeHalf | double | 0 |
| DriveUpTimeOpen | double | 0 |
| MotorDelay | integer | 0 |
| ShutterType | integer | 0 |
| TargetID | integer | 0 |

### Shutter Control Legacy

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DriveDownTimeClose | double | 0 |
| DriveDownTimeDown | double | 0 |
| DriveDownTimeHalf | double | 0 |
| DriveUpTimeDown | double | 0 |
| DriveUpTimeHalf | double | 0 |
| DriveUpTimeOpen | double | 0 |
| HandlerScript | integer | 0 |
| MotorDelay | integer | 0 |
| ShutterType | integer | 0 |
| TransmitDevice | integer | 0 |
| TransmitDevice2 | integer | 0 |

Siemens Configurator

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DefaultInterval | integer | 5000 |
| PLCExport | string | |
| Type | integer | 1 |

Siemens Device

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Address | integer | 0 |
| Area | integer | 7 |
| AreaAddress | integer | 0 |
| Bit | integer | 0 |
| DataType | integer | 1 |
| EmulateStatus | boolean | 1 |
| Factor | double | 0 |
| Length | integer | 0 |
| Poller | integer | 0 |
| ReadOnly | boolean | |

Siemens Gateway

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| ClientTSAPHi | integer | 1 |
| ClientTSAPLo | integer | 2 |
| ServerTSAPHi | integer | 1 |
| ServerTSAPLo | integer | 2 |

Skin Control

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DefaultSkin | string | |

SMS

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| APIID | integer | 0 |
| Password | string | |
| Sender | string | |
| Username | string | |

SMS REST

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| APIKey | string | |

SMTP

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Host | string | smtp.gmail.com |
| Password | string | Salat15! |
| Port | integer | 587 |
| Recipient | string | pp@symcon.de |
| SenderAddress | string | pp@symcon.de |
| SenderName | string | Pascal Piotrowski |
| UseAuthentication | boolean | 1 |
| Username | string | pp@symcon.de |
| UseSSL | boolean | 1 |
| VerifyHost | boolean | 1 |
| VerifyPeer | boolean | 1 |

SSDP Control

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

SSE Client

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Active | boolean | |
| Headers | string | [] |
| URL | string | |
| VerifyHost | boolean | 1 |
| VerifyPeer | boolean | 1 |

Store Control

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

Text Parser

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Rules | string | [] |

Text To Speech

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| TTSAudioOutput | string | |
| TTSEngine | string | |

ThermoJet

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

TMEX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Open | boolean | |
| Port | integer | 1 |

UDP Socket

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| BindIP | string | 0.0.0.0 |
| BindPort | integer | 50001 |
| EnableBroadcast | boolean | 1 |
| EnableReuseAddress | boolean | 1 |
| Host | string | |
| Open | boolean | 1 |
| Port | integer | 50001 |

Universal RX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | string | 00000000 |
| RORG | integer | 165 |
| ScriptID | integer | 0 |

Util Control

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| EnableRequestLicenseData | boolean | |
| EnableSendUsageData | boolean | |
| LastCheckedVersion | string | |

UVR1611

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceCount | integer | 1 |
| Interval | integer | 0 |

Velleman USB

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

Virtual IO

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Open | boolean | |

VoIP

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| AcceptList | string | [] |
| Active | boolean | |
| AuthID | string | |
| DefaultAction | integer | 0 |
| Domain | string | |
| HandlerScript | integer | 0 |
| Password | string | |
| Username | string | |

WebFront Visualization

| EIGENSCHAFT | TYP | STANDA |
|---|---|---|
| AllowPasswordlessOnWAN | boolean | |
| AutoStart | string | [] |
| BooleanCheckBox | boolean | 1 |
| EnableMobile | boolean | 1 |
| GraphContinuousSpan | boolean | |
| GraphDynamicScaling | boolean | |
| GraphHighDensity | boolean | |
| GraphRawDensity | boolean | |
| GraphShowExtrema | boolean | |
| GraphShowLegend | boolean | |
| GraphTimeSpan | integer | 1 |
| IgnorePasswordOnLAN | boolean | |
| Items | string | [{"ClassName":"ClockWidget","Configuration":"","ID":"<br>{"ClassName":"Category","Configuration":"{\"baseID\":<br>\"Übersicht\"}","ID":"root","ParentID":"roottp","Positio<br>{"ClassName":"TabPane","Configuration":"{\"subTitle\"<br>\"IPS\"}","ID":"roottp","ParentID":"","Position":0,"Visibl<br>{"ClassName":"Category","Configuration":"{}","Positior |
| MobileID | integer | 0 |
| Nested | boolean | 1 |
| Password | string | |
| Skin | string | |

WebHook Control

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Hooks | string | [{"Hook":"\/hook\/notification-response\/0","TargetID":10196}] |

WebOAuth Control

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| ClientIDs | string | [] |

WebServer

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Active | boolean | 1 |
| AuthPassword | string | |
| AuthUsername | string | |
| Certificate | string | *removed* |
| CertificateAuthority | string | |
| DHParameters | string | *removed* |
| EnableBasicAuth | boolean | |
| EnableLogFile | boolean | |
| EnableSSL | boolean | |
| HomeDir | string | webfront |
| Port | integer | 82 |
| PrivateKey | string | *removed* |
| Server | string | 0.0.0.0 |

WinLIRC

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

WMRS200

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DeviceID | integer | 0 |
| DeviceType | integer | 0 |

WMRS200 Gateway

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

WS Client

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Active | boolean | |
| Headers | string | [] |
| URL | string | |
| VerifyCertificate | boolean | 1 |

WuT Counter

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Impulses | integer | 1000 |
| IONum | integer | 0 |

WuT Gateway

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| Interval | integer | 0 |
| IPAddress | string | |
| Password | string | |

WuT Input

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| IONum | integer | 0 |

WuT Output

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| IONum | integer | 0 |

WuT ThermoHygro

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| Interval | integer | 0 |
| IPAddress | string | |

XBee Gateway

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |

XBee Splitter

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DeviceID | integer | 0 |

xComfort Binary Input

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DataPoint | integer | 0 |

xComfort Configurator

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DPData | string | |

xComfort Dimmer

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DataPoint | integer | 0 |
| EmulateStatus | boolean | |

xComfort Energy

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DataPoint | integer | 0 |
| EnergyType | integer | 0 |

xComfort Gateway

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| GatewayMode | integer | 0 |

xComfort Heating

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DataPoint | integer | 0 |
| TemperatureVariableID | integer | 0 |

xComfort HRV

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DataPoint | integer | 0 |
| TemperatureVariableID | integer | 0 |

xComfort Humidity

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DataPoint | integer | 0 |

xComfort Impulse

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DataPoint | integer | 0 |
| Impulses | integer | 1000 |

xComfort One Channel Heating

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DataPoint | integer | 0 |
| TemperatureVariableID | integer | 0 |

xComfort Remote

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DataPoint | integer | 0 |

xComfort Room Control

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DataPoint | integer | 0 |

xComfort Shutter

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DataPoint | integer | 0 |
| EmulateStatus | boolean | |

xComfort Switch

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DataPoint | integer | 0 |
| EmulateStatus | boolean | |

xComfort Temperature

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DataPoint | integer | 0 |

xComfort Value RX

| EIGENSCHAFT | TYP | STANDARDWERT |
| --- | --- | --- |
| DataPoint | integer | 0 |
| ValueType | integer | 0 |

xComfort Value TX

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DataPoint | integer | 0 |
| ValueType | integer | 0 |

Z-Wave Configurator

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| SupportSecurity | boolean | 1 |
| UseNetworkWide | boolean | 1 |
| UseNormalPower | boolean | 1 |

Z-Wave Discovery

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|

Z-Wave Gateway

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| FailedThreshold | integer | 10 |
| GatewayMode | integer | 1 |
| OptimizeSchedule | integer | 1 |
| SendRetryCount | integer | 3 |
| SendRetryWait | integer | 500 |
| SendTimeout | integer | 1 |

Z-Wave Module

| EIGENSCHAFT | TYP | STANDARDWERT |
|---|---|---|
| DisableOptimize | boolean | |
| DisableTransmitQueue | boolean | |
| EnforceBasicClass | boolean | |
| EnforceSimpleAssociations | boolean | |
| Interval | integer | 0 |
| LockVariables | boolean | |
| MultiInstanceID | integer | 0 |
| NodeID | integer | 0 |

## BEISPIEL

```
IPS_SetProperty($id, "Open", true); //I/O Instanz soll aktiviert
sein.
IPS_ApplyChanges($id); //Neue Konfiguration übernehmen

// Zur Erstellung dieser Tabelle
<?php

$moduleIDs= IPS_GetModuleList();
$sortedModules = Array();
```

```
foreach($moduleIDs as $moduleID) {
$module = IPS_GetModule($moduleID);

//import only our buildin modules
if($module['LibraryID'] == "{0945206A-47AA-4FDD-9093-99051E410E82}")
$sortedModules[$module['ModuleName']] = $moduleID;
}

ksort($sortedModules, SORT_STRING | SORT_FLAG_CASE);

foreach($sortedModules as $moduleName => $moduleID) {
//create
$iids = IPS_GetInstanceListByModuleID($moduleID);
if (sizeof($iids) > 0) {
$iid = $iids[0];
}
else {
$iid = IPS_CreateInstance($moduleID);
}
$properties = json_decode(IPS_GetConfiguration($iid), true);
ksort($properties, SORT_STRING | SORT_FLAG_CASE);

//build text
echo "__" . $moduleName . "__" . PHP_EOL;
echo PHP_EOL;
echo "Eigenschaft | Typ | Standardwert" . PHP_EOL;
echo "-------------------------- | ------- | --------------------" .
PHP_EOL;
foreach($properties as $key => $value) {
if ($moduleName == "WebServer") {
switch($key) {
case "Certificate":
case "DHParameters":
case "PrivateKey":
$value = "_removed_";
break;
}
}
echo str_pad($key, 27) . "| " . str_pad(gettype($value), 8) . "| ".
$value . PHP_EOL;
}
echo PHP_EOL;
echo PHP_EOL;
//cleanup
$childrenIDs = IPS_GetChildrenIDs($iid);
foreach($childrenIDs as $childrenID) {
IPS_DeleteVariable($childrenID);
}

IPS_DeleteInstance($iid);
}
```

# Referenzen

| | |
|---|---|
| IPS_GetReferenceList | gibt die IDs aller Referenzen wieder |

# IPS_GetReferenceList

```
array IPS_GetReferenceList (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Ein Array von Integer-Werten aller IDs der referenzierten Objekte

## BESCHREIBUNG

Diese Funktion gibt ein Array der IDs aller referenzierten Objekte der Instanz mit der ID **InstanzID** wieder. Falls keine Referenz existiert, ist das Array leer. (z.B. das "Archive Control" referenziert alle Variablen dir dort geloggt werden)

## BEISPIEL

```
IPS_GetReferenceList(12345);
```

# StatusVariablen

Die Dokumentation dieser Funktion ist nur noch aus Kompatibilitätsgründen vorhanden und sollte ab IP-Symcon 2.5 nicht mehr verwendet werden. Diese Funktion wird ab Version 2.5 nur noch emuliert und sollte durch die entsprechenden Ersatzfunktionen ausgetauscht werden. Seit Version 4.0 kann diese Funktion nur noch genutzt werden, wenn der Spezialschalter "Kompatibilitätsfunktionen" aktiviert ist.

Ab Version 2.5 haben alle Objekte das Feld "Ident", welches über die Funktion IPS_GetObjectIDByIdent gesucht werden kann

Status Variablen sind normale IP-Symcon Variablen, die jedoch an eine Instanz fest gebunden sind. Status Variable sind als “Nur-Lesen” markiert, sodass diese nicht aus PHP Skripten heraus modifiziert werden können, um den Status als solchen nur vom jeweiligen Gerät änderbar zu machen.
Der Vorteil der Status Variablen ist, dass diese über die Instanz unter einem eindeutigen und nicht veränderbaren Namen ansprechbar sind. Das ist im Gegensatz zu den Objektnamen von großem Vorteil, da diese vom Benutzer änderbar sind und dadurch in jeder IP-Symcon Umgebung anders sein können.
Im Gegensatz dazu, können über die VariableIdents der Status Variablen die IDs der dazugehören IP-Symcon Variable erfragt werden, wodurch ein IP-Symcon umgebungsunabhängiger Zugriff möglich ist.
Bei einem Zugriff auf Variablen sollte die Funktion IPS_GetObjectIDByIdent verwendet werden.

# Verbindungen

| | |
|---|---|
| IPS_ConnectInstance | erstellt eine (Daten-)Verbindung zwischen zwei Instanzen |
| IPS_DisconnectInstance | löst eine (Daten-)Verbindung zwischen zwei Instanzen |
| IPS_GetCompatibleInstances | liefert kompatible Instanzen zu einer Instanz |
| IPS_IsInstanceCompatible | prüft, ob eine Verbindung zwischen zwei Instanzen möglich ist |

# IPS_ConnectInstance

```
boolean IPS_ConnectInstance (int $InstanzID, int $ConnectionID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| ConnectionID | ID der neu zu verbindenen Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion verbindet die Instanz **InstanzID** mit der Instanz **ConnectionID** von der Hardwaresicht (physikalisch) gesehen aus, so dass Daten ausgetauscht werden können. Beispiel wäre in FS20 Gerät, dass mit einem FHZ Splitter verbunden wird.



Jede Instanz kann nur eine verbundene Instanz haben.

## BEISPIEL

```
IPS_ConnectInstance(12345, 23456);
```

# IPS_DisconnectInstance

```
boolean IPS_DisconnectInstance (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Die Funktion löst eine verbundene Instanz **InstanzID** von der übergeordneten Instanz. Die Angabe der übergeordneten Instanz ist nicht notwendig, da es nur eine geben kann und somit die zu lösende Verbindung eindeutig ist.

## BEISPIEL

```
IPS_DisconnectInstance(12345);
```

# IPS_GetCompatibleInstances

```
boolean IPS_GetCompatibleInstances (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der zu prüfenden Instanz |

## RÜCKGABEWERT

Array von Integer-Werten aller kompatiblen InstanzIDs zur **InstanzID**

## BESCHREIBUNG

Die Funktion ermittelt die IDs aller in IP-Symcon verfügbaren Instanzen, die zur **InstanzID** kompatibel sind. Die IDs werden in einem Array aufgelistet.

## BEISPIEL

```
print_r(IPS_GetCompatibleInstances(12345));

/* liefert z.B.:
Array
(
[0] => 22222
[1] => 33333
[2] => 44444
[3] => 55555
usw. ...
)
*/
```

# IPS_IsInstanceCompatible

```
boolean IPS_IsInstanceCompatible (int $InstanzID, int
$ÜbergeordneteID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| ÜbergeordneteID | ID der potentiellen übergeordneten Instanz |

## RÜCKGABEWERT

TRUE, wenn die Instanz kompatibel sind, sonst FALSE

## BESCHREIBUNG

Die Funktion ermittelt, ob die Instanz mit der ID **ÜbergeordneteID** eine valide übergeordnete Instanz zu der Instanz mit der ID **InstanzID** ist.

## BEISPIEL

```
if (IPS_IsInstanceCompatible(12345, 23456))
echo "Instanz ist zur anderen Instanz kompatibel!";
```

# Kategorieverwaltung

| | |
|---|---|
| IPS_CategoryExists | prüft, ob eine Kategorie bereits existiert |
| IPS_CreateCategory | legt eine neue Kategorie an |
| IPS_DeleteCategory | löscht eine bestehende Kategorie |
| IPS_GetCategoryIDByName | sucht und liefert die ID einer bestehenden Kategorie über dessen Namen |
| IPS_GetCategoryList | liefert eine Liste aller existierenden Kategorien |

# IPS_CategoryExists

```
boolean IPS_CategoryExists (int $KategorieID)
```

## PARAMETERLISTE

| | |
|---|---|
| KategorieID | ID der zu prüfenden Kategorie |

## RÜCKGABEWERT

Falls die KategorieID im System existiert, wird **TRUE** zurückgemeldet, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion prüft, ob die Kategorie mit der ID **KategorienID** existiert.

## BEISPIEL

```
if (IPS_CategoryExists(45724))
echo "Kategorie existiert bereits!";
```

# IPS_CreateCategory

```
int IPS_CreateCategory ()
```

## RÜCKGABEWERT

ID der neu erstellten Kategorie

## BESCHREIBUNG

Die Funktion legt eine neue Kategorie an. Es wird kein Parameter benötigt. Nach Ausführung der Funktion erscheint im Kategorienbaum von IP-Symcon eine neue Kategorie mit der Bezeichnung z.B. ”Unnamed Object (ID: 48490)”. Mit Hilfe der Funktion IPS_SetName kann der Kategorie ein aussagefähiger Name gegeben werden. Der Name sollte nicht für die Identifikation verwendet werden. Hierfür ist die ID zuständig.
Mit der Funktion IPS_SetParent kann die Kategorie einem anderen Objekt untergeordnet werden und stellt so eine Unterkategorie dar.
Die Funktion liefert eine ID, mit deren Hilfe die erzeugte Kategorie eindeutig identifiziert werden kann.

## BEISPIEL

```
// Anlegen einer neuen Kategorie mit dem namen "Regenerfassung"
$CatID = IPS_CreateCategory(); // Kategorie anlegen
IPS_SetName($CatID, "Regenerfassung"); // Kategorie benennen
IPS_SetParent($CatID, 12345); // Kategorie einsortieren unter dem
Objekt mit der ID "12345"
```

# IPS_DeleteCategory

```
boolean IPS_DeleteCategory (int $KategorieID)
```

## PARAMETERLISTE

| | |
|---|---|
| KategorieID | ID der zu löschenden Kategorie |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion löscht die vorhandene Kategorie mit der ID **KategorienID**.



Die Kategorie darf nur gelöscht werden, wenn keine Unterobjekte mehr vorhanden sind.

## BEISPIEL

```
// Löschen der Kategorie 47788
IPS_DeleteCategory(47788);
```

# IPS_GetCategoryIDByName

```
int IPS_GetCategoryIDByName (string $KategorieName, int
$ÜbergeordneteID)
```

## PARAMETERLISTE

| | |
|---|---|
| KategorieName | Zu suchender KategorieName |
| ÜbergeordneteID | Objekt, in deren direkten Unterobjekten gesucht werden soll |

## RÜCKGABEWERT

ID der gefundenen Kategorie, sonst FALSE

## BESCHREIBUNG

Die Funktion versucht die ID der Kategorie mit dem Namen **KategorienName**, deren übergeordnetes Objekt die ID **ÜbergeordneteID** besitzt, zu ermitteln. Es wird die ID der ersten gefundenen Kategorie verwendet, deren Name mit **KategorienName** übereinstimmt. Falls eine solche Kategorie gefunden wurde, wird deren ID zurückgemeldet, andernfalls FALSE.

**Vorsicht:** Da Namen in IP-Symcon beliebig vergeben werden können, und damit **nicht eindeutig** sind, kann die Funktion eine falsche ID zurückgeben.

Diese Funktion generiert eine **Warnung**. Sollte der Name nicht gefunden werden, benutzen Sie den @ Operator, falls Sie diese Meldung unterdrücken wollen. Die genau Funktionsweise dieses Operators können Sie dem PHP Handbuch entnehmen.

## BEISPIEL

```
$KategorieID = @IPS_GetCategoryIDByName("Regenerfassung",
$ÜbergeordneteID);
if ($KategorieID === false)
echo "Kategorie nicht gefunden!";
else
echo "Die Kategorien-ID lautet: ". $KategorieID;
```

# IPS_GetCategoryList

```
array IPS_GetCategoryList ()
```

## RÜCKGABEWERT

Ein Array von Integer-Werten aller IDs der Kategorien in IP-Symcon

## BESCHREIBUNG

Die Funktion ermittelt die IDs aller in IP-Symcon eingerichteten Kategorien. Die IDs werden in einem Array aufgelistet. Falls keine Kategorie existiert, ist das Array leer.

## BEISPIEL

```
$alleKategorien = IPS_GetCategoryList();
print_r($alleKategorien);

/* liefert z.B.:
Array
(
[0] => 0
[1] => 37659
[2] => 18326
usw. ...
)
*/
```

# Linkverwaltung

| | |
|---|---|
| IPS_CreateLink | legt einen neuen Link an |
| IPS_DeleteLink | löscht einen bestehenden Link |
| IPS_GetLink | liefert umfassende Informationen über einen Link |
| IPS_GetLinkIDByName | ermittelt die ID eines Links anhand des Namens |
| IPS_GetLinkList | liefert eine Liste aller existierenden Links |
| IPS_LinkExists | prüft, ob ein Link bereits existiert |
| IPS_SetLinkTargetID | setzt die Verknüpfung für den Link |

# IPS_CreateLink

Benötigt: IP-Symcon >= 2.1

```
int IPS_CreateLink ()
```

## RÜCKGABEWERT

ID des neu erstellten Links

## BESCHREIBUNG

Die Funktion legt eine neuen Link an. Er benötigt keine Parameter. Nach Ausführung der Funktion erscheint im Kategorienbaum von IP-Symcon ein neues Objekt mit der Bezeichnung z.B. ”Unnamed Object (ID: 48490)”. Mit Hilfe der Funktion IPS_SetName kann dem Objekt (hier der Link) ein aussagefähiger Name gegeben werden. Der Name ist jedoch für die Identifikation irrelevant. Hierfür ist immer die ID zuständig.
Weiterhin sollte der Link mit einem anderen Objekt verknüpft werden. Dies kann über die Funktion IPS_SetLinkTargetID gemacht werden.
Die Funktion liefert eine ID, mit deren Hilfe der erzeugte Link eindeutig identifiziert werden kann.

## BEISPIEL

```
// Anlegen eines neuen Links mit dem Namen "Regenerfassung"
$LinkID = IPS_CreateLink(); // Link anlegen
IPS_SetName($LinkID, "Regenerfassung"); // Link benennen
IPS_SetParent($LinkID, 12345); // Link einsortieren unter dem Objekt
mit der ID "12345"
IPS_SetLinkTargetID($LinkID, 54321); // Link verknüpfen
```

# IPS_DeleteLink

Benötigt: IP-Symcon >= 2.1

```
boolean IPS_DeleteLink (int $LinkID)
```

## PARAMETERLISTE

| | |
|---|---|
| LinkID | ID des zu löschenden Links |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion löscht den Link mit der ID **LinkID**.



Der Link darf nur gelöscht werden, wenn keine Unterobjekte mehr vorhanden sind.

## BEISPIEL

```
//Löschen des Links 47788
IPS_DeleteLink(47788);
```

# IPS_GetLink

Benötigt: IP-Symcon >= 2.1

```
array IPS_GetLink (int $LinkID)
```

## PARAMETERLISTE

| | |
|---|---|
| LinkID | Die zu zeigende LinkID |

## RÜCKGABEWERT

Folgende Informationen stehen als **key => value** Paare zur verfügung:

| INDEX | TYP | BESCHREIBUNG |
|---|---|---|
| **LinkID** | integer | ObjektID |
| **LinkChildID** (bis 3.4) | integer | ID des Objektes mit dem dieser Link verknüpft ist. Siehe IPS_SetLinkTargetID (ab 4.0 dauerhaft ersetzt durch TargetID) |
| **TargetID** (ab 2.6) | integer | ID des Objektes mit dem dieser Link verknüpft ist. Siehe IPS_SetLinkTargetID |

## BESCHREIBUNG

Die Funktion liefert ein Array mit Informationen über den Link mit der ID **LinkID**. Sollte der Link nicht vorhanden sein, wird eine Warnung generiert.



Ab Version 4.0 wird die bis 3.4 vorhandene Funktion als "array IPS_GetLinkCompatibility( integer $LinkID )" zur Verfügung gestellt.

## BEISPIEL

```
// Ab Version 4.0
print_r(IPS_GetLink(19668));
/* Beispielausgabe:
Array
(
[LinkID] => 19668
[TargetID] => 14444
)
*/
```

```
print_r(IPS_GetLinkCompatibility(19668));
/* Beispielausgabe:
Array
(
[LinkID] => 19668
[LinkChildID] => 14444
[TargetID] => 14444
)
*/


// Bis Version 3.4
print_r(IPS_GetLink(19668));
/* Beispielausgabe:
Array
(
[LinkID] => 19668
[LinkChildID] => 14444
[TargetID] => 14444
)
*/

// Bis Version 2.5
print_r(IPS_GetLink(19668));
/* Beispielausgabe:
Array
(
[LinkID] => 19668
[LinkChildID] => 14444
)
*/
```

# IPS_GetLinkIDByName

Benötigt: IP-Symcon >= 2.1

```
int IPS_GetLinkIDByName (string $LinkName, int $ÜbergeordneteID)
```

## PARAMETERLISTE

| | |
|---|---|
| LinkName | Zu suchender LinkName |
| ÜbergeordneteID | Objekt, in deren direkten Unterobjekten gesucht werden soll |

## RÜCKGABEWERT

ID des gefundenen Links, sonst FALSE

## BESCHREIBUNG

Die Funktion versucht die ID des Links mit dem Namen **LinkName**, deren übergeordnetes Objekt die ID **ÜbergeordneteID** besitzt, zu ermitteln. Es wird die ID des ersten gefundenen Links verwendet, deren Name mit **LinkName** übereinstimmt. Falls ein solcher Link gefunden wurde, wird deren ID zurückgemeldet, andernfalls FALSE.

**Vorsicht:** Da Namen in IP-Symcon beliebig vergeben werden können, und damit **nicht eindeutig** sind, kann die Funktion eine falsche ID zurückmelden.

Diese Funktion generiert eine **Warnung**, sollte der Name nicht gefunden werden. Sollte der Name nicht gefunden werden, muss der @ Operator genutzt, falls diese Meldung unterdrückt werden soll. Die genau Funktionsweise dieses Operators kann dem PHP Handbuch entnommen werden.

## BEISPIEL

```
$LinkID = @IPS_GetLinkIDByName("Verlinkte Regenerfassung",
$ÜbergeordneteID);
if ($LinkID === false)
echo "Link nicht gefunden!";
else
```

```
echo "Die Link-ID lautet: ". $LinkID;
```

# IPS_GetLinkList

Benötigt: IP-Symcon >= 2.1

```
array IPS_GetLinkList ()
```

## RÜCKGABEWERT

Ein Array von Integer-Werten aller IDs der Links in IP-Symcon

## BESCHREIBUNG

Die Funktion ermittelt die IDs aller in IP-Symcon eingerichteten Links . Die IDs werden in einem Array aufgelistet. Falls kein Link existiert, ist das Array leer.

## BEISPIEL

```
$alleLinks = IPS_GetLinkList();
print_r($alleLinks);

/* liefert z.B.:
Array
(
[0] => 0
[1] => 37659
[2] => 18326
usw. ...
)
*/
```

# IPS_LinkExists

Benötigt: IP-Symcon >= 2.1

```
boolean IPS_LinkExists (int $LinkID)
```

## PARAMETERLISTE

| | |
|---|---|
| LinkID | ID der zu prüfenden Links |

## RÜCKGABEWERT

Falls die LinkID im System existiert, wird **TRUE** zurückgemeldet, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion prüft, ob der Link mit der ID **LinkID** existiert.

## BEISPIEL

```
if (IPS_LinkExists(45724))
echo "Link existiert bereits!";
```

# IPS_SetLinkTargetID

Benötigt: IP-Symcon >= 2.6

```
boolean IPS_SetLinkTargetID (int $LinkID, int $VerknüpftesObjekt)
```

## PARAMETERLISTE

| | |
|---|---|
| LinkID | ID des zu verändernden Links |
| VerknüpftesObjekt | ID des zu verknüpfenden Objekts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion verknüpft ein bestimmtes Objekt **VerknüpftesObjekt** mit dem Link **LinkID**, sodass der Link auf dieses Objekt zeigt.

## BEISPIEL

```
IPS_SetLinkTargetID($LinkID, 12345); //Auf Objekt 12345 verweisen
```

# Medienverwaltung

| | |
|---|---|
| IPS_CreateMedia | legt ein neues Medienobjekt im MedienPool an |
| IPS_DeleteMedia | löscht ein Medienobjekt aus dem MedienPool |
| IPS_GetMedia | liefert umfassende Information über ein bestimmtes Medienobjekt |
| IPS_GetMediaContent | liefert den Inhalt eines Medienobjekts |
| IPS_GetMediaID | ermittelt die ID eines Medienobjekts |
| IPS_GetMediaIDByFile | ermittelt die ID eines Medienobjekts anhand der Datei |
| IPS_GetMediaIDByName | ermittelt die ID eines Medienobjekts anhand des Namens |
| IPS_GetMediaList | liefert eine Liste aller Medienobjekte im MedienPool |
| IPS_GetMediaListByType | liefert eine Liste aller Medienobjekte eines bestimmten Typs |
| IPS_MediaExists | prüft, ob sich ein bestimmtes Medienobjekt bereits im MedienPool befindet |
| IPS_SendMediaEvent | sendet eine Mitteilung, dass ein Medienobjekt sich geändert hat |
| IPS_SetMediaCached | aktiviert, dass ein Medienobjekt nur im Arbeitsspeicher bearbeitet wird |
| IPS_SetMediaContent | setzt den Inhalt eines Medienobjekts |
| IPS_SetMediaFile | bindet eine Mediendatei an ein Medienobjekt |

# IPS_CreateMedia

```
int IPS_CreateMedia (int $MedienTyp)
```

## PARAMETERLISTE

| WERT | BESCHREIBUNG |
|---|---|
| 0 | Legt ein **IPSView Formular** an |
| 1 | Legt ein **Image Objekt** an |
| 2 | Legt ein **Sound Objekt** an |
| 3 | Legt ein **Stream Objekt** an |
| 4 | Legt ein **Chart Objekt** an |
| 5 | Legt ein **Dokument Objekt** an |

MedienTyp

## RÜCKGABEWERT

ID des neu erstellten Medienobjekts

## BESCHREIBUNG

Die Funktion legt ein leeres Medienobjekt vom Typ **MedienTyp** an.
Die Funktion liefert eine ID, mit deren Hilfe das erzeugte Medienobjekt eindeutig identifiziert werden kann. Das erzeugte Objekt ist nach dem Anlegen noch leer und muss mit der Funktion IPS_SetMediaFile noch mit einer realen Mediendatei verbunden werden.

## BEISPIEL

```
// Anlegen eines neuen Links mit dem Namen "Alarm"
$ImageFile = "C:\\Bilder\\Alarmsymbol.png"; // Image-Datei
$MediaID = IPS_CreateMedia(1); // Image im MedienPool anlegen
IPS_SetMediaFile($MediaID, $ImageFile, true); // Image im MedienPool
mit Image-Datei verbinden
IPS_SetName($MediaID, "Alarm"); // Medienobjekt benennen
IPS_SetParent($MediaID, 12345); // Medienobjekt einsortieren unter
dem Objekt mit der ID "12345"
```

# IPS_DeleteMedia

```
boolean IPS_DeleteMedia (int $MedienID, boolean $DateiLöschen)
```

## PARAMETERLISTE

| | |
|---|---|
| MedienID | ID des zu löschenden Medienobjekts |
| DateiLöschen | TRUE, wenn Datei gelöscht werden soll; FALSE, wenn die Datei in den 'deleted' Ordner verschoben werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Die Funktion entfernt das Medienobjekt mit der ID **MedienID** aus dem MedienPool.



Das Medienobjekt darf nur gelöscht werden, wenn keine Unterobjekte mehr vorhanden sind.

## BEISPIEL

```
IPS_DeleteMedia($MediaID, true);
```

# IPS_GetMedia

```
array IPS_GetMedia (int $MedienID)
```

## PARAMETERLISTE

| | |
|---|---|
| MedienID | Die zu zeigende MedienID |

## RÜCKGABEWERT

Folgende Informationen stehen als **key => value** Paare zur Verfügung:

| INDEX | TYP | BESCHREIBUNG |
|---|---|---|
| **IsAvailable** (bis 3.4) | boolean | TRUE, wenn MediaFile vorhanden (ab Version 4.0 ersetzt durch MediaIsAvailable) |
| **IsLinked** (bis 3.4) | boolean | TRUE, wenn Datei außerhalb des Media Ordners liegt (entfällt ab Version 4.0) |
| **LastUpdate** (bis 3.4) | integer | Unix Timestamp der letzten Aktualisierung (ab Version 4.0 ersetzt durch MediaUpdated) |
| **MediaCRC** | string | CRC32 der Datei |
| **MediaFile** | string | Pfad zur Datei |
| **MediaID** | integer | MediaID |
| **MediaIsAvailable** (ab 4.0) | boolean | TRUE, wenn MediaFile vorhanden |
| **MediaIsCached** (ab 4.0) | boolean | TRUE, wenn das Medienobjekt in den Arbeitsspeicher geladen werden soll. Siehe IPS_SetMediaCached |
| **MediaSize** | integer | Größe in Byte |
| **MediaType** | integer | Medien-Typ (0: Formular, 1: Bild, 2: Ton, 3: Stream, 4: Chart, 5: Dokument) |
| **MediaUpdated** (ab 4.0) | integer | Unix Timestamp der letzten Aktualisierung |
| **SendEvent** (bis 3.4) | boolean | TRUE, wenn automatisch Dateiänderungen versendet werden sollen (entfällt ab Version 4.0) |

## BESCHREIBUNG

Die Funktion liefert ein Array mit Informationen über Medien mit der ID **MedienID**.
(info: Ab Version 4.0 wird die bis 3.4 vorhandene Funktion als
"array IPS_GetMediaCompatibility( integer $MedienID )" zur Verfügung gestellt.[/i

## BEISPIEL

```
// Ab Version 4.0
print_r(IPS_GetMedia(45699));
/* Beispielausgabe:
Array
(
[MediaCRC] => E2D2C1D1
[MediaFile] => media\45699.bin
[MediaID] => 45699
[MediaIsAvailable] => 1
[MediaIsCached] =>
[MediaSize] => 8192
[MediaType] => 0
[MediaUpdated] => 1214421546
)
*/

print_r(IPS_GetMediaCompatibility(45699));
/* Beispielausgabe:
Array
(
[IsAvailable] => 1
[IsLinked] =>
[LastUpdate] => 1214421546
[MediaCRC] => E2D2C1D1
[MediaFile] => media\45699.bin
[MediaID] => 45699
[MediaSize] => 8192
[MediaType] => 0
[SendEvent] =>
)
*/



// Bis Version 3.4
print_r(IPS_GetMedia(45699));
/* Beispielausgabe:
Array
(
[IsAvailable] => 1
[IsLinked] =>
[LastUpdate] => 1214421546
[MediaCRC] => E2D2C1D1
[MediaFile] => media\45699.bin
[MediaID] => 45699
[MediaSize] => 8192
[MediaType] => 0
[SendEvent] =>
)
*/
```

# IPS_GetMediaContent

Benötigt: IP-Symcon >= 3.1

```
string IPS_GetMediaContent (int $MedienID)
```

## PARAMETERLISTE

| | |
|---|---|
| MedienID | ID des Medienobjekts, dessen Inhalt geliefert werden soll |

## RÜCKGABEWERT

Der Inhalt des Medienobjekts in Base64 kodiert

## BESCHREIBUNG

Diese Funktion liefert den Base64 kodierten Inhalt für das Medienobjekt mit der ID **MedienID**. Dabei wird der Inhalt aus der im Medienobjekt definierten Datei ausgelesen.



Ein praktisches Beispiel für diese Funktion ist das Einlesen eines Bildes in eine externe Applikation, welche z.B. über die JSON-RPC API keinen Zugriff auf das Dateisystem hat.

## BEISPIEL

```
$Inhalt = base64_decode(IPS_GetMediaContent($MediaID));
```

# IPS_GetMediaID

```
int IPS_GetMediaID (string $MedienName)
```

## PARAMETERLISTE

| | |
|---|---|
| MedienName | Zu suchender Medienname |

## RÜCKGABEWERT

ID des gefundenen Medienobjekts, sonst 0

## BESCHREIBUNG

Die Funktion versucht, die ID des Medienobjekts mit dem Namen **MedienName** zu ermitteln. Es wird die ID des ersten gefundenen Medienobjekts verwendet, dessen Name mit **MedienName** übereinstimmt. Falls ein solches Medienobjekt gefunden wurde, wird dessen ID zurückgemeldet, andernfalls 0 (Null).
Eine sicherere Methode, die ID zu bestimmen, bietet die Funktion IPS_GetMediaIDByName.



**Vorsicht:** Da Namen in IP-Symcon beliebig vergeben werden können, und damit **nicht eindeutig** sind, kann die Funktion eine falsche ID zurückmelden.

## BEISPIEL

```
$MediaID = IPS_GetMediaID("Regenmenge");
if ($MediaID == 0)
echo "Medienobjekt nicht gefunden!";
else
echo "Die Medienobjekt-ID lautet: ". $MediaID;
```

# IPS_GetMediaIDByFile

```
int IPS_GetMediaIDByFile (string $MedienPfad)
```

## PARAMETERLISTE

| | |
|---|---|
| MedienPfad | Relativer MedienPfad vom KernelDir gesehen |

## RÜCKGABEWERT

ID des gefundenen Medienobjekts, sonst 0 und Warning

## BESCHREIBUNG

Die Funktion versucht, die ID von Medien mit dem Pfad **MedienPfad** zu finden. Sollte der MediePfad in IP-Symcon nicht vorhanden sein, wird eine Warnung generiert. Diese kann mit dem @ Operator abgefangen werden.

## BEISPIEL

```
$MediaID = @IPS_GetMediaIDByFile("media\\help.png");
if ($MediaID == 0)
echo "Bild nicht gefunden!";
else
echo "Die Media-ID lautet: ". $MediaID;
```

# IPS_GetMediaIDByName

```
int IPS_GetMediaIDByName (string $MedienName, int $ÜbergeordneteID)
```

## PARAMETERLISTE

| | |
|---|---|
| MedienName | Zu suchender MedienName |
| ÜbergeordneteID | Objekt, in deren direkten Unterobjekten gesucht werden soll |

## RÜCKGABEWERT

ID des gefundenen Medienobjekts, sonst FALSE

## BESCHREIBUNG

Die Funktion versucht die ID von Median mit dem Namen **MediaName**, dessen übergeordnetes Objekt die ID **ÜbergeordneteID** besitzt, zu ermitteln. Es wird die ID des ersten gefundenen Medien verwendet, dessen Name mit **MediaName** übereinstimmt. Falls ein solches Medium gefunden wurde, wird dessen ID zurückgemeldet, andernfalls FALSE.
Da hier im Gegensatz zu IPS_GetMediaID ein zusätzliches Kriterium (ÜbergeordneteID) bei der Suche verwendet wird, ist die Wahrscheinlichkeit die richtige ID zu finden erheblich größer. Trotzdem gilt:



**Vorsicht:** Da Namen in IP-Symcon beliebig vergeben werden können, und damit **nicht eindeutig** sind, kann die Funktion eine falsche ID zurückmelden.



Diese Funktion generiert eine **Warnung**, sollte der Name nicht gefunden werden. Benutzen Sie den @ Operator, falls Sie diese Meldung unterdrücken wollen. Die genau Funktionsweise dieses Operators können Sie dem PHP Handbuch entnehmen.

## BEISPIEL

```
$MedienID = @IPS_GetMediaIDByName("MeinBild", $ÜbergeordneteID);
if ($MedienID === false)
echo "Bild nicht gefunden!";
else
```

```
echo "Die Media-ID lautet: ". $MedienID;
```

# IPS_GetMediaList

```
array IPS_GetMediaList ()
```

## RÜCKGABEWERT

Ein Array von Integer-Werten aller IDs der Medienobjekte in IP-Symcon

## BESCHREIBUNG

Die Funktion ermittelt die IDs aller im MedienPool registrierten Medienobjekte. Die IDs werden in einem Array aufgelistet. Falls kein Medienobjekt existiert, ist das Array leer.
Es werden Objekte aller Medientypen aufgelistet. Sollen nur Objekte eines bestimmten Medientyps berücksichtigt werden, so kann hierfür die Funktion IPS_GetMediaListByType verwendet werden.

## BEISPIEL

```
$alleMedienobjekte = IPS_GetMediaList();
print_r($alleMedienobjekte);

/* liefert z.B.:
Array
(
[0] => 37659
[1] => 18326
usw. ...
*/
```

# IPS_GetMediaListByType

```
array IPS_GetMediaListByType (int $MedienTyp)
```

## PARAMETERLISTE

| WERT | BESCHREIBUNG |
|---|---|
| 0 | IPSView Formular |
| 1 | Image Objekt |
| 2 | Sound Objekt |
| 3 | Stream Objekt |
| 4 | Chart Objekt |
| 5 | Dokument Objekt |

MedienTyp

## RÜCKGABEWERT

Ein Array von Integer-Werten aller IDs der Medienobjekt vom Typ **MedienTyp** in IP-Symcon

## BESCHREIBUNG

Die Funktion ermittelt die IDs aller im MedienPool registrierten Medienobjekte eines bestimmten **MedienTyps**. Die IDs werden in einem Array aufgelistet. Falls kein Medienobjekt des angeforderten Typs existiert, ist das Array leer.
Es werden nur Objekte des Medientyps **MedienTyp** aufgelistet. Sollen alle Medienobjekte aufgelistet werden, so kann hierfür die Funktion IPS_GetMediaList verwendet werden.

## BEISPIEL

```
$alleImageObjekte = IPS_GetMediaListByType(1); // nur Image-Objekte
auflisten
print_r($alleImageObjekte);

/* liefert z.B.:
Array
(
[0] => 37659
[1] => 18326
usw. ...
)
*/
```

# IPS_MediaExists

```
boolean IPS_MediaExists (int $MedienID)
```

## PARAMETERLISTE

| | |
|---|---|
| MedienID | ID des zu rüfenden Medienobjekts |

## RÜCKGABEWERT

Falls die MedienID im System existiert, wird **TRUE** zurückgemeldet, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion prüft, ob das Medienobjekt mit der ID **MediaID** existiert.

## BEISPIEL

```
if (IPS_MediaExists(34881))
echo "Ein Medienobjekt mit dieser ID existiert!";
```

# IPS_SendMediaEvent

```
boolean IPS_SendMediaEvent (int $MedienID)
```

## PARAMETERLISTE

| | |
|---|---|
| MedienID | ID des zu prüfenden Medienobjekts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Die Funktion sendet eine Mitteilung, dass sich das Medienobjekt mit der ID MedienID verändert hat. Diese Nachricht wird z.B. verwendet, um ein Bild/Forumular zu aktualisieren.

## BEISPIEL

```
IPS_SendMediaEvent(12345);
```

# IPS_SetMediaCached

```
boolean IPS_SetMediaCached (int $MedienID, boolean $CacheAktiviert)
```

## PARAMETERLISTE

| | |
|---|---|
| MedienID | ID des zu verändernden Medienobjekts |
| CacheAktiviert | TRUE, wenn das Medienobjekt nur im Cache verarbeitet werden soll, sonst FALSE |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Die Funktion aktiviert das Cachen für ein Medienobjekt mit der ID MedienID. Das Medienobjekt wird im Arbeitsspeicher verändert wenn die Funktionen IPS_SetMediaContent und IPS_GetMediaContent genutzt werden. Beim Herunterfahren von IP-Symcon wird das Medienobjekt automatisch auf den physikalischen Speicher geschrieben und somit persistent.



Diese Funktion dient zur Schonung der SSD's und Flashspeicher.

## BEISPIEL

```
$MediaID = IPS_CreateMedia(1); // Image im MedienPool anlegen
IPS_SetMediaCached($MediaID, true);
// Das Cachen für das Mediaobjekt wird aktiviert.
// Beim ersten Zugriff wird dieses von der Festplatte ausgelesen
// und zukünftig nur noch im Arbeitsspeicher verarbeitet.
```

# IPS_SetMediaContent

Benötigt: IP-Symcon >= 3.1

```
boolean IPS_SetMediaContent (int $MedienID, string $Inhalt)
```

## PARAMETERLISTE

| | |
|---|---|
| MedienID | ID des Medienobjekts, dem der Inhalt zugeordnet werden soll |
| Inhalt | Base64 kodierter Inhalt |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion setzt den Inhalt für das Medienobjekt mit der ID **MedienID**. Dabei wird der Dateiname, der vorher per IPS_SetMediaFile definiert wurde, genutzt.
Es wird eine Fehlermeldung zurückgegeben, sofern kein gültiger Dateiname vorhanden ist, oder der Inhalt nicht korrekt Base64 kodiert wurde.



Diese Funktion ist mit Vorsicht zu verwenden, da der Inhalt ohne Warnung überschrieben wird.

## BEISPIEL

```
$MediaID = IPS_CreateMedia(1); //Bild hochladen
IPS_SetMediaFile($MediaID, "BlackPixel.gif", False);
IPS_SetMediaContent($MediaID,
"R0lGODlhAQABAIAAAAUEBAAAACwAAAAAAQABAAACAkQBADs=");
```

# IPS_SetMediaFile

```
boolean IPS_SetMediaFile (int $MedienID, string $DateiName, boolean
$DateiMussExistieren)
```

## PARAMETERLISTE

| | |
|---|---|
| MedienID | ID des zu verändernden Medienobjekts |
| DateiName | Dateiname/-pfad der zu verknüpfenden Datei |
| DateiMussExistieren | TRUE, wenn Existenz geprüft werden soll, sonst FALSE |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Die Funktion bindet den Dateinamen **DateiName** an das Medienobjekt mit der ID **MedienID**. Der Parameter **DateiMussExistieren** gibt an, ob intern geprüft werden soll, ob die Datei vorhanden ist. Die zuzuordnende Mediendatei muss zum Medientyp des Medienobjekts passen.



Der DateiName muss eindeutig sein. Wenn der DateiName bereits einem anderen Medienobjekt zugehörig ist, wird ein Fehler zurückgegeben.



Es ist unbedingt notwendig, dass sich die Dateien innerhalb des Hauptordners “IP-Symcon” oder einem Unterordner befinden. Aus Sicherheitsgründen sind andere Verzeichnisse nicht zulässig. Dies gilt nicht für das Hinzufügen von Dateien, sondern einzig für den Zugriff.

## BEISPIEL

```
$ImageFile = "Alarmsymbol.png"; // Image-Datei
$MediaID = IPS_CreateMedia(1); // Image im MedienPool anlegen
IPS_SetMediaFile($MediaID, $ImageFile, true); // Image im MedienPool
mit Image-Datei verbinden
```

# Modulverwaltung

| | |
|---|---|
| IPS_GetCompatibleModules | liefert alle Kompatiblen Module zu einem bestimmten Modul |
| IPS_GetLibrary | gibt Details zu einer Bibliothek |
| IPS_GetLibraryList | liefert eine Liste der Bibliotheken |
| IPS_GetLibraryModules | gibt eine Liste der Module einer Bibliothek |
| IPS_GetModule | gibt Details zu einem Modul |
| IPS_GetModuleList | gibt eine Liste aller Module |
| IPS_GetModuleListByType | gibt eine Liste aller Module des typs |
| IPS_IsModuleCompatible | überprüft, ob zwei Modul miteinander Kompatibel sind |
| IPS_LibraryExists | überprüft, ob Bibliothek existiert |
| IPS_ModuleExists | überprüft, ob Modul existiert |

# IPS_GetCompatibleModules

```
array IPS_GetCompatibleModules (string $ModulID)
```

## PARAMETERLISTE

| | |
|---|---|
| ModulID | GUID |

## RÜCKGABEWERT

Array von String-Werten aller kompatiblen ModulIDs zu **ModulID**

## BESCHREIBUNG

Die Funktion ermittelt die GUIDs aller in IP-Symcon verfügbaren Module, die zur **ModulID** kompatibel sind. Die IDs werden in einem Array aufgelistet.

## BEISPIEL

```
print_r(IPS_GetCompatibleModules("{57040540-4432-4220-8D2D-
4676B57E223D}"));

/* liefert z.B.:
Array
(
[0] => {4CB91589-CE01-4700-906F-26320EFCF6C4}
[1] => {6DC3D946-0D31-450F-A8C6-C42DB8D7D4F1}
[2] => {3CFF0FD9-E306-41DB-9B5A-9D06D38576C3}
[3] => {8062CF2B-600E-41D6-AD4B-1BA66C32D6ED}
usw. ...
)
*/
```

# IPS_GetLibrary

```
array IPS_GetLibrary (string $BibliothekID)
```

## PARAMETERLISTE

| | |
|---|---|
| BibliothekID | Die zu zeigende BibliothekID (GUIDs) |

## RÜCKGABEWERT

Folgende Informationen stehen als **key => value** Paare zur Verfügung:

| INDEX | TYP | BESCHREIBUNG |
|---|---|---|
| Author | string | Name des Autors |
| Build | integer | Fortlaufende Zahl zur Versionierung |
| Date | integer | Unix-Timestamp |
| LibraryID | string | BibliothekID (GUID) |
| Name | string | Name der Bibliothek |
| URL | string | Internetadresse zur Bibliothek |
| Version | integer | HighByte: Major Version, LowByte: Minor Version |

## BESCHREIBUNG

Die Funktion liefert ein Array mit Informationen über die Bibliothek mit der GUID **BibliothekID**.

## BEISPIEL

```
print_r(IPS_GetLibrary("{4414D465-0A32-49AC-929A-D948CBCDE06E}"));

/* liefert z.B.:
Array
(
[LibraryID] => {4414D465-0A32-49AC-929A-D948CBCDE06E}
[Author] => Symcon GmbH
[URL] => www.symcon.de
[Name] => SzenenSteuerung
[Version] => 1.1
[Build] => 0
[Date] => 0
)
*/
```

# IPS_GetLibraryList

```
array IPS_GetLibraryList ()
```

## RÜCKGABEWERT

Ein Array von String-Werten aller GUIDs der Bibliotheken in IP-Symcon

## BESCHREIBUNG

Die Funktion ermittelt die GUIDs aller in IP-Symcon verfügbaren Bibliotheken. Die IDs werden in einem Array aufgelistet.

## BEISPIEL

```
print_r(IPS_GetLibraryList());

/* liefert z.B.:
Array
(
[0] => {FF95B199-B3BD-424C-9AEF-3F004BC672B6}
[1] => {6EC74E99-C6FD-4E03-9195-E7BD90E6C07E}
[2] => {2CE84600-0A3C-438C-AC22-86439A1E5DF0}
usw. ...
)
*/
```

# IPS_GetLibraryModules

```
array IPS_GetLibraryModules (string $BibliothekID)
```

## PARAMETERLISTE

| | |
|---|---|
| BibliothekID | Die zu zeigende BibliothekID |

## RÜCKGABEWERT

Ein Array von String-Werten aller GUIDs der Module in einer Bibliothek in IP-Symcon

## BESCHREIBUNG

Die Funktion ermittelt die GUIDs der verfügbaren Module in einer angegebenen Bibliothek. Die IDs werden in einem Array aufgelistet.

## BEISPIEL

```
print_r(IPS_GetLibraryModules("{7DC57F9A-C095-4CDE-
A6F0-2CB35A29A8FE}"));

/* liefert z.B.:
Array
(
[0] => {57040540-4432-4220-8D2D-4676B57E223D}
[1] => {48FCFDC1-11A5-4309-BB0B-A0DB8042A969}
[2] => {56800073-A809-4513-9618-1C593EE1240C}
[3] => {2FD7576A-D2AD-47EE-9779-A502F23CABB3}
usw. ...
)
*/
```

# IPS_GetModule

```
array IPS_GetModule (string $ModulID)
```

## PARAMETERLISTE

| | |
|---|---|
| ModulID | Die zu zeigende ModulID (GUIDs) |

## RÜCKGABEWERT

Folgende Informationen stehen als **key => value** Paare zur Verfügung:

| INDEX | TYP | BESCHREIBUNG |
|---|---|---|
| Aliases | array | Array von alternativen Namen (string) für das Modul |
| ChildRequirements | array | Array von GUIDs (string), die von untergeordnete Objekten erwartet werden |
| ParentRequirements | array | Array von GUIDs (string), die von übergeordneten Objekten erwartet werden |
| Implemented | array | Array von GUIDs (string), die das Modul anbietet |
| LibraryID | string | BibliothekID (GUIDs), in dem das Modul enthalten ist |
| ModuleID | string | ModulID (GUIDs) |
| ModuleName | string | Name des Moduls |
| ModuleType | integer | Typ des Moduls (0: Kern, 1: I/O, 2: Splitter, 3: Gerät, 4: Konfigurator, 5: Discovery, 6: Visualisierung) |
| Prefix | string | Prefix des Moduls zum Aufruf der dazugehörigen PHP-Funktionen (seit Version 6.1) |
| Translation | array | Übersetzungen für den ModuleName und Aliases (seit Version 7.0) |
| URL | string | URL zur Dokumentationsseite |
| Vendor | string | System-/Herstellerkennzeichnung |

## BESCHREIBUNG

Die Funktion liefert ein Array mit Informationen über das Modul mit der GUID **ModulID**.

## BEISPIEL

```
print_r(IPS_GetModule("{BAEA5454-4256-48AA-982B-538201A374D4}"));

/* liefert z.B.:
```

```
Array
(
[ParentRequirements] => Array
(
[0] => {42DFD4E4-5831-4A27-91B9-6FF1B2960260}
)

[ChildRequirements] => Array
(
)

[Implemented] => Array
(
[0] => {8A4D3B17-F8D7-4905-877F-9E69CEC3D579}
)

[Vendor] => KNX
[Aliases] => Array
(
[0] => DPT 013.x - 4-Byte Signed Value
)

[Translation] => Array
(
[de] => Array
(
[KNX DPT 13] => KNX DPT 13
[DPT 013.x - 4-Byte Signed Value] => DPT 013.x - 4-Byte
vorzeichenbehaftet Wert
)
)

[URL] => https://www.symcon.de/service/dokumentation/modulreferenz/
knx/
[LibraryID] => {0945206A-47AA-4FDD-9093-99051E410E82}
[ModuleID] => {BAEA5454-4256-48AA-982B-538201A374D4}
[ModuleName] => KNX DPT 13
[ModuleType] => 3
)
*/
```

# IPS_GetModuleList

```
array IPS_GetModuleList ()
```

## RÜCKGABEWERT

Ein Array von String-Werten aller GUIDs der Module in IP-Symcon

## BESCHREIBUNG

Die Funktion ermittelt die GUIDs aller in IP-Symcon verfügbaren Module. Die IDs werden in einem Array aufgelistet.

## BEISPIEL

```
print_r(IPS_GetModuleList());

/* liefert z.B.:
Array
(
[0] => {2D8B0172-166C-45B0-B979-61A787809E22}
[1] => {CCC52C33-D77A-4AE9-A6CF-462F152532A0}
[2] => {11842D31-3CA4-4F1B-BBA9-E4A0FE1873AF}
[3] => {4CB91589-CE01-4700-906F-26320EFCF6C4}
usw. ...
)
*/

//Alle Modulnamen mit GUID ausgeben
foreach(IPS_GetModuleList() as $guid)
{
$module = IPS_GetModule($guid);
// Nur Module aus der Kern-Bibliothek
if($module['LibraryID'] == "{0945206A-47AA-4FDD-9093-99051E410E82}")
{
$pair[$module['ModuleName']] = $guid;
}
}
ksort($pair);
foreach($pair as $key=>$guid)
{
echo $key." = ".$guid."\n";
}

/*
ALL2x = {D805EC4C-7D17-4E84-98D3-A441AA71ACA3}
```

```
ALL3690 = {BBD04875-8CC5-412A-B848-B2AB6F08C425}
ALL3691 = {07CAE61E-01FA-4086-A2D1-55C442D8C8BF}
ALL4000 = {2D8B0172-166C-45B0-B979-61A787809E22}
ALLUniversal = {FB6B8DD0-CD8E-4CE7-8613-D7BD6436720A}
Archive Control = {43192F0B-135B-4CE7-A0A7-1475603F3060}
CMI = {AAAB8DD0-CD8E-4CE7-8613-D913371701AE}
Calendar Control = {1B9A5AF2-F2D2-4827-A13D-FFA6AB350371}
Client Socket = {3CFF0FD9-E306-41DB-9B5A-9D06D38576C3}
Connect Control = {9486D575-BE8C-4ED8-B5B5-20930E26DE6F}
Cutter = {AC6C6E74-C797-40B3-BA82-F135D941D1A2}
DMX Gateway = {B1E43BF6-770A-4FD7-B4FE-6D265F93746B}
DMX Output = {E19C2E41-7347-4A3B-B7D9-A9A88E0D133E}
DMX RGB = {1AF36760-644E-4DA9-8776-01BA6DF6178E}
DNSSD Control = {780B2D48-916C-4D59-AD35-5A429B2355A5}
Dummy Module = {485D0419-BE97-4548-AA9C-C083EB82E61E}
EnOcean Configurator = {9BA823CF-83C6-4A5F-DBC2-295CA971833E}
EnOcean D20500 = {8DFBE8BC-FC2D-61F0-375C-C8C2A13CDC1A}
EnOcean D20501 = {927296DC-B74F-F0B8-2C4E-6B98236EDD94}
EnOcean D20502 = {B4D91880-71C1-626B-94D7-521565495264}
EnOcean Discovery = {29E1D618-F3CB-6868-59C6-4DCD02F8C78D}
EnOcean EEP A50201RX = {432FF87E-4497-48D6-8ED9-EE7104A50201}
EnOcean EEP A50202RX = {432FF87E-4497-48D6-8ED9-EE7104A50202}
EnOcean EEP A50203RX = {432FF87E-4497-48D6-8ED9-EE7104A50203}
EnOcean EEP A50204RX = {432FF87E-4497-48D6-8ED9-EE7104A50204}
EnOcean EEP A50205RX = {432FF87E-4497-48D6-8ED9-EE7104A50205}
EnOcean EEP A50206RX = {432FF87E-4497-48D6-8ED9-EE7104A50206}
EnOcean EEP A50207RX = {432FF87E-4497-48D6-8ED9-EE7104A50207}
EnOcean EEP A50208RX = {432FF87E-4497-48D6-8ED9-EE7104A50208}
EnOcean EEP A50209RX = {432FF87E-4497-48D6-8ED9-EE7104A50209}
EnOcean EEP A5020ARX = {432FF87E-4497-48D6-8ED9-EE7104A5020A}
EnOcean EEP A5020BRX = {432FF87E-4497-48D6-8ED9-EE7104A5020B}
EnOcean EEP A50210RX = {432FF87E-4497-48D6-8ED9-EE7104A50210}
EnOcean EEP A50211RX = {432FF87E-4497-48D6-8ED9-EE7104A50211}
EnOcean EEP A50212RX = {432FF87E-4497-48D6-8ED9-EE7104A50212}
EnOcean EEP A50213RX = {432FF87E-4497-48D6-8ED9-EE7104A50213}
EnOcean EEP A50214RX = {432FF87E-4497-48D6-8ED9-EE7104A50214}
EnOcean EEP A50215RX = {432FF87E-4497-48D6-8ED9-EE7104A50215}
EnOcean EEP A50216RX = {432FF87E-4497-48D6-8ED9-EE7104A50216}
EnOcean EEP A50217RX = {432FF87E-4497-48D6-8ED9-EE7104A50217}
EnOcean EEP A50218RX = {432FF87E-4497-48D6-8ED9-EE7104A50218}
EnOcean EEP A50219RX = {432FF87E-4497-48D6-8ED9-EE7104A50219}
EnOcean EEP A5021ARX = {432FF87E-4497-48D6-8ED9-EE7104A5021A}
EnOcean EEP A5021BRX = {432FF87E-4497-48D6-8ED9-EE7104A5021B}
EnOcean EEP A50220RX = {432FF87E-4497-48D6-8ED9-EE7104A50220}
EnOcean EEP A50230RX = {432FF87E-4497-48D6-8ED9-EE7104A50230}
EnOcean EEP A50401RX = {432FF87E-4497-48D6-8ED9-EE7104A50401}
EnOcean EEP A50402RX = {432FF87E-4497-48D6-8ED9-EE7104A50402}
EnOcean EEP A50403RX = {432FF87E-4497-48D6-8ED9-EE7104A50403}
EnOcean EEP A50601RX = {432FF87E-4497-48D6-8ED9-EE7104A50601}
EnOcean EEP A50602RX = {432FF87E-4497-48D6-8ED9-EE7104A50602}
EnOcean EEP A50603RX = {432FF87E-4497-48D6-8ED9-EE7104A50603}
EnOcean EEP A50701RX = {432FF87E-4497-48D6-8ED9-EE7104A50701}
EnOcean EEP A50702RX = {432FF87E-4497-48D6-8ED9-EE7104A50702}
EnOcean EEP A50703RX = {432FF87E-4497-48D6-8ED9-EE7104A50703}
EnOcean EEP A50801RX = {432FF87E-4497-48D6-8ED9-EE7104A50801}
```

```
EnOcean EEP A50802RX = {432FF87E-4497-48D6-8ED9-EE7104A50802}
EnOcean EEP A50803RX = {432FF87E-4497-48D6-8ED9-EE7104A50803}
EnOcean EEP A50901RX = {432FF87E-4497-48D6-8ED9-EE7104A50901}
EnOcean EEP A50902RX = {432FF87E-4497-48D6-8ED9-EE7104A50902}
EnOcean EEP A50904RX = {432FF87E-4497-48D6-8ED9-EE7104A50904}
EnOcean EEP A50905RX = {432FF87E-4497-48D6-8ED9-EE7104A50905}
EnOcean EEP A50906RX = {432FF87E-4497-48D6-8ED9-EE7104A50906}
EnOcean EEP A50907RX = {432FF87E-4497-48D6-8ED9-EE7104A50907}
EnOcean EEP A50908RX = {432FF87E-4497-48D6-8ED9-EE7104A50908}
EnOcean EEP A51001RX = {432FF87E-4497-48D6-8ED9-EE7104A51001}
EnOcean EEP A51002RX = {432FF87E-4497-48D6-8ED9-EE7104A51002}
EnOcean EEP A51003RX = {432FF87E-4497-48D6-8ED9-EE7104A51003}
EnOcean EEP A51004RX = {432FF87E-4497-48D6-8ED9-EE7104A51004}
EnOcean EEP A51005RX = {432FF87E-4497-48D6-8ED9-EE7104A51005}
EnOcean EEP A51006RX = {432FF87E-4497-48D6-8ED9-EE7104A51006}
EnOcean EEP A51007RX = {432FF87E-4497-48D6-8ED9-EE7104A51007}
EnOcean EEP A51008RX = {432FF87E-4497-48D6-8ED9-EE7104A51008}
EnOcean EEP A51009RX = {432FF87E-4497-48D6-8ED9-EE7104A51009}
EnOcean EEP A5100ARX = {432FF87E-4497-48D6-8ED9-EE7104A5100A}
EnOcean EEP A5100BRX = {432FF87E-4497-48D6-8ED9-EE7104A5100B}
EnOcean EEP A5100CRX = {432FF87E-4497-48D6-8ED9-EE7104A5100C}
EnOcean EEP A5100DRX = {432FF87E-4497-48D6-8ED9-EE7104A5100D}
EnOcean EEP A51010RX = {432FF87E-4497-48D6-8ED9-EE7104A51010}
EnOcean EEP A51011RX = {432FF87E-4497-48D6-8ED9-EE7104A51011}
EnOcean EEP A51012RX = {432FF87E-4497-48D6-8ED9-EE7104A51012}
EnOcean EEP A51013RX = {432FF87E-4497-48D6-8ED9-EE7104A51013}
EnOcean EEP A51014RX = {432FF87E-4497-48D6-8ED9-EE7104A51014}
EnOcean EEP A51015RX = {432FF87E-4497-48D6-8ED9-EE7104A51015}
EnOcean EEP A51016RX = {432FF87E-4497-48D6-8ED9-EE7104A51016}
EnOcean EEP A51017RX = {432FF87E-4497-48D6-8ED9-EE7104A51017}
EnOcean EEP A51018RX = {432FF87E-4497-48D6-8ED9-EE7104A51018}
EnOcean EEP A51019RX = {432FF87E-4497-48D6-8ED9-EE7104A51019}
EnOcean EEP A5101ARX = {432FF87E-4497-48D6-8ED9-EE7104A5101A}
EnOcean EEP A5101BRX = {432FF87E-4497-48D6-8ED9-EE7104A5101B}
EnOcean EEP A5101CRX = {432FF87E-4497-48D6-8ED9-EE7104A5101C}
EnOcean EEP A5101DRX = {432FF87E-4497-48D6-8ED9-EE7104A5101D}
EnOcean EEP A5101FRX = {432FF87E-4497-48D6-8ED9-EE7104A5101F}
EnOcean EEP A51020RX = {432FF87E-4497-48D6-8ED9-EE7104A51020}
EnOcean EEP A51021RX = {432FF87E-4497-48D6-8ED9-EE7104A51021}
EnOcean EEP A51022RX = {BBE179DE-1241-CB89-6803-492049D5546B}
EnOcean EEP A51023RX = {9880FC1A-AD89-5749-12A0-AF6C99D50B28}
EnOcean EEP A51101RX = {432FF87E-4497-48D6-8ED9-EE7104A51101}
EnOcean EEP A51102RX = {432FF87E-4497-48D6-8ED9-EE7104A51102}
EnOcean EEP A51103RX = {432FF87E-4497-48D6-8ED9-EE7104A51103}
EnOcean EEP A51104RX = {432FF87E-4497-48D6-8ED9-EE7104A51104}
EnOcean EEP A51200RX = {432FF87E-4497-48D6-8ED9-EE7104A51200}
EnOcean EEP A51201RX = {432FF87E-4497-48D6-8ED9-EE7104A51201}
EnOcean EEP A51202RX = {432FF87E-4497-48D6-8ED9-EE7104A51202}
EnOcean EEP A51203RX = {432FF87E-4497-48D6-8ED9-EE7104A51203}
EnOcean EEP A51301RX = {432FF87E-4497-48D6-8ED9-EE7104A51301}
EnOcean EEP A51302RX = {432FF87E-4497-48D6-8ED9-EE7104A51302}
EnOcean EEP A51303RX = {432FF87E-4497-48D6-8ED9-EE7104A51303}
EnOcean EEP A51304RX = {432FF87E-4497-48D6-8ED9-EE7104A51304}
EnOcean EEP A51305RX = {432FF87E-4497-48D6-8ED9-EE7104A51305}
EnOcean EEP A51401RX = {432FF87E-4497-48D6-8ED9-EE7104A51401}
```

```
EnOcean EEP A51402RX = {432FF87E-4497-48D6-8ED9-EE7104A51402}
EnOcean EEP A51403RX = {432FF87E-4497-48D6-8ED9-EE7104A51403}
EnOcean EEP A51404RX = {432FF87E-4497-48D6-8ED9-EE7104A51404}
EnOcean EEP A51405RX = {432FF87E-4497-48D6-8ED9-EE7104A51405}
EnOcean EEP A51406RX = {432FF87E-4497-48D6-8ED9-EE7104A51406}
EnOcean EEP A51407RX = {B3A8A26C-FF2F-491E-9279-E981959293C0}
EnOcean EEP A51408RX = {2C9E6E3A-319F-7ADF-024C-E8DEA7209A52}
EnOcean EEP A51409RX = {29CBDDA5-6CA6-4C17-9360-1979DCBD1E43}
EnOcean EEP A5140ARX = {1F3E63B6-F354-44FF-98DE-BEB63F36310C}
EnOcean EEP A52006 = {D25790E7-DC98-7EE3-4D74-2F0484392795}
EnOcean EEP A52012RX = {432FF87E-4497-48D6-8ED9-EE7104A52012}
EnOcean EEP A52012TX = {432FF87E-4497-48D6-8ED9-AA7104A52012}
EnOcean EEP A53001RX = {432FF87E-4497-48D6-8ED9-EE7104A53001}
EnOcean EEP A53002RX = {432FF87E-4497-48D6-8ED9-EE7104A53002}
EnOcean EEP A53003RX = {432FF87E-4497-48D6-8ED9-EE7104A53003}
EnOcean EEP A53004RX = {432FF87E-4497-48D6-8ED9-EE7104A53004}
EnOcean EEP A53701RX = {432FF87E-4497-48D6-8ED9-EE7104A53701}
EnOcean EEP A53808RX = {432FF87E-4497-48D6-8ED9-EE7104A53808}
EnOcean EEP A53809RX = {432FF87E-4497-48D6-8ED9-EE7104A53809}
EnOcean EEP D20100 = {8328F257-35B7-4034-AA23-20B0A30CAF11}
EnOcean EEP D20101 = {5F30BDCD-487F-44EE-9795-82C5DD429F2D}
EnOcean EEP D20102 = {C3825266-A6DE-44B6-97B4-C0BCFB6BD52B}
EnOcean EEP D20103 = {EAF7026C-A0DB-4B25-8DFD-D4FAB7A3EFCF}
EnOcean EEP D20104 = {5B38D44B-A570-4381-B719-8BFB58387AA1}
EnOcean EEP D20105 = {19FDCECC-0735-4ADA-8304-3AF53C919668}
EnOcean EEP D20106 = {1881B19B-A03C-437D-934D-28C598F13C39}
EnOcean EEP D20107 = {06DFB7CC-9945-4E1D-B08D-C505F4A7D2B4}
EnOcean EEP D20108 = {C885AB2C-7D83-4274-9420-EFEA5B8B0908}
EnOcean EEP D20109 = {BFDADCB0-BAB5-45BC-BF7A-1D19EA0FF31D}
EnOcean EEP D2010A = {6A37AC89-EFF0-40D8-88A1-8B24478FEE76}
EnOcean EEP D2010B = {F054B2EB-D812-47ED-B73C-22FBC0ADA667}
EnOcean EEP D2010C = {0389DA40-1B2A-446C-9448-03DED7977DC0}
EnOcean EEP D2010D = {744C3ACE-4680-4E1D-966F-CC959B25DFD6}
EnOcean EEP D2010E = {7E11E659-98E1-45A1-A9DB-29D337423C92}
EnOcean EEP D2010F = {AE62BF01-6C90-4FC2-8E1E-75EB5516BCF0}
EnOcean EEP D20110 = {699264EC-BEF1-4426-B501-687EA30C03B1}
EnOcean EEP D20111 = {21DDC756-E3CD-4B2D-B2F4-86092918A832}
EnOcean EEP D20112 = {47804DFF-5534-4256-8938-2C5D018736A3}
EnOcean EEP D20113 = {783B1089-4A50-4973-A522-C5288B9E1909}
EnOcean EEP D20114 = {A08C2716-CF6A-472C-A303-5C37C57A65BB}
EnOcean EEP D20300 = {8DA41914-45A6-8F0C-C935-178F76A36C42}
EnOcean EEP D2030A = {B24CF984-5CD7-B0D7-9661-41E6F2CEA0E7}
EnOcean EEP D20310 = {C14DBE59-E9B6-C9B7-7DEF-D98B8ADBA6F3}
EnOcean EEP D20601 = {6E5F79FE-3F0B-4CCE-8786-E8DD8893185B}
EnOcean EEP D20700 = {832ABD55-1173-B194-EB3F-D23681113A01}
EnOcean EEP D21101 = {3B1E2E8A-7C09-F58C-2841-3EF2D7C67F7A}
EnOcean EEP D21102 = {ED6AB19B-5A4A-80FD-BC4B-C07EE7F9915A}
EnOcean EEP D21103 = {7973060A-0241-E1E5-6E4B-CACBB08ACA76}
EnOcean EEP D21104 = {CC1482D1-6AD9-2199-C5E5-AD4EB6926455}
EnOcean EEP D21105 = {360F32BD-D9D5-5233-966A-63EAF54F671C}
EnOcean EEP D21106 = {8B2F1F85-A51C-0606-7CF3-8F980FEADB12}
EnOcean EEP D21107 = {F84B8E1B-B0DF-9383-1F6E-56EFAFB405A8}
EnOcean EEP D21108 = {1828E2A3-D83B-1433-46BF-AD0F11FF07DC}
EnOcean EEP D21430 = {1858BB3E-78BD-B090-7798-F1C0C26D912E}
EnOcean EEP D21500 = {75CA6030-43B9-8CF6-5901-E2DEF86E5249}
```

```
EnOcean EEP D23200 = {F60551AF-5D2B-4560-8BDD-7D1474C7A490}
EnOcean EEP D23201 = {C3B79AA9-54DB-4AE4-8ED4-3F954C723465}
EnOcean EEP D23202 = {0A03908E-694E-4CB1-BD35-AF7777504BFE}
EnOcean EEP D2A001 = {EA335B84-7E0A-F982-32CE-B50B60FC8E39}
EnOcean EEP D50001RX = {432FF87E-4497-48D6-8ED9-EE7104D50001}
EnOcean EEP F60101RX = {B9ED3CC4-A354-0420-A7D3-9AFC6C721FB3}
EnOcean EEP F60201RX = {432FF87E-4497-48D6-8ED9-EE7104F60201}
EnOcean EEP F60202RX = {432FF87E-4497-48D6-8ED9-EE7104F60202}
EnOcean EEP F60203RX = {432FF87E-4497-48D6-8ED9-EE7104F60203}
EnOcean EEP F60204RX = {432FF87E-4497-48D6-8ED9-EE7104F60204}
EnOcean EEP F60301RX = {432FF87E-4497-48D6-8ED9-EE7104F60301}
EnOcean EEP F60302RX = {432FF87E-4497-48D6-8ED9-EE7104F60302}
EnOcean EEP F60401RX = {432FF87E-4497-48D6-8ED9-EE7104F60401}
EnOcean EEP F60402RX = {432FF87E-4497-48D6-8ED9-EE7104F60402}
EnOcean EEP F60500RX = {74DB7CEE-18F9-478D-9FBC-CA857F807C50}
EnOcean EEP F60501RX = {432FF87E-4497-48D6-8ED9-EE7104F60501}
EnOcean EEP F60502RX = {8D42EFEE-119C-4CA8-BED2-66D18921749E}
EnOcean EEP F61000RX = {432FF87E-4497-48D6-8ED9-EE7104F61000}
EnOcean EEP F61001RX = {C31A7F85-4E76-E9C9-F920-2953DE737560}
EnOcean EltakoDimmer = {48909406-A2B9-4990-934F-28B9A80CD079}
EnOcean EltakoFABH130 = {03AEBCA9-1140-969E-7C72-728A1F30CD55}
EnOcean EltakoFABH65S = {0E8E3267-5CFA-4A52-AB3E-899170D16116}
EnOcean EltakoFAFT60 = {0BE195DC-6002-4D99-A566-3B9B0B57FAD6}
EnOcean EltakoFAH60 = {AF827EB8-08A3-434D-9690-424AFF06C698}
EnOcean EltakoFHK14 = {7C25F5A6-ED34-4FB4-8A6D-D49DFE636CDC}
EnOcean EltakoFKC = {40C99CC9-EC04-49C8-BB9B-73E21B6FDDDD}
EnOcean EltakoFRW = {40C99CC9-EC04-49C8-BB9B-73E21B6FEEEE}
EnOcean EltakoFSS12 = {7124C1BC-B260-4C5E-BF00-B38D3C7B5CB7}
EnOcean EltakoFTR65 = {40C99CC9-EC04-49C8-BB9B-73E21B6FFF65}
EnOcean EltakoFWS61 = {9E4572C0-C306-4F00-B536-E75B4950F094}
EnOcean EltakoFZS = {40C99CC9-EC04-49C8-BB9B-73E21B6FCCCC}
EnOcean EltakoRGBW = {71072C70-7ADF-6CC5-8D6F-7368132584F8}
EnOcean EltakoShutter = {1463CAE7-C7D5-4623-8539-DD7ADA6E92A9}
EnOcean EltakoSwitch = {FD46DA33-724B-489E-A931-C00BFD0166C9}
EnOcean Gateway = {A52FEFE9-7858-4B8E-A96E-26E15CB944F7}
EnOcean Hoppe = {1C8D7E80-3ED1-4117-BB53-9C5F61B1BEF3}
EnOcean Opus = {9B1F32CD-CD74-409A-9820-E5FFF064449A}
EnOcean PTM200 = {40C99CC9-EC04-49C8-BB9B-73E21B6FA265}
EnOcean PTM200RX = {63484585-F8AD-4780-BAFD-3C0353641046}
EnOcean RCM100 = {8492CEAF-ED62-4634-8A2F-B09A7CEDDE5B}
EnOcean STM100 = {FA1479DE-C0C1-433D-98BC-EA7C298D1AA5}
EnOcean STM250 = {B01DE819-EA69-4FC1-91AB-4D9FF8D55370}
EnOcean Shutter = {1463CAE7-C7D5-4623-8539-DD7ADA6E92AA}
EnOcean SmartDrive MX = {004A19B5-6E60-41A3-813C-7653ABAB6116}
EnOcean Thermokon SR0x = {B4249BC6-5BA8-45E3-B506-86680935D4EE}
EnOcean Thermokon Thanos = {54E114A3-F9B6-4747-A07E-D2A064629627}
EnOcean Valve Actuator = {004A19B5-6E60-41A3-813C-A11DC00B6116}
EnOcean alphaEOS SENSE TF-H = {F98CD462-0147-4A79-A0DF-448580E01629}
Event Control = {ED573B53-8991-4866-B28C-CBE44C59A2DA}
FHT = {A89F8DFA-A439-4BF1-B7CB-43D047208DDD}
FHZ = {57040540-4432-4220-8D2D-4676B57E223D}
FS10 = {6D508C91-F197-44A9-A1AB-A27F97A18A5F}
FS10 Gateway = {753E7267-7558-49D3-ACFB-86755C28318D}
FS20 = {48FCFDC1-11A5-4309-BB0B-A0DB8042A969}
FS20EX = {56800073-A809-4513-9618-1C593EE1240C}
```

```
HID = {E6D7692A-7F4C-441D-827B-64062CFE1C02}
HMS = {2FD7576A-D2AD-47EE-9779-A502F23CABB3}
HTTP Client = {4CB91589-CE01-4700-906F-26320EFCF6C4}
Heating Control = {FF7AF0F4-D616-4EF5-B73A-D49C56CF4C92}
Heating Control Legacy = {3F52FA69-77F5-4DE6-8B2A-347452AC5F8F}
HomeMatic Configurator = {5214C3C6-91BC-4FE1-A2D9-A3920261DA74}
HomeMatic Device = {EE4A81C6-5C90-4DB7-AD2F-F6BBD521412E}
HomeMatic Discovery = {3718244C-71A2-B20D-F754-DF5C79340AB4}
HomeMatic Socket = {A151ECE9-D733-4FB9-AA15-7F7DD10C58AF}
IMAP = {CABFCCA1-FBFF-4AB7-B11B-9879E67E152F}
IPS-868 AD = {134A1CFF-3CAA-4D2E-8517-F9BFDE7555C9}
IPS-868 Accelerometer = {304B6E9A-A518-43C6-80E8-D066879E7FD1}
IPS-868 AirQuality = {2B2842ED-0BB9-45FE-934C-1DFE07E1A724}
IPS-868 Configurator = {0A3CCE10-7613-1C54-0721-57023C98EFDA}
IPS-868 Counter = {134A1CFF-3CAA-4D2E-8517-F9BFDE7544C9}
IPS-868 DA = {134A1CFF-3CAA-4D2E-8517-F9BFDE7566C9}
IPS-868 Data = {984FFE86-71BE-48D3-943E-45277BECB52D}
IPS-868 Discovery = {F2B8A368-F9F2-7919-B299-79DC5B64B8A4}
IPS-868 Display = {9542B617-C603-4E8A-BFDE-DD69663F0226}
IPS-868 Display Input = {FBF6F86C-4901-4957-8D08-2B2CBC1DF71A}
IPS-868 Display Output = {C8CF38E9-BEAA-4149-A9ED-C8AAE04E8A6D}
IPS-868 Gateway = {995946C3-7995-48A5-86E1-6FB16C3A0F8A}
IPS-868 Input = {5509B26C-F2F7-428F-973B-FD5D07C80111}
IPS-868 Level = {C7A069AB-E2BE-4AD2-BB51-A7B52D6FAC4E}
IPS-868 Servo = {D2C92D2D-FC07-4545-AEBA-4294FD9D3035}
IPS-868 Servo Input = {7D07671F-65FF-4D9A-B521-9F0964DD9399}
IPS-868 Stripe = {7AF52E55-4994-44AE-9DB8-752895F1EB76}
IPS-868 Stripe Input = {26B22717-924E-4D58-B091-C98E49812238}
IPS-868 Thermo = {95D715AA-AB5A-4423-AD7B-5081EB542971}
IPS-868 Tracker = {50460BD9-FA93-4DE0-976A-55208C90DF15}
IPS-868 WatchDogTimer = {9F88A462-BFDA-47C5-990C-B9E20D2F861E}
IRTrans Device = {899DBC68-0147-4528-B95D-D43C568D4E56}
IRTrans Gateway = {0F0F74EF-2304-4C23-840F-EC1C5B9A9A82}
Image Grabber = {5A5D5DBD-53AB-4826-8B09-71E9E4E981E5}
JSON Decoder = {9F976F04-7DC4-BB2D-38CA-67C1AFC84C7A}
KNX Configurator = {33765ABB-CFA5-40AA-89C0-A7CEA89CFE7A}
KNX DPT 1 = {F3058AB2-AFDC-4479-A053-5F4599DF6F5B}
KNX DPT 10 = {2D26EF69-C3BC-4738-879D-FB493DD166BF}
KNX DPT 11 = {4A5AE804-59E2-41EA-ABCE-3BC80F33543C}
KNX DPT 12 = {AD992707-B7DC-4255-96DE-B70057CEC989}
KNX DPT 13 = {BAEA5454-4256-48AA-982B-538201A374D4}
KNX DPT 14 = {E8C3C9ED-A6FA-45FF-A942-A206FEEE360C}
KNX DPT 15 = {16B72DE7-3C8B-4709-8819-6A260E890A12}
KNX DPT 16 = {2146FF76-FAE6-46B6-9630-73E5F3FAE190}
KNX DPT 17 = {E8A0352C-B13F-4691-AAF5-586DE502569E}
KNX DPT 18 = {8B2A23B9-6B96-4F76-9EC8-5365BA7E9441}
KNX DPT 19 = {8456D139-6771-4821-8A37-9304BEAA1BFD}
KNX DPT 2 = {434A6354-235F-4484-8883-9FF30616E3FB}
KNX DPT 20 = {C9304609-D1CD-4F07-BE7C-E5546A91A5DF}
KNX DPT 200 = {04ED3286-C982-436F-A01E-8A03C2E976A8}
KNX DPT 201 = {5F14B9D5-297D-4C25-9E45-92BF7B7C3F02}
KNX DPT 202 = {22726542-A2B6-4FBB-BB98-BA5927932C23}
KNX DPT 203 = {3243B56D-DE90-4BED-8E07-E4DEE777C6AF}
KNX DPT 204 = {06E5F74A-BC1F-4F69-BBE6-8AD663058CD9}
KNX DPT 205 = {28837D84-085F-4701-ADA0-CF9A8F49F591}
```

```
KNX DPT 206 = {F5E3C9F5-A0E8-4919-BB38-0551AE5886A7}
KNX DPT 207 = {EE68B239-8E3A-4F54-9501-96CB4B2700A4}
KNX DPT 209 = {D9A125D0-8D82-4808-A024-0FC5E82799D6}
KNX DPT 21 = {8941E278-1901-478D-AFC8-807E3515DD28}
KNX DPT 210 = {68E7C9CE-724E-4CBE-8206-A61AD9A6A328}
KNX DPT 211 = {0636647E-EB02-4EE3-857E-0B69C0E8AA3D}
KNX DPT 212 = {4D8AD2DA-5669-4BEB-B9C7-983DA48A7777}
KNX DPT 213 = {DEA88D19-78B6-4D51-89FF-E3958151863D}
KNX DPT 214 = {AC33D286-FBBC-40BE-91CC-C470FB896C04}
KNX DPT 215 = {619D8815-1E90-4A7A-B51D-14DCB2FCE0E0}
KNX DPT 216 = {F5339A58-E891-4A22-AF29-3C801D3092EE}
KNX DPT 217 = {9CF216A5-370F-4BD9-92F0-B7BF87933A76}
KNX DPT 218 = {DB54CF66-4ED5-4457-A1EF-A3DF52EC135F}
KNX DPT 219 = {7EA00021-4300-4B9F-A274-863E1DD41695}
KNX DPT 22 = {DD0F4EF5-3C71-44F1-A346-397D34809D30}
KNX DPT 220 = {14BBE7DF-8442-4ED8-9D7D-C00B3E1B892A}
KNX DPT 221 = {3C16BD4B-45BF-435A-84DA-D7908F2D6140}
KNX DPT 222 = {3AD960E1-E129-4D96-BFAC-696AC149C629}
KNX DPT 223 = {EA6A0F52-ECEF-46D2-A942-96DDED471732}
KNX DPT 224 = {7245A626-1864-4834-A181-EE4FB4AD50DC}
KNX DPT 225 = {CC6C64D0-2F48-4497-A6A6-2120BAE5CD52}
KNX DPT 229 = {3F20B758-71DD-4C0B-A169-D4B59B023555}
KNX DPT 23 = {7CA94B7E-6460-4D68-8568-EE4B0912C5BF}
KNX DPT 230 = {88546CFE-1265-494C-B80C-1A3A6E1E671D}
KNX DPT 231 = {44A18680-880B-4ACC-9E84-8FB79FB9D727}
KNX DPT 232 = {DE2E3FBF-1588-4293-B76C-C90446D9E5F1}
KNX DPT 234 = {C667FC98-9FE4-4953-87ED-2B75F59704A4}
KNX DPT 235 = {960A48AD-36AB-4174-A298-F31E99014738}
KNX DPT 236 = {8A57A40D-AE55-42D5-BF33-4F1E0DDA1F8C}
KNX DPT 237 = {E3E1F95B-3356-4F6A-A419-9486125E023A}
KNX DPT 238 = {117D49A5-7D8F-4509-8669-F44972F072DD}
KNX DPT 239 = {F8AE906F-76E2-4464-96DA-B9A6D19049C9}
KNX DPT 240 = {CC5F383D-975B-44E9-BC18-C64FA2698CC0}
KNX DPT 241 = {A26BC82A-5953-4385-9FAA-5A5D3814D2B5}
KNX DPT 242 = {F565EF9F-30A5-4E1C-BA10-7BC15AB4EFB4}
KNX DPT 249 = {34B83F38-03F2-4D15-9031-70197CE9456E}
KNX DPT 25 = {9FB4BD67-5E67-4E94-B8C8-71D609048BBB}
KNX DPT 251 = {9165E3E2-8ED2-4152-BEFF-97793823FE4D}
KNX DPT 26 = {135F8FCF-DAFC-46AA-85EC-94C911FD21DF}
KNX DPT 27 = {EDE5D3E1-FB5A-48EE-A944-0A4ED0FBACC1}
KNX DPT 28 = {3E175751-4B84-4E8E-98C1-BF0D12835F2C}
KNX DPT 29 = {AA9ED268-3B37-4962-85BB-CBC73CB11618}
KNX DPT 3 = {EB931B1A-3A4A-47E9-AC67-D2A0D285B60C}
KNX DPT 30 = {2E7AADF5-F1DF-476D-B207-22D204C4AD28}
KNX DPT 31 = {5AD8B9C3-5A87-4426-B9A1-C7935022A680}
KNX DPT 4 = {6891595A-BF7C-4F4C-8A3B-7EA7346C8657}
KNX DPT 5 = {EBD0EE8D-DFA5-449F-BBE4-49CFC5F0EEB4}
KNX DPT 6 = {BAC22632-D6ED-4D88-82EA-726D700C4F2A}
KNX DPT 7 = {DCAFED37-B20B-4EEB-AA44-799C9B804557}
KNX DPT 8 = {7852DBDE-C68A-4A3E-88AF-155B99605371}
KNX DPT 9 = {95CDE37D-72CE-4360-BB91-573757C274A0}
KNX Discovery = {3519D734-79C9-4E7A-9E9B-91B6374D1C10}
KNX EIS Group = {D62B95D3-0C5E-406E-B1D9-8D102E50F64B}
KNX Gateway = {1C902193-B044-43B8-9433-419F09C641B8}
KNX RGB = {4D7F7548-0979-4ABD-9BB3-81F9477C0903}
```

```
KNX RGBW = {81F54858-72B1-4C2C-8CE3-7E00A3168378}
KNX Shutter = {24A9D68D-7B98-4D74-9BAE-3645D435A9EF}
KS300 = {9D21F700-6F67-4FBB-ACC2-AA42420A0486}
LCN Configurator = {0F64973F-4669-4272-BDB3-6338D0350269}
LCN Data = {A26E7E5A-A7C5-4063-8BE0-ED8BB26F8411}
LCN Display = {A26E7E5A-A7C5-4063-8BE0-ED8BB26F8522}
LCN Gateway = {9BDFC391-DEFF-4B71-A76B-604DBA80F207}
LCN Module = {0E31FED6-E465-4621-95D4-AAF2683C41EC}
LCN RGBW = {839D1208-D4BB-4FA7-86FA-AA9FBF96ED17}
LCN Shutter = {C81E019F-6341-4748-8644-1C29D99B813E}
LCN ShutterMotor = {C81E019F-6341-4748-8644-1C29D99B813F}
LCN Unit = {2D871359-14D8-493F-9B01-26432E3A710F}
LCN Value = {0102BDC9-3B85-4A11-968D-7D314DA07C06}
LevelJet = {D64B904C-5312-443C-A2F3-03201ED9811C}
Location Control = {45E97A63-F870-408A-B259-2933F7EABF74}
M-Bus Configurator = {73336CB8-B5E5-187E-8BB8-CC4F32BF127D}
M-Bus Device = {B53BE2E5-892D-4537-94AC-EAC68A469188}
M-Bus Discovery = {6BED8C28-002F-66B7-07B5-4145D4408F34}
M-Bus Gateway = {301AB802-23CD-4DE2-91D1-6E3BC9BF03FC}
MQTT Client = {F7A0DD2E-7684-95C0-64C2-D2A9DC47577B}
MQTT Client Configurator = {2408C0E0-E672-7FE5-414B-F78C9B9244E4}
MQTT Client Device = {91D174F2-AE0F-B8D8-5EF4-6232B9083CCF}
MQTT Server = {C6D2AEB3-6E1F-4B2E-8E69-3A1A00246850}
MQTT Server Configurator = {CC4F15B1-81C2-4F45-8D53-972F0C9C8103}
MQTT Server Device = {01C00ADD-D04E-452E-B66A-D253278743FE}
Media Player = {2999EBBB-5D36-407E-A52B-E9142A45F19C}
ModBus Address = {CB197E50-273D-4535-8C91-BB35273E3CA5}
ModBus Configurator = {8C1E6AC6-4C45-067A-3696-6B4284B75C4B}
ModBus Gateway = {A5F663AB-C400-4FE5-B207-4D67CC030564}
Module Control = {B8A5067A-AFC2-3798-FEDC-BCD02A45615E}
Multicast Socket = {BAB408E0-0A0F-48C3-B14E-9FB2FA81F66A}
Notification Control = {D4B231D6-8141-4B9E-9B32-82DA3AEEAB78}
OZW Configurator = {62BDA2F6-6206-4DC3-9431-4328B5890836}
OZW DataPoint = {422A36CD-5565-4CDC-9FC4-242D3C810901}
OZW Device = {CE54AFB0-625A-4F37-B5C1-EAF1FD2A1DA6}
OZW Gateway = {1D60A51E-FF04-4D82-9E0D-04B3A11EF13F}
OneWire Configurator = {F462BFF3-6772-4720-8450-49E6E2820643}
OneWire Discovery = {1739B440-4A46-F5FF-8640-F9DF3D14C06B}
OneWire F05 = {F1B54BB1-DC7D-42D9-A973-6CA4789E358F}
OneWire F10 = {685D4911-57CC-43AF-BA36-183EF2C8518F}
OneWire F12 = {50DB3978-CF6A-4BCE-87DF-5BB45D900628}
OneWire F1D = {88A63FF8-832B-48E3-B989-A416C7908E6A}
OneWire F20 = {5DF182B0-01DF-4D8A-82CD-E646FD9BF0B2}
OneWire F26 = {301FA314-65F8-4317-8BF5-729CF8664F54}
OneWire F28 = {766E337F-A707-48CC-A323-70D7E77E3F8C}
OneWire F29 = {6A75828A-25CD-4CF3-83EA-DAAB914030A7}
OneWire F2C = {E02955B3-49E4-47A9-A9ED-2C71401F6D6E}
OneWire F3A = {BD0F2622-F67C-4248-9A04-316DF13914C3}
OneWire Gateway = {CED1D815-2477-4B05-8F65-0E4475913063}
POP3 = {69CA7DBF-5FCE-4FDF-9F36-C05E0136ECFD}
Popup Module = {5EA439B8-FB5C-4B81-AA35-1D14F4EA9821}
Presence Control = {B263AFBC-950F-462C-95B1-E3ACACE0B9B0}
Register Variable = {F3855B3C-7CD6-47CA-97AB-E66D346C037F}
SMS = {96102E00-FD8C-4DD3-A3C2-376A44895AB1}
SMS REST = {96102E00-FD8C-4DD3-A3C2-376A44895AC2}
```

```
SMTP = {375EAF21-35EF-4BC4-83B3-C780FD8BD88A}
SSDP Control = {FFFFA648-B296-E785-96ED-065F7CEE6F29}
SSE Client = {2FADB4B7-FDAB-3C64-3E2C-068A4809849A}
Serial Port = {6DC3D946-0D31-450F-A8C6-C42DB8D7D4F1}
Server Socket = {8062CF2B-600E-41D6-AD4B-1BA66C32D6ED}
Shutter Control = {9512FC2C-7999-4BD0-B322-B6C49BD09127}
Shutter Control Legacy = {542CC907-CA63-4E7A-A8C7-92F74639FA4C}
Siemens Configurator = {691B4B06-C774-BA52-6C63-B88D5EDD79EB}
Siemens Device = {932076B1-B18E-4AB6-AB6D-275ED30B62DB}
Siemens Gateway = {1B0A36F7-343F-42F3-8181-0748819FB324}
Skin Control = {E8C1043A-E2C2-4CC8-88DC-CBF02D45E15E}
Store Control = {F45B5D1F-56AE-4C61-9AB2-C87C63149EC3}
TMEX = {ABEE57FB-E243-461D-B2F7-4D176ACB7C7C}
Text Parser = {4B00C7F7-1A6D-4795-A2D2-08151854D259}
Text To Speech = {684CC410-6777-46DD-A33F-C18AC615BB94}
ThermoJet = {C3380B6E-2351-4A19-9668-A17960F97E51}
UDP Socket = {82347F20-F541-41E1-AC5B-A636FD3AE2D8}
UVR1611 = {0A8F9D69-E78B-4EAB-8E26-C8A4ACF7FA25}
Universal RX = {23D1E4CD-77E9-4F4B-B8B3-53AE81222AF6}
Util Control = {B69010EA-96D5-46DF-B885-24821B8C8DBD}
Velleman USB = {8CE70CD0-6674-4907-B3CE-F6E5235C9938}
Virtual IO = {6179ED6A-FC31-413C-BB8E-1204150CF376}
VoIP = {A4224A63-49EA-445F-8422-22EF99D8F624}
WMRS200 = {DD2A4676-82C6-4154-9AAD-DE30668D53B0}
WMRS200 Gateway = {E4FDC411-95D5-453C-B731-0CEB0483E663}
WS Client = {D68FD31F-0E90-7019-F16C-1949BD3079EF}
WebFront Visualization = {3565B1F2-8F7B-4311-A4B6-1BF1D868F39E}
WebHook Control = {015A6EB8-D6E5-4B93-B496-0D3F77AE9FE1}
WebOAuth Control = {F99BF07D-CECA-438B-A497-E4B55F139D37}
WebServer = {D83E9CCF-9869-420F-8306-2B043E9BA180}
WinLIRC = {19E51FC2-064B-4A51-8995-11FEFF7F129A}
WuT Counter = {5F1C5261-07A2-4A7E-9AC4-88AF9ED29420}
WuT Gateway = {01CA6888-C833-484E-A3F3-806535421CB7}
WuT Input = {C3D0F82C-CD07-4AA8-AE5C-7AD983FE91F3}
WuT Output = {E85C40B3-C1E9-4A60-85C7-6CDDA3D8D7BF}
WuT ThermoHygro = {2EF634A4-D96D-4018-BD90-94E487A89D49}
XBee Gateway = {7FA47C08-E31B-44A6-9E50-20C4DDD3E081}
XBee Splitter = {9D5DCE79-1A97-4531-9D10-68839F4BEAAC}
Z-Wave Configurator = {2D7CA355-2C51-4430-8F67-4E397EAAEA19}
Z-Wave Discovery = {5754F842-9D3E-BF8D-E3FC-8B1237AEC7E5}
Z-Wave Gateway = {4EF72D56-BF9F-4347-8F0A-2035D241116F}
Z-Wave Module = {101352E1-88C7-4F16-998B-E20D50779AF6}
dS Apartment = {555C65F5-F436-467F-A2B1-DAA1BB02CA3B}
dS Configurator = {7CADC358-60B0-4D91-9825-27E9029B62C4}
dS Expert = {1822D865-2093-32A1-6999-280C82E669EB}
dS Joker = {DB54D0EC-64C4-4941-9B64-2337183C9C2E}
dS Light = {DB54D0DB-64C3-4930-9B53-2337183C9C1D}
dS Shutter = {3DDA1E2B-B807-4680-AB6D-E7E8FBD6093A}
dS Splitter = {8D7872F4-CAC3-409D-926B-CCF1BA9E937B}
dS Zone = {4E0FE8A0-1A1E-460A-9446-2D18B3635310}
ekey Fingerprint = {9AB0DD38-A396-4D35-81A1-2FB6A3992332}
xComfort Binary Input = {3040A77D-3E9C-42D4-A1B6-329EFE8086DB}
xComfort Configurator = {5DD921D4-4712-443F-B89F-03434A4DBF94}
xComfort Dimmer = {8050FEEC-C875-4BDD-9143-D15134B89D35}
xComfort Energy = {814067F0-EACB-43C3-99BD-5CB9B2F8FB9E}
```

```
xComfort Gateway = {D2DCE381-19A7-4D14-B819-49C0539BC350}
xComfort HRV = {E4693C3F-95F1-48B6-9443-4A6B3EE0FACA}
xComfort Heating = {586D59EE-04A8-4896-B49D-63B4DD9618EF}
xComfort Humidity = {3EBA1AB7-72CA-48D2-8F89-813E085D41BB}
xComfort Impulse = {A374DCF0-CEDE-4EB7-B6A8-E92787E19B25}
xComfort One Channel Heating = {62444FDF-FBF0-8F98-50B3-
B1E85ADD86E2}
xComfort Remote = {DCBD8143-83AB-4068-8FC0-0C92A93AA8A8}
xComfort Room Control = {1A1C4C67-C99D-4D3E-8A34-23581CE8CCAA}
xComfort Shutter = {1B7B5B7D-CAA9-4AB5-B9D8-EC805EC955AD}
xComfort Switch = {27DD9788-802E-45B7-BA54-FB97141398F7}
xComfort Temperature = {591B4A05-E5BF-4EEA-BC34-36E6F1CC9D56}
xComfort Value RX = {DA2FCC12-2DE1-404A-8A5E-1C6AF05F96A2}
xComfort Value TX = {ED6A1E00-81C7-416F-9F97-1F2CC8F45B15}
*/
```

# IPS_GetModuleListByType

Benötigt: IP-Symcon >= 3.3

```
array IPS_GetModuleListByType (int $ModulTyp)
```

## PARAMETERLISTE

| WERT | BESCHREIBUNG |
|---|---|
| 0 | Kern |
| 1 | I/O |
| 2 | Splitter |
| 3 | Gerät |
| 4 | Konfigurator |
| 5 | Discovery |
| 6 | Visualisierung |

ModulTyp

## RÜCKGABEWERT

Ein Array von String-Werten aller GUIDs der Module in IP-Symcon mit dem angegebenen Typ

## BESCHREIBUNG

Die Funktion ermittelt die GUIDs aller in IP-Symcon verfügbaren Module des Typs **ModulTyp**. Die IDs werden in einem Array aufgelistet.

## BEISPIEL

```
print_r(IPS_GetModuleListByType(0));

/* liefert z.B.:
Array
(
[0] => {2D8B0172-166C-45B0-B979-61A787809E22}
[1] => {CCC52C33-D77A-4AE9-A6CF-462F152532A0}
[2] => {11842D31-3CA4-4F1B-BBA9-E4A0FE1873AF}
[3] => {4CB91589-CE01-4700-906F-26320EFCF6C4}
usw. ...
)
*/
```

# IPS_IsModuleCompatible

```
boolean IPS_IsModuleCompatible (string $ModulID, string
$ÜbergeordneteID)
```

## PARAMETERLISTE

| | |
|---|---|
| ModulID | GUIDs |
| ÜbergeordneteID | GUIDs |

## RÜCKGABEWERT

TRUE, wenn die Module kompatibel sind, sonst FALSE

## BESCHREIBUNG

Die Funktion ermittelt, ob das Modul mit der GUID **ÜbergeordneteID** ein valides übergeordnetes Modul des Moduls mit der GUID **ModulID** ist.

## BEISPIEL

```
if (IPS_IsModuleCompatible("{48FCFDC1-11A5-4309-BB0B-A0DB8042A969}",
"{57040540-4432-4220-8D2D-4676B57E223D}"))
echo "FS20 Modul ist zum FHZ Modul kompatibel!";
```

# IPS_LibraryExists

```
boolean IPS_LibraryExists (string $BibliothekID)
```

## PARAMETERLISTE

| | |
|---|---|
| BibliothekID | GUIDs |

## RÜCKGABEWERT

Falls die BibliothekID im System existiert, wird **TRUE** zurückgemeldet, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion prüft, ob die Bibliothek mit der GUID **BibliothekID** existiert.

## BEISPIEL

```
if (IPS_LibraryExists("{7DC57F9A-C095-4CDE-A6F0-2CB35A29A8FE}"))
echo "ELV Bibliothek ist vorhanden!";
```

# IPS_ModuleExists

```
boolean IPS_ModuleExists (string $ModulID)
```

## PARAMETERLISTE

| | |
|---|---|
| ModulID | GUIDs |

## RÜCKGABEWERT

Falls die ModulID im System existiert, wird **TRUE** zurückgemeldet, andernfalls **FALSE**

## BESCHREIBUNG

Die Funktion prüft, ob das Modul mit der GUID **ModulID** existiert.

## BEISPIEL

```
if (IPS_ModuleExists("{48FCFDC1-11A5-4309-BB0B-A0DB8042A969}"))
echo "FS20 Modul ist vorhanden!";
```

# Objektverwaltung

| | |
|---|---|
| IPS_GetChildrenIDs | liefert eine Aufstellung aller untergeordneten Objekte |
| IPS_GetLocation | liefert den kompletten Pfad des Objekts |
| IPS_GetName | liefert den Namen des Objekts |
| IPS_GetObject | liefert umfassende Informationen über ein Objekt |
| IPS_GetObjectIDByIdent | liefert die ID eines Objekts anhand des Identifikators |
| IPS_GetObjectIDByName | liefert die ID eines Objekts anhand des Namens |
| IPS_GetObjectList | liefert eine Aufstellung aller existierenden Objekte |
| IPS_GetParent | liefert die ParentID des Objekts |
| IPS_HasChildren | prüft, ob Objekt untergeordnete Objekte hat |
| IPS_IsChild | prüft, ob Objekt einem anderen Objekt untergeordnet ist |
| IPS_ObjectExists | prüft, ob ein bestimmtes Objekt existiert |
| IPS_SetDisabled | Setzt ein Objekt für das WebFront inaktiv |
| IPS_SetHidden | setzt die Sichtbarkeit eines beliebigen Objekts |
| IPS_SetIcon | ordnet einem beliebigen Objekt ein Icon zu |
| IPS_SetIdent | ordnet einem beliebigen Objekt einen Identifikator zu |
| IPS_SetInfo | ordnet einem beliebigen Objekt erweiterte Informationen zu |
| IPS_SetName | ordnet einem beliebigen Objekt einen Namen zu |
| IPS_SetParent | verschiebt ein Objekt im logischen Strukturbaum |
| IPS_SetPosition | ordnet einem beliebigen Objekt eine Position zu |

# IPS_GetChildrenIDs

```
array IPS_GetChildrenIDs (int $ObjektID)
```

## PARAMETERLISTE

| | |
|---|---|
| ObjektID | ID für die die Unterobjekte zurückgegeben werden sollen |

## RÜCKGABEWERT

Ein Array von Integer-Werten aller IDs der Unterobjekte von Objekt **ObjektID**

## BESCHREIBUNG

Die Funktion ermittelt die IDs aller Unterobjekte von Objekt **ObjektID**. Die IDs werden in einem Array aufgelistet.

## BEISPIEL

```
print_r(IPS_GetChildrenIDs(32102));

/* liefert z.B.:
Array
(
[0] => 11650
[1] => 25578
[2] => 30202
usw. ...
)
*/
```

# IPS_GetLocation

```
string IPS_GetLocation (int $ObjektID)
```

## PARAMETERLISTE

| | |
|---|---|
| ObjektID | ID für die der Pfad/Namen erstellt werden soll |

## RÜCKGABEWERT

Pfad/Name des Objekts im logischen Objektbaum

## BESCHREIBUNG

Diese Funktion dient dazu den kompletten Pfad inklusive Namen eines Objektes zu erhalten



Ab Version 4.0: Wenn das Objekt ein Ereignis ist und keinen Namen hat (Unbenanntes Objekt), wird der beschreibende Name zurückgegeben



Diese Funktion dient nur zur schnellen Visualisierung des Pfades. Da im Namen eines Objektes Backslashes "\" enthalten sein können, ist der Rückgabewert nicht eindeutig. Eine Weiterverwendung mit z.B. explode ist daher nicht sinnvoll.

## BEISPIEL

```
echo IPS_GetLocation(47359);
//gibt z.B. zurück: "FHT8b\LowBattery"
```

# IPS_GetName

```
string IPS_GetName (int $ObjektID)
```

## PARAMETERLISTE

| | |
|---|---|
| ObjektID | ID für die der Namen zurückgegeben werden soll |

## RÜCKGABEWERT

Name des Objekts im logischen Objektbaum

## BESCHREIBUNG

Die Funktion gibt den Namen eines Objektes zurück.



Ab Version 4.0: Wenn das Objekt ein Ereignis ist und keinen Namen hat (Unbenanntes Objekt), wird der beschreibende Name zurückgegeben

## BEISPIEL

```
echo IPS_GetName(47359);
// Gibt z.B. zurück: "Meine Tischlampe"

// Sonderfall bei unbenannten Ereignissen (ab Version 4.0)
echo IPS_GetName(48864 /*[TestSkript\Unbenanntes Objekt (ID:
48864)]*/);
// Gibt z.B. zurück: "Bei Variablenänderung der Variable
"Testordner\IRTrans USB/LAN/WLAN\Fernbedienung"
```

# IPS_GetObject

```
array IPS_GetObject (int $ObjektID)
```

## PARAMETERLISTE

| | |
|---|---|
| ObjektID | Die zu zeigende ObjektID |

## RÜCKGABEWERT

Folgende Informationen stehen als **key => value** Paare zur Verfügung:

| INDEX | TYP | BESCHREIBUNG |
|---|---|---|
| **ChildrenIDs** | array | Untergeordnete ObjektIDs. Siehe: IPS_GetChildrenIDs |
| **HasChildren** | boolean | TRUE, wenn das Objekt Unterobjekte hat. Siehe: IPS_HasChildren |
| **ObjectID** | integer | ObjektID |
| **ObjectIcon** | string | Dateiname ohne Erweiterung des Icons. Siehe: IPS_SetIcon |
| **ObjectIdent** | string | Identifikator des Objekts. Siehe IPS_SetIdent |
| **ObjectInfo** | string | Beschreibung, die der Benutzer angeben kann. Siehe: IPS_SetInfo |
| **ObjectIsDisabled** (ab 4.0) | boolean | TRUE, wenn das Objekt inaktiv gesetzt ist. Siehe IPS_SetDisabled |
| **ObjectIsHidden** | boolean | TRUE, wenn das Objekt in der Visualisierung versteckt sein soll. Siehe IPS_SetHidden |
| **ObjectIsReadOnly** | boolean | TRUE, wenn das Objekt schreibgeschützt ist. (Wird zur Zeit nur für Status Variablen benutzt) |
| **ObjectName** | string | Name des Objekts. Siehe: IPS_SetName |
| **ObjectPosition** | integer | Position, die das Objekt in der Visualisierung haben soll. Dieser Wert muss nicht unbedingt eindeutig sein. Siehe: IPS_SetPosition |
| **ObjectType** | integer | Objekt-Typ (0: Kategorie, 1: Instanz, 2: Variable, 3: Skript, 4: Ereignis, 5: Media, 6: Link) |
| **ObjectSummary** | string | Kurzbeschreibung eines Objektes, welches vom Modul zur Laufzeit bei Bedarf generiert wird. |
| **ParentID** | integer | Übergeordnetes Objekt. 0 = Kein übergeordnetes Objekt. Siehe: IPS_SetParent |

## BESCHREIBUNG

Die Funktion liefert ein Array mit Informationen über das Objekt mit der ID **ObjektID**.

Sollte das Objekt nicht vorhanden sein, wird eine Warnung generiert.

## BEISPIEL

```
print_r(IPS_GetObject(19668));
/* liefert z.B.:
Array
(
[ParentID] => 26691
[ObjectID] => 43502
[ObjectType] => 1
[ObjectIdent] =>
[ObjectName] => MX Gerät (ID: 1)
[ObjectInfo] => Dies ist eine Instanz von einem Gerät
[ObjectIcon] => IPS
[ObjectSummary] =>
[ObjectPosition] => 1
[ObjectIsReadOnly] =>
[ObjectIsHidden] =>
[ObjectIsDisabled] =>
[ObjectIsLocked] =>
[HasChildren] => 1
[ChildrenIDs] => Array
(
[0] => 43430
[1] => 55892
[2] => 36597
[3] => 18993
[4] => 34524
[5] => 12269
[6] => 44450
[7] => 23394
[8] => 11693
[9] => 13654
)
)
*/
```

# IPS_GetObjectIDByIdent

Benötigt: IP-Symcon >= 2.5

```
int IPS_GetObjectIDByIdent (string $ObjektIdent, int
$ÜbergeordneteID)
```

## PARAMETERLISTE

| | |
|---|---|
| ObjektIdent | Zu suchender ObjektIdentifikator |
| ÜbergeordneteID | Objekt, in dessen direkten Unterobjekten gesucht werden soll |

## RÜCKGABEWERT

ID des gefundenen Objekts, sonst FALSE

## BESCHREIBUNG

Die Funktion versucht die ID des Objekts mit dem Identifikator **ObjektIdent**, dessen übergeordnetes Objekt die ID **ÜbergeordneteID** besitzt, zu ermitteln. Falls ein solches Objekt gefunden wurde, wird dessen ID zurückgegeben, andernfalls FALSE.



Der Parameter "ObjektIdent" darf kein Leerstring ("") sein.



__Unterschied zu IPS_GetObjectIDByName:__ Da ObjektIdents innerhalb einer Ebene in IP-Symcon eindeutig sind, wird entweder eine einzelne ID zurückgegeben oder eine Warnung generiert.



Sollte der Ident nicht gefunden werden, generiert diese Funktion eine **Warnung**. Es muss der @ Operator genutzt werden, falls diese Meldung unterdrückt werden soll. Die genau Funktionsweise dieses Operators kann dem PHP Handbuch entnommen werden.

## BEISPIEL

```
//Diese Funktion kann als Ersatz für IPS_StatusVariableExists
```

```
verwendet werden.
//Beispiel für IPS_StatusVariableExists($id, "StatusVariable");
echo !(@IPS_GetObjectIDByIdent("StatusVariable", $id) === false);

//Ferner kann diese Funktion auch ein Ersatz für
IPS_GetStatusVariableID darstellen.
//Beispiel für IPS_GetStatusVariableID($id, "StatusVariable");
echo IPS_GetObjectIDByIdent("StatusVariable", $id);


$id = IPS_GetObjectIDByIdent($VariableIdent, $InstanceID);
$v = IPS_GetVariable($id);

// Array ab 4.0
$sv = Array(
"VariableID" => $id,
"VariableIdent" => $VariableIdent,
"VariableName" => "N/A",
"VariablePosition" => 0,
"VariableProfile" => $v['VariableProfile'],
"VariableType" => $v['VariableType'],
"VariableHasAction" => ($v['VariableAction'] > 0),
"VariableUseAction" => ($v['VariableAction'] > 0)
);

// Array bis 3.4
$sv = Array(
"VariableID" => $id,
"VariableIdent" => $VariableIdent,
"VariableName" => "N/A",
"VariablePosition" => 0,
"VariableProfile" => $v['VariableProfile'],
"VariableType" => $v['VariableValue']['ValueType'],
"VariableHasAction" => ($v['VariableAction'] > 0),
"VariableUseAction" => ($v['VariableAction'] > 0)
);
```

# IPS_GetObjectIDByName

```
int IPS_GetObjectIDByName (string $ObjektName, int $ÜbergeordneteID)
```

## PARAMETERLISTE

| | |
|---|---|
| ObjektName | Zu suchender ObjektName |
| ÜbergeordneteID | Objekt, in dessen direkten Unterobjekten gesucht werden soll |

## RÜCKGABEWERT

ID des gefundenen Objekts, sonst FALSE

## BESCHREIBUNG

Die Funktion versucht die ID des Objekts mit dem Namen **ObjektName**, dessen übergeordnetes Objekt die ID **ÜbergeordneteID** besitzt, zu ermitteln. Es wird die ID des ersten gefundenen Objekts verwendet, dessen Name mit **ObjektName** übereinstimmt. Falls ein solches Objekt gefunden wurde, wird dessen ID zurückgemeldet, andernfalls FALSE.



Der Parameter "ObjektName" darf kein Leerstring ("") sein.



**Vorsicht:** Da Namen in IP-Symcon beliebig vergeben werden können, und damit **nicht eindeutig** sind, kann die Funktion eine falsche ID zurückmelden.



Diese Funktion generiert eine **Warnung**. Sollte der Name nicht gefunden werden, muss der @ Operator genutzt, falls diese Meldung unterdrückt werden soll. Die genau Funktionsweise dieses Operators kann dem PHP Handbuch entnommen werden.

## BEISPIEL

```
$ObjektID = @IPS_GetObjectIDByName("Regenerfassung", $ParentID);
if ($ObjektID === false)
```

```
echo "Objekt nicht gefunden!";
else
echo "Die Objekt-ID lautet: ". $ObjektID;
```

# IPS_GetObjectList

```
array IPS_GetObjectList ()
```

## RÜCKGABEWERT

Ein Array von Integer-Werten aller IDs in IP-Symcon

## BESCHREIBUNG

Die Funktion ermittelt die IDs aller in IP-Symcon registrierten Objekte. Die IDs werden in einem Array aufgelistet.

## BEISPIEL

```
print_r(IPS_GetObjectList());

/* liefert z.B.:
Array
(
[0] => 0
[1] => 10573
[2] => 11363
[3] => 11650
[4] => 14114
usw. ...
)
*/
```

# IPS_GetParent

```
int IPS_GetParent (int $ObjektID)
```

## PARAMETERLISTE

| | |
|---|---|
| ObjektID | ID für die die ParentID zurückgegeben werden soll |

## RÜCKGABEWERT

ParentID des Objekts im logischen Objektbaum

## BESCHREIBUNG

Die Funktion gibt die ParentID eines Objektes zurück.

## BEISPIEL

```
echo IPS_GetParent(47359);
//gibt z.B. zurück: 0 für die Hauptkategorie
```

# IPS_HasChildren

```
boolean IPS_HasChildren (int $ObjektID)
```

## PARAMETERLISTE

| | |
|---|---|
| ObjektID | ID des zu prüfenden Objekts |

## RÜCKGABEWERT

TRUE, falls das Objekte Unterobjekte hat, andernfalls FALSE

## BESCHREIBUNG

Die Funktion prüft, ob ein Objekt mit der ID **ObjektID** untergeordnete Objekte hat.

## BEISPIEL

```
if (IPS_HasChildren(0))
echo "Das Root Objekt hat untergeordnete Objekte!";
```

# IPS_IsChild

```
boolean IPS_IsChild (int $ObjektID, int $ParentID, boolean
$Rekursiv)
```

## PARAMETERLISTE

| | |
|---|---|
| ObjektID | ID des zu prüfenden Objekts |
| ParentID | ID des vermeintlichen Elternobjekts |
| Rekursiv | False, wenn nur eine Ebene geprüft werden soll, sonst True |

## RÜCKGABEWERT

TRUE, wenn ObjektID Kind von ParentID ist, sonst FALSE

## BESCHREIBUNG

Die Funktion prüft, ob ein Objekt mit der ID **ObjektID** dem Objekt **ParentID** untergeordnet ist. Ist der Parameter **Rekursiv** aktiviert, werden alle untergeordneten Objekte auf allen untergeordneten Ebenen überprüft. Andernfalls werden nur die direkt untergeordneten Objekte überprüft.

## BEISPIEL

```
if (IPS_IsChild(0, 12345, true))
echo "Das Objekt 12345 ist dem Root Objekt untergeordnet!";
```

# IPS_ObjectExists

```
boolean IPS_ObjectExists (int $ObjektID)
```

## PARAMETERLISTE

| | |
|---|---|
| ObjektID | Die zu prüfende ObjektID |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion prüft, ob ein Objekt mit der ID **ObjektID** existiert.

## BEISPIEL

```
if (IPS_ObjectExists(34881))
echo "Ein Objekt mit dieser ID existiert!";
```

# IPS_SetDisabled

Benötigt: IP-Symcon >= 4.0

```
boolean IPS_SetDisabled (int $ObjektID, boolean $Inaktiv)
```

## PARAMETERLISTE

| | |
|---|---|
| ObjektID | ID des zu verändernden Objekts |
| Inaktiv | TRUE wenn das Objekt inaktiv gesetzt werden soll; FALSE wenn es aktiv gesetzt werden soll. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion setzt das Objekt mit der ID **ObjektID** inaktiv, d.h. es ist noch in den Visualisierungen sichtbar aber nicht mehr klick-/nutzbar.

## BEISPIEL

```
// Das Objekt wird inaktiv gesetzt.
IPS_SetDisabled(47381, true);
```

# IPS_SetHidden

Benötigt: IP-Symcon >= 2.1

```
boolean IPS_SetHidden (int $ObjektID, boolean $Versteckt)
```

## PARAMETERLISTE

| | |
|---|---|
| ObjektID | ID des zu verändernden Objekts |
| Versteckt | TRUE, wenn unsichtbar |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion setzt die Sichtbarkeit des Objekts mit der ID **ObjektID**. Unsichtbare Objekte werden samt Unterobjekten von der Visualisierung ausgeschlossen und in der Konfiguration als grau hinterlegt dargestellt.

## BEISPIEL

```
IPS_SetHidden(47381, true); //Objekt verstecken
```

# IPS_SetIcon

Benötigt: IP-Symcon >= 2.1

```
boolean IPS_SetIcon (int $ObjektID, string $Icon)
```

## PARAMETERLISTE

| | |
|---|---|
| ObjektID | ID des zu verändernden Objekts |
| Icon | Dateiname des Icons ohne Pfad/Erweiterung |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion ordnet dem Objekt mit der ID **ObjektID** ein Icon zu.



Eine Liste der verfügbaren Icons können Sie unter Icons finden.

## BEISPIEL

```
IPS_SetIcon(47381, "Tree");
```

# IPS_SetIdent

Benötigt: IP-Symcon >= 2.5

```
boolean IPS_SetIdent (int $ObjektID, string $Ident)
```

## PARAMETERLISTE

| | |
|---|---|
| ObjektID | ID des zu verändernden Objekts |
| Ident | Neuer Identifikator für das Objekt |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion ordnet dem Objekt mit der ID **ObjektID** den Identifikator **Ident** zu. Der Identifikator ist ein für den Endbenutzer nicht sichtbarer String. Erlaubte Zeichen sind "a..z","A..Z","0..9","_". Dieser kann genutzt werden, um Objekte innerhalb einer Ebene auffindbar zu machen, ohne die eindeutige und systemabhängige ID zu nutzen. Über den Identifikator werden z.B. alle Statusvariablen der Instanzen realisiert. Die Funktion kann auf alle Objekte angewandt werden, die eine ID zur Identifikation besitzen.



Es kann nur ein Objekt mit demselben Identifikator innerhalb einer Ebene existieren. Somit ist eine Abfrage über IPS_GetObjectIDByIdent eindeutig.

## BEISPIEL

```
IPS_SetIdent(47381, "TEMPERATURE_3");
```

# IPS_SetInfo

```
boolean IPS_SetInfo (int $ObjektID, string $Info)
```

## PARAMETERLISTE

| | |
|---|---|
| ObjektID | ID des zu verändernden Objekts |
| Info | Neuer Beschreibungstext für das Objekt |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion ordnet dem Objekt mit der ID **ObjektID** die Beschreibung **Info** zu. Diese können benutzt werden, um z.B. spezifische Konfigurationen oder spezielle Informationen pro Objekt zu hinterlegen, die zu einem späteren Zeitpunkt wiederverwendet werden können.

## BEISPIEL

```
IPS_SetInfo(47381, "USB Soundkarte – Port A, Keller – Grünes
Kabel");
```

# IPS_SetName

```
boolean IPS_SetName (int $ObjektID, string $Name)
```

## PARAMETERLISTE

| | |
|---|---|
| ObjektID | ID des zu verändernden Objekts |
| Name | Neuer Name für das Objekt |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Die Funktion ordnet dem Objekt mit der ID **ObjektID** den Namen **Name** zu. Der Name ersetzt im Strukturbaum den bisherigen Namen (Standard: unbekanntes Objekt: #12345) und kann so zur besseren Lesbarkeit beitragen. Die Funktion kann auf alle Objekte angewandt werden, die eine ID zur Identifikation besitzen.

## BEISPIEL

```
// Setzt den angezeigten Namen des Objekts mit der ID 12345 auf
"Sicherheitsvorkehrungen"
IPS_SetName(12345, "Sicherheitsvorkehrungen");
```

# IPS_SetParent

```
boolean IPS_SetParent (int $ObjektID, int $ÜbergeordneteID)
```

## PARAMETERLISTE

| | |
|---|---|
| ObjektID | ID des zu verändernden Objekts |
| ÜbergeordneteID | ID des neuen übergeordneten Objekts im logischen Objektbaum |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Die Funktion verschiebt das Objekt mit der ID **ObjektID** im IP-Symcon Strukturbaum unter das Objekt mit der ID **ÜbergeordneteID**. Das Objekt ist danach dem Übergeordneten-Objekt optisch untergeordnet. Die neue Zuordnung wird im logischen Strukturbaum (Logical Treeview) sichtbar. Auf diese Weise kann die logische Ansicht so angepasst werden, dass sie bestmöglich die Zusammengehörigkeit und/oder Abhängigkeit verschiedener Komponenten widerspiegelt.
Durch die Zuweisung an ein neues Übergeordnetes-Objekt ändert sich nichts an bestehenden physikalischen Verbindungen. Sie dient lediglich einer besseren Darstellungsweise.
Falls das Objekt einen 'Ident' besitzt, darf kein gleicher 'Ident' an der neuen Zielposition existieren.

## BEISPIEL

```
IPS_SetParent(47381, 15361);
```

# IPS_SetPosition

Benötigt: IP-Symcon >= 2.1

```
boolean IPS_SetPosition (int $ObjektID, int $Position)
```

## PARAMETERLISTE

| | |
|---|---|
| ObjektID | ID des zu verändernden Objekts |
| Position | Positionswert des Objekts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion ordnet dem Objekt mit der ID **ObjektID** eine **Position** zu. Die Position wird in der Visualisierung und Verwaltungskonsole verwendet, um die Objekte in der gewünschten Reihenfolge anzuzeigen. Defaultwert ist Null. Der Positionswert muss nicht eindeutig sein. Sollten zwei Positionen denselben Wert haben, wird primär nach Typ und dann nach Namen sortiert.

## BEISPIEL

```
IPS_SetPosition(47381, 5);
```

# Programminformationen

| | |
|---|---|
| IPS_FunctionExists | prüft, ob eine bestimmte Funktion existiert |
| IPS_GetFunction | liefert umfassende Informationen über eine Funktion |
| IPS_GetFunctionList | liefert eine Liste mit allen Verfügbaren Funktionsnamen |
| IPS_GetFunctionListByModuleID | liefert eine Liste mit allen verfügbaren Funktionen eines Moduls |
| IPS_GetKernelArchitecture | liefert die Architektur für die IP-Symcon erstellt wurde |
| IPS_GetKernelDate | liefert das Datum der Erstellung des Kernels |
| IPS_GetKernelDir | liefert den Pfad zum IP-Symcon Programmverzeichnis |
| IPS_GetKernelDirEx | liefert den Pfad zum IP-Symcon Installationsverzeichnis |
| IPS_GetKernelDirSpace | Liefert freien/verfügbaren/gesamt Speicherplatz auf dem Laufwerk, auf dem alle Daten von IP-Symcon gespeichert werden |
| IPS_GetKernelPlatform | liefert das Betriebssystem auf dem IP-Symcon läuft |
| IPS_GetKernelRevision | liefert eine eindeutige Revisionskennung |
| IPS_GetKernelRunlevel | liefert den Runlevel des Kernels |
| IPS_GetKernelStartTime | liefert den Zeitpunkt, an dem IP-Symcon gestartet wurde |
| IPS_GetKernelVersion | liefert die Versionsnummer des verwendeten Kernels |
| IPS_GetLogDir | liefert den Pfad zum IP-Symcon Logverzeichnis |
| IPS_GetSystemLanguage | liefert die Sytemsprache des Betriebssystems auf dem IP-Symcon läuft |
| IPS_GetUpdateChannel | liefert den aktuellen konfigurierten Update-Kanal zurück |
| IPS_LogMessage | schreibt eine benutzerdefinierte Meldung in die Log-Datei |

# IPS_FunctionExists

Benötigt: IP-Symcon >= 2.6

```
boolean IPS_FunctionExists (string $Funktionsname)
```

## PARAMETERLISTE

| | |
|---|---|
| Funktionsname | Name, der zu prüfenden Funktion |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Die Funktion prüft, ob die Funktion mit dem Namen Funktionsname innerhalb von IP-Symcon registriert ist.

## BEISPIEL

```
if (IPS_FunctionExists("GetValue"))
echo "Die Funktion GetValue existiert!";
```

# IPS_GetFunction

Benötigt: IP-Symcon >= 2.6

```
array IPS_GetFunction (string $Funktionsname)
```

## PARAMETERLISTE

| | |
|---|---|
| Funktionsname | Name, der zu prüfenden Funktion |

## RÜCKGABEWERT

Ein Array mit folgenden **key => value** Paaren

| INDEX | TYP | BESCHREIBUNG |
|---|---|---|
| **FunctionName** | string | Funktionsname |
| **Parameters** | array | siehe Tabelle "Parameterinformationen" |
| **Result** | array | siehe Tabelle "Rückgabeinformationen" |

*Tabelle: Parameterinformationen*

| INDEX | TYP | BESCHREIBUNG |
|---|---|---|
| Description | string | Name des Rückgabewert |
| Enumeration | array | Namentliche Darstellung der möglichen Integerwerte |
| Type_ | integer | Variablentyp des Parameters/Rückgabewerts: 0=Boolean, 1=Integer, 2=Float, 3=String, 4=Variant, 5=Array |

*Tabelle: Rückgabeinformationen*

| INDEX | TYP | BESCHREIBUNG |
|---|---|---|
| Description | string | Name des Rückgabewert |
| Enumeration | array | Namentliche Darstellung der möglichen Integerwerte |
| Type_ | integer | Variablentyp des Parameters/Rückgabewerts: 0=Boolean, 1=Integer, 2=Float, 3=String, 4=Variant, 5=Array |

## BESCHREIBUNG

Die Funktion liefert weitere Informationen zu einer in IP-Symcon registrierten Funktion. Der Parameter Funktionsname gibt an, welche Funktion inspiziert werden soll.

## BEISPIEL

```
// Information der Funktion IPS_GetFunction und IPS_CreateMedia
```

```
abfragen
var_dump(IPS_GetFunction("IPS_GetFunction"));
var_dump(IPS_GetFunction("IPS_CreateMedia"));

// Ausgabe "IPS_GetFunction"
/*
array(3) {
["FunctionName"]=>
string(15) "IPS_GetFunction"
["Result"]=>
array(3) {
["Type_"]=>
int(5)
["Description"]=>
string(6) "Result"
["Enumeration"]=>
array(0) {
}
}
["Parameters"]=>
array(1) {
[0]=>
array(3) {
["Type_"]=>
int(3)
["Description"]=>
string(12) "FunctionName"
["Enumeration"]=>
array(0) {
}
}
}
*/

// Ausgabe "IPS_CreateMedia"
/*
array(3) {
["FunctionName"]=>
string(15) "IPS_CreateMedia"
["Result"]=>
array(3) {
["Type_"]=>
int(1)
["Description"]=>
string(6) "Result"
["Enumeration"]=>
array(0) {
}
}
["Parameters"]=>
array(1) {
[0]=>
array(3) {
["Type_"]=>
int(1)
```

```
["Description"]=>
string(9) "MediaType"
["Enumeration"]=>
array(6) {
[0]=>
string(6) "mtForm"
[1]=>
string(7) "mtImage"
[2]=>
string(7) "mtSound"
[3]=>
string(8) "mtStream"
[4]=>
string(7) "mtChart"
[5]=>
string(10) "mtDocument"
}
}
}
}
*/
```

# IPS_GetFunctionList

Benötigt: IP-Symcon >= 2.6

```
array IPS_GetFunctionList (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | InstanzID, für zu suchende Funktion. 0, um alle zu suchen. |

## RÜCKGABEWERT

Ein Array von Strings aller Funktionsnamen

## BESCHREIBUNG

Die Funktion ermittelt alle in der aktuellen Konfiguration von IP-Symcon verfügbaren IPS-Funktionen. Die Funktionen werden in einem Array aufgelistet. Die Anzahl der Funktionen ist abhängig von der derzeitigen Konfiguration von IP-Symcon. Wenn z.B. eine Instanz des TTS-Moduls (Text-To-Speech) konfiguriert ist, dann werden zusätzlich auch alle TTS-Funktionen aufgeführt.
Als Parameter kann die InstanzID als Filter angegeben werden. Somit werden nur die Funktionen angezeigt, die die betreffende Instanz veröffentlicht. Wird als InstanzID 0 angegeben, so ist der Filtermechanismus nicht aktiv und es werden alle in IP-Symcon verfügbaren Funktionen ausgegeben.
Details zu den einzelnen Funktionen können über IPS_GetFunction ermitteln werden.

## BEISPIEL

```
$alleFunktionen = IPS_GetFunctionList(0);
print_r(IPS_GetFunction($alleFunktionen[48])); // nur Funktion 48
ausgeben (IPS_CreateScript)

/* liefert:
Array
(
[FunctionName] => IPS_CreateScript
[Parameters] => Array
(
[0] => Array
(
[Description] => ScriptType
[Type_] => 1
)
```

```
)
[Result] => Array
(
[Description] => Result
[Type_] => 1
)
)
*/

//Exportiert alle IP-Symcon Funktionen mit einer Parameterliste
$instanceid = 0; //0 = Alle Funktionen, sonst Filter auf InstanzID
$fs = IPS_GetFunctionList($instanceid);
asort($fs);
$typestr = Array("boolean", "integer", "float", "string", "variant",
"array");
foreach($fs as $f) {
$f = IPS_GetFunction($f);
echo sprintf("[%7s]", $typestr[$f['Result']['Type_']]) . " - ".
$f['FunctionName']."(";
$a = Array();
foreach($f['Parameters'] as $p) {
if(isset($p['Enumeration']) && sizeof($p['Enumeration']) > 0) {
$b=Array();
foreach($p['Enumeration'] as $k => $v) {
$b[] = $k."=".$v;
}
$type = "integer/enum[".implode(", ", $b)."]";
} else {
$type = $typestr[$p['Type_']];
}
$a[]=$type." $".$p['Description'];
}
echo implode(", ", $a).");\n";
}
```

# IPS_GetFunctionListByModuleID

```
array IPS_GetFunctionListByModuleID (string $ModuleID)
```

## PARAMETERLISTE

| | |
|---|---|
| ModuleID | GUID des Moduls |

## RÜCKGABEWERT

Ein Array von Strings aller Funktionen

## BESCHREIBUNG

Die Funktion ermittelt alle verfügbaren Funktionen des Moduls mit der GUID **ModuleID**.
Die Funktionen werden in einem Array aufgelistet.
Details zu dem einzelnen Modul kann über IPS_GetModule ermittelt werden.
Details zu den einzelnen Funktionen können über IPS_GetFunction ermitteln werden.

## BEISPIEL

```
// Alle Funktionen von "DPT 200.x - Binary Value & Status" von KNX
print_r(IPS_GetFunctionListByModuleID("{04ED3286-C982-436F-
A01E-8A03C2E976A8}"));

/* liefert:
Array
(
[0] => IPS_RequestAction
[1] => IPS_StopSearch
[2] => IPS_StartSearch
[3] => IPS_IsSearching
[4] => KNX_WriteDPT200
[5] => IPS_GetConfigurationForm
[6] => IPS_SupportsSearching
[7] => KNX_RequestStatus
[8] => IPS_GetConfiguration
[9] => IPS_GetProperty
[10] => IPS_SetProperty
[11] => IPS_SetConfiguration
[12] => IPS_GetConfigurationForParent
[13] => IPS_HasChanges
[14] => IPS_ResetChanges
[15] => IPS_ApplyChanges
```

```
[16] => IPS_GetReferenceList
[17] => IPS_Translate
[18] => KNX_RenameVariables
)
*/
```

# IPS_GetKernelArchitecture

Benötigt: IP-Symcon >= 6.0

```
string IPS_GetKernelArchitecture ()
```

## RÜCKGABEWERT

Die Architektur für die IP-Symcon erstellt wurde

## BESCHREIBUNG

Die Funktion liefert einen String mit der Architektur für die IP-Symcon erstellt wurde

## BEISPIEL

```
echo IPS_GetKernelArchitecture();
// Beispielausgabe:
/*
// MacOS, Windows 32Bit
i386

// Linux, Windows 64Bit
amd64

// Linux, Raspberry Pi
armhf

// Linux, Raspberry Pi
arm64
*/
```

# IPS_GetKernelDate

Benötigt: IP-Symcon >= 4.4

```
int IPS_GetKernelDate ()
```

## RÜCKGABEWERT

Das Erstelldatum in UnixTimeStamp

## BESCHREIBUNG

Die Funktion liefert einen Integerwert, welcher als UnixTimeStamp das Erstelldatum des Kernels ausgibt.

## BEISPIEL

```
echo IPS_GetKernelDate();
// Beispielausgabe
// 1511046903
```

# IPS_GetKernelDir

```
string IPS_GetKernelDir ()
```

## RÜCKGABEWERT

Der vollständige Pfad zum IP-Symcon Programmverzeichnis

## BESCHREIBUNG

Die Funktion liefert einen String mit dem vollständigen Pfad zum IP-Symcon Programmverzeichnis (Settings, Skripte, Medien...). Je nach Betriebssystem enthält der Pfad ein abschließendes Slash ("/") oder Backslash ("\").

## BEISPIEL

```
echo IPS_GetKernelDir();
// Beispielausgabe:
/*
// Windows
// ab Version 5.3
C:\ProgramData\Symcon\
// bis Version 5.2
C:\Programme\IP-Symcon\

// Linux, RaspberryPi
/var/lib/symcon/

// MacOS
/Library/Application Support/Symcon/
*/
```

# IPS_GetKernelDirEx

Benötigt: IP-Symcon >= 4.0

```
string IPS_GetKernelDirEx ()
```

## RÜCKGABEWERT

Der vollständige Pfad zum IP-Symcon Installationsverzeichnis

## BESCHREIBUNG

Die Funktion liefert einen String mit dem vollständigen Pfad zu den IP-Symcon Installationsverzeichnis. Je nach Betriebssystem enthält der Pfad ein abschließendes Slash ("/") oder Backslash ("\").

## BEISPIEL

```
echo IPS_GetKernelDirEx();

// Beispielausgabe:
/*
// Windows
C:\Programme\IP-Symcon\

// Linux, RaspberryPi
/usr/share/symcon/

// MacOS
/Applications/Symcon.app/Contents/Service/
*/
```

# IPS_GetKernelDirSpace

Benötigt: IP-Symcon >= 6.2

```
array IPS_GetKernelDirSpace ()
```

## RÜCKGABEWERT

Liefert gesamten/freien/verfügbaren Speicherplatz

## BESCHREIBUNG

Die Funktion liefert ein Array mit den Keys Total/Free/Available. Diese geben den gesamten/freien/verfügbaren Speicherplatz von dem Laufwerk an, auf dem alle Daten von IP-Symcon gespeichert werden.

## BEISPIEL

```
print_r(IPS_GetKernelDirSpace());
// Beispielausgabe:
/*
Array
(
[Total] => 511582613504
[Free] => 96859312128
[Available] => 96859312128
)
*/
```

# IPS_GetKernelPlatform

Benötigt: IP-Symcon >= 4.4

```
string IPS_GetKernelPlatform ()
```

## RÜCKGABEWERT

Das Betriebssystem auf dem IP-Symcon läuft

## BESCHREIBUNG

Die Funktion liefert einen String mit dem Betriebssystem auf dem der IP-Symcon Server läuft

## BEISPIEL

```
echo IPS_GetKernelPlatform();
// Beispielausgabe:
/*
// SymBox
SymBox

// Windows
Windows

// Linux
Ubuntu
Ubuntu (Docker)

// RaspberryPi
Raspberry Pi
Raspberry Pi (Docker)

// MacOS
Mac
*/
```

# IPS_GetKernelRevision

Benötigt: IP-Symcon >= 4.4

```
string IPS_GetKernelRevision ()
```

## RÜCKGABEWERT

Eindeutige Revisionskennung

## BESCHREIBUNG

Die Funktion liefert einen String mit einer eindeutigen Revisionskennung.

## BEISPIEL

```
echo IPS_GetKernelRevision();
// Beispielausgabe:
// 8edcb65b27506d687dd6a4d1cee470f86e2d89e1
```

# IPS_GetKernelRunlevel

Benötigt: IP-Symcon >= 4.0

```
int IPS_GetKernelRunlevel ()
```

## RÜCKGABEWERT

Runlevel des Kernels.

## BESCHREIBUNG

Die Funktion liefert den Runlevel des Systems zurück.

| RUNLEVEL | RÜCKGABEWERT | BESCHREIBUNG |
|---|---|---|
| KR_CREATE | 10101 | Kernel wird erstellt |
| KR_INIT | 10102 | Kernel wird initialisiert. z.B. werden Module geladen und Instanzen erstellt |
| KR_READY | 10103 | Kernel ist bereit und läuft |
| KR_UNINIT | 10104 | Kernel wird heruntergefahren und alles sauber beendet |
| KR_SHUTDOWN | 10105 | Kernel wurde vollständig beendet |

## BEISPIEL

```
echo IPS_GetKernelRunlevel(); // liefert z.B. für KR_READY: 10103
```

# IPS_GetKernelStartTime

Benötigt: IP-Symcon >= 4.0

```
int IPS_GetKernelStartTime ()
```

## RÜCKGABEWERT

Zeitpunkt, an dem IP-Symcon gestartet wurde

## BESCHREIBUNG

Die Funktion liefert Datum und Uhrzeit des Startzeitpunkts von IP-Symcon als Unix Timestamp.

## BEISPIEL

```
// liefert z.B. 1204675012
echo IPS_GetKernelStartTime();
// liefert z.B. "2008.03.04 23:56:52"
echo date("Y.m.d H:i:s", IPS_GetKernelStartTime());
```

# IPS_GetKernelVersion

```
string IPS_GetKernelVersion ()
```

## RÜCKGABEWERT

Programmversion. Major/Minorversion durch Punkt (".") getrennt.

## BESCHREIBUNG

Die Funktion liefert einen String mit der Nummer der IP-Symcon Programmversion. Durch Abfrage der Programmversion kann entschieden werden, ob ein Skript unter der gemeldeten Version lauffähig ist.

## BEISPIEL

```
echo IPS_GetKernelVersion(); // liefert z.B.: "2.00"
```

# IPS_GetLogDir

Benötigt: IP-Symcon >= 4.0

```
string IPS_GetLogDir ()
```

## RÜCKGABEWERT

Der vollständige Pfad zum IP-Symcon Logverzeichnis

## BESCHREIBUNG

Die Funktion liefert einen String mit dem vollständigen Pfad zum IP-Symcon Logverzeichnis. Je nach Betriebssystem enthält der Pfad ein abschließendes Slash (“/”) oder Backslash (“\”).

## BEISPIEL

```
echo IPS_GetLogDir();
// Beispielausgabe:
/*
//Windows
"C:\Programme\IP-Symcon\logs\"

// Linux, RaspberryPi
"/var/log/symcon/"

// MacOS
"/Library/Logs/Symcon"
*/
```

# IPS_GetSystemLanguage

Benötigt: IP-Symcon >= 6.1

```
string IPS_GetSystemLanguage ()
```

## RÜCKGABEWERT

Die Systemsprache des Betriebssystems auf dem IP-Symcon läuft

## BESCHREIBUNG

Die Funktion liefert einen String mit der Systemsprache des Betriebssystems auf dem der IP-Symcon Server läuft zurück.

## BEISPIEL

```
echo IPS_GetSystemLanguage();
// Beispielausgabe:
/*
// Deutsch
de_DE

// Englisch
en_US
*/
```

# IPS_GetUpdateChannel

Benötigt: IP-Symcon >= 6.1

```
string IPS_GetUpdateChannel ()
```

## RÜCKGABEWERT

aktuell konfigurierter Update-Kanal

## BESCHREIBUNG

Die Funktion liefert einen String mit dem aktuell konfigurierten Update-Kanal zurück.

## BEISPIEL

```
echo IPS_GetUpdateChannel(); //liefert z.B.: stable, beta oder
testing
```

# IPS_LogMessage

```
boolean IPS_LogMessage (string $Sender, string $Meldung)
```

## PARAMETERLISTE

| | |
|---|---|
| Sender | Der angezeigte Absender der Meldung |
| Meldung | Die anzuzeigende Meldung |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion schreibt unter Angabe des Absenders **Sender** den String **Meldung** in die IP-Symcon Log-Datei. Der String **Sender** gibt dabei den Verursacher der Meldung an, z.B. den Namen des Skripts. Die Ausgabe der Meldung kann als “Custom Message” live im Meldungsfenster beobachtet werden.

## BEISPIEL

```
$SkriptStart = microtime(true);
//hier viel berechnen...
$SkriptLaufzeit = microtime(true) - $SkriptStart;
IPS_LogMessage($_IPS['SELF'], "Laufzeit beträgt ". $SkriptLaufzeit.
"sek");
```

# Skriptverwaltung

| | |
|---|---|
| IPS_CreateScript | legt ein neues Skript an |
| IPS_DeleteScript | entfernt eine bestehendes Skript aus der Skriptliste |
| IPS_GetScript | liefert umfassende Informationen über ein Skript |
| IPS_GetScriptContent | liefert den Inhalt eines Skripts |
| IPS_GetScriptEventList | liefert alle Ereignisse die einem Skript zugeordnet sind |
| IPS_GetScriptFile | ermittelt den Dateinamen zur SkriptID |
| IPS_GetScriptIDByFile | liefert die ID eines Skripts andhand der Datei |
| IPS_GetScriptIDByName | liefert die ID eines Skripts andhand des Namens |
| IPS_GetScriptList | liefert eine Aufstellung aller existierenden Skripte |
| IPS_GetScriptTimer | ermittelt den Startwert des ScriptTimers |
| IPS_ScriptExists | prüft, ob ein bestimmtes Skript existiert |
| IPS_SetScriptContent | setzt den Inhalt eines Skripts |
| IPS_SetScriptFile | ordnet einem Skript einen Dateinamen zu |
| IPS_SetScriptTimer | setzt den Startwert des ScriptTimers |

# IPS_CreateScript

```
int IPS_CreateScript (int $SkriptTyp)
```

## PARAMETERLISTE

| WERT | BESCHREIBUNG |
|---|---|
| SkriptTyp | |
| 0 | Erzeugt ein **PHP Skript** |
| 1 | Erzeugt einen **Ablaufplan** |
| 2 | Erzeugt einen **IPSWorkflow** |

## RÜCKGABEWERT

ID des neu erstellten Skripts

## BESCHREIBUNG

Die Funktion legt ein Skript vom Typ **SkriptTyp** an.
Die Funktion liefert eine ID, mit deren Hilfe das erzeugte Skript eindeutig identifiziert werden kann. Die Funktion erzeugt auch automatisch eine passende Skriptdatei, die normalerweise vom Format SkriptID.ips.php ist. Falls Sie eine andere Datei zuordnen möchten, kann dies über die Funktion IPS_SetScriptFile getan werden. Es sollte in dem Zusammenhang auch die alte Skriptdatei gelöscht werden, um unnötige Skriptdateien zu vermeiden.

## BEISPIEL

```
$ScriptID = IPS_CreateScript(0);
echo"Die Skript-ID lautet: ". $ScriptID;
```

# IPS_DeleteScript

```
boolean IPS_DeleteScript (int $SkriptID, boolean $DateiLöschen)
```

## PARAMETERLISTE

| | |
|---|---|
| SkriptID | ID des zu löschenden Skripts |
| DateiLöschen | TRUE, wenn Datei endgültig gelöscht werden soll; FALSE, wenn die Datei in den 'deleted' Ordner verschoben werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Die Funktion entfernt das Skript mit der ID SkriptID aus der internen Skriptliste. Das Skript erscheint nicht mehr im Strukturbaum und kann auch nicht mehr bearbeitet werden. Nicht endgültig gelöschte, aber verschobene, Skripte werden in den "/scripts/deleted/" Ordner verschoben.



Das Skript darf nur gelöscht/verschoben werden, wenn keine Unterobjekte mehr vorhanden sind.

## BEISPIEL

```
// Skript mitsamt Datei endgültig löschen
IPS_DeleteScript($ScriptID, true);
```

# IPS_GetScript

```
array IPS_GetScript (int $SkriptID)
```

## PARAMETERLISTE

| | |
|---|---|
| SkriptID | Die zu zeigende SkriptID |

## RÜCKGABEWERT

Folgende Informationen stehen als **key => value** Paare zur verfügung:

| INDEX | TYP | BESCHREIBUNG |
|---|---|---|
| **IsBroken** (bis 3.4) | boolean | TRUE, wenn beim letzten Skriptaufruf Fehler aufgetreten sind, andernfalls FALSE (ab 4.0 ersetzt durch ScriptIsBroken) |
| **LastExecute** (bis 3.4) | integer | Unix Timestamp des letzten Aufrufs (ab 4.0 ersetzt durch ScriptExecuted) |
| **ScriptCRC** (ab 5.1) | string | CRC32 der Datei |
| **ScriptIsBroken** (ab 4.0) | boolean | TRUE, wenn beim letzten Skriptaufruf Fehler aufgetreten sind, andernfalls FALSE |
| **ScriptExecuted** (ab 4.0) | integer | Unix Timestamp des letzten Aufrufs |
| **ScriptFile** | string | Dateiname des Skripts |
| **ScriptID** | integer | SkriptID |
| **ScriptSize** (ab 5.1) | integer | Größe in Byte |
| **ScriptType** | integer | Skript-Typ (0: PHP Skript, 1: Ablaufplan, 2: IPSWorkflow) |
| **ScriptUpdated** (ab 5.1) | integer | Unix Timestamp der letzten Aktualisierung |

## BESCHREIBUNG

Die Funktion liefert ein Array mit Informationen über das Skript mit der ID **SkriptID**.



Ab Version 4.0 wird die bis 3.4 vorhandene Funktion als "array IPS_GetScriptCompatibility( integer $SkriptIDID )" zur Verfügung gestellt.

## BEISPIEL

```
// Ab Version 5.1
print_r(IPS_GetScript(46413));
/* Beispielausgabe:
Array
(
[ScriptID] => 46413
[ScriptType] => 0
[ScriptFile] => 46413.ips.php
[ScriptUpdated] => 1542622765
[ScriptExecuted] => 1542295981
[ScriptCRC] => 8132DEB6
[ScriptSize] => 42
[ScriptIsBroken] =>
)
*/

// Ab Version 4.0
print_r(IPS_GetScript(46413));
/* Beispielausgabe:
Array
(
[ScriptIsBroken] =>
[ScriptExecuted] => 1204933792
[ScriptFile] => 46413.ips.php
[ScriptID] => 46413
[ScriptType] => 0
)
*/

print_r(IPS_GetScriptCompatibility(46413));
/* Beispielausgabe:
Array
(
[IsBroken] =>
[LastExecute] => 1204933792
[ScriptFile] => 46413.ips.php
[ScriptID] => 46413
[ScriptType] => 0
)
*/


// Bis Version 3.4
print_r(IPS_GetScript(46413));
/* Beispielausgabe:
Array
(
[IsBroken] =>
[LastExecute] => 1204933792
[ScriptFile] => 46413.ips.php
[ScriptID] => 46413
[ScriptType] => 0
)
*/
```

# IPS_GetScriptContent

Benötigt: IP-Symcon >= 3.1

```
string IPS_GetScriptContent (int $SkriptID)
```

## PARAMETERLISTE

| | |
|---|---|
| SkriptID | ID des Skripts, dessen Inhalt geliefert werden soll |

## RÜCKGABEWERT

Der Inhalt des Skripts

## BESCHREIBUNG

Diese Funktion liefert den Inhalt für das Skript mit der ID **SkriptID**. Dabei wird der Inhalt aus der im Skript-Objekt definierten Datei ausgelesen.



Ein praktisches Beispiel für diese Funktion ist das Einlesen eines Skripts in einen externen Editor, welcher z.B. über die JSON-RPC API keinen Zugriff auf das Dateisystem hat.

## BEISPIEL

```
$Inhalt = IPS_GetScriptContent($ScriptID);
```

# IPS_GetScriptEventList

```
array IPS_GetScriptEventList (int $SkriptID)
```

## PARAMETERLISTE

| | |
|---|---|
| SkriptID | SkriptID, zu der verknüpfte Ereignisse gesucht werden sollen |

## RÜCKGABEWERT

Ein Array von Integer-Werten aller EreignisIDs, die mit dem Skript **SkriptID** verknüpft sind

## BESCHREIBUNG

Die Funktion liefert ein Array, dessen Elemente jeweils die IDs zu den Ereignissen des Skripts mit der ID **SkriptID** enthalten. Die Ereignisse können dann mit der Funktion IPS_GetEvent weiterverarbeitet werden.

## BEISPIEL

```
$events = IPS_GetScriptEventList($_IPS['SELF']);
print_r($events); // ermittelt alle Ereignisse des aktuellen Skripts
```

# IPS_GetScriptFile

```
string IPS_GetScriptFile (int $SkriptID)
```

## PARAMETERLISTE

| | |
|---|---|
| SkriptID | ID des Skripts, dem der Dateiname zugeordnet werden soll |

## RÜCKGABEWERT

Der Dateiname des Skripts

## BESCHREIBUNG

Die Funktion gibt den Dateinamen vom Skript mit der ID **SkriptID** zurück. Dadurch kann die Datei z.B. direkt in andere Skripte inkludiert werden.

## BEISPIEL

```
include(IPS_GetScriptFile($ScriptID));
```

# IPS_GetScriptIDByFile

```
int IPS_GetScriptIDByFile (string $Dateipfad)
```

## PARAMETERLISTE

| | |
|---|---|
| Dateipfad | Relativer Dateipfad vom script-Order gesehen |

## RÜCKGABEWERT

ID des gefundenen Skripts, sonst FALSE

## BESCHREIBUNG

Die Funktion versucht die ID des Skripts herausfinden, welches zur Skriptdatei **Dateipfad** zugeordnet ist. Falls ein solches Skript gefunden wurde, wird dessen ID zurückgemeldet, andernfalls FALSE.



Diese Funktion generiert eine **Warnung**, sollte der Name nicht gefunden werden. Es muss der @ Operator genutzt werden, falls diese Meldung unterdrückt werden soll. Die genaue Funktionsweise dieses Operators kann dem PHP Handbuch entnommen werden.

## BEISPIEL

```
$ScriptID = @IPS_GetScriptIDByFile("12345.ips.php");
if ($ScriptID === false)
echo "Skriptdatei nicht gefunden!";
else
echo "Die Skript-ID lautet: ". $ScriptID;
```

# IPS_GetScriptIDByName

```
int IPS_GetScriptIDByName (string $SkriptName, int $ÜbergeordneteID)
```

## PARAMETERLISTE

| | |
|---|---|
| SkriptName | zu suchender SkriptName |
| ÜbergeordneteID | Objekt in deren direkten Unterobjekten gesucht werden soll |

## RÜCKGABEWERT

ID des gefundenen Skripts, sonst FALSE

## BESCHREIBUNG

Die Funktion versucht die ID des Skripts mit dem Namen **SkriptName**, dessen übergeordnetes Objekt die ID **ÜbergeordneteID** besitzt, zu ermitteln. Es wird die ID des ersten gefundenen Skripts verwendet, dessen Name mit **SkriptName** übereinstimmt. Falls ein solches Skript gefunden wurde, wird dessen ID zurückgemeldet, andernfalls FALSE. Da hier im Gegensatz zu IPS_GetScriptID ein zusätzliches Kriterium (ÜbergeordneteID) bei der Suche verwendet wird, ist die Wahrscheinlichkeit die richtige ID zu finden erheblich größer. Trotzdem gilt:

**Vorsicht:** Da Namen in IP-Symcon beliebig vergeben werden können, und damit **nicht eindeutig** sind, kann die Funktion eine falsche ID zurückmelden.

Diese Funktion generiert eine **Warnung**, sollte der Name nicht gefunden werden. Es muss der @ Operator genutzt werden, falls diese Meldung unterdrückt werden soll. Die genaue Funktionsweise dieses Operators kann dem PHP Handbuch entnommen werden.

## BEISPIEL

```
$SkriptID = @IPS_GetScriptIDByName("Regenerfassung",
$ÜbergeordneteID);
if ($SkriptID === false)
echo "Skript nicht gefunden!";
else
```

```
echo "Die Skript-ID lautet: ". $SkriptID;
```

# IPS_GetScriptList

```
array IPS_GetScriptList ()
```

## RÜCKGABEWERT

Ein Array von Integer-Werten aller IDs der Skripte in IP-Symcon

## BESCHREIBUNG

Die Funktion ermittelt die IDs aller in IP-Symcon registrierten Skripte. Die IDs werden in einem Array aufgelistet. Falls kein Skript existiert, ist das Array leer.

## BEISPIEL

```
$alleSkripte = IPS_GetScriptList();
print_r($alleSkripte);

/* liefert z.B.:
Array
(
[0] => 37659
[1] => 18326
usw. ...
*/
```

# IPS_GetScriptTimer

```
int IPS_GetScriptTimer (int $SkriptID)
```

## PARAMETERLISTE

| | |
|---|---|
| SkriptID | ID des Skripts, dessen Timer ausgelesen werden soll |

## RÜCKGABEWERT

Zeit in Sekunden, die der Timer das Skript zyklisch aufruft. (Es ist nicht die Restzeit.)

## BESCHREIBUNG

Die Funktion liefert einen Integer-Wert mit dem Startwert des ScriptTimers des Skripts mit der ID **SkriptID**. Der Wert wird in Sekunden angegeben. Mit dieser Funktion kann man prüfen, auf welchen Wert der ScriptTimer eines Skripts gesetzt ist und ob dieser korrekt ist.
Steht der Wert auf 0, so ist der Timer abgeschaltet und es erfolgt kein Timer-Event.

## BEISPIEL

```
$TimerWert = IPS_GetScriptTimer($ScriptID); // ermittelt den Wert
des ScriptTimers
```

# IPS_ScriptExists

```
boolean IPS_ScriptExists (int $SkriptID)
```

## PARAMETERLISTE

| | |
|---|---|
| SkriptID | ID der zu prüfenden Skripte |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Die Funktion prüft, ob das Skript mit der ID SkriptID existiert.

## BEISPIEL

```
if (IPS_ScriptExists(34881))
echo "Ein Skript mit dieser ID existiert!";
```

# IPS_SetScriptContent

Benötigt: IP-Symcon >= 3.1

```
boolean IPS_SetScriptContent (int $SkriptID, string $Inhalt)
```

## PARAMETERLISTE

| | |
|---|---|
| SkriptID | ID des Skripts, dem der Inhalt zugeordnet werden soll |
| Inhalt | Inhalt inklusive PHP Tags (<? und ?>) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Diese Funktion setzt den Inhalt für das Skript mit der ID SkriptID. Dabei wird der Dateiname, der standardmäßig definiert ist, genutzt, sofern kein anderer per IPS_SetScriptFile definiert wurde.



Diese Funktion ist mit Vorsicht zu verwenden, da der Inhalt ohne Warnung überschrieben wird.

## BEISPIEL

```
$ScriptID = IPS_CreateScript(0);
IPS_SetName($ScriptID, "Mein Skript");
IPS_SetScriptContent($ScriptID, "<? echo 'Test!'; ?>");
```

# IPS_SetScriptFile

```
boolean IPS_SetScriptFile (int $SkriptID, string $Dateiname)
```

## PARAMETERLISTE

| | |
|---|---|
| SkriptID | ID des Skripts, dem der Dateiname zugeordnet werden soll |
| Dateiname | Dateiname des PHP Skripts (relativ zum "/scripts" Ordner) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion bindet den Dateinamen **DateiName** an das Skript mit der ID **SkriptID**. Normalerweise besteht der Dateiname eines Skripts nur aus dessen ID mit der Extension ".ips.php". Mit dieser Funktion ist es möglich einen anderen Dateinamen zu verwenden. Der Dateiname hat nichts mit dem Skriptnamen zu tun, der im Location Tree zu sehen ist. Skriptname und Dateiname können völlig verschieden sein. Die SkriptID bleibt auch nach der Zuordnung erhalten. Weder der Skriptname noch der Dateiname können direkt zur Identifikation des Skripts verwendet werden. Von IP-Symcon wird hierfür immer nur die numerische SkriptID benutzt.

## BEISPIEL

```
$ScriptName = "Example.ips.php"; //Skriptdatei
$ScriptID = IPS_CreateScript(0);
IPS_SetName($ScriptID, "Example");

$Script = IPS_GetScript($ScriptID);
//Datei umbenennen in $ScriptName
rename(IPS_GetKernelDir().'/scripts/'.$Script['ScriptFile'],
IPS_GetKernelDir().'/scripts/'.$ScriptName);
//Dem Skript den neuen Dateinamen ("Example.ips.php") zuweisen
IPS_SetScriptFile($ScriptID, $ScriptName);
```

# IPS_SetScriptTimer

```
boolean IPS_SetScriptTimer (int $SkriptID, int $TimerWert)
```

## PARAMETERLISTE

| | |
|---|---|
| SkriptID | ID des Skripts, dessen Timer eingestellt werden solll |
| TimerWert | Zeit in Sekunden, in denen das Skript zyklisch aufgerufen werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion setzt den Integer-Startwert des ScriptTimers des Skripts mit der ID **SkriptID** auf den Wert **TimerWert** in Sekunden. Mit dieser Funktion kann man den ScriptTimer auf einen bestimmten Wert setzen. Der Wert wird in Sekundenschritten heruntergezählt. Beim Erreichen des Werts 0 wird das Skript gestartet und der ScriptTimer erneut auf den Startwert gesetzt. Dieser Vorgang wiederholt sich bis der Timer explizit gelöscht (auf 0 gesetzt) wurde.



Diese Funktion funktioniert nur sekundengenau, wenn Werte kleiner als 3600 angegeben werden. Bei Werten über 3600 kann es je nach Startzeit/Teiler zu unregelmäßigkeiten kommen, da der SkriptTimer an die Limitation von normalen Ereignissen und dessen Zeitmuster gebunden ist. z.B. wird ein SkriptTimer, welcher alle 5 Stunden starten soll und um 6h früh gesetzt wird, täglich um 6h, 11h, 16h, 21h und dann wieder um 1h, statt wie vielleicht erwartet um 2h, gestartet. Werte über die Tagesgrenze hinaus werden zu vollen Tagen abgerundet.

Steht der Wert auf 0, so ist der Timer abgeschaltet, und es erfolgt kein Ausführung.

## BEISPIEL

```
//veranlasst, dass das Skript alle 10 Sekunden ausgeführt wird
IPS_SetScriptTimer($ScriptID, 10);
```

# Variablenverwaltung

## IPS_CREATEVARIABLE

legt eine Variable an

## IPS_DELETEVARIABLE

löscht eine Variable

## IPS_GETVARIABLE

liefert umfassende Informationen zu einer bestimmten Variablen

## IPS_GETVARIABLEEVENTLIST

liefert alle Ereignisse die einer Variable zugeordnet sind

## IPS_GETVARIABLEIDBYNAME

liefert die ID einer Variablen anhand des Namens

## IPS_GETVARIABLELIST

liefert eine Liste aller existierenden Variablen

## IPS_GETVARIABLEPRESENTATION

Gibt alle Parameter der aktuell genutzten Darstellung zurück

## IPS_SETVARIABLECUSTOMACTION

ordnet der Variable ein benutzerdefiniertes Aktionsskript zu

## IPS_SETVARIABLECUSTOMPRESENTATION

ordnet der Variable eine benutzerdefinierte Darstellung zu

## IPS_SETVARIABLECUSTOMPROFILE

ordnet der Variable ein benutzerdefiniertes Profil zu

## IPS_VARIABLEEXISTS

prüft, ob eine bestimmte Variable existiert

## VARIABLENDARSTELLUNG

Funktionen in dieser Gruppe dienen der Verwaltung von Variablendarstellungen

| | |
|---|---|
| IPS_GetPresentation | liefert umfassende Informationen über eine bestimmte Darstellung |
| IPS_GetPresentationForm | liefert das Konfigurationsformular einer Darstellung für einen Variablentypen |
| IPS_GetPresentations | liefert alle Darstellungen des Systems |
| IPS_PresentationExists | prüft ob eine bestimmte Darstellung existiert |
| Vorlagenverwaltung | Funktionen in dieser Gruppe dienen der Verwaltung von Vorlagen |

| | |
|---|---|
| IPS_CreateTemplate | erstellt eine neue Vorlage für eine bestimmte Darstellung |
| IPS_DeleteTemplate | löscht eine Vorlage |
| IPS_GetTemplate | liefert umfassende Informationen über eine Vorlage |
| IPS_GetTemplateList | liefert alle verfügbaren Vorlagen |
| IPS_GetTemplateListByPresentation | liefert eine Liste der Vorlagen einer Darstellung |
| IPS_SetTemplateName | setzt den Namen einer Vorlage |
| IPS_SetTemplateValues | setzt die Werte einer Vorlage |
| IPS_TemplateExists | prüft ob eine Vorlage existiert |

## VARIABLENPROFILE

| | |
|---|---|
| IPS_CreateVariableProfile | legt ein Variablenprofil an |
| IPS_DeleteVariableProfile | löscht ein Variablenprofil |
| IPS_GetVariableProfile | liefert umfassende Informationen zu einem bestimmten Variablenprofil |
| IPS_GetVariableProfileList | liefert eine Liste aller existierenden Variablenprofile |
| IPS_GetVariableProfileListByType | liefert eine Liste aller existierenden Variablenprofile eines bestimmten Typs |
| IPS_SetVariableProfileAssociation | modifiziert die Assoziationstabelle eines Variablenprofils |
| IPS_SetVariableProfileDigits | setzt die Anzahl der Nachkommastellen eines Variablenprofils |
| IPS_SetVariableProfileIcon | setzt das Icon eines Variablenprofils |
| IPS_SetVariableProfileText | setzt das Prefix und Suffix eines Variablenprofils |
| IPS_SetVariableProfileValues | setzt den Minimalwert, Maximalwert und die Schrittweite eines Variablenprofils |
| IPS_VariableProfileExists | prüft, ob ein bestimmtes Variablenprofil existiert |

# IPS_CreateVariable

```
int IPS_CreateVariable (int $VariablenTyp)
```

## PARAMETERLISTE

| WERT | BESCHREIBUNG |
|---|---|
| VariablenTyp | |
| 0 | Legt eine Variable vom Typ **Boolean** an |
| 1 | Legt eine Variable vom Typ **Integer** an |
| 2 | Legt eine Variable vom Typ **Float** an |
| 3 | Legt eine Variable vom Typ **String** an |

## RÜCKGABEWERT

ID der neu erstellten Variable

## BESCHREIBUNG

Die Funktion legt eine neue IPS-Variable vom Typ VariablenTyp an.
Die Funktion liefert eine ID, mit deren Hilfe die erzeugte Variable eindeutig identifiziert werden kann.

## BEISPIEL

```
// Anlegen einer Float-Variablen mit dem Namen "Raumtemperatur
Wohnzimmer"
$VarID_Raumtemperatur = IPS_CreateVariable(2);
IPS_SetName($VarID_Raumtemperatur, "Raumtemperatur Wohnzimmer"); //
Variable benennen
IPS_SetParent($VarID_Raumtemperatur, 12345); // Variable
einsortieren unter dem Objekt mit der ID "12345"
```

# IPS_DeleteVariable

```
boolean IPS_DeleteVariable (int $VariablenID)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenID | ID der zu löschenden Variable |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Die Funktion löscht eine vorhandene Variable mit der ID VariablenID.



Die Variable darf nur gelöscht werden, wenn keine Unterobjekte mehr vorhanden sind.

## BEISPIEL

```
// Löschen der Variablen 47788
IPS_DeleteVariable(47788);
```

# IPS_GetVariable

```
array IPS_GetVariable (int $VariablenID)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenID | Die zu zeigende VariablenID |

## RÜCKGABEWERT

Folgende Informationen stehen als **key => value** Paare zur Verfügung:

| INDEX | TYP | BESCHREIBUNG |
|---|---|---|
| **VariableAction** | integer | ID der Instanz, die bei einer gewünschten Änderungen der Variable den Änderungswunsch entgegen nimmt. 0 bei keiner oder einer deaktivierten Verknüpfung. |
| **VariableChanged** | integer | Unix-Timestamp, wenn die Variable zuletzt verändert wurde |
| **VariableCustomAction** | integer | ID des Skriptes, dass bei einer gewünschten Änderung der Variable über die Visualisierung gestartet werden soll. Sofern aktuell eine vorhandene Standardaktion deaktiviert wird, ist die nichtvorhandene SkriptID Eins (1) angegeben. (Ab Version 2.5 #2300) |
| **VariableCustomProfile** | string | Name vom Benutzer gegebenen Profil. Leer, falls kein Profil angeben. |
| **VariableID** | integer | VariablenID |
| **VariableIsLocked** | boolean | Gibt an, ob diese Variable oberhalb des Variablenlimits ist und deshalb nicht beschrieben werden kann (Verfügbar ab Version 2.3). |
| **VariableProfile** | string | Name vom System gegebenen Profil. Leer, falls kein Profil angegben |
| **VariableType** (ab 4.0) | integer | Enthält den Variablentyp (0: Boolean, 1: Integer, 2: Float, 3: String) |
| **VariableUpdated** | integer | Unix-Timestamp, wenn die Variable zuletzt aktualisiert wurde |
| **VariableValue** (bis 3.4) | array | Siehe Tabelle Variablenwert (durch VariableType und GetValue ersetzt) |

*Tabelle: VariablenWert (bis 3.4)*

| INDEX | TYP | BESCHREIBUNG |
|---|---|---|
| **ValueType** | integer | Variablentyp (0: Boolean, 1: Integer, 2: Float, 3: String) |
| **ValueBoolean** | boolean | Wert der Variable abhängig vom **ValueType** |
| **ValueInteger** | integer | Wert der Variable abhängig vom **ValueType** |
| **ValueFloat** | float | Wert der Variable abhängig vom **ValueType** |
| **ValueString** | string | Wert der Variable abhängig vom **ValueType** |

## BESCHREIBUNG

Die Funktion liefert ein Array mit Informationen über die Variable mit der ID **VariablenID**.



Um den Wert einer Variable zu bekommen muss ab Version 4.0 die Funktion GetValue genutzt werden



Ab Version 4.0 wird die bis 3.4 vorhandene Funktion als "array IPS_GetVariableCompatibility( integer $VariablenID )" zur Verfügung gestellt.

## BEISPIEL

```
// Ab Version 4.0
print_r(IPS_GetVariable(40770));
/* Beispielausgabe:
Array
(
[VariableChanged] => 1246039629
[VariableCustomAction] => 0
[VariableCustomProfile] => BoolProfile
[VariableID] => 40770
[VariableIsLocked] =>
[VariableProfile] => ~Switch
[VariableType] => 0
[VariableUpdated] => 1246039629
)
*/

print_r(IPS_GetVariableCompatibility(40770)); // Ab Version 4.0
/* Beispielausgabe:
Array
(
[VariableChanged] => 1246039629
[VariableCustomAction] => 0
[VariableCustomProfile] => BoolProfile
[VariableID] => 40770
```

```
[VariableIsLocked] =>
[VariableProfile] => ~Switch
[VariableUpdated] => 1246039629
[VariableValue] => Array
(
[ValueArray] => Array
(
)

[ValueBoolean] => 1
[ValueFloat] => 0
[ValueIndex] => Array
(
[IndexInt] => 0
[IndexStr] =>
[IndexType] => 0
)

[ValueInteger] => 0
[ValueString] =>
[ValueType] => 0
)
)
*/

// Bis Version 3.4
print_r(IPS_GetVariable(40770));
/* Beispielausgabe:
Array
(
[VariableChanged] => 1246039629
[VariableCustomAction] => 0
[VariableCustomProfile] => BoolProfile
[VariableID] => 40770
[VariableIsLocked] =>
[VariableProfile] => ~Switch
[VariableUpdated] => 1246039629
[VariableValue] => Array
(
[ValueArray] => Array
(
)

[ValueBoolean] => 1
[ValueFloat] => 0
[ValueIndex] => Array
(
[IndexInt] => 0
[IndexStr] =>
[IndexType] => 0
)

[ValueInteger] => 0
[ValueString] =>
[ValueType] => 0
```

```
)
)
*/
```

# IPS_GetVariableEventList

```
array IPS_GetVariableEventList (int $VariablenID)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenID | VariablenID, zu der verknüpfte Ereignisse gesucht werden sollen |

## RÜCKGABEWERT

Ein Array von Integer-Werten aller EreignisIDs, die mit der Variable **VariablenID** verknüpft sind

## BESCHREIBUNG

Die Funktion liefert ein Array, dessen Elemente jeweils die IDs zu den Ereignissen sind, in der die Variable **VariableID** verwendet wird. Die Ereignisse können dann mit der Funktion IPS_GetEvent weiterverarbeitet werden.

## BEISPIEL

```
$events = IPS_GetVariableEventList(12345);
print_r($events); // ermittelt alle Ereignisse des aktuellen Skripts
```

# IPS_GetVariableIDByName

```
int IPS_GetVariableIDByName (string $VariablenName, int
$ÜbergeordneteID)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenName | Zu suchender VariablenName |
| ÜbergeordneteID | Objekt, in deren direkten Unterobjekten gesucht werden soll |

## RÜCKGABEWERT

ID der gefundenen Variable, sonst FALSE

## BESCHREIBUNG

Die Funktion versucht die ID der Variablen mit dem Namen **VariablenName**, deren übergeordnetes Objekt die ID **ÜbergeordneteID** besitzt, zu ermitteln. Es wird die ID der ersten gefundenen Variablen verwendet, deren Name mit **VariablenName** übereinstimmt. Falls eine solche Variable gefunden wurde, wird deren ID zurückgemeldet, andernfalls FALSE.
Da hier im Gegensatz zu IPS_GetVariableID ein zusätzliches Kriterium (ÜbergeordneteID) bei der Suche verwendet wird, ist die Wahrscheinlichkeit die richtige ID zu finden erheblich größer. Trotzdem gilt:



**Vorsicht:** Da Namen in IP-Symcon beliebig vergeben werden können, und damit **nicht eindeutig** sind, kann die Funktion eine falsche ID zurückmelden.



Diese Funktion generiert eine **Warnung**, sollte der Name nicht gefunden werden. Es muss der @ Operator genutzt werden, falls diese Meldung unterdrückt werden soll. Die genaue Funktionsweise dieses Operators kann dem PHP Handbuch entnommen werden.

## BEISPIEL

```
$VariablenID = @IPS_GetVariableIDByName("Regenmenge",
$ÜbergeordneteID);
if ($VariablenID === false)
```

```
echo "Variable nicht gefunden!";
else
echo "Die Variablen-ID lautet: ". $VariablenID;
```

# IPS_GetVariableList

```
array IPS_GetVariableList ()
```

## RÜCKGABEWERT

Ein Array von Integer-Werten aller IDs der Variablen in IP-Symcon

## BESCHREIBUNG

Die Funktion ermittelt die IDs aller in IP-Symcon registrierten IPS-Variablen. Die IDs werden in einem Array aufgelistet. Falls keine Variable existiert, ist das Array leer.

## BEISPIEL

```
$alleVariablen = IPS_GetVariableList();
print_r($alleVariablen);

/* liefert z.B.:
Array
(
[0] => 37659
[1] => 18326
usw. ...
*/
```

# IPS_GetVariablePresentation

Benötigt: IP-Symcon >= 8.1

```
array IPS_GetVariablePresentation (int $VariablenID)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenID | VariablenID der Variable deren Darstellung ausgegeben werden soll |

## RÜCKGABEWERT

Ein Array mit allen Parametern der aktuellen Variablendarstellung

## BESCHREIBUNG

Die Funktion liefert die aktuelle Darstellung der Variable mit **VariablenID**. Die Darstellung liegt als Array vor und beinhaltet alle Darstellungsparameter.

| KEY | BESCHREIBUNG |
|---|---|
| PRESENTATION | Die als GUID formatierte ID der genutzten Darstellung |
| TEMPLATE | Die als GUID formatierte ID verwendeten Vorlage |
| *Darstellungsparameter* | Die Darstellungsparameter unterscheiden sich je nach Darstellung. Die Parameter befinden sich auf der gleichen Ebene wie der PRESENTATION Parameter |

### MÖGLICHE DARSTELLUNGEN

| DARSTELLUNG |
|---|
| Aufzählung |
| Datum/Uhrzeit |
| Dauer |
| Farbe |
| Rollladen |
| Schalter |
| Schieberegler |
| Webinhalt |
| Wertanzeige |
| Werteingabe |

## BEISPIEL

```
print_r(IPS_GetVariablePresentation(12345));
/* Beispielausgabe
Array
(
[DIGITS] => 0
[CUSTOM_GRADIENT] => []
[ICON] => temperature-half
[DECIMAL_SEPARATOR] => Client
[GRADIENT_TYPE] => 1
[MAX] => 25
[PRESENTATION] => {6B9CAEEC-5958-C223-30F7-BD36569FC57A}
[INTERVALS] => []
[INTERVALS_ACTIVE] =>
[MIN] => 15
[PERCENTAGE] =>
[PREFIX] =>
[STEP_SIZE] => 1
[SUFFIX] => °C
[THOUSANDS_SEPARATOR] =>
[USAGE_TYPE] => 0
)
*/
```

# IPS_SetVariableCustomAction

```
boolean IPS_SetVariableCustomAction (int $VariablenID, int
$SkriptID)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenID | ID der Variable, der die Aktion zugeordnet werden soll |
| SkriptID | ID eines Skriptes, das als Aktions aufgeführt werden soll. Sofern eine vorhandene Standardaktion deaktiviert werden soll, kann die nichtvorhandene SkriptID Eins (1) angegeben werden, wodurch die Standardaktion ausgeschaltet wird. Wenn eine gültige SkriptID angegebenen wird, so hat diese Vorrang vor der Standardaktion. Soll die Standardaktion, sofern verfügbar, aktiviert werden, so muss die SkriptID Null (0) angegeben werden. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion vergibt der Variable **VariablenID** eine benutzerdefinierte Aktion, die in der Visualisierung benutzt wird, um eine Änderung der Variable an die richtige Funktion des Moduls weiterzuleiten. Ein Beispiel für ein Aktionsskript kann hier gefunden werden: Aktionsskripte

## BEISPIEL

```
IPS_SetVariableCustomAction(12345, 44431);
```

# IPS_SetVariableCustomPresentation

Benötigt: IP-Symcon >= 8.0

```
boolean IPS_SetVariableCustomPresentation (int $VariablenID, array
$Presentation)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenID | ID der Variable, der die Darstellung zugeordnet werden soll |
| Presentation | Die Konfiguration der Darstellung. Details sind unter dem Abschnitt Darstellungen zu finden. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion vergibt der Variable **VariablenID** die **Darstellung** konfigurierte Darstellung.

## BEISPIEL

```
// Setzt eine Variable auf die Darstellung Schieberegler
IPS_SetVariableCustomPresentation(12345, ['PRESENTATION' =>
VARIABLE_PRESENTATION_SLIDER]);
```

# IPS_SetVariableCustomProfile

Benötigt: IP-Symcon >= 3.0

```
boolean IPS_SetVariableCustomProfile (int $VariablenID, string
$ProfilName)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenID | ID der Variable, der das Profil zugeordnet werden soll |
| ProfilName | Name des Profils. Verfügbare Profile können über IPS_GetVariableProfileList abgefragt werden. Wenn ein Leerstring übergeben wird, dann wird das benutzerdefinierte Profil deaktiviert. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion vergibt der Variable **VariablenID** das benutzerdefinierte Profil **ProfilName**.



Diese Funktion überprüft vor dem Setzen die Existenz des zu setzenden Profils und die Übereinstimmung von Profiltyp und Variablentyp.



Standardprofile, die nicht geändert werden können, fangen mit einer Tilde (~) an.

## BEISPIEL

```
IPS_SetVariableCustomProfile(12345, "~Switch");
```

# IPS_VariableExists

```
boolean IPS_VariableExists (int $VariablenID)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenID | ID der zu prüfenden Variable |

## RÜCKGABEWERT

Falls die VariablenID im System existiert, wird **TRUE** zurückgemeldet, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion prüft, ob die Variable mit der ID **VariablenID** existiert.

## BEISPIEL

```
if (IPS_VariableExists(44788))
echo "Variable existiert bereits!";
```

# Variablendarstellung

## IPS_GETPRESENTATION

liefert umfassende Informationen über eine bestimmte Darstellung

## IPS_GETPRESENTATIONFORM

liefert das Konfigurationsformular einer Darstellung für einen Variablentypen

## IPS_GETPRESENTATIONS

liefert alle Darstellungen des Systems

## IPS_PRESENTATIONEXISTS

prüft ob eine bestimmte Darstellung existiert

## VORLAGENVERWALTUNG

Funktionen in dieser Gruppe dienen der Verwaltung von Vorlagen

| | |
|---|---|
| IPS_CreateTemplate | erstellt eine neue Vorlage für eine bestimmte Darstellung |
| IPS_DeleteTemplate | löscht eine Vorlage |
| IPS_GetTemplate | liefert umfassende Informationen über eine Vorlage |
| IPS_GetTemplateList | liefert alle verfügbaren Vorlagen |
| IPS_GetTemplateListByPresentation | liefert eine Liste der Vorlagen einer Darstellung |
| IPS_SetTemplateName | setzt den Namen einer Vorlage |
| IPS_SetTemplateValues | setzt die Werte einer Vorlage |
| IPS_TemplateExists | prüft ob eine Vorlage existiert |

# IPS_GetPresentation

Benötigt: IP-Symcon >= 8.0

```
array IPS_GetPresentation (string $DarstellungsID)
```

## PARAMETERLISTE

| | |
|---|---|
| DarstellungsID | Die ID der Darstellung, als GUID formatiert |

## RÜCKGABEWERT

Folgende Informationen stehen als **key => value** Paare zur Verfügung:

| INDEX | TYP | BESCHREIBUNG |
|---|---|---|
| caption | string | Der Anzeigename der Darstellung. |
| chart | array | Ein Array mit den Zuweisungen der Parameter für die Darstellung eines Diagramms (Siehe chart Tabelle) |
| condition | array | Eine Liste an Bedingungen welche die Parameter der Darstellung erfüllen müssen. |
| conversions | array | Eine Liste von Anweisungen die angewandt werden, um Parameter für verschiedene Variablentypen anzupassen |
| form | array/ string | Das Konfigurationsformular bestimmt wie die Parameter im Variablen-Dialog oder dem Vorlagenmanager angepasst werden können. |
| format | array/ string | Die Formatierung von Werten innerhalb dieser Darstellung |
| id | string | Die ID der Darstellung, als GUID formatiert |
| locale | array | Die Lokalisierung der Darstellung |
| presentationParameters | array | Eine Liste, welche die Standardwerte der Parameter dieser Darstellung beschreibt |
| presentationValue | string | Der Standartwert welcher in der Darstellungsvorschau angezeigt wird. Kann alternativ auch PHP-Code als String oder Array enthalten, welcher den Wert zurückgibt |
| restrictions | array | Eine Liste an Einschränkungen, welche festlegt für welche Variablen diese Darstellung verfügbar ist |

chart Tabelle
Jeder Wert kann entweder eine Zahl sein oder der Name des Parameters, dessen Wert verwendet werden soll.

| INDEX | TYP | BESCHREIBUNG |
| --- | --- | --- |
| min | integer/float/string | Parameter für den Minimalwert |
| max | integer/float/string | Parameter für den Maximalwert |
| stepSize | integer/float/string | Parameter für die Schrittweite |

## BESCHREIBUNG

Die Funktion liefert ein Array mit Informationen über die Variablendarstellung mit der ID DarstellungsID.

## BEISPIEL

```
IPS_GetPresentation('{6B9CAEEC-5958-C223-30F7-BD36569FC57A}' /*
Schieberegler*/);
```

# IPS_GetPresentationForm

Benötigt: IP-Symcon >= 8.0

```
string IPS_GetPresentationForm (string $DarstellungsID, int
$Variablentyp, array $Parameter)
```

## PARAMETERLISTE

| | |
|---|---|
| DarstellungsID | Die ID der Darstellung, als GUID formatiert |
| Variablentyp | Der Typ der Variablen deren Formular zurückgegeben werden soll |
| Parameter | Die Parameterlist, die für das Formular genutzt werden soll |

## RÜCKGABEWERT

Das Formular der beschriebenen Darstellung als JSON kodierter String.

## BESCHREIBUNG

Die Funktion gibt das Formular für die **DarstellungsID** unter Berücksichtigung des **Variablentyp** und der **Parameter**.

## BEISPIEL

```
IPS_GetPresentationForm('{6B9CAEEC-5958-C223-30F7-BD36569FC57A}' /*
Schieberegler*/, 1 /* Integer */, []);
```

# IPS_GetPresentations

Benötigt: IP-Symcon >= 8.0

```
string IPS_GetPresentations ()
```

## RÜCKGABEWERT

Ein JSON-kodieret Liste aller verfügbaren Darstellungen.
Die Felder der einzelnen Darstellungen sind hier näher beschrieben.

## BESCHREIBUNG

Gibt eine JSON-kodieret Liste aller verfügbaren Darstellungen zurück.

## BEISPIEL

```
// Alle Darstellung als Array
$presentations = json_decode(IPS_GetPresentations(), true);
```

# IPS_PresentationExists

Benötigt: IP-Symcon >= 8.0

```
boolean IPS_PresentationExists (string $DarstellungsID)
```

## PARAMETERLISTE

| | |
|---|---|
| DarstellungsID | Die ID der zu prüfenden Darstellung, als GUID formatiert |

## RÜCKGABEWERT

Falls die Darstellung im System existiert, wird **TRUE** zurückgemeldet, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion prüft, ob die Darstellung mit der ID **DarstellungsID** existiert.

## BEISPIEL

```
if (IPS_PresentationExists('{60AE6B26-B3E2-BDB1-A3A1-BE232940664B}'
/* Schieberegler */)) {
echo 'Darstellung existiert bereits!';
}
```

# Vorlagenverwaltung

| | |
|---|---|
| IPS_CreateTemplate | erstellt eine neue Vorlage für eine bestimmte Darstellung |
| IPS_DeleteTemplate | löscht eine Vorlage |
| IPS_GetTemplate | liefert umfassende Informationen über eine Vorlage |
| IPS_GetTemplateList | liefert alle verfügbaren Vorlagen |
| IPS_GetTemplateListByPresentation | liefert eine Liste der Vorlagen einer Darstellung |
| IPS_SetTemplateName | setzt den Namen einer Vorlage |
| IPS_SetTemplateValues | setzt die Werte einer Vorlage |
| IPS_TemplateExists | prüft ob eine Vorlage existiert |

# IPS_CreateTemplate

Benötigt: IP-Symcon >= 8.0

```
string IPS_CreateTemplate (string $DarstellungsID)
```

## PARAMETERLISTE

| | |
|---|---|
| DarstellungsID | Die ID der Darstellung, als GUID formatiert, für die ein neue Vorlage angelegt werden soll |

## RÜCKGABEWERT

Die ID der neu erstellten Vorlage als GUID formatiert

## BESCHREIBUNG

Erstellt eine neue Vorlage für die Darstellung mit der ID **DarstellungsID**. Die ID der Vorlage wird zurückgegeben.

## BEISPIEL

```
$vorlage = IPS_CreateTemplate('{6B9CAEEC-5958-C223-30F7-
BD36569FC57A}' /* Schieberegler */);
```

# IPS_DeleteTemplate

Benötigt: IP-Symcon >= 8.0

```
boolean IPS_DeleteTemplate (string $VorlagenID)
```

## PARAMETERLISTE

| | |
|---|---|
| VorlagenID | Die ID der zu löschenden Vorlage, als GUID formatiert |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion löscht die Vorlage mit der ID **VorlagenID**.

## BEISPIEL

```
// Löscht die Vorlage mit der GUID {8422E41F-00AA-33F7-E8A4-
BB516EC0056B}
IPS_DeleteTemplate('{8422E41F-00AA-33F7-E8A4-BB516EC0056B}');
```

# IPS_GetTemplate

Benötigt: IP-Symcon >= 8.0

```
array IPS_GetTemplate (string $VorlagenID)
```

## PARAMETERLISTE

| | |
|---|---|
| VorlagenID | Die ID der Vorlage, als GUID formatiert |

## RÜCKGABEWERT

Folgende Informationen stehen als **key=>value** Paare zur Verfügung:

| INDEX | TYP | BESCHREIBUNG |
|---|---|---|
| TemplateID | string | Die ID der Vorlage |
| PresentationID | string | Die ID der Darstellung zu der diese Vorlage gehört |
| DisplayName | string | Der Anzeigename der Vorlage |
| Values | array | Ein Array mit den Werten der Vorlage |
| ReadOnly | bool | Ob die Vorlage verändert werden kann |

## BESCHREIBUNG

Die Funktion liefert einen Array mit Informationen über die Vorlage mit der ID **VorlagenID**.

## BEISPIEL

```
// Liefert Informationen über die Raumtemperatur Vorlage
print_r(IPS_GetTemplate('{868B087E-A38D-2155-EBE0-157AFBBF9E8C}' /*
Raumptemperatur */);
/* Beispielausgabe:
Array
(
[TemplateID] => {868B087E-A38D-2155-EBE0-157AFBBF9E8C}
[PresentationID] => {6B9CAEEC-5958-C223-30F7-BD36569FC57A}
[DisplayName] => Raumtemperatur
[Values] => Array
(
[MIN] => 15
[DIGITS] => 1
[CUSTOM_GRADIENT] => []
```

```
[DECIMAL_SEPARATOR] => Client
[GRADIENT_TYPE] => 1
[MAX] => 25
[INTERVALS] => []
[INTERVALS_ACTIVE] =>
[PERCENTAGE] =>
[PREFIX] =>
[STEP_SIZE] => 0.5
[SUFFIX] => °C
[THOUSANDS_SEPARATOR] =>
[USAGE_TYPE] => 0
)

[IsReadOnly] => 1
)
*/
```

# IPS_GetTemplateList

Benötigt: IP-Symcon >= 8.0

```
array IPS_GetTemplateList ()
```

## RÜCKGABEWERT

Eine Liste der IDs aller verfügbaren Vorlagen

## BESCHREIBUNG

Die Funktion gibt die IDs aller verfügbaren Vorlagen in einem Array zurück. Mit der Funktion IPS_GetTemplate können mehr Informationen über die einzelnen Vorlagen abgerufen werden.

## BEISPIEL

```
print_r(IPS_GetTemplateList());
```

# IPS_GetTemplateListByPresentation

Benötigt: IP-Symcon >= 8.0

```
array IPS_GetTemplateListByPresentation (string $DarstellungsID)
```

## PARAMETERLISTE

| | |
|---|---|
| DarstellungsID | Die ID der Darstellung deren Vorlagen aufgelistet werden sollen |

## RÜCKGABEWERT

Ein Array der IDs einer Darstellung

## BESCHREIBUNG

Die Funktion liefert die IDs aller Vorlagen der Darstellung mit der ID **DarstellungsID**. Mit der Funktion IPS_GetTemplate können mehr Informationen über die Einzelnen Vorlagen abgerufen werden.

## BEISPIEL

```
print_r(IPS_GetTemplateListByPresentation('{6B9CAEEC-5958-C223-30F7-
BD36569FC57A}' /* Schieberegler */));
```

# IPS_SetTemplateName

Benötigt: IP-Symcon >= 8.0

```
boolean IPS_SetTemplateName (string $VorlagenID, string $Name)
```

## PARAMETERLISTE

| | |
|---|---|
| VorlagenID | Die ID der Vorlage deren Name angepasst werden soll |
| Name | Der Wert auf den der Name der Vorlage gesetzt werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion setzt den Namen der Vorlage mit der ID **VorlagenID** auf den Wert **Name**.

## BEISPIEL

```
IPS_SetTemplateName('{3C257416-99FE-0386-D8D3-20437FF62CEB}' , 'Neue
Vorlage');
```

# IPS_SetTemplateValues

Benötigt: IP-Symcon >= 8.0

```
boolean IPS_SetTemplateValues (string $VorlagenID, array $Werte)
```

## PARAMETERLISTE

| | |
|---|---|
| VorlagenID | Die ID der Vorlage deren Werte angepasst werden sollen |
| Werte | Ein Array auf welchen die Werte der Vorlage gesetzt werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion setzt die Werte der Vorlage mit der ID **VorlagenID** auf den Wert **Werte**

## BEISPIEL

```
IPS_SetTemplateValues('{E35A875A-E9BC-4C6D-A550-F4DA423BCF18}',
['PROFILE' => '~Intensity.100']);
```

# IPS_TemplateExists

Benötigt: IP-Symcon >= 8.0

```
boolean IPS_TemplateExists (string $VorlagenID)
```

## PARAMETERLISTE

| | |
|---|---|
| VorlagenID | ID der zu prüfenden Vorlage als GUID formatiert |

## RÜCKGABEWERT

Falls eine Vorlage mit der **VorlagenGUID** existiert, wird TRUE zurückgemeldet, andernfalls FALSE.

## BESCHREIBUNG

Die Funktion prüft ob die Vorlage mit der ID **VorlagenID** existiert.

## BEISPIEL

```
if (IPS_TemplateExists('{868B087E-A38D-2155-EBE0-157AFBBF9E8C}' /*
Raumtemperatur */)) {
echo 'Vorlage existiert bereits!';
}
```

# Variablenprofile

| | |
|---|---|
| IPS_CreateVariableProfile | legt ein Variablenprofil an |
| IPS_DeleteVariableProfile | löscht ein Variablenprofil |
| IPS_GetVariableProfile | liefert umfassende Informationen zu einem bestimmten Variablenprofil |
| IPS_GetVariableProfileList | liefert eine Liste aller existierenden Variablenprofile |
| IPS_GetVariableProfileListByType | liefert eine Liste aller existierenden Variablenprofile eines bestimmten Typs |
| IPS_SetVariableProfileAssociation | modifiziert die Assoziationstabelle eines Variablenprofils |
| IPS_SetVariableProfileDigits | setzt die Anzahl der Nachkommastellen eines Variablenprofils |
| IPS_SetVariableProfileIcon | setzt das Icon eines Variablenprofils |
| IPS_SetVariableProfileText | setzt das Prefix und Suffix eines Variablenprofils |
| IPS_SetVariableProfileValues | setzt den Minimalwert, Maximalwert und die Schrittweite eines Variablenprofils |
| IPS_VariableProfileExists | prüft, ob ein bestimmtes Variablenprofil existiert |

# IPS_CreateVariableProfile

```
boolean IPS_CreateVariableProfile (string $ProfilName, int
$VariablenTyp)
```

## PARAMETERLISTE

| | |
|---|---|
| ProfilName | Name des Profils. Erlaubt sind A-Z, a-z, Punkt, Komma, Unterstrich |
| VariablenTyp | |

| WERT | BESCHREIBUNG |
|---|---|
| 0 | Legt ein Variablenprofil für VariablenTyp **Boolean** an |
| 1 | Legt ein Variablenprofil für VariablenTyp **Integer** an |
| 2 | Legt ein Variablenprofil für VariablenTyp **Float** an |
| 3 | Legt ein Variablenprofil für VariablenTyp **String** an |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion legt ein neues Variablenprofile für den Typ **VariablenTyp** an.

## BEISPIEL

```
//Anlegen eines Profils für Boolean-Variablen
IPS_CreateVariableProfile("Schalter", 0);
//... hier die weitere Konfiguration des Profils
```

# IPS_DeleteVariableProfile

```
boolean IPS_DeleteVariableProfile (string $ProfilName)
```

## PARAMETERLISTE

| | |
|---|---|
| ProfilName | Name des zu löschenden Profils |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion löscht ein vorhandenes Variablenprofil mit dem Namen **ProfilName**.

## BEISPIEL

```
//Löschen des Schalter Profils
IPS_DeleteVariableProfile("Schalter");
```

# IPS_GetVariableProfile

```
array IPS_GetVariableProfile (string $ProfilName)
```

## PARAMETERLISTE

| | |
|---|---|
| ProfilName | Name des Variablenprofils |

## RÜCKGABEWERT

Folgende Informationen stehen als **key => value** Paare zur Verfügung:

| INDEX | TYP | BESCHREIBUNG |
|---|---|---|
| Associations | array | Array mit Wert, Name, Icon, Color Paaren (siehe Associations Tabelle) |
| Icon | string | Icon des Variablenprofils |
| IsReadOnly | boolean | TRUE, wenn es ein systemerstelltes Profil ist, welches nicht geändert werden kann. |
| MaxValue | float | Maximalwert für die Visualisierung |
| MinValue | float | Minimalwert für die Visualisierung |
| StepSize | float | Schrittweite für die Visualisierung; 0, wenn Assoziationstabelle genutzt werden soll |
| Digits | integer | Anzahl der Nachkommastellen |
| Prefix | string | Präfix für die Visualisierung |
| Suffix | string | Suffix für die Visualisierung |
| ProfileName | string | Name des Profils (~ = Systemprofil) |
| ProfileType | integer | Typ des Profils (Siehe ProfilTyp Tabelle) |

Associations Tabelle

| INDEX | TYP | BESCHREIBUNG |
|---|---|---|
| Value | float | Wert, zu dem der Name und das Icon verknüpft werden sollen |
| Name | string | Name zum angegebenen Wert |
| Icon | string | Icon zum angegebenen Wert |
| Color | integer | Farbwert im HTML Farbcode (z.b. 0x0000FF für Blau). Sonderfall: -1 für transparent |

ProfilTyp

Profiltyp und der Variablentyp (IPS_CreateVariable) müssen identisch sein, damit eine Anzeige im WebFront korrekt dargestellt werden kann.

| WERT | BESCHREIBUNG |
|---|---|
| 0 | Ein Profil vom Typ **Boolean** |
| 1 | Ein Profil vom Typ **Integer** |
| 2 | Ein Profil vom Typ **Float** |
| 3 | Ein Profil vom Typ **String** |

## BESCHREIBUNG

Die Funktion liefert ein Array mit Informationen über das Variableprofil mit dem Namen **ProfilName**.

## BEISPIEL

```
print_r( IPS_GetVariableProfile("~WindDirection") );

/* liefert z.B.:
Array
(
[Associations] => Array
(
)

[Digits] => 1
[Icon] => WindDirection
[IsReadOnly] => 1
[MaxValue] => 360
[MinValue] => 0
[Prefix] =>
[ProfileName] => ~WindDirection
[ProfileType] => 2
[StepSize] => 60
[Suffix] => °
)
*/
```

# IPS_GetVariableProfileList

```
array IPS_GetVariableProfileList ()
```

## RÜCKGABEWERT

Ein Array von String-Werten aller Namen der Variablenprofile

## BESCHREIBUNG

Die Funktion ermittelt in IP-Symcon vorhandenen Variablenprofile. Die Namen werden in einem Array aufgelistet.

## BEISPIEL

```
$Profile = IPS_GetVariableProfileList();
print_r($Profile);

/* liefert z.B.:
Array
(
[0] => ~Temperature
[1] => ~Humidity
[2] => ~AirPressure
usw. ...
*/
```

# IPS_GetVariableProfileListByType

```
array IPS_GetVariableProfileListByType (int $VariablenTyp)
```

## PARAMETERLISTE

| | WERT | BESCHREIBUNG |
|---|---|---|
| VariablenTyp | 0 | Sucht vom Typ **Boolean** |
| | 1 | Sucht vom Typ **Integer** |
| | 2 | Sucht vom Typ **Float** |
| | 3 | Sucht vom Typ **String** |

## RÜCKGABEWERT

Ein Array von String-Werten aller Namen der Variablenprofile

## BESCHREIBUNG

Die Funktion ermittelt in IP-Symcon vorhandenen Variablenprofile vom Typ **Variablentyp**. Die Namen werden in einem Array aufgelistet.

## BEISPIEL

```
$Profile = IPS_GetVariableProfileListByType(0);
print_r($Profile);

/* liefert z.B.:
Array
(
[0] => ~Switch
[1] => ~Alert
[2] => ~Alert.Reversed
usw. ...
*/
```

# IPS_SetVariableProfileAssociation

```
boolean IPS_SetVariableProfileAssociation (string $ProfilName,
variant $Wert, string $Name, string $Icon, int $Farbe)
```

## PARAMETERLISTE

| | |
|---|---|
| ProfilName | Name des Profils. Verfügbare Profile können über IPS_GetVariableProfileList abgefragt werden. |
| Wert | Wert, zu dem der Name, das Icon und die Farbe verknüpft werden sollen |
| Name | Name zum angegebenen Wert |
| Icon | Icon zum angegebenen Wert |
| Farbe | Farbwert im HTML Farbcode (z.b. 0x0000FF für Blau). Sonderfall: -1 für transparent |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion setzt den **Namen**, das **Icon** und die **Farbe** für einen **Wert** eines Variablenprofils mit Namen **ProfilName**. Die Farbe wird nicht angezeigt für Profile vom Typ Wertanzeige/Slider (siehe Objekt-Darstellung).



Wenn **Name** und **Icon** leer sind, wird die Assoziation zum angegebenen **Wert** gelöscht. Bei Boolean Profilen kann keine Assoziation gelöscht werden.



Die maximale Anzahl von gleichzeitiger Assoziationen pro Profil ist 128.



Standardprofile, die nicht geändert werden können, fangen mit einer Tilde (~) an.

## BEISPIEL

```
//Anlegen für Wert 1 in der Farbe weiß
IPS_SetVariableProfileAssociation("Temperatur", 1, "Wert 1",
"Speaker", 0xFFFFFF);

//Anlegen für Wert "sum" in der Farbe gelb
IPS_SetVariableProfileAssociation("Jahreszeit", "sum", "Sommer",
"Sun", 0xFFFF00);

//Löschen für Wert 1
IPS_SetVariableProfileAssociation("Temperatur", 1, "", "", -1);
```

# IPS_SetVariableProfileDigits

```
boolean IPS_SetVariableProfileDigits (string $ProfilName, int
$Nachkommastellen)
```

## PARAMETERLISTE

| | |
|---|---|
| ProfilName | Name des Profils. Verfügbare Profile können über IPS_GetVariableProfileList abgefragt werden. |
| Nachkommastellen | Anzahl der in der Visualisierung angezeigten Nachkommastellen |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion setzt die Anzahl der **Nachkommastellen** eines Variablenprofils mit Namen **ProfilName**.



Standardprofile, die nicht geändert werden können, fangen mit einer Tilde (~) an.

## BEISPIEL

```
IPS_SetVariableProfileDigits("Temperatur", 1);
```

# IPS_SetVariableProfileIcon

```
boolean IPS_SetVariableProfileIcon (string $ProfilName, string
$Icon)
```

## PARAMETERLISTE

| | |
|---|---|
| ProfilName | Name des Profils. Verfügbare Profile können über IPS_GetVariableProfileList abgefragt werden. |
| Icon | Icon zum angegebenen Wert |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion setzt das standardmäßig genutzte Icon eines Variablenprofils mit Namen **ProfilName**.



Die Präzedenz der genutzten Icons verhält sich wie unter Icons beschrieben.



Standardprofile, die nicht geändert werden können, fangen mit einer Tilde (~) an.

## BEISPIEL

```
IPS_SetVariableProfileIcon("Temperatur", "Temperature");
```

# IPS_SetVariableProfileText

```
boolean IPS_SetVariableProfileText (string $ProfilName, string
$Präfix, string $Suffix)
```

## PARAMETERLISTE

| | |
|---|---|
| ProfilName | Name des Profils. Verfügbare Profile können über IPS_GetVariableProfileList abgefragt werden. |
| Präfix | Prefix für den Wert |
| Suffix | Suffix für den Wert |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion setzt den Präfix und Suffix eines Variablenprofils mit Namen **ProfilName**.



Standardprofile, die nicht geändert werden können, fangen mit einer Tilde (~) an.

## BEISPIEL

```
IPS_SetVariableProfileText("Schalter", "", "%");
```

# IPS_SetVariableProfileValues

```
boolean IPS_SetVariableProfileValues (string $ProfilName, float $Minimalwert, float $Maximalwert, float $Schrittweite)
```

## PARAMETERLISTE

| | |
|---|---|
| ProfilName | Name des Profils. Verfügbare Profile können über IPS_GetVariableProfileList abgefragt werden. |
| Minimalwert | Der für die Visualisierung genutzte Minimalwert. Diese Soft-Limitation beeinflusst nicht den Variablenwert. |
| Maximalwert | Der für die Visualisierung genutzte Maximalwert. Diese Soft-Limitation beeinflusst nicht den Variablenwert. |
| Schrittweite | Die für die Visualisierung genutzte Schrittweite zur Erstellung der Sollwert-Veränderngsleiste. Eine Schrittweite von **0** aktiviert die Assoziationsliste. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion setzt den **MinimalWert**, **MaximalWert** und die **Schrittweite** eines Variablenprofils mit Namen **ProfilName**.



Standardprofile, die nicht geändert werden können, fangen mit einer Tilde (~) an.

## BEISPIEL

```
IPS_SetVariableProfileValues("Temperatur", -10, 40, 0.5);
```

# IPS_VariableProfileExists

```
boolean IPS_VariableProfileExists (string $ProfilName)
```

## PARAMETERLISTE

| | |
|---|---|
| ProfilName | Name des gesuchten Variablenprofils |

## RÜCKGABEWERT

Falls das Profil mit dem Namen **ProfilName** im System existiert, wird **TRUE** zurückgemeldet, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion prüft, ob das Variablenprofil mit dem Namen **ProfilName** existiert.

## BEISPIEL

```
if (IPS_VariableProfileExists("Temperatur"))
echo "Profil existiert bereits!";
```

# Variablenzugriff

| | |
|---|---|
| GetValue | holt den Wert einer IP-Symcon Variable unabhängig des Typs |
| GetValueBoolean | holt den Wert einer IP-Symcon Variable vom Typ Boolean |
| GetValueFloat | holt den Wert einer IP-Symcon Variable vom Typ Float |
| GetValueFormatted | holt den Wert einer IP-Symcon Variable und formatiert ihn anhand des Profils |
| GetValueFormattedEx | gibt einen Wert formatiert anhand des Profils einer Variable zurück |
| GetValueInteger | holt den Wert einer IP-Symcon Variable vom Typ Integer |
| GetValueString | holt den Wert einer IP-Symcon Variable vom Typ String |
| HasAction | prüft, ob eine Aktion definiert ist |
| RequestAction | führe die Aktion einer IP-Symcon Variable aus |
| RequestActionEx | führe die Aktion einer IP-Symcon Variable mit individuellen Sender aus |
| SetValue | setzt den Wert einer IP-Symcon Variable unabhängig des Typs |
| SetValueBoolean | setzt den Wert einer IP-Symcon Variable vom Typ Boolean |
| SetValueFloat | setzt den Wert einer IP-Symcon Variable vom Typ Float |
| SetValueInteger | setzt den Wert einer IP-Symcon Variable vom Typ Integer |
| SetValueString | setzt den Wert einer IP-Symcon Variable vom Typ String |

# GetValue

```
variant GetValue (int $VariablenID)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenID | ID der Variable |

## RÜCKGABEWERT

Wert der angegebenen Variable im Typ der angegebenen Variable

## BESCHREIBUNG

Die Funktion liefert als Ergebnis den Inhalt der IP-Symcon-Variablen mit der ID **VariablenID**. Der zurückgelieferte Variablentyp ist derjenige, der beim Anlegen der Variablen (IPS_CreateVariable) gewählt wurde. Wurde die Variable als String angelegt, liefert die Funktion also immer einen String.

## BEISPIEL

```
$Fenster_offen = GetValue(47788); // Boolean-Variable: true oder
false
$aktuelles_Projekt = GetValue(46250); // String-Variable: z.B.
"Regenerfassung"
```

# GetValueBoolean

```
boolean GetValueBoolean (int $VariablenID)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenID | ID der Variable |

## RÜCKGABEWERT

Wert der angegebenen Variable

## BESCHREIBUNG

Die Funktion liefert als Ergebnis den Inhalt der IP-Symcon Variablen mit der ID **VariablenID**. Die gewählte Variable muss vom Typ Boolean sein. Wenn nicht, bricht die Funktion mit einem Fehler ab. Eine Konvertierung wird nicht vorgenommen. Für eine typunsichere Abfrage kann die Funktion GetValue genutzt werden.

## BEISPIEL

```
$Fenster_offen = GetValueBoolean(47788); // Boolean-Variable: true
oder false
```

# GetValueFloat

```
float GetValueFloat (int $VariablenID)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenID | ID der Variable |

## RÜCKGABEWERT

Wert der angegebenen Variable

## BESCHREIBUNG

Die Funktion liefert als Ergebnis den Inhalt der IP-Symcon-Variablen mit der ID **VariablenID**. Die gewählte Variable muss vom Typ Float sein. Wenn nicht, bricht die Funktion mit einem Fehler ab. Eine Konvertierung wird nicht vorgenommen. Für eine typunsichere Abfrage können Sie die GetValue Funktion nutzen.

## BEISPIEL

```
$aktuelle_Temperatur = GetValueFloat(47788);
```

# GetValueFormatted

```
string GetValueFormatted (int $VariablenID)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenID | ID der Variable |

## RÜCKGABEWERT

Wert der angegebenen Variable passend zum Profil formatiert

## BESCHREIBUNG

Die Funktion liefert als Ergebnis den Inhalt der IP-Symcon Variablen mit der ID **VariablenID**, nachdem er passend zum Variablenprofil formatiert wurde. Der zurückgelieferte Variablentyp ist dabei durch die Formatierung immer String.

## BEISPIEL

```
echo GetValueFormatted(47788); //Profil = ~Switch
//Ergibt: An oder Aus

echo GetValueFormatted(46250); //Profil = ~Temperature
//Ergibt: 17,5°C
```

# GetValueFormattedEx

Benötigt: IP-Symcon >= 5.5

```
string GetValueFormattedEx (int $VariablenID, variant $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenID | ID der Variable |
| Wert | Wert, welcher formatiert werden soll |

## RÜCKGABEWERT

Wert passend zum Profil der Variable formatiert

## BESCHREIBUNG

Die Funktion liefert als Ergebnis den Inhalt Wert **Wert**, nachdem er passend zum Variablenprofil der Variable mit der ID **VariablenID** formatiert wurde. Der tatsächliche Wert der Variable mit der ID **VariablenID** ist dabei unerheblich. Der zurückgelieferte Variablentyp ist durch die Formatierung immer String.

## BEISPIEL

```
echo GetValueFormattedEx(12345, true); //Profil = ~Switch
//Ergibt: An

echo GetValueFormattedEx(23456, 17.5); //Profil = ~Temperature
//Ergibt: 17,5°C
```

# GetValueInteger

```
int GetValueInteger (int $VariablenID)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenID | ID der Variable |

## RÜCKGABEWERT

Wert der angegebenen Variable

## BESCHREIBUNG

Die Funktion liefert als Ergebnis den Inhalt der IP-Symcon Variablen mit der ID **VariablenID**. Die gewählte Variable muss vom Typ Integer sein. Wenn nicht, bricht die Funktion mit einem Fehler ab. Eine Konvertierung wird nicht vorgenommen. Für eine typunsichere Abfrage kann die Funktion GetValue genutzt werden.

## BEISPIEL

```
$Anzahl_Personen = GetValueInteger(47788);
```

# GetValueString

```
string GetValueString (int $VariablenID)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenID | ID der Variable |

## RÜCKGABEWERT

Wert der angegebenen Variable

## BESCHREIBUNG

Die Funktion liefert als Ergebnis den Inhalt der IP-Symcon Variablen mit der ID **VariablenID**. Die gewählte Variable muss vom Typ String sein. Wenn nicht, bricht die Funktion mit einem Fehler ab. Eine Konvertierung wird nicht vorgenommen. Für eine typunsichere Abfrage kann die Funktion GetValue genutzt werden.

## BEISPIEL

```
$aktuelles_Projekt = GetValueString(47788);
```

# HasAction

Benötigt: IP-Symcon >= 5.3

```
boolean HasAction (int $VariablenID)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenID | ID der Variable |

## RÜCKGABEWERT

TRUE, falls die Variable eine Aktion hat, andernfalls FALSE

## BESCHREIBUNG

Die Funktion prüft die Variable mit der ID **VariablenID** auf eine vorhandene Aktion.



Die Funktion prüft nur auf das Vorhandensein einer Aktion, nicht dessen Gültig- oder Korrektheit

## BEISPIEL

```
// Prüft die Variable mit der ID 12356, ob diese eine Aktion hat.
HasAction(12345);
```

# RequestAction

Benötigt: IP-Symcon >= 5.0

```
boolean RequestAction (int $VariablenID, variant $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenID | ID der Variable |
| Wert | Wert für die Aktion |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion führt die Aktion der Variable mit der ID **VariablenID** mit dem Wert **Wert** aus. Hierbei wird die gleiche Schaltaktion wie im WebFront und anderen Visualisierungen verwendet. Ist für die Variable ein eigenes Aktionsskript definiert, so wird dies Skript ausgeführt. Ansonsten wird, falls vorhanden, die Standardaktion einer Instanzvariable ausgeführt. Existiert weder eine eigene noch eine Standardaktion, so führt die Funtkion RequestAction zu einem Fehler. Üblicherweise schalten Variablenaktionen die dazugehörige Eigenschaft eines Gerätes entsprechend des neuen Wertes und setzen abschließend den Wert der Variablen entsprechend. Kommt es beim Schalten zu einem Fehler, wird dieser ausgegeben. Es ist darauf zu achten, dass der Typ des Wertes zum Typ der Variablen passt. Ist die Zielvariable vom Typ String, so sollte auch Wert vom Typ String sein. Des Weiteren steht im Skript die Systemvariable mit $_IPS["SENDER"] = Action zur Verfügung.



Falls der Typ der Variablen und des Wertes nicht übereinstimmen, versucht PHP eine geeignete Konvertierung vorzunehmen. Das Ergebnis dieser Konvertierung kann aber vom erwarteten Ergebnis abweichen. Es ist daher besser von vornherein den richtigen Typ zu übergeben.



Es muss darauf geachtet werden, dass in einem Aktionsskript kein RequestAction auf sich selbst aufgerufen wird. Dies verursacht eine Endlosschleife.

## BEISPIEL

```
// Die Boolean Variable 29117 auf "true" schalten
RequestAction(29117, true);

// Die Integer Variable 46250 auf 18 schalten
RequestAction(46250, 18);
```

# RequestActionEx

Benötigt: IP-Symcon >= 5.0

```
boolean RequestActionEx (int $VariablenID, variant $Wert, string
$Sender)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenID | ID der Variable |
| Wert | Wert für die Aktion |
| Sender | Name des Sender |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion führt die Aktion der Variable mit der ID **VariablenID** mit dem Wert **Wert** und dem Sender **Sender** aus. Hierbei wird die gleiche Schaltaktion wie im WebFront und anderen Visualisierungen verwendet. Ist für die Variable ein eigenes Aktionsskript definiert, so wird dies Skript ausgeführt. Ansonsten wird, falls vorhanden, die Standardaktion einer Instanzvariable ausgeführt. Existiert weder eine eigene noch eine Standardaktion, so führt die Funtkion RequestAction zu einem Fehler. Üblicherweise schalten Variablenaktionen die dazugehörige Eigenschaft eines Gerätes entsprechend des neuen Wertes und setzen abschließend den Wert der Variablen entsprechend. Kommt es beim Schalten zu einem Fehler, wird dieser ausgegeben. Es ist darauf zu achten, dass der Typ des Wertes zum Typ der Variablen passt. Ist die Zielvariable vom Typ String, so sollte auch Wert vom Typ String sein. Des Weiteren steht im Skript die Systemvariable mit $_IPS["SENDER"] = Action zur Verfügung.



Falls der Typ der Variablen und des Wertes nicht übereinstimmen, versucht PHP eine geeignete Konvertierung vorzunehmen. Das Ergebnis dieser Konvertierung kann aber vom erwarteten Ergebnis abweichen. Es ist daher besser von vornherein den richtigen Typ zu übergeben.

## BEISPIEL

```
// Boolean-Variable mit Sender Alarmanlage einschalten
```

```
RequestActionEx(12345, true, "Alarmanlage");

// Integer-Variable mit Sender TemperaturKlimaanlage einstellen
RequestActionEx(23456, 18, "TemperaturKlimaanlage");
```

# SetValue

```
boolean SetValue (int $VariablenID, variant $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenID | ID der Variable |
| Wert | Neuer Wert der Variable |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion schreibt den neuen Wert **Wert** in die IP-Symcon Variable mit der ID **VariablenID**. Es ist darauf zu achten, dass der Typ des Wertes zum Typ der Variablen passt. Ist die Zielvariable vom Typ String, so sollte auch Wert vom Typ String sein.



Falls der Typ der Variablen und des Wertes nicht übereinstimmen, versucht PHP eine geeignete Konvertierung vorzunehmen. Das Ergebnis dieser Konvertierung kann aber vom erwarteten Ergebnis abweichen. Es ist daher besser von vornherein den richtigen Typ zu übergeben.

## BEISPIEL

```
SetValue(29117, true); // Boolean-Variable: Merker für Schulferien
setzen
SetValue(46250, $aktuelles_Projekt); // String-Variable: z.B.
"Regenerfassung"
```

# SetValueBoolean

```
boolean SetValueBoolean (int $VariablenID, boolean $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenID | ID der Variable |
| Wert | TRUE/FALSE |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion setzt den Inhalt der IP-Symcon Variablen mit der ID **VariablenID**. Die gewählte Variable muss vom Typ Boolean sein. Wenn nicht, bricht die Funktion mit einem Fehler ab. Eine Konvertierung wird nicht vorgenommen. Für ein typunsicheres Setzen der Variable kann die Funktion SetValue genutzt werden.

## BEISPIEL

```
SetValueBoolean(47788, true); // Alarmanlage einschalten
```

# SetValueFloat

```
boolean SetValueFloat (int $VariablenID, float $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenID | ID der Variable |
| Wert | 64Bit Gleitkommazahl |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion setzt den Inhalt der IP-Symcon Variablen mit der ID **VariablenID**. Die gewählte Variable muss vom Typ Float sein. Wenn nicht, bricht die Funktion mit einem Fehler ab. Eine Konvertierung von Integer/Boolean (0/1) wird vorgenommen. Für ein typunsicheres Setzen der Variable kann die Funktion SetValue genutzt werden.

## BEISPIEL

```
SetValueFloat(47788, $aktuelle_Temperatur);
```

# SetValueInteger

```
boolean SetValueInteger (int $VariablenID, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenID | ID der Variable |
| Wert | -2,147,483,648 .. 2,147,483,647 |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion setzt den Inhalt der IP-Symcon Variablen mit der ID **VariablenID**. Die gewählte Variable muss vom Typ Integer sein. Wenn nicht, bricht die Funktion mit einem Fehler ab. Eine Konvertierung von Float (Trunc)/Boolean (0/1) wird vorgenommen. Für ein typunsicheres Setzen der Variable können Sie die SetValue Funktion nutzen.

## BEISPIEL

```
SetValueInteger(47788, $Anzahl_Personen);
```

# SetValueString

```
boolean SetValueString (int $VariablenID, string $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenID | ID der Variable |
| Wert | Text |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion setzt den Inhalt der IP-Symcon Variablen mit der ID **VariablenID**. Die gewählte Variable muss vom Typ String sein. Wenn nicht, bricht die Funktion mit einem Fehler ab. Eine Konvertierung von Float/Integer/Boolean (False/True) wird vorgenommen. Für ein typunsicheres Setzen der Variable kann die Funktion SetValue genutzt werden.

## BEISPIEL

```
SetValueString(47788, $aktuelles_Projekt);
```

# Entwicklerbereich

## ERWEITERTE FUNKTIONEN

| FUNKTION | BESCHREIBUNG |
|---|---|
| Systembefehle | Es können alle Befehle zum Konfigurieren von IP-Symcon genutzt werden, die normalerweise nur für native Modulentwickler verfügbar waren. Über PHP kann vollständig auf das System zugegriffen werden – In der Befehlsreferenz kann ein Überblick über alle Möglichkeiten gewonnen werden. |
| Modulbefehle | Jedes Modul und jede Option der Module, welche über die Verwaltungskonsole verändert werden können, können auch durch Skripte ausgelesen und verändert werden. Dadurch ist die komplette Kontrolle über die angeschlossenen Hardwaremodule möglich. Die Modulreferenz bietet eine komplette Auflistung der Aktionsbefehle. |
| PHP | IP-Symcon bietet das volle Potential der PHP Skriptsprache. Es kann nicht nur auf alle PHP Befehle zugegriffen, sondern auch über die eigene php.ini Anpassungen vorgenommen und Extensions geladen werden. Über diese sind z.B. erweiterte Grafikfunktionen (GD2) oder MySQL, MSSQL oder andere Datenbanken ansprechbar. Der in IP-Symcon integrierte WebServer macht eine seperate LAMP Installation überflüssig. |
| Client/Server Modell | Der IP-Symcon Dienst fügt sich nahtlos in das Serverbetriebssystem ein. Es muss keine interaktive Session gestartet werden – Das bedeutet kein Ärger und keine Problematik bei unerwarteten oder auch geplanten Neustarts des Systems. Nach dem Start ist das System wieder verfügbar. Mit der Verwaltungskonsole kann immer komfortabel auf das System zugegriffen werden. Das System kann bequem von der Workstation aus erweitert werden. Ein direktes "am Server sitzen" oder auch ein einloggen per TeamViewer/AnyDesk/RDP auf den Server ist nicht mehr nötig. Veränderungen in der Verwaltungskonsole werden serverseitig sofort übernommen und können dadurch direkt ausprobiert werden. |
| JSON-RPC Schnittstelle | Die JSON-RPC Schnittstelle, welche Standardmäßig seit IP-Symcon 3.0 verfügbar ist, erlaubt den Zugriff auf alle Funktionen der Befehlsreferenz und Modulreferenz, sodass sich eine Integration in jede Applikation so einfach gestaltet, wie das Schreiben eines PHP Skriptes innerhalb von IP-Symcon. Dabei kann auf die bereits vorhandene Dokumentation der Befehle zurückgegriffen werden. Einige Beispiele zur Integration können unter Datenaustausch gefunden werden. |
| WebServer | Der integrierte WebServer kann auf mehreren Ports betrieben werden. Eine Möglichkeit der SSL Verschlüsslung mit eigenen Zertifikaten und ein Webalizer kompatibles Logfile kann pro Instanz konfiguriert werden. |
| WebFront | Das mit IP-Symcon mitgelieferte Web Interface mit dem Namen “WebFront” bietet eine schnelle Visualisierungsmöglichkeit. Sobald Geräte in IP-Symcon eingerichtet sind, können von überall per LAN/WLAN/Web auf die Geräte zugegriffen werden. Einfach das WebFront aufrufen und schalten. Dokumentationen zu den Erweiterungsmöglichkeiten des WebFronts können im Entwicklerbereich gefunden werden. Für den optimalen Betrieb wird ein aktueller Browser vorausgesetzt. |

## SDK/TOOLS

Mit dem IP-Symcon SDK ist es Möglich Module für IP-Symcon zu erstellen.

# Datenaustausch

## ERREICHBARKEIT

Die seit IP-Symcon 3.0 verfügbare JSON-RPC API ist standardmäßig über den Port 3777 erreichbar. Dabei ist der Port 3777 nur am lokalen System gebunden und erfordert keinerlei Authentifizierung.
Weiterhin ist die JSON-RPC API auf jedem in IP-Symcon eingerichteten WebServer aktiv. Dadurch können auf jeder Host/Port/SSL Konfiguration das WebFront und die Mobilen Apps einen Zugriff zu IP-Symcon bekommen.

## KENNWORT UND FUNKTIONEN

Die JSON-RPC API wird über den Lizenz-Benutzernamen und das Fernzugriff Kennwort authentifiziert. Wenn kein Kennwort angegeben wird, so wird keinerlei Zugriff gewährt. Ein Sonderfall sind einige WFC_* Funktionen, welche über das Kennwort des jeweiligen WebFront-Konfigurators authentifiziert werden. In diesem Fall wird als Benutzername "webfront" verwendet. Diese Funktionen ermöglichen den isolierten Zugriff auf die vom Konfigurator definierten Teilbereiche des Objektbaums. Ausschließlich die WFC_GetConfigurators Funktion wird ohne jegliche Authentifizierung zur Verfügung gestellt. Diese ist erforderlich, um die Auflistung der WebFront-Konfiguratoren für das WebFront und die Mobilen Apps zu ermöglichen.
Zusätzlich wird dadurch die IP-Adressen Autostart Funktion realisiert. Übertragen werden in dieser Funktion nur die ID/Titel/Icon/Position/Editierbarkeit des jeweiligen Konfigurators, ob ein Kennwort erforderlich ist und ob dieser Konfigurator für die Mobile Apps aktiviert wurde. Standardmäßig ist die Nutzung für Mobile Apps im Konfigurator deaktiviert, um ein Sicherheitsproblem auf den Basis-Baum (ID = 0) zu vermeiden.

## AUTHENTIFIZIERUNG

Die im WebServer angebotene Basis-Authentifizierung gilt, sofern der WebServer für das WebFront genutzt wird, ausschließlich für den "user"-Ordner, der für benutzerdefinierte Inhalte freigegeben ist. Wenn der WebServer für ein anderes Verzeichnis genutzt wird, so gilt die Authentifizierung für alle Inhalte, die innerhalb dieses Verzeichnisses liegen - jedoch nicht für die JSON-RPC API, da diese den vorher genannten Authentifizierungsregeln unterliegt.



Für die Funktionsaufrufe können die normalen Befehle der Befehlsreferenz und Modulreferenz verwendet werden.



Als Benutzername muss Ihr Lizenz-Benutzername verwendet werden. Das Kennwort kann über Fernzugriff aktivieren gesetzt werden.

## JSON-RPC SCHNITTSTELLE (ÜBER PHP/IP-SYMCON)

### KERNELVERSION AUSLESEN

```
$rpc = new JSONRPC("http://user:password@127.0.0.1:3777/api/");
$result = $rpc->IPS_GetKernelVersion();
echo "KernelVersion: ".$result;
```

### SKRIPTE STARTEN

```
$rpc = new JSONRPC("http://user:password@127.0.0.1:3777/api/");
$rpc->IPS_RunScript(34956);
```

### VARIABLEN AUSLESEN

```
$rpc = new JSONRPC("http://user:password@127.0.0.1:3777/api/");
echo $rpc->GetValueFormatted(58383);
```

### VARIABLEN VERÄNDERN

```
$rpc = new JSONRPC("http://user:password@127.0.0.1:3777/api/");
$rpc->SetValue(38809, 18.5);
```

### HOMEMATIC GERÄT SCHALTEN

```
$rpc = new JSONRPC("http://user:password@127.0.0.1:3777/api/");
$rpc->HM_WriteValueBoolean(20558, "STATE", false);
```

### RÜCKGABEWERT BEI NICHT ERREICHBARKEIT

```
try {
$rpc = new JSONRPC("http://user:password@127.0.0.1:3777/api/");
$rpc->IPS_GetKernelDir();
} catch (JSONRPCException $e) {
echo 'RPC Problem: ', $e->getMessage(), "\n";
} catch (Exception $e) {
echo 'Server Problem: ', $e->getMessage(), "\n";
}
```

## WEBSOCKET SCHNITTSTELLE (ÜBER JAVASCRIPT)

Änderungen im System können über die Websocket Schnittstelle schnellstmöglich empfangen werden.

### GESAMTE API

```
//IP-Symcon 5.1/IP-Symcon 5.2
//Normal
var connection = new WebSocket('ws://127.0.0.1:3777/api/',
[btoa('email:fernzugriffpw').replace(/=/g, '')]);
//SSL
var connection = new WebSocket('wss://127.0.0.1:3777/api/',
[btoa('email:fernzugriffpw').replace(/=/g, '')]);

//ab IP-Symcon 5.3
//Normal
var connection = new WebSocket('ws://127.0.0.1:3777/api/',
[encodeURIComponent(btoa('email:fernzugriffpw'))]);
//SSL
var connection = new WebSocket('wss://127.0.0.1:3777/api/',
[encodeURIComponent(btoa('email:fernzugriffpw'))]);
```

### WEBFRONT API

```
//IP-Symcon 5.1
//Normal
var connection = new WebSocket('ws://127.0.0.1:3777/api/wfc/12345',
[btoa('webfront:webfrontpw').replace(/=/g, '')]);
//SSL
var connection = new WebSocket('wss://127.0.0.1:3777/api/wfc/12345',
[btoa('webfront:webfrontpw').replace(/=/g, '')]);

//IP-Symcon 5.2
//Normal
var connection = new WebSocket('ws://127.0.0.1:3777/wfc/12345/api/',
[btoa('webfront:webfrontpw').replace(/=/g, '')]);
//SSL
var connection = new WebSocket('wss://127.0.0.1:3777/wfc/12345/
```

```
api/', [btoa('webfront:webfrontpw').replace(/=/g, '')]);

//ab IP-Symcon 5.3
//Normal
var connection = new WebSocket('ws://127.0.0.1:3777/wfc/12345/api/',
[encodeURIComponent(btoa('webfront:webfrontpw'))]);
//SSL
var connection = new WebSocket('wss://127.0.0.1:3777/wfc/12345/
api/', [encodeURIComponent(btoa('webfront:webfrontpw'))]);
```

## JSON-RPC SCHNITTSTELLE (ÜBER CURL)

Aufruf via Kommandozeile

### VARIABLEN VERÄNDERN

Beim Benutzername mail%40provider.de oder Passwort darf @ nicht geschrieben werden. => Als Ersatz: %40
Benutzername (Mail), Passwort und IP ohne <,> eintragen!



Zur Kodierung und Berücksichtigung von Sonderzeichen kann die Seite urlencoder.org für Benutzername und Passwort genutzt werden.

```
//SetValue Integer mit den Parametern ID = 12345 und Wert = 42
curl -i -X POST -H "Content-Type: application/json" -d
"{\"jsonrpc\": \"2.0\", \"id\": \"0\", \"method\": \"SetValue\",
\"params\": [12345, 42]}" http://
<mail%40provider.de>:<Passwort>@<IP>:3777/api/
//SetValue Float mit den Parametern ID = 12346 und Wert = 1.23
curl -i -X POST -H "Content-Type: application/json" -d
"{\"jsonrpc\": \"2.0\", \"id\": \"0\", \"method\": \"SetValue\",
\"params\": [12346, 1.23]}" http://
<mail%40provider.de>:<Passwort>@<IP>:3777/api/
//SetValue Boolean mit den Parametern ID = 12347 und Wert = false
curl -i -X POST -H "Content-Type: application/json" -d
"{\"jsonrpc\": \"2.0\", \"id\": \"0\", \"method\": \"SetValue\",
\"params\": [12347, false]}" http://
<mail%40provider.de>:<Passwort>@<IP>:3777/api/
//SetValue String mit den Parametern ID = 12348 und Wert = "mein
String"
curl -i -X POST -H "Content-Type: application/json" -d
"{\"jsonrpc\": \"2.0\", \"id\": \"0\", \"method\": \"SetValue\",
\"params\": [12348, \"mein String\"]}" http://
<mail%40provider.de>:<Passwort>@<IP>:3777/api/
//Rückgabewert bei Erfolg
{"jsonrpc":"2.0","id":"0","result":true}
```

## VARIABLEN AUSLESEN

```
//GetValue mit den Parametern ID = 12345
curl -i -X POST -H "Content-Type: application/json" -d
"{\"jsonrpc\": \"2.0\", \"id\": \"0\", \"method\": \"GetValue\",
\"params\": [12345]}" http://
<mail%40provider.de>:<Passwort>@<IP>:3777/api/
//Rückgabewert
{"jsonrpc":"2.0","id":"0","result":42}
```

# Kompatibilitätsfunktionen

Kompatibilitätsfunktionen sind die Funktionen, welche mittlerweile nicht mehr verwendet werden sollen und standardmässig deaktiviert sind. Diese wurden aufgrund von Performance abgeschafft oder durch verbesserte Ersatzfunktionen abgelöst. Aus Kompatibilitätsgründen zu alten IP-Symcon Installationen sind diese aber bei Bedarf über den Spezialschalter "Kompatibilitätsfunktionen" aktivierbar. Dies benötigt aber viele Ressourcen und es wird stark davon abgeraten.

# IPS_GetInstanceChildrenIDs

```
array IPS_GetInstanceChildrenIDs (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der zu überprüfenden Instanz |

## RÜCKGABEWERT

Ein Array von InstanzIDs der Unterinstanzen

## BESCHREIBUNG

Diese Funktion sollte ab IP-Symcon 2.6 nicht mehr verwendet werden und wurde mit IP-Symcon 4.0 entfernt.

Die Funktion gibt alle Unterinstanzen von einer bestimmte Instanz **InstanzID** aus, gesehen von der Hardwaresicht (physikalisch).

Diese Funktion ist nur sinnvoll bei I/O und Splitter Instanzen.

## BEISPIEL

```
print_r(IPS_GetInstanceChildrenIDs(12345));

// Beispiel um die veraltete Funktion "IPS_GetInstanceChildrenIDs"
zu ersetzen.
// Dazu wird IPS_GetInstance genutzt

$InstanceIDs = IPS_GetInstanceList();
foreach($InstanceIDs as $IID)
if(IPS_GetInstance($IID)['ConnectionID'] == 12345)
echo $IID . PHP_EOL;
```

# IPS_GetInstanceParentID

```
int IPS_GetInstanceParentID (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der zu überprüfenden Instanz |

## RÜCKGABEWERT

ID der übergeordneten Instanz

## BESCHREIBUNG



Diese Funktion sollte ab IP-Symcon 2.6 nicht mehr verwendet werden und wurde mit IP-Symcon 4.0 entfernt.

Die Funktion gibt die ID der übergeordneten Instanz für Instanz **InstanzID** aus (von der Hardwaresicht (physikalisch) gesehen).



Diese Funktion ist nur sinnvoll bei Device und Splitter Instanzen.

## BEISPIEL

```
// Bis Version 2.5
echo IPS_GetInstanceParentID(12345);

// Ab Version 2.6
echo IPS_GetInstance(12345)['ConnectionID'];
```

# IPS_GetScriptID

```
int IPS_GetScriptID (string $SkriptName)
```

## PARAMETERLISTE

| | |
|---|---|
| SkriptName | Zu suchender SkriptName |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG



Diese Funktion kann ab IP-Symcon 2.5 nicht mehr verwendet werden. Alte IP-Symcon 1.0 Funktionen wurden mit diesem Release entfernt.

Die Funktion versucht, die ID des Skripts mit dem Namen **SkriptName** zu ermitteln. Es wird die ID des ersten gefundenen Skripts verwendet, dessen Name mit **SkriptName** übereinstimmt. Falls ein solches Skript gefunden wurde, wird dessen ID zurückgemeldet, andernfalls 0 (Null).
Eine sicherere Methode, die ID zu bestimmen, bietet die Funktion IPS_GetScriptIDByName.



**Vorsicht:** Da Namen in IP-Symcon beliebig vergeben werden können, und damit **nicht eindeutig** sind, kann die Funktion eine falsche ID zurückmelden.

## BEISPIEL

```
//Fehlermeldung mit @ unterdrücken
$ScriptID = @IPS_GetScriptID("Regenerfassung");
if ($ScriptID === false)
echo "Skript nicht gefunden!";
else
echo "Die Skript-ID lautet: ". $ScriptID;
```

# IPS_GetStatusVariable

```
array IPS_GetStatusVariable (int $InstanzID, string $VariableIdent)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| VariableIdent | Status Variable Identifikator. Eine Auflistung kann über IPS_GetStatusVariableIdents abgerufen werden. |

## RÜCKGABEWERT

Folgende Informationen stehen als **key => value** Paare zur Verfügung:

| INDEX | TYP | BESCHREIBUNG |
|---|---|---|
| **VariableID** | integer | Aktuelle ID der verknüpften IP-Symcon Variable |
| **VariableIdent** | string | Status Variablen Identifikator |
| **VariableName** | string | Standard Name der angelegten Variable |
| **VariablePosition** | integer | Standard Position der angelegten Variable |
| **VariableProfile** | string | Standard Profilname der angelegten Variable |
| **VariableType** | integer | Kompatibler Variablentyp (Siehe IPS_GetVariable) |
| **VariableHasAction** | boolean | Gibt an, ob intern eine Aktion definiert ist, die mit dieser Status Variable verknüpft wurde. |
| **VariableUseAction** | boolean | Gibt an, wie die vorhandene Aktion in der Visualisierung benutzt werden soll. |

## BESCHREIBUNG

Die Dokumentation dieser Funktion ist nur noch aus Kompatibilitätsgründen vorhanden und sollte ab IP-Symcon 2.5 nicht mehr verwendet werden. Diese Funktion wird ab Version 2.5 nur noch emuliert und sollte durch die entsprechenden Ersatzfunktionen ausgetauscht werden. Eine Ersatzfunktion befindet sich im Beispiel (siehe unten).

Die Funktion liefert ein Array mit Informationen über die Status Variable von der Instanz **InstanzID** und der Status Variable mit dem Status Variablen Identifikator **VariableIdent**.

## BEISPIEL

```
// bis Version 2.5
if(IPS_StatusVariableExists(12345, "StatusVariable"))
{
print_r(IPS_GetStatusVariable(12345, "StatusVariable"));
}

// ab Version 2.6 Ersatzfunktion
$VariableIdent = "Statusvariable";
$InstanceID = 12345;

$id = IPS_GetObjectIDByIdent($VariableIdent, $InstanceID);
$v = IPS_GetVariable($id);
$result = Array(
"VariableID" => $id,
"VariableIdent" => $VariableIdent,
"VariableName" => "N/A",
"VariablePosition" => 0,
"VariableProfile" => $v['VariableProfile'],
"VariableType" => $v['VariableType'],
"VariableHasAction" => ($v['VariableAction'] > 0),
"VariableUseAction" => ($v['VariableAction'] > 0)
);
print_r($result);
```

# IPS_GetStatusVariableID

```
int IPS_GetStatusVariableID (int $InstanzID, string $VariableIdent)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| VariableIdent | Status Variable Identifikator. Eine Auflistung kann über IPS_GetStatusVariableIdents abgerufen werden. |

## RÜCKGABEWERT

VariablenID der Variable, die der Status Variable zugeordnet ist

## BESCHREIBUNG



Die Dokumentation dieser Funktion ist nur noch aus Kompatibilitätsgründen vorhanden und sollte ab IP-Symcon 2.5 nicht mehr verwendet werden. Diese Funktion wird ab Version 2.5 nur noch emuliert und sollte durch die entsprechenden Ersatzfunktionen ausgetauscht werden. Eine Ersatzfunktion befindet sich im Beispiel (siehe unten).

Die Funktion gibt die VariablenID der Variable zurück, die mit der Status Variable der Instanz **InstanzID** und dem Status Variablen Identifikator **VariableIdent** verknüpft ist.

## BEISPIEL

```
//Zustand von z.B. einem FS20 Gerät darstellen ohne die ID der
"Status" Variable zu kennen
echo GetValue(IPS_GetStatusVariableID(12345, "StatusVariable"));

// ab Version 2.6 Ersatzfunktion
$InstanceID = "StatusVariable";
$VariableIdent = 12345;

$migrateIdents = Array(
"F05_StatusVar" => "Status",
"F10_TemperatureVar" => "Temperature",
"F12_Var0" => "Status0",
"F12_Var1" => "Status1",
"F1D_CounterVar1" => "Counter1",
```

```
"F1D_CounterVar2" => "Counter2",
"F20_Var0" => "Port0",
"F20_Var1" => "Port1",
"F20_Var2" => "Port2",
"F20_Var3" => "Port3",
"F26_TemperatureVar" => "Temperature",
"F26_VDDVar" => "VDD",
"F26_VADVar" => "VAD",
"F26_XSENSVar" => "XSENS",
"F28_TemperatureVar" => "Temperature",
"F29_Var0" => "Status0",
"F29_LatchVar0" => "Latch0",
"F29_Var1" => "Status1",
"F29_LatchVar1" => "Latch1",
"F29_Var2" => "Status2",
"F29_LatchVar2" => "Latch2",
"F29_Var3" => "Status3",
"F29_LatchVar3" => "Latch3",
"F29_Var4" => "Status4",
"F29_LatchVar4" => "Latch4",
"F29_Var5" => "Status5",
"F29_LatchVar5" => "Latch5",
"F29_Var6" => "Status6",
"F29_LatchVar6" => "Latch6",
"F29_Var7" => "Status7",
"F29_LatchVar7" => "Latch7",
"F2C_PositionVar" => "Position",
"F3A_Var0" => "Status0",
"F3A_Var1" => "Status1"
);

if(isset($migrateIdents[$VariableIdent]))
{
$VariableIdent = $migrateIdents[$VariableIdent];
}

$result = IPS_GetObjectIDByIdent($VariableIdent, $InstanceID);
echo $result;
```

# IPS_GetStatusVariableIdents

```
array IPS_GetStatusVariableIdents (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |

## RÜCKGABEWERT

Ein Array von Strings, die die Status Variable Indentifikatoren der Instanz angeben

## BESCHREIBUNG

Die Dokumentation dieser Funktion ist nur noch aus Kompatibilitätsgründen vorhanden und sollte ab IP-Symcon 2.5 nicht mehr verwendet werden. Diese Funktion wird ab Version 2.5 nur noch emuliert und sollte durch die entsprechenden Ersatzfunktionen ausgetauscht werden. Eine Ersatzfunktion befindet sich im Beispiel (siehe untenSpezialschalter).

Die Funktion ermittelt alle der Instanz **InstanzID** bekannten Status Variablen und liefert die Status Variablen Identifikatoren in einem Array von Werten zurück.

## BEISPIEL

```
print_r(IPS_GetStatusVariableIdents(12345));

// ab Version 2.6 Ersatzfunktion
$InstanceID = 12345

$r = Array();
$cids = IPS_GetChildrenIDs($InstanceID);
foreach($cids as $cid) {
$o = IPS_GetObject($cid);
if($o['ObjectIdent'] != "") {
$r[] = $o['ObjectIdent'];
}
}
$result = $r;
print_r($result);
```

# IPS_GetUptime

```
int IPS_GetUptime ()
```

## RÜCKGABEWERT

Zeitpunkt, an dem IP-Symcon gestartet wurde

## BESCHREIBUNG

Die Funktion liefert Datum und Uhrzeit des Startzeitpunkts von IP-Symcon als Unix Timestamp.



Diese Funktion ist nur bis Version 3.4 implementiert. Ab Version 4.0 wird diese durch IPS_GetKernelStartTime ersetzt.

## BEISPIEL

```
// liefert z.B. 1204675012
echo IPS_GetUptime();
// liefert z.B. "2008.03.04 23:56:52"
echo date("Y.m.d H:i:s", IPS_GetUptime());
```

# IPS_GetVariableID

```
int IPS_GetVariableID (string $VariablenName)
```

## PARAMETERLISTE

| | |
|---|---|
| VariablenName | zu suchender VariablenName |

## RÜCKGABEWERT

ID der gefundenen Variable, sonst **FALSE** und ein **Warning**

## BESCHREIBUNG



Diese Funktion kann ab IP-Symcon 2.5 nicht mehr verwendet werden. Alte IP-Symcon 1.0 Funktionen wurden mit diesem Release entfernt.

Die Funktion versucht, die ID der Variablen mit dem Namen **VariablenName** zu ermitteln. Es wird die ID der ersten gefundenen Variablen verwendet, deren Name mit **VariablenName** übereinstimmt. Falls eine solche Variable gefunden wurde, wird deren ID zurückgemeldet, andernfalls FALSE.
Eine sicherere Methode, die ID zu bestimmen, bietet die Funktion IPS_GetVariableIDByName.



**Vorsicht:** Da Namen in IP-Symcon beliebig vergeben werden können, und damit **nicht eindeutig** sind, kann die Funktion eine falsche ID zurückmelden.

## BEISPIEL

```
//Fehlermeldung mit @ unterdrücken
$VarID = @IPS_GetVariableID("Regenmenge");
if ($VarID === false)
echo "Variable nicht gefunden!";
else
echo "Die Variablen-ID lautet: ". $VarID;
```

# IPS_HasInstanceChildren

```
boolean IPS_HasInstanceChildren (boolean $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der zu überprüfenden Instanz |

## RÜCKGABEWERT

Der Rückgabewert ist **TRUE**, wenn der Instanz eine oder mehrere Instanzen untergeordnet sind. Sonst, **FALSE**.

## BESCHREIBUNG



Diese Funktion sollte ab IP-Symcon 2.6 nicht mehr verwendet werden und wurde mit IP-Symcon 4.0 entfernt.

Die Funktion gibt an, ob eine bestimmte Instanz **InstanzID**, von der Hardware (physikalischen) Sicht gesehen aus, Unterinstanz - z.B. einen Splitter oder Device Modul - hat.



Diese Funktion ist nur Sinnvoll bei I/O und Splitter Instanzen.

## BEISPIEL

```
if(IPS_HasInstanceParent(12345))
{
echo "Hat eine hat untergeordnete Instanzen";
}
```

# IPS_HasInstanceParent

```
boolean IPS_HasInstanceParent (int $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der zu überprüfenden Instanz |

## RÜCKGABEWERT

Der Rückgabewert ist **TRUE**, wenn die Instanz eine übergeordnete Instanz besitzt. Sonst, **FALSE**.

## BESCHREIBUNG



Diese Funktion sollte ab IP-Symcon 2.6 nicht mehr verwendet werden und wurde mit IP-Symcon 4.0 entfernt.

Die Funktion gibt an, ob eine bestimmte Instanz **InstanzID**, von der Hardware (physikalischen) Sicht gesehen aus, überliegenden Instanz z.B. einen Splitter oder I/O Modul hat.



Diese Funktion ist nur sinnvoll bei Device und Splitter Instanzen.

## BEISPIEL

```
// Bis Version 2.5
if(IPS_HasInstanceParent(12345))
{
echo "Hat eine übergeordnete Instanz";
}

// Ab Version 2.6
if(IPS_GetInstance(12345)['ConnectionID'] > 0)
{
echo "Hat eine übergeordnete Instanz";
}
```

# IPS_IsPersistent

```
boolean IPS_IsPersistent (int $ObjektID)
```

## PARAMETERLISTE

| | |
|---|---|
| ObjektID | zu prüfende ObjektID |

## RÜCKGABEWERT

TRUE, wenn das Objekt gespeichert wird, sonst FALSE

## BESCHREIBUNG



Diese Funktion sollte ab IP-Symcon 2.6 nicht mehr verwendet werden. Da alle Objekte in IP-Symcon persistent sind, ist diese Funktion überflüssig.

Die Funktion prüft, ob ein Objekt vom System in die Settings abgespeichert wird. Das gilt zur Zeit für alle IDs größer als 10000. Andernfalls wird das Objekt beim Beenden verworfen.

## BEISPIEL

```
if (IPS_IsPersistent(34881))
echo "Das Objekt wird beim Beenden abgespeichert!";
```

# IPS_SetLinkChildID

Benötigt: IP-Symcon >= 2.1

```
boolean IPS_SetLinkChildID (int $LinkID, int $VerknüpftesObjekt)
```

## PARAMETERLISTE

| | |
|---|---|
| LinkID | ID des zu verändernden Links |
| VerknüpftesObjekt | ID des zu verknüpfenden Objekts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG



Diese Funktion sollte ab IP-Symcon 2.6 nicht mehr verwendet werden. Stattdessen ist es empfohlen die Funktion IPS_SetLinkTargetID zu nutzen. Seit IP-Symcon 4.0 kann diese Funktion nur noch benutzt werden, wenn der entprechende Spezialschalter gesetzt wurde.

Die Funktion verknüpft ein bestimmtes Objekt **VerknüpftesObjekt** mit dem Link **LinkID**, sodass der Link auf dieses Objekt zeigt.

## BEISPIEL

```
IPS_SetLinkChildID($LinkID, 12345); //Auf Objekt 12345 verweisen
```

# IPS_SetStatusVariableUseAction

```
boolean IPS_SetStatusVariableUseAction (int $InstanzID, string
$VariableIdent, boolean $BenutzteAktion)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| VariableIdent | Status Variable Identifikator. Eine Auflistung kann über IPS_GetStatusVariableIdents abgerufen werden. |
| BenutzteAktion | Gibt an, ob die Aktion in der Visualisierung genutzt werden soll. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Dokumentation dieser Funktion ist nur noch aus Kompatibilitätsgründen vorhanden und sollte ab IP-Symcon 2.5 nicht mehr verwendet werden. Diese Funktion wird ab Version 2.5 nur noch emuliert und sollte durch die entsprechenden Ersatzfunktionen ausgetauscht werden. Eine Ersatzfunktion befindet sich im Beispiel (siehe unten).

Die Funktion setzt, ob die intern verknüpfte Aktion einer bestimmten Instanz **InstanzID** und der Status Variable mit dem Status Variablen Identifikator **VariableIdent** benutzt werden soll.

## BEISPIEL

```
if(IPS_StatusVariableExists(12345, "StatusVariable"))
{
IPS_SetStatusVariableUseAction(12345, "StatusVariable", false); //
Per WebFront nicht mehr schaltbar machen
}

// ab Version 2.6 Ersatzfunktion
$VariableIdent = "Statusvariable";
```

```
$InstanceID = 12345;
$UseAction = false;

$id = IPS_GetObjectIDByIdent($VariableIdent, $InstanceID);

if ($UseAction) {
IPS_SetVariableCustomAction($id, 0); // Standardaktion aktivieren
} else {
IPS_SetVariableCustomAction($id, 1); // Standardaktion deaktivieren
}
```

# IPS_StatusVariableExists

```
boolean IPS_StatusVariableExists (int $InstanzID, string
$VariableIdent)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID der Instanz |
| VariableIdent | Status Variable Identifikator. Eine Auflistung kann über IPS_GetStatusVariableIdents geholt werden. |

## RÜCKGABEWERT

Der Rückgabewert ist **TRUE**, wenn die Status Variable vorhanden ist, sonst **FALSE**

## BESCHREIBUNG



Die Dokumentation dieser Funktion ist nur noch aus Kompatibilitätsgründen vorhanden und sollte ab IP-Symcon 2.5 nicht mehr verwendet werden. Diese Funktion wird ab Version 2.5 nur noch emuliert und sollte durch die entsprechenden Ersatzfunktionen ausgetauscht werden. Eine Ersatzfunktion befindet sich im Beispiel (siehe unten).

Die Funktion überprüft, ob eine bestimmte Instanz **InstanzID** die Status Variable mit dem Status Variablen Identifikator **VariableIdent** besitzt.

## BEISPIEL

```
if(IPS_StatusVariableExists(12345, "StatusVariable"))
{
echo IPS_GetStatusVariableID(12345, "StatusVariable");
}

// ab Version 2.6 Ersatzfunktion
$InstanceID = "StatusVariable";
$VariableIdent = 12345;

$result = !(@IPS_GetObjectIDByIdent($VariableIdent, $InstanceID) ===
false);
echo $result;
```

# Limitationen

Die nachfolgend beschriebenen Beschränkungen sind aus Gründen der Performance und Sicherheit des System erforderlich. Eine Veränderbarkeit dieser ist nicht geplant.

| LIMITATION | BESCHREIBUNG |
|---|---|
| Maximale Anzahl der Objekte | 50000. Der Adressraum von 10000-59999 wird dabei genutzt. Je nach Lizenz/IP-Symcon Version kann dieser Wert geringer sein. |
| Maximale Größe einer String-Variable | Ab 1024kb wird ein Fehler erzeugt und der Inhalt nicht geschriebenen. Aus Performance- und Stabilitätsgründen sollten Daten über 8kb anderweitig ausgelagert werden. Die 1024kb Grenze dient nur als Hard-Limit, um katastrophale Auswirkungen zur Laufzeit durch fehlerhafte Skripte einzudämmen. |
| Maximale Größe des RegisterVariable Puffers | Eine Warnung wird ab 8kb Größe in das Log geschrieben. Es werden nur die letzten 64kb an Daten im Puffer behalten. Ältere Daten werden ohne weitere Warnung verworfen. |
| Maximale Größe des Cutter Puffers | Eine Warnung wird ab 8kb Größe in das Log geschrieben. Es werden nur die letzten 64kb an Daten im Puffer behalten. Ältere Daten werden ohne weitere Warnung verworfen. |
| Maximale Anzahl an PHP-Threads | Die Standardeinstellung von 50 PHP-Threads ist für fast alle Systeme völlig ausreichend. Das Minimum liegt bei 25 PHP-Threads. Wenn ein Wert kleiner als 25 gewählt wird, werden trotzdem 25 PHP-Threads erstellt. Sollten mehr PHP-Threads benötigt werden, so kann der Wert auf z.B. 100 erhöht werden. Viel höhere Werte sind in den wenigsten Fällen sinnvoll und konsumieren unnötig viel RAM. |
| Maximaler Speicher pro PHP-Thread | Die Standardeinstellung ist 32MB pro PHP-Thread. Maximal kann dieser Wert auf 64M pro Thread in der php.ini verändert werden. Konsultieren Sie dafür die PHP-Dokumentation. |
| Maximaler Speicher pro Buffers | Der Buffer besitzt ein Softlimt von 256kB und ein Hardlimit von 1024kB. Das heißt ab 256kB größe wirft IP-Symcon eine "Warning" in die Meldungen und die maximale Größe des Buffers beträgt 1024kB. Daten größer als 1024kB werden abgeschnitten. |
| Abfrage von Daten über AC_* Funktionen | Es werden maximal 10000 Zeilen zurückgegeben, unabhängig davon, ob das Limit höher gewählt wurde. Es wird keinerlei Warnung ausgegeben. Abfragen sollten entsprechend optimiert werden, so dass auf die aggregierten Daten zugegriffen werden kann. Das minimiert die Belastung der Datenbank und somit minimiert es auch Zugriffszeiten für die WebFront-Visualisierung der einzelnen Zugriffe. |
| Maximale Assoziationen pro Variablenprofil | Es sind maximal 128 (ab 5.0; vorher 32) Assoziationen pro Variablenprofil erlaubt. Der Hintergrund ist, dass jede Änderung an den Profilassoziationen an alle WebFronts übertragen werden muss, was bei sehr vielen Assoziationen einen hohen Overhead hat. Außerdem ist dieses Feature nicht für z.B. eine Playlisten-Auswahl konzipiert, da diese das Design und Layout des WebFronts negativ beeinflussen würde. |

# Mirroring

Benötigt: IP-Symcon >= 7.0



Mirroring ist eine kostenpflichtige Erweiterung, welche zu jeder bestehenden IP-Symcon Lizenz erworben werden kann. Für eine passende Demo-Version wenden Sie sich bitte an unseren Support. Die Erweiterung kann direkt im Shop erworben werden.

Ein System kann als Mirror eines anderen Server-Systems konfiguriert werden. In diesem Fall wird der Mirror regelmäßig vom Server aktualisiert und passt seine Konfiguration an die des Servers an. Sollte der Server ausfallen, so kann der Mirror verwendet werden und die Aufgaben des Servers übernehmen, wodurch die Verfügbarkeit des Gesamtsystems optimiert wird.



Mirroring beinhaltet diverse Parameter, welche passend zum Anwendungsfall konfiguriert werden sollten. Damit die Inbetriebnahme dieser Erweiterung optimal erfolgt, empfehlen wir Kontakt zu unserem Support aufzunehmen, damit gemeinsam die optimale Konfiguration für den Anwendungsfall erarbeitet und umgesetzt werden kann.

Zur Verwendung von Mirroring muss beim Server der Fernzugriff aktiv sein. In der Kern Instanz "Replikation" können die initialen Einstellungen für das Mirroring vorgenommen werden. Diese werden anschließen in der Datei "replication.json" im gleichen Verzeichnis wie die "settings.json" - siehe Installation des entsprechenden Betriebssystems - angelegt. Änderungen nach dem initialen Aktivieren der Replikation können nur noch in dieser Datei vorgenommen werden, da ansonsten das restliche System ein 1:1 Mirror vom Haupt-System ist.

## PARAMETER DER REPLICATION.JSON

| PARAMETER | TYP | BESCHREIBUNG |
|---|---|---|
| Active | Boolean | Mirroring ist aktiv, falls der Wert auf true gesetzt ist, ansonsten ist es inaktiv und das System agiert normal. |
| Mode | Integer | Betriebsmodus des gespiegelten Systems<br>0 - Parallel: Das System ist voll einsatzbereit und all seine Verbindungen sind aktiv. Somit kann das System direkt verwendet werden, falls der Server ausfällt. Dafür muss aber entweder ein Gateway die Verbindung von mehreren Geräten unterstützen oder ein eigenes Gateway vorhanden sein. Sollten das gespiegelte System verwendet werden, obwohl der Server noch läuft, so werden diese Änderungen bei der nächsten Spiegelung überschrieben und verfallen somit<br>1 - Standby: Das System ist im Standby und all seine Verbindungen sind inaktiv. Bei Ausfall des Servers muss daher das gespiegelte System einmal heruntergefahren werden, der Parameter Active auf false gesetzt werden und das System wieder hochgefahren werden. Dann werden die benötigten Verbindungen hergestellt und das gespiegelte System verwendet werden. |
| Host | String | Die URL unter welcher das Server-System erreichbar ist |
| Port | Integer | Der Port unter welchem das Server-System IP-Symcon bereitstellt |
| UseSSL | Boolean | Bei true wird die Verbindung zum Client via SSL verschlüsselt |
| VerifyPeer | Boolean | Bei true wird verifiziert, dass das SSL Zertifikat des Hosts korrekt ist (nur falls UseSSL ebenfalls true ist) |
| VerifyHost | Boolean | Bei true wird verifiziert, dass die Host URL aus dem SSL Zertifikat mit dem Host aus der Eigenschaft übereinstimmt (nur sichtbar falls Benutze SSL aktiv ist) |
| Username | String | Der Benutzername, also die Lizenzadresse, des Server-Systems |
| Password | String | Das Passwort für den Fernzugriff des Server-Systems |
| Timeout | Integer | Der Timeout bei Anfragen an den Server in Millisekunden. Benötigt eine Anfrage länger, wird die Verbindung als fehlerhaft markiert und keine Spiegelung durchgeführt |
| Synchronization | Object | Detaileinstellungen um Aspekte des Servers nicht zu spiegeln und dabei eigene Einstellungen beizubehalten |

## PARAMETER FÜR SYNCHRONIZATION

| PARAMETER | TYP | BESCHREIBUNG |
|---|---|---|
| ExcludeObjects | Array | Objekte, deren IDs in dieser Liste stehen, werden bei einer Spiegelung nicht aktualisiert und behalten die Konfiguration des gespiegelten Systems bei |
| ExcludeProfiles | Array | Variablenprofile, deren Namen in dieser Liste stehen, werden bei einer Spiegelung nicht aktualisiert und behalten die Konfiguration des gespiegelten Systems bei |

# SDK/Tools

Das SDK dient dazu eigene Module zu entwickeln und diese über das "Module Control" einbinden zu können. Die so entwickelten Module können anderen Nutzern zur Verfügung gestellt werden.

## SDK

Ab Version 3.0
SDK für Excel
SDK für Skins
Ab Version 4.0
SDK für PHP

## TOOLS

Guid Generator
Module Generator
Module Validator
Network-Configuration-Tool
PhpStorm
Recoverytool
Visual Studio Code

# SDK (Excel)

Benötigt: IP-Symcon >= 3.0

## BESCHREIBUNG

Ab IP-Symcon 3.0 unterstützt IP-Symcon die JSON-RPC Schnittstelle und somit kann die Datenbank einfach in Microsoft Excel eingelesen werden.

## VORAUSSETZUNGEN

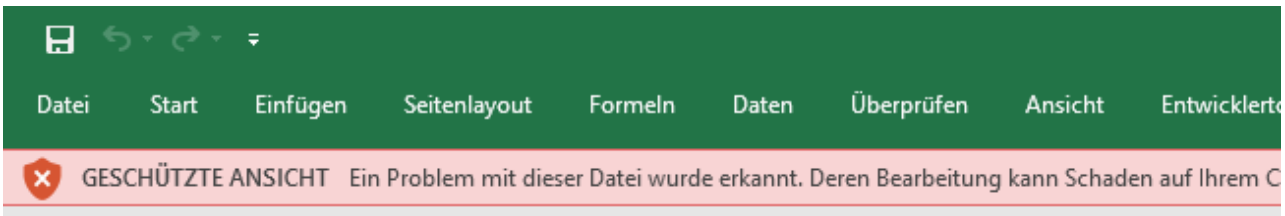
IP-Symcon 3.0 oder höher
Microsoft Excel oder Vergleichbares mit Visual Basic for Applications Unterstützung
"Visual Basic for Applications" Kenntnisse (optional)
Download der vorbereiteten Excel-Datei

## EINRICHTEN IN MICROSOFT EXCEL

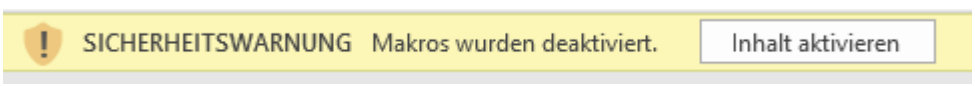
Die heruntergeladene Datei muss entpackt und in Excel geöffnet werden.

### ERSTMALIGES ÖFFNEN DER DATEI

Bei dem ersten Öffnen der Datei erscheinen zwei Warnungen, welche durchgeklickt und erlaubt werden müssen.
Erste Warnung:

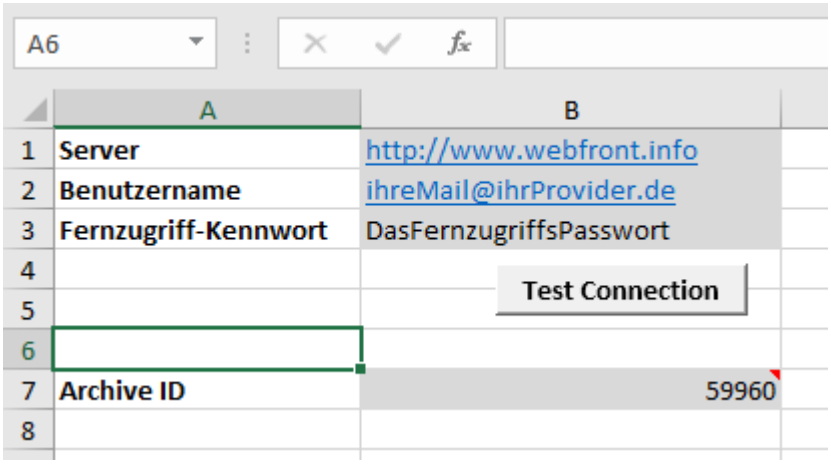



Wenn auf den Schriftzug der Meldung geklickt wird, muss "Trotzdem bearbeiten" angewählt werden. Dies ist eine Standard Sicherheitsmaßnahme, da die Datei Skripte enthält.
Im Anschluss erscheint entweder ein Popup oder eine gelbe Sicherheitswarnung, welche beide vor den enthaltenen Makros warnen. Diese müssen für die Funktionalität aktiviert werden. Somit Entweder im Popup "Makros aktivieren" oder in der gelben Meldung "Inhalt aktivieren" anklicken.




### ZUGRIFF KONFIGURIEREN

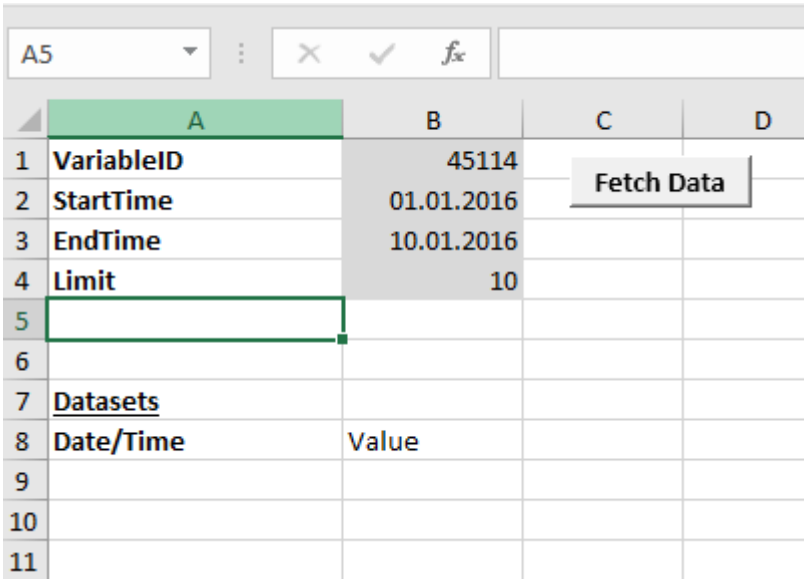
In dem Reiter "Konfiguration" kann die Verbindung zur IPS-Datenbank getestet werden. Es ist drauf zu achten, dass der Fernzugriff aktiviert ist.




In der Konfigurationstabelle können Server, Benutzername, Fernzugriff-Kennwort angegeben werden.
Bei Server können vollständige URLs oder IP-Adressen angegeben werden.
Bei IP-Adressen ist darauf zu achten, dass diese in dem Format "http://192.168.1.1:3777" angegeben wird.
Das Fernzugriff-Kennwort und der Benutzername ist nur nötig, wenn nicht lokal (127.0.0.1) auf die Daten zugegriffen werden soll.
Über "Test Connection" wird die Verbindung aufgebaut und die InstanzID des Archivs wird ausgelesen.

## ROHDATEN AUSLESEN

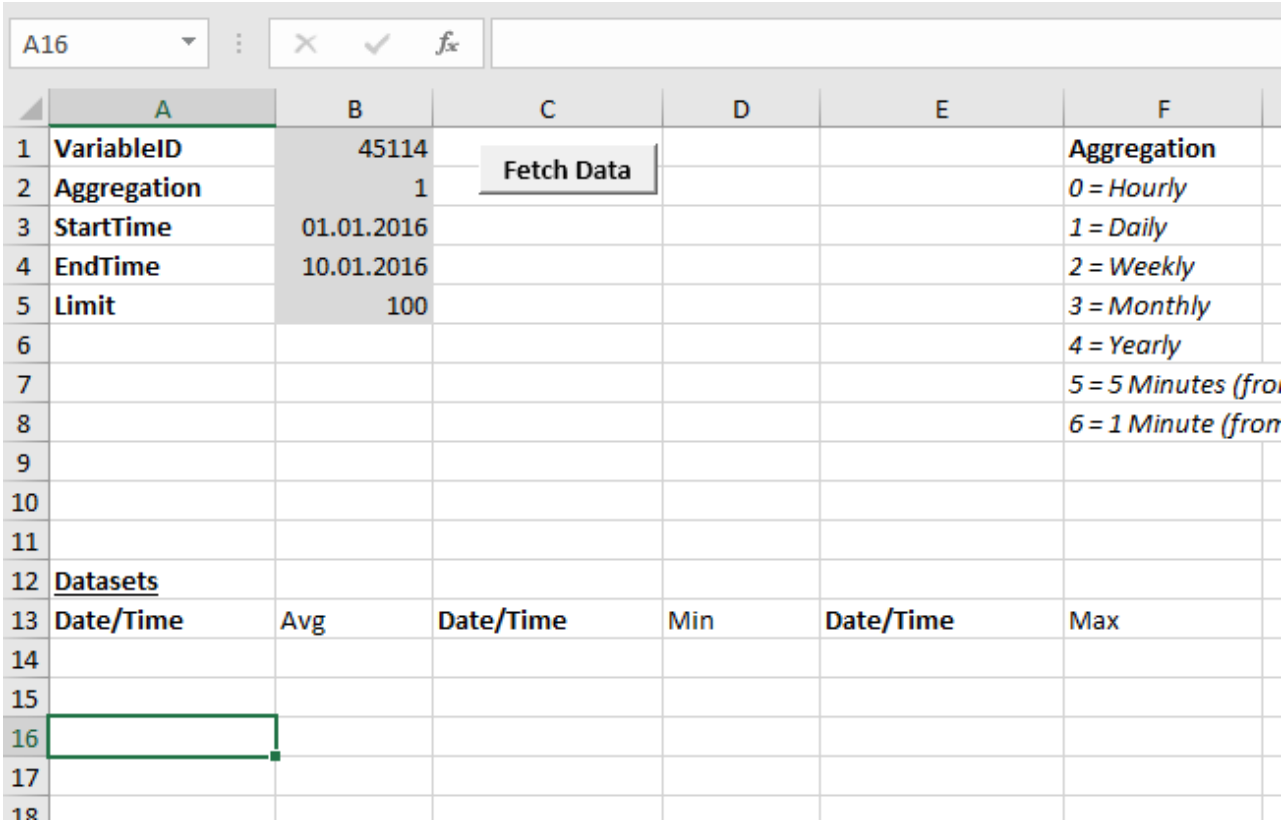
Im Reiter "Rohdaten" können die Rohdaten einer Varaible ausgelesen werden. Diese werden dann Zeilenweise mit Datum und Wert ausgegeben.




Eingetragen werden muss die VariablenID. Start- und Endzeit können 0 sein und setzten damit keine Grenzen. Limit setzt die maximale Anzahl an Datensätze, die ausgelesen werden sollen. Systembedingt gibt es allerdings ein Limit von 10000 Datensätzen.
Über "Fetch Data" werden die Daten eingelesen und in die Tabelle eingetragen. Der aktuellste Datensatz erscheint zuerst.

## AGGREGATION AUSLESEN

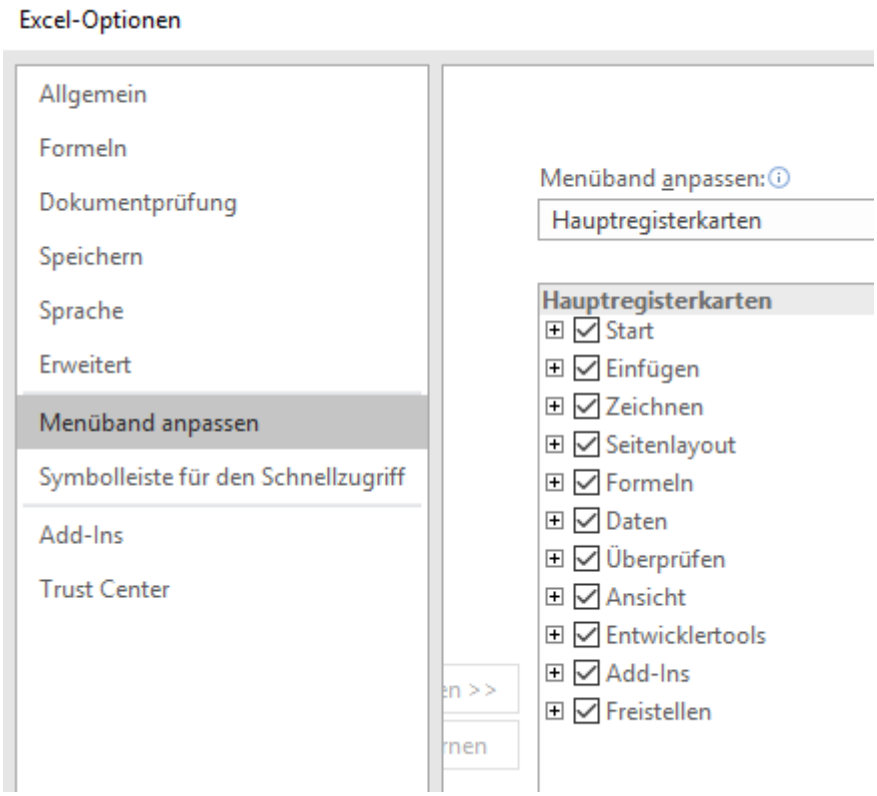
Der dritte Reiter beinhaltet die "Aggregation". Auch werden wieder die VariablenID, Start-/Endzeit und das Limit benötigt.
Zusätzlich kann die Aggregation gewählt werden. Mit den Werten 0-6 kann zwischen minutengenauer bis zu jährlicher Aggregation gewählt werden.



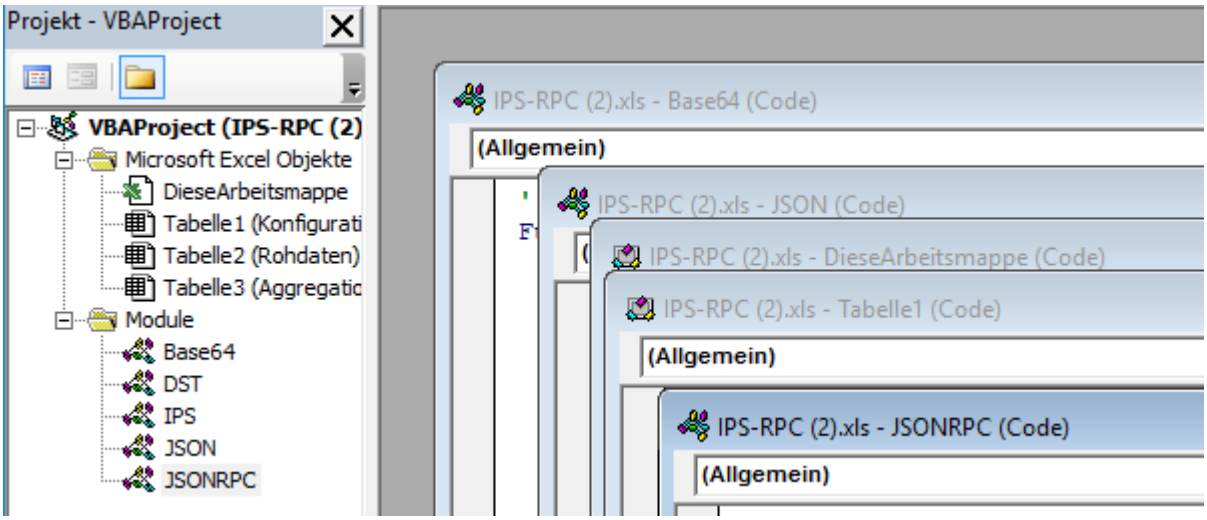

Über "Fetch Data" werden die Daten eingelesen und in die Tabelle eingetragen. Es wird ein Average, sowie Minima und Maxima innerhalb eine Aggregation ausgegeben.

## VISUAL BASIC FOR APPLICATIONS UND DIE KONFIGURATION

Um die Skripte und Makros, welche das Auslesen ermöglichen, zu verwalten oder ändern, sollten die Entwicklertools in der Schnellleiste aktiviert werden.
Dies kann einfach via "Rechtsklick->Datei->Menüband anpassen" ausgewählt werden. Dort muss nur das Menü "Entwicklertools" aktiviert werden.




Einmal aktiviert findet man unter diesem Menüpunkt den Button "Visual Basic" mit dem man in den Entwicklerbereich kommt.
Dies sieht wiefolgt aus:




Hier können die benötigten Skripte angepasst werden und genutzt werden um die Daten aus IP-Symcon weiterzuverarbeiten.
Auch befindet sich in diesen Skripten die Logik zum Verbindungsaufbau zu IP-Symcon, sowie die dafür benötigte JSON-RPC Schnittstelle.

# SDK (PHP)

Benötigt: IP-Symcon >= 4.0

## BESCHREIBUNG

Ab der Version 4.0 können Module mithilfe von PHP erstellt werden.

## VORAUSSETZUNGEN

PHP Kenntnisse
IP-Symcon 4.0 oder höher

## VIDEO-TUTORIALS

Für viele Themen rund um das SDK stehen Video-Tutorials auf unserem YouTube-Kanal bereit.

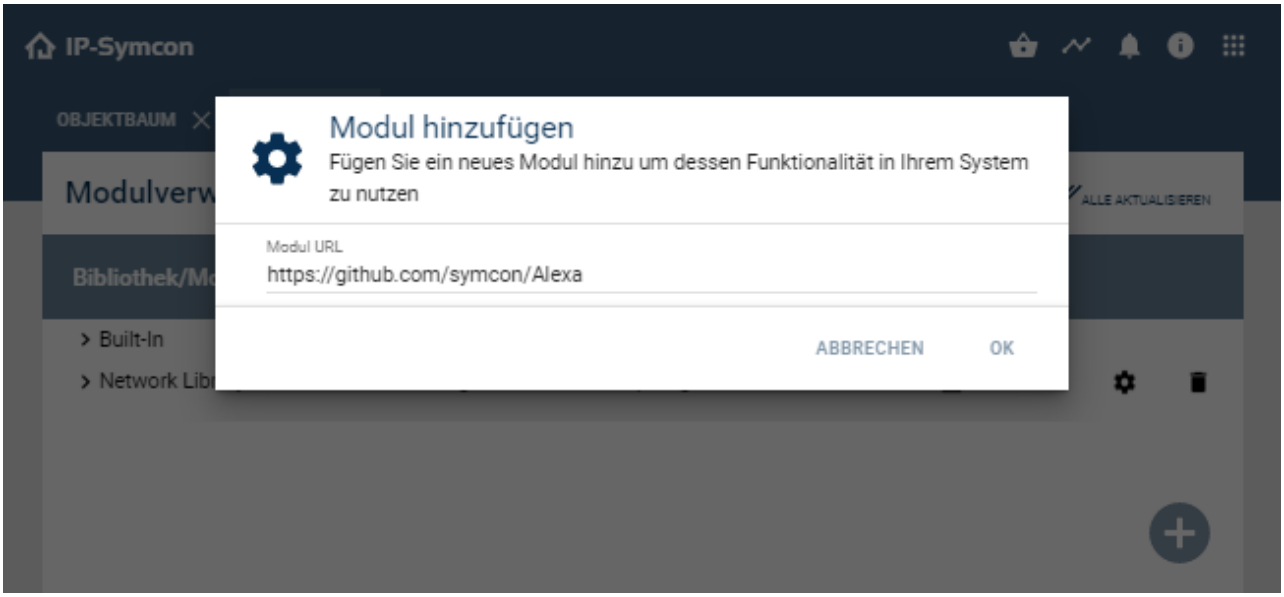
## EINBINDUNG IN IP-SYMCON

Die Datenstruktur kann hier nachgesehen werden.
Der erste Schritt wäre das Erstellen der Bibliothek , sowie das Einrichten aller benötigen Dateien.
Diese können ebenfalls der Strukturübersicht entnommen werden.
Anschließend können die Module entwickelt werden.



Um ein Modul in der lokalen Entwicklung neuzuladen ist jeweils ein kurzer Neustart des Dienstes nötig. Dieser Schritt ist bei einem Update via Repository nicht nötig.

Nach vollendeter Entwicklung wird es empfohlen, dass die Bibliothek samt Modul/-en in einem Repository zusammengefasst wird.
Beispielsweise wie in diesem: Rechenmodule
Anschließend können die Module, direkt über die Repository-URL, im Module Control hinzugefügt werden. Auch ist es mögich dieses Modul für den Module-Store einzureichen.


IP-Symcon
OBJEKTBAUM
Modulverw
ALLE AKTUALISIEREN
Bibliothek/Mo
> Built-In
> Network Libr
Modul hinzufügen
Fügen Sie ein neues Modul hinzu um dessen Funktionalität in Ihrem System zu nutzen
Modul URL
https://github.com/symcon/Alexa
ABBRECHEN
OK

# Aktionen

Benötigt: IP-Symcon >= 6.0

## BESCHREIBUNG

Es ist möglich eigene Aktionen als Teil der Bibliothek zu definieren. Diese können nach der Installation der Bibliothek wie unter Aktionen beschrieben verwendet werden.
Werden Aktionen abgespeichert oder konfiguriert resultiert dies in drei Werten: Dem Aktionsziel, der AktionsID und den Aktionsparametern. Das Aktionsziel stellt hierbei das Zielobjekt dar, beispielsweise im Falle eines Ereignisses das darunterliegende Objekt. Die AktionsID identifiziert eindeutig die ausgewählte Aktion. Die Parameter sind eine Menge an Werten, welche individuell zur Aktion gehören. Hierdurch kann ein Benutzer eine Aktion weiter parametrisieren und beispielsweise angeben auf welchen Wert eine Variable geschaltet werden soll.

## STRUKTUR

Für jede Aktion, welche Teil der Bibliothek sein soll, muss eine JSON-Datei im Unterordner "actions" der Bibliothek erstellt werden. Diese JSON-Datei muss ein JSON-Objekt beinhalten, welches der folgenden Struktur folgt.

| PARAMETER | DATENTYP | BESCHREIBUNG |
|---|---|---|
| id | string | Einmalige GUID zur eindeutigen Identifikation der Aktion. GUID Generator |
| caption | string | Sichtbare Bezeichnung der Aktion |
| form | array/ string | Dieses Feld kann ein Array von Konfigurationselementen beinhalten wie in den Bereichen "elements" oder "actions" der Konfigurationsformulare. Alternativ kann das Formular durch ein PHP-Skript dynamisch generiert werden. In diesem Fall enthält dieses Feld den entsprechenden Code, welcher das Konfigurationsformular als Objekt zurückgibt. Der PHP-Code kann auf mehrere Strings in einem Array aus Strings aufgeteilt werden. Soll eine bereits definierte Aktion bearbeitet werden, beispielsweise in einem Ereignis, so stehen die definierten Parameter als Systemvariablen zur Verfügung, siehe SelectAction. Die Einstellungen des Formulars stellen die Parameter der Aktion dar, wobei jedes benannte Konfigurationselement seinen Wert unter seinem "name" zur Verfügung stellt. |
| action | array/ string | PHP-Code, welcher die Ausführung der Aktion definiert. Wird die Aktion ausgeführt, so wird der PHP-Code aus diesem Feld ausgeführt. Dabei stehen die Parameter als Systemvariablen zur Verfügung. |
| priority (optional) | int | Priorität der Aktion, in einer Auswahl werden Aktionen nach der Priorität sortiert |
| category (optional) | string | (**default:** "other") Die Kurzform der Kategorie der Aktion. Mögliche Kategorien sind unter category aufgelistet. |
| restrictions (optional) | array | (**default:** []) Eine Liste von Restriktionen, welche einschränken, unter welchen Bedingungen diese Aktion angeboten wird. Die möglichen Restriktionen sind in den Tabellen restrictions beschrieben. |
| locale (optional) | object | (**default:** []) Hier kann wie in der Lokalisierung eines Moduls die Aktion lokalisiert werden. Hierdurch wird sowohl die Form als auch die caption lokalisiert. |
| format (optional) | string | (**default:** Inhalt von caption) Dieser String stellt eine Kurzformatierung der Aktion mit ihren Parametern dar, welche an verschiedenen Stellen angezeigt werden kann. Die Formatierung wird in ICU-Syntax notiert. Dort können als Variablen die Parameter (Name entsprechend des Parameters) sowie des Ziel (Name "TARGET") verwendet werden. Für die Formatierung stehen einige zusätzliche Symcon-spezifische Typen zur Verfügung. Der Inhalt von format wird durch die locale übersetzt. |
| description (optional) | string | (**default:** "") Dieser Text wird als Beschreibung für die Aktion in der Aktionsauswahl angezeigt. Der Inhalt von description wird durch die locale übersetzt. |
| readable (optional) | array/ string | (**default:** Inhalt von action, wobei alle Statusvariablen von Parametern durch die eingestellten Werte ersetzt werden) PHP-Code, welcher die Aktion als lesbaren Code zurückgibt. Dies wird beispielsweise für "Befehle einfügen" im Skripteditor verwendet. Dabei stehen die Parameter als Systemvariablen zur Verfügung. |
| | | (**default:** Nicht veraltet) Wenn dieser Parameter gesetzt ist, wird die Aktion als veraltet betrachtet und wird in einer Auswahl nicht angezeigt. Wenn ein Dialog zum Bearbeiten dieser Aktion |

## CATEGORY

| KURZFORM | NAME | BESCHREIBUNG |
|---|---|---|
| target | Zielspezifisch | Aktionen, die zu einem spezifischen Ziel gehören, üblicherweise einem Modul |
| math | Mathematische Operationen | Mathematische Operationen um Objekte zu modifizieren |
| set | Wert setzen | Aktionen, die den Wert einer Variable setzen |
| switch | Wert schalten | Aktionen, die den Wert einer Variable schalten |
| expert | Experte | Aktionen, die tieferes Verständniss über IP-Symcon oder PHP erfordern |
| other | Andere Aktionen | Aktionen, die in keine andere Kategorie passen |

## RESTRICTIONS

Restriktionen für alle Objekttypen

| PARAMETER | DATENTYP | BESCHREIBUNG |
|---|---|---|
| objectType | array | Die Aktion wird nur für Ziele der angegebenen Typen angeboten (0 = Kategorie, 1 = Instanz, 2 = Variable, 3 = Skript, 4 = Ereignis, 5 = Media, 6 = Link) |
| includeEnvironments | array | Eine Liste von Umgebungen in welchen die Aktion angeboten wird, siehe Umgebungen |
| excludeEnvironments | array | Eine Liste von Umgebungen in welchen die Aktion nicht angeboten wird. Bei Übereinstimmungen mit "includeEnvironments" wird die Aktion nicht angezeigt. Siehe Umgebungen |
| writable | bool | Ist diese Restriktion auf true gesetzt, so wird die Aktion nur angeboten, falls das Zielobjekt nicht schreibgeschützt sein. Ist das Ziel eine Variable, so darf diese zusätzlich keine Variablenaktion haben. |

Restriktionen für Zielobjekte vom Typ Instanz

| PARAMETER | DATENTYP | BESCHREIBUNG |
|---|---|---|
| hasActionStatusVariables | bool | Ist diese Restriktion auf true gesetzt, so wird die Aktion nur angeboten, falls die Instanz mindestens eine schaltbare Statusvariable hat |
| hasInstanceFunctions | bool | Ist diese Restriktion auf true gesetzt, so wird die Aktion nur angeboten, falls die Instanz mindestens eine öffentliche Funktion besitzt, welche nicht vom IPSModule geerbt wurde |
| moduleID | array | Die Aktion wird nur angeboten, falls die ModulID der Zielinstanz in dieser Liste enthalten ist |
| hasIdent | array | Die Aktion wird nur angeboten, falls die Zielinstanz für jeden Eintrag dieser Liste ein Kind mit dem entsprechenden Ident hat (seit IP-Symcon 6.1) |
| hasNoIdent | array | Die Aktion wird nur angeboten, falls die Zielinstanz für jeden Eintrag dieser Liste KEIN Kind mit dem entsprechenden Ident hat (seit IP-Symcon 8.0) |

Restriktionen für Zielobjekte vom Typ Variable

| PARAMETER | DATENTYP | BESCHREIBUNG |
|---|---|---|
| requestAction | bool | Ist diese Restriktion auf true gesetzt, so wird die Aktion nur angeboten, falls die Variable über eine Variablenaktion verfügt |
| variableType | array | Die Aktion wird nur angeboten, falls der Variablentyp der Zielvariablen in dieser Liste enthalten ist (0 = Boolean, 1 = Integer, 2 = Float, 3 = String) |
| profilesInclude | array | Die Aktion wird nur angeboten, falls der Name ihres aktuell verwendetes Profils in dieser Liste enthalten ist |
| profilesExclude | array | Die Aktion wird nur angeboten, falls der Name ihres aktuell verwendeten Profils NICHT in dieser Liste enthalten ist |
| profileIsEnum | bool | Ist diese Restriktion auf true gesetzt, so wird die Aktion nur angeboten, falls das Profil der Variablen eine Aufzählung darstellt (Schrittweite = 0, mindestens eine Assoziation) |
| profileIsPercentage | bool | Ist diese Restriktion auf true gesetzt, so wird die Aktion nur angeboten, falls das Profil der Variablen einen Prozentwert darstellt (Suffix = "%", Max Wert > Min Wert) |
| presentation | array | Die Aktion wird nur angeboten, falls die ID der verwendeten Darstellung in dieser Liste enthalten ist |

Restriktionen für Zielobjekte vom Typ Automation

| PARAMETER | DATENTYP | BESCHREIBUNG |
|---|---|---|
| scriptType | array | Die Aktion wird nur angeboten, falls der Automationstyp der Automation in dieser Liste enthalten ist (0 = PHP-Skript, 1 = Ablaufplan, 2 = IPSWorkflow) (ab Version 6.1) |

Restriktionen für Zielobjekte vom Typ Ereignis

| PARAMETER | DATENTYP | BESCHREIBUNG |
|---|---|---|
| eventType | array | Die Aktion wird nur angeboten, falls der Ereignistyp des Ereignis in dieser Liste enthalten ist (0 = Ausgelöst, 1 = Zyklisch, 2 = Wochenplan) (ab Version 6.3) |

Umgebungen

| UMGEBUNG | BESCHREIBUNG |
|---|---|
| Default | Die Standardumgebung in welcher die meisten Aktionen definiert sind, viele Auswahlen beschränken sich auf diese Aktionen oder bieten sie zusätzlich an |
| TestCommands | Aktionsauswahl im "Befehle testen"-Dialog einer Instanz, hier werden "Default"-Aktionen nicht mit angeboten |
| EventTrigger | Aktionsauswahl in einem ausgelösten Ereignis |
| EventCyclic | Aktionsauswahl in einem zyklischen Ereignis |
| EventSchedule | Aktionsauswahl in einem Wochenplan |
| FlowScript | Aktionsauswahl im Ablaufplan |
| ScriptEditor | Aktionsauswahl im "Befehl hinzufügen"-Dialog eines Skripteditors |
| Weitere Umgebungen | Es können eigene Umgebungen definiert werden, um beispielsweise spezielle Aktionen als Teil einer Instanzkonfiguration anzubieten |

## SYSTEMVARIABLEN

Sowohl in einem möglichen PHP-Skript für die Form als auch in dem Skript für die Ausführung kann via Systemvariablen auf das Ziel der Aktion und die eingestellten Parameter zugegriffen werden. So stehen folgende Systemvariablen in diesen Skripten zur Verfügung:

| SYSTEMVARIABLE | BESCHREIBUNG |
|---|---|
| $_IPS['TARGET'] | ObjektID des Ziels der Aktion |
| $_IPS[(Name des Parameters)] (z.B. $_IPS['VALUE']) | Der eingestellte bzw. ausgewählte Wert des Parameters. Diese Felder stehen in einem Skript für die Form nicht zur Verfügung, falls die Aktion initial ausgewählt wird und somit noch keine Parameter ausgewählt wurden. Daher sollte dort mit der PHP-Funktion isset geprüft werden, ob die Systemvariablen vorhanden sind. |
| Weitere Systemvariablen | Weitere Systemvariablen stehen basierend auf der Ausführung der Aktion zur Verfügung. Sollte es hierbei zu Überschneidungen mit Parameternamen kommen, werden die Parameterwerte verwendet und es kann nicht auf die regulären Systemvariablen zugegriffen werden. |

## FORMATIERUNG

Im Feld format können alle Typen des ICU-Formats verwendet werden. Zusätzlich wurden Symcon-spezifische Typen eingeführt. Wie bei ICU üblich, bestehen diese aus drei Teilen: Der Parameterbezeichnung, der Typbezeichnung und optionalen zusätzlichen Parametern.

profile

Der Wert wird auf Basis des Profils der Zielvariablen dargestellt. Hat das Ziel kein Profil, so

wird der Wert direkt dargestellt.
Beispiele: {VALUE, profile}

valueFormatted

Es wird der formatierte Wert der Variablen mit der ObjektID des Wertes dargestellt.
Beispiele: {VARIABLE, valueFormatted}

object

Es wird der Name des Objekts mit der ObjektID des Wertes dargestellt.

| PARAMETER | BESCHREIBUNG |
|---|---|
| ident (optional) | Wird als Parameter "ident" gefolgt von einer Parameterbezeichnung angegeben, so wird stattdessen der Name des Kindes mit dem Ident, welcher dem Parameter entspricht, dargestellt. Steht der Wert hinter ident in einfachen Anführungszeichen ('), so wird der Wert dazwischen direkt als Ident verwendet. |

Beispiele: {VARIABLE, object}, {TARGET, object, ident IDENT}

action

Es wird die Formatierung der Aktion, welche als JSON-String codiert im Wert steht, dargestellt.

| PARAMETER | BESCHREIBUNG |
|---|---|
| targetID (optional) | Wird als Parameter "targetID" gefolgt von einer Parameterbezeichnung angegeben, so wird die ZielID der formatierten Aktion auf den Wert des entsprechenden Parameters gesetzt. Wird zusätzlich direkt danach "ident" gefolgt von einer Parameterbezeichnung angegeben, so wird die ZielID stattdessen auf die ObjektID des Kindes mit dem Ident, welcher dem Parameter entspricht, dargestellt. Steht der Wert hinter ident in einfachen Anführungszeichen ('), so wird der Wert dazwischen direkt als Ident verwendet. |
| linkTarget (optional) | Wird der Parameter "linkTarget" gesetzt, so wird die ZielID des Links als Ziel der formatierten Aktion verwendet. Bei "linkTarget" müssen keine zusätzlichen Parameter angegeben werden. "linkTarget" kann mit targetID kombiniert werden, indem "linkTarget" danach angegeben wird. Dann wird das beschriebene Ziel als Link betrachtet und dessen Ziel für die Aktion verwendet. |

Beispiele: {ACTION, action}, {ACTION, action, targetID VARIABLE}, {ACTION, action, targetID TARGET ident IDENT}, {ACTION, action, linkTarget}, {ACTION, action, targetID LINK linkTarget}

scheduleAction

Es werden die Namen von Wochenplanaktionen von Wochenplänen unter dem angegebenen Elternobjekt mit der ID des Wertes dargestellt.

| PARAMETER | BESCHREIBUNG |
|---|---|
| parent | Wird als Parameter "parent" gefolgt von einer Parameterbezeichnung angegeben, so wird als Elternobjekt des Objekt mit dem Wert als ID verwendet |
| action | Wird als Parameter "action" gefolgt von einer AktionsID in einfachen Anführungszeichen (') angegeben, so werden nur Namen von Wochenplanaktionen dargestellt, welche die entsprechende Aktion ausführen |

Beispiele: {SCHEDULE_ACTION, scheduleAction, parent PARENT}, {SCHEDULE_ACTION, scheduleAction, parent PARENT action '{7938A5A2-0981-5FE0-BE6C-8AA610D654EB}'}

selectPresentation

Verschiedene Formatierungen je nach verwendeter Darstellung einer Variablen

| PARAMETER | BESCHREIBUNG |
| --- | --- |
| ID einer Darstellung | Wird als Parameter die ID einer Darstellung gefolgt von einer Formatierung verwendet, so wird die Formatierung genutzt, falls die Variable die Darstellung verwendet |
| other | Wird als Parameter "other" gefolgt von einer Formatierung verwendet, so wird die Formatierung genutzt, falls die ID der verwendeten Darstellung sonst nicht als Parameter zu finden war |

Beispiele: {TARGET, selectPresentation, {6B9CAEEC-5958-C223-30F7-BD36569FC57A}{Schieberegler} {05CC3CC2-A0B2-5837-A4A7-A07EA0B9DDFB}{Farbe} other{Sonstige Darstellung}}

## BEISPIELE

switchValueString.json

```
{
"id": "{A4D52B67-BE4B-4AD0-964F-B9BA2556AAB0}",
"caption": "Switch to Value",
"form": [
{
"type": "ValidationTextBox",
"name": "VALUE",
"caption": "Value"
}
],
"action": "RequestAction($_IPS['TARGET'], $_IPS['VALUE']);",
"restrictions": {
"objectType": [ 2 ],
"variableType": [ 3 ],
"profileIsEnum": false,
"requestAction": true
},
"priority": 10,
"locale": {
"de": {
"Switch to Value": "Schalte auf Wert",
"Value": "Wert",
"Set to {VALUE, profile}": "Setze auf {VALUE, profile}"
}
},
"format": "Set to {VALUE, profile}"
}
```

setValueIntegerPreviousReminder.json

```
{
"id": "{92971E6F-4AC1-BEF8-B325-9E044D27B3EB}",
"caption": "Set to Value",
"form": [
"$form = [",
" {",
```

```
" 'type' => 'SelectColor',",
" 'name' => 'VALUE',",
" 'allowTransparent' => false,",
" 'caption' => 'Value',",
" 'writable' => true",
" }",
"];",
"if (isset($_IPS['VALUE'])) {",
" $form[] = {",
" 'type' => 'Label'",
" 'caption' => 'Previously selected value: ' . $_IPS['VALUE']",
" };",
"}",
"return $form;"
],
"action": "SetValue($_IPS['TARGET'], $_IPS['VALUE']);",
"restrictions": {
"objectType": [ 2 ],
"variableType": [ 1 ],
"profilesExclude": [ "~HexColor" ]
},
"priority": 5,
"locale": {
"de": {
"Set to Value": "Setze auf Wert",
"Value": "Wert",
"Set to {VALUE, profile}": "Setze auf {VALUE, profile}"
}
},
"format": "Set to {VALUE, profile}"
}
```

switchStatusVariable.json

```
{
"id": "{E616C2B2-A827-1712-4AE0-841C57B3DD74}",
"caption": "Switch Status Variable",
"form": [
"$options = [];",
"$firstVariable = false;",
"foreach (IPS_GetChildrenIDs($_IPS['TARGET']) as $childID) {",
" $object = IPS_GetObject($childID);",
" if (($object['ObjectType'] === 2) && ($object['ObjectIdent'] !==
'')) {",
" if (!HasAction($childID)) {",
" continue;",
" }",
" if (!$firstVariable) {",
" $firstVariable = $object;",
" }",
" $options[] = [",
" 'caption' => $object['ObjectName'],",
" 'value' => $object['ObjectIdent']",
" ];",
" }",
"}",
```

```
"return [",
" [",
" 'type' => 'Select',",
" 'name' => 'IDENT',",
" 'caption' => 'Status Variable',",
" 'options' => $options,",
" 'value' => $firstVariable['ObjectIdent'],",
" 'onChange' => 'IPS_UpdateFormField(\"ACTION\", \"targetID\",
IPS_GetObjectIDByIdent($IDENT, ' . $_IPS['TARGET'] . '), $id);'",
" ],",
" [",
" 'type' => 'SelectAction',",
" 'name' => 'ACTION',",
" 'caption' => 'Action',",
" 'targetID' => isset($_IPS['IDENT']) ?
IPS_GetObjectIDByIdent($_IPS['IDENT'], $_IPS['TARGET']) :
$firstVariable['ObjectID']",
" ]",
"];"
],
"action": [
"$targetVariableID = IPS_GetObjectIDByIdent($_IPS['IDENT'],
$_IPS['TARGET']);",
"$action = json_decode($_IPS['ACTION'], true);",
"echo IPS_RunActionWait($action['actionID'], $targetVariableID,
$action['parameters']);"
],
"restrictions": {
"objectType": [ 1 ],
"hasActionStatusVariables": true
},
"locale": {
"de": {
"Switch Status Variable": "Schalte Statusvariable",
"Status Variable": "Statusvariable",
"{TARGET, object, ident IDENT}: Switch Variable": "{TARGET, object,
ident IDENT}: Schalte Variable"
}
},
"format": "{TARGET, object, ident IDENT}: {ACTION, action, targetID
TARGET ident IDENT}",
"readable": [
"$targetVariableID = IPS_GetObjectIDByIdent($_IPS['IDENT'],
$_IPS['TARGET']);",
"$action = json_decode($_IPS['ACTION'], true);",
"$action['parameters']['TARGET'] = $targetVariableID;",
"$action['parameters']['ENVIRONMENT'] = $_IPS['ENVIRONMENT'];",
"$action['parameters']['PARENT'] = $_IPS['PARENT'];",
"echo IPS_GetActionReadableCode($action['actionID'],
$action['parameters']) . \"\\n\";"
]
}
```

multiplySetVariable.json (veraltete Aktion)

```
{
```

```
"id": "{085176EC-BEE5-3732-15E5-8C82875CABD4}",
"caption": "Multiply with Value of other Variable",
"form": [
"$variable = IPS_GetVariable($_IPS['TARGET']);",
"$types = [];",
"if ($variable['VariableType'] === 2) {",
" $types = [1, 2];",
"}",
"else {",
" $types = [ $variable['VariableType'] ];",
"}",
"return [",
" [",
" 'type' => 'SelectVariable',",
" 'name' => 'VARIABLE',",
" 'caption' => 'Variable',",
" 'validVariableTypes' => $types",
" ]",
"];"
],
"action": "SetValue($_IPS['TARGET'], GetValue($_IPS['TARGET']) *
GetValue($_IPS['VARIABLE']));",
"restrictions": {
"objectType": [ 2 ],
"variableType": [ 1, 2 ],
"profileIsEnum": false,
"writable": true
},
"locale": {
"de": {
"Multiply with Value of other Variable": "Multipliziere mit Wert
einer anderen Variablen",
"Variable": "Variable",
"Multiply by value of {VARIABLE, object}": "Multipliziere mit Wert
von {VARIABLE, object}",
"Set the target variable to its current value multiplied with the
value of another variable": "Setze die Zielvariable auf ihren
aktuellen Wert multipliziert mit dem Wert einer anderen Variablen"
}
},
"format": "Multiply by value of {VARIABLE, object}",
"category": "math",
"description": "Set the target variable to its current value
multiplied with the value of another variable",
"deprecated": [
"return [",
" 'actionID' => '{A3153696-013A-41B1-A001-5E8085D95465}',",
" 'parameters' => [",
" 'DYNAMIC' => true,",
" 'FACTOR' => 0,",
" 'VARIABLE' => $_IPS['VARIABLE']",
" ]",
"];"
]
}
```

# Bibliotheken

Benötigt: IP-Symcon >= 4.0

## BESCHREIBUNG

Die Bibliothek ist die Basis für jede Modulentwicklung. Es können auch mehrere Module zu einer Bibliothek zusammengefasst werden. Die benötigte Verzeichnisstruktur kann unter Struktur eingesehen werden.
Diese kann über ein Repository (wir empfehlen Github oder Bitbucket) zur Verfügung gestellt werden.

## EINBINDUNG IN IP-SYMCON

Die Datei library.json muss vorhanden sein, welche im Hauptverzeichnis liegt.
Anhand der Verzeichnisstruktur kann IP-Symcon die gesamte Bibliothek über das "Module Control" einlesen.

| PARAMETER | DATENTYP | BESCHREIBUNG |
|---|---|---|
| id | string | Jedes Modul besitzt zur eindeutigen Identifikation eine eigene GUID (siehe unten Infobox). GUID Generator |
| author | string | Unter welchem Namen wurde die Bibliothek entwickelt |
| compatibility (ab 4.3) | array | Überprüft ob die benötigte Version installiert ist. Es kann auf Kernelversion und/oder Datum geprüft werden. (Weitere Beschreibung siehe Tabelle) |
| name | string | Der Name der gesamten Bibliothek. (A-Z, a-z, 0-9, Leerzeichen, Unterstrich sind erlaubte Zeichen. Leerzeichen und Unterstriche dürfen dabei jedoch nicht am Anfang oder Ende sein. Ein leerer Name ist ebenfalls nicht gültig.) |
| url | string | URL zu der Homepage (Muss mit http:// oder https:// beginnen. Darf Alternativ auch leer sein) |
| version | string | Versionsnummer. Diese kann als String beliebig definiert werden. Wir empfehlen eine Darstellung in der Form "Zahl.Zahl". Bsp: "4.2" |
| build | integer | Buildnummer |
| date | integer | Unix-Zeitstempel |



Die GUID ist eine UUID und hat das Format 8-4-4-4-12. Die Zahlen geben die Anzahl der Ziffern an. Die Ziffern bestehen aus Zeichen zwischen 0-9 und A-F. Es müssen immer Bindestriche und geschweifte Klammern vorhanden sein und es dürfen nur Großbuchstaben verwendet werden. (Beispiel: {12345678-90AB-CDEF-1234-567890ABCDEF} )

### COMPATIBILITY

| PARAMETER | DATENTYP | BESCHREIBUNG |
| --- | --- | --- |
| version (optional) | string | Mindest-Version als String. Bsp: "4.2" |
| date (optional) | integer | Datum als UnixTimestamp. Bsp: 1491343200 |

## BEISPIELE

library.json

```
{
"id": "{F96B257F-85E7-47CF-8340-8FE850AACD10}",
"author": "Symcon GmbH",
"name": "Misc Modules",
"url": "https://www.symcon.de",
"compatibility": {
"version": "4.2",
"date": 1491343200
},
"version": "1.0",
"build": 0,
"date": 0
}
```

# Darstellungen

Benötigt: IP-Symcon >= 8.0

## BESCHREIBUNG

Eine visuelle Übersicht aller Darstellungen gibt es in der Objekt-Darstellung.
Die Darstellung einer Variable kann über die Funktionen IPS_SetVariableCustomPresentation, RegisterVariable* und MaintainVariable gesetzt werden. Die Darstellung einer Variable wird über einen Array mit folgenden "Key"->"Value" Paaren bestimmt.

| KEY | BESCHREIBUNG |
|---|---|
| PRESENTATION | Die als GUID formatierte ID der zu nutzenden Darstellung |
| TEMPLATE | (optional) Die als GUID formatierte ID der zu nutzenden Vorlage |
| *Darstellungsparameter* | (optional) Anstelle einer Vorlage können auch beliebig viele Parameter einer Darstellung gesetzt werden. Die Parameter werden auf der gleichen Ebene gesetzt wie der PRESENTATION Parameter |



Wird ein Parameter direkt gesetzt werden die Werte der Vorlage ignoriert

Die verfügbaren Darstellungen entsprechenden Vorlagen sind hier zu finden.
Beispiel:
Die Darstellung einer Variable wird bei Erstellung auf einen Schieberegler mit einem eigenen Suffix gesetzt.

```
$this->RegisterVariableFloat('Value', 'Value', [
'PRESENTATION' => VARIABLE_PRESENTATION_SLIDER,
'SUFFIX' => ' %'
]);
```

### DARSTELLUNGEN

**DARSTELLUNG**

- Aufzählung
- Datum/Uhrzeit
- Dauer
- Farbe
- Rollladen
- Schalter
- Schieberegler
- Webinhalt
- Wertanzeige
- Werteingabe

# Aufzählung

Benötigt: IP-Symcon >= 8.0

## PARAMETER

Informationen zu den visuellen Eigenschaften und Bedeutungen der Parameter sind in der Objekt-Darstellung zu finden.

| NAME | TYP | PARAMETER | BESCHREIBUNG |
|---|---|---|---|
| ICON | String | Standard Icon | Icons |
| OPTIONS | String | Optionen | JSON-kodierte Liste von Objekten, die genauen Parameter sind unter Optionen beschrieben |
| LAYOUT | Integer | Anordnung | 0: Spalte, 1: Reihe, 2: Gitter |

## OPTIONEN

| NAME | TYP | PARAMETER | BESCHREIBUNG |
|---|---|---|---|
| Value | Boolean/String/Integer/ Float | Wert | Abhängig vom Typ der Variable |
| Caption | String | Beschriftung | |
| IconActive | Boolean | Überschreibe Icon | |
| IconValue | String | Icon | Icons |
| Color | Integer | Farbe | |

# Datum/Uhrzeit

Benötigt: IP-Symcon >= 8.0

## PARAMETER

Informationen zu den visuellen Eigenschaften und Bedeutungen der Parameter sind in der Objekt-Darstellung zu finden.

| NAME | TYP | PARAMETER | BESCHREIBUNG |
|---|---|---|---|
| DATE | Integer | Datumsanzeige | 0: Kein, 1: Jahr, Monat und Tag, 2: Monat und Tag, 3: Jahr und Tag |
| MONTH_TEXT | Boolean | Formatiere Monat als | false: Zahl, true: Text |
| DAY_OF_THE_WEEK | Boolean | Zeige Wochentag | |
| TIME | Integer | Zeitanzeige | 0: Keine, 1: Stunden und Minuten, 2: Stunden, Minuten und Sekunden |

# Dauer

Benötigt: IP-Symcon >= 8.0

## PARAMETER

Informationen zu den visuellen Eigenschaften und Bedeutungen der Parameter sind in der Objekt-Darstellung zu finden.

| NAME | TYPE | PARAMETER | BESCHREIBUNG |
|---|---|---|---|
| COUNTDOWN_TYPE | Integer | Darstellungstyp | 0: Wert in Variable, 1: Dauer bis zum Wert in der Variable, 2: Dauer seit Wert in der Variable |
| FORMAT | Integer | Format | 0: Nur Sekunden, 1: Minuten und Sekunden, 2: Stunden, Minuten und Sekunden |
| MILLISECONDS | Boolean | Zeige Millisekunden | |

# Farbe

## PARAMETER

Informationen zu den visuellen Eigenschaften und Bedeutungen der Parameter sind in der Objekt-Darstellung zu finden.

| NAME | TYP | PARAMETER | BESCHREIBUNG |
|---|---|---|---|
| ENCODING | Integer | Kodierung | 0: RGB, 1: CMYK, 2: HSV, 3: HSL 4: xy |
| PRESET_VALUES | String | Standardwerte | JSON-kodierte Liste von Objekten, die genauen Parameter sind unter Farben beschrieben. Nicht verfügbar bei Kodierung xy |
| COLOR_SPACE | Integer | Farbraum | 0: Benutzerdefiniert, 1: sRGB, 2: AdobeRGB, 3: DCI-P3, 4: Rec2020 |
| CUSTOM_COLOR_SPACE | String | Benutzerdefinierter Farbraum | JSON-kodierte Liste von Objekten, die genauen Parameter sind unter Benutzerdefinierter Farbraum beschrieben |
| COLOR_CURVE | Integer | Farbkurve | 0: Keine, 1: Benutzerdefiniert, 2: Tageslicht, 3: Tageslicht (Frühling), 4: Tageslicht (Sommer), 5. Tageslicht (Winter) |
| CUSTOM_COLOR_CURVE | String | Benutzerdefinierte Farbkurve | JSON-kodierte Liste von Objekten, die genauen Parameter sind unter Benutzerdefinierter Farbkurve beschrieben |

## FARBEN

| PARAMETER | TYP | BESCHREIBUNG |
|---|---|---|
| Color | Integer | Farbe als Integer |

## BENUTZERDEFINIERTER FARBRAUM

Ein Farbraum wird durch 4 Objekte mit den Folgenden Parametern beschrieben. Der 1. Eintrag ist dabei für Rot, der 2. für Grün, der 3. für Blau und der 4. für den Weißpunkt.

| PARAMETER | TYP | BESCHREIBUNG |
|---|---|---|
| x | Float | X-Position der Farbe auf der CIE-Normfarbtafel im Bereich 0-1 |
| y | Float | Y-Position der Farbe auf der CIE-Normfarbtafel im Bereich 0-1 |

Beispiel:

```
// sRGB
[
{"x":0.64,"y":0.33},
{"x":0.3,"y":0.6},
{"x":0.15,"y":0.06},
{"x":0.3127,"y":0.329}
]
```

## BENUTZERDEFINIERTER FARBKURVE

Eine Farbkurve besteht aus einer beliebigen Anzahl von Farben, die etwa die Farbtemperatur über den Verlauf eines Tages dartellen.

| PARAMETER | TYP | BESCHREIBUNG |
|---|---|---|
| x | Float | X-Position der Farbe auf der CIE-Normfarbtafel im Bereich 0-1 |
| y | Float | Y-Position der Farbe auf der CIE-Normfarbtafel im Bereich 0-1 |

Beispiel:

```
// Tageslicht (Winter)
[
{"x": 0.477, "y" 0.4137},
{"x": 0.4599, "y" 0.4106},
{"x": 0.433, "y" 0.4027},
{"x": 0.4103, "y" 0.3932},
{"x": 0.3918, "y" 0.3835},
{"x": 0.3761, "y" 0.374},
{"x": 0.3631, "y" 0.3652},
{"x": 0.352, "y" 0.357},
{"x": 0.3439, "y" 0.3507},
{"x": 0.3367, "y" 0.3447},
{"x": 0.3348, "y" 0.343},
]
```

# Rollladen

Benötigt: IP-Symcon >= 8.0

## PARAMETER

Informationen zu den visuellen Eigenschaften und Bedeutungen der Parameter sind in der Objekt-Darstellung zu finden.

| NAME | TYP | PARAMETER | BESCHREIBUNG |
|---|---|---|---|
| USAGE_TYPE | Integer | Verwendungstyp | 0: Offen, 1: Rotation |
| OPEN_OUTSIDE_VALUE | Integer/ Float | Offen bei / Außen bei | |
| CLOSE_INSIDE_VALUE | Integer/ Float | Geschlossen bei / Innen bei | |
| MAX_ROTATION_INSIDE | Integer/ Float | Maximale Rotation nach innen | |
| MAX_ROTATION_OUTSIDE | Integer/ Float | Maximale Rotation nach außen | |
| SUN_POSITION | Integer | Sonnenposition | 0: Links, 1: Rechts, 2: (Keine) |

# Legacy Profil

## PARAMETER

Informationen zu den visuellen Eigenschaften und Bedeutungen der Parameter sind in der Objekt-Darstellung zu finden.

| NAME | TYP | PARAMETER | BESCHREIBUNG |
| --- | --- | --- | --- |
| PROFILE | String | Profil | Variablenprofile |

# Schalter

Benötigt: IP-Symcon >= 8.0

## PARAMETER

Informationen zu den visuellen Eigenschaften und Bedeutungen der Parameter sind in der Objekt-Darstellung zu finden.

| NAME | TYP | PARAMETER | BESCHREIBUNG |
|---|---|---|---|
| USE_ICON_FALSE | Boolean | Individuelle Icons passend zum Wert | |
| ICON | String | Icon | Icons |
| ICON_TRUE | String | Icon für true | Icons |
| ICON_FALSE | String | Icon für false | Icons |
| GLOW_COLOR | Integer | Farbe des Leuchtens wenn aktiv | |
| GLOW_INTENSITY | Integer | Intensität des Leuchtens wenn aktiv | |
| USAGE_TYPE | Integer | Verwendung der Variable | 0: An/Aus, 1: Stumm schalte, 2: Keine hiervon |

# Schieberegler

Benötigt: IP-Symcon >= 8.0

## PARAMETER

Informationen zu den visuellen Eigenschaften und Bedeutungen der Parameter sind in der Objekt-Darstellung zu finden.

| NAME | TYP | PARAMETER | BESCHREIBUNG |
|---|---|---|---|
| MIN | Integer/ Float | Minimaler Wert | |
| MAX | Integer/ Float | Maximaler Wert | |
| STEP_SIZE | Integer/ Float | Schrittweite | |
| GRADIENT_TYPE | Integer | Gradient | 0: Standard, 1: Temperatur, 2: Farbtemperatur, 3: Benutzerdefiniert |
| CUSTOM_GRADIENT | String | Benutzerdefinierter Gradient | JSON-kodierte Liste von Objekten mit den Parametern 'Value' und 'Color', wobei Color analog zu SelectColor codiert wird |
| USAGE_TYPE | Integer | Verwendung der Variable | 0: Temperatur, 1: Farbtemperatur, 2: Intensität, 3: Lautstärke, 4: Fortschritt, 5: Keine hiervon |
| PREFIX | String | Präfix | |
| SUFFIX | String | Suffix | |
| PERCENTAGE | Boolean | Anzeigetyp | false: Absolut, true: Prozentual |
| THOUSANDS_SEPARATOR | String | Tausendertrennzeichen | Client: Standard vom Nutzer, sonst direkt der Wert |
| DIGITS | Integer | Nachkommastellen | |
| DECIMAL_SEPARATOR | String | Dezimaltrennzeichen | Client: Standard vom Nutzer, sonst direkt der Wert |
| ICON | String | Icon | |
| INTERVALS_ACTIVE | Boolean | Verwende aktualisierte Parameter für spezifische Intervalle | |
| INTERVALS | String | Intervalle | JSON-kodierte Liste von Objekten, die genauen Parameter sind unter Intervalle beschrieben |

## INTERVALLE

| NAME | TYP | PARAMETER | BESCHREIBUNG |
|---|---|---|---|
| IntervalMinValue | Integer/ Float | Anfang des Intervalls | |
| IntervalMaxValue | Integer/ Float | Ende des Intervalls | |
| ConstantActive | Boolean | Anzeige | false: Formatierter Wert, true: Konstante |
| ConstantValue | String | Konstante | |
| ConversionFactor | Integer/ Float | Umrechnungsfaktor | |
| PrefixActive | Boolean | Überschreibe Präfix | |
| PrefixValue | String | Präfix | |
| SuffixActive | Boolean | Überschreibe Suffix | |
| SuffixValue | String | Suffix | |
| DigitsActive | Boolean | Überschreibe Nachkommastellen | |
| DigitsValue | Integer | Nachkommastellen | |
| IconActive | Boolean | Überschreibe Icon | |
| IconValue | String | Icon | |

# Webinhalt

Benötigt: IP-Symcon >= 8.0

## PARAMETER

Informationen zu den visuellen Eigenschaften und Bedeutungen der Parameter sind in der Objekt-Darstellung zu finden.

| NAME | TYP | PARAMETER | BESCHREIBUNG |
|---|---|---|---|
| HTML_TYPE | Integer | Anzeigetyp | 0: HTML Inhalt, 1: Webseite |
| PADDING | Boolean | Ränder entfernen | |

# Wertanzeige

Benötigt: IP-Symcon >= 8.0

## PARAMETER

Informationen zu den visuellen Eigenschaften und Bedeutungen der Parameter sind in der Objekt-Darstellung zu finden.

| NAME | TYPE | PARAMETER | BESCHREIBUNG |
|---|---|---|---|
| ICON | String | Standardicon | Icons |
| COLOR | Integer | Standardfarbe | |
| PREFIX | String | Präfix | |
| SUFFIX | String | Suffix | |

### FLOAT UND INTEGER

| NAME | TYPE | PARAMETER | BESCHREIBUNG |
|---|---|---|---|
| USAGE_TYPE | Integer | Verwendung der Variable | 5: Keine hiervon 0, 1: Temperatur |
| PERCENTAGE | Boolean | Anzeigetyp | true: Prozent, false: Absolut |
| MIN | Integer/ Float | Minimaler Wert | |
| MAX | Integer/ Float | Maximaler Wert | |
| THOUSANDS_SEPARATOR | String | Tausendertrennzeichen | Client: Standard vom Nutzer, sonst direkt der Wert |
| DIGITS | Integer | Nachkommastellen | |
| DECIMAL_SEPARATOR | String | Dezimaltrennzeichen | Client: Standard vom Nutzer, sonst direkt der Wert |
| INTERVALS_ACTIVE | Boolean | Verwende aktualisierte Parameter für spezifische Intervalle | |
| INTERVALS | String | Intervalle | JSON-kodierte Liste von Objekten, die genauen Parameter sind unter Intervalle beschrieben |

### INTERVALLE

| NAME | TYP | PARAMETER | BESCHREIBUNG |
|---|---|---|---|
| IntervalMinValue | Integer/ Float | Anfang des Intervalls | |
| IntervalMaxValue | Integer/ Float | Ende des Intervalls | |
| ConstantActive | Boolean | Anzeige | false: Formatierter Wert, true: Konstante |
| ConstantValue | String | Konstante | |
| ConversionFactor | Integer/ Float | Umrechnungsfaktor | |
| PrefixActive | Boolean | Überschreibe Präfix | |
| PrefixValue | String | Präfix | |
| SuffixActive | Boolean | Überschreibe Suffix | |
| SuffixValue | String | Suffix | |
| DigitsActive | Boolean | Überschreibe Nachkommastellen | |
| DigitsValue | Integer | Nachkommastellen | |
| IconActive | Boolean | Überschreibe Icon | |
| IconValue | String | Icon | |
| ColorActive | Boolean | Überschreibe Farbe | |
| Color | Integer | Farbe | |

### BOOLEAN UND STRING

| NAME | TYPE | PARAMETER | BESCHREIBUNG |
|---|---|---|---|
| MULTILINE | Boolean | Mehrere Zeilen | |
| OPTIONS | String | Optionen | JSON-kodierte Liste von Objekten, die genauen Parameter sind unter Optionen beschrieben |

## OPTIONEN

| NAME | TYP | PARAMETER | BESCHREIBUNG |
|---|---|---|---|
| Value | Boolean/String | Wert | Abhängig vom Typ der Variable |
| Caption | String | Beschriftung | |
| IconActive | Boolean | Überschreibe Icon | |
| Icon | String | Icon | Icons |
| Color Active | Boolean | Farbe überschreiben | |
| Color | Integer | Farbe | |

# Werteingabe

Benötigt: IP-Symcon >= 8.0

## PARAMETER

Informationen zu den visuellen Eigenschaften und Bedeutungen der Parameter sind in der Objekt-Darstellung zu finden.

| NAME | TYP | PARAMETER |
|---|---|---|
| PRÄFIX | String | Präfix |
| SUFFIX | String | Suffix |
| MULTILINE | Boolean | Mehrzeilige Eingabe |

# Datenfluss

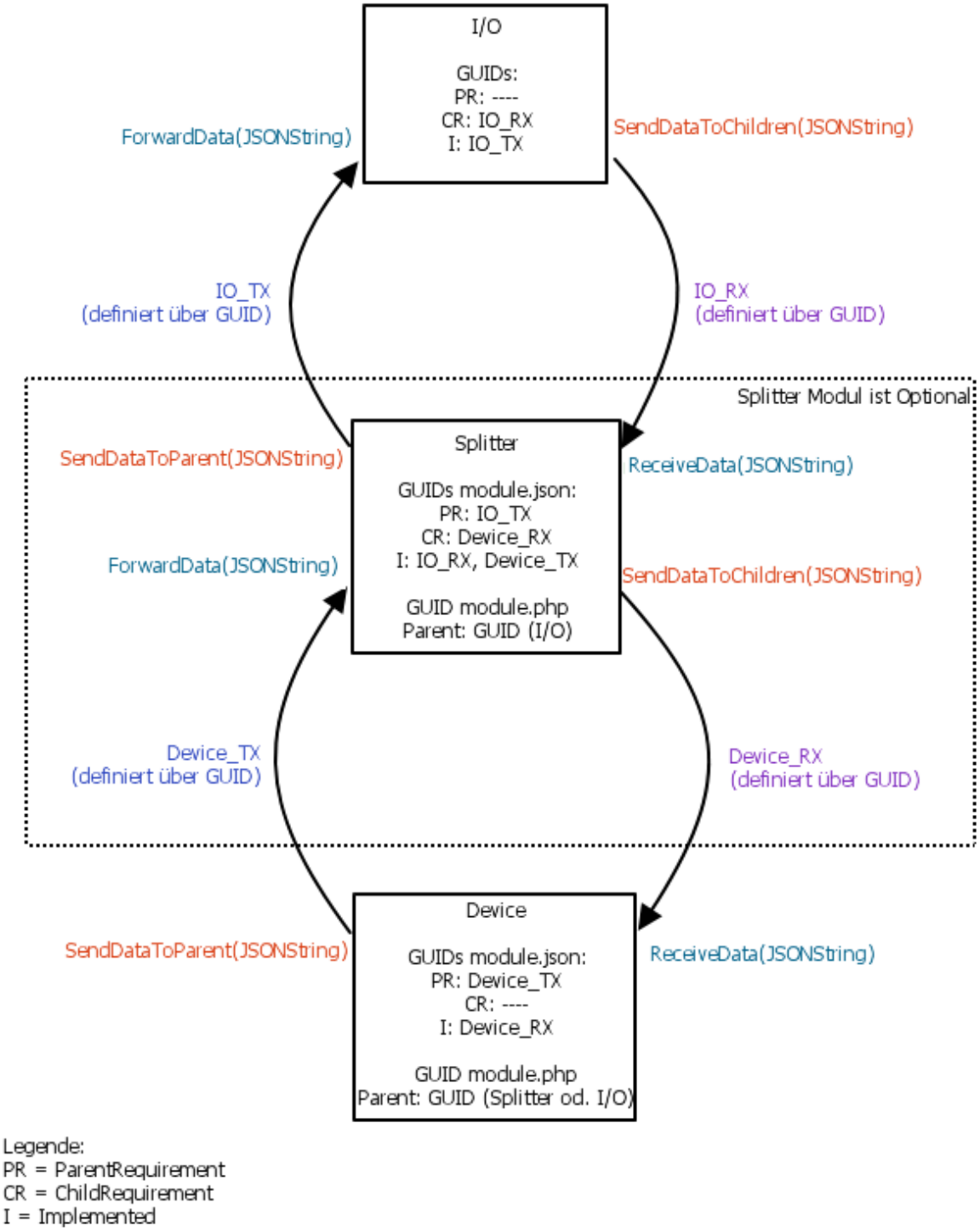
## BESCHREIBUNG

Der Datenfluss soll erklären, wie Daten zwischen "Device <-> Splitter <-> I/O" in IP-Symcon versendet werden und was dafür eingerichtet werden muss.



Für eine einfache Erstellung eines Moduls inklusive des Datenflusses, wird die Nutzung des Modul-Generators empfohlen.

*Typischer Aufbau*




## DATENFLUSSRICHTUNG

Es gibt 2 Datenflussrichtungen.

### I/O ZU DEVICE

Diese Flussrichtung ist, wie in der Abbildung sichtbar, von der übgeordneten Instanz via SendDataToChildren zu den Children. Die ankommenden Daten werden innerhalb der Children von der überschreibaren Funktion ReceiveData verarbeitet.

Zwischen I/O und Device kann optional und falls nötig ein Splitter eingepflegt werden.

### DEVICE ZU I/O

Diese Flussrichtung ist, wie in der Abbildung sichtbar, von der untergeordneten Instanz via SendDataToParent zu dem Parent. Die ankommenden Daten werden innerhalb des Parents von der überschreibaren Funktion ForwardData verarbeitet.
Zwischen Device und I/O kann optional und falls nötig ein Splitter eingepflegt werden.

## I/O MODULE

Bei selbstentwickelten Modulen sind die GUIDs für Devices und Splitter in der jeweiligen module.json Datei anzugeben.
Diese können über den GUID Generator erstellt werden.
*Seitens der I/O stehen folgende Datenpakete für I/O Typen zur Verfügung.*

| I/O MODUL | MODUL GUID | UNTERSTÜTZE DATENPAKETE |
|---|---|---|
| Client Socket | {3CFF0FD9-E306-41DB-9B5A-9D06D38576C3} | Simpel |
| HID | {E6D7692A-7F4C-441D-827B-64062CFE1C02} | Erweitert (Event) |
| HTTP Client | {4CB91589-CE01-4700-906F-26320EFCF6C4} | Erweitert (HTTP Request) |
| Multicast Socket | {BAB408E0-0A0F-48C3-B14E-9FB2FA81F66A} | Simpel<br>Erweitert (Socket) |
| Serial Port | {6DC3D946-0D31-450F-A8C6-C42DB8D7D4F1} | Simpel |
| Server Sent Event Client | {2FADB4B7-FDAB-3C64-3E2C-068A4809849A} | Erweitert (SSE) |
| Server Socket | {8062CF2B-600E-41D6-AD4B-1BA66C32D6ED} | Simpel<br>Erweitert (Socket) |
| UDP Socket | {82347F20-F541-41E1-AC5B-A636FD3AE2D8} | Simpel<br>Erweitert (Socket)<br>Erweitert (UDP) |
| Virtual I/O | {6179ED6A-FC31-413C-BB8E-1204150CF376} | Simpel<br>Erweitert (Socket) |
| WebSocket Client | {D68FD31F-0E90-7019-F16C-1949BD3079EF} | Simpel |

### DATENPAKETE

Die Datenpakete werden JSONString kodiert.
Diese beinhalten einmal die GUID des Datenpakettyps und die eigentlichen Daten.
Über die GUID als Identifikator weiß das jeweilge Modul, welche Daten in dem Datenpaket sind und deren Formatierung.

### SIMPEL

RX GUID: {018EF6B5-AB94-40C6-AA53-46943E824ACF}

TX GUID: {79827379-F36E-4ADA-8A95-5F8D1DC92FA9}

| PARAMETER | DATENTYP | BESCHREIBUNG |
|---|---|---|
| Buffer | String | Beliebiger Dateninhalt |

```
// Beispiel für das Senden an den Parent (TX Paket) vom Typ Simpel
public function SendData() {
$this->SendDataToParent(json_encode([
'DataID' => "{79827379-F36E-4ADA-8A95-5F8D1DC92FA9}",
'Buffer' => utf8_encode("Hallo Welt String"),
]));
}

// Empfangene Daten vom Parent (RX Paket) vom Typ Simpel
public function ReceiveData($JSONString) {
$data = json_decode($JSONString);
$data['Buffer'] = utf8_decode($Data['Buffer']);

//Im Meldungsfenster zu Debug zwecken ausgeben
IPS_LogMessage("DATA", print_r($data, true));
}
```

### ERWEITERT (EVENT)

RX GUID: {FD7FF32C-331E-4F6B-8BA8-F73982EF5AA7}
TX GUID: {4A550680-80C5-4465-971E-BBF83205A02B}

| PARAMETER | DATENTYP | BESCHREIBUNG |
|---|---|---|
| Buffer | String | Beliebiger Dateninhalt |
| EventID | Integer | ID für das Event |

```
// Beispiel für das Senden an den Parent (TX Paket) vom Typ
Erweitert (Event)
public function SendData() {
$this->SendDataToParent(json_encode([
'DataID' => "{4A550680-80C5-4465-971E-BBF83205A02B}",
'Buffer' => utf8_encode("Hallo Welt String"),
'EventID' => 123
]));
}

// Empfangene Daten vom Parent (RX Paket) vom Typ Erweitert (Event)
public function ReceiveData($JSONString) {
$data = json_decode($JSONString);
$data['Buffer'] = utf8_decode($Data['Buffer']);

//Im Meldungsfenster zu Debug zwecken ausgeben
IPS_LogMessage("DATA", print_r($data, true));
}
```

## ERWEITERT (SOCKET)

RX GUID: {7A1272A4-CBDB-46EF-BFC6-DCF4A53D2FC7}
TX GUID: {C8792760-65CF-4C53-B5C7-A30FCC84FEFE}

| PARAMETER | DATENTYP | BESCHREIBUNG |
|---|---|---|
| Buffer | String | Beliebiger Dateninhalt |
| Type | Integer | Typ der Verbindung (0 = Data, 1 = Connect, 2 = Disconnect) |
| ClientIP | String | IP-Adresse der Verbindung |
| ClientPort | Integer | Port der Verbindung |

```
// Beispiel für das Senden an den Parent (TX Paket) vom Typ
Erweitert (Socket)
public function SendData() {
$this->SendDataToParent(json_encode([
'DataID' => "{C8792760-65CF-4C53-B5C7-A30FCC84FEFE}",
'Buffer' => utf8_encode("Hallo Welt String"),
'Type' => 0,
'ClientIP' => "192.168.0.8",
'ClientPort' => 502
]));
}

// Empfangene Daten vom Parent (RX Paket) vom Typ Erweitert (Socket)
public function ReceiveData($JSONString) {
$data = json_decode($JSONString);
$data['Buffer'] = utf8_decode($Data['Buffer']);

//Im Meldungsfenster zu Debug zwecken ausgeben
IPS_LogMessage("DATA", print_r($data, true));
}
```

## ERWEITERT (UDP)

RX GUID: {9082C662-7864-D5CA-863F-53999200D897}
TX GUID: {8E4D9B23-E0F2-1E05-41D8-C21EA53B8706}

| PARAMETER | DATENTYP | BESCHREIBUNG |
|---|---|---|
| Buffer | String | Beliebiger Dateninhalt |
| ClientIP | String | IP-Adresse der Verbindung, wenn leer dann wird die im UDP Socket eingestellte Hostadresse genutzt. (Wird ignoriert, wenn Broadcast aktiv ist) |
| ClientPort | Integer | Port der Verbindung, wenn auf 0 gesetzt wird der im UDP Socket eingestellte Port genutzt |
| Broadcast | Boolean | Broadcast de-/aktiviert (False = Deaktiviert, True = Aktiviert) |

```
// Beispiel für das Senden an den Parent (TX Paket) vom Typ
Erweitert (UDP)
```

```
public function SendData() {
$this->SendDataToParent(json_encode([
'DataID' => "{8E4D9B23-E0F2-1E05-41D8-C21EA53B8706}",
'Buffer' => utf8_encode("Hallo Welt String"),
'ClientIP' => "192.168.0.8",
'ClientPort' => 502
'Broadcast' => false,
]));
}

// Empfangene Daten vom Parent (RX Paket) vom Typ Erweitert (UDP)
public function ReceiveData($JSONString) {
$data = json_decode($JSONString);
$data['Buffer'] = utf8_decode($Data['Buffer']);

//Im Meldungsfenster zu Debug zwecken ausgeben
IPS_LogMessage("DATA", print_r($data, true));
}
```

### ERWEITERT (SSE)

RX GUID: {5A709184-B602-D394-227F-207611A33BDF}

| PARAMETER | DATENTYP | BESCHREIBUNG |
|---|---|---|
| Event | String | Beliebiger Eventtyp |
| Data | String | Beliebiger Dateninhalt |
| Retry | String | In Millisekunden bis zu einem erneuten Sendeversuch |
| ID | String | ID für das Event |

TX GUID: {79827379-F36E-4ADA-8A95-5F8D1DC92FA9}
TX wird nicht ausgewertet muss aber gesetzt sein

```
// Empfangene Daten vom Parent (RX Paket) vom Typ Erweitert (SSE)
public function ReceiveData($JSONString) {
$data = json_decode($JSONString);

//Im Meldungsfenster zu Debug zwecken ausgeben
IPS_LogMessage("DATA", print_r($data, true));
}
```

Weitere Informationen gibt es unter w3.org/TR/eventsource

### ERWEITERT (HTTP REQUEST)

RX GUID: {018EF6B5-AB94-40C6-AA53-46943E824ACF}

| PARAMETER RX | DATENTYP | BESCHREIBUNG |
|---|---|---|
| Buffer | String | Beliebiger Dateninhalt |

TX GUID: {D4C1D08F-CD3B-494B-BE18-B36EF73B8F43}

| PARAMETER TX | DATENTYP | BESCHREIBUNG |
|---|---|---|
| RequestMethod | String | GET oder POST |
| RequestURL | String | URL der anzufragenden Webseite |
| RequestData | String | Welcher Datenwert angefragt werden soll |
| Timeout | Integer | Millisekunden bis ein Timeout kommt |

```
// Beispiel für das Senden an den Parent (TX Paket) vom Typ
Erweitert (HTTP Request)
public function SendData() {
$this->SendDataToParent(json_encode([
'DataID' => "{D4C1D08F-CD3B-494B-BE18-B36EF73B8F43}",
'RequestMethod' => utf8_encode("POST"),
'RequestURL' => utf8_encode("https://reqbin.com/echo/post/form"),
'RequestData' => utf8_encode("dummy-post-data"),
'Timeout' => 10000
]));
}

// Empfangene Daten vom Parent (RX Paket) vom Typ Simpel
public function ReceiveData($JSONString) {
$data = json_decode($JSONString);
$data['Buffer'] = utf8_decode($Data['Buffer']);

//Im Meldungsfenster zu Debug zwecken ausgeben
IPS_LogMessage("DATA", print_r($data, true));
}
```

## VORAUSSETZUNG UND EINRICHTUNG

Die GUIDs für die verwendeten Datenpakettypen werden in der module.json definiert. Damit ein Datenfluss zwischen 2 Instanzen etabliert werden kann, muss der Sendende bei dem passenden Requirement die genutzte Kommunikationsart eintragen.
Der Empfänger trägt seinerseits unter Implemented die GUID der Kommunikationsart ein. Wenn mehrere GUIDs verschiedener Datenpakettypen in die module.json eingetragen werden, kann ein Modul auch mehrere Datenpakettypen interpretieren und verarbeiten.



Für IPSModule: Ebenfalls muss von jedem Child der jeweilige Parent via ConnectParent, RequireParent oder ForceParent eingerichtet sein. Beim neuerem IPSModuleStrict werden die notwendigen Instanzen für den Datenfluss automatisch von der Verwaltungskonsole erstellt. Wenn diese Automatik nicht passend ist, kann das jeweilige Modul über die Funktion GetCompatibleParents die Automatik der Verwaltungskonsole beeinflussen.

## BEISPIEL

Kommunikation zwischen I/O, Splitter und Device.
Das Device hat den Splitter als Parent und der Splitter hat das I/O als Parent.
*Auszug aus den jeweiligen module.json*
Beispiel GUIDs der Datenpakettypen:
IO_TX: {65465465-6546-6546-6546-65465465}
IO_RX: {AE3AE3A-AE3A-AE3A-AE3A-AE3AE3AE}
Device_TX: {78978978-7897-7897-7897-78978978}
Device_RX: {12312312-1231-1231-1231-12312312}

```
// Innerhalb des I/Os (Parent vom Splitter)
// I/Os haben normalerweise keine weitere Parents. Somit sollte die
Liste normalerweise leer sein.
"parentRequirements": [],
// GUID des Datenpakettyps, welches via SendDataToChildren genutzt
wird
"childRequirements": ["{AE3AE3A-AE3A-AE3A-AE3A-AE3AE3AE}"],
// Liste GUIDs der Datenpakettypen auf die gehorcht wird.
// Diese wird von der Funktion ForwardData (Daten kommen vom Child)
verarbeitet.
"implemented": ["{65465465-6546-6546-6546-65465465}"],

// Innerhalb des Splitters (Parent vom Device, Child vom I/O)
// GUID des Datenpakettyps, welches via SendDataToParent genutzt
wird
"parentRequirements": ["{65465465-6546-6546-6546-65465465}"],
// GUID des Datenpakettyps, welches via SendDataToChildren genutzt
wird
"childRequirements": ["{12312312-1231-1231-1231-12312312}"],
// Liste GUIDs der Datenpakettypen auf die gehorcht wird.
// Diese wird in wird entweder von der Funktion ForwardData (Daten
kommen vom Child) oder ReceiveData (Daten kommen vom Parent)
verarbeitet.
"implemented": ["{AE3AE3A-AE3A-AE3A-AE3A-AE3AE3AE}",
"{78978978-7897-7897-7897-78978978}"],

// Innerhalb des Device (Child vom Splitter)
// GUID des Datenpakettyps, welches via SendDataToParent genutzt
wird
"parentRequirements": ["{78978978-7897-7897-7897-78978978}"],
// Devices haben normalerweise keine weitere Children. Somit sollte
die Liste normalerweise leer sein.
"childRequirements": [],
// Liste GUIDs der Datenpakettypen auf die gehorcht wird.
// Diese wird von der Funktion ReceiveData (Daten kommen vom Parent)
verarbeitet.
"implemented": ["{12312312-1231-1231-1231-12312312}"],
```

# Datenverwaltung

Benötigt: IP-Symcon >= 4.0

## BESCHREIBUNG

Ein Modul hat grundlegend vier verschiedene Formen Daten zu verwalten. Dies sind Eigenschaften, Attribute, Buffer und Statusvariablen. Diese Unterscheiden sich in Aufgabe, Zugriffsmöglichkeiten und Persistenz.
Eigenschaften
Attribute
Buffer
Statusvariablen

### EIGENSCHAFTEN

Eigenschaften sind persistente Daten eines Moduls, welche vom Benutzer konfiguriert werden sollen/müssen. Diese Daten werden erst bei "Übernehmen" gespeichert. Eigenschaften sind zum Beispiel benötigte Logindaten, Geräte-ID oder Intervall zur Abfrage von Sensorwerten. Dies geschieht im Regelfall über die Konfigurationsseite, welche über die form.json definiert wird.

| FUNKTIONSNAME | KURZBESCHREIBUNG |
|---|---|
| ReadPropertyBoolean | Liefert den Wert einer Boolean Eigenschaft |
| ReadPropertyFloat | Liefert den Wert einer Float Eigenschaft |
| ReadPropertyInteger | Liefert den Wert einer Integer Eigenschaft |
| ReadPropertyString | Liefert den Wert einer String Eigenschaft |
| RegisterPropertyBoolean | Erstellt eine Boolean Eigenschaft |
| RegisterPropertyFloat | Erstellt eine Float Eigenschaft |
| RegisterPropertyInteger | Erstellt eine Integer Eigenschaft |
| RegisterPropertyString | Erstellt eine String Eigenschaft |

```
public function Create() {
//Never delete this line!
parent::Create();

$this->RegisterPropertyBoolean("EmulateStatus", true);
$this->RegisterPropertyFloat("Faktor", 0.5);
$this->RegisterPropertyInteger("DeviceID", 0);
$this->RegisterPropertyString("Text", "");
}
```

### ATTRIBUTE

Attribute sind persistente Daten eines Moduls, welche einzig vom Modul selbst gesetzt und sofort gespeichert werden. Dies sind zum Beispiel Tokens bei verschlüsselten

Verbindungen oder gespeicherte Werte einer Szenensteuerung.

| FUNKTIONSNAME | KURZBESCHREIBUNG |
|---|---|
| ReadAttributeBoolean | Liefert den Wert eines Boolean Attributs |
| ReadAttributeFloat | Liefert den Wert eines Float Attributs |
| ReadAttributeInteger | Liefert den Wert eines Integer Attributs |
| ReadAttributeString | Liefert den Wert eines String Attributs |
| RegisterAttributeBoolean | Erstellt ein Boolean Attribut |
| RegisterAttributeFloat | Erstellt ein Float Attribut |
| RegisterAttributeInteger | Erstellt ein Integer Attribut |
| RegisterAttributeString | Erstellt ein String Attribut |
| WriteAttributeBoolean | Schreibt in ein Boolean Attribut |
| WriteAttributeFloat | Schreibt in ein Float Attribut |
| WriteAttributeInteger | Schreibt in ein Integer Attribut |
| WriteAttributeString | Schreibt in ein String Attribut |

Beispiel

```
public function Create() {
//Never delete this line!
parent::Create();

$this->RegisterAttributeBoolean("BoolAttr", true);
$this->RegisterAttributeInteger("IntAttr", 5);
$this->RegisterAttributeFloat("FloatAttr", 3.7);
$this->RegisterAttributeString("StrAttr", "lalala");
}

public function BumpAndShow() {

var_dump($this->ReadAttributeBoolean("BoolAttr"));
var_dump($this->ReadAttributeInteger("IntAttr"));
var_dump($this->ReadAttributeFloat("FloatAttr"));
var_dump($this->ReadAttributeString("StrAttr"));

$this->WriteAttributeBoolean("BoolAttr", !$this-
>ReadAttributeBoolean("BoolAttr"));
$this->WriteAttributeInteger("IntAttr", $this-
>ReadAttributeInteger("IntAttr")*2);
$this->WriteAttributeFloat("FloatAttr", $this-
>ReadAttributeFloat("FloatAttr")+0.1);
$this->WriteAttributeString("StrAttr", $this-
>ReadAttributeString("StrAttr") . "öäü");
}
```

### BUFFER

Buffer sind nicht persistente Daten eines Moduls, welche einzig vom Modul selbst verwaltet werden sollen. Dies sind zum Beispiel eingehende Datensätze, welche nicht in

einem übermittelt werden, sondern nach und nach eintreffen und somit zusammengesetzt werden müssen. Erst bei einem vollständigem Datensatz werden diese vom Buffer genommen und vom Modul verarbeitet.

| FUNKTIONSNAME | KURZBESCHREIBUNG |
|---|---|
| GetBuffer | Liefert den Inhalt eines Buffers |
| GetBufferList | Liefert ein Array aller Buffer |
| SetBuffer | Erstellt einen Buffer |

Beispiel

```
public function ReceiveData($JSONString) {
//Decode JSONString
$data = json_decode($JSONString);

//Parse and write values to our buffer
$this->SetBuffer("Test", utf8_decode($data->Buffer));

//Print buffer
IPS_LogMessage("IOTest", $this->GetBuffer("Test"));
}
```

## STATUSVARIABLEN

Statusvariablen sind persistente Daten eines Moduls, welche vom Modul jederzeit verändert werden können. Diese sind im Objektbaum sichtbar und stehen zur weiteren Verarbeitung und Darstellung in den Visualisierungen zur Verfügung. Statusvariablen sind zum Beispiel Sensorwerte, im WebFront darzustellende Aktorenwerte und Statuswerte zur Weiterverarbeitung.

| FUNKTIONSNAME | KURZBESCHREIBUNG |
|---|---|
| DisableAction | Deaktiviert die Standardaktion |
| EnableAction | Aktiviert die Standardaktion |
| GetValue | Liefert den Wert einer Variable |
| SetValue | Setzt den Wert einer Variable |
| MaintainAction | Ruft DisableAction oder EnableAction auf |
| MaintainVariable | Konfiguriert eine Statusvariable |
| RegisterVariableBoolean | Erstellt eine Boolean Statusvariable |
| RegisterVariableFloat | Erstellt eine Float Statusvariable |
| RegisterVariableInteger | Erstellt eine Integer Statusvariable |
| RegisterVariableString | Erstellt eine String Statusvariable |
| RequestAction | Setzt den Wert einer Statusvariable |
| UnregisterVariable | Löscht eine Statusvariable |

Beispiel

```
public function Create() {
```

```
//Never delete this line!
parent::Create();

//Variables
$this->RegisterVariableString("TextData", "TextData", "");
IPS_SetHidden($this->GetIDForIdent("TextData"), true);
$this->RegisterVariableString("SimulationView",
"Simulationsvorschau", "~HTMLBox");
$this->RegisterVariableInteger("SimulationCounter",
"Simulationszähler" , "");
$this->RegisterVariableFloat("Faktor", "Zoom Faktor Wandanzeige",
"Faktor.Anzeige");
$this->RegisterVariableBoolean("Active", "Simulation aktiv",
"~Switch");
$this->EnableAction("Active");
}
```

# HTML-SDK

Benötigt: IP-Symcon >= 7.1

## BESCHREIBUNG

Das HTML-SDK ermöglicht es PHP-Modulen eine individuelle Objektdarstellung zu nutzen. Diese Darstellung ermöglicht dank HTML volle Flexibilität. Durch Nachrichten zwischen der HTML-Darstellung und dem Modul kann die Darstellung zur Laufzeit aktualisiert werden oder das Modul über Interaktion durch den Benutzer informieren.

## HTML ALS DARSTELLUNG NUTZEN

Soll das HTML-SDK verwendet werden, so muss die Darstellung mit der Funktion SetVisualizationType aktiviert werden. Soll dann die Kachel in der Visualisierung dargestellt werden, wird die Funktion GetVisualizationTile verwendet um den HTML-Inhalt zurückzugeben.

## ÜBERSETZUNG

Analog zu Konfigurationsformularen werden auch die Texte der HTML-Elemente der Darstellung basierend auf der Sprache des Benutzers lokalisiert.

### ICONS

Die Symcon Icons können innerhalb der HTML Darstellung verwendet werden, indem die icons.js als Skript eingebunden wird:

```
<script src="/icons.js"></script>
```

Wenn das Skript eingebunden ist, können Icons entsprechend der Dokumentation von Font Awesome angezeigt werden. Es sind alle regulär in Symcon verwendbaren Icons nutzbar, also Classic Icons mit dem Stil Light per CSS Klasse fa-light sowie Brands über die Klasse fa-brands. Die Symcon-spezifischen Icons können mit der Klasse fa-kit genutzt werden.

## NACHRICHTEN

Die Funktion GetVisualizationTile wird initial einmalig aufgerufen. Soll die Darstellung zur Laufzeit angepasst werden, so muss dies über Nachrichten geschehen. Alternativ ermöglichen Nachrichten einen Kanal von der Darstellung zum Modul um Variablen zu schalten oder andere Interaktion durchzuführen.

Es ist zwar möglich eine vergleichbare Kommunikation auch über eine HTML-Box zu realisieren. Dies ist allerdings zum einen recht aufwendig und birgt zum anderen das Risiko Sicherheitslücken zu erstellen, welche ausgenutzt werden könnten. Die Kommunikation via HTML-SDK ist im Gegenzug in beide Richtungen mit dem Passwort der Visualisierung abgesichert.

### MODUL ZU DARSTELLUNG

Soll eine Nachricht vom Modul zur Darstellung geschickt werden, so muss im Modul die Funktion UpdateVisualizationValue verwendet werden. Um den Inhalt dieser Nachricht in der Darstellung zu empfangen und zu verarbeiten, muss via JavaScript die Funktion handleMessage implementiert werden. Diese erhält als Eingabe genau den Inhalt, der mit UpdateVisualizationTile abgeschickt wurde. Das Format dieser Daten ist absolut frei und sollte passend zum Modul gewählt werden.

### DARSTELLUNG ZU MODUL

Die Darstellung kann via JavaScript die Funktion requestAction verwenden. Diese Funktion führt im Gegenzug auf Seite des Moduls die PHP-Funktion RequestAction aus, wodurch das Modul auf die Nachricht reagieren kann.

## SPEZIELLE FUNKTIONEN IN JAVASCRIPT

Das HTML-SDK bietet eine Reihe von Funktionen, welche innerhalb der Darstellung via JavaScript verwendet werden können.

| FUNKTION | BESCHREIBUNG |
|---|---|
| handleMessage | Diese Funktion muss definiert werden um Nachrichten vom Modul zu empfangen |
| requestAction | Diese Funktion schickt eine Nachricht ans Modul |
| translate | Übersetzt Text basierend auf der Lokalisierung |
| translateHTML | Übersetzt den Text von HTML-Elementen basierend auf der Lokalisierung |

## WEBINAR UND BEISPIELE

Bei der Veröffentlichung wurde die Funktionalität des HTML-SDK im Rahmen eines Webinars auf YouTube vorgestellt.
In unserem Test-Repository können auch einige Beispielimplementation eingesehen und ausprobiert werden:

| BEISPIEL | BESCHREIBUNG |
|---|---|
| Entenzähler | Ein kleiner Zähler mit ausführlichen Kommentaren |
| Wärmepumpe | Ein komplexes und umfangreiches Beispiel zur Darstellung einer Wärmepumpe |

# handleMessage

Benötigt: IP-Symcon >= 7.1

```
void handleMessage (variant Daten)
```

## PARAMETERLISTE

| | |
|---|---|
| Daten | beliebige Daten |

## RÜCKGABEWERT

Keine Rückgabe

## BESCHREIBUNG

Diese Funktion muss implementiert werden, falls die Darstellung Nachrichten des Moduls empfangen und verarbeiten soll. Wird im Modul die Funktion UpdateVisualizationValue verwendet, wird dadurch im Gegenzug diese Funktion mit dem gleichen Parameterwert aufgerufen.

## BEISPIEL

```
function handleMessage(data) {
// In diesem Beispiel beinhaltet data einfach einen Text, welcher
möglicherweise übersetzt und angezeigt werden soll
document.getElementById('info').textContent = translate(data);
}
```

# requestAction

Benötigt: IP-Symcon >= 7.1

```
void requestAction (string Ident, variant Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| Ident | Ein Ident |
| Wert | Ein Wert |

## RÜCKGABEWERT

Keine Rückgabe

## BESCHREIBUNG

Diese Funktion schickt eine Nachricht an das Modul und führt dort die Funktion RequestAction mit den gegebenen Parameter aus.



Auch wenn RequestAction üblicherweise zum Schalten einer Variable mit dem gegebenen Ident verwendet wird, ist die Funktion nicht darauf beschränkt. So kann bei Verwendung eines Idents, der zu keiner Variablen passt, beliebig anderes Verhalten ausgelöst werden.

## BEISPIEL

```
// Schalte Zähler auf 5
requestAction('Counter', 5);
```

# translate

Benötigt: IP-Symcon >= 7.1

```
string translate (string Original)
```

## PARAMETERLISTE

| | |
|---|---|
| Original | Der zu übersetzende englische Originaltext |

## RÜCKGABEWERT

Der übersetzte Text

## BESCHREIBUNG

Übersetzt den Text *original* basierend auf der Lokalisierung des Moduls und der Sprache des Benutzers. Existiert keine passende Übersetzung, wird der Originaltext wieder zurückgegeben.

## BEISPIEL

```
// Gebe den übersetzten Text in der Konsole aus
console.log(translate('Hallo Welt'));
```

# translateHTML

Benötigt: IP-Symcon >= 7.1

```
void translateHTML (object Knoten)
```

## PARAMETERLISTE

| | |
|---|---|
| Knoten | Ein HTML-Element |

## RÜCKGABEWERT

Keine Rückgabe

## BESCHREIBUNG

Übersetzt und überschreibt den Textinhalt des HTML-Elements *knoten* und all seinen Kindern.



Beim initialen Laden der Visualisierung wird diese Funktion einmal automatisch auf das gesamte Dokument angewandt. Daher ist eine Verwendung eigentlich nur notwendig, wenn Elemente dynamisch angepasst werden.

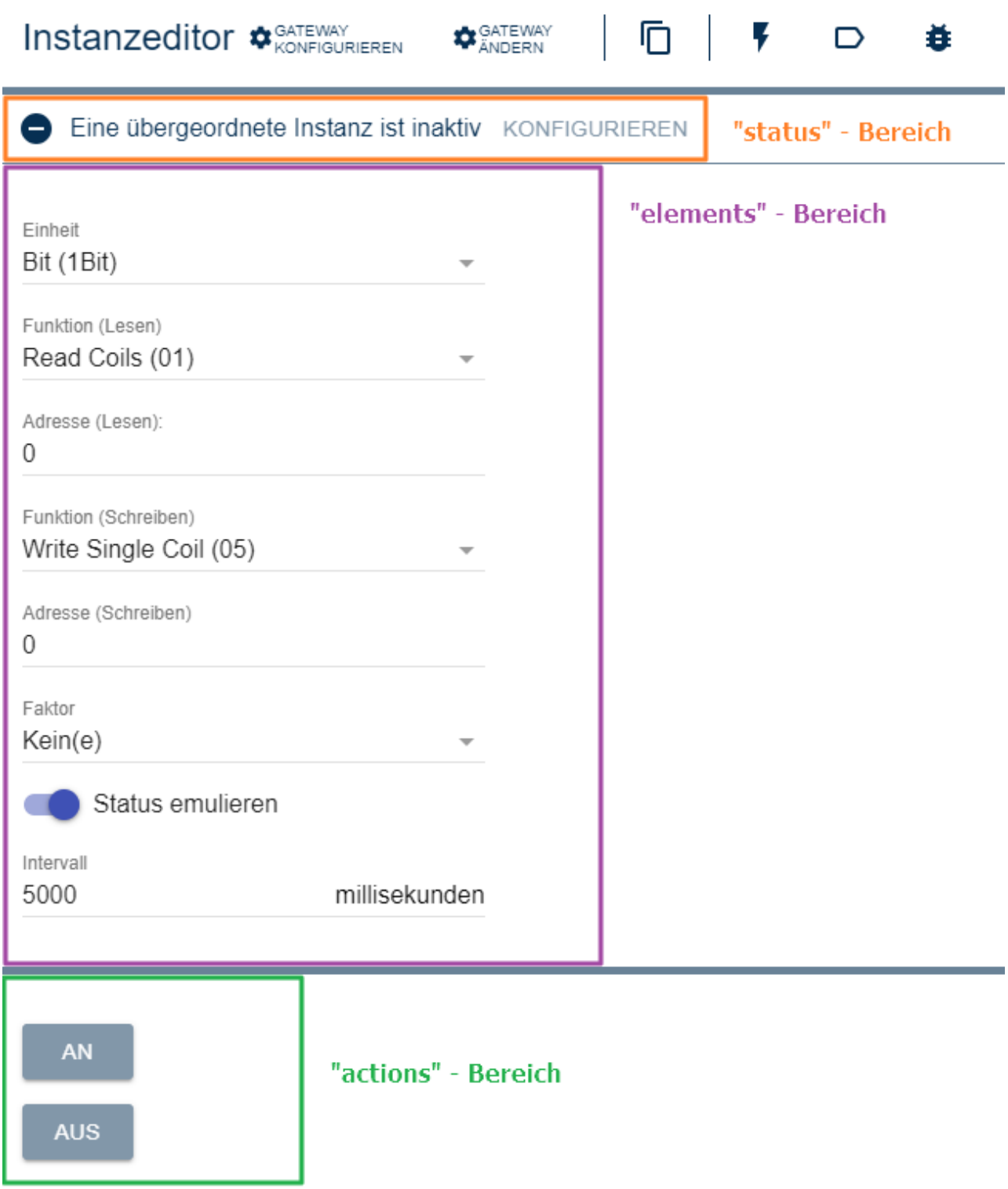
## BEISPIEL

```
// Übersetze Inhalt des Textblockes 'info'
translateHTML(document.getElementById('info'));
```

# Konfigurationsformulare

Benötigt: IP-Symcon >= 4.0
Die Konfigurationsformulare sind JSON Dateien, welche zur Konfiguration der Eigenschaften von IP-Symcon Instanzen genutzt werden.
Diese werden normalerweise von der Verwaltungskonsole geladen und entsprechend angezeigt.
Die JSON-Datei, welche das Konfigurationsformular beschreibt, besitzt bis zu zwei Bereiche die unabhängig voneinander individuell beschrieben werden können.
Eine Übersetzung der "caption" und "label" kann über die Lokalisierungen via der locale.json hinzugefügt werden.

## FORMULARBEREICHE




**elements**
Im "elements"-Bereich definierte Formularfelder können die Eigenschaften einer Instanz setzen. Der Name eines Formularfeldes, sofern verfügbar, entspricht dem Namen der Eigenschaft, welche über IPS_SetProperty gesetzt wird. Soll der "elements" Bereich nicht sichtbar sein, so ist dieser nicht zu definieren.

**actions**
Der "actions"-Bereich ist eine sogenannte Testumgebung. Die hier definierten Formularfelder haben nicht die Möglichkeit Eigenschaften der jeweiligen Instanz zu verändern. Soll der "actions" Bereich nicht sichtbar sein, so ist dieser nicht zu definieren. Generell gilt: Die Testumgebung kann nur genutzt werden, wenn Veränderungen im Elemente-Bereich übernommen wurden. Dadurch wird verhindert, das nicht übernommene Einstellungen getestet werden.

**status**

Der "status"-Bereich definiert mögliche Statusmeldungen. Im "status"-Bereich sind keine Formularfelder erlaubt, sondern nur Statusmeldungen, welche unter Statusmeldung erklärt werden. Soll das Modul keine Statusmeldungen anzeigen, so ist dieser Bereich nicht zu definieren.

## ÜBERSICHT DER EINFÜGBAREN FORMULARFELDER

| FORMULARFELD / TYPE | AB IP-SYMCON | BESCHREIBUNG |
|---|---|---|
| Button | 4.0 | Button |
| CheckBox | 4.0 | Checkbox |
| Configurator | 5.0 | Beschreibt einen Konfigurator |
| ExpansionPanel | 5.0 | Auf- und zuklappbare Formularfelder |
| HorizontalSlider | 4.0 | Ein horizontaler Schieberegler |
| Image | 5.0 | Erstellt ein Bild auf der Modulseite |
| IntervalBox | 4.0 | Angabe von Zahlen mit Textbeschreibung/Einheit (veraltet) |
| Label | 4.0 | Eine Beschriftung |
| List | 4.2 | Eine editierbare Liste |
| NumberSpinner | 4.0 | Eingabefeld, bei dem nur Zahlen erlaubt sind. (optional mit Nachkommastellen) |
| PasswordTextBox | 4.0 | Eingabefeld für Passwörter |
| PopupAlert | 5.0 | Alarm, welcher ein Popup sofort öffnet |
| PopupButton | 5.0 | Button, welcher ein Popup öffnet |
| RowLayout | 5.0 | Zeilenansicht von mehreren Formularfeldern |
| Select | 4.0 | DropDown-Menü |
| SelectCategory | 4.0 | Auswahldialog für eine Kategorie |
| SelectColor | 4.2 | Auswahldialog für eine Farbe |
| SelectDate | 5.0 | Auswahldialog für ein Datum |
| SelectDateTime | 5.0 | Auswahldialog für ein Datum und Uhrzeit |
| SelectEvent | 4.0 | Auswahldialog für ein Ereignis |
| SelectFile | 4.2 | Auswahldialog für eine Datei |
| SelectInstance | 4.0 | Auswahldialog für eine Instanz |
| SelectLink | 4.0 | Auswahldialog für einen Link |
| SelectMedia | 4.0 | Auswahldialog für eine Media-Datei |
| SelectObject | 4.0 | Auswahldialog für ein Objekt |
| SelectScript | 4.0 | Auswahldialog für ein Skript |
| SelectTime | 5.0 | Auswahldialog für eine Uhrzeit |
| SelectVariable | 4.0 | Auswahldialog für eine Variable |
| Tree | 5.0 | Erstellt einen Baum |
| ValidationTextBox | 4.0 | Eingabefeld für Text |

## BEISPIELE

Grundaufbau der JSON Datei

```
{
"elements":
[
... //Hier Formularfelder einfügen
],
"actions":
[
... //Hier Formularfelder einfügen
],
"status":
[
... //Hier Statusmeldungen einfügen
]
}
```

Vollständiges Beispiel anhand eines SMS Moduls

```
{
"elements":
[
{ "type": "ValidationTextBox", "name": "Username", "caption":
"Username" },
{ "type": "ValidationTextBox", "name": "Password", "caption":
"Password" },
{ "type": "NumberSpinner", "name": "APIID", "caption": "APIID" },
{ "type": "ValidationTextBox", "name": "Sender", "caption": "Sender"
},
{ "type": "Select", "name": "SMSType", "caption": "Type",
"options": [
{ "label": "SMS", "value": 0 },
{ "label": "Combi-SMS", "value": 1 },
{ "label": "MMS", "value": 2 }
]}
],
"actions":
[
{ "type": "ValidationTextBox", "name": "Number", "caption": "Number"
},
{ "type": "ValidationTextBox", "name": "Message", "caption":
"Message" },
{ "type": "Button", "label": "Send Message", "onClick":
"SMS_Send($id, $Number, $Message);" },
{ "type": "Button", "label": "Read Balance", "onClick": "echo
'Balance: '.sprintf('%.2f', SMS_RequestBalance($id)).' credits';" }
],
"status":
[
{ "code": 102, "icon": "active", "caption": "Login information
valid" },
{ "code": 201, "icon": "error", "caption": "Authentication failed"
```

```
},
{ "code": 202, "icon": "error", "caption": "No credits left" }
]
}
```

# Button

Ein Button der bei Betätigung eine Aktion ausführt.




## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung des Buttons |
| confirm (optional) | (**default:** "") Beinhaltet dieser Parameter keinen leeren String, so wird bei Betätigung des Buttons der Text in einem Ja/Nein-Dialog angezeigt. Die onClick-Aktion wird nur ausgeführt, falls der Dialog mit "Ja" bestätigt wird. Ist der Parameter leer, so wird die Aktion direkt ausgeführt. (ab IP-Symcon 5.0) |
| download (optional) | (**default:** "") Ist dieser Parameter nicht leer und beinhaltet die Ausgabe des onClick-Skriptes eine Data-URL, so wird die Rückgabe als Textdatei mit dem Namen das download-Parameters heruntergeladen. (ab IP-Symcon 7.0) |
| enabled (optional) | (**default:** true) Bei true kann der Button verwendet werden, ansonsten wird er deaktiviert dargestellt (ab IP-Symcon 5.2) |
| ~~label~~ | ~~Name und sichtbare Beschriftung des Buttons~~ *(veraltet, nicht änderbar)* |
| link (optional) | Ist dieser Parameter true, dann wird die Ausgabe des onClick-Skriptes als Link geöffnet. Ist er false, so wird die Ausgabe als Dialog im Konfigurationsformular angezeigt. Ist der Parameter nicht gesetzt, dann wird auf Basis der Ausgabe entschieden, ob diese als Link geöffnet oder im Dialog angezeigt wird (siehe Infobox). (ab IP-Symcon 5.1) |
| name (optional) | Name des Buttons |
| onClick | Skript, welches bei einem Klick auf den Button ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden (Arrays werden ab Version 6.0 unterstützt). Es sind keine PHP-Tags erforderlich. Es stehen einige weitere Variablen zur Verfügung, welche im Infofeld weiter unten erörtert werden. |
| type | Button |
| visible (optional) | (**default:** true) Bei true ist der Button sichtbar, ansonsten ist er unsichtbar (ab IP-Symcon 5.2) |
| width (optional) | Feste Breite des Buttons in Pixel oder % als String, z.B. "40%" oder "250px". Ist der Wert nicht gesetzt oder "", wird die Breite automatisch auf Basis der Beschriftung gewählt (ab IP-Symcon 5.2) |

Die Variable $id steht in allen Bereichen zur Verfügung. Zusätzlich werden für alle Formularfelder Variablen erstellt, sofern ein Name vergeben wurde. Die Variable ist dann entsprechend als $Name nutzbar. Die Variable $Name ist nur innerhalb des selben Scope (z.B. actions oder elements) verfügbar.

Steht in dem Parameter onClick ein "echo" in Kombination mit 'http'/'https' wird der Browser geöffnet und die URL aufgerufen. Ist ein "echo" in Kombination mit einem 'mailto:' vorhanden wird das Standard-Mailprogramm mit allen gesetzten Parametern geöffnet. Ist ein "echo" in Kombination mit einem 'link:' vorhanden wird 'link:' herausgefiltert und der verbleibende Inhalt wird im Browser geöffnet. Dadurch sind etwa relative Links möglich. (Verfügbar ab IP-Symcon 4.1)

## BEISPIEL

```
//Das Beispiel gibt die InstanzID zurück.
{ "type": "Button", "caption": "An", "onClick": "echo $id;" }

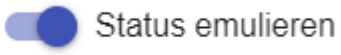
//Sonderbeispiel:
//Das Beispiel öffnet bei "Klick" den Browser mit der angegebenen
Adresse: 'https://www.symcon.de'
{
"type": "Button",
"caption": "Symcon",
"onClick": "echo 'https://www.symcon.de';"
}

//Das Beispiel fragt, ob der Wert einer Variablen tatsächlich auf 0
gesetzt werden soll
{
"type": "Button",
"caption": "Reset",
"onClick": "SetValue(12345, 0);",
"confirm": "Dies setzt die Variable zurück auf 0. Dabei können
wichtige Daten verloren gehen. Sind Sie sicher?"
}

//Das Beispiel führt ein mehrzeiliges Skript aus
{
"type": "Button",
"caption": "Reset",
"onClick": [
"SetValue(12345, 0);",
"echo 'Wert wurde zurückgesetzt';"
]
}
```

# CheckBox

Erstellt eine Checkbox.
Falls im "elements"-Bereich erstellt, setzt die Checkbox eine Eigenschaft bei Übernahme auf **True** oder **False**.
Der Parameter **"name"** definiert dabei welche Eigenschaft gesetzt wird.




## PARAMETER

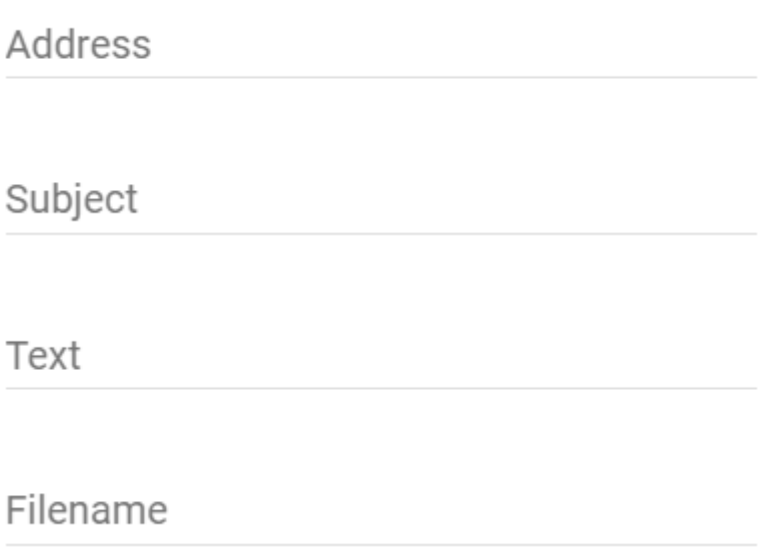
| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung |
| download (optional) | (**default:** "") Ist dieser Parameter nicht leer und beinhaltet die Ausgabe des onChange-Skriptes eine Data-URL, so wird die Rückgabe als Textdatei mit dem Namen das download-Parameters heruntergeladen. (ab IP-Symcon 7.0) |
| enabled (optional) | (**default:** true) Bei true kann die CheckBox verwendet werden, ansonsten wird sie deaktiviert dargestellt (ab IP-Symcon 5.2) |
| link (optional) | (**default:** false) Ist dieser Parameter true, dann wird die Ausgabe des onChange-Skriptes als Link geöffnet. Ist er false, so wird die Ausgabe als Dialog im Konfigurationsformular angezeigt. (ab IP-Symcon 6.0) |
| name (optional) | Name der Checkbox/der zu setzenden Eigenschaft |
| onChange (optional) | (**default:** "") Skript, welches bei einer Veränderung der Checkbox durch einen Klick ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden (Arrays werden ab Version 6.0 unterstützt). Es hat die gleichen Eigenschaften wie onClick des Button (ab IP-Symcon 5.2) |
| type | CheckBox |
| value (optional) | (**default:** false) Der Wert der CheckBox - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben (ab IP-Symcon 5.2) |
| visible (optional) | (**default:** true) Bei true ist die CheckBox sichtbar, ansonsten ist sie unsichtbar (ab IP-Symcon 5.2) |
| width (optional) | Feste Breite der Checkbox in Pixel oder % als String, z.B. "40%" oder "250px". Ist der Wert nicht gesetzt oder "", wird die Breite automatisch auf Basis der Beschriftung gewählt (ab IP-Symcon 5.3) |

## BEISPIEL

```
{ "type": "CheckBox", "name": "status", "caption": "Status
emulieren" }
```

# ColumnLayout

Benötigt: IP-Symcon >= 6.0
Erstellt eine Spalte, in welcher Formularfelder untereinander dargestellt werden. Dies ist insbesondere hilfreich um den Inhalt dynamisch anzupassen.




## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| items | Liste mit Konfigurationselementen innerhalb der Spalte |
| name (optional) | Name der Spalte |
| type | ColumnLayout |
| visible (optional) | (**default:** true) Bei true ist die Spalte sichtbar, ansonsten ist sie unsichtbar |

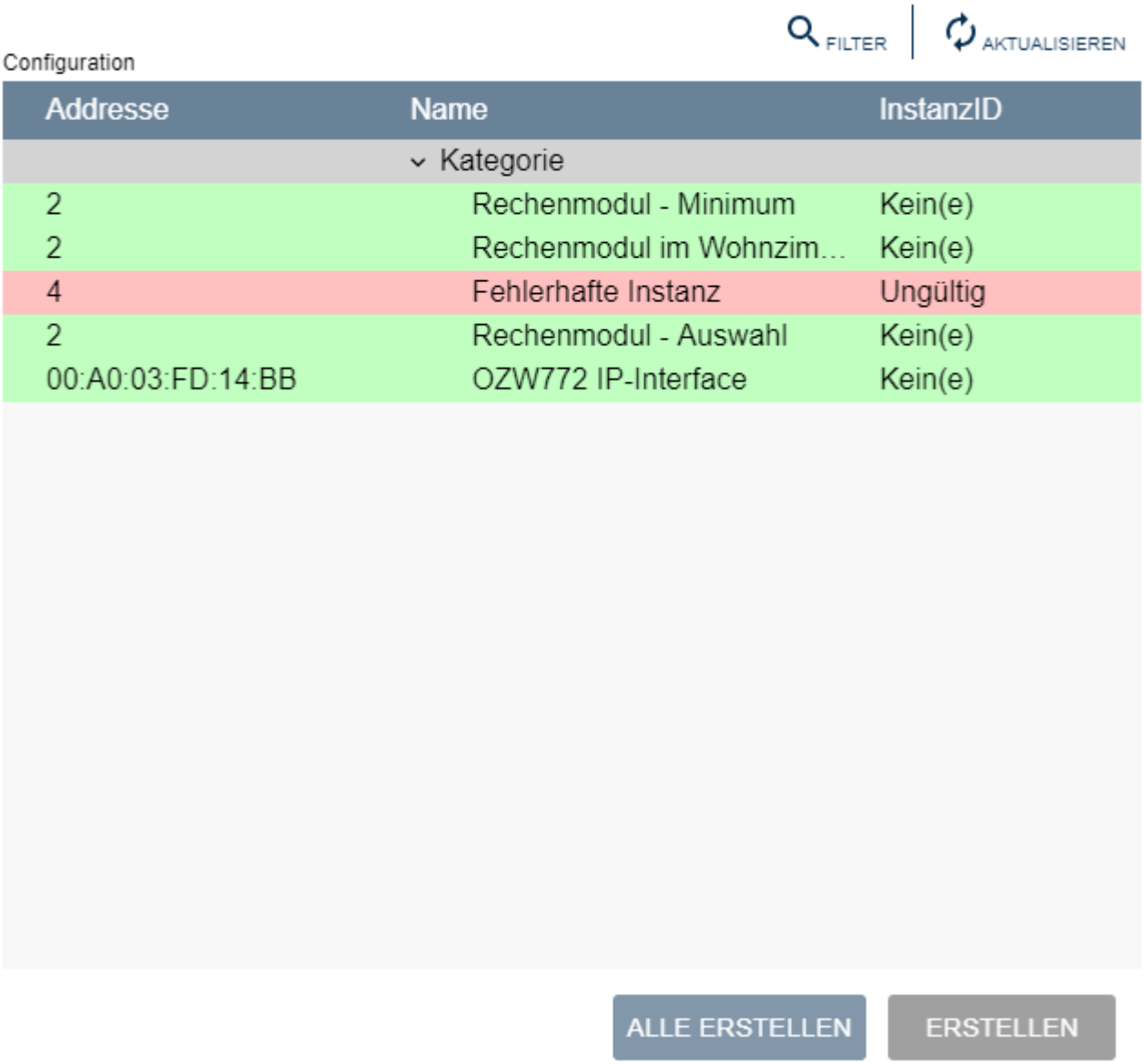
## BEISPIEL

```
// Spaltenlayout
{
"type": "ColumnLayout",
"items": [
{
"type": "SelectLocation",
"name": "Location",
"caption": "My Location"
},{
"type": "Button",
"caption": "Test",
"onClick": "TM_Update($id);"
},{
"type": "Button",
"caption": "Test2",
```

```
"onClick": "TM_Update2($id);"
}
]
}
```

# Configurator

Benötigt: IP-Symcon >= 5.0
Beschreibt das Aussehen und Funktionalität eines Konfigurators.




## ALLGEMEINE PARAMETER

| PARAMETER | BESCHREIBUNG |
| --- | --- |
| caption (optional) | Sichtbare Beschriftung der Konfigurationsliste |
| columns (optional) | (**default:** Spalten für Adresse ("address") und Name ("name") ) Spalten der Konfigurationsliste |
| discoveryInterval (optional) | (**default:** 0) Macht die Konsole für die Gerätesuche eine Anfrage an das beinhaltende Konfigurationsformular, gibt dieser Parameter an, wie viele Sekunden vergangen sein müssen, bevor eine erneute Anfrage gestartet werden darf; Befinden sich innerhalb eines Konfigurationsformulares mehrere Elemente vom Typ Configurator, so wird der maximale Wert verwendet (ab IP-Symcon 6.3) |
| delete (optional) | (**default:** false) Wenn true, dann wird hinter jedem Eintrag mit InstanzID ein Button eingeblendet, welcher die entsprechende Instanz löscht und die Zeile im Konfigurator aktualisiert |
| name (optional) | Name der Konfigurationsliste |
| onExpand (optional) | (**default:** "") Skript, welches nach der Ausklappen eines Eintrages des Konfigurators ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden. In dieser Ausführung bietet der Baum als Variablenwert nicht den aktuell ausgewählten Eintrag sondern den ausgeklappten Eintrag, siehe Ausgewählte Zeile als Variable in on-Aktionen. Ansonsten hat es die gleichen Eigenschaften wie onClick des Button (ab IP-Symcon 6.4) |
| rowCount (optional) | (**default:** 0) sichtbare Anzahl von Zeilen; bei mehr Zeilen in der Liste wird ein Scrollbalken eingeblendet - ist der Wert 0, so füllt die Konfigurationsliste den restlichen noch verfügbaren Platz im Instanzeditor. Nur ein Konfigurationselement kann den gesamten restlichen Platz erhalten. |
| sort (optional) | Sortierung der Tabelle. Wird dies nicht gesetzt bleibt die Liste unsortiert |
| type | Configurator |
| values (optional) | Einträge der Form |
| visible (optional) | (**default:** true) Bei true ist der Configurator sichtbar, ansonsten ist er unsichtbar (ab IP-Symcon 5.2) |



Sollen die Parameter columns, sort oder values per UpdateFormField verändert werden, so müssen die neuen Werte als JSON codiert werden.

## PARAMETER FÜR COLUMNS

| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung einer Spalte |
| ~~label~~ | ~~Sichtbare Beschriftung einer Spalte~~ *(veraltet)* |
| name | eindeutiger Name einer Spalte um diese zu referenzieren; der Name darf nicht "id", "parent", "create", "expanded" oder "instanceID" sein |
| quickFilter (optional) | (**default:** true) Bei true wird der Text dieser Spalte beim Schnellfilter in der Suche betrachtet. Sollte der Wert bei allen Spalten auf false gesetzt sein, so wird der Schnellfilter ausgeblendet. (ab IP-Symcon 7.2) |
| visible (optional) | (**default:** true) Bei false wird die Spalte nicht angezeigt |
| width | Breite der Spalte in Pixeln, als CSS-String (z.B. "100px"); eine einzige Spalte darf den Wert "auto" haben, wodurch die Breite dieser Spalte den restlich verfügbaren Platz nutzt. Im Falle eines Baumes muss bei einer Spalte der Wert "auto" gesetzt sein. An dieser Spalte werden die Erweiterungs-Symbole dargestellt. |

## PARAMETER FÜR SORT

| PARAMETER | BESCHREIBUNG |
|---|---|
| column | Name der Spalte, nach welcher die Liste sortiert wird |
| direction | Die Richtung der Sortierung; "ascending" für eine aufsteigende Sortierung und "descending" für eine absteigende Sortierung |

## PARAMETER FÜR VALUES

| PARAMETER | BESCHREIBUNG |
|---|---|
| address (optional) | (**default**: "") Wurden die Standardspalten verwendet, so kann mit diesem Pvarameter der Wert für die Adresse gesetzt werden |
| create (optional) | (**default**: error) Vorgabe der Konfiguration der erstellten Instanz (Mehr Informationen) |
| expanded (optional) | (**default**: false) Falls dieser Wert true ist, wird der Eintrag beim Laden des Konfigurationsformulares direkt ausgeklappt |
| instanceID (optional) | (**default**: 0) InstanzID des Gerätes im Objektbaum, 0 wenn das Gerät noch nicht erstellt wurde |
| id (optional) | Ein Identifikator, mit welchem der Eintrag für eine Baumdarstellung als Elternknoten adressiert werden kann, id muss größer als 0 sein |
| name (optional) | (**default**: "") Wurden die Standardspalten verwendet, so kann mit diesem Parameter der Wert für den Namen gesetzt werden |
| parent (optional) | (**default**: 0) In diesem Feld kann die id eines Wertes gesetzt werden, welcher in der Baumdarstellung der Elternknoten von diesem Wert wird. Hiefür muss der Eltern-Wert im Array vor dem Kind stehen. Ist "parent" 0, so wird das Element auf oberster Ebene eingefügt. |

## PARAMETER FÜR CREATE

Es gibt verschiedene Möglichkeiten die zu erstellenden Instanzen und deren initiale Konfiguration zu definieren.

### VARIANTE 1: KONFIGURATION EINER EINZELNEN INSTANZ

In der ersten Variante wird eine einzelne Instanz und deren Konfiguration definiert. In diesem Fall ist create ein Objekt, welches Informationen über die verwendete ModulID und die initiale Konfiguration enthält:
*Parameter für Einrichtung*

| PARAMETER | BESCHREIBUNG |
|---|---|
| moduleID | Die Modul ID der zu erstellenden bzw. benötigten Instanz |
| configuration | Die benötigte Konfiguration der Instanz, siehe IPS_SetConfiguration, nicht als JSON-String codiert |
| location (optional) | (**default:** []) Die Liste von Strings gibt die initiale Position im Objektbaum an. Bei einer leeren Liste wird die Instanz in der Hauptkategorie erstellt. Ansonten werden hierarchisch die Kategorien mit dem jeweiligen Namen verwendet und bei Bedarf erstellt. |
| info (optional) | (**default:** "") Die initiale Beschreibung der neu erstellten Instanz; wird eine bereits existierende Instanz verwendet, wird die Beschreibung nicht überschrieben. |
| position (optional) | (**default:** 0) Die initiale Position der neu erstellten Instanz; wird eine bereits existierende Instanz verwendet, wird die Position nicht überschrieben. |
| name (optional) | (**default:** Modulname oder Wert der Spalte "name") Der initiale Name der neu erstellten Instanz; wird eine bereits existierende Instanz verwendet, wird der Name nicht überschrieben. (ab IP-Symcon 5.5) |
| noDiscovery (optional) | (**default:** false) Steht dieser Parameter auf true, so wird dieser Eintrag bei der Gerätesuche nicht als Gerät angezeigt und kann auch nicht als gelesen markiert werden. Bei einer Konfiguration als Kette ( Variante 2 ) wird der Wert des ersten Eintrags der Kette verwendet. Bei mehreren möglichen Konfigurationen (Variante 3 ) muss jede Konfiguration diesen Parameter auf true haben, damit die Auswahl nicht in der Gerätesuche auftaucht. (ab IP-Symcon 5.4) |
| statusVariables (optional) | (**default:** {}) Dies Objekt gibt eine initiale Konfiguration der Statusvariablen vor. Für jeden Parameter dieses Statusvariablen-Objektes wird die Variable mit dem Ident des Parameternamens konfiguriert. Der dazugehörige Parameter ist ein weiteres Objekt, welches die Konfiguration der Variablen definiert. Die Variablenkonfiguration wird nach der Übernahme der Instanzkonfiguration angewandt. Variablen, die erst später erstellt werden, können hier nicht konfiguriert werden. (ab IP-Symcon 5.3) |

*Parameter für Statusvariablen*

| PARAMETER | BESCHREIBUNG |
|---|---|
| name (optional) | (**default:** Vorgabe durch erstellte Instanz) Die Statusvariable wird auf diesen Wert umbenannt. Der Name wird durch die Lokalisierung des Konfigurators übersetzt. |

Wird eine Instanz erstellt, so wird sie mit der von der Einrichtung vorgegebenen ModulID und Konfiguration erstellt. Wie bei der Instanzerstellung üblich, wird die neue Instanz mit validen, physikalisch übergeordneten Instanzen verknüpft, wenn möglich. Hierbei wird bevorzugt die dem Konfigurator physikalisch übergeordnete Instanz verwendet.

### VARIANTE 2: KONFIGURATION EINER KETTE

Alternativ kann eine Kette von Konfigurationen übergeben werden. Diese definiert die gesamte physikalische Kette für die neu erstellte Instanz, üblicherweise also Gerät, Splitter und I/O. In dieser Variante stellt data.create eine Liste dar, deren Elemente wie in Variante 1 die Instanzen der physikalischen Kette definiert. Die Liste beginnt hierbei mit dem zu erstellenden Gerät und setzt dann mit der jeweils nächsten physikalisch übergeordneten Instanz fort.
Bei der Erstellung durch eine Kette, werden bevorzugt existierende übergeordnete Instanzen wiederverwendet, sofern diese und deren physikalisch übergeordneten Instanzen die Anforderungen der Kette erfüllen. Das Gerät selbst wird allerdings in jedem Falle neu erstellt.

### VARIANTE 3: MEHRERE MÖGLICHE KONFIGURATIONEN

Als letzte Möglichkeit können mehrere Konfigurationen zur Auswahl gestellt werden. Hierbei wird create als Objekt mit Schlüssel/Wert-Paaren erstellt, wobei jeder Wert wieder einer Konfiguration wie in Variante 1 oder Variante 2 beschrieben entspricht. Soll bei dieser Darstellung eine Instanz erstellt werden, so wird dem Benutzer eine Auswahl der Bezeichnungen vorgegeben. Die Instanz wird mit den Einstellungen des ausgewählten Elements erstellt. Hierbei können die Schlüssel "moduleID" und "configuration" nicht verwendet werden.

## FARBEN IN DER KONFIGURATIONSLISTE

Was die einzelnen Farben bei Meldungen und im Konfigurator bedeuten kann unter den Farbcodes für Meldungen und Konfiguratoren nachgelesen werden. Vom Element werden diese Informationen folgendermaßen dargestellt:

| FARBE | BEDINGUNG |
|---|---|
| grün | instanceID ist 0, create ist definiert |
| rot | instanceID ist nicht 0, create ist nicht definiert oder die definierte instanceID existiert nicht |
| grau | Die Instanz mit der ID instanceID ist nicht konfiguriert wie von create vorgegeben. Stellt create eine Kette dar, so wird die Zeile auch als grau dargestellt, falls ein physikalisches Elternteil anders als in der Kette definiert konfiguriert ist |
| weiß | instanceID ist 0, create ist nicht definiert (Kategorie) |
| weiß | Die Instanz mit der ID instanceID ist konfiguriert wie von create vorgegeben. Stellt create eine Kette dar, so wird die Zeile nur dann als weiß dargestellt, wenn alle physikalischen Elternteile wie in der Kette definiert konfiguriert sind |

## DAS FORMULARFELD CONFIGURATOR IN DER GERÄTESUCHE VERWENDEN

Die Konsole bietet erstellbare Instanzen zentral über die Gerätesuche an. Dies geschieht, indem bei Instanzen des Typs Konfigurator und Discovery regelmäßig GetConfigurationForm aufgerufen wird und der Parameter values bei Configurator-Feldern ausgewertet wird. Es werden nur Konfigurator- und Discovery-Instanzen in der Gerätesuche angezeigt, welche mindestens ein Configurator-Feld beinhalten. In diesem

darf values auch leer sein. Soll die eigene Instanz bei der Gerätesuche betrachtet werden, so müssen daher folgende Bedingungen erfüllt sein:

- Die Instanz muss vom Typ Konfigurator oder Discovery sein
- Die Suche nach Geräten muss automatisch beim Ausführen von GetConfigurationForm gestartet werden. Dauert die Suche länger, so sollte in einer initialen Rückgabe values leer gelassen werden, nach abgeschlossener Suche müssen die erstellbaren Geräte aber in der Rückgabe von GetConfigurationForm enthalten sein
- Die Suche sollte bei wiederholtem Aufruf von GetConfigurationForm nicht erneut gestartet werden sollte, falls diese aktuell schon läuft
- Eine Beispielimplementation kann hier gefunden werden: SymconTest

## AUSGEWÄHLTE ZEILE ALS VARIABLE IN ON-AKTIONEN

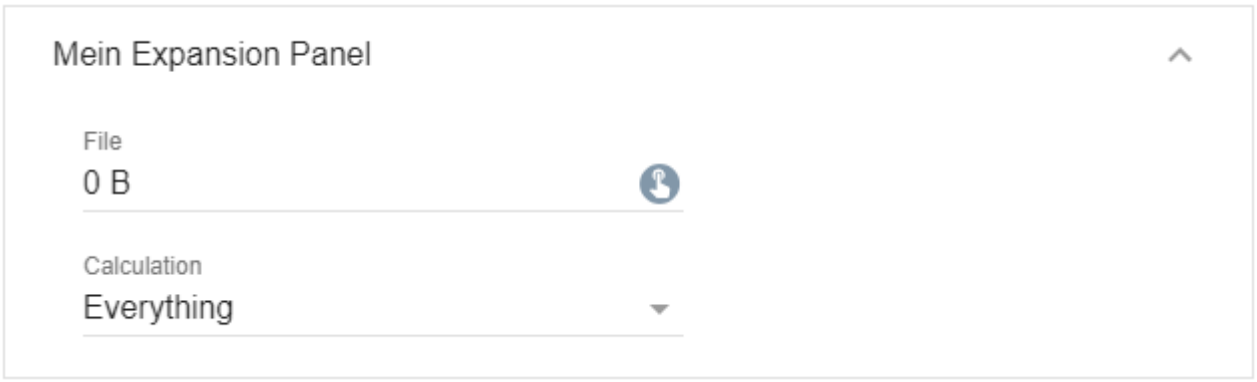
Die Verwendung der Werte des Konfigurators in Skripten erfolgt analog zu Listen. Dort stehen sämtliche Werte zur Verfügung, inklusive expanded, instanceID und create. Dennoch speichert der Konfigurator keine Werte in einer Eigenschaft ab.

## BEISPIEL

```
{
"type": "Configurator",
"values": [{
"name": "Rechenmodul (Umbenannte Statusvariablen)",
"address": "Adresse",
"create": {
"moduleID": "{A7B0B43B-BEB0-4452-B55E-CD8A9A56B052}",
"configuration": {},
"statusVariables": {
"Minimum": {
"name": "Alternativer Name für Minimum"
},
"Maximum": {
"name": "Alternativer Name für Maximum"
}
}
}
}]
})
```

# ExpansionPanel

Benötigt: IP-Symcon >= 5.0
Erstellt ein Expansion Panel, welches weitere Formularfelder enthält und auf- und zugeklappt werden kann, um so den Inhalt platzsparend darzustellen.




## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Titel des Expansion Panels |
| download (optional) | (**default:** "") Ist dieser Parameter nicht leer und beinhaltet die Ausgabe des onClick-Skriptes eine Data-URL, so wird die Rückgabe als Textdatei mit dem Namen das download-Parameters heruntergeladen. (ab IP-Symcon 7.0) |
| expanded (optional) | (**default:** false) Bei true ist das Expansion Panel ausgeklappt, ansonsten ist es eingeklappt (ab IP-Symcon 5.2) |
| items | Liste mit Formularfeldern innerhalb des Expansion Panels (nicht änderbar) |
| link (optional) | (**default:** false) Ist dieser Parameter true, dann wird die Ausgabe des onClick-Skriptes als Link geöffnet. Ist er false, so wird die Ausgabe als Dialog im Konfigurationsformular angezeigt. (ab IP-Symcon 6.0) |
| name (optional) | Name des Buttons |
| onClick (optional) | (**default:** "") Skript, welches bei einem Klick auf das Ausklappicon ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden (Arrays werden ab Version 6.0 unterstützt). Es hat die gleichen Eigenschaften wie onClick des Button (ab IP-Symcon 5.3) |
| type | ExpansionPanel |
| visible (optional) | (**default:** true) Bei true ist das Expansion Panel sichtbar, ansonsten ist es unsichtbar (ab IP-Symcon 5.2) |
| width (optional) | Feste Breite des Expansion Panels in Pixel oder % als String, z.B. "40%" oder "250px". Ist der Wert nicht gesetzt oder "", wird automatisch die volle Breite genutzt (ab IP-Symcon 5.3) |

## BEISPIEL

```
// Richtet ein Expansion Panel ein
{
"type": "ExpansionPanel",
"caption": "Mein Expansion Panel",
"items": [
{
"type": "SelectFile",
"name": "File",
"caption": "File"
},
{
"type": "Select",
"name": "Calculation",
"caption": "Calculation",
"options": [
{ "caption": "Everything", "value": 0 },
{ "caption": "Sum", "value": 1 },
{ "caption": "Minimum", "value": 2 },
{ "caption": "Maximum", "value": 3 },
{ "caption": "Average", "value": 4 },
{ "caption": "Count", "value": 5 }
]
}
]
}
```

# HorizontalSlider

Erstellt einen horizontalen Schieberegler der zwischen den Werten von "**minimum**" und "**maximum**" verschoben werden kann.




## PARAMETER

| PARAMETER | BESCHREIBUNG |
| --- | --- |
| caption | Beschriftung des Schiebereglers |
| download (optional) | (**default:** "") Ist dieser Parameter nicht leer und beinhaltet die Ausgabe des onChange-Skriptes eine Data-URL, so wird die Rückgabe als Textdatei mit dem Namen das download-Parameters heruntergeladen. (ab IP-Symcon 7.0) |
| displayValue (optional) | (**default:** false) Bei true wird der Wert des Schiebereglers rechts vom Regler angezeigt. (ab IP-Symcon 8.1) |
| enabled (optional) | (**default:** true) Bei true kann der Schieberegler verwendet werden, ansonsten wird er deaktiviert dargestellt (ab IP-Symcon 5.2) |
| link (optional) | (**default:** false) Ist dieser Parameter true, dann wird die Ausgabe des onChange-Skriptes als Link geöffnet. Ist er false, so wird die Ausgabe als Dialog im Konfigurationsformular angezeigt. (ab IP-Symcon 6.0) |
| maximum (optional ab IP-Symcon 5.3) | (**default:** 100) Maximumwert der im Schieberegler einstellbar ist |
| minimum (optional ab IP-Symcon 5.3) | (**default:** 0) Minimumwert der im Schieberegler einstellbar ist |
| name (optional) | Name des Schiebereglers |
| onChange (optional ab IP-Symcon 5.3) | (**default:** "") Skript, welches bei Verschiebung des Reglers ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden (Arrays werden ab Version 6.0 unterstützt). Es hat die gleichen Eigenschaften wie onClick des Button |
| prefix (optional) | (**default:** "") Sichtbare Beschriftung/Einheit vor der Wertanzeige (ab IP-Symcon 8.1) |
| stepSize (optional) | (**default:** 1) Schrittgröße, in welcher der Schieberegler verstellt werden kann (ab IP-Symcon 5.3) |
| suffix (optional) | (**default:** "") Sichtbare Beschriftung/Einheit hinter der Wertanzeige (ab IP-Symcon 8.1) |
| type | HorizontalSlider |
| value (optional) | (**default:** minimum) Der Wert des Schiebereglers - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben (ab IP-Symcon 5.2) |
| visible (optional) | (**default:** true) Bei true ist der Schieberegler sichtbar, ansonsten ist er unsichtbar (ab IP-Symcon 5.2) |
| width (optional) | (**default:** 300px) Feste Breite des Schiebereglers in Pixel oder % als String, z.B. "40%" oder "250px" (ab IP-Symcon 5.3) |

## BEISPIEL

```
//Das Beispiel gibt die InstanzID zurück.
{ "type": "HorizontalSlider", "name": "Slider", "caption": "ID-
Schieberegler", "minimum": 0, "maximum": 16, "onChange": "echo $id;"
}
```

# Image

Benötigt: IP-Symcon >= 5.0
Erstellt ein Bild, welches auf der Modulseite dargestellt wird. Das Bild kann direkt angegeben werden oder aus einem Medienobjekt geladen werden.

## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| download (optional) | (**default:** "") Ist dieser Parameter nicht leer und beinhaltet die Ausgabe des onClick-Skriptes eine Data-URL, so wird die Rückgabe als Textdatei mit dem Namen das download-Parameters heruntergeladen. (ab IP-Symcon 7.0) |
| image (optional) | Inhalt des Bildes als Data-URI (Mehr Infos, nicht kombinierbar mit **mediaID**, entweder **image** oder **mediaID** muss gesetzt sein) |
| link (optional) | Ist dieser Parameter true, dann wird die Ausgabe des onClick-Skriptes als Link geöffnet. Ist er false, so wird die Ausgabe als Dialog im Konfigurationsformular angezeigt. Ist der Parameter nicht gesetzt, dann wird auf Basis der Ausgabe entschieden, ob diese als Link geöffnet oder im Dialog angezeigt wird (siehe Infobox). (ab IP-Symcon 6.0) |
| mediaID (optional) | MedienID des dargestellten Bildes oder Streams (ab IP-Symcon 5.5) (Nicht kombinierbar mit **image**, entweder **image** oder **mediaID** muss gesetzt sein) |
| name (optional) | Name des Bildes |
| type | Image |
| visible (optional) | (**default:** true) Bei true ist das Bild sichtbar, ansonsten ist es unsichtbar (ab IP-Symcon 5.2) |
| center (optional) | (**default:** false) Ist **center** gesetzt, so wird das Bild innerhalb des dargestellten Bereiches horizontal zentriert, ansonsten ist es linksbündig. Innerhalb eines RowLayout hat diese Einstellung keine Funktion. |
| width (optional) | Feste Breite des Bildes in Pixel oder % als String, z.B. "40%" oder "250px". Ist der Wert nicht gesetzt oder "", wird die Breite automatisch auf Basis des Bildes gewählt (ab IP-Symcon 6.0) |
| onClick | Skript, welches bei einem Klick auf das Bild ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden. Es hat die gleichen Eigenschaften wie onClick des Button. (ab IP-Symcon 6.0) |



Steht in dem Parameter onClick ein "echo" in Kombination mit 'http'/'https' wird der Browser geöffnet und die URL aufgerufen. Ist ein "echo" in Kombination mit einem 'mailto:' vorhanden wird das Standard-Mailprogramm mit allen gesetzten Parametern geöffnet (Verfügbar ab IP-Symcon 6.0)

## BEISPIEL

```
// Bild base64-codiert darstellen
{
"type": "Image",
"image": "
iVBORw0KGgoAAAANSUhEUgAAAAUAAAAFCAYAAACNbyblAAAAHElEQVQI12P4//8/
w38GIAXDIBKE0DHxgljNBAAO9TXL0Y4OHwAAAABJRU5ErkJggg=="
}

// Bild aus Medium darstellen
{
"type": "Image",
"mediaID": 12345
}
```

# IntervalBox (veraltet)

Ab 5.0 ist die IntervalBox veraltet und wird zu einem zukünftigen Zeitpunkt entfernt. Der NumberSpinner unterstützt den Parameter **suffix** und bietet somit sämtliche Funktionen der IntervalBox. Daher sind die Parameter der IntervalBox auch nicht änderbar.

Erstellt eine IntervalBox welche ein Zahleneingabefeld hat, welches eine Beschriftung/ Einheit hinter dem Eingabefeld besitzt.
Falls im "elements"-Bereich erstellt, setzt die IntervalBox eine Eigenschaft bei Übernahme auf den eingetragenen Zahlenwert.
Der Parameter **"name"** definiert dabei welche Eigenschaft gesetzt wird.




## PARAMETER

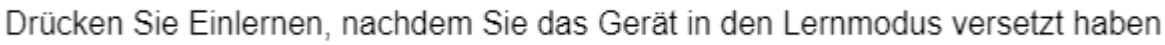
| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung/Einheit hinter dem Eingabefeld |
| type | IntervalBox |
| name (optional) | Name der Intervalbox/der zu setzenden Eigenschaft |

## BEISPIEL

```
{ "type": "IntervalBox", "name": "CacheInterval", "caption":
"Sekunden" }
```

# Label

Erstellt ein Label, welches der Übersicht dient.




## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| bold (optional) | (**default:** false) Bei true wird das Label fett dargestellt |
| caption | Sichtbare Beschriftung |
| color (optional) | (**default:** 0 (schwarz)) Das Label wird in der angegebenen Farbe dargestellt |
| italic (optional) | (**default:** false) Bei true wird das Label kursiv dargestellt |
| ~~label~~ | ~~sichtbare Beschriftung~~ *(veraltet, nicht änderbar)* |
| link (optional) | (**default:** true) Ist dieser Parameter gesetzt, so werden URLs innerhalb der Beschriftung, welche mit "http://" oder "https://" beginnen als Links dargestellt (ab IP-Symcon 5.3) |
| name (optional) | Name des Labels |
| type | Label |
| underline (optional) | (**default:** false) Bei true wird das Label unterstrichen dargestellt |
| visible (optional) | (**default:** true) Bei true ist das Label sichtbar, ansonsten ist es unsichtbar (ab IP-Symcon 5.2) |
| width (optional) | Feste Breite des Labels in Pixel oder % als String, z.B. "40%" oder "250px". Ist der Wert nicht gesetzt oder "", wird die Breite automatisch auf Basis der Beschriftung gewählt (ab IP-Symcon 5.3) |

## BEISPIEL

```
{ "type": "Label", "caption": "Drücken Sie Einlernen, nachdem Sie
das Gerät in den Lernmodus versetzt haben" }
```

# List

Benötigt: IP-Symcon >= 4.2
Erstellt eine Liste mit der Überschrift **caption**.
Die angezeigten Spalten werden in **columns** definert.
Initiale Einträge der Liste können via **values** vorgegeben werden.




## ALLGEMEINE PARAMETER

| PARAMETER | BESCHREIBUNG |
| --- | --- |
| add (optional) | (**default:** false) Wenn true, dann wird unter der Tabelle ein "Hinzufügen"-Button eingeblendet, welcher der Tabelle ein neues Element mit Standardwerten hinzufügt |
| caption (optional) | Sichtbare Beschriftung der Liste |
| columns | Spalten der Liste |
| delete (optional) | (**default:** false) Wenn true, dann wird hinter jedem Eintrag in der Liste ein Button eingeblendet, welcher das entsprechende Element löscht |
| download (optional) | (**default:** "") Ist dieser Parameter nicht leer und beinhaltet die Ausgabe des onAdd, onChangeOrder, onDelete oder onEdit-Skriptes eine Data-URL, so wird die Rückgabe als Textdatei mit dem Namen das download-Parameters heruntergeladen. (ab IP-Symcon 7.0) |
| form (optional) | (**default:** basierend auf den edit-Parametern von columns) Ein individuelles Formular zum Bearbeiten oder Hinzufügen von Zeilen. Die Definition des Formulars erfolg analog zur Definition des Formulars für Aktionen statisch oder als PHP-Code. Wie das Formular ausgewertet und gefüllt wird ist im Abschnitt Individuelle Formulare zum Bearbeiten erklärt (ab IP-Symcon 7.0) |
| link (optional) | (**default:** false) Ist dieser Parameter true, dann wird die Ausgabe des onAdd, onChangeOrder, onDelete oder onEdit-Skriptes als Link geöffnet. Ist er false, so wird die Ausgabe als Dialog im Konfigurationsformular angezeigt. (ab IP-Symcon 6.0) |
| name (optional) | Name der Liste / der zu setzenden Eigenschaft |
| onAdd (optional) | (**default:** "") Skript, welches nach dem Hinzufügen eines Eintrages der Liste ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden (Arrays werden ab Version 6.0 unterstützt). In dieser Ausführung bietet die Liste als Variablenwert nicht den aktuell ausgewählten Eintrag sondern den hinzugefügten Eintrag, siehe Ausgewählte Zeile als Variable in on-Aktionen. Ansonsten hat es die gleichen Eigenschaften wie onClick des Button (ab IP-Symcon 5.3) |
| onChangeOrder (optional) | (**default:** "") Skript, welches nach dem Verschieben eines Eintrages der Liste ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden. In dieser Ausführung bietet die Liste als Variablenwert nicht den aktuell ausgewählten Eintrag sondern den verschobenen Eintrag, siehe Ausgewählte Zeile als Variable in on-Aktionen. Ansonsten hat es die gleichen Eigenschaften wie onClick des Button (ab IP-Symcon 6.3) |
| onDelete (optional) | (**default:** "") Skript, welches nach dem Löschen eines Eintrages der Liste ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden (Arrays werden ab Version 6.0 unterstützt). In dieser Ausführung bietet die Liste als Variablenwert nicht den aktuell ausgewählten Eintrag sondern den gelöschten Eintrag, siehe Ausgewählte Zeile als |

## PARAMETER FÜR COLUMNS

| PARAMETER | BESCHREIBUNG |
|---|---|
| add (optional) | initialer Wert in dieser Spalte bei einem neu hinzugefügtem Eintrag, notwendig wenn **add** bei der Liste auf true gesetzt ist |
| align (optional) | (**default:** "left") Ausrichtung der Spalte (ab IP-Symcon 5.2)<br>"left": Inhalt wird linksbündig dargestellt<br>"center": Inhalt wird zentriert dargestellt<br>"right": Inhalt wird rechtsbündig dargestellt |
| caption | Sichtbare Beschriftung einer Spalte |
| confirm (optional) | (**default:** "") Abfragetext, bevor eine onClick-Aktion ausgeführt wird. Nur wenn die Abfrage mit "Ja" bestätigt wird, wird die Aktion ausgeführt. Ist kein Abfragetext definiert, so wird die Aktion direkt ausgeführt (ab IP-Symcon 5.2) |
| download (optional) | (**default:** "") Ist dieser Parameter nicht leer und beinhaltet die Ausgabe des onClick-Skriptes dieser Spalte eine Data-URL, so wird die Rückgabe als Textdatei mit dem Namen das download-Parameters heruntergeladen. (ab IP-Symcon 7.1) |
| edit (optional) | Beschreibung eines editierbaren Elements |
| ~~label~~ | ~~sichtbare Beschriftung einer Spalte~~ *(veraltet)* |
| link (optional) | (**default:** false) Ist dieser Parameter true, dann wird die Ausgabe des onClick-Skriptes dieser Spalte als Link geöffnet. Ist er false, so wird die Ausgabe als Dialog im Konfigurationsformular angezeigt. (ab IP-Symcon 7.1) |
| name | Eindeutiger Name einer Spalte um diese zu referenzieren; der Name darf nicht "rowColor", "editable" oder "deletable" sein, da diese Werte für andere Funktionen belegt sind, siehe Parameter für values |
| onClick (optional) | (**default:** "") Skript, welches bei einem Klick auf ein Feld dieser Spalte ausgeführt wird. Es hat die gleichen Eigenschaften wie onClick des Button. Das Skript wird nur ausgeführt, sofern dieses Feld nicht leer, also 0, false oder "" ist (ab IP-Symcon 5.2) |
| quickFilter (optional) | (**default:** false) Bei true wird der Text dieser Spalte beim Schnellfilter in der Suche betrachtet. Sollte der Wert bei allen Spalten auf false gesetzt sein, so wird der Schnellfilter ausgeblendet. (ab IP-Symcon 7.2) |
| sortColumn (optional) | Ist sortColumn gesetzt und wird nach dieser Spalte sortiert, so werden statt den Werten dieser Spalte, die Werte der Spalte mit dem Namen dieses Parameters verglichen |
| save (optional) | (**default:** true, falls das Element editierbar ist, sonst false) Bei true sollen die Werte dieser Spalte als Modul-Eigenschaften gesichert werden, siehe Speichern und Laden von Listen |
| visible (optional) | (**default:** true) Bei false wird die Spalte nicht angezeigt |
| width | Breite der Spalte in Pixeln, als CSS-String (z.B. "100px"); eine einzige Spalte darf den Wert "auto" haben, wodurch die Breite dieser Spalte den restlich verfügbaren Platz nutzt |

## PARAMETER FÜR EDIT

| PARAMETER | BESCHREIBUNG |
| --- | --- |
| type | Typ der Darstellung, entsprechend der möglichen Eingaben: CheckBox, IntervalBox, NumberSpinner, PasswordTextBox, Select, SelectCategory, SelectColor, SelectEvent, SelectFile, SelectInstance, SelectLink, SelectMedia, SelectObject, SelectScript, SelectVariable, ValidationTextBox, List, Tree |
| Weitere Parameter | Die weiteren Parameter ergeben sich je nach gewähltem Typ; die Parameter "name" und "caption" des Typs werden ignoriert |



Wird der "type" List oder Tree verwendet, so wird der Wert dieser Spalte nicht noch einmal JSON-codiert. Es ist also lediglich erforderlich den Wert der übergeordneten Liste einmalig zu parsen. Jede Unterliste bzw. Unterbaum steht dann direkt als Objekt zur Verfügung. Analog müssen die Werte in "values" auch nicht doppelt codiert werden.



Der Parameter "add" der Spalte wird entsprechend der Lokalisierung übersetzt, sofern der "type" von "edit" ValidationTextBox oder PasswordTextBox ist oder "edit" gar nicht gesetzt ist



Es ist möglich mit der Funktion UpdateFormField nur Teile von von columns oder values zu aktualisieren, indem eine Punktnotation verwendet wird. In dem Fall wird als Ziel columns.x.y bzw. values.x.y verwendet. x bezieht sich dabei auf den Index der betroffenen Spalte bzw. des betroffenen Wertes und y auf den Namen des Feldes. Es ist auch möglich einen einzelne Spalte bzw. einen einzelnen Wert komplett zu bearbeiten indem als Ziel columns.x bzw. values.x verwendet wird. Um also beispielsweise die Breite der zweiten Spalte anzupassen, kann der Befehl $this->UpdateFormField('list', 'columns.1.width', '200px'); verwendet werden.

## PARAMETER FÜR SORT

| PARAMETER | BESCHREIBUNG |
| --- | --- |
| column | Name der Spalte, nach welcher die Liste sortiert wird |
| direction | Die Richtung der Sortierung; "ascending" für eine aufsteigende Sortierung und "descending" für eine absteigende Sortierung |

## PARAMETER FÜR VALUES

| PARAMETER | BESCHREIBUNG |
|---|---|
| deletable (optional) | (**default:** true) Bei false kann die Zeile nicht gelöscht werden (ab IP-Symcon 5.4) |
| editable (optional) | (**default:** true) Bei false kann die Zeile nicht editiert werden, obwohl die Liste an sich editierbare Spalten besitzt (ab IP-Symcon 5.4) |
| rowColor (optional) | (**default:** transparent) Hintergrundfarbe der Zeile als Hex-Code (Defaultfarben s.u.), leerer String für Transparenz |
| Weitere Parameter (optional) | (**default:** "") Für jeden initialen Eintrag der Liste werden hier die Werte zu allen Spalten definiert. Hierfür wird für jede Spalte ein Parameter mit dem Namen der Spalte eingetragen. Die Belegung des Parameters entspricht dem Wert dieser Spalte bei dem entsprechenden Eintrag. |

| FARBE | HEX-CODE |
|---|---|
| Hellgrün | #C0FFC0 |
| Lila | #C0C0FF |
| Gelb | #FFFFC0 |
| Grau | #DFDFDF |
| Rot | #FFC0C0 |

Was die einzelnen Farben bei Meldungen und im Konfigurator bedeuten kann unter den Farbcodes für Meldungen und Konfiguratoren nachgelesen werden.

## SPEICHERN UND LADEN VON LISTEN

Beim Speichern der Liste werden alle Listenelemente, die editierbar sind oder den Paramter **save** gesetzt haben, als Modul-Eigenschaften gesichert.
Wird die Liste erneut geladen und ist der Parameter **loadValuesFromConfiguration** nicht auf false gesetzt, so werden zuerst die Werte eingetragen, die als Modul-Eigenschaften gesichert wurden. Steht **loadValuesFromConfiguration** auf false, werden keine Werte aus den Eigenschaften geladen sondern nur wie in den weiteren Schritten beschrieben geladen. Sollen dennoch Werte aus der Eigenschaft als Grundlage verwendet werden, so müssen diese selbstständig in **values** eingetragen werden. Die weiteren Elemente, inklusive der Farbe, werden mit den initialen Werten aus **values** aufgefüllt.
Existieren mehr Zeilen als Elemente in **values**, so werden Elemente der überzähligen Zeilen, welche nicht als Modul-Eigenschaften gesichert waren, mit dem Initialwert **add** der entsprechenden Spalte gefüllt.

## AUSGEWÄHLTE ZEILE ALS VARIABLE IN ON-AKTIONEN

In der Definition des Konfigurationsformulares können die Werte der Tabelle als Variable **$Name** verwendet werden. Auf das Objekt kann wie auf ein Array zugegriffen werden. Wird auf das entsprechende Objekt mit einem String-Index zugegriffen, so wird der entsprechende Wert wie in values definiert der ausgewählten Zeile zurückgegeben. Wird mit einem numerischen Index zugegriffen oder das Objekt iteriert, so wird die entsprechende Zeile zurückgegeben, deren einzelne Parameter widerum per String-Index referenziert werden können.
Beispiel:

```
//Gebe den Namen der aktuell gewählte Zeile der Tabelle devices aus
{ "type": "Button", "caption": "Ausgabe", "onClick":
"print_r($devices['name']);" }

//Gebe den Namen der ersten Zeile der Tabelle devices aus
{ "type": "Button", "caption": "Ausgabe", "onClick":
"print_r($devices[0]['name']);" }
```

## INDIVIDUELLE FORMULARE ZUM BEARBEITEN

Standardmäßig wird beim Bearbeiten oder Hinzufügen einer Zeile ein Formular auf Basis der edit-Parameter der einzelnen Spalten verwendet. Ist allerdings der Parameter "form" gesetzt, so wird stattdessen das davon beschriebene Formular verwendet.
Wird dabei eine Zeile bearbeitet, wird der Wert jedes Formularfeld basierend auf dem Wert mit gleichem Namen in der bearbeiteten Zeile gesetzt, sofern vorhanden. Beim Hinzufügen werden die initialen Werte der Formularfelder stattdessen basierend auf dem add-Wert der Spalte mit gleichem Namen gesetzt, falls solch eine Spalte existiert. Beim Bestätigen des Dialogs werden für die Zeile alle Werte des Dialogs mit dem Namen des Elements gespeichert. Falls dann noch Spalten existieren, deren Namen nicht im Formular vorkommen, wird deren Wert entsprechend des add-Wertes gesetzt.
Beim Übernehmen der Konfiguration werden alle Werte, die zu einer Spalte passen wie üblich basierend auf dem save-Parameter abgespeichert. Passt ein Wert zu keiner Spalte, so wird er auf jeden Fall abgespeichert.
Teile eines individuellen Formulars können wie im regulären Formular via UpdateFormField dynamisch angepasst werden.
Beispiele zur Verwendung des form-Parameters können im Test-Repository gefunden werden.

## WERTE AUS DER TABELLE IN PHP REGISTRIEREN UND AUSLESEN

Wurde die Tabelle im Konfigurationsformular im Bereich **elements** definiert, kann die Tabelle durch die Funktion RegisterPropertyString registriert werden. Die Tabelle wird dann als JSON-codierter String gespeichert.
Die Funktion ReadPropertyString kann genutzt werden um den gespeicherten String auszulesen. In einem nächsten Schritt sollte die Funktion **json_decode** verwendet werden um den eingelesenen String in ein Array zu konvertieren.
Beispiel:

```
$arrString = $this->ReadPropertyString("devices");
$arr = json_decode($arrString);
```

## BEISPIEL

```
//Einfache editierbare Liste von Instanzen und dazugehörigen Werten
//Die Liste enthält initial einen Eintrag
{
```

```
"type": "List",
"name": "Devices",
"caption": "Devices",
"rowCount": 5,
"add": true,
"delete": true,
"sort": {
"column": "Name",
"direction": "ascending"
},
"columns": [{
"caption": "InstanceID",
"name": "InstanceID",
"width": "75px",
"add": 0,
"edit": {
"type": "SelectInstance"
}
}, {
"caption": "Name",
"name": "Name",
"width": "auto",
"add": ""
}, {
"caption": "State",
"name": "State",
"width": "40px",
"add": "New!"
}, {
"caption": "Temperature",
"name": "Temperature",
"width": "75px",
"add": 20.0,
"edit": {
"type": "NumberSpinner",
"digits": 2
}
}],
"values": [{
"InstanceID": 12435,
"Name": "ABCD",
"State": "OK!",
"Temperature": 23.31,
"rowColor": "#ff0000" //rot
}]
}
```

Wird in der Beispieltabelle nun der Wert **instanceID** des ersten Eintrages auf 54321 und **temperature** auf 37.0 geändert und die Tabelle gespeichert, so werden die neuen Werte für **instanceID** und **temperature** als Modul-Eigenschaften persistent gespeichert. Da die weiteren Elemente, also **name**, **state** und die Farbe der Zeile **rowColor**, nicht editierbar sind, werden diese auch nicht gespeichert.
Wird die Tabelle nun erneut geladen, so werden zuerst die gespeicherten Werte für **instanceID** und **temperature** geladen. Die nicht-editierbaren Felder **name** und **state** werden dann mit den Werten aus **values**, also "ABCD" und "OK!" gefüllt. Die Farbe wird ebenfalls aus **values** geladen, wodurch die Zeile einen roten Hintergrund erhält.
Hätte man der Tabelle zusätzlich eine weitere Zeile hinzugefügt und die Werte **instanceID**

und **temperature** auf 314 bzw. -5.3 gesetzt, so würden diese ebenfalls aus den Modul-Eigenschaften geladen werden. Da sich in **values** lediglich ein Element befindet, werden die nicht-editierbaren Felder entsprechend der Standardwerte aus **add** gefüllt. Das Feld **name** der zweiten Zeile würde leer bleiben und **state** würde auf "OK!" gesetzt werden. Da keine Farbe definiert ist, wäre der Hintergrund dieser Zeile transparent.

```
// Liste mit alternativer Sortierung
{
"type": "List",
"name": "sortList",
"rowCount": 5,
"add": false,
"columns": [
{
"caption": "Text",
"name": "Text",
"width": "auto",
"sortColumn": "Number"
},
{
"caption": "Number",
"name": "Number",
"width": "200px",
"visible": false
}
],
"values": [
{
"Text": "One",
"Number": "1"
},
{
"Text": "Two",
"Number": "2"
},
{
"Text": "Three",
"Number": "3"
}
]
}
```

# NumberSpinner

Erstellt ein Eingabefeld für Zahlen mit der Überschrift **caption**.
Falls im "elements"-Bereich erstellt, setzt der NumberSpinner eine Eigenschaft bei Übernahme auf den eingetragenen Zahlenwert.
Der Parameter **"name"** definiert dabei welche Eigenschaft gesetzt wird.



Der Paramter digits dient dazu ob die übergebene Zahl als Integer(Ganzzahlig) oder Float(Fließkommazahl) interpretiert wird.




## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung des NumberSpinners |
| digits (optional) | (**default:** 0) Gibt die Anzahl der Nachkommastellen an. (Nicht kombinierbar mit hex) |
| enabled (optional) | (**default:** true) Bei true kann der NumberSpinner verwendet werden, ansonsten wird er deaktiviert dargestellt (ab IP-Symcon 5.2) |
| hex (optional) | (**default:** false) Wenn true, dann ist eine hexadezimale Eingabe {1..9,A..F} erlaubt. (Nicht kombinierbar mit digits) |
| minimum (optional) | (**default:** kein Minimalwert) Definiert einen Minimalwert - Ist der eingetragene Wert kleiner als der Minimalwert, so wird ein Fehler angezeigt und Änderungen können nicht übernommen werden (ab IP-Symcon 5.3) |
| maximum (optional) | (**default:** kein Maximallwert) Definiert einen Maximalwert - Ist der eingetragene Wert größer als der Maximalwert, so wird ein Fehler angezeigt und Änderungen können nicht übernommen werden (ab IP-Symcon 5.3) |
| moreDigits (optional) | (**default:** false) Bei false ist der NumberSpinner auf die angegebene Anzahl von Nachkommastellen beschränkt. Wird diese überschritten, so wird ein Fehler angezeigt und Änderungen können nicht übernommen werden. Bei true können beliebig viele Nachkommastellen angegeben werden (Nicht kombinierbar mit hex; ab IP-Symcon 6.0) |
| name (optional) | Name des NumberSpinners/der zu setzenden Eigenschaft |
| onChange (optional) | (**default:** "") Skript, welches bei einer Veränderung des Dropdown-Wertes ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden. Es hat die gleichen Eigenschaften wie onClick des Button (ab IP-Symcon 8.1) |
| suffix (optional) | (**default:** "") Sichtbare Beschriftung/Einheit hinter dem Eingabefeld |
| type | NumberSpinner |
| value (optional) | (**default:** 0) Der Wert des NumberSpinners - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben (ab IP-Symcon 5.2) |
| visible (optional) | (**default:** true) Bei true ist der NumberSpinner sichtbar, ansonsten ist er unsichtbar (ab IP-Symcon 5.2) |
| width (optional) | (**default:** 300px) Feste Breite des Eingabefeldes in Pixel oder % als String, z.B. "40%" oder "250px" (ab IP-Symcon 5.3) |

## BEISPIEL

```
// ohne digits/hex Parameter
{ "type": "NumberSpinner", "name": "Port", "caption": "Port" }
// mit digits Parameter
{ "type": "NumberSpinner", "name": "Latitude", "caption": "Latitude
(N)", "digits": 2}
// mit hex Parameter
{ "type": "NumberSpinner", "name": "Value", "caption": "Hexvalue",
"hex": true}
// mit Suffix
{ "type": "NumberSpinner", "name": "Seconds", "caption": "Seconds",
```

```
 "suffix": "seconds"}
```

# OpenObjectButton

Benötigt: IP-Symcon >= 5.3
Ein Button der bei Betätigung ein Objekt öffnet. Bei einer Instanz wird die Instanzkonfiguration geöffnet und bei einem Skript der Skripteditor. Bei allen anderen Objekttypen wird der Bearbeiten-Dialog im Objektbaum geöffnet.




## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung des Buttons |
| enabled (optional) | (**default:** true) Bei true kann der Button verwendet werden, ansonsten wird er deaktiviert dargestellt |
| name (optional) | Name des Buttons |
| objectID | ID des zu öffnenden Objektes |
| type | OpenObjectButton |
| visible (optional) | (**default:** true) Bei true ist der Button sichtbar, ansonsten ist er unsichtbar |
| width (optional) | Feste Breite des Buttons in Pixel oder % als String, z.B. "40%" oder "250px". Ist der Wert nicht gesetzt oder "", wird die Breite automatisch auf Basis der Beschriftung gewählt |

## BEISPIEL

```
{ "type": "OpenObjectButton", "caption": "Zum Connect Control",
"objectID": 42563 }
```

# PasswordTextBox

Erstellt ein Eingabefeld für Passwörter mit der Überschrift **caption**.
Falls im "elements"-Bereich erstellt, setzt die PasswordTextBox eine Eigenschaft bei Übernahme auf den eingetragenen String.
Der Parameter **"name"** definiert dabei welche Eigenschaft gesetzt wird.




## PARAMETER

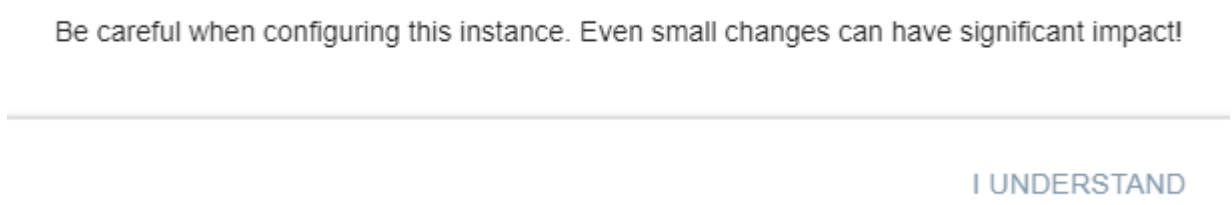
| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung des Passwort-Eingabefeldes |
| enabled (optional) | (**default:** true) Bei true kann das Passwort-Eingabefeld verwendet werden, ansonsten wird es deaktiviert dargestellt (ab IP-Symcon 5.2) |
| name (optional) | Name des Passwort-Eingabefeldes/der zu setzenden Eigenschaft |
| onChange (optional) | (**default:** "") Skript, welches bei einer Veränderung des Dropdown-Wertes ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden. Es hat die gleichen Eigenschaften wie onClick des Button (ab IP-Symcon 8.1) |
| type | PasswordTextBox |
| validate (optional) | (**default:** "") Erfüllt der Wert nicht den hier angegebenen regulären Ausdruck, so wird ein Fehler angezeigt und Änderungen können nicht übernommen werden (ab IP-Symcon 5.3) |
| value (optional) | (**default:** "") Der Wert des Passwort-Eingabefeldes - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben (ab IP-Symcon 5.2) |
| visible (optional) | (**default:** true) Bei true ist das Passwort-Eingabefeld sichtbar, ansonsten ist es unsichtbar (ab IP-Symcon 5.2) |
| width (optional) | (**default:** 300px) Feste Breite des Eingabefeldes in Pixel oder % als String, z.B. "40%" oder "250px" (ab IP-Symcon 5.3) |

## BEISPIEL

```
{ "type": "PasswordTextBox", "name": "Password", "caption":
"Passwort" }
```

# PopupAlert

Benötigt: IP-Symcon >= 5.0
Enthält das Konfigurationsformular dieses Element, so wird es beim Laden der Konfigurationsseite geöffnet. Nachdem das Popup geschlossen wurde, ist es zusätzlich möglich dies mithilfe von Dynamik (visible auf true) erneut zu öffnen.




## ALLGEMEINE PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| name (optional) | Name des PopupAlert |
| popup | Definition des zu öffnenden Popups (Parameter sind ab IP-Symcon 6.1 änderbar durch Verknüpfung der Paramternamen, beispielsweise popup.closeCaption) |
| type | PopupAlert |
| visible (optional) | (**default:** true) Ist visible initial true, so wird das Popup beim Öffnen der Instanzkonfiguration geöffnet, ansonsten passiert nichts. Durch Änderung auf true kann das Popup im Nachhinein geöffnet werden. Eine Änderung auf false schließt ein aktuell geöffnetes Popup. (ab IP-Symcon 5.2) |

## PARAMETER FÜR POPUP

| PARAMETER | BESCHREIBUNG |
|---|---|
| buttons (optional) | (**default:** []) Eine Liste von Buttons welche im Popup unten rechts dargestellt wird. An erster Stelle wird immer der Schließen-Button dargestellt, danach folgen die hier beschriebenen Buttons in Reihenfolge (ab IP-Symcon 6.1) |
| closeCaption (optional) | (**default:** "OK") Die Beschriftung des Schließen-Buttons unten rechts im Popup |
| items | Auflistung von Formularfeldern im Popup (nicht änderbar) |

## PARAMETER FÜR BUTTONS

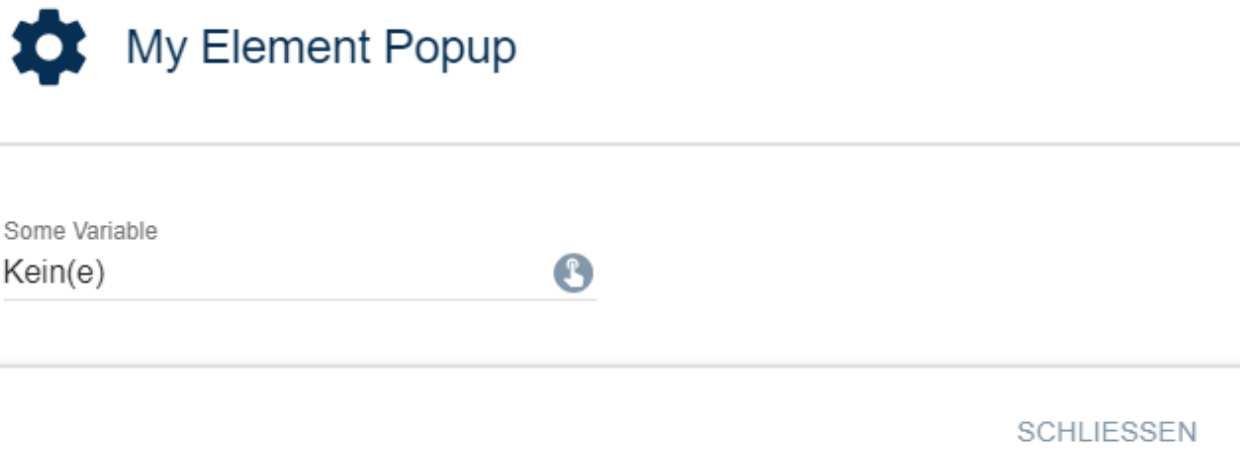
| PARAMETER | BESCHREIBUNG |
| --- | --- |
| caption | Sichtbare Beschriftung des Buttons |
| onClick | Skript, welches bei einem Klick auf den Button ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden. Es sind keine PHP-Tags erforderlich. Es hat die gleichen Eigenschaften wie onClick des Button. Sofern das Skript keine Ausgabe macht, wird das Popup nach Abarbeitung des Skriptes geschlossen. Ansonsten wird die Ausgabe als rote Fehlermeldung unten im Popup angzeigt. Eine Ausnahme besteht hier wenn die Rückgabe mit dem Präfix "MESSAGE:" beginnt. In diesem Fall wird das Popup geschlossen und die Rückgabe hinter dem Präfix in einem Meldungsdialog angezeigt. |
| enabled (optional) | (**default:** true) Bei true kann der Button verwendet werden, ansonsten wird er deaktiviert dargestellt |

## BEISPIEL

```
{
"type": "PopupAlert",
"popup": {
"closeCaption": "I understand",
"items": [
{
"type": "Label",
"caption": "Be careful when configuring this instance. Even small
changes can have significant impact!"
}
]
}
},
{
"type": "PopupAlert",
"popup": {
"items": [
{
"type": "SelectVariable",
"name": "VariableActionTest",
"caption": "Some Action Variable"
}
]
}
}
```

# PopupButton

Benötigt: IP-Symcon >= 5.0
Erstellt einen Button, welcher durch einen Klick ein Popup öffnet, welches weitere Formularfelder enthält.




## ALLGEMEINE PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung des Buttons |
| download (optional) | (**default:** "") Ist dieser Parameter nicht leer und beinhaltet die Ausgabe des onClick-Skriptes eine Data-URL, so wird die Rückgabe als Textdatei mit dem Namen das download-Parameters heruntergeladen. (ab IP-Symcon 7.0) |
| enabled (optional) | (**default:** true) Bei true kann der Button verwendet werden, ansonsten wird er deaktiviert dargestellt. Dieser Parameter hat keine Auswirkung auf ein möglicherweise geöffnetes Popup (ab IP-Symcon 5.2) |
| link (optional) | (**default:** false) Ist dieser Parameter true, dann wird die Ausgabe des onClick-Skriptes als Link geöffnet. Ist er false, so wird die Ausgabe als Dialog im Konfigurationsformular angezeigt. (ab IP-Symcon 6.0) |
| name (optional) | Name des PopupButtons |
| onClick (optional) | (**default:** "") Skript, welches bei einem Klick auf den Button ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden (Arrays werden ab Version 6.0 unterstützt). Es hat die gleichen Eigenschaften wie onClick des Button (ab IP-Symcon 5.3) |
| popup | Definition des zu öffnenden Popups (Parameter sind ab IP-Symcon 6.1 änderbar durch Verknüpfung der Paramternamen, beispielsweise popup.closeCaption) |
| type | PopupButton |
| visible (optional) | (**default:** true) Bei true ist der Button sichtbar, ansonsten ist er unsichtbar. Dieser Parameter hat keine Auswirkung auf ein möglicherweise geöffnetes Popup (ab IP-Symcon 5.2) |
| width (optional) | Feste Breite des Buttons in Pixel oder % als String, z.B. "40%" oder "250px". Ist der Wert nicht gesetzt oder "", wird die Breite automatisch auf Basis der Beschriftung gewählt (ab IP-Symcon 5.2) |

## PARAMETER FÜR POPUP

| PARAMETER | BESCHREIBUNG |
|---|---|
| buttons (optional) | (**default:** []) Eine Liste von Buttons welche im Popup unten rechts dargestellt wird. An erster Stelle wird immer der Schließen-Button dargestellt, danach folgen die hier beschriebenen Buttons in Reihenfolge (ab IP-Symcon 6.1) |
| caption (optional) | (**default:** "") Sichtbarer Titel des Popups |
| closeCaption (optional) | (**default:** "Schließen") Die Beschriftung des Schließen-Buttons unten rechts im Popup (ab Version 6.1) |
| items | Auflistung von Formularfeldern im Popup (nicht änderbar) |

## PARAMETER FÜR BUTTONS

| PARAMETER | BESCHREIBUNG |
| --- | --- |
| caption | Sichtbare Beschriftung des Buttons |
| onClick | Skript, welches bei einem Klick auf den Button ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden. Es sind keine PHP-Tags erforderlich. Es hat die gleichen Eigenschaften wie onClick des Button. Sofern das Skript keine Ausgabe macht, wird das Popup nach Abarbeitung des Skriptes geschlossen. Ansonsten wird die Ausgabe als rote Fehlermeldung unten im Popup angzeigt. Eine Ausnahme besteht hier wenn die Rückgabe mit dem Präfix "MESSAGE:" beginnt. In diesem Fall wird das Popup geschlossen und die Rückgabe hinter dem Präfix in einem Meldungsdialog angezeigt. |
| enabled (optional) | (**default:** true) Bei true kann der Button verwendet werden, ansonsten wird er deaktiviert dargestellt |

## BEISPIEL

```
{
"type": "PopupButton",
"caption": "Open Popup",
"popup": {
"caption": "My Element Popup",
"items": [
{
"type": "SelectVariable",
"name": "VariableTest",
"caption": "Some Variable"
}
]
}
}
```

# ProgressBar

Benötigt: IP-Symcon >= 5.2
Erstellt einen Fortschrittsbalken.
Der Fortschrittsbalken kann durch die Verwendung von UpdateFormField dazu verwendet werden, den Benutzer über einen möglicherweise länger dauernden Vorgang zu informieren.
Es ist möglich entweder einen fortschreitenden Balken darzustellen oder einen unbestimmten Balken, welcher eine Ladeanimation anzeigt.




## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| caption (optional) | Sichtbare Beschriftung des Fortschrittsbalkens |
| current (optional) | (**default:** 0) Der aktuelle Wert des Fortschrittsbalkens - Wird der Balken fortschreitend dargestellt, so ist er ungefüllt, falls der aktuelle Wert dem minimalen Wert des Fortschrittsbalkens entspricht, komplett gefüllt, falls der aktuelle Wert dem maximalen Wert des Fortschrittsbalken entspricht und ansonsten entsprechend anteilig gefüllt. |
| indeterminate (optional) | (**default:** false) Bei true ist der Fortschrittsbalken unbestimmt und zeigt eine Animation an, ansonsten wird der Balken fortschreitend auf Basis von current, minimum und maximum dargestellt |
| maximum (optional) | (**default:** 100) Der maximale Wert des Fortschrittsbalkens |
| minimum (optional) | (**default:** 0) Der minimale Wert des Fortschrittsbalkens |
| name (optional) | Name des Fortschrittsbalken |
| type | ProgressBar |
| visible (optional) | (**default:** true) Bei true ist der Fortschrittsbalken sichtbar, ansonsten ist er unsichtbar |
| width (optional) | (**default:** 300px) Feste Breite des Fortschrittsbalkens in Pixel oder % als String, z.B. "40%" oder "250px" (ab IP-Symcon 5.3) |

## BEISPIEL

```
{
"type": "ProgressBar",
"name": "IndeterminateProgress",
"indeterminate": true,
"caption": "Laden..."
},
{
```

```
"type": "ProgressBar",
"name": "DeterminateProgress",
"minimum": 0,
"maximum": 50,
"current": 20,
"caption": "20 / 50"
}
```

# QrCode

Benötigt: IP-Symcon >= 7.0
Erstellt einen QR-Code.




## PARAMETER

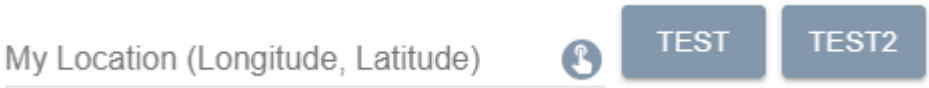
| PARAMETER | BESCHREIBUNG |
|---|---|
| caption (optional) | (**default:** "") Sichtbare Beschriftung |
| center (optional) | (**default:** false) Bei true wird der QR-Code mittig angezeigt, ansonsten linksbündig |
| description | (**default:** true) Bei true wird der im QR-Code enthaltene Text unter diesem angezeigt |
| name (optional) | Name des QR-Codes/der zu setzenden Eigenschaft |
| source | Die im QR-Code codierte Zeichenkette |
| type | QrCode |
| visible (optional) | (**default:** true) Bei true ist der QR-Code sichtbar, ansonsten ist er unsichtbar |

## BEISPIEL

```
{ "type": "QrCode", "name": "Beispiel", "caption": "Beispieltext",
"source": "Ein Beispieltext" }
```

# RowLayout

Benötigt: IP-Symcon >= 5.0
Erstellt eine Zeile, in welcher Formularfelder nebeneinander dargestellt werden.




## PARAMETER

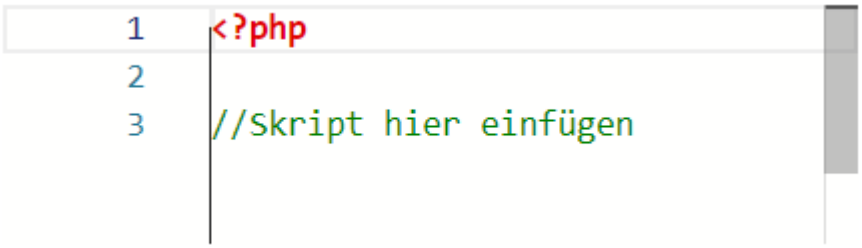
| PARAMETER | BESCHREIBUNG |
|---|---|
| items | Liste mit Konfigurationselementen innerhalb der Zeile (ab IP-Symcon 6.0 änderbar) |
| name (optional) | Name der Zeile |
| type | RowLayout |
| visible (optional) | (**default:** true) Bei true ist die Zeile sichtbar, ansonsten ist sie unsichtbar (ab IP-Symcon 5.2) |

## BEISPIEL

```
// Zeilenlayout
{
"type": "RowLayout",
"items": [
{
"type": "SelectLocation",
"name": "Location",
"caption": "My Location"
},{
"type": "Button",
"caption": "Test",
"onClick": "TM_Update($id);"
},{
"type": "Button",
"caption": "Test2",
"onClick": "TM_Update2($id);"
}
]
}
```

# ScriptEditor

Benötigt: IP-Symcon >= 6.0
Erstellt einen Skripteditor mit der Überschrift **caption**.
Falls im "elements"-Bereich erstellt, setzt der ScriptEditor eine Eigenschaft bei Übernahme auf den eingetragenen String.
Der Parameter **"name"** definiert dabei welche Eigenschaft gesetzt wird.




## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| caption (optional) | (**default:** "") sichtbare Beschriftung des Eingabefeldes |
| enabled (optional) | (**default:** true) Bei true kann das Eingabefeld verwendet werden, ansonsten wird es deaktiviert dargestellt |
| name (optional) | Name des Eingabefeldes/der zu setzenden Eigenschaft |
| type | ScriptEditor |
| value (optional) | (**default:** Platzhalterskript mit PHP-Tag) Der Wert des Eingabefeldes - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben |
| visible (optional) | (**default:** true) Bei true ist das Eingabefeld sichtbar, ansonsten ist es unsichtbar |
| width (optional) | (**default:** 100%) Feste Breite des Eingabefeldes in Pixel oder % als String, z.B. "40%" oder "250px", |
| rowCount (optional) | (**default:** 5) Anzahl der dargestellten Zeilen des Editors. Beinhaltet das Skript selbst mehr Zeilen, so wird ein Scrollbalken angezeigt |

## BEISPIEL

```
{ "type": "ScriptEditor", "name": "Script" }
```

# Select

Erstellt ein DropDown-Menü mit der Überschrift **caption**.
Falls im "elements"-Bereich erstellt, setzt das DropDown-Menü eine Eigenschaft bei Übernahme auf den ausgewählten Wert **value**.
Der Parameter **"name"** definiert dabei welche Eigenschaft gesetzt wird.



Bei MultiSelect ist darauf zu achten, dass alle Value-Arrays die gleichen Namen beinhalten



Wird Select innerhalb von Listen verwendet, gibt es kein MultiSelect.




## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung des DropDown-Menüs |
| download (optional) | (**default:** "") Ist dieser Parameter nicht leer und beinhaltet die Ausgabe des onChange-Skriptes eine Data-URL, so wird die Rückgabe als Textdatei mit dem Namen das download-Parameters heruntergeladen. (ab IP-Symcon 7.0) |
| enabled (optional) | (**default:** true) Bei true kann das DropDown-Menü verwendet werden, ansonsten wird es deaktiviert dargestellt (ab IP-Symcon 5.2) |
| link (optional) | (**default:** false) Ist dieser Parameter true, dann wird die Ausgabe des onChange-Skriptes als Link geöffnet. Ist er false, so wird die Ausgabe als Dialog im Konfigurationsformular angezeigt. (ab IP-Symcon 6.0) |
| name (optional) | Name des DropDown-Menüs/der zu setzenden Eigenschaft |
| options | Array, welches alle Auswahlmöglichkeiten beinhaltet |
| onChange (optional) | (**default:** "") Skript, welches bei einer Veränderung des Dropdown-Wertes ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden (Arrays werden ab Version 6.0 unterstützt). Es hat die gleichen Eigenschaften wie onClick des Button (ab IP-Symcon 5.2) |
| type | Select |
| value (optional) | (**default:** erste definierte Option) Der Wert des DropDown-Menüs - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben (ab IP-Symcon 5.2) |
| visible (optional) | (**default:** true) Bei true ist das DropDown-Menü sichtbar, ansonsten ist es unsichtbar (ab IP-Symcon 5.2) |
| width (optional) | (**default:** 300px) Feste Breite des Dropdown-Menüs in Pixel oder % als String, z.B. "40%" oder "250px" (ab IP-Symcon 5.3) |

## PARAMETER FÜR OPTIONS

| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Beschriftung einer Auswahlmöglichkeit |
| ~~label~~ | ~~Beschriftung einer Auswahlmöglichkeit~~ *(veraltet)* |
| value | Wert der bei Auswahl übergeben wird. Kann auch ein weiteres Array beinhalten, welches mehreren Eigenschaften die Werte übergibt. |

## BEISPIEL

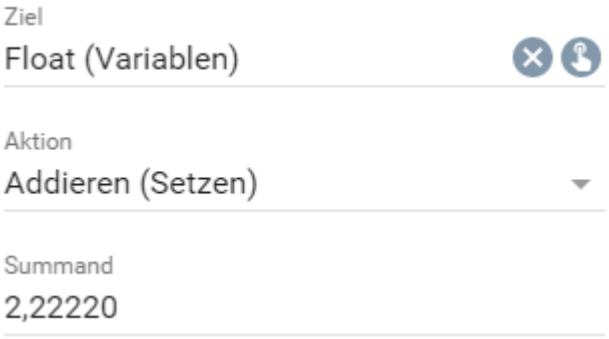
```
//Einfache Valueübergabe
//Eigenschaft "DeviceType" wird auf 0..3 gesetzt
//DropDown-Menü hat 4 Einträge
{ "type": "Select", "name": "DeviceType", "caption": "Einheit",
"options": [
{ "caption": "Bit (1Bit)", "value": 0 },
{ "caption": "Bits (2Bit)", "value": 1 },
{ "caption": "Bits (4Bit)", "value": 2 },
```

```
{ "caption": "Byte (8Bit)", "value": 3 }
]
}

//Mehrfache Valueübergabe (MultiSelect)
//"name" des Select (hier: "OutputDevice") muss gesetzt sein, wird
jedoch nicht zum Setzen der Werte genutzt.
//Eigenschaften "DeviceName", "DeviceChannels" und "DeviceNum"
werden gesetzt.
{ "type": "Select", "name": "OutputDevice", "caption":
"Ausgabegerät",
"options": [
{"caption" : "No sound",
"value": [
{"name": "DeviceName", "value": "No sound"},
{"name": "DeviceChannels", "value": 0},
{"name": "DeviceNum", "value": 0}
]
},
{"caption" : "Kopfhörer",
"value": [
{"name": "DeviceName", "value": "Kopfhörer"},
{"name": "DeviceChannels", "value": 2},
{"name": "DeviceNum", "value": 1}
]
},
{"caption" : "Lautsprecher",
"value": [
{"name": "DeviceName", "value": "Stereolautsprecher"},
{"name": "DeviceChannels", "value": 2},
{"name": "DeviceNum", "value": 2}
]
}
]
}
```

# SelectAction

Benötigt: IP-Symcon >= 6.0
Erstellt eine Eingabe für eine Aktion .




Falls im "elements"-Bereich erstellt, wird bei Übernahme die Eigenschaft **name** auf ein JSON-codiertes Objekt gesetzt, welches folgende Parameter enthält:

| PARAMETER | BESCHREIBUNG |
|---|---|
| actionID | ID der Aktion |
| parameters | Parameter der Aktion |

## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| caption (optional) | (**default:**: "") Sichtbare Beschriftung (ab IP-Symcon 7.1) |
| enabled (optional) | (**default:** true) Bei true kann die Auswahl verwendet werden, ansonsten wird sie deaktiviert dargestellt |
| environment (optional) | (**default:** "Default") Legt die Umgebung der Aktionsauswahl fest, hierüber kann bestimmt werden, welche Arten von Aktionen zur Auswahl stehen, siehe auch Aktionen |
| highestPriorityOnly (optional) | (**default:** false) Bei false verfügt das Element eine Auswahl über alle passenden Aktionen, ansonsten wird ohne Auswahl automatisch die Aktion mit der höchsten Priorität ausgewählt |
| includeDefaultEnvironment (optional) | (**default:** true) Bei false werden nur Aktionen zur Auswahl angeboten, welche die Umgebung *environment* unterstützen, bei true werden zusätzlich alle Aktionen der Umgebung "Default" angeboten |
| name (optional) | Name der Auswahl/der zu setzenden Eigenschaft |
| targetID (optional) | (**default:** -2) Bei -2 verfügt das Element über eine Zielauswahl, ansonsten ist das Ziel mit dieser ID fest als Ziel gesetzt, wobei -1 für allgemeine Aktionen steht |
| type | SelectAction |
| value (optional) | (**default:** keine Aktion ausgewählt) Der Wert der Auswahl - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben |
| visible (optional) | (**default:** true) Bei true ist die Auswahl sichtbar, ansonsten ist sie unsichtbar |
| width (optional) | (**default:** 300px) Feste Breite des Eingabefeldes in Pixel oder % als String, z.B. "40%" oder "250px" |
| saveTarget (optional) | (**default:** true) Bei true wird das Ziel als TARGET als Teil der parameters gespeichert |
| saveEnvironment (optional) | (**default:** true) Bei true wird die Umgebung als ENVIRONMENT als Teil der parameters gespeichert |
| saveParent (optional) | (**default:** true) Bei true wird die InstanzID als PARENT als Teil der parameters gespeichert |

## BEISPIEL 1: ZIEL, UMGEBUNG UND INSTANZID WERDEN KONSTANT ALS TEIL DER EIGENSCHAFT GESPEICHERT

Standardmäßig werden alle benötigten Parameter als Rückgabe von SelectAction abgespeichert. Dadurch kann die Aktion am einfachsten ausgeführt werden.

```
{ "type": "SelectAction", "name": "PropertyAction", "targetID":
12345 }
// PHP-Code im Modul
$action = json_decode($this->ReadPropertyString('PropertyAction'),
true);
IPS_RunAction($action['actionID'], $action['parameters']);
```

## BEISPIEL 2: ZIEL, UMGEBUNG UND INSTANZID WERDEN DYNAMISCH GESETZT

Speichert man die Zusatzparameter nicht als Rückgabe ab, so müssen diese manuell hinzugefügt werden. Hierdurch können beispielsweise Aktionen zwischen Instanzen ausgetauscht werden und es werden keine redundanten Daten abgespeichert.

```
{ "type": "SelectAction", "name": "PropertyAction", "targetID":
12345, "saveTarget": false, "saveEnvironment": false, "saveParent":
false }
// PHP-Code im Modul
$action = json_decode($this->ReadPropertyString('PropertyAction'),
true);
$parameters = $action['parameters'];
$parameters['TARGET'] = 12345;
$parameters['ENVIRONMENT'] = 'Default';
$parameters['PARENT'] = $this->InstanceID;
IPS_RunAction($action['actionID'], $parameters);
```

# SelectCategory

Erstellt einen Auswahldialog mit der Beschriftung **caption**.
Falls im "elements"-Bereich erstellt, wird bei Übernahme die Eigenschaft **name** auf die ID der ausgewählten Kategorie gesetzt.




## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung |
| download (optional) | (**default:** "") Ist dieser Parameter nicht leer und beinhaltet die Ausgabe des onChange-Skriptes eine Data-URL, so wird die Rückgabe als Textdatei mit dem Namen das download-Parameters heruntergeladen. (ab IP-Symcon 7.0) |
| enabled (optional) | (**default:** true) Bei true kann die Kategorieauswahl verwendet werden, ansonsten wird sie deaktiviert dargestellt (ab IP-Symcon 5.2) |
| link (optional) | (**default:** false) Ist dieser Parameter true, dann wird die Ausgabe des onChange-Skriptes als Link geöffnet. Ist er false, so wird die Ausgabe als Dialog im Konfigurationsformular angezeigt. (ab IP-Symcon 6.0) |
| name (optional) | Name der Kategorieauswahl/der zu setzenden Eigenschaft |
| onChange (optional) | (**default:** "") Skript, welches bei einer Veränderung des Wertes der Kategorieauswahl ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden (Arrays werden ab Version 6.0 unterstützt). Es hat die gleichen Eigenschaften wie onClick des Button (ab IP-Symcon 5.2) |
| type | SelectCategory |
| value (optional) | (**default:** 1) Der Wert der Kategorieauswahl - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben (ab IP-Symcon 5.2) |
| visible (optional) | (**default:** true) Bei true ist die Kategorieauswahl sichtbar, ansonsten ist sie unsichtbar (ab IP-Symcon 5.2) |
| width (optional) | (**default:** 300px) Feste Breite der Kategorieauswahl in Pixel oder % als String, z.B. "40%" oder "250px" (ab IP-Symcon 5.3) |

## BEISPIEL

```
{ "type": "SelectCategory", "name": "PropertyCategoryID", "caption":
"Target" }
```

# SelectColor

Benötigt: IP-Symcon >= 4.2
Erstellt einen Auswahldialog mit der Beschriftung **caption**.
Falls im "elements"-Bereich erstellt, wird bei Übernahme die Eigenschaft **name** auf den Wert der ausgewählten Farbe als Hex-String gesetzt.
Wird der Wert gelöscht, wird dieser auf Transparent(Wert = -1) gesetzt.




## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| allowTransparent (optional) | (**default:** true) Wenn gesetzt, so kann die Auswahl gelöscht und so auf "Transparent" gesetzt werden. Ansonsten kann der Wert nicht gelöscht und somit nicht auf "Transparent" gesetzt werden. |
| caption | Sichtbare Beschriftung |
| download (optional) | (**default:** "") Ist dieser Parameter nicht leer und beinhaltet die Ausgabe des onChange-Skriptes eine Data-URL, so wird die Rückgabe als Textdatei mit dem Namen das download-Parameters heruntergeladen. (ab IP-Symcon 7.0) |
| enabled (optional) | (**default:** true) Bei true kann die Auswahl verwendet werden, ansonsten wird sie deaktiviert dargestellt (ab IP-Symcon 5.2) |
| link (optional) | (**default:** false) Ist dieser Parameter true, dann wird die Ausgabe des onChange-Skriptes als Link geöffnet. Ist er false, so wird die Ausgabe als Dialog im Konfigurationsformular angezeigt. (ab IP-Symcon 6.0) |
| name (optional) | Name der Auswahl/der zu setzenden Eigenschaft |
| onChange (optional) | (**default:** "") Skript, welches bei einer Änderung der Farbe ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden (Arrays werden ab Version 6.0 unterstützt). Es hat die gleichen Eigenschaften wie onClick des Button (ab IP-Symcon 5.2) |
| type | SelectColor |
| value (optional) | (**default:** -1 falls allowTransparent, sonst 0) Der Wert der Auswahl - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben (ab IP-Symcon 5.2) |
| visible (optional) | (**default:** true) Bei true ist die Auswahl sichtbar, ansonsten ist sie unsichtbar (ab IP-Symcon 5.2) |
| width (optional) | (**default:** 300px) Feste Breite der Farbauswahl in Pixel oder % als String, z.B. "40%" oder "250px" (ab IP-Symcon 5.3) |

## BEISPIEL

```
{ "type": "SelectColor", "name": "HexColorEigenschaft", "caption":
"Color" }
```

# SelectCondition

Benötigt: IP-Symcon >= 6.1
Erstellt eine Eingabe für Bedingungen.




Falls im "elements"-Bereich erstellt, wird bei Übernahme die Eigenschaft **name** auf eine JSON-codiertes Liste gesetzt, wobei jedes Element ein Objekt mit folgenden Parametern darstellt:

| PARAMETER | BESCHREIBUNG |
|---|---|
| id | ID der Bedingung |
| parentID | ID der übergeordneten Bedingung |
| operation | Operation zur Verknüpfung der einzelnen Regeln (0: AND, 1: OR, 2: NAND (noch nicht implementiert), 3: NOR (noch nicht implementiert)) |
| rules | Objekt welches die Regeln der Bedingung beschreibt |

### RULES

| PARAMETER | BESCHREIBUNG |
|---|---|
| variable | Liste mit den Variablenregeln |
| date | Liste mit den Datumsregeln |
| time | Liste mit den Uhrzeitregeln |
| dayOfTheWeek | Liste mit den Wochentagsregeln |

### DARSTELLUNG EINER REGEL

| PARAMETER | BESCHREIBUNG |
| --- | --- |
| id | Eindeutige ID der Regel innerhalb der Regelliste |
| comparison | Vergleichsoperator mit welchem der aktuelle Wert mit dem Vergleichswert verglichen wird (0: =, 1: ≠, 2: >, 3: ≥, 4: <, 5: ≤) |
| value | Der Vergleichswert, der Typ hängt von der Regelart ab<br>*Variable*: Falls "type" = 0 steht hier der absolute Vergleichswert passend zum Typ der Variable, bei "type" = 1 steht hier eine VariablenID mit deren Wert verglichen wird<br>*Datum*: Der Wert wird als Objekt mit den Feldern "year", "month" und "day" abgespeichert, analog zur Darstellung von IPS_SetEventConditionDateRule<br>*Uhrzeit*: Der Wert wird als Objekt mit den Feldern "hour", "minute" und "second" abgespeichert, analog zur Darstellung von IPS_SetEventConditionTimeRule<br>*Wochentag*: Der Vergleichstag wird als Zahl gespeichert (1: Montag, 2: Dienstag, 3: Mittwoch, 4: Donnerstag, 5: Freitag, 6: Samstag, 7: Sonntag) |
| variableID (nur bei Variablenregeln) | Die ID der Variable, welche den aktuellen Wert beinhaltet |
| type (nur bei Variablenregeln) | Bei 0 wird die Variable mit einem absoluten Wert verglichen, bei 1 mit dem Wert einer anderen Variablen |



Anstatt eine konfigurierte Bedingung selbstständig auszuwerten sollte die Funktion IPS_IsConditionPassing verwendet werden. Dort kann der Wert des SelectCondition direkt als Parameter verwendet werden.



Falls die abgespeicherte Bedingung verschachtelte Bedingungen beinhaltet oder "multi" auf false steht und die Bedingung mehrere Regeln beinhaltet, wird die Regel als "Komplex" angezeigt und kann nicht bearbeitet werden.

## PARAMETER

| PARAMETER | BESCHREIBUNG |
| --- | --- |
| caption (optional) | (**default:**: "") Sichtbare Beschriftung (ab IP-Symcon 7.1) |
| enabled (optional) | (**default:** true) Bei true kann die Auswahl verwendet werden, ansonsten wird sie deaktiviert dargestellt |
| multi (optional) | (**default:** false) Bei false wird vom Element eine einzelne Regel definiert, bei true eine gesamte Bedingung mit beliebig vielen Regeln |
| name (optional) | Name der Auswahl/der zu setzenden Eigenschaft |
| type | SelectCondition |
| value (optional) | (**default:** keine Bedingung) Der Wert der Auswahl - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben |
| visible (optional) | (**default:** true) Bei true ist die Auswahl sichtbar, ansonsten ist sie unsichtbar |
| width (optional) | (**default:** 300px, falls multi = false, sonst 100%) Feste Breite des Eingabefeldes in Pixel oder % als String, z.B. "40%" oder "250px" |

## BEISPIEL

```
{ "type": "SelectCondition", "name": "PropertyCondition", "multi":
true }
```

# SelectDate

Benötigt: IP-Symcon >= 5.0
Erstellt eine Eingabe für ein Datum mit der Beschriftung **caption**.




Falls im "elements"-Bereich erstellt, wird bei Übernahme die Eigenschaft **name** auf ein JSON-codiertes Objekt gesetzt, welches folgende Parameter enthält:

| PARAMETER | BESCHREIBUNG |
|---|---|
| day | Tag des Datums |
| month | Monat des Datums |
| year | Jahr des Datums |

Wurde kein Zeitpunkt ausgewählt, so stehen alle Parameter auf 0.

## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung |
| enabled (optional) | (**default:** true) Bei true kann die Auswahl verwendet werden, ansonsten wird sie deaktiviert dargestellt (ab IP-Symcon 5.2) |
| name (optional) | Name der Auswahl/der zu setzenden Eigenschaft |
| type | SelectDate |
| value (optional) | (**default:** kein Zeitpunkt ausgewählt) Der Wert der Auswahl - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben (ab IP-Symcon 5.2) |
| visible (optional) | (**default:** true) Bei true ist die Auswahl sichtbar, ansonsten ist sie unsichtbar (ab IP-Symcon 5.2) |
| width (optional) | (**default:** 300px) Feste Breite des Eingabefeldes in Pixel oder % als String, z.B. "40%" oder "250px" (ab IP-Symcon 5.3) |

## BEISPIEL

```
{ "type": "SelectDate", "name": "PropertyDate", "caption": "Date" }
```

# SelectDateTime

Benötigt: IP-Symcon >= 5.0
Erstellt eine Eingabe für ein Datum und eine Uhrzeit mit der Beschriftung **caption**.




Falls im "elements"-Bereich erstellt, wird bei Übernahme die Eigenschaft **name** auf ein JSON-codiertes Objekt gesetzt, welches folgende Parameter enthält:

| PARAMETER | BESCHREIBUNG |
|---|---|
| day | Tag des Datums |
| hour | Stunden der Uhrzeit |
| minute | Minuten der Uhrzeit |
| month | Monat des Datums |
| second | Sekunden der Uhrzeit |
| year | Jahr des Datums |

Wurde kein Zeitpunkt ausgewählt, so stehen alle Parameter auf 0.

## PARAMETER

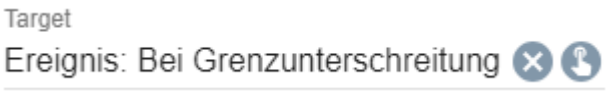
| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung |
| enabled (optional) | (**default:** true) Bei true kann die Auswahl verwendet werden, ansonsten wird sie deaktiviert dargestellt (ab IP-Symcon 5.2) |
| name (optional) | Name der Auswahl/der zu setzenden Eigenschaft |
| type | SelectDateTime |
| value (optional) | (**default:** kein Zeitpunkt ausgewählt) Der Wert der Auswahl - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben (ab IP-Symcon 5.2) |
| visible (optional) | (**default:** true) Bei true ist die Auswahl sichtbar, ansonsten ist sie unsichtbar (ab IP-Symcon 5.2) |
| width (optional) | (**default:** 300px) Feste Breite des Eingabefeldes in Pixel oder % als String, z.B. "40%" oder "250px" (ab IP-Symcon 5.3) |

## BEISPIEL

```
{ "type": "SelectDateTime", "name": "PropertyDateTime", "caption":
"Date + Time" }
```

# SelectEvent

Erstellt einen Auswahldialog mit der Beschriftung **caption**.
Falls im "elements"-Bereich erstellt, wird bei Übernahme die Eigenschaft **name** auf die ID des ausgewählten Ereignisses gesetzt.




## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung |
| download (optional) | (**default:** "") Ist dieser Parameter nicht leer und beinhaltet die Ausgabe des onChange-Skriptes eine Data-URL, so wird die Rückgabe als Textdatei mit dem Namen das download-Parameters heruntergeladen. (ab IP-Symcon 7.0) |
| enabled (optional) | (**default:** true) Bei true kann die Ereignisauswahl verwendet werden, ansonsten wird sie deaktiviert dargestellt (ab IP-Symcon 5.2) |
| link (optional) | (**default:** false) Ist dieser Parameter true, dann wird die Ausgabe des onChange-Skriptes als Link geöffnet. Ist er false, so wird die Ausgabe als Dialog im Konfigurationsformular angezeigt. (ab IP-Symcon 6.0) |
| name (optional) | Name der Ereignisauswahl/der zu setzenden Eigenschaft |
| onChange (optional) | (**default:** "") Skript, welches bei einer Veränderung des Wertes der Ereignisauswahl ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden (Arrays werden ab Version 6.0 unterstützt). Es hat die gleichen Eigenschaften wie onClick des Button (ab IP-Symcon 5.2) |
| type | SelectEvent |
| value (optional) | (**default:** 0) Der Wert der Ereignisauswahl - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben (ab IP-Symcon 5.2) |
| visible (optional) | (**default:** true) Bei true ist die Ereignisauswahl sichtbar, ansonsten ist sie unsichtbar (ab IP-Symcon 5.2) |
| width (optional) | (**default:** 300px) Feste Breite der Ereignisauswahl in Pixel oder % als String, z.B. "40%" oder "250px" (ab IP-Symcon 5.3) |

## BEISPIEL

```
{ "type": "SelectEvent", "name": "PropertyEventID", "caption":
"Target" }
```

# SelectFile

Benötigt: IP-Symcon >= 4.2
Erstellt einen Auswahldialog mit der Beschriftung **caption**.
Falls im "elements"-Bereich erstellt, wird bei Übernahme die Eigenschaft **name** auf den Base64 kodierten Wert der ausgewählten Datei gesetzt. Dies bedeutet, dass der Inhalt der ausgewählten Datei als Base64 kodierter String gespeichert wird.




## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung |
| download (optional) | (**default:** "") Ist dieser Parameter nicht leer und beinhaltet die Ausgabe des onChange-Skriptes eine Data-URL, so wird die Rückgabe als Textdatei mit dem Namen das download-Parameters heruntergeladen. (ab IP-Symcon 7.0) |
| enabled (optional) | (**default:** true) Bei true kann die Dateiauswahl verwendet werden, ansonsten wird sie deaktiviert dargestellt (ab IP-Symcon 5.2) |
| extensions (optional) | Erlaubte Erweiterungen der Datei. Durch Komma getrennt und mit Punkt eingeleitet Bsp: .jpg,.gif,.txt |
| link (optional) | (**default:** false) Ist dieser Parameter true, dann wird die Ausgabe des onChange-Skriptes als Link geöffnet. Ist er false, so wird die Ausgabe als Dialog im Konfigurationsformular angezeigt. (ab IP-Symcon 6.0) |
| name (optional) | Name der Dateiauswahl/der zu setzenden String-Eigenschaft |
| onChange (optional) | (**default:** "") Skript, welches bei einer Veränderung des Wertes der Dateiauswahl ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden (Arrays werden ab Version 6.0 unterstützt). Es hat die gleichen Eigenschaften wie onClick des Button (ab IP-Symcon 5.2) |
| type | SelectFile |
| value (optional) | (**default:** 0) Der Wert der Dateiauswahl - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben (ab IP-Symcon 5.2) |
| visible (optional) | (**default:** true) Bei true ist die Dateiauswahl sichtbar, ansonsten ist sie unsichtbar (ab IP-Symcon 5.2) |
| width (optional) | (**default:** 300px) Feste Breite der Dateiauswahl in Pixel oder % als String, z.B. "40%" oder "250px" (ab IP-Symcon 5.3) |

## BEISPIEL

```
{ "type": "SelectFile", "name": "PropertyStringFile", "caption":
"Target", "extensions": ".jpg,.gif,.txt" }
```

# SelectIcon

Erstellt einen Auswahldialog mit der Beschriftung **caption**.
Falls im "elements"-Bereich erstellt, wird bei Übernahme die Eigenschaft **name** auf den Namen des ausgewählten Icons gesetzt.
Wird der Wert gelöscht, wird dieser auf einen leeren String gesetzt.




## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung |
| download (optional) | (**default:** "") Ist dieser Parameter nicht leer und beinhaltet die Ausgabe des onChange-Skriptes eine Data-URL, so wird die Rückgabe als Textdatei mit dem Namen das download-Parameters heruntergeladen. (ab IP-Symcon 7.0) |
| enabled (optional) | (**default:** true) Bei true kann die Auswahl verwendet werden, ansonsten wird sie deaktiviert dargestellt |
| link (optional) | (**default:** false) Ist dieser Parameter true, dann wird die Ausgabe des onChange-Skriptes als Link geöffnet. Ist er false, so wird die Ausgabe als Dialog im Konfigurationsformular angezeigt. |
| name (optional) | Name der Auswahl/der zu setzenden Eigenschaft |
| onChange (optional) | (**default:** "") Skript, welches bei einer Änderung des Icons ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden (Arrays werden ab Version 6.0 unterstützt). Es hat die gleichen Eigenschaften wie onClick des Button |
| type | SelectIcon |
| value (optional) | (**default:** "") Der Wert der Auswahl - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben (ab IP-Symcon 5.2) |
| visible (optional) | (**default:** true) Bei true ist die Auswahl sichtbar, ansonsten ist sie unsichtbar (ab IP-Symcon 5.2) |
| width (optional) | (**default:** 300px) Feste Breite der Farbauswahl in Pixel oder % als String, z.B. "40%" oder "250px" |

## BEISPIEL

```
{ "type": "SelectIcon", "name": "IconEigenschaft", "caption": "Icon"
}
```

# SelectInstance

Erstellt einen Auswahldialog mit der Beschriftung **caption**.
Falls im "elements"-Bereich erstellt, wird bei Übernahme die Eigenschaft **name** auf die ID der ausgewählten Instanz gesetzt.




## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung |
| download (optional) | (**default:** "") Ist dieser Parameter nicht leer und beinhaltet die Ausgabe des onChange-Skriptes eine Data-URL, so wird die Rückgabe als Textdatei mit dem Namen das download-Parameters heruntergeladen. (ab IP-Symcon 7.0) |
| enabled (optional) | (**default:** true) Bei true kann die Instanzauswahl verwendet werden, ansonsten wird sie deaktiviert dargestellt (ab IP-Symcon 5.2) |
| link (optional) | (**default:** false) Ist dieser Parameter true, dann wird die Ausgabe des onChange-Skriptes als Link geöffnet. Ist er false, so wird die Ausgabe als Dialog im Konfigurationsformular angezeigt. (ab IP-Symcon 6.0) |
| name (optional) | Name der Instanzauswahl/der zu setzenden Eigenschaft |
| onChange (optional) | (**default:** "") Skript, welches bei einer Veränderung des Wertes der Instanzauswahl ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden (Arrays werden ab Version 6.0 unterstützt). Es hat die gleichen Eigenschaften wie onClick des Button (ab IP-Symcon 5.2) |
| type | SelectInstance |
| validModules (optional) | (**default:** []) Die Auswahl wird auf Instanzen eingeschränkt, welche eine der definierten ModulIDs haben. Wenn die Liste leer ist sind alle Instanz erlaubt. Entspricht die ausgewählte Instanz keiner der ModulIDs so wird ein Fehler angezeigt und Änderungen können nicht übernommen werden. (ab IP-Symcon 5.5) |
| value (optional) | (**default:** 1) Der Wert der Instanzauswahl - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben (ab IP-Symcon 5.2) |
| visible (optional) | (**default:** true) Bei true ist die Instanzauswahl sichtbar, ansonsten ist sie unsichtbar (ab IP-Symcon 5.2) |
| width (optional) | (**default:** 300px) Feste Breite der Instanzauswahl in Pixel oder % als String, z.B. "40%" oder "250px" (ab IP-Symcon 5.3) |

## BEISPIEL

```
{ "type": "SelectInstance", "name": "PropertyInstanceID", "caption":
```

```
  "Target" }
```

# SelectLink

Erstellt einen Auswahldialog mit der Beschriftung **caption**.
Falls im "elements"-Bereich erstellt, wird bei Übernahme die Eigenschaft **name** auf die ID des ausgewählten Links gesetzt.




## PARAMETER

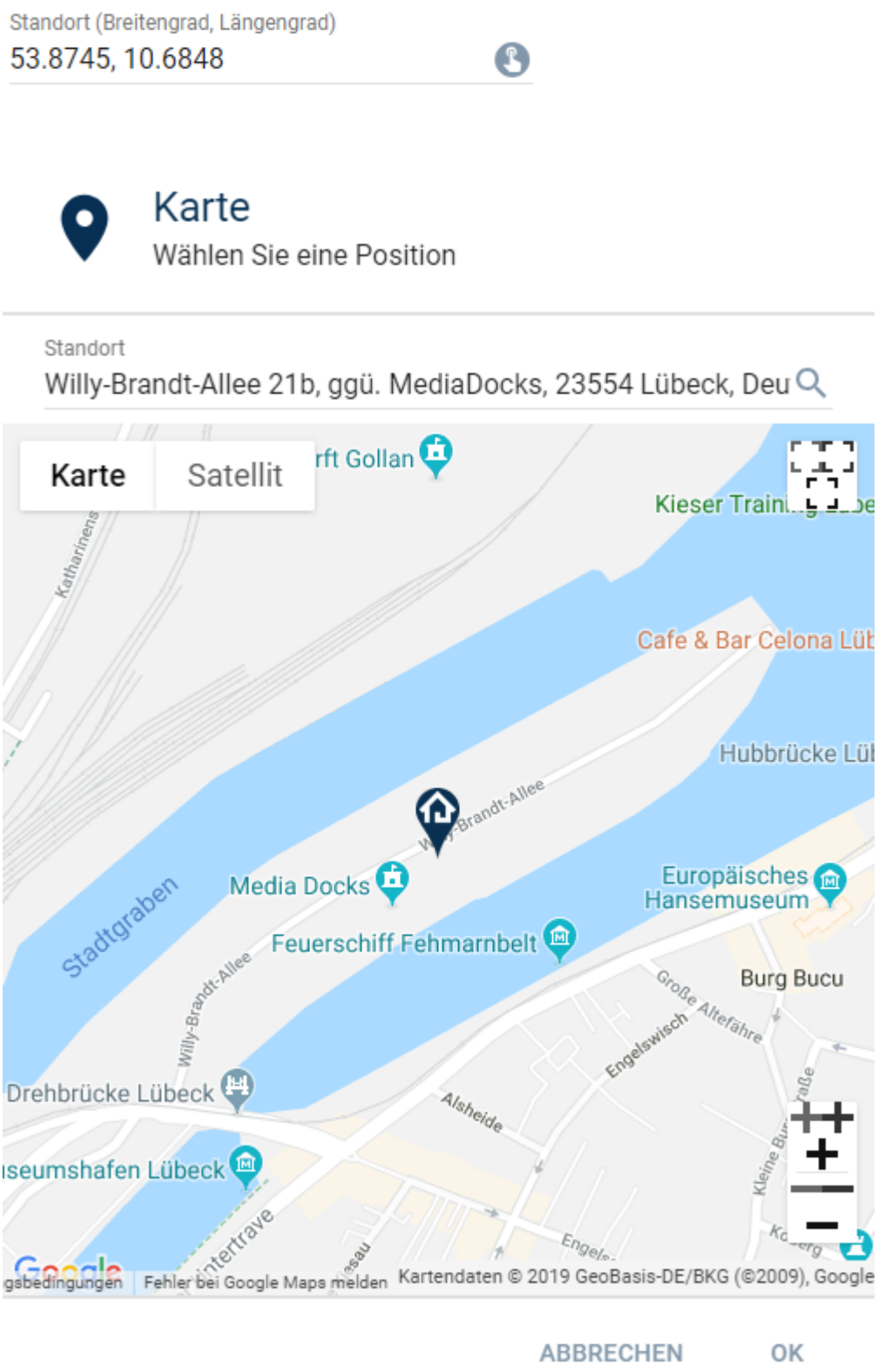
| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung |
| download (optional) | (**default:** "") Ist dieser Parameter nicht leer und beinhaltet die Ausgabe des onChange-Skriptes eine Data-URL, so wird die Rückgabe als Textdatei mit dem Namen das download-Parameters heruntergeladen. (ab IP-Symcon 7.0) |
| enabled (optional) | (**default:** true) Bei true kann die Linkauswahl verwendet werden, ansonsten wird sie deaktiviert dargestellt (ab IP-Symcon 5.2) |
| link (optional) | (**default:** false) Ist dieser Parameter true, dann wird die Ausgabe des onChange-Skriptes als Link geöffnet. Ist er false, so wird die Ausgabe als Dialog im Konfigurationsformular angezeigt. (ab IP-Symcon 6.0) |
| name (optional) | Name der Linkauswahl/der zu setzenden Eigenschaft |
| onChange (optional) | (**default:** "") Skript, welches bei einer Veränderung des Wertes der Linkauswahl ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden (Arrays werden ab Version 6.0 unterstützt). Es hat die gleichen Eigenschaften wie onClick des Button (ab IP-Symcon 5.2) |
| type | SelectLink |
| value (optional) | (**default:** 1) Der Wert der Linkauswahl - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben (ab IP-Symcon 5.2) |
| visible (optional) | (**default:** true) Bei true ist die Linkauswahl sichtbar, ansonsten ist sie unsichtbar (ab IP-Symcon 5.2) |
| width (optional) | (**default:** 300px) Feste Breite der Linkauswahl in Pixel oder % als String, z.B. "40%" oder "250px" (ab IP-Symcon 5.3) |

## BEISPIEL

```
{ "type": "SelectLink", "name": "PropertyLinkID", "caption":
"Target" }
```

# SelectLocation

Benötigt: IP-Symcon >= 5.0
Erstellt eine Eingabe für eine Position mit der Beschriftung **caption**. Hier kann die Position bei Internetzugriff per Karte ausgewählt werden. Alternativ können Längen- und Breitengrad auch manuell angegeben werden.




Falls im "elements"-Bereich erstellt, wird bei Übernahme die Eigenschaft **name** auf ein JSON-codiertes Objekt gesetzt, welches folgende Parameter enthält:

| PARAMETER | BESCHREIBUNG |
| --- | --- |
| latitude | Breitengrad der Position |
| longitude | Längengrad der Position |

Wurde keine Position ausgewählt, so stehen alle Parameter auf 0.

## PARAMETER

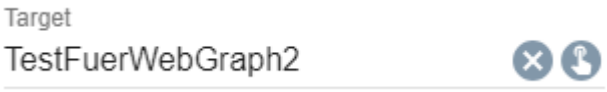
| PARAMETER | BESCHREIBUNG |
| --- | --- |
| caption | Sichtbare Beschriftung |
| enabled (optional) | (**default:** true) Bei true kann die Auswahl verwendet werden, ansonsten wird sie deaktiviert dargestellt (ab IP-Symcon 5.2) |
| name (optional) | Name der Auswahl/der zu setzenden Eigenschaft |
| type | SelectLocation |
| value (optional) | (**default:** keine Position ausgewählt) Der Wert der Auswahl - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben (ab IP-Symcon 5.2) |
| visible (optional) | (**default:** true) Bei true ist die Auswahl sichtbar, ansonsten ist sie unsichtbar (ab IP-Symcon 5.2) |
| width (optional) | (**default:** 300px) Feste Breite des Eingabefeldes in Pixel oder % als String, z.B. "40%" oder "250px" (ab IP-Symcon 5.3) |

## BEISPIEL

```
{ "type": "SelectLocation", "name": "Location", "caption": "Home" }
```

# SelectMedia

Erstellt einen Auswahldialog mit der Beschriftung **caption**.
Falls im "elements"-Bereich erstellt, wird bei Übernahme die Eigenschaft **name** auf die ID des ausgewählten Mediums gesetzt.




## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung |
| download (optional) | (**default:** "") Ist dieser Parameter nicht leer und beinhaltet die Ausgabe des onChange-Skriptes eine Data-URL, so wird die Rückgabe als Textdatei mit dem Namen das download-Parameters heruntergeladen. (ab IP-Symcon 7.0) |
| enabled (optional) | (**default:** true) Bei true kann die Medienauswahl verwendet werden, ansonsten wird sie deaktiviert dargestellt (ab IP-Symcon 5.2) |
| link (optional) | (**default:** false) Ist dieser Parameter true, dann wird die Ausgabe des onChange-Skriptes als Link geöffnet. Ist er false, so wird die Ausgabe als Dialog im Konfigurationsformular angezeigt. (ab IP-Symcon 6.0) |
| name (optional) | Name der Medienauswahl/der zu setzenden Eigenschaft |
| onChange (optional) | (**default:** "") Skript, welches bei einer Veränderung des Wertes der Medienauswahl ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden (Arrays werden ab Version 6.0 unterstützt). Es hat die gleichen Eigenschaften wie onClick des Button (ab IP-Symcon 5.2) |
| type | SelectMedia |
| value (optional) | (**default:** 1) Der Wert der Medienauswahl - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben (ab IP-Symcon 5.2) |
| visible (optional) | (**default:** true) Bei true ist die Medienauswahl sichtbar, ansonsten ist sie unsichtbar (ab IP-Symcon 5.2) |
| width (optional) | (**default:** 300px) Feste Breite der Medienauwahl in Pixel oder % als String, z.B. "40%" oder "250px" (ab IP-Symcon 5.3) |

## BEISPIEL

```
{ "type": "SelectMedia", "name": "PropertyMediaID", "caption":
"Target" }
```

# SelectModule

Benötigt: IP-Symcon >= 5.5
Erstellt einen Auswahldialog mit der Beschriftung **caption**.
Falls im "elements"-Bereich erstellt, wird bei Übernahme die Eigenschaft **name** auf die ID der ausgewählten Instanz gesetzt.




## PARAMETER

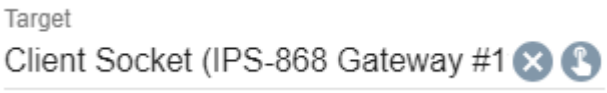
| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung |
| download (optional) | (**default:** "") Ist dieser Parameter nicht leer und beinhaltet die Ausgabe des onChange-Skriptes eine Data-URL, so wird die Rückgabe als Textdatei mit dem Namen das download-Parameters heruntergeladen. (ab IP-Symcon 7.0) |
| enabled (optional) | (**default:** true) Bei true kann die Modulauswahl verwendet werden, ansonsten wird sie deaktiviert dargestellt |
| link (optional) | (**default:** false) Ist dieser Parameter true, dann wird die Ausgabe des onChange-Skriptes als Link geöffnet. Ist er false, so wird die Ausgabe als Dialog im Konfigurationsformular angezeigt. (ab IP-Symcon 6.0) |
| moduleID | Die ID des Moduls, dessen Instanzen zur Auswahl angeboten werden sollen |
| name (optional) | Name der Modulauswahl/der zu setzenden Eigenschaft |
| onChange (optional) | (**default:** "") Skript, welches bei einer Veränderung des Wertes der Modulauswahl ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden (Arrays werden ab Version 6.0 unterstützt). Es hat die gleichen Eigenschaften wie onClick des Button |
| type | SelectModule |
| value (optional) | (**default:** 1) Der Wert der Modulauswahl - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben |
| visible (optional) | (**default:** true) Bei true ist die Modulauswahl sichtbar, ansonsten ist sie unsichtbar |
| width (optional) | (**default:** 300px) Feste Breite der Modulauswahl in Pixel oder % als String, z.B. "40%" oder "250px" |

## BEISPIEL

```
{ "type": "SelectModule", "name": "PropertyModuleInstanceID",
"caption": "VoIP Modul" , "moduleID":
"{A4224A63-49EA-445F-8422-22EF99D8F624}"}
```

# SelectObject

Erstellt einen Auswahldialog mit der Beschriftung **caption**.
Falls im "elements"-Bereich erstellt, wird bei Übernahme die Eigenschaft **name** auf die ID des ausgewählten Objekts gesetzt.




## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung |
| download (optional) | (**default:** "") Ist dieser Parameter nicht leer und beinhaltet die Ausgabe des onChange-Skriptes eine Data-URL, so wird die Rückgabe als Textdatei mit dem Namen das download-Parameters heruntergeladen. (ab IP-Symcon 7.0) |
| enabled (optional) | (**default:** true) Bei true kann die Objektauswahl verwendet werden, ansonsten wird sie deaktiviert dargestellt (ab IP-Symcon 5.2) |
| link (optional) | (**default:** false) Ist dieser Parameter true, dann wird die Ausgabe des onChange-Skriptes als Link geöffnet. Ist er false, so wird die Ausgabe als Dialog im Konfigurationsformular angezeigt. (ab IP-Symcon 6.0) |
| name (optional) | Name der Objektauswahl/der zu setzenden Eigenschaft |
| onChange (optional) | (**default:** "") Skript, welches bei einer Veränderung des Wertes der Objektauswahl ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden (Arrays werden ab Version 6.0 unterstützt). Es hat die gleichen Eigenschaften wie onClick des Button (ab IP-Symcon 5.2) |
| type | SelectObject |
| value (optional) | (**default:** 1) Der Wert der Objektauswahl - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben (ab IP-Symcon 5.2) |
| visible (optional) | (**default:** true) Bei true ist die Objektauswahl sichtbar, ansonsten ist sie unsichtbar (ab IP-Symcon 5.2) |
| width (optional) | (**default:** 300px) Feste Breite der Objektauswahl in Pixel oder % als String, z.B. "40%" oder "250px" (ab IP-Symcon 5.3) |

## BEISPIEL

```
{ "type": "SelectObject", "name": "ObjectID", "caption": "Target" }
```

# SelectProfile

Benötigt: IP-Symcon >= 5.5
Erstellt einen Auswahldialog mit der Beschriftung **caption**.
Falls im "elements"-Bereich erstellt, wird bei Übernahme die Eigenschaft **name** auf den Namen des ausgewählten Profils gesetzt.




## PARAMETER

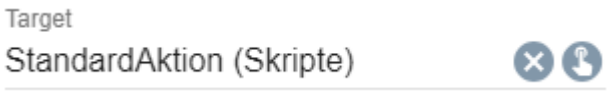
| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung |
| download (optional) | (**default:** "") Ist dieser Parameter nicht leer und beinhaltet die Ausgabe des onChange-Skriptes eine Data-URL, so wird die Rückgabe als Textdatei mit dem Namen das download-Parameters heruntergeladen. (ab IP-Symcon 7.0) |
| enabled (optional) | (**default:** true) Bei true kann die Profilauswahl verwendet werden, ansonsten wird sie deaktiviert dargestellt |
| link (optional) | (**default:** false) Ist dieser Parameter true, dann wird die Ausgabe des onChange-Skriptes als Link geöffnet. Ist er false, so wird die Ausgabe als Dialog im Konfigurationsformular angezeigt. (ab IP-Symcon 6.0) |
| name (optional) | Name der Profilauswahl/der zu setzenden Eigenschaft |
| onChange (optional) | (**default:** "") Skript, welches bei einer Veränderung des Wertes der Profilauswahl ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden (Arrays werden ab Version 6.0 unterstützt). Es hat die gleichen Eigenschaften wie onClick des Button |
| profileType (optional) | (**default:** -1) Der Typ des Profile welches ausgewählten werden kann (-1: Alle, 0: Boolean, 1: Integer, 2: Float, 3: String) |
| type | SelectProfile |
| value (optional) | (**default:** "") Der Wert der Profilauswahl - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben |
| visible (optional) | (**default:** true) Bei true ist die Profilauswahl sichtbar, ansonsten ist sie unsichtbar |
| width (optional) | (**default:** 300px) Feste Breite der Profilauswahl in Pixel oder % als String, z.B. "40%" oder "250px" |

## BEISPIEL

```
{ "type": "SelectProfile", "name": "PropertyProfileName", "caption":
"Profil - Integer" , "profileType": 1}
```

# SelectScript

Erstellt einen Auswahldialog mit der Beschriftung **caption**.
Falls im "elements"-Bereich erstellt, wird bei Übernahme die Eigenschaft **name** auf die ID des ausgewählten Skripts gesetzt.




## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung |
| download (optional) | (**default:** "") Ist dieser Parameter nicht leer und beinhaltet die Ausgabe des onChange-Skriptes eine Data-URL, so wird die Rückgabe als Textdatei mit dem Namen das download-Parameters heruntergeladen. (ab IP-Symcon 7.0) |
| enabled (optional) | (**default:** true) Bei true kann die Skriptauswahl verwendet werden, ansonsten wird sie deaktiviert dargestellt (ab IP-Symcon 5.2) |
| link (optional) | (**default:** false) Ist dieser Parameter true, dann wird die Ausgabe des onChange-Skriptes als Link geöffnet. Ist er false, so wird die Ausgabe als Dialog im Konfigurationsformular angezeigt. (ab IP-Symcon 6.0) |
| name (optional) | Name der Skriptauswahl/der zu setzenden Eigenschaft |
| onChange (optional) | (**default:** "") Skript, welches bei einer Veränderung des Wertes der Skriptauswahl ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden (Arrays werden ab Version 6.0 unterstützt). Es hat die gleichen Eigenschaften wie onClick des Button (ab IP-Symcon 5.2) |
| type | SelectScript |
| value (optional) | (**default:** 1) Der Wert der Skriptauswahl - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben (ab IP-Symcon 5.2) |
| visible (optional) | (**default:** true) Bei true ist die Skriptauswahl sichtbar, ansonsten ist sie unsichtbar (ab IP-Symcon 5.2) |
| width (optional) | (**default:** 300px) Feste Breite der Skriptauswahl in Pixel oder % als String, z.B. "40%" oder "250px" (ab IP-Symcon 5.3) |

## BEISPIEL

```
{ "type": "SelectScript", "name": "RXObjectID", "caption": "Target"
}
```

# SelectTime

Benötigt: IP-Symcon >= 5.0
Erstellt eine Eingabe für eine Uhrzeit mit der Beschriftung **caption**.




Falls im "elements"-Bereich erstellt, wird bei Übernahme die Eigenschaft **name** auf ein JSON-codiertes Objekt gesetzt, welches folgende Parameter enthält:

| PARAMETER | BESCHREIBUNG |
|---|---|
| hour | Stunden der Uhrzeit |
| minute | Minuten der Uhrzeit |
| second | Sekunden der Uhrzeit |

Wurde kein Zeitpunkt ausgewählt, so stehen alle Parameter auf 0.

## PARAMETER

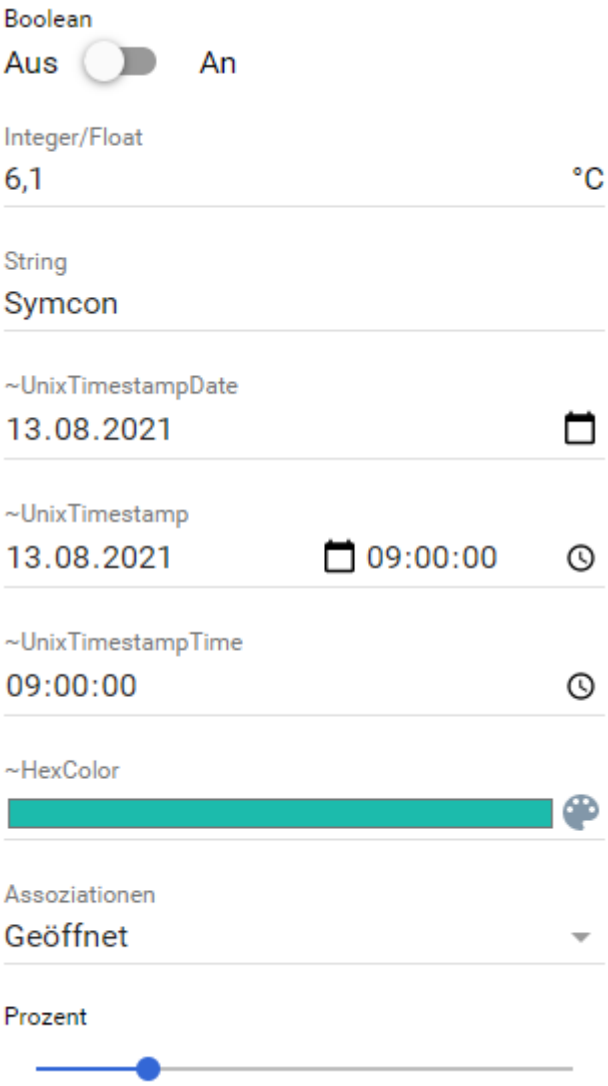
| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung |
| enabled (optional) | (**default:** true) Bei true kann die Auswahl verwendet werden, ansonsten wird sie deaktiviert dargestellt (ab IP-Symcon 5.2) |
| name (optional) | Name der Auswahl/der zu setzenden Eigenschaft |
| type | SelectTime |
| value (optional) | (**default:** kein Zeitpunkt ausgewählt) Der Wert der Auswahl - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben (ab IP-Symcon 5.2) |
| visible (optional) | (**default:** true) Bei true ist die Auswahl sichtbar, ansonsten ist sie unsichtbar (ab IP-Symcon 5.2) |
| width (optional) | (**default:** 300px) Feste Breite des Eingabefeldes in Pixel oder % als String, z.B. "40%" oder "250px" (ab IP-Symcon 5.3) |

## BEISPIEL

```
{ "type": "SelectTime", "name": "PropertyTime", "caption": "Time" }
```

# SelectValue

Benötigt: IP-Symcon >= 6.3
Erstellt einen Auswahldialog, der basierend auf dem Typen und dem Profil der gewählten **variableID** ein passendes Formularfeld anzeigt.
Falls im "elements"-Bereich erstellt, wird bei Übernahme die Eigenschaft **name** auf den derzeit ausgewählten Wert gesetzt.




## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung |
| enabled (optional) | (**default:** true) Bei true kann die Wertauswahl verwendet werden, ansonsten wird sie deaktiviert dargestellt (ab IP-Symcon 5.2) |
| name (optional) | Name der Wertauswahl/der zu setzenden Eigenschaft |
| onChange (optional) | (**default:** "") Skript, welches bei einer Veränderung des Wertes der Wertauswahl ausgeführt wird. Es hat die gleichen Eigenschaften wie onClick des Button |
| type | SelectValue |
| value (optional) | (**default:** null) Der JSON kodierte Wert der Auswahl. Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben |
| visible (optional) | (**default:** true) Bei true ist die Wertauswahl sichtbar, ansonsten ist sie unsichtbar |
| variableID | (**default:** 1) Die Variable, die für die Auswahl des Elementes herangezogen wird |
| width (optional) | (**default:** 300px) Feste Breite der Wertauswahl in Pixel oder % als String, z.B. "40%" oder "250px" |



Der Wert **value** wird als JSON kodierter String gespeichert.

## BEISPIEL

```
{
"type": "SelectValue",
"name": "Wert",
"caption": "Wert",
"variableID": 12345
}
```

# SelectVariable

Erstellt einen Auswahldialog mit der Beschriftung **caption**.
Falls im "elements"-Bereich erstellt, wird bei Übernahme die Eigenschaft **name** auf die ID der ausgewählten Variablen gesetzt.




## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung |
| download (optional) | (**default:** "") Ist dieser Parameter nicht leer und beinhaltet die Ausgabe des onChange-Skriptes eine Data-URL, so wird die Rückgabe als Textdatei mit dem Namen das download-Parameters heruntergeladen. (ab IP-Symcon 7.0) |
| enabled (optional) | (**default:** true) Bei true kann die Variablenauswahl verwendet werden, ansonsten wird sie deaktiviert dargestellt (ab IP-Symcon 5.2) |
| link (optional) | (**default:** false) Ist dieser Parameter true, dann wird die Ausgabe des onChange-Skriptes als Link geöffnet. Ist er false, so wird die Ausgabe als Dialog im Konfigurationsformular angezeigt. (ab IP-Symcon 6.0) |
| name (optional) | Name der Variablenauswahl/der zu setzenden Eigenschaft |
| onChange (optional) | (**default:** "") Skript, welches bei einer Veränderung des Wertes der Variablenauswahl ausgeführt wird. Es hat die gleichen Eigenschaften wie onClick des Button (ab IP-Symcon 5.2) |
| requiredAction (optional) | (**default:** 0) Gibt an ob die Variable eine Aktion haben muss (ab IP-Symcon 6.0)<br>0: Es ist egal, ob die Variable eine Aktion hat oder nicht<br>1: Die Variable muss eine Aktion haben<br>2: Die Variable darf keine Aktion haben |
| requiredLogging (optional) | (**default:** 0) Die Art auf die die Variable geloggt sein darf (ab IP-Symcon 6.0)<br>0: Es ist egal, ob die Variable geloggt ist oder nicht<br>1: Die Variable muss geloggt sein<br>2: Die Variable darf nicht geloggt sein<br>3: Die Variable muss als Standard geloggt sein<br>4: Die Variable muss als Zähler geloggt sein |
| type | SelectVariable |
| validVariableTypes (optional) | (**default:** []) Die Typen der Variablen, die ausgewählt werden können. Bei einem leeren Array sind alle Variablentypen zulässig (0: Boolean, 1: Integer, 2: Float, 3: String)(ab IP-Symcon 6.0) |
| value (optional) | (**default:** 1) Der Wert der Variablenauswahl - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben (ab IP-Symcon 5.2) |
| visible (optional) | (**default:** true) Bei true ist die Variablenauswahl sichtbar, ansonsten ist sie unsichtbar (ab IP-Symcon 5.2) |
| width (optional) | (**default:** 300px) Feste Breite der Variablenauswahl in Pixel oder % als String, z.B. "40%" oder "250px" (ab IP-Symcon 5.3) |



Entspricht die Variable nicht den durch die Parameter validVariableTypes, requiredAction oder requiredLogging definierten Anforderungen, wird ein Fehler angezeigt und Änderungen können nicht übernommen werden

## BEISPIEL

```
//Jede Variable kann ausgewählt werden
{
"type": "SelectVariable",
"name": "PropertyVariableID",
"caption": "Target"
}

//Die ausgewählte Variable muss ein Integer oder Float sein,
//sowie geloggt sein und eine Aktion haben
{
"type": "SelectVariable",
"name": "PropertyVariableID",
"caption": "Target",
"validVariableTypes": [1, 2],
"requiredAction": 1,
"requiredLogging": 1
}
```

# Statusmeldung

Erstellt eine Statusmeldung.

Statusmeldungen können nur im "status"-Bereich genutzt werden. Keine anderen Formularfeldtypen können im "status"-Bereich genutzt werden.

## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung |
| code | Statuscode (siehe Tabelle) |
| icon | Symbol welches angezeigt wird |

## ÜBERSICHT DER STATUSCODES

| CODE | BESCHREIBUNG |
|---|---|
| 101 | Instanz wird erstellt |
| 102 | Instanz ist aktiv |
| 103 | Instanz wird gelöscht |
| 104 | Instanz ist inaktiv |
| 105 | Instanz wurde nicht erstellt |
| >=200 | Instanz ist fehlerhaft |

Codes >=200 müssen selbstdefiniert werden.

## ÜBERSICHT DER MÖGLICHEN ICONS

active




error




inactive




## BEISPIEL

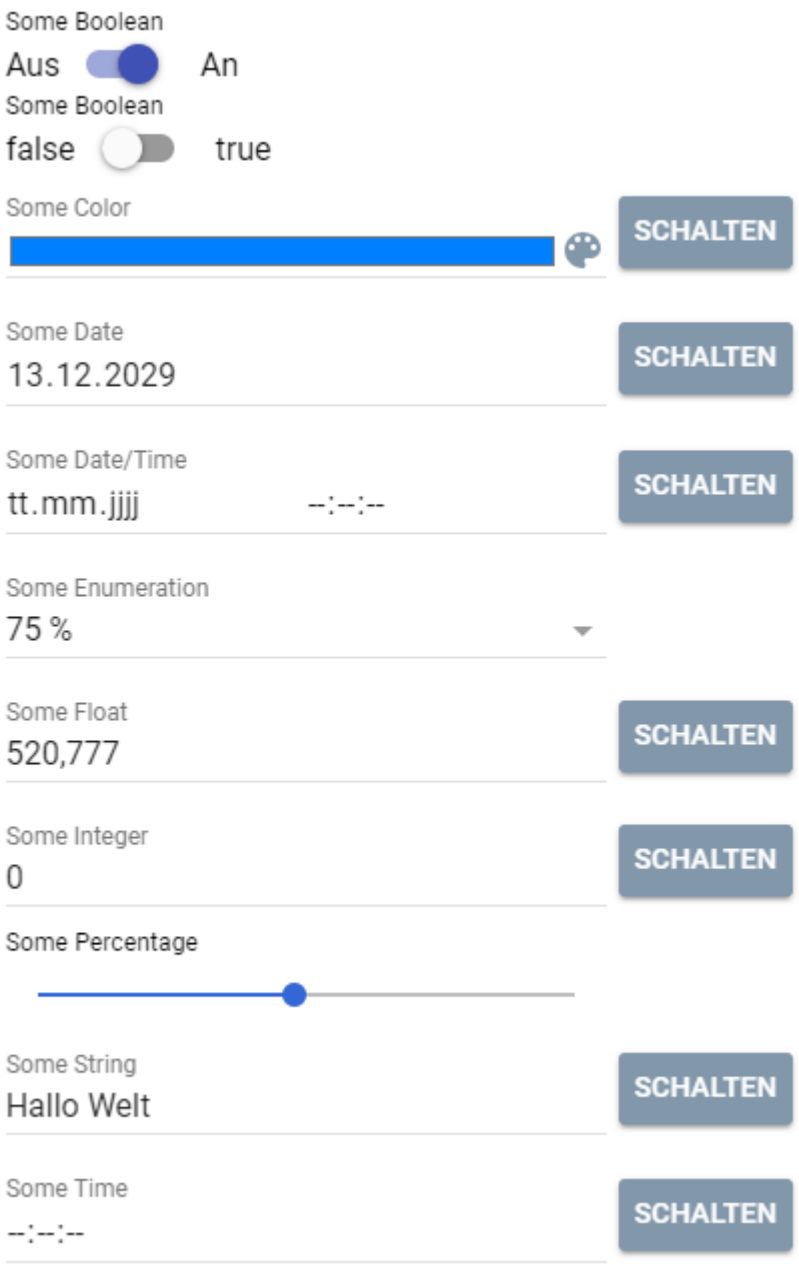
```
// Active icon
{ "code": 102, "icon": "active", "caption": "Logged in!" }

// Error icon
{ "code": 200, "icon": "error", "caption": "Error!" }

// Inactive icon
{ "code": 104, "icon": "inactive", "caption": "Logged out!" }
```

# TestCenter

Benötigt: IP-Symcon >= 5.0
Generiert einen Testbereich, in welchem alle schaltbaren Statusvariablen der Instanz geschaltet werden können. Die Bedienelemente basieren hierbei auf Typ und Profil der Variablen und orientieren sich an den Schaltelementen des WebFronts.




## PARAMETER

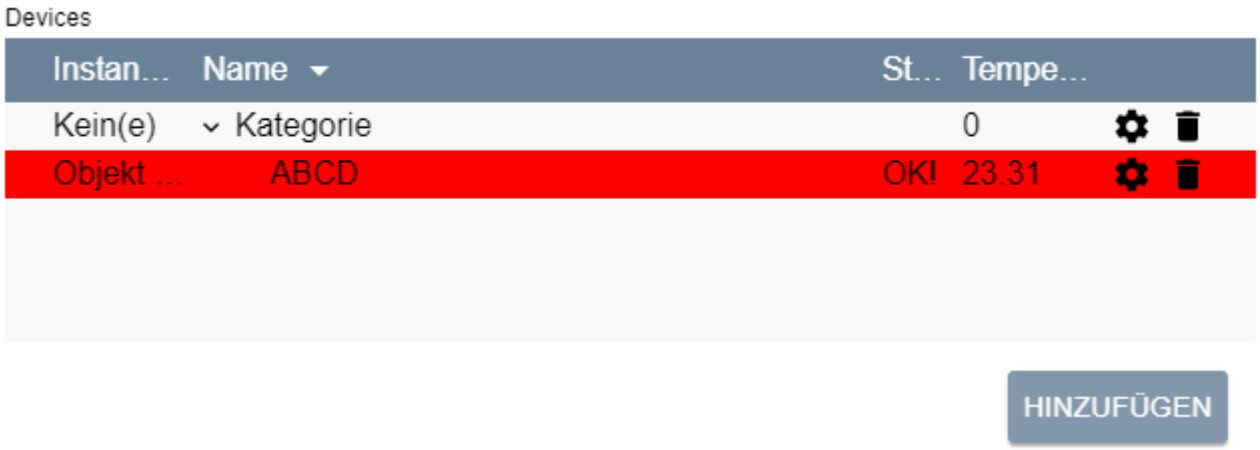
| PARAMETER | BESCHREIBUNG |
| --- | --- |
| caption (optional) | (**default:** "") Sichtbare Beschriftung (ab IP-Symcon 7.2) |
| enabled (optional) | (**default:** true) Bei true kann der Testbereich verwendet werden, ansonsten wird er deaktiviert dargestellt (ab IP-Symcon 5.2) |
| name (optional) | Name des Testbereichs |
| type | TestCenter |
| visible (optional) | (**default:** true) Bei true ist der Testbereich sichtbar, ansonsten ist er unsichtbar (ab IP-Symcon 5.2) |

## BEISPIEL

```
{
"type": "TestCenter"
}
```

# Tree

Benötigt: IP-Symcon >= 5.0
Erstellt einen Baum mit der Überschrift **caption**.
Die angezeigten Spalten werden in **columns** definert.
Initiale Einträge des Baumes können via **values** vorgegeben werden.




## ALLGEMEINE PARAMETER

| PARAMETER | BESCHREIBUNG |
| --- | --- |
| add (optional) | (**default:** false) Wenn true, dann wird unter der Tabelle ein "Hinzufügen"-Button eingeblendet, welcher dem Baum ein neues Element unter dem angewählten Knoten hinzufügt |
| caption (optional) | Sichtbare Beschriftung des Baumes |
| columns | Spalten des Baumes |
| delete (optional) | (**default:** false) Wenn true, dann wird hinter jedem Eintrag in des Baumes ein Button eingeblendet, welcher das entsprechende Element löscht |
| download (optional) | (**default:** "") Ist dieser Parameter nicht leer und beinhaltet die Ausgabe des onAdd, onDelete, onEdit oder onExpand-Skriptes eine Data-URL, so wird die Rückgabe als Textdatei mit dem Namen das download-Parameters heruntergeladen. (ab IP-Symcon 7.0) |
| form (optional) | (**default:** basierend auf den edit-Parametern von columns) Ein individuelles Formular zum Bearbeiten oder Hinzufügen von Zeilen. Die Definition des Formulars erfolg analog zur Definition des Formulars für Aktionen statisch oder als PHP-Code. Die Verwendung des Formulars funktioniert analog zum gleichnamigen Parameter der List (ab IP-Symcon 7.0) |
| link (optional) | (**default:** false) Ist dieser Parameter true, dann wird die Ausgabe des onAdd, onDelete, onEdit und onExpand-Skriptes als Link geöffnet. Ist er false, so wird die Ausgabe als Dialog im Konfigurationsformular angezeigt. (ab IP-Symcon 6.0) |
| name (optional) | Name des Baumes |
| onAdd (optional) | (**default:** "") Skript, welches nach dem Hinzufügen eines Eintrages des Baumes ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden (Arrays werden ab Version 6.0 unterstützt). In dieser Ausführung bietet der Baum als Variablenwert nicht den aktuell ausgewählten Eintrag sondern den hinzugefügten Eintrag, siehe Verwendung des Bauminhaltes in on-Aktionen. Ansonsten hat es die gleichen Eigenschaften wie onClick des Button (ab IP-Symcon 5.3) |
| onDelete (optional) | (**default:** "") Skript, welches nach dem Löschen eines Eintrages des Baumes ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden (Arrays werden ab Version 6.0 unterstützt). In dieser Ausführung bietet der Baum als Variablenwert nicht den aktuell ausgewählten Eintrag sondern den gelöschten Eintrag, siehe Verwendung des Bauminhaltes in on-Aktionen. Ansonsten hat es die gleichen Eigenschaften wie onClick des Button (ab IP-Symcon 5.3) |
| onEdit (optional) | (**default:** "") Skript, welches nach der Bearbeitung eines Eintrages des Baumes ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden (Arrays werden ab Version 6.0 unterstützt). In dieser Ausführung bietet der Baum als Variablenwert nicht den aktuell ausgewählten Eintrag sondern den bearbeiteten Eintrag, siehe Verwendung des Bauminhaltes in on-Aktionen. Ansonsten hat es die gleichen Eigenschaften wie onClick des Button (ab IP-Symcon 5.3) |

## PARAMETER FÜR COLUMNS

| PARAMETER | BESCHREIBUNG |
|---|---|
| add (optional) | initialer Wert in dieser Spalte bei einem neu hinzugefügtem Eintrag, notwendig wenn **add** bei dem Baum auf true gesetzt ist |
| caption | Sichtbare Beschriftung einer Spalte |
| confirm (optional) | (**default:** "") Abfragetext, bevor eine onClick-Aktion ausgeführt wird. Nur wenn die Abfrage mit "Ja" bestätigt wird, wird die Aktion ausgeführt. Ist kein Abfragetext definiert, so wird die Aktion direkt ausgeführt (ab IP-Symcon 5.2) |
| download (optional) | (**default:** "") Ist dieser Parameter nicht leer und beinhaltet die Ausgabe des onClick-Skriptes dieser Spalte eine Data-URL, so wird die Rückgabe als Textdatei mit dem Namen das download-Parameters heruntergeladen. (ab IP-Symcon 7.1) |
| edit (optional) | Beschreibung eines editierbaren Elements |
| ~~label~~ | ~~sichtbare Beschriftung einer Spalte~~ *(veraltet)* |
| link (optional) | (**default:** false) Ist dieser Parameter true, dann wird die Ausgabe des onClick-Skriptes dieser Spalte als Link geöffnet. Ist er false, so wird die Ausgabe als Dialog im Konfigurationsformular angezeigt. (ab IP-Symcon 7.1) |
| name | eindeutiger Name einer Spalte um diese zu referenzieren; der Name darf nicht "id", "parent", "expanded", "rowColor", "editable" oder "deletable" sein, da diese Werte für andere Funktionen belegt sind, siehe Parameter für values |
| onClick (optional) | (**default:** "") Skript, welches bei einem Klick auf ein Feld dieser Spalte ausgeführt wird. Es hat die gleichen Eigenschaften wie onClick des Button. Das Skript wird nur ausgeführt, sofern dieses Feld nicht leer, also 0, false oder "" ist (ab IP-Symcon 5.2) |
| quickFilter (optional) | (**default:** false) Bei true wird der Text dieser Spalte beim Schnellfilter in der Suche betrachtet. Sollte der Wert bei allen Spalten auf false gesetzt sein, so wird der Schnellfilter ausgeblendet. (ab IP-Symcon 7.2) |
| save (optional) | (**default:** true, falls das Element editierbar ist, sonst false) Bei true sollen die Werte dieser Spalte als Modul-Eigenschaften gesichert werden, siehe Speichern und Laden von Bäumen |
| visible (optional) | (**default:** true) Bei false wird die Spalte nicht angezeigt |
| width | Breite der Spalte in Pixeln, als CSS-String (z.B. "100px"); genau eine Spalte muss den Wert "auto" haben, wodurch die Breite dieser Spalte den restlich verfügbaren Platz nutzt. Darüber hinaus werden an dieser Spalte die Erweiterungs-Symbole des Baumes dargestellt. |

## PARAMETER FÜR EDIT

| PARAMETER | BESCHREIBUNG |
|---|---|
| type | Typ der Darstellung, entsprechend der möglichen Eingaben: CheckBox, IntervalBox, NumberSpinner, PasswordTextBox, Select, SelectCategory, SelectColor, SelectEvent, SelectFile, SelectInstance, SelectLink, SelectMedia, SelectObject, SelectScript, SelectVariable, ValidationTextBox, List, Tree |
| Weitere Parameter | Die weiteren Parameter ergeben sich je nach gewähltem Typ; die Parameter "name" und "caption" des Typs werden ignoriert |

Wird der "type" List oder Tree verwendet, so wird der Wert dieser Spalte nicht noch einmal JSON-codiert. Es ist also lediglich erforderlich den Wert des übergeordneten Baumes einmalig zu parsen. Jede Unterliste bzw. Unterbaum steht dann direkt als Objekt zur Verfügung. Analog müssen die Werte in "values" auch nicht doppelt codiert werden.

Der Parameter "add" der Spalte wird entsprechend der Lokalisierung übersetzt, sofern der "type" von "edit" ValidationTextBox oder PasswordTextBox ist oder "edit" gar nicht gesetzt ist

## PARAMETER FÜR SORT

| PARAMETER | BESCHREIBUNG |
|---|---|
| column | Name der Spalte, nach welcher der Baum sortiert wird |
| direction | Die Richtung der Sortierung; "ascending" für eine aufsteigende Sortierung und "descending" für eine absteigende Sortierung |

## PARAMETER FÜR VALUES

| PARAMETER | BESCHREIBUNG |
|---|---|
| deletable (optional) | (**default:** true) Bei false kann die Zeile nicht gelöscht werden (ab IP-Symcon 5.4) |
| editable (optional) | (**default:** true) Bei false kann die Zeile nicht editiert werden, obwohl der Baum an sich editierbare Spalten besitzt (ab IP-Symcon 5.4) |
| expanded (optional) | (**default:** false) Falls dieser Wert true ist, wird der Eintrag beim Laden des Konfigurationsformulares direkt ausgeklappt |
| id | Ein Identifikator, mit welchem der Eintrag in der Baumdarstellung als Elternknoten adressiert werden kann, id muss größer als 0 sein |
| parent (optional) | (**default:** 0) In diesem Feld kann die id eines Wertes gesetzt werden, welcher in der Baumdarstellung der Elternknoten von diesem Wert wird. Hiefür muss der Eltern-Wert im Array "values" vor dem Kind stehen. Ist "parent" 0, so wird das Element auf oberster Ebene eingefügt. |
| rowColor (optional) | (**default:** transparent) Hintergrundfarbe der Zeile als Hex-Code (z.B. Rot: "#ff0000" ), leerer String für Transparenz |
| Weitere Parameter (optional) | (**default:** "") Für jeden initialen Eintrag des Baumes werden hier die Werte zu allen Spalten definiert. Hierfür wird für jede Spalte ein Parameter mit dem Namen der Spalte eingetragen. Die Belegung des Parameters entspricht dem Wert dieser Spalte bei dem entsprechenden Eintrag. |

## SPEICHERN UND LADEN VON BÄUMEN

Das Speichern und Laden von Bäumen erfolgt fast analog zu Listen.
In der Eigenschaft werden zusätzlich die Werte für **id** und **parent** gespeichert.
Beim Laden ist zu beachten, dass bei der Verwendung von values immer der Identifikator des Knotens angegeben werden muss, damit diese dem passenden Knoten aus der Eigenschaft zugewiesen werden können.

## VERWENDUNG DES BAUMINHALTES IN ON-AKTIONEN

Die Verwendung der Werte des Baumes in Skripten erfolgt analog zu Listen. Darüber hinaus beinhaltet jeder Eintrag den Wert "expanded", welcher angibt ob der Eintrag aktuell eingeklappt (false) oder ausgeklappt (true) ist.

## BEISPIEL

```
{
"type": "Tree",
"name": "Devices",
"caption": "Devices",
"rowCount": 5,
"add": true,
"delete": true,
"sort": {
"column": "Name",
"direction": "ascending"
},
"columns": [{
"caption": "InstanceID",
"name": "InstanceID",
"width": "75px",
"add": 0,
"edit": {
"type": "SelectInstance"
}
}, {
"caption": "Name",
"name": "Name",
"width": "auto",
"add": ""
}, {
"caption": "State",
"name": "State",
"width": "40px",
"add": "New!"
}, {
"caption": "Temperature",
"name": "Temperature",
"width": "75px",
"add": 20.0,
"edit": {
"type": "NumberSpinner",
"digits": 2
}
}],
"values": [{
```

```
"id": 1,
"InstanceID": 0,
"Name": "Kategorie",
"State": "",
"Temperature": 0
},{
"id": 2,
"parent": 1,
"InstanceID": 12435,
"Name": "ABCD",
"State": "OK!",
"Temperature": 23.31,
"rowColor": "#ff0000"
}]
}
```

# ValidationTextBox

Erstellt ein Eingabefeld für Text mit der Überschrift **caption**.
Falls im "elements"-Bereich erstellt, setzt die ValidationTextBox eine Eigenschaft bei Übernahme auf den eingetragenen String.
Der Parameter **"name"** definiert dabei welche Eigenschaft gesetzt wird.




## PARAMETER

| PARAMETER | BESCHREIBUNG |
|---|---|
| caption | Sichtbare Beschriftung des Eingabefeldes |
| enabled (optional) | (**default:** true) Bei true kann das Eingabefeld verwendet werden, ansonsten wird es deaktiviert dargestellt (ab IP-Symcon 5.2) |
| multiline (optional) | (**default:** false) Bei true, wird das Eingabefeld bei Bedarf auf mehrere Zeilen erweitert. Dies kann sowohl durch bewusst gesetzte Zeilenumbrüche, z.B. per Enter, als auch bei einem Überlauf passieren. Bei false bleibt das Eingabefeld stets bei einer Zeile (ab IP-Symcon 5.4) |
| name (optional) | Name des Eingabefeldes/der zu setzenden Eigenschaft |
| onChange (optional) | (**default:** "") Skript, welches bei einer Veränderung des Dropdown-Wertes ausgeführt wird. Sofern das Skript aus mehreren Zeilen besteht, können die einzelnen Zeilen auch als Array übergeben werden. Es hat die gleichen Eigenschaften wie onClick des Button (ab IP-Symcon 8.1) |
| type | ValidationTextBox |
| validate (optional) | (**default:** "") Erfüllt der Wert nicht den hier angegebenen regulären Ausdruck, so wird ein Fehler angezeigt und Änderungen können nicht übernommen werden (ab IP-Symcon 5.3) |
| value (optional) | (**default:** "") Der Wert des Eingabefeldes - Sofern es eine dazugehörige Eigenschaft gibt, wird dieser Parameter im elements-Bereich durch die Eigenschaft überschrieben (ab IP-Symcon 5.2) |
| visible (optional) | (**default:** true) Bei true ist das Eingabefeld sichtbar, ansonsten ist es unsichtbar (ab IP-Symcon 5.2) |
| width (optional) | (**default:** 300px) Feste Breite des Eingabefeldes in Pixel oder % als String, z.B. "40%" oder "250px" (ab IP-Symcon 5.3) |

## BEISPIEL

```
{ "type": "ValidationTextBox", "name": "IPAddress", "caption":
"Host" }
```

# Konstanten

Benötigt: IP-Symcon >= 5.1

## KONSTANTEN

| DEFINE | BESCHREIBUNG |
|---|---|
| EVENTCONDITIONCOMPARISON | Vergleichskonditionen für bedingte Ereignisse |
| EVENTCYCLICDATETYPE | Datumstyp bei zyklischem Ereignis |
| EVENTCYCLICTIMETYPE | Zeittyp des Ereignisses |
| EVENTTRIGGERTYPE | Auslösetyp bei ausgelösten Ereignis |
| EVENTTYPE | Typ des Ereignisses |
| MEDIATYPE | Typ der Mediainstanz |
| MODULETYPE | Typ des Moduls |
| OBJECTTYPE | Typ des Objekts |
| SCRIPTTYPE | Typ des Skripts |
| VARIABLETYPE | Typ der Variable |

### EVENTCONDITIONCOMPARISON

Vergleichskonditionen für bedingte Ereignisse

| KONSTANTE | WERT | BESCHREIBUNG |
|---|---|---|
| EVENTCONDITIONCOMPARISON_EQUAL | 0 | Gleich |
| EVENTCONDITIONCOMPARISON_NOTEQUAL | 1 | Ungleich |
| EVENTCONDITIONCOMPARISON_GREATER | 2 | Größer als |
| EVENTCONDITIONCOMPARISON_GREATEROREQUAL | 3 | GrößerGleich |
| EVENTCONDITIONCOMPARISON_SMALLER | 4 | Kleiner als |
| EVENTCONDITIONCOMPARISON_SMALLEROREQUAL | 5 | KleinerGleich |

### EVENTCYCLICDATETYPE

Datumstyp bei zyklischem Ereignis

| KONSTANTE | WERT | BESCHREIBUNG |
|---|---|---|
| EVENTCYCLICDATETYPE_NONE | 0 | Kein Datumstyp |
| EVENTCYCLICDATETYPE_ONCE | 1 | Einmalig |
| EVENTCYCLICDATETYPE_DAY | 2 | Täglich |
| EVENTCYCLICDATETYPE_WEEK | 3 | Wöchentlich |
| EVENTCYCLICDATETYPE_MONTH | 4 | Monatlich |
| EVENTCYCLICDATETYPE_YEAR | 5 | Jährlich |

## EVENTCYCLICTIMETYPE

Zeittyp des Ereignisses

| KONSTANTE | WERT | BESCHREIBUNG |
|---|---|---|
| EVENTCYCLICTIMETYPE_ONCE | 0 | Einmalig |
| EVENTCYCLICTIMETYPE_SECOND | 1 | Sekündlich |
| EVENTCYCLICTIMETYPE_MINUTE | 2 | Minütlich |
| EVENTCYCLICTIMETYPE_HOUR | 3 | Stündlich |

## EVENTTRIGGERTYPE

Auslösetyp bei ausgelösten Ereignis

| KONSTANTE | WERT | BESCHREIBUNG |
|---|---|---|
| EVENTTRIGGERTYPE_ONUPDATE | 0 | Bei Variablenaktualisierung |
| EVENTTRIGGERTYPE_ONCHANGE | 1 | Bei Variablenänderung |
| EVENTTRIGGERTYPE_ONLIMITEXCEED | 2 | Bei Grenzwertüberschreitung |
| EVENTTRIGGERTYPE_ONLIMITDROP | 3 | Bei Grenzwertunterschreitung |
| EVENTTRIGGERTYPE_ONVALUE | 4 | Bei bestimmten Wert |

## EVENTTYPE

Typ des Ereignisses

| KONSTANTE | WERT | BESCHREIBUNG |
|---|---|---|
| EVENTTYPE_TRIGGER | 0 | Ausgelöstes Ereignis |
| EVENTTYPE_CYCLIC | 1 | Zyklisches Ereignis |
| EVENTTYPE_SCHEDULE | 2 | Wochenplan Ereignis |

## MEDIATYPE

Typ der Mediainstanz

| KONSTANTE | WERT | BESCHREIBUNG |
|---|---|---|
| MEDIATYPE_IPSVIEW | 0 | IPSView |
| MEDIATYPE_IMAGE | 1 | Bild |
| MEDIATYPE_SOUND | 2 | Ton |
| MEDIATYPE_STREAM | 3 | Stream |
| MEDIATYPE_CHART | 4 | Chart |
| MEDIATYPE_DOCUMENT | 5 | Dokument |

## MODULETYPE

Typ des Moduls

| KONSTANTE | WERT | BESCHREIBUNG |
|---|---|---|
| MODULETYPE_CORE | 0 | Kern |
| MODULETYPE_IO | 1 | I/O |
| MODULETYPE_SPLITTER | 2 | Splitter |
| MODULETYPE_DEVICE | 3 | Geräte |
| MODULETYPE_CONFIGURATOR | 4 | Konfigurator |
| MODULETYPE_DISCOVERY | 5 | Erkennungsdienst |
| MODULETYPE_VISUALIZATION | 6 | Visualisierung |

## OBJECTTYPE

Typ des Objekts

| KONSTANTE | WERT | BESCHREIBUNG |
|---|---|---|
| OBJECTTYPE_CATEGORY | 0 | Kategorie |
| OBJECTTYPE_INSTANCE | 1 | Instanz |
| OBJECTTYPE_VARIABLE | 2 | Variablen |
| OBJECTTYPE_SCRIPT | 3 | Skript |
| OBJECTTYPE_EVENT | 4 | Ereignis |
| OBJECTTYPE_MEDIA | 5 | Media |
| OBJECTTYPE_LINK | 6 | Link |

## SCRIPTTYPE

Typ des Skripts

| KONSTANTE | WERT | BESCHREIBUNG |
|---|---|---|
| SCRIPTTYPE_PHP | 0 | PHP-Skript |
| SCRIPTTYPE_FLOW | 1 | Ablaufplan |
| SCRIPTTYPE_IPSWORKFLOW | 2 | IPSWorkflow |

## VARIABLETYPE

Typ der Variable

| KONSTANTE | WERT | BESCHREIBUNG |
|---|---|---|
| VARIABLETYPE_BOOLEAN | 0 | Boolean |
| VARIABLETYPE_INTEGER | 1 | Integer |
| VARIABLETYPE_FLOAT | 2 | Float |
| VARIABLETYPE_STRING | 3 | String |

## DARSTELLUNGEN

| GUID | KONSTANTE |
|---|---|
| {6B9CAEEC-5958-C223-30F7-BD36569FC57A} | VARIABLE_PRESENTATION_SLIDER |
| {4153A8D4-5C33-C65F-C1F3-7B61AAF99B1C} | VARIABLE_PRESENTATION_LEGACY |
| {3319437D-7CDE-699D-750A-3C6A3841FA75} | VARIABLE_PRESENTATION_VALUE_PRESENTAT |
| {6F477326-1683-A2FD-D2E7-477F366ECB62} | VARIABLE_PRESENTATION_VALUE_INPUT |
| {9DE1D610-5106-97FB-714D-1AADEDF8377A} | VARIABLE_PRESENTATION_WEB_CONTENT |
| {05CC3CC2-A0B2-5837-A4A7-A07EA0B9DDFB} | VARIABLE_PRESENTATION_COLOR |
| {497C4845-27FA-6E4F-AE37-5D951D3BDBF9} | VARIABLE_PRESENTATION_DATE_TIME |
| {60AE6B26-B3E2-BDB1-A3A1-BE232940664B} | VARIABLE_PRESENTATION_SWITCH |
| {6075FC22-69AF-B110-3749-C24138883082} | VARIABLE_PRESENTATION_SHUTTER |
| {52D9E126-D7D2-2CBB-5E62-4CF7BA7C5D82} | VARIABLE_PRESENTATION_ENUMERATION |
| {08A6AF76-394E-D354-48D5-BFC690488E4E} | VARIABLE_PRESENTATION_DURATION |

## VORLAGEN

| GUID | KONSTANTE |
|---|---|
| {B4C70F3E-6613-DA1A-7279-5DEE8DEB1B24} | VARIABLE_TEMPLATE_DATE |
| {362DA268-56A2-E771-5E53-17E38B5D82E6} | VARIABLE_TEMPLATE_TIME |
| {BB0E9933-0403-BD3A-D1C9-255646934B00} | VARIABLE_TEMPLATE_DATE_TIME |
| {868B087E-A38D-2155-EBE0-157AFBBF9E8C} | VARIABLE_TEMPLATE_SLIDER_ROOM_TEMPER |
| {66062309-21A9-26C0-213F-775C52E1473B} | VARIABLE_TEMPLATE_SLIDER_COLOR_TEMPER |
| {BC799412-0C66-551F-CAEC-7566F5D52BD9} | VARIABLE_TEMPLATE_SLIDER_ENERGY |
| {8EC19DF0-89FB-A77E-ED7D-047A949CF292} | VARIABLE_TEMPLATE_SLIDER_POWER |
| {3BE75DE9-7D84-C082-2E77-9ED3AEE04D63} | VARIABLE_TEMPLATE_SHUTTER_LAMELLA_RIG |
| {22A0DF9C-C200-154A-641B-3A3CB096DB6D} | VARIABLE_TEMPLATE_SHUTTER_LAMELLA_LEF |
| {90AF8F8F-183F-BBFD-E078-35FAB6DCFE4F} | VARIABLE_TEMPLATE_VALUE_PRESENTATION_I |
| {2FED3D39-073D-6037-901B-2586A1AB5569} | VARIABLE_TEMPLATE_VALUE_PRESENTATION_I |
| {C899FCFA-063E-897E-9DA4-28ADD278EED5} | VARIABLE_TEMPLATE_VALUE_PRESENTATION_I |
| {7BD38CF5-07F2-5B5B-8F7F-15398B823BFC} | VARIABLE_TEMPLATE_VALUE_PRESENTATION_I |
| {C90EF36A-165E-D0B0-032C-F468F483D42B} | VARIABLE_TEMPLATE_VALUE_PRESENTATION_I |
| {0C711895-2F8E-DBFE-1700-84173491D229} | VARIABLE_TEMPLATE_COLOR_RAINBOW |
| {A7467E68-5C39-5BD9-C0C8-BCE6004FEEAA} | VARIABLE_TEMPLATE_COLOR_FOREST |

# Lokalisierungen

Benötigt: IP-Symcon >= 4.1
Die JSON-Datei "locale.json" beinhaltet die Übersetzungen für "label" und "caption" der Konfigurationsseite. Hier sind auch abstufende Sprachenkürzel möglich. Zum Beispiel können auch "de_DE" oder "de_AT" genutzt werden.

## BEISPIELE

Grundaufbau der JSON Datei
Die Sprachenkürzel geben an in welche Sprache übersetzt werden soll

```
{
"translations": {
"de": {
"Zu übersetzendes Wort 1": "Übersetzung ins Deutsche - Wort 1",
"Zu übersetzendes Wort 2": "Übersetzung ins Deutsche - Wort 2"
},
"de_DE": {
"Zu übersetzendes Wort 1": "Übersetzung ins Deutsche - Wort 1",
"Zu übersetzendes Wort 2": "Übersetzung ins Deutsche - Wort 2"
},
"de_CH": {
"Zu übersetzendes Wort 1": "Übersetzung ins Schwiizerdütsch - Wort
1",
"Zu übersetzendes Wort 2": "Übersetzung ins Schwiizerdütsch - Wort
2"
}
}
}
```

Vollständiges Beispiel anhand eines SMS Moduls (siehe Beispiel Konfigurationsformulare )

```
{
"translations": {
"de": {
"Username": "Benutzername",
"Password": "Passwort",
"Sender": "Absender",
"Type": "Typ",
"Combi-SMS": "Kombi-SMS",
"Number": "Nummer",
"Message": "Nachricht",
"Send Message": "Sende Nachricht",
"Read Balance": "Frage Guthaben ab",
"Login information valid": "Logindaten sind gültig",
"Authentication failed": "Authentifizierung ist fehlgeschlagen",
"No credits left": "Kein Guthaben vorhanden"
```

```
  }
  }
  }
```

# Module

Benötigt: IP-Symcon >= 4.0

## BESCHREIBUNG

Ein Modul besteht grundlegend aus 2 Dateien. Der module.php und module.json.
Optional kann eine Konfigurationsseite (form.json) bereitgestellt werden.

## FORM.JSON

Weitere Informationen zur Konfigurationsseite können unter Konfigurationsformulare nachgeschlagen werden.

## LOCALE.JSON

Weitere Informationen zur Übersetzung der Konfigurationsseite können unter Lokalisierungen nachgeschlagen werden.

## MODULE.JSON

Diese Datei beinhaltet Rahmeninformation, die unabdingbar sind, damit das Modul identifizierbar ist und korrekt eingebunden werden kann.

| PARAMETER | DATENTYP | BESCHREIBUNG |
|---|---|---|
| id | string | Einmalige GUID zur eindeutigen Identifikation. GUID Generator |
| name | string | Der Name des Moduls. (A-Z, a-z, 0-9, Leerzeichen, Unterstrich sind erlaubte Zeichen. Leerzeichen und Unterstriche dürfen dabei jedoch nicht am Anfang oder Ende sein. Ein leerer Name ist ebenfalls nicht gültig.) |
| type | integer | Type des Moduls (0: Kern, 1: I/O, 2: Splitter, 3: Gerät, 4: Konfigurator, 5: Discovery) |
| vendor | string | Herstellername und der Name des Menüpunkts unter dem das Gerät in "Instanz hinzufügen" gefunden werden kann. Wenn nichts angegeben wird, wird das Gerät unter "(Sonstige)" eingetragen. |
| aliases | array [string] | Zusätzliche Gerätenamen/-varianten |
| url | string | URL zur Dokumentationnsseite des Moduls (Muss mit http:// oder https:// beginnen. Darf Alternativ auch "" (leer) sein) |
| parentRequirements | array [string] | Datenfluss GUIDs, wodurch kompatible übergeordnete Instanzen ermittelt werden. Die übergeordnete Instanz muss mindestens eine dieser Datenfluss GUIDs als Implemented führen, um kompatibel zu sein |
| childRequirements | array [string] | Datenfluss GUIDs, wodurch kompatible untergeordnete Instanzen ermittelt werden. Die untergeordnete Instanz muss mindestens eine dieser Datenfluss GUIDs als Implemented führen, um kompatibel zu sein |
| implemented | array [string] | Unterstützte Datafluss GUIDs müssen in den jeweiligen ReceiveData/ForwardData Funktionen korrekt ausgewertet und unterstützt werden, sofern diese hier aufgelistet werden |
| prefix | string | Prefix, welcher den Funktionen vorausgeschrieben wird. Darf nur Zahlen und Buchstaben enthalten. |

```
{
"id": "{E5AA629B-75BD-45C0-9BCB-845C102B0411}",
"name": "ModulnameXYZ",
"type": 3,
"vendor": "",
"aliases":
[
"Name1UnterstütztesGerät",
"Name2UnterstütztesGerät"
],
"url": "https://www.symcon.de",
"parentRequirements": [],
"childRequirements": [],
"implemented": [],
"prefix": "ABC"
}
```

## MODULE.PHP

Dies ist die eigentliche Klassendatei, welche die Funktionen beinhaltet die hinterher Daten verarbeiten und weiterleiten.



Der Klassenname muss identisch zum "name" Parameter sein, welcher in der module.json definiert wurde. Einziger erlaubter Unterschied sind Leerzeichen. Diese müssen im Klassennamen innerhalb der module.php entfernt werden.



Funktionsnamen dürfen nur aus folgenden Zeichen bestehen: "a..z","A..Z","0..9". Des weiteren darf "$InstanceID" nicht als Parametername verwendet werden.

Seit IP-Symcon 8.1 gibt es die verbesserte Basisklasse IPSModuleStrict.
IPSModule ist weiter verfügbar, sollte jedoch für neue Module nicht mehr verwendet werden.
Die Folgende Tabelle zeigt die Unterschiede auf:

| MERKMAL | IPSMODULE | IPSMODULESTRICT |
|---|---|---|
| Type Hints | Optional für public Funktionen | Immer erforderlich |
| Fehlermeldungen bei fehlenden Type Hints in public Funktionen | Nur Warnungen | Führt zu Fehlern |
| Unterstützung veralteter Typen | Integer/Boolean erlaubt | Bitte int/bool benutzen |
| Rückgabe von `RegisterVariable*` | Aktuelle Variablen-ID as int | boolean, ob Variable erstellt wurde, um ggf. Startwert zu setzen) |
| Schreibzugriff auf erstellte Variablen | Immer, auch Kundenseitig per SetValue | Nur über `$this->SetValue` (Variablen sind über ReadOnly gesichert) |
| Datenfluss Verbindungen | Komplex über ConnectParent/ RequireParent/ ForceParent | Automatisch über die Kompatibilität und GetCompatibleParents() |
| Datenfluss Kodierung | UTF-8 (utf8_encode/ utf8_decode), Problematisch ab PHP 9.0 | HEX-kodiert (bin2hex/hex2bin) und einfacher erkennbar wann eine Kodierung erforderlich ist |

### VORLAGE MINIMAL

```
// Klassendefinition
class ModulnameXYZ extends IPSModuleStrict {
```

```
/**
* Die folgenden Funktionen stehen automatisch zur Verfügung, wenn
das Modul über das "Module Control" eingefügt wurden.
* Die Funktionen werden, mit dem selbst eingerichteten Prefix, in
PHP und JSON-RPC wie folgt zur Verfügung gestellt:
*
* ABC_MeineErsteEigeneFunktion($id);
*
*/
public function MeineErsteEigeneFunktion(): void {
echo $this->InstanceID;
}

}
```

## VORLAGE KLASSISCH

```
// Klassendefinition
class ModulnameXYZ extends IPSModuleStrict {

// Überschreibt die interne IPS_Create($id) Funktion
public function Create(): void {
// Diese Zeile nicht löschen.
parent::Create();
}

// Überschreibt die interne IPS_ApplyChanges($id) Funktion
public function ApplyChanges(): void {
// Diese Zeile nicht löschen
parent::ApplyChanges();
}

/**
* Die folgenden Funktionen stehen automatisch zur Verfügung, wenn
das Modul über die "Module Control" eingefügt wurden.
* Die Funktionen werden, mit dem selbst eingerichteten Prefix, in
PHP und JSON-RPC wie folgt zur Verfügung gestellt:
*
* ABC_MeineErsteEigeneFunktion($id);
*
*/
public function MeineErsteEigeneFunktion(): void {
// Selbsterstellter Code
}

}
```

# __construct

Benötigt: IP-Symcon >= 4.0

```
void __construct (string $InstanzID)
```

## PARAMETERLISTE

| | |
|---|---|
| InstanzID | ID dieser Instanz |

## RÜCKGABEWERT

Keine Rückgabe

## BESCHREIBUNG

Der Konstruktor wird bei jeglichem Funktionsaufruf aufgerufen und kümmert sich um die Instanzierung.
Die Basisklasse nutzt den Konstruktor um z.B. die InstanzID zu setzen.



Sollen eigene Erweiterungen hinzugefügt werden, muss der Konstruktor von der Basisklasse überschrieben werden.

## BEISPIEL

```
// Diese Funktion muss normalerweise nicht überschrieben werden.
```

# ApplyChanges

Benötigt: IP-Symcon >= 4.0

```
void ApplyChanges ()
```

## RÜCKGABEWERT

Keine Rückgabe

## BESCHREIBUNG

Wird ausgeführt, wenn auf der Konfigurationsseite "Übernehmen" gedrückt wird und nach dem unittelbaren Erstellen der Instanz.



Die ApplyChanges Funktion wird von IP-Symcon aufgerufen. Sie muss somit von der Basisklasse überschrieben werden, um eigene Erweiterungen hinzuzufügen

## BEISPIEL

```
// IPSModuleStrict
public function ApplyChanges(): void {

// Diese Zeile nicht löschen
parent::ApplyChanges();

// ConnectParent/RequireParent/ForceParent ist nicht verfügbar -
kann aber durch die neue Funktion "GetCompatibleParents()" ersetzt
werden.

}

// IPSModule
public function ApplyChanges() {

// Diese Zeile nicht löschen
parent::ApplyChanges();

// Wenn keine übergeordenete Instanz vorhanden ist, erstelle eine
neue eigene VirtualIO Instanz
$this->RequireParent("{6179ED6A-FC31-413C-BB8E-1204150CF376}");

}
```

# ConnectParent

Benötigt: IP-Symcon >= 4.0

```
boolean ConnectParent (string $ÜbergeordneteGUID)
```

## PARAMETERLISTE

| | |
|---|---|
| ÜbergeordneteGUID | GUID |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG



Diese Funktion ist nur für IPSModule verfügbar. Die Alternative für IPSModuleStrict ist GetCompatibleParents.

Die Funktion verbindet die Instanz mit einer vorhandenen übergeordneten kompatiblen (Parent) Instanz. Sollte diese nicht vorhanden sein, wird diese erstellt und anschließend verbunden. Eine Liste der verfügbaren ModulIDs (GUID) kann hier entnommen werden.



Der Normalfall für die Funktion ist ein Device-Modul, welches sich an einen Splitter hängen möchte. Dabei wird nur, sofern kein Splitter vorhanden ist, ein neuer erstellt.

## BEISPIEL

```
// IPSModuleStrict
ConnectParent/RequireParent/ForceParent ist nicht verfügbar,
kann aber durch die neue Funktion "GetCompatibleParents()" ersetzt
werden.

// IPSModule
public function Create() {

// Die Zeile niemals entfernen!
parent::Create();
```

```
// Verbinde mit vorhandenem Splitter oder erstelle gegebenfalls
einen neuen
$this->ConnectParent("{46C969BF-3465-4E3E-B2A5-E404FB969735}");

}
```

# Create

Benötigt: IP-Symcon >= 4.0

```
void Create ()
```

## RÜCKGABEWERT

Keine Rückgabe

## BESCHREIBUNG

Im Gegensatz zu Construct wird diese Funktion nur einmalig beim Erstellen der Instanz und Start von IP-Symcon aufgerufen. Deshalb sollten hier Statusvariablen und Modul-Eigenschaften erstellt werden, die das Modul dauerhaft braucht.
Häufig genutzte Funktionen:
RegisterPropertyString
RegisterPropertyInteger
RegisterPropertyFloat
RegisterPropertyBoolean



Die Create Funktion wird von IP-Symcon aufgerufen. Sie muss somit von der Basisklasse überschrieben werden, um eigene Erweiterungen hinzuzufügen

## BEISPIEL

```
// IPSModuleStrict
public function Create(): void {

// Diese Zeile nicht entfernen
parent::Create();

// Modul-Eigenschaftserstellung
$this->RegisterPropertyString("Benutzername", "MaxMustermann");
$this->RegisterPropertyInteger("Zahl", 123);
$this->RegisterPropertyFloat("Faktor", 0.5);
$this->RegisterPropertyBoolean("Geöffnet", true);

}

// IPSModule
public function Create() {
```

```
// Beispiel identisch. Bitte die geänderte Funktionssignatur
beachten.

}
```

# Destroy

Benötigt: IP-Symcon >= 4.1

```
void Destroy ()
```

## RÜCKGABEWERT

Keine Rückgabe

## BESCHREIBUNG

Diese Funktion wird beim Löschen der Instanz im laufenden Betrieb und bei Update via "Module Control" aufgerufen. Die Funktion wird nicht beim Beenden von IP-Symcon aufgerufen.



Die Destroy Funktion wird von IP-Symcon aufgerufen. Sie muss somit von der Basisklasse überschrieben werden, um eigene Erweiterungen hinzuzufügen

## BEISPIEL

```
// IPSModuleStrict
public function Destroy(): void {

// Diese Zeile nicht entfernen
parent::Destroy();

}

// IPSModule
public function Destroy() {

// Beispiel identisch. Bitte die geänderte Funktionssignatur
beachten.

}
```

# DisableAction

Benötigt: IP-Symcon >= 4.0

```
boolean DisableAction (string $Ident)
```

## PARAMETERLISTE

| | |
|---|---|
| Ident | Ident der Variable |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Deaktiviert die Standardaktion der Statusvariable. Dadurch ist diese in der Visualisierung nicht mehr veränderbar, sondern kann nur gelesen werden. Sofern EnableAction nicht aufrufen wurde, muss diese Funktion nicht aufgerufen werden, da alle Statusvariablen standardmäßig ohne Standardaktion erstellt werden.

## BEISPIEL

```
// Deaktiviert die Standardaktion der Statusvariable
$this->DisableAction("Status");
```

# EnableAction

Benötigt: IP-Symcon >= 4.0

```
boolean EnableAction (string $Ident)
```

## PARAMETERLISTE

| | |
|---|---|
| Ident | Ident der Variable |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Aktiviert die Standardaktion der Statusvariable. Dadurch ist diese in der Visualisierung veränderbar und kann auch beschrieben werden. Diese Funktion muss aufgerufen werden, da alle Statusvariablen standardmäßig ohne Standardaktion erstellt werden. Sofern die Standardaktion aktiviert ist, muss auf Änderungsanfragen innerhalb von RequestAction reagiert werden.

## BEISPIEL

```
// Aktiviert die Standardaktion der Statusvariable
$this->EnableAction("Status");
```

# ForceParent

Benötigt: IP-Symcon >= 4.0

```
boolean ForceParent (string $ModulID)
```

## PARAMETERLISTE

| | |
|---|---|
| ModulID | GUID |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG



Diese Funktion ist nur für IPSModule verfügbar. Die Alternative für IPSModuleStrict ist GetCompatibleParents.

Es wird eine Verbindung mit einer kompatiblen übergeordneten Instanz mit der GUID **ModulID** forciert. Falls die bisherige Verbindung nicht mit der gewünschten **ModulID** übereinstimmt, wird diese getrennt. Anschließend wird eine neue Instanz mit der übergordneten ModuleID erstellt und verbunden.
Eine Liste der verfügbaren ModulIDs kann hier entnommen werden.



Der Normalfall für die Funktion ist ein Splitter-Modul, welches je nach Konfiguration in der Oberfläche ein anderes I/O Modul benötigt.



Diese Funktion wird normalerweise nur in der Create() oder ApplyChanges() Methode aufgerufen.

## BEISPIEL

```
// IPSModuleStrict
ConnectParent/RequireParent/ForceParent ist nicht verfügbar,
```

```
kann aber durch die neue Funktion "GetCompatibleParents()" ersetzt
werden.

// IPSModule
public function ApplyChanges() {

// Die Zeile niemals entfernen!
parent::ApplyChanges();

// Je nach Konfiguration unterschiedliche I/O Instanzen erstellen
switch($this->ReadPropertyInteger("GatewayMode")) {
case 0: //ClientSocket bei Modus 0 erstellen
$this->ForceParent("{3CFF0FD9-E306-41DB-9B5A-9D06D38576C3}");
break;
case 1: //SerialPort bei Modus 1 erstellen
$this->ForceParent("{6DC3D946-0D31-450F-A8C6-C42DB8D7D4F1}");
break;
}

}
```

# ForwardData

Benötigt: IP-Symcon >= 4.0

```
string ForwardData (string $JSONString)
```

## PARAMETERLISTE

| | |
|---|---|
| JSONString | Datenpaket im JSON Format |

## RÜCKGABEWERT

Resultat der Funktion, welches an die aufrufendene untergeordnete Instanz zurückgegeben wird

## BESCHREIBUNG

Diese Funktion wird von IP-Symcon aufgerufen und verarbeitet zugesendete Daten und schickt diese an die übergeordnete Instanz weiter.
Daten können über die Funktion SendDataToParent versendet werden.
Weitere Informationen zur Datenweiterleitung können unter Datenfluss gefunden werden.



Die ForwardData Funktion wird von IP-Symcon aufgerufen. Sie muss somit von der Basisklasse überschrieben werden, um eigene Erweiterungen hinzuzufügen

## BEISPIEL

```
// IPSModuleStrict
public function ForwardData(string $JSONString): string {

// Beispiel innerhalb einer Gateway/Splitter Instanz

// Empfangene Daten von der Device Instanz
$data = json_decode($JSONString);
IPS_LogMessage("ForwardData", utf8_decode($data->Buffer));

// Hier würde man den Buffer im Normalfall verarbeiten
// z.B. CRC prüfen, in Einzelteile zerlegen

// Weiterleiten zur I/O Instanz
```

```
$resultat = $this->SendDataToParent(json_encode(Array("DataID" =>
"{79827379-F36E-4ADA-8A95-5F8D1DC92FA9}", "Buffer" => $data-
>Buffer)));

// Weiterverarbeiten und durchreichen
return $resultat;

}

// IPSModule
public function ForwardData($JSONString) {

// Beispiel identisch. Bitte die geänderte Funktionssignatur
beachten.

}
```

# GetBuffer

Benötigt: IP-Symcon >= 4.1

```
string GetBuffer (string $Name)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Der Name des Buffers |

## RÜCKGABEWERT

Der Inhalt des Buffers

## BESCHREIBUNG

Diese Funktion liefert den Inhalt des Buffers mit dem Namen **Name**. Wenn der Buffer leer ist, wird ein leerer String zurückgegeben.



Der Inhalt des Buffers ist nur solange vorhanden, wie IP-Symcon läuft. Mit einem Neustart verschwinden alle Informationen

## BEISPIEL

```
// Gibt den Inhalt des Buffers "Databuffer" wieder
$Bufferdata = $this->GetBuffer("DataBuffer");
```

# GetBufferList

Benötigt: IP-Symcon >= 5.0

```
array GetBufferList ()
```

## RÜCKGABEWERT

Namensliste aller Buffer

## BESCHREIBUNG

Diese Funktion liefert ein Array mit den Namen aller Buffer.

## BEISPIEL

```
// Gibt das Array der Buffernamen wieder
$BufferList = $this->GetBufferList();
```

# GetCompatibleParents

Benötigt: IP-Symcon >= 8.1

```
string GetCompatibleParents ()
```

## RÜCKGABEWERT

Empfohlener Typ und GUIDs der übergeordneten Instanz

## BESCHREIBUNG

TODO!

## BEISPIEL

```
// Benötige eine neue Instanz und bevorzuge einen Serial Port bei
der Erstellung
public function GetCompatibleParents() {

return '{"type": "require", "moduleIDs": ["{6DC3D946-0D31-450F-A8C6-
C42DB8D7D4F1}"]}';

}
```

# GetConfigurationForm

Benötigt: IP-Symcon >= 4.1

```
string GetConfigurationForm ()
```

## RÜCKGABEWERT

Inhalt der Konfigurationsseite

## BESCHREIBUNG

Der Inhalt kann überschrieben werden um so eine selbsterstellte Konfigurationsseite zu übergeben. So können dynamisch Inhalte wiedergespiegelt werden. Die "form.json" auf dem Dateisystem wird dabei vollkommen ignoriert.



Wird diese Funktion nicht im eigenen Modul definiert, so wird standarmässig der Inhalt der form.json weitergegeben.

## BEISPIEL

```
// IPSModuleStrict
public function GetConfigurationForm(): string {

return '{ "actions": [ { "type": "Label", "label": "The current time
is '.date("d.m.y H:i").'" } ] }';

}

// IPSModule
public function GetConfigurationForm() {

// Beispiel identisch. Bitte die geänderte Funktionssignatur
beachten.

}
```

# GetConfigurationForParent

Benötigt: IP-Symcon >= 4.2

```
string GetConfigurationForParent ()
```

## RÜCKGABEWERT

Inhalt der Konfiguration der übergeordenten Instanz

## BESCHREIBUNG

Der Konfiguration-String wird der übergeordeneten Instanz übergeben. Diese liest den String aus und setzt die eingetragenen Werte. Über diese Funktion eingelesene Werte können über die Konfigurationsseite nicht mehr verändert werden.



Wird diese Funktion nicht im eigenen Modul definiert, so wird standarmässig nichts verändert und die übergeordenete Instanz bleibt frei konfigurierbar.

## BEISPIEL

```
// IPSModuleStrict
public function GetConfigurationForParent(): string {

// Übergeordnete Instanz ist ein "SerialPort"
return "{\"BaudRate\": \"57600\", \"StopBits\": \"1\", \"DataBits\":
\"8\", \"Parity\": \"None\"}";

}

// IPSModule
public function GetConfigurationForParent() {

// Beispiel identisch. Bitte die geänderte Funktionssignatur
beachten.

}
```

# GetIDForIdent

Benötigt: IP-Symcon >= 4.0

```
variant GetIDForIdent (string $Ident)
```

## PARAMETERLISTE

| | |
|---|---|
| Ident | Ident des Objekts nach dem gesucht werden soll. |

## RÜCKGABEWERT

ID des gefundenen Objekts, sonst FALSE

## BESCHREIBUNG

Der Befehl sucht nach der ObjektID des Objekt mit dem Ident **Ident** und gibt diese zurück. Sollte der Ident nicht vorhanden sein, wird eine Fehlermeldung erzeugt.

## BEISPIEL

```
// Statusvariable mit dem Ident "Status" auf True setzen
SetValue($this->GetIDForIdent("Status"), true);
```

# GetMessageList

Benötigt: IP-Symcon >= 5.0

```
array GetMessageList ()
```

## RÜCKGABEWERT

Array aller aktiven Nachrichten

## BESCHREIBUNG

Diese Funktion liefert ein Array mit allen aktiven Nachrichten, welche über RegisterMessage registriert wurden.
Eine Liste der NachrichtIDs befindet sich hier: Nachrichten



Um ein eine Registrierung aufzuheben kann UnregisterMessage genutzt werden.

## BEISPIEL

```
// Gibt ein Array der aktiven registrierten Messages wieder
print_r($this->GetMessageList());

// Beispielausgabe
array(1) {
[12345]=> //ID der Instanz
array(1) {
[0]=>
int(10505) //NachrichtenID => IM_CHANGESTATUS
}
}
```

# GetReferenceList

Benötigt: IP-Symcon >= 5.1

```
array GetReferenceList ()
```

## RÜCKGABEWERT

Array mit Integer-Werten der IDs aller Referenzen

## BESCHREIBUNG

Diese Funktion liefert ein Array mit den IDs aller Referenzen.

## BEISPIEL

```
$ReferenceList = $this->GetReferenceList();
```

# GetStatus

Benötigt: IP-Symcon >= 5.1

```
int GetStatus ()
```

## RÜCKGABEWERT

Aktueller Status

## BESCHREIBUNG

Diese Funktion liefert den aktuellen Status der Instanz.
*Tabelle: Status der Instanz*

| WERT | STATUS |
|---|---|
| 101 | Instanz wird erstellt |
| 102 | Instanz ist aktiv |
| 103 | Instanz wird gelöscht |
| 104 | Instanz ist inaktiv |
| 105 | Instanz wurde nicht erstellt |
| 106 | Instanz ist im Standby |
| >=200 | Instanz ist fehlerhaft |

Für weitere Informationen zu Status Codes, siehe Nachrichten

## BEISPIEL

```
// Fragt den aktuellen Status der Instanz ab
$currentStatus = $this->GetStatus();
```

# GetTimerInterval

Benötigt: IP-Symcon >= 5.2

```
int GetTimerInterval (string $Name)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Der Name des Timers dessen Intervall abgefragt werden soll. |

## RÜCKGABEWERT

Aktuell eingestelltes Intervall in Millisekunden

## BESCHREIBUNG

Fragt den Intervall des Timers **Name** ab.



Wie in RegisterTimer beschrieben, ist der Timer stateless und bei einem Neustart vom IP-Symcon-Server muss der Timer neugesetzt werden, ansonsten nimmt er den Wert von RegisterTimer.

## BEISPIEL

```
// Fragt den Intervall des Timers "Update" ab
echo $this->GetTimerInterval("Update");
```

# GetValue

Benötigt: IP-Symcon >= 5.0

```
variant GetValue (string $Ident)
```

## PARAMETERLISTE

| | |
|---|---|
| Ident | Ident der Statusvariable |

## RÜCKGABEWERT

Der Inhalt der Statusvariable

## BESCHREIBUNG

Diese Funktion liefert den Wert der Statusvariable mit dem Ident **Ident**.

## BEISPIEL

```
// Gibt den Wert der Statusvariable "Statusvariable1" zurück
$StatusvariableValue = $this->GetValue("Statusvariable1");
```

# GetVisualizationTile

Benötigt: IP-Symcon >= 7.1

```
string GetVisualizationTile ()
```

## RÜCKGABEWERT

Initiale Darstellung einer Darstellung via HTML-SDK

## BESCHREIBUNG

Soll das HTML-SDK verwendet werden, so muss diese Funktion überschrieben werden, um den HTML-Inhalt zurückzugeben.



Zur einfacheren Bearbeitung lohnt es sich meist den konstanten Teil der Darstellung in einer HTML-Datei zu definieren anstatt es in dieser Funktion komplett auszuschreiben

## BEISPIEL

```
// IPSModuleStrict
public function GetVisualizationTile(): string {

// Gebe direkt den Inhalt der module.html im gleichen Verzeichnis
zurück
return file_get_contents('./module.html');

}

// IPSModule
public function GetVisualizationTile() {

// Beispiel identisch. Bitte die geänderte Funktionssignatur
beachten.

}
```

# HasActiveParent

Benötigt: IP-Symcon >= 5.1

```
boolean HasActiveParent ()
```

## RÜCKGABEWERT

TRUE, falls alle physikalisch übergeordneten Instanzen aktiv sind, andernfalls FALSE

## BESCHREIBUNG

Diese Funktion prüft, ob alle physikalisch übergeordneten Instanzen aktiv sind. Ist in der Kette mindestens eine Instanz nicht aktiv, so gibt die Funktion FALSE zurück. Sind alle Instanzen aktiv, so gibt die Funktion TRUE zurück.



Sollte kein Parent vorhanden sein, liefert die Funktion ebenfalls FALSE zurück.

## BEISPIEL

```
// Breche mit Fehlermeldung ab, falls Splitter oder I/O nicht bereit
sind
if (!$this->HasActiveParent()) {
echo "Fehler: Übergeordnete Instanzen sind nicht aktiv";
return;
}
```

# LogMessage

Benötigt: IP-Symcon >= 5.0

```
boolean LogMessage (string $Nachricht, int $Typ)
```

## PARAMETERLISTE

| | |
|---|---|
| Nachricht | Inhalt der Nachricht |

Typ

| TYP | WERT | BESCHREIBUNG |
|---|---|---|
| KL_DEBUG | 10206 | Eine Debugmeldung |
| KL_ERROR | 10205 | Eine Fehlermeldung |
| KL_MESSAGE | 10201 | Eine Standardnachricht |
| KL_NOTIFY | 10203 | Eine Benachrichtigung |
| KL_WARNING | 10204 | Eine Warnungsmeldung |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion schreibt unter Angabe des Typs **Typ** und der Nachricht **Nachricht** eine Nachricht ins Logfile. Zusätzlich werden diese im Meldungsfenster innerhalb der Konsole angezeigt.



Je nach Typ unterscheiden sich die Nachrichten in der Darstellung innerhalb des Meldungsfensters. (siehe Meldungen)



KL_DEBUG werden ausschließlich ins Log geschrieben. Bei Deaktivierung des Spezialschalter "LogfileVerbose" werden diese nichtmal ins Log geschrieben.



Der Nachrichtentyp KL_CUSTOM steht einzig der Funktion IPS_LogMessage zur Verfügung. Dieser kann nicht in Modulen verwendet werden.

## BEISPIEL

```
// Senden einer Warnungsmeldung im Meldungsfenster
$this->LogMessage("Dies ist eine Warnung", KL_WARNING);
```

# MaintainAction

Benötigt: IP-Symcon >= 4.0

```
boolean MaintainAction (string $Ident, boolean $AktionAktivieren)
```

## PARAMETERLISTE

| | |
|---|---|
| Ident | Ident der Variable |
| AktionAktivieren | Enable wenn **True**, Disable wenn **False** |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion ruft bei **AktionAktivieren** == True die Funktion EnableAction und bei **AktionAktivieren** == False die Funktion DisableAction mit dem Ident **Ident** auf.

## BEISPIEL

```
// Wir haben eine Aktion nur, wenn der Gerätetyp == 5 ist
$this->MaintainAction("SpecialData", $this-
>ReadPropertyInteger("DeviceType") == 5);
```

# MaintainVariable

Benötigt: IP-Symcon >= 4.0

```
variant MaintainVariable (string $Ident, string $Name, int $Typ,
variant $ProfilOderDarstellung, int $Position, boolean $Beibehalten)
```

## PARAMETERLISTE

| | |
|---|---|
| Ident | Ident der Statusvariable |
| Name | Name der Statusvariable |
| Typ | Typ der Statusvariable |
| ProfilOderDarstellung | Name des Variablenprofils oder Konfiguration der Darstellung welche verwendet werden soll |
| Position | Postion im Objektbaum und somit auch im WebFront |
| Beibehalten | Register wenn **True**, Unregister wenn **False** |

## RÜCKGABEWERT

| KLASSE | TYP | BESCHREIBUNG |
|---|---|---|
| IPSModuleStrict | boolean | Gibt zurück, ob die Variable erstellt wurde. Der Rückgabewert kann genutzt werden, um z.B. einen Initialwert zu setzen |
| IPSModule | integer | VariablenID der erstellten Statusvariable |

## BESCHREIBUNG

Wenn der Parameter **Beibehalten** == true ist, dann wird die Typ **Typ** passende Statusvariable mit dem Ident **Ident** und dem Namen **Name** erstellt.
Wenn der Parameter **Beibehalten** == false ist, dann wird die Statusvariable mit dem Ident **Ident** gelöscht.



Darstellung für Variablen gibt es seit der Version 8.0



Der Wert von **Typ** enthält den Variablentyp (0: Boolean, 1: Integer, 2: Float, 3: String). Siehe auch IPS_CreateVariable oder IPS_GetVariable

## BEISPIEL

```
// Wir haben diese Statusvariable nur, wenn der Gerätetyp == 5 ist

// IPSModuleStrict
$erstellt = $this->MaintainVariable("Status", "Status des Device",
3, "MeinVariablenProfilfuerStatus", 0, $this-
>ReadPropertyInteger("DeviceType") == 5);
if ($erstellt) {
// Initialer Wert soll true sein
$this->SetValue("Status", true);
}

// IPSModule
$variablenID = $this->MaintainVariable("Status", "Status des
Device", 3, "MeinVariablenProfilfuerStatus", 0, $this-
>ReadPropertyInteger("DeviceType") == 5);
```

# MessageSink

Benötigt: IP-Symcon >= 4.1

```
void MessageSink (int $Zeitstempel, int $SenderID, int $NachrichtID,
array $Daten)
```

## PARAMETERLISTE

| | |
|---|---|
| Zeitstempel | Fortlaufender Zähler Zeitstempel |
| SenderID | ID des Senders |
| NachrichtID | ID der Nachricht |
| Daten | Daten der Nachricht |

## RÜCKGABEWERT

Keine Rückgabe

## BESCHREIBUNG

Der Inhalt der Funktion kann überschrieben werden um eigene Reaktionen auf bestimmte Nachrichten auszuführen.
Die Funktion wird nur bei registrierten NachrichtenIDs/SenderIDs-Kombinationen aufgerufen.
Der Zeitstempel wird für jede versendete Nachricht inkrementiert.
RegisterMessage und UnregisterMessage fügt Filterregeln hinzu oder entfernt diese wieder.
Eine Liste der NachrichtIDs befindet sich hier: Nachrichten



Wird diese Funktion nicht im eigenen Modul definiert, so wird die Standard MessageSink() ausgeführt.



Der Inhalt von **Daten** kann je nach Nachrichtentyp varieren. Diese sind noch undokumentiert.

## BEISPIEL

```
// IPSModuleStrict
```

```
public function MessageSink(int $TimeStamp, int $SenderID, int
$Message, array $Data) {

IPS_LogMessage("MessageSink", "Message from SenderID ".$SenderID."
with Message ".$Message."\r\n Data: ".print_r($Data, true));
}

// IPSModule
public function MessageSink($TimeStamp, $SenderID, $Message, $Data)
{

// Beispiel identisch. Bitte die geänderte Funktionssignatur
beachten.

}
```

# Migrate

Benötigt: IP-Symcon >= 7.0

```
string Migrate (string $JSONData)
```

## PARAMETERLISTE

| | |
|---|---|
| JSONData | Persistenz (Konfiguration, Attribute) der Instanz |

## RÜCKGABEWERT

JSON-kodiertes Objekt mit der neuen Konfiguration/Attributen.
Leerer String, sofern keine Änderungen notwendig sind.

## BESCHREIBUNG

Diese Funktion wird nur beim Start von IP-Symcon oder nach eine Modul-Update aufgerufen. Sie wird nicht beim erstmaligem Erstellen einer Instanz aufgerufen. Die gelieferte Persistenz ist unabhängig von der bereits im Create registrierten Properties/ Attributen. Dadurch kann diese beliebig angepasst/korrigiert werden, sodass diese für die neuen Properties/Attribute passt. Dadurch können z.B. alte Properties (z.B. wegen einem Tippfehler) im Namen oder Inhalt korrigiert werden.
Die neuen Werte sind in den Properties/Attributen direkt im darauf folgenden ApplyChanges verfügbar.



Die Migrate Funktion wird von IP-Symcon aufgerufen. Sie muss somit von der Basisklasse überschrieben werden, um eigene Erweiterungen hinzuzufügen

## BEISPIEL

```
// IPSModuleStrict
public function Migrate(string $JSONData): string {

// Diese Zeile nicht entfernen
parent::Migrate($JSONData);

// Beispielinhalt von JSONData
/*
{
"attributes": {
```

```
"MeinAttribut": "MeinWert"
},
"configuration": {
"MeineKonfiguration": "MeinWert"
}
}
*/

// Eigenschaften/Attribute migrieren
$j = json_decode($JSONString);
$j->attributes->NeuesAttribut = $j->attributes->MeinAttribut;
$j->configuration->NeueKonfiguration = $j->configuration-
>MeineKonfiguration;
return json_encode($j);

}

// IPSModule
public function Migrate($JSONData) {

// Beispiel identisch. Bitte die geänderte Funktionssignatur
beachten.

}
```

# ReadAttributeBoolean

Benötigt: IP-Symcon >= 5.1

```
boolean ReadAttributeBoolean (string $Name)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name des Attributs |

## RÜCKGABEWERT

Der Wert des Attributs

## BESCHREIBUNG

Die Funktion gibt den Wert des Attributs mit dem Namen **Name** zurück.

Diese Funktion kann nicht in der Create Methode aufgerufen werden

Veränderte Werte sind im Vergleich zu Properties sofort verfügbar

## BEISPIEL

```
$this->ReadAttributeBoolean("CurrentState");
```

# ReadAttributeFloat

Benötigt: IP-Symcon >= 5.1

```
float ReadAttributeFloat (string $Name)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name des Attributs |

## RÜCKGABEWERT

Der Wert des Attributs

## BESCHREIBUNG

Die Funktion gibt den Wert des Attributs mit dem Namen **Name** zurück.



Diese Funktion kann nicht in der Create Methode aufgerufen werden



Veränderte Werte sind im Vergleich zu Properties sofort verfügbar

## BEISPIEL

```
$this->ReadAttributeFloat("Median");
```

# ReadAttributeInteger

Benötigt: IP-Symcon >= 5.1

```
int ReadAttributeInteger (string $Name)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name des Attributs |

## RÜCKGABEWERT

Der Wert des Attributs

## BESCHREIBUNG

Die Funktion gibt den Wert des Attributs mit dem Namen **Name** zurück.



Diese Funktion kann nicht in der Create Methode aufgerufen werden



Veränderte Werte sind im Vergleich zu Properties sofort verfügbar

## BEISPIEL

```
$this->ReadAttributeInteger("SequenceCounter")
```

# ReadAttributeString

Benötigt: IP-Symcon >= 5.1

```
string ReadAttributeString (string $Name)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name des Attributs |

## RÜCKGABEWERT

Der Wert des Attributs

## BESCHREIBUNG

Die Funktion gibt den Wert des Attributs mit dem Namen **Name** zurück.



Diese Funktion kann nicht in der Create Methode aufgerufen werden



Veränderte Werte sind im Vergleich zu Properties sofort verfügbar

## BEISPIEL

```
$this->ReadAttributeString("Token");
```

# ReadPropertyBoolean

Benötigt: IP-Symcon >= 4.0

```
boolean ReadPropertyBoolean (string $Name)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name der Eigenschaft |

## RÜCKGABEWERT

Der Wert der Eigenschaft

## BESCHREIBUNG

Die Funktion gibt den Wert der Eigenschaft mit dem Namen **Name** zurück.



Veränderte Werte sind erst verfügbar, wenn IPS_ApplyChanges aufgerufen wurde

## BEISPIEL

```
$this->ReadPropertyBoolean("EmulateStatus");
```

# ReadPropertyFloat

Benötigt: IP-Symcon >= 4.0

```
float ReadPropertyFloat (string $Name)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name der Eigenschaft |

## RÜCKGABEWERT

Der Wert der Eigenschaft

## BESCHREIBUNG

Die Funktion gibt den Wert der Eigenschaft mit dem Namen **Name** zurück.



Veränderte Werte sind erst verfügbar, wenn IPS_ApplyChanges aufgerufen wurde

## BEISPIEL

```
$this->ReadPropertyFloat("Faktor");
```

# ReadPropertyInteger

Benötigt: IP-Symcon >= 4.0

```
int ReadPropertyInteger (string $Name)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name der Eigenschaft |

## RÜCKGABEWERT

Der Wert der Eigenschaft

## BESCHREIBUNG

Die Funktion gibt den Wert der Eigenschaft mit dem Namen **Name** zurück.



Veränderte Werte sind erst verfügbar, wenn IPS_ApplyChanges aufgerufen wurde

## BEISPIEL

```
$this->ReadPropertyInteger("GatewayMode");
```

# ReadPropertyString

Benötigt: IP-Symcon >= 4.0

```
string ReadPropertyString (string $Name)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name der Eigenschaft |

## RÜCKGABEWERT

Der Wert der Eigenschaft

## BESCHREIBUNG

Die Funktion gibt den Wert der Eigenschaft mit dem Namen **Name** zurück.



Veränderte Werte sind erst verfügbar, wenn IPS_ApplyChanges aufgerufen wurde

## BEISPIEL

```
$this->ReadPropertyString("Benutzername");
```

# ReceiveData

Benötigt: IP-Symcon >= 4.0

```
string ReceiveData (string $JSONString)
```

## PARAMETERLISTE

| | |
|---|---|
| JSONString | Datenpaket im JSON Format |

## RÜCKGABEWERT

optionale Antwort an die übergordnete Instanz

## BESCHREIBUNG

Diese Funktion wird von IP-Symcon aufgerufen und verarbeitet zugesendete Daten und schickt diese gegenenfalls an alle untergeordneten Instanzen weiter.
Daten können über die Funktion SendDataToChildren versendet werden.
Weitere Informationen zur Datenweiterleitung können unter Datenfluss gefunden werden.



Die ReceiveData Funktion wird von IP-Symcon aufgerufen. Sie muss somit von der Basisklasse überschrieben werden, um eigene Erweiterungen hinzuzufügen



Bis IP-Symcon 5.3 hatte diese Funktion keinen Rückgabewert.

## BEISPIEL

```
// IPSModuleStrict
public function ReceiveData(string $JSONString): string {

// Beispiel innerhalb einer Gateway/Splitter Instanz

// Empfangene Daten vom I/O
$data = json_decode($JSONString);
IPS_LogMessage("ReceiveData", utf8_decode($data->Buffer));
```

```
// Hier werden die Daten verarbeitet

// Weiterleitung zu allen Gerät-/Device-Instanzen
$resultate = $this->SendDataToChildren(json_encode(Array("DataID" =>
"{66164EB8-3439-4599-B937-A365D7A68567}", "Buffer" => $data-
>Buffer)));

//Wenn eine untergeordnete Instanz ein Resultat liefert, kann dieses
genutzt werden.
foreach($resultate as $resultat) {
IPS_LogMessage("IOSplitter RECV-RES", $resultat);
}

}

// Beispiel innerhalb einer Geräte/Device Instanz
public function ReceiveData(string $JSONString): string {

// Empfangene Daten vom Gateway/Splitter
$data = json_decode($JSONString);
IPS_LogMessage("ReceiveData", utf8_decode($data->Buffer));

// Datenverarbeitung und schreiben der Werte in die Statusvariablen
SetValue($this->GetIDForIdent("Value"), $data->Buffer);

// Sende Resultat zurück an das/den Gateway/Splitter
return "OK von " . $this->InstanceID;

}

// IPSModule
public function ReceiveData($JSONString) {

// Beispiel identisch. Bitte die geänderte Funktionssignatur
beachten.

}
```

# RegisterAttributeBoolean

Benötigt: IP-Symcon >= 5.1

```
boolean RegisterAttributeBoolean (string $Name, boolean
$Standardwert)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name des Attributs |
| Standardwert | Standardwert des Attributs |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion erstellt ein Attribut vom Typ Boolean mit dem Namen **Name** und dem Standardwert **Standardwert**



Diese Funktion kann ausschließlich in der Create Methode aufgerufen werden!

## BEISPIEL

```
public function Create(): void {

// Diese Zeile nicht löschen oder ändern.
parent::Create();

$this->RegisterAttributeBoolean("CurrentState", true);

}
```

# RegisterAttributeFloat

Benötigt: IP-Symcon >= 5.1

```
boolean RegisterAttributeFloat (string $Name, float $Standardwert)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name des Attributs |
| Standardwert | Standardwert des Attributs |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion erstellt ein Attribut vom Typ Float mit dem Namen **Name** und dem Standardwert **Standardwert**



Diese Funktion kann ausschließlich in der Create Methode aufgerufen werden!

## BEISPIEL

```
public function Create(): void {

// Diese Zeile nicht löschen oder ändern.
parent::Create();

$this->RegisterAttributeFloat("Median", 0.5);

}
```

# RegisterAttributeInteger

Benötigt: IP-Symcon >= 5.1

```
boolean RegisterAttributeInteger (string $Name, int $Standardwert)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name des Attributs |
| Standardwert | Standardwert des Attributs |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion erstellt ein Attribut vom Typ Integer mit dem Namen **Name** und dem Standardwert **Standardwert**



Diese Funktion kann ausschließlich in der Create Methode aufgerufen werden!

## BEISPIEL

```
public function Create(): void {

// Diese Zeile nicht löschen oder ändern.
parent::Create();

$this->RegisterAttributeInteger("SequenceCounter", 0);

}
```

# RegisterAttributeString

Benötigt: IP-Symcon >= 5.1

```
boolean RegisterAttributeString (string $Name, string $Standardwert)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name des Attributs |
| Standardwert | Standardwert des Attributs |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion erstellt ein Attribut vom Typ String mit dem Namen **Name** und dem Standardwert **Standardwert**



Diese Funktion kann ausschließlich in der Create Methode aufgerufen werden!

## BEISPIEL

```
public function Create(): void {

// Diese Zeile nicht löschen oder ändern.
parent::Create();

$this->RegisterAttributeString("Token", "");

}
```

# RegisterMessage

Benötigt: IP-Symcon >= 4.1

```
boolean RegisterMessage (int $SenderID, int $NachrichtID)
```

## PARAMETERLISTE

| | |
|---|---|
| SenderID | ID des Senders |
| NachrichtID | ID der Nachricht |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion registriert das Modul für den Empfang der Nachricht **NachrichtID** und der ID **SenderID**.
Eine Liste der NachrichtIDs befindet sich hier: Nachrichten



Ein mehrfaches Aufrufen mit den selben Parametern registriert dennoch nur einmalig

## BEISPIEL

```
$this->RegisterMessage(12345 /* InstanzID */, 10505 /*
IM_CHANGESTATUS */);
```

# RegisterOnceTimer

Benötigt: IP-Symcon >= 5.5

```
boolean RegisterOnceTimer (string $Name, string $Skriptinhalt)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name des Timers |
| Skriptinhalt | PHP Skript ohne PHP Tags (<?php … ) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion erstellt einen Timer mit dem Namen **Name** und dem Inhalt **Skriptinhalt**. Der Timer wird einmalig und sofort ausgeführt.



Der Inhalt von **Skriptinhalt** wird jedes mal überschrieben, sodass auch nach Änderungen immer der korrekte Inhalt garantiert werden kann.

## BEISPIEL

```
// Erstellt einen Timer mit dem Namen "Update".
$this->RegisterOnceTimer("Update", "echo 'Hallo Welt';");
```

# RegisterPropertyBoolean

Benötigt: IP-Symcon >= 4.0

```
boolean RegisterPropertyBoolean (string $Name, boolean
$Standardwert)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name der Eigenschaft |
| Standardwert | Standardwert der Eigenschaft |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion erstellt eine Modul-Eigenschaft vom Typ Boolean mit dem Namen **Name** und dem Standardwert **Standardwert**



Diese Funktion kann ausschließlich in der Create Methode aufgerufen werden!

## BEISPIEL

```
public function Create(): void {

// Diese Zeile nicht löschen oder ändern.
parent::Create();

$this->RegisterPropertyBoolean("EmulateStatus", true);

}
```

# RegisterPropertyFloat

Benötigt: IP-Symcon >= 4.0

```
boolean RegisterPropertyFloat (string $Name, float $Standardwert)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name der Eigenschaft |
| Standardwert | Standardwert der Eigenschaft |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion erstellt eine Eigenschaft vom Typ Float mit dem Namen **Name** und dem Standardwert **Standardwert**



Diese Funktion kann ausschließlich in der Create Methode aufgerufen werden!

## BEISPIEL

```
public function Create(): void {

// Diese Zeile nicht löschen oder ändern.
parent::Create();

$this->RegisterPropertyFloat("Faktor", 0.5);

}
```

# RegisterPropertyInteger

Benötigt: IP-Symcon >= 4.0

```
boolean RegisterPropertyInteger (string $Name, int $Standardwert)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name der Eigenschaft |
| Standardwert | Standardwert der Eigenschaft |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion erstellt eine Eigenschaft vom Typ Integer mit dem Namen **Name** und dem Standardwert **Standardwert**



Diese Funktion kann ausschließlich in der Create Methode aufgerufen werden!

## BEISPIEL

```
public function Create(): void {

// Diese Zeile nicht löschen oder ändern.
parent::Create();

$this->RegisterPropertyInteger("GatewayMode", 0);

}
```

# RegisterPropertyString

Benötigt: IP-Symcon >= 4.0

```
boolean RegisterPropertyString (string $Name, string $Standardwert)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name der Eigenschaft |
| Standardwert | Standardwert der Eigenschaft |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion erstellt eine Eigenschaft vom Typ String mit dem Namen **Name** und dem Standardwert **Standardwert**



Diese Funktion kann ausschließlich in der Create Methode aufgerufen werden!

## BEISPIEL

```
public function Create(): void {

// Diese Zeile nicht löschen oder ändern.
parent::Create();

$this->RegisterPropertyString("Benutzername", "MaxMustermann");

}
```

# RegisterReference

Benötigt: IP-Symcon >= 5.1

```
boolean RegisterReference (int $ID)
```

## PARAMETERLISTE

| | |
|---|---|
| ID | ID des Objekts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion registriert die ID **ID** als referenziert, sodass das Modul bei einer Referenzsuche gefunden wird.



Ein mehrfaches Aufrufen mit der selben ID registriert dennoch nur einmalig

## BEISPIEL

```
$this->RegisterReference(10505 /* ObjektID */);
```

# RegisterScript

Benötigt: IP-Symcon >= 4.0

```
variant RegisterScript (string $Ident, string $Name, string $Inhalt,
int $Position)
```

## PARAMETERLISTE

| | |
|---|---|
| Ident | Ident des Skripts |
| Name | Name des Skripts |
| Inhalt | Inhalt, welcher in das Skript eingetragen werden soll.<br>**Default** == "<?php<br>//Autogenerated script" |
| Position | Position im Objektbaum und somit auch in der Visualisierung.<br>**Default** == 0 |

## RÜCKGABEWERT

| KLASSE | TYP | BESCHREIBUNG |
|---|---|---|
| IPSModule | integer | Gibt die ObjektID des erstellten oder bereits vorhandenen Skripts zurück. |
| IPSModuleStrict | boolean | Gibt zurück, ob das Script erstellt wurde. Der Rückgabewert kann genutzt werden, um z.B. initiale Konfigurationen vorzunehmen. |

## BESCHREIBUNG

Erstellt ein Skript mit dem Ident **Ident**, Namen **Name**, Inhalt **Inhalt** und der Position **Postion**, sofern der angegebene Ident nicht schon vorhanden ist.
In beiden Fällen wird die ObjektID des erstellten oder bereits vorhanden Skripts zurückgegeben.



Der Inhalt von **Inhalt** wird jedes mal überschrieben, sodass auch nach Änderungen immer der korrekte Inhalt garantiert werden kann. Für Skripte welche bearbeitet werden sollen, ist es empfohlen ein Konfigurationsformular mit SelectScript-Auswahl zu nutzen

## BEISPIEL

```
// IPSModuleStrict
$erstellt = $this->RegisterScript("TestSkript", "Mein TestSkript");
if ($erstellt) {
// Dinge tun, wenn Skript frisch erstellt wurde
}

// IPSModule
$skriptID = $this->RegisterScript("TestSkript", "Mein TestSkript");
```

# RegisterTimer

Benötigt: IP-Symcon >= 4.0

```
boolean RegisterTimer (string $Name, int $Intervall, string
$Skriptinhalt)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name des Timers |
| Intervall | Intervall in Millisekunden mit dem der Timer erstellt werden soll. 0 = Niemals |
| Skriptinhalt | PHP Skript ohne PHP Tags (<?php … ) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion erstellt einen Timer mit dem Namen **Name**. Es kann noch angegeben werden, welches **Intervall** beim Erstellen gesetzt werden soll und der **Skriptinhalt** wird definiert.
Der Timer ist stateless. Das heißt bei jedem Neustart wird der Timer mit dem hier eingestellten **Intervall** neuerstellt. Intervalleinstellungen müssen in der ApplyChanges() neu geladen werden.



Der Timer kann nur in der "Create()"-Funktion erstellt werden. Soll der Timer ausgeschaltet werden oder nicht starten: SetTimerInterval auf 0 setzen.



Der Inhalt von **Skriptinhalt** wird jedes mal überschrieben, sodass auch nach Änderungen immer der korrekte Inhalt garantiert werden kann.

## BEISPIEL

```
// Erstellt einen Timer mit dem Namen "Update" und einem Intervall
von 5 Sekunden.
```

```
$this->RegisterTimer("Update", 5000, "echo 'Hallo Welt';");
```

# RegisterVariableBoolean

Benötigt: IP-Symcon >= 4.0

```
variant RegisterVariableBoolean (string $Ident, string $Name, array
$Darstellung, int $Position)
```

## PARAMETERLISTE

| | |
|---|---|
| Ident | Ident der Statusvariable |
| Name | Name der Statusvariable |
| Darstellung | Die Konfiguration der Darstellung als Array. Ein Profil kann über das Legacy Profil gesetzt werden.<br>**Default** == "" |
| Position | Position im Objektbaum und somit auch in der Visualisierung.<br>**Default** == 0 |

## RÜCKGABEWERT

| KLASSE | TYP | BESCHREIBUNG |
|---|---|---|
| IPSModuleStrict | boolean | Gibt zurück, ob die Variable erstellt wurde. Der Rückgabewert kann genutzt werden, um z.B. einen initialen Wert zu setzen. |
| IPSModule | integer | VariablenID der erstellten Statusvariable. |

## BESCHREIBUNG

Diese Funktion erstellt eine Statusvariable vom Typ Boolean mit dem Ident **Ident** dem Namen **Name**. Es kann noch angegeben werden, welche **Darstellung** genutzt werden und auf welcher **Position** die Statusvariable stehen soll.
Der Aufbau der Darstellungen ist hier näher beschrieben.



Darstellungen können ab der Version 8.0 verwendet werden. In vorherigen Versionen können Profile über den **Presentation** Parameter als String gesetzt werden.



Diese Funktion kann auch aufgerufen werden wenn die Variable bereits existiert. In diesem Fall wird ausschließlich das die Darstellung aktualisiert, sofern diese abweichend ist. Sollte der Variablentyp anders sein, wird die Variable ohne Rückfrage gelöscht und neu mit dem korrektem Variablentyp neu erstellt.



Über GetIDForIdent kann die VariablenID ermittelt werden, sodass diese nirgends gespeichert werden muss.

## BEISPIEL

```
// IPSModuleStrict
$erstellt = $this->RegisterVariableBoolean("Schalter",
"Lichtschalter", ["PRESENTATION" => VARIABLE_PRESENTATION_SWITCH]);
if ($erstellt) {
// Initialen Wert setzen
$this->SetValue("Schalter", true);
}

// IPSModule
$variablenID = $this->RegisterVariableBoolean("Schalter",
"Lichtschalter", ["PRESENTATION" => VARIABLE_PRESENTATION_SWITCH]);
```

# RegisterVariableFloat

Benötigt: IP-Symcon >= 4.0

```
variant RegisterVariableFloat (string $Ident, string $Name, array
$Darstellung, int $Position)
```

## PARAMETERLISTE

| | |
|---|---|
| Ident | Ident der Statusvariable |
| Name | Name der Statusvariable |
| Darstellung | Die Konfiguration der Darstellung als Array. Ein Profil kann über das Legacy Profil gesetzt werden.<br>**Default** == "" |
| Position | Position im Objektbaum und somit auch in der Visualisierung.<br>**Default** == 0 |

## RÜCKGABEWERT

| KLASSE | TYP | BESCHREIBUNG |
|---|---|---|
| IPSModuleStrict | boolean | Gibt zurück, ob die Variable erstellt wurde. Der Rückgabewert kann genutzt werden, um z.B. einen initialen Wert zu setzen. |
| IPSModule | integer | VariablenID der erstellten Statusvariable. |

## BESCHREIBUNG

Diese Funktion erstellt eine Statusvariable vom Typ Float mit dem Ident **Ident** dem Namen **Name**. Es kann noch angegeben werden, welche **Darstellung** genutzt werden und auf welcher **Position** die Statusvariable stehen soll.
Der Aufbau der Darstellungen ist hier näher beschrieben.



Darstellungen können ab der Version 8.0 verwendet werden. In vorherigen Versionen können Profile über den **Presentation** Parameter als String gesetzt werden.



Diese Funktion kann auch aufgerufen werden wenn die Variable bereits existiert. In diesem Fall wird ausschließlich die Darstellung aktualisiert, sofern dieses abweichend ist. Sollte der Variablentyp anders sein, wird die Variable ohne Rückfrage gelöscht und neu mit dem korrektem Variablentyp neu erstellt.



Über GetIDForIdent kann die VariablenID ermittelt werden, sodass diese nirgends gespeichert werden muss.

## BEISPIEL

```
// IPSModuleStrict
$erstellt = $this->RegisterVariableFloat("Faktor", "Zoom Faktor
Wandanzeige", ["PRESENTATION"=> VARIABLE_PRESENTATION_SLIDER,
'SUFFIX' => ' %']);
if ($erstellt) {
// Initialen Wert setzen
$this->SetValue("Faktor", 5.8);
}

// IPSModule
$variablenID = $this->RegisterVariableFloat("Faktor", "Zoom Faktor
Wandanzeige", ["PRESENTATION"=> VARIABLE_PRESENTATION_SLIDER,
'SUFFIX' => ' %']);
```

# RegisterVariableInteger

Benötigt: IP-Symcon >= 4.0

```
variant RegisterVariableInteger (string $Ident, string $Name, array
$Darstellung, int $Position)
```

## PARAMETERLISTE

| | |
|---|---|
| Ident | Ident der Statusvariable |
| Name | Name der Statusvariable |
| Darstellung | Die Konfiguration der Darstellung als Array. Ein Profil kann über das Legacy Profil gesetzt werden.<br>**Default** == "" |
| Position | Position im Objektbaum und somit auch in der Visualisierung.<br>**Default** == 0 |

## RÜCKGABEWERT

| KLASSE | TYP | BESCHREIBUNG |
|---|---|---|
| IPSModuleStrict | boolean | Gibt zurück, ob die Variable erstellt wurde. Der Rückgabewert kann genutzt werden, um z.B. einen initialen Wert zu setzen. |
| IPSModule | integer | VariablenID der erstellten Statusvariable. |

## BESCHREIBUNG

Diese Funktion erstellt eine Statusvariable vom Typ Integer mit dem Ident **Ident** dem Namen **Name**. Es kann noch angegeben werden, welche **Darstellung** genutzt wird und auf welcher **Position** die Statusvariable stehen soll.
Der Aufbau der Darstellungen ist hier näher beschrieben.



Darstellungen können ab der Version 8.0 verwendet werden. In vorherigen Versionen können Profile über den **Presentation** Parameter als String gesetzt werden.



Diese Funktion kann auch aufgerufen werden wenn die Variable bereits existiert. In diesem Fall wird ausschließlich die Darstellung aktualisiert, sofern dieses abweichend ist. Sollte der Variablentyp anders sein, wird die Variable ohne Rückfrage gelöscht und neu mit dem korrektem Variablentyp neu erstellt.



Über GetIDForIdent kann die VariablenID ermittelt werden, sodass diese nirgends gespeichert werden muss.

## BEISPIEL

```
// IPSModuleStrict
$erstellt = $this->RegisterVariableInteger("Helligkeit", "Helligkeit
der Lampe", ["PRESENTATION"=> VARIABLE_PRESENTATION_SLIDER, 'SUFFIX'
=> ' lx']);
if ($erstellt) {
// Initialen Wert setzen
$this->SetValue("Helligkeit", 50);
}

// IPSModule
$variablenID = $this->RegisterVariableInteger("Helligkeit",
"Helligkeit der Lampe", ["PRESENTATION"=>
VARIABLE_PRESENTATION_SLIDER, 'SUFFIX' => ' lx']);
```

# RegisterVariableString

Benötigt: IP-Symcon >= 4.0

```
variant RegisterVariableString (string $Ident, string $Name, array
$Darstellung, int $Position)
```

## PARAMETERLISTE

| | |
|---|---|
| Ident | Ident der Statusvariable |
| Name | Name der Statusvariable |
| Darstellung | Die Konfiguration der Darstellung als Array. Ein Profil kann über das Legacy Profil gesetzt werden.<br>**Default** == "" |
| Position | Position im Objektbaum und somit auch in der Visualisierung.<br>**Default** == 0 |

## RÜCKGABEWERT

| KLASSE | TYP | BESCHREIBUNG |
|---|---|---|
| IPSModuleStrict | boolean | Gibt zurück, ob die Variable erstellt wurde. Der Rückgabewert kann genutzt werden, um z.B. einen initialen Wert zu setzen. |
| IPSModule | integer | VariablenID der erstellten Statusvariable. |

## BESCHREIBUNG

Diese Funktion erstellt eine Statusvariable vom Typ String mit dem Ident **Ident** dem Namen **Name**. Es kann noch angegeben werden, welche **Darstellung** genutzt werden und auf welcher **Position** die Statusvariable stehen soll.
Der Aufbau der Darstellungen ist hier näher beschrieben.



Darstellungen können ab der Version 8.0 verwendet werden. In vorherigen Versionen können Profile über den **Presentation** Parameter als String gesetzt werden.



Diese Funktion kann auch aufgerufen werden wenn die Variable bereits existiert. In diesem Fall wird ausschließlich die Darstellung aktualisiert, sofern dieses abweichend ist. Sollte der Variablentyp anders sein, wird die Variable ohne Rückfrage gelöscht und neu mit dem korrektem Variablentyp neu erstellt.



Über GetIDForIdent kann die VariablenID ermittelt werden, sodass diese nirgends gespeichert werden muss.

## BEISPIEL

```
// IPSModuleStrict
$erstellt = $this->RegisterVariableString("Name", "Mein Name",
["PRESENTATION"=> VARIABLE_PRESENTATION_VALUE_INPUT]);
if ($erstellt) {
// Initialen Wert setzen
$this->SetValue("Name", "Peter");
}

// IPSModule
$variablenID = $this->RegisterVariableString("Name", "Mein Name",
["PRESENTATION"=> VARIABLE_PRESENTATION_VALUE_INPUT]);
```

# ReloadForm

Benötigt: IP-Symcon >= 5.2

```
boolean ReloadForm ()
```

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis TRUE, andernfalls FALSE.

## BESCHREIBUNG

Diese Funktion lädt das Formular der Instanzkonfiguration in jeder offenen Instanzkonfiguration neu. Aktuelle Eingaben gehen dabei verloren.

## BEISPIEL

```
// Lade das Formular neu
$this->ReloadForm();
```

# RequestAction

```
void RequestAction (string $Ident, variant $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| Ident | Ident der Variable |
| Wert | Der Wert, welcher gesetzt werden soll |

## RÜCKGABEWERT

Keine Rückgabe

## BESCHREIBUNG

Wird aufgerufen wenn in der Visualisierung auf z.B. eine Schaltfläche geklickt wird.



Die RequestAction Funktion wird von IP-Symcon aufgerufen. Sie muss somit von der Basisklasse überschrieben werden, um eigene Erweiterungen hinzuzufügen

## BEISPIEL

```
// IPSModuleStrict
public function RequestAction(string $Ident, mixed $Value): void {

switch($Ident) {
case "TestVariable":
//Hier würde normalerweise eine Aktion z.B. das Schalten ausgeführt
werden
//Ausgaben über 'echo' werden an die Visualisierung zurückgeleitet

//Neuen Wert in die Statusvariable schreiben
SetValue($this->GetIDForIdent($Ident), $Value);
break;
default:
throw new Exception("Invalid Ident");
}

}

// IPSModule
```

```
public function RequestAction($Ident, $Value) {

// Beispiel identisch. Bitte die geänderte Funktionssignatur
beachten.

}
```

# RequireParent

Benötigt: IP-Symcon >= 4.0

```
boolean RequireParent (string $ÜbergordneteGUID)
```

## PARAMETERLISTE

| | |
|---|---|
| ÜbergordneteGUID | GUID |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG



Diese Funktion ist nur für IPSModule verfügbar. Die Alternative für IPSModuleStrict ist GetCompatibleParents.

Sofern keine Verbindung existiert, wird eine übergeordnete Instanz mit der GUID **ÜbergordneteGUID** erstellt. Selbst dann, wenn eine kompatible übergeordneten Instanz vorhanden wäre.



Der Normalfall für die Funktion ist ein Splitter-Modul, welches beim Erstellen eine eigene I/O Instanz benötigt. Alternativ ist eventuell auch ForceParent passend.



Diese Funktion wird normalerweise nur in der Create() oder ApplyChanges() Methode aufgerufen.

## BEISPIEL

```
// IPSModuleStrict
ConnectParent/RequireParent/ForceParent ist nicht verfügbar,
kann aber durch die neue Funktion "GetCompatibleParents()" ersetzt
werden.
```

```
// IPSModule
public function Create() {

// Die Zeile niemals entfernen!
parent::Create();

// Verbinde mit neu erstellten Splitter, falls noch keine Verbindung
besteht
$this->RequireParent("{46C969BF-3465-4E3E-B2A5-E404FB969735}");

}
```

# SendDataToChildren

Benötigt: IP-Symcon >= 4.0

```
array SendDataToChildren (string $Daten)
```

## PARAMETERLISTE

| | |
|---|---|
| Daten | JSON kodierter String |

## RÜCKGABEWERT

Ergebnisse, welches ReceiveData von den übergeordneten Instanzen liefert

## BESCHREIBUNG

Diese Funktion sendet Daten an alle verbundenen untergeordneten Instanzen.
Weitere Informationen sind unter Datenfluss zu finden.



Bis IP-Symcon 5.3 hatte diese Funktion keinen Rückgabewert.

## BEISPIEL

```
// Beispiel ist bei ReceiveData zu finden
```

# SendDataToParent

Benötigt: IP-Symcon >= 4.0

```
string SendDataToParent (string $Daten)
```

## PARAMETERLISTE

| | |
|---|---|
| Daten | JSON kodierter String |

## RÜCKGABEWERT

Ergebnis, welches ForwardData von der übergeordneten Instanz liefert

## BESCHREIBUNG

Diese Funktion sendet Daten an eine verbundene übergeordnete Instanz.
Weitere Informationen sind unter Datenfluss zu finden.

## BEISPIEL

```
// Beispiel ist bei ForwardData zu finden
```

# SendDebug

Benötigt: IP-Symcon >= 4.0

```
boolean SendDebug (string $Meldungsname, string $Daten, int $Format)
```

## PARAMETERLISTE

| | |
|---|---|
| Meldungsname | Name/Titel der Debugausgabe |
| Daten | Inhalt der Debugmeldung |
| Format | Voreinstellung für automatische Formationsauswahl (0 = Text, 1 = Hex) |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Gibt die Debugmeldung **Daten** unter dem Meldungsname **Meldungsname** und dem Format **Format** aus.
**Format** gibt dabei an ob bei automatischer Darstellung entweder Text oder Hex Ausgabe ausgewählt wird.

## BEISPIEL

```
// Beispiel aus der Anwesenheitssimulation
$this->SendDebug("Fetch", "Fetched day -".$day." with
".sizeof($data['Data'])." valid device(s)", 0);

// Beispielausgabe automatisch als Textausgabe im Debugfenster
konfiguriert
"Fetch" - "Fetched day -28 with 2 valid device(s)"
```

# SetBuffer

Benötigt: IP-Symcon >= 4.1

```
boolean SetBuffer (string $Name, string $Daten)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name des Buffers |
| Daten | Daten welche in den Buffer gepackt werden sollen. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion schreibt Daten **Daten** in den Buffer mit dem Namen **Name**.



Der Buffer besitzt ein Softlimt von 256kB und ein Hardlimit von 1024kB. Das heißt ab 256kB größe wirft IP-Symcon eine "Warning" in die Meldungen und die maximale Größe des Buffers beträgt 1024kB. Daten größer als 1024kB werden abgeschnitten.



Der Inhalt des Buffers ist nur solange vorhanden, wie IP-Symcon läuft. Mit einem Neustart verschwinden alle Informationen.

## BEISPIEL

```
// Schreibt "Hallo Welt" in den Buffer "Databuffer"
$this->SetBuffer("DataBuffer", "Hallo Welt");
```

# SetForwardDataFilter

Benötigt: IP-Symcon >= 4.1

```
boolean SetForwardDataFilter (string $ErforderlicheRegexRegel)
```

## PARAMETERLISTE

| | |
|---|---|
| ErforderlicheRegexRegel | Regexregel welche als Filter angewendet werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den Filter mit der Regular Expression **ErforderlicheRegexRegel**. ForwardData() wird nur aufgerufen wenn der gesetzte Filter zutrifft.

Zum Testen für den RegEx-Ausdruck kann die Seite www.regex101.com genutzt werden.

Bei bestimmten Sonderfällen kann es zu einer Änderung von Groß- zu Kleinschreibung beim PHP json_encode() kommen. So wird z.B. aus dem Sonderzeichen kodierten JSON_String ""\u001D"" nach einem json_decode und erneutem json_encode ""\u001d"".

## BEISPIEL

```
// Filter für ForwardData hinzufügen
public function ApplyChanges(): void {

[...]
$this->SetForwardDataFilter(".*Hallo.*");
[...]

}
```

```
// Wird nur aufgerufen, wenn "Hallo" im $JSONString gefunden wird
public function ForwardData(string $JSONString): string {

return "OK";

}
```

# SetReceiveDataFilter

Benötigt: IP-Symcon >= 4.1

```
boolean SetReceiveDataFilter (string $ErforderlicheRegexRegel)
```

## PARAMETERLISTE

| | |
|---|---|
| ErforderlicheRegexRegel | Regexregel welche als Filter angewendet werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den Filter mit der Regular Expression **ErforderlicheRegexRegel**. RecieveData() wird nur aufgerufen wenn der gesetzte Filter zutrifft.



Zum Testen für den RegEx-Ausdruck kann die Seite www.regex101.com genutzt werden.



Bei bestimmten Sonderfällen kann es zu einer Änderung von Groß- zu Kleinschreibung beim PHP json_encode() kommen. So wird z.B. aus dem Sonderzeichen kodierten JSON_String ""\u001D"" nach einem json_decode und erneutem json_encode ""\u001d"".

## BEISPIEL

```
// Filter für ReceiveData hinzufügen
public function ApplyChanges(): void {

[...]
$this->SetReceiveDataFilter(".*Hallo.*");
[...]

}
```

```
// Wird nur aufgerufen, wenn "Hallo" im $JSONString gefunden wird
public function ReceiveData(string $JSONString): string {

return "";

}
```

# SetStatus

Benötigt: IP-Symcon >= 4.0

```
boolean SetStatus (int $Statuswert)
```

## PARAMETERLISTE

| | |
|---|---|
| Statuswert | Der Statuswert auf den die Instanz gesetzt werden soll. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den Satus einer Instanz auf den Wert von **Statuswert**.
*Tabelle: Status der Instanz*

| WERT | STATUS |
|---|---|
| 101 | Instanz wird erstellt |
| 102 | Instanz ist aktiv |
| 103 | Instanz wird gelöscht |
| 104 | Instanz ist inaktiv |
| 105 | Instanz wurde nicht erstellt |
| 106 | Instanz ist im Standby |
| >=200 | Instanz ist fehlerhaft |

Ein Fehlerstatus über 200 kann selbst definiert werden.
Dies hat auch direkten Einfluss auf die Anzeige bei den Konfigurationsformularen im "status"-Bereich

## BEISPIEL

```
// Setzt des Status auf "inaktiv"
$this->SetStatus(104);
```

# SetSummary

Benötigt: IP-Symcon >= 4.1

```
boolean SetSummary (string $Kurzinfo)
```

## PARAMETERLISTE

| | |
|---|---|
| Kurzinfo | Inhalt der gesetzt werden soll |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion setzt die Kurzinformation **Kurzinfo** des Objekts, welche in der Verwaltungskonsole angezeigt wird und durch IPS_GetObject wieder ausgelesen werden kann.

## BEISPIEL

```
// Setzt eine mögliche IP-Adresse in die Kurzbeschreibung um eine
schnelle Wiedererkennbarkeit zu gewährleisten.
$this->SetSummary($this->ReadPropertyString("IPAddress"));
```

# SetTimerInterval

Benötigt: IP-Symcon >= 4.0

```
boolean SetTimerInterval (string $Name, int $Intervall)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Der Name des Timers dessen Intervall gesetzt werden soll. |
| Intervall | Der Intervall in Millisekunden auf den der Timer gesetzt werden soll. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Setzt den Intervall des Timers **Name** auf den Wert von **Interval**.

Wie in RegisterTimer beschrieben, ist der Timer stateless und bei einem Neustart vom IP-Symcon-Server muss der Timer neugesetzt werden, ansonsten nimmt er den Wert von RegisterTimer.

## BEISPIEL

```
// Setzt den Intervall des Timers "Update" auf 5 Sekunden
$this->SetTimerInterval("Update", 5000);
```

# SetValue

Benötigt: IP-Symcon >= 5.0

```
boolean SetValue (string $Ident, variant $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| Ident | Ident der Statusvariable |
| Wert | Wert, welcher in die Statusvariable geschrieben werden sollen. |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion schreibt den Wert **Wert** in die Statusvariable mit dem Ident **Ident**.

## BEISPIEL

```
// Schreibt 123 in die Statusvariable "Statusvariable1"
$this->SetValue("Statusvariable1", 123);
```

# SetVisualizationType

Benötigt: IP-Symcon >= 7.1

```
boolean SetVisualizationType (int $VisualisierungsTyp)
```

## PARAMETERLISTE

| | |
|---|---|
| VisualisierungsTyp | Visualisierungstyp für individuelle Visualisierung<br>**0**: keine individuelle Visualisierung<br>**1**: Visualisierung via HTML-SDK |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Mit dieser Funktion kann eine individuelle Darstellung der Instanz genutzt werden. Je nach angegebenem **VisualisierungsTyp** bietet die Instanz bei 0 keine individuelle Darstellung. Ansonsten wird eine individuelle Darstellung angeboten, welche bei 1 via HTML-SDK umgesetzt wird.

## BEISPIEL

```
// Aktiviere HTML-SDK
$this->SetVisualizationType(1);
```

# Translate

Benötigt: IP-Symcon >= 4.3

```
string Translate (string $Text)
```

## PARAMETERLISTE

| | |
|---|---|
| Text | Zu übersetzender Text |

## RÜCKGABEWERT

Übersetzter Text

## BESCHREIBUNG

Übersetzt den String **Text** anhand der ausgewählten Systemsprache und der locale.json.



Wenn keine locale.json vorhanden ist oder der String nicht in der locale.json gefunden werden kann, wird der ursprüngliche Text zurückgegeben.

## BEISPIEL

```
// module.php
$label = sprintf($this->Translate("The current time is %s"),
date("d.m.y H:i"));

// locale.json
{
"translations": {
"de": {
"The current time is %s": "Die aktuelle Zeit ist %s"
}
}
}
```

# UnregisterMessage

Benötigt: IP-Symcon >= 4.1

```
boolean UnregisterMessage (int $SenderID, int $NachrichtID)
```

## PARAMETERLISTE

| | |
|---|---|
| SenderID | ID des Senders |
| NachrichtID | ID der Nachricht |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion entfernt eine Registrierung des Moduls für den Empfang der Nachricht **NachrichtID** und der ID **SenderID**.
Eine Liste der NachrichtIDs befindet sich hier: Nachrichten

## BEISPIEL

```
// Das Modul "horcht" nicht mehr auf Nachrichten der Instanz 12345
mit der NachrichtID 10505
$this->UnregisterMessage(12345 /* InstanzID */, 10505 /*
IM_CHANGESTATUS */);

// Alle Nachrichten des Modules sollen gelöscht werden
foreach ($this->GetMessageList() as $senderID => $messages) {
foreach ($messages as $message) {
$this->UnregisterMessage($senderID, $message);
}
}
```

# UnregisterReference

Benötigt: IP-Symcon >= 5.1

```
boolean UnregisterReference (int $ID)
```

## PARAMETERLISTE

| | |
|---|---|
| ID | ID des Objekts |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion entfernt die ID **ID** als referenziert für das Modul.

## BEISPIEL

```
$this->UnregisterReference(10505 /* ObjektID */);
```

# UnregisterVariable

Benötigt: IP-Symcon >= 4.0

```
boolean UnregisterVariable (string $Ident)
```

## PARAMETERLISTE

| | |
|---|---|
| Ident | Ident der Statusvariable |

## RÜCKGABEWERT

| KLASSE | TYP | BESCHREIBUNG |
|---|---|---|
| IPSModuleStrict | boolean | Gibt zurück, ob die Variable entfernt wurde. Der Rückgabewert kann genutzt werden, um z.B. weitere Dinge aufzuräumen. |
| IPSModule | boolean | Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**. |

## BESCHREIBUNG

Der Befehl löscht die Statusvariable mit dem Ident **Ident**

## BEISPIEL

```
$this->UnregisterVariable("Temperatur");
```

# UpdateFormField

Benötigt: IP-Symcon >= 5.2

```
boolean UpdateFormField (string $Feld, string $Parameter, variant
$Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| Feld | Name des zu ändernden Formularfeldes |
| Parameter | Name des zu ändernden Parameters |
| Wert | Neuer Wert des Parameters |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Dies Funktion setzt den Parameter **Parameter** das Formularfeldes mit dem Namen **Feld** in jeder offenen Instanzkonfiguration dieser Instanz auf den Wert **Wert**. Sollte ein Formularfeld mit diesem Namen im Element- und im Aktionsbereich existieren, so werden beide aktualisiert.

Es ist nicht möglich die Parameter **name** und **type** von Formularfeldern zu verändern. Weitere nicht veränderbare Parameter sind in der jeweiligen Dokumentation aufgeführt.

Bei komplexen Parameter, also Listen oder Objekte, muss der neue Wert JSON-codiert übergeben werden.

## BEISPIEL

```
// Verstecke den Button
$this->UpdateFormField("MyButton", "visible", false);
```

# UpdateVisualizationValue

Benötigt: IP-Symcon >= 7.1

```
boolean UpdateVisualizationValue (variant $Daten)
```

## PARAMETERLISTE

| | |
|---|---|
| Daten | beliebige Daten, welche an die Visualisierung geschickt werden |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Diese Funktion schickt **Daten** an die Visualisierung via HTML-SDK. Die Visualisierung muss zum Empfangen wiederum die Funktion handleMessage implementiert haben. Diese Funktion wird dann mit **Daten** als Parameter aufgerufen.



Es existiert keine feste Definition, wie die Daten codiert werden müssen. Somit kann bei der Entwicklung eines Moduls eine eigene Definition gewählt werden, welche gut zum Modul passt

## BEISPIEL

```
// In diesem Beispiel wird eine einfache Zahl als Zählerstand
übergeben
$this->UpdateVisualizationValue(5);
```

# WriteAttributeBoolean

Benötigt: IP-Symcon >= 5.1

```
boolean WriteAttributeBoolean (string $Name, boolean $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name des Attributs |
| Wert | Wert des Attributs |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion schreibt den Wert **Wert** in das Attribut mit dem Namen **Name**.



Diese Funktion kann nicht in der Create Methode aufgerufen werden



Veränderte Werte eines Attributs sind im Vergleich zu Properties sofort verfügbar

## BEISPIEL

```
$this->WriteAttributeBoolean("CurrentState", false);
```

# WriteAttributeFloat

Benötigt: IP-Symcon >= 5.1

```
boolean WriteAttributeFloat (string $Name, float $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name des Attributs |
| Wert | Wert des Attributs |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion schreibt den Wert **Wert** in das Attribut mit dem Namen **Name**.



Diese Funktion kann nicht in der Create Methode aufgerufen werden



Veränderte Werte eines Attributs sind im Vergleich zu Properties sofort verfügbar

## BEISPIEL

```
$this->WriteAttributeFloat("Median", 5.5);
```

# WriteAttributeInteger

Benötigt: IP-Symcon >= 5.1

```
boolean WriteAttributeInteger (string $Name, int $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name des Attributs |
| Wert | Wert des Attributs |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion schreibt den Wert **Wert** in das Attribut mit dem Namen **Name**.



Diese Funktion kann nicht in der Create Methode aufgerufen werden



Veränderte Werte eines Attributs sind im Vergleich zu Properties sofort verfügbar

## BEISPIEL

```
$this->WriteAttributeInteger("SequenceCounter", 4);
```

# WriteAttributeString

Benötigt: IP-Symcon >= 5.1

```
boolean WriteAttributeString (string $Name, string $Wert)
```

## PARAMETERLISTE

| | |
|---|---|
| Name | Name des Attributs |
| Wert | Wert des Attributs |

## RÜCKGABEWERT

Konnte der Befehl erfolgreich ausgeführt werden, liefert er als Ergebnis **TRUE**, andernfalls **FALSE**.

## BESCHREIBUNG

Die Funktion schreibt den Wert **Wert** in das Attribut mit dem Namen **Name**.



Diese Funktion kann nicht in der Create Methode aufgerufen werden



Veränderte Werte eines Attributs sind im Vergleich zu Properties sofort verfügbar

## BEISPIEL

```
$this->WriteAttributeString("Token",
"08da50bd109c7fb1bec49d15ae86e55f");
```

# Nachrichten

Benötigt: IP-Symcon >= 4.1

## BESCHREIBUNG

Alle Nachrichten haben eine eigene ID, welche die Art der Nachricht beschreibt.
Dies ist eine fünfstellige Zahl und setzt sich wiefolgt zusammen:
Basis + Kategorie + Typ = Gesamtwert



Ab Version 5.0 stehen die Werte automatisch als Konstante zur Verfügung

## ÜBERSICHT ALLER NACHRICHTIDS

## NACHRICHTENBASIS

Nachrichten von IP-Symcon haben die Wertebasis 10000, solche von Modulinstanzen (noch undokumentiert) die Wertebasis 20000.

| DEFINE | WERT | BESCHREIBUNG |
|---|---|---|
| IPS_BASE | 10000 | Wertebasis für Kernel |
| IPS_MODULBASE | 20000 | Wertebasis für Module |

## NACHRICHTENKATEGORIEN

| DEFINE | WERT | BESCHREIBUNG |
|---|---|---|
| IPS_BASE | 0 | Service Nachrichten |
| IPS_KERNELMESSAGE | 100 | Kernel Manager |
| IPS_LOGMESSAGE | 200 | Meldungen Manager |
| IPS_MODULEMESSAGE | 300 | Modul Manager |
| IPS_OBJECTMESSAGE | 400 | Objekt Manager |
| IPS_INSTANCEMESSAGE | 500 | Instanz Manager |
| IPS_SEARCHMESSAGE | 510 | Such Manager |
| IPS_VARIABLEMESSAGE | 600 | Variablen Manager |
| IPS_SCRIPTMESSAGE | 700 | Script Manager |
| IPS_EVENTMESSAGE | 800 | Event Manager |
| IPS_MEDIAMESSAGE | 900 | Medien Manager |
| IPS_LINKMESSAGE | 1000 | Link Manager |
| IPS_FLOWMESSAGE | 1100 | Flow Manager |
| IPS_ENGINEMESSAGE | 1200 | Script Engine |
| IPS_PROFILEMESSAGE | 1300 | Profile Pool |
| IPS_TIMERMESSAGE | 1400 | Timer Pool |
| IPS_ACTIONMESSAGE | 1500 | Action Pool |
| IPS_LICENSENMESSAGE | 1600 | License Pool |

## NACHRICHTENTYPEN

### IPS_BASE

Wertebasis 0

| DEFINE | WERT | GESAMTWERT | BESCHREIBUNG |
|---|---|---|---|
| IPS_KERNELSTARTED (ab 4.2) | 1 | 10001 | Wird nach KR_READY gesendet und synchron abgearbeitet |
| IPS_KERNELSHUTDOWN (ab 4.2) | 2 | 10002 | Wird vor KR_UNINIT gesendet und synchron abgearbeitet |

### IPS_KERNELMESSAGE

Wertebasis 100

| DEFINE | WERT | GESAMTWERT | BESCHREIBUNG |
|---|---|---|---|
| KR_CREATE | 1 | 10101 | Kernel wurde erstellt |
| KR_INIT | 2 | 10102 | Kernel Komponenten werden initialisiert, Module geladen und Settings eingelesen |
| KR_READY | 3 | 10103 | Kernel ist bereit und läuft |
| KR_UNINIT | 4 | 10104 | "Shutdown"-Befehl erhalten, finalisiere alles geladene |
| KR_SHUTDOWN | 5 | 10105 | Finalisierung abgeschlossen, entferne Kernel |

## IPS_LOGMESSAGE

Wertebasis 200

| DEFINE | WERT | GESAMTWERT | BESCHREIBUNG |
|---|---|---|---|
| KL_MESSAGE | 1 | 10201 | Normale Nachricht |
| KL_SUCCESS | 2 | 10202 | Erfolg |
| KL_NOTIFY | 3 | 10203 | Änderungsbenachrichtung |
| KL_WARNING | 4 | 10204 | Warnung |
| KL_ERROR | 5 | 10205 | Fehlermeldung |
| KL_DEBUG | 6 | 10206 | Debug Information |
| KL_CUSTOM | 7 | 10207 | Sonstige Nachrichten |

## IPS_MODULEMESSAGE

Wertebasis 300

| DEFINE | WERT | GESAMTWERT | BESCHREIBUNG |
|---|---|---|---|
| ML_LOAD | 1 | 10301 | Modul geladen |
| ML_UNLOAD | 2 | 10302 | Modul entladen |

## IPS_OBJECTMESSAGE

Wertebasis 400

| DEFINE | WERT | GESAMTWERT | BESCHREIBUNG |
|---|---|---|---|
| OM_REGISTER | 1 | 10401 | Objekt erstellt |
| OM_UNREGISTER | 2 | 10402 | Objekt entfernt |
| OM_CHANGEPARENT | 3 | 10403 | Übergeordnetes Objekt hat sich geändert |
| OM_CHANGENAME | 4 | 10404 | Name hat sich geändert |
| OM_CHANGEINFO | 5 | 10405 | Beschreibung hat sich geändert |
| OM_CHANGETYPE | 6 | 10406 | Typ hat sich geändert |
| OM_CHANGESUMMARY | 7 | 10407 | Kurzinfo hat sich geändert |
| OM_CHANGEPOSITION | 8 | 10408 | Position hat sich geändert |
| OM_CHANGEREADONLY | 9 | 10409 | "Nur-Lesen"-Status hat sich geändert |
| OM_CHANGEHIDDEN | 10 | 10410 | Sichtbarkeit hat sich geändert |
| OM_CHANGEICON | 11 | 10411 | Icon hat sich geändert |
| OM_CHILDADDED | 12 | 10412 | Untergeordnetes Objekt hinzugefügt |
| OM_CHILDREMOVED | 13 | 10413 | Untergeordnetes Objekt entfernt |
| OM_CHANGEIDENT | 14 | 10414 | Ident hat sich geändert |
| OM_CHANGEDISABLED | 15 | 10415 | Bedienbarkeit hat sich geändert |
| OM_CHANGELOCKED | 16 | 10416 | Sperre hat sich geändert |

## IPS_INSTANCEMESSAGE

Wertebasis 500

| DEFINE | WERT | GESAMTWERT | BESCHREIBUNG |
|---|---|---|---|
| IM_CREATE | 1 | 10501 | Instanz erstellt |
| IM_DELETE | 2 | 10502 | Instanz entfernt |
| IM_CONNECT | 3 | 10503 | Instanzinterface verfügbar |
| IM_DISCONNECT | 4 | 10504 | Instanzinterface nicht mehr verfügbar |
| IM_CHANGESTATUS | 5 | 10505 | Status hat sich geändert |
| IM_CHANGESETTINGS | 6 | 10506 | Einstellungen haben sich geändert |
| IM_CHANGEATTRIBUTE | 7 | 10507 | Attribut hat sich geändert |
| IM_ADDATTRIBUTE | 8 | 10508 | Attribut wurde hinzugefügt |
| IM_REMOVEATTRIBUTE | 9 | 10509 | Attribut wurde entfernt |

## IPS_SEARCHMESSAGE

Wertebasis 510

| DEFINE | WERT | GESAMTWERT | BESCHREIBUNG |
|---|---|---|---|
| IM_SEARCHSTART | 1 | 10511 | Suche wurde gestartet |
| IM_SEARCHSTOP | 2 | 10512 | Suche wurde gestoppt |
| IM_SEARCHUPDATE | 3 | 10513 | Suche hat neue Ergebnisse |

### IPS_FORMMESSAGE

Wertebasis 520

| DEFINE | VALUE | TOTAL VALUE | DESCRIPTION |
|---|---|---|---|
| IM_FORMFIELDCREATE | 1 | 10521 | Formularfeld erstellt |
| IM_FORMFIELDDELETE | 2 | 10522 | Formularfeld entfernt |
| IM_FORMFIELDUPDATE | 3 | 10523 | Formularfeld hat sich geändert |
| IM_FORMRELOAD | 4 | 10524 | Lade Formularfeld erneut |

### IPS_VARIABLEMESSAGE

Wertebasis 600

| DEFINE | WERT | GESAMTWERT | BESCHREIBUNG |
|---|---|---|---|
| VM_CREATE | 1 | 10601 | Variable wurde erstellt |
| VM_DELETE | 2 | 10602 | Variable wurde entfernt |
| VM_UPDATE | 3 | 10603 | Variable wurde aktualisiert |
| VM_CHANGEPROFILENAME | 4 | 10604 | Variablenprofilname wurde geändert |
| VM_CHANGEPROFILEACTION | 5 | 10605 | Variablenprofilaktion wurde geändert |
| VM_CHANGEDLOCKED | 6 | 10606 | Sperre wurde geändert |

### IPS_SCRIPTMESSAGE

Wertebasis 700

| DEFINE | WERT | GESAMTWERT | BESCHREIBUNG |
|---|---|---|---|
| SM_CREATE | 1 | 10701 | Skript wurde erstellt |
| SM_DELETE | 2 | 10702 | Skript wurde entfernt |
| SM_CHANGEFILE | 3 | 10703 | Skript wurde Datei angehangen |
| SM_BROKEN | 4 | 10704 | Skript Fehlerstatus hat sich geändert |
| SM_UPDATE | 5 | 10704 | Skript wurde aktualisiert |

### IPS_EVENTMESSAGE

Wertebasis 800

| DEFINE | WERT | GESAMTWERT | BESCHREIBUN |
|---|---|---|---|
| EM_CREATE | 1 | 10801 | Ereignis wurde erste |
| EM_DELETE | 2 | 10802 | Ereignis wurde entfe |
| EM_UPDATE | 3 | 10803 | Ereignis wurde aktua |
| EM_CHANGEACTIVE | 4 | 10804 | Ereignisaktivierung h geändert |
| EM_CHANGELIMIT | 5 | 10805 | Ereignisaufruflimit h geändert |
| EM_CHANGESCRIPT | 6 | 10806 | Ereignisskriptinhalt h geändert |
| EM_CHANGETRIGGER | 7 | 10807 | Ereignisauslöser hat geändert |
| EM_CHANGETRIGGERVALUE | 8 | 10808 | Ereignisgrenzwert ha geändert |
| EM_CHANGETRIGGEREXECUTION | 9 | 10809 | Ereignisgrenzwertau hat sich geändert |
| EM_CHANGECYCLIC | 10 | 10810 | zyklisches Ereignis ha geändert |
| EM_CHANGECYCLICDATEFROM | 11 | 10811 | Startdatum hat sich geändert |
| EM_CHANGECYCLICDATETO | 12 | 10812 | Enddatum hat sich g |
| EM_CHANGECYCLICTIMEFROM | 13 | 10813 | Startzeit hat sich geä |
| EM_CHANGECYCLICTIMETO | 14 | 10814 | Endzeit hat sich geär |
| EM_ADDSCHEDULEACTION | 15 | 10815 | Eintrag in der Aktionstabelle des Wochenplans wurde hinzugefügt |
| EM_REMOVESCHEDULEACTION | 16 | 10816 | Eintrag in der Aktionstabelle des Wochenplans wurde entfernt |
| EM_CHANGESCHEDULEACTION | 17 | 10817 | Eintrag in der Aktionstabelle des Wochenplans hat sic geändert |
| EM_ADDSCHEDULEGROUP | 18 | 10818 | Gruppierung der Wochenplantage wu hinzugefügt |
| EM_REMOVESCHEDULEGROUP | 19 | 10819 | Gruppierung der Wochenplantage wu entfernt |
| EM_CHANGESCHEDULEGROUP | 20 | 10820 | Gruppierung der Wochenplantage hat geändert |
| EM_ADDSCHEDULEGROUPPOINT | 21 | 10821 | Schaltpunkt einer Gruppierung wurde hinzugefügt |
| | | | Schaltpunkt eines |

## IPS_MEDIAMESSAGE

Wertebasis 900

| DEFINE | WERT | GESAMTWERT | BESCHREIBUNG |
|---|---|---|---|
| MM_CREATE | 1 | 10901 | Medienobjekt wurde erstellt |
| MM_DELETE | 2 | 10902 | Medienobjekt wurde entfernt |
| MM_CHANGEFILE | 3 | 10903 | Datei des Medienobjekts wurde geändert |
| MM_AVAILABLE | 4 | 10904 | Verfügbarkeit des Medienobjekts hat sich geändert |
| MM_UPDATE | 5 | 10905 | Medienobjekt wurde aktualisiert |
| MM_CHANGECACHED | 6 | 10906 | Cacheoption vom Medienobjekt hat sich geändert |

## IPS_LINKMESSAGE

Wertebasis 1000

| DEFINE | WERT | GESAMTWERT | BESCHREIBUNG |
|---|---|---|---|
| LM_CREATE | 1 | 11001 | Link wurde erstellt |
| LM_DELETE | 2 | 11002 | Link wurde entfernt |
| LM_CHANGETARGET | 3 | 11003 | Ziel des Links hat sich geändert |

## IPS_FLOWMESSAGE

Wertebasis 1100

| DEFINE | WERT | GESAMTWERT | BESCHREIBUNG |
|---|---|---|---|
| FM_CONNECT | 1 | 11101 | Instanz wurde verbunden |
| FM_DISCONNECT | 2 | 11102 | Instanz wurde getrennt |
| FM_CHILDADDED | 3 | 11103 | Untergeordnete Instanz wurde mit dieser Instanz verbunden |
| FM_CHILDREMOVED | 4 | 11104 | Untergeordnete Instanz wurde von dieser Instanz getrennt |

## IPS_ENGINEMESSAGE

Wertebasis 1200

| DEFINE | WERT | GESAMTWERT | BESCHREIBUNG |
|---|---|---|---|
| SE_UPDATE | 1 | 11201 | Scriptengine wurde neu geladen |
| SE_EXECUTE | 2 | 11202 | Script wurde ausgeführt |
| SE_RUNNING | 3 | 11203 | Script wird ausgeführt |
| SE_FLOWSCRIPT | 4 | 11204 | Bei Ablaufplan Status (entweder alle Schritte abgeschlossen oder abgebrochen) |

## IPS_PROFILEMESSAGE

Wertebasis 1300

| DEFINE | WERT | GESAMTWERT | BESCHREIBUNG |
|---|---|---|---|
| PM_CREATE | 1 | 11301 | Profil wurde erstellt |
| PM_DELETE | 2 | 11302 | Profil wurde entfernt |
| PM_CHANGETEXT | 3 | 11303 | Profilprefix/Profilsuffix hat sich geändert |
| PM_CHANGEVALUES | 4 | 11304 | Profilwerte haben sich geändert |
| PM_CHANGEDIGITS | 5 | 11305 | Profilnachkommastellen haben sich geändert |
| PM_CHANGEICON | 6 | 11306 | Profilicon hat sich geändert |
| PM_ASSOCIATIONADDED | 7 | 11307 | Profilassoziation wurde hinzugefügt |
| PM_ASSOCIATIONREMOVED | 8 | 11308 | Profilassoziation wurde entfernt |
| PM_ASSOCIATIONCHANGED | 9 | 11309 | Profilassoziation hat sich geändert |

### IPS_TIMERMESSAGE

Wertebasis 1400

| DEFINE | WERT | GESAMTWERT | BESCHREIBUNG |
|---|---|---|---|
| TM_REGISTER | 1 | 11401 | Timer wurde erstellt |
| TM_UNREGISTER | 2 | 11402 | Timer wurde entfernt |
| TM_CHANGEINTERVAL | 3 | 11403 | Timer Interval hat sich geändert |

### IPS_ACTIONMESSAGE

Wertebasis 1500

| DEFINE | WERT | GESAMTWERT | BESCHREIBUNG |
|---|---|---|---|
| AM_FORMFIELDCREATE | 1 | 11501 | Aktuell ungenutzt |
| AM_FORMFIELDDELETE | 2 | 11502 | Aktuell ungenutzt |
| AM_FORMFIELDUPDATE | 3 | 11503 | Form-Element benötigt ein Update |
| AM_FORMRELOAD | 4 | 11504 | Konfigurationsformular neuladen |

### IPS_LICENSEMESSAGE

Wertebasis 1600

| DEFINE | WERT | GESAMTWERT | BESCHREIBUNG |
|---|---|---|---|
| LP_UPDATELICENSE | 1 | 11601 | Lizenz wurde geändert |
| LP_UPDATESUBSCRIPTION | 2 | 11602 | Subskription wurde aktualisiert |

## NACHRICHTENPARAMETER

Werte von verschiedenen Nachrichtenparametern

## SEARCH HANDLING

| DEFINE | WERT | GESAMTWERT | BESCHREIBUNG |
|---|---|---|---|
| IF_UNKNOWN | 0 | 0 | Unbekannter Wert |
| IF_NEW | 1 | 1 | Gerät ist erstellt aber nicht konfiguriert |
| IF_OLD | 2 | 2 | Gerät ist konfiguriert und sollte eine InstanzID haben |
| IF_CURRENT | 3 | 3 | Gerät ist konfiguriert und die InstanzID gehört dem suchenden Gerät |
| IF_UNSUPPORTED | 4 | 4 | Gerät wird vom Modul nicht unterstützt |

## STATUS CODES

| DEFINE | WERT | GESAMTWERT | BESCHREIBUNG |
|---|---|---|---|
| IS_SBASE | 100 | 100 | Wertebasis für Status Codes |
| IS_CREATING | 1 | 101 | Instanz wird erstellt |
| IS_ACTIVE | 2 | 102 | Instanz ist aktiv |
| IS_DELETING | 3 | 103 | Instanz wird gelöscht |
| IS_INACTIVE | 4 | 104 | Instanz ist inaktiv |
| IS_NOTCREATED | 5 | 105 | Instanz wurde nicht erstelltv |

## ERROR CODES

| DEFINE | WERT | GESAMTWERT | BESCHREIBUNG |
|---|---|---|---|
| IS_EBASE | 200 | 200 | Allgemeiner Fehlerfall |

# Referenzen

Benötigt: IP-Symcon >= 5.1

## BESCHREIBUNG

Referenzen zeigen an, welche Objekte einen selbst zur fehlerfreien Funktionsweise benötigen. So wird zum Beispiel bei einer Referenzsuche auf eine Variable ein ausgelöstes Ereignis aufgeführt, welches bei Variablenänderung auslöst. Ohne die Variable kann das Ereignis nicht ordentlich funktionieren.
Abgerufen werden die Referenzen standardmässig über das Kontextmenü im Objektbaum. In der dann angezeigten Liste kann dann zu den Objekten gesprungen werden.




## REFERENZEN VERWALTEN

Im Modul können Referenzen hinzufügt und entfernt werden.

| FUNKTION | BESCHREIBUNG |
|---|---|
| GetReferenceList | Diese Funktion liefert ein Array mit allen Referenzen. |
| RegisterReference | Diese Funktion registriert mithilfe der ZielobjektID eine Referenz, welches dann bei einer Referenzsuche gefunden wird. |
| UnregisterReference | Diese Funktion entfernt eine Referenz für das Modul. |

# Store

Ab der Version 5.1 gibt es in IP-Symcon einen Module Store, in welchem entwickelte Module angeboten werden können. Die Grundlage hierfür ist eine Bibliothek, welche komfortabel gefunden und installiert werden kann.



Allgemeine Informationen zum Module Store kann man hier finden:
Module Store

Für jede Bibliothek können bis zu drei verschiedene Kanäle angeboten werden: Stable, Beta und Testing

**Stable** ist der Kanal für die stabile Produktivversion der Bibliothek. Veröffentlichungen auf diesem Kanal werden durch das Symcon-Team geprüft.
**Beta** ist der offene Testkanal. Jeder Benutzer hat hierauf Zugriff und kann die Beta-Version installieren.
**Testing** beschreibt einen internen Testkanal. Hierzu müssen Benutzer explizit eingeladen werden, damit diese Zugriff auf den Kanal haben. Ist man nicht eingeladen, so erhält man keinen Einblick in diese Version.

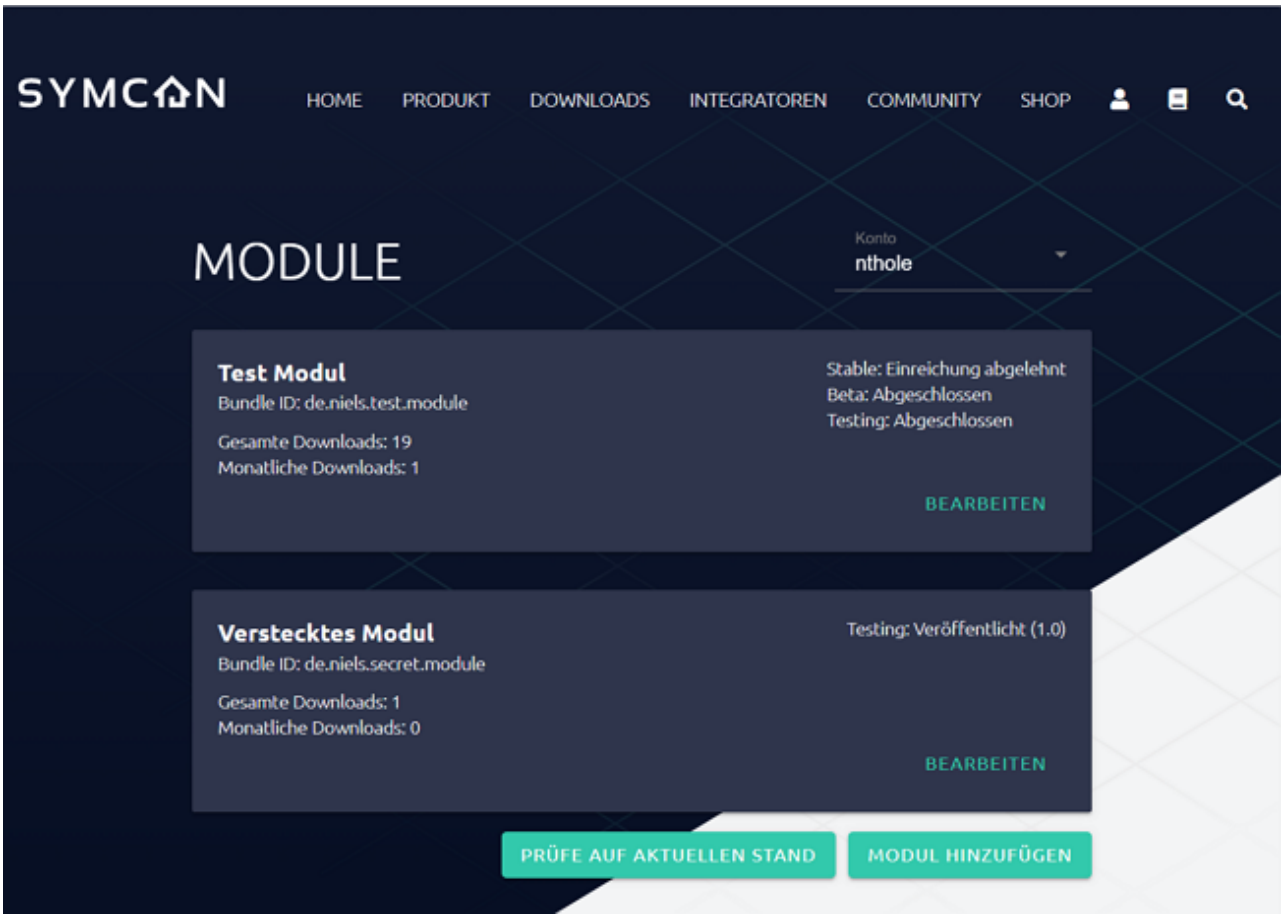
# Einreichen

## BESCHREIBUNG

Soll eine entwickelte Bibliothek im Module Store angeboten werden, so muss dies über den Entwicklerbereich auf der Accountseite eingereicht werden.

## ENTWICKLERBEREICH

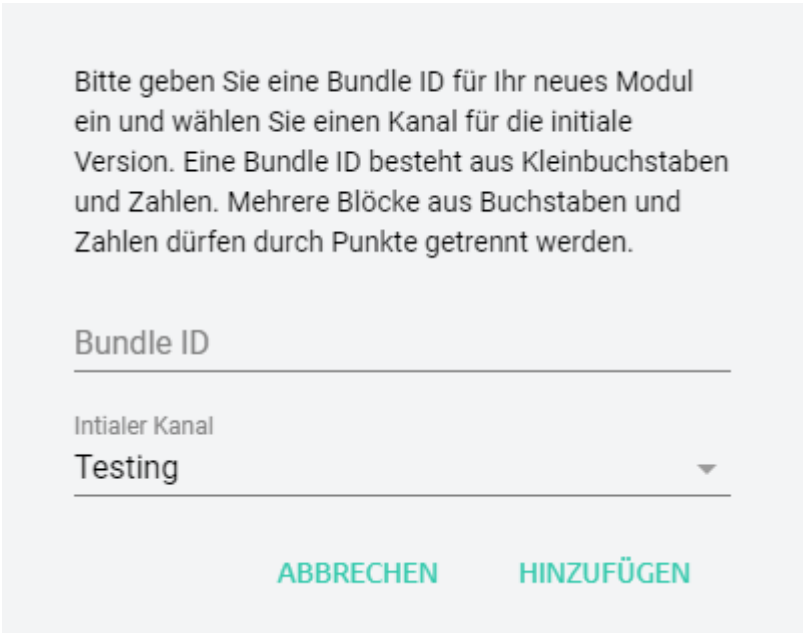
Der Entwicklerbereich ist auf der Accountseite unter "Entwicklerbereich" zu erreichen.




Hier werden alle bisherigen Einreichungen aufgelistet. Es werden die Downloads des Moduls sowie der Zustand der einzelnen Kanäle dargestellt. Über den Button "Bearbeiten" kann ein Modul angepasst werden. Über den Button "Modul hinzufügen" können neue Module für eine Einreichung vorbereitet werden. Mit dem Button "Prüfe auf aktuellen Stand" wird automatisch geprüft welche Module aktuell auf dem aktuellen Stand des Hauptbranches des dazugehörigen Repositories sind. Diese Funktion wird nur für Repositories auf GitHub unterstützt. Bei einer größeren Anzahl von Modulen werden entsprechend viele Anfragen an die GitHub-API gemacht, welche ohne Verknüpfung mit einem GitHub-Benutzerkonto schnell aufgebraucht werden können. Wird das Konto des persönlichen Bereichs mit GitHub verbunden wird dieses Limit erheblich erhöht. Falls dem eigenen Konto Zugriff auf andere Konten gewährt wurde, können durch die Auswahl "Konto" die Einreichungen dieser Konten geprüft und bearbeitet werden.

## MODUL HINZUFÜGEN

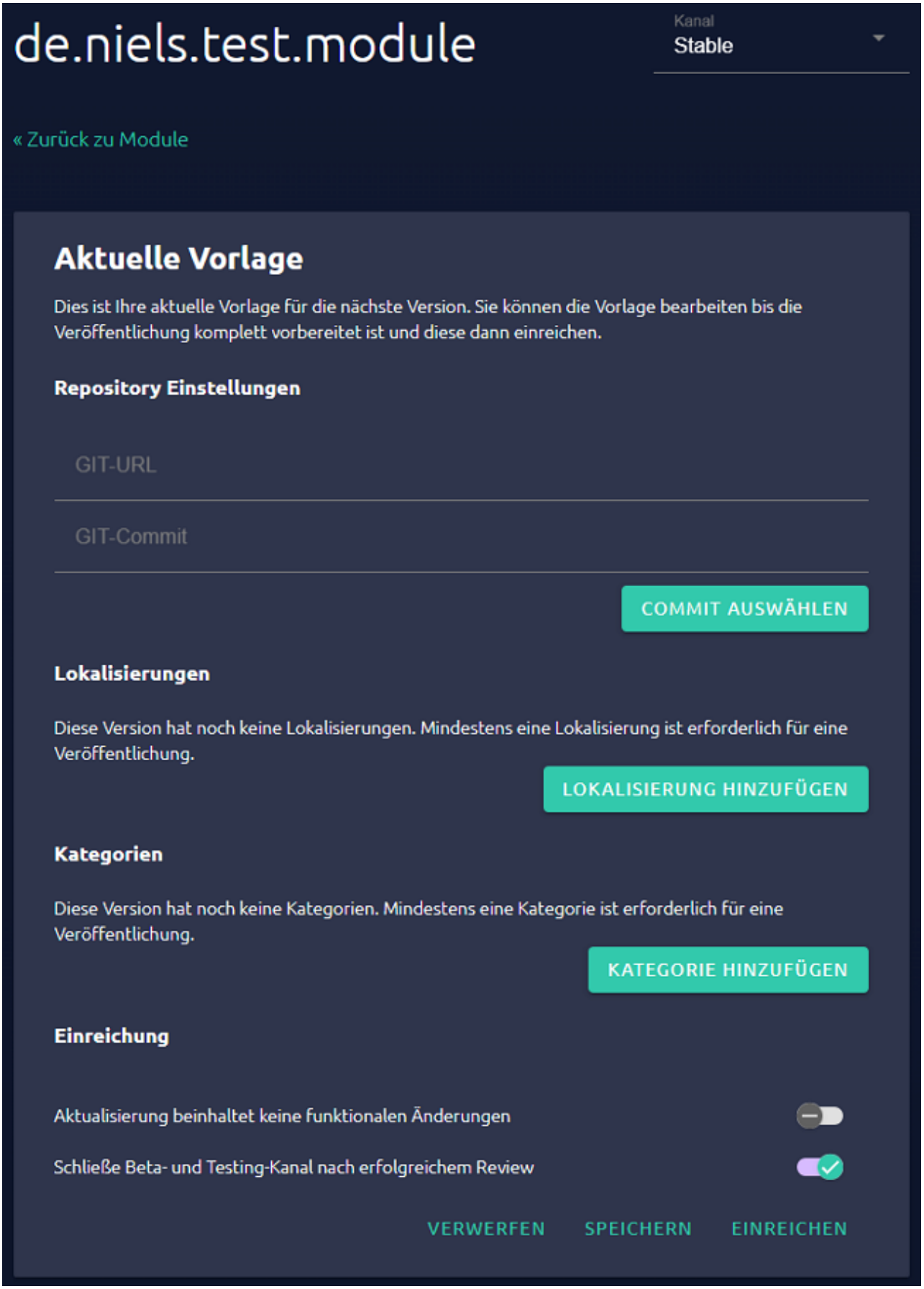
Nach einem Klick auf "Modul hinzufügen" erscheint ein Dialog, in welchem eine Bundle ID und ein initialer Kanal ausgewählt wird.




Die Bundle ID ist eine eindeutige ID, welche auch bei neuen Versionen gleich bleibt, und aus Kleinbuchstaben und Punkten besteht. Es wird empfohlen die umgekehrte Domänennotation zu verwenden, z.B. de.symcon.alexa. Der initiale Kanal ist der Kanal für die erste Einreichung. Später können problemlos weitere Kanäle hinzugefügt werden.

## MODUL EINREICHEN

Unter "Modul bearbeiten" kann man den aktuellen Status nach Kanälen prüfen und die nächste Einreichung vorbereiten.

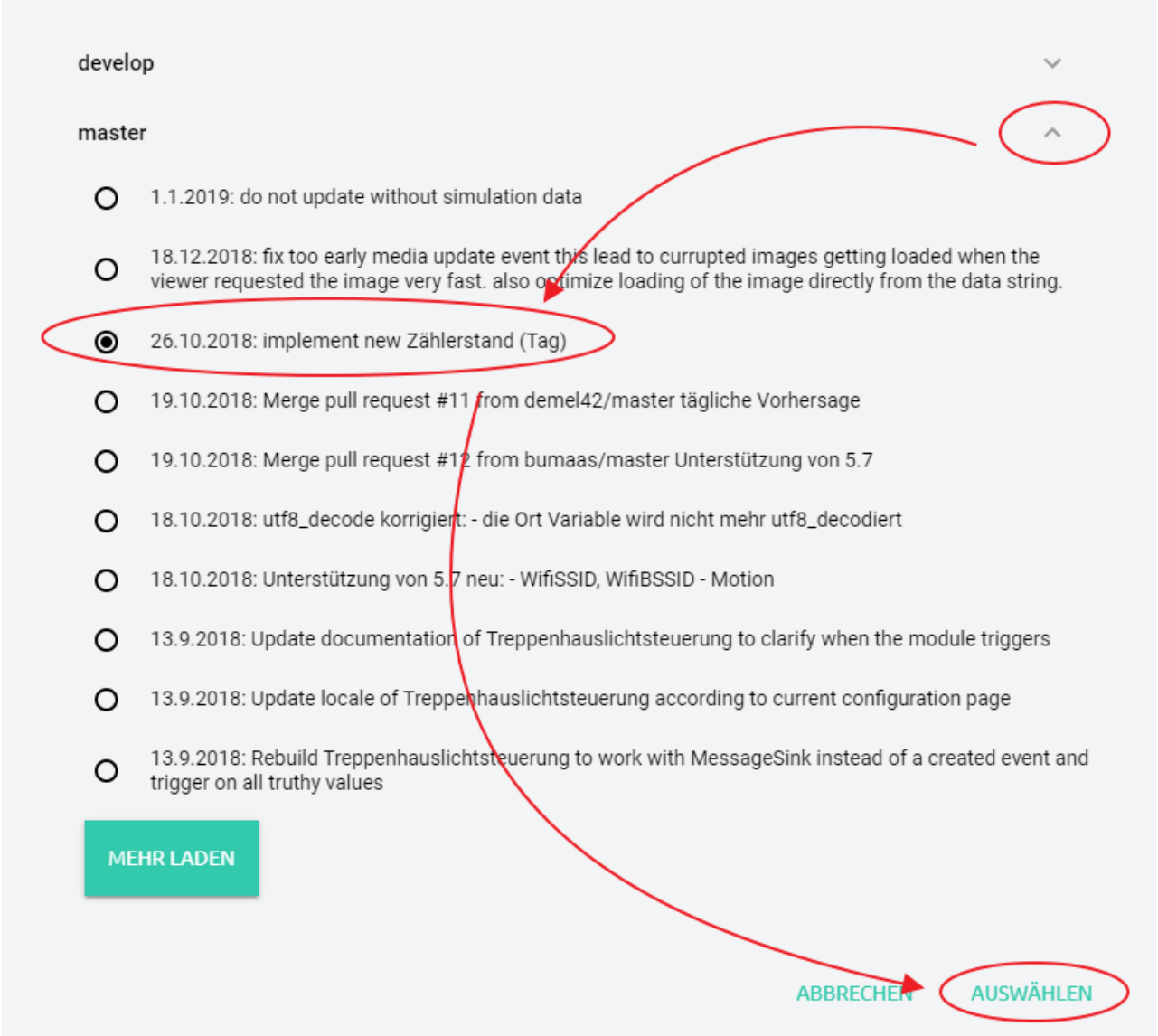



Unter GIT-URL wird die Adresse des Git Repository eingetragen, unter welcher die aktuelle Bibliothek zu finden ist. Um private Repositories zu verwenden, muss das Konto des persönlichen Bereichs mit einem GitHub-Benutzerkonto verbunden werden, welches Zugriff auf das private Repository hat.

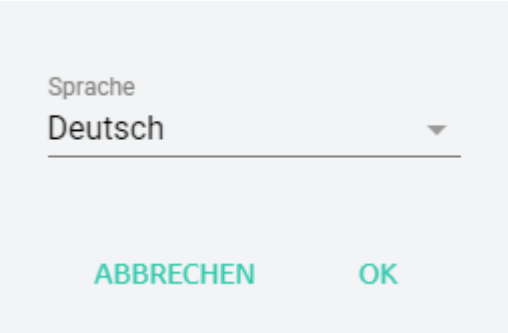


Aktuell werden private Repositories nur bei GitHub unterstützt.

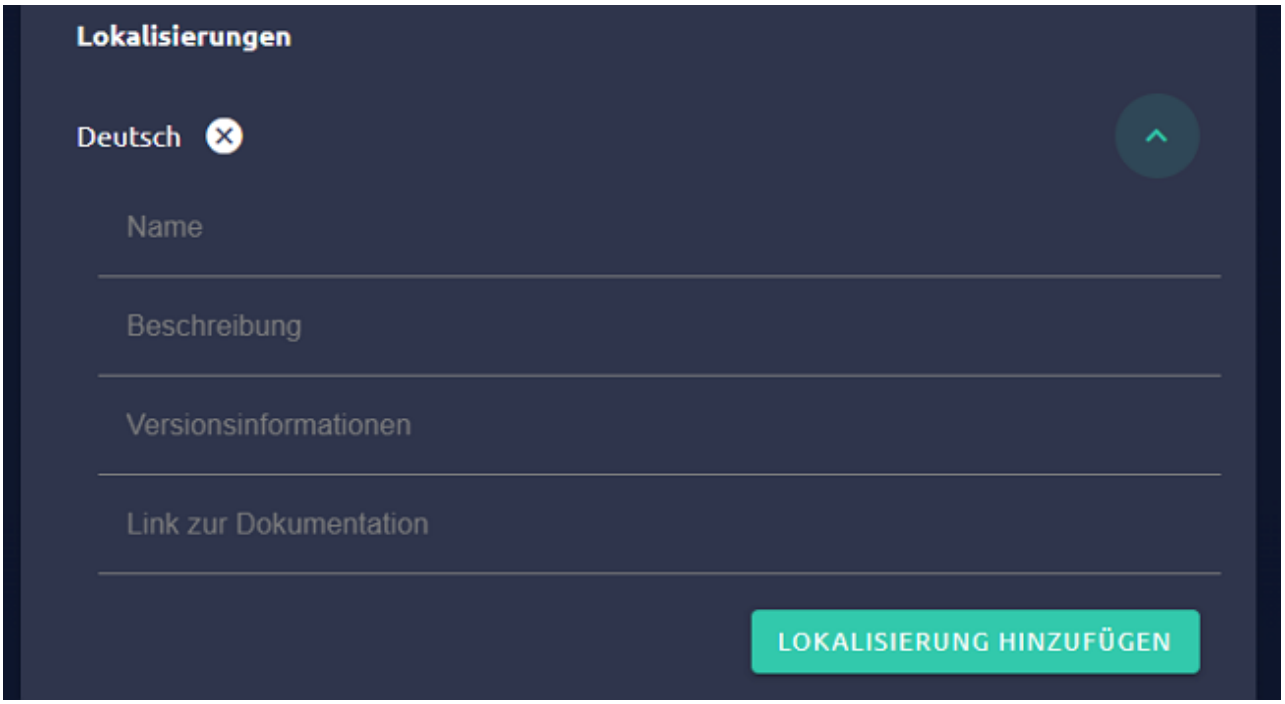
Nach der URL wird der Commit ausgewählt, welcher veröffentlicht werden soll. Hierzu kann man auf das Zahnrad in der Zeile "GIT-Commit" klicken, den gewünschten Commit anklicken und per "Auswählen" bestätigen. Der Dialog bietet die Commits nach Branches aufgeteilt an. Wird der gewünschte Commit nicht angezeigt, können per "Mehr laden" mehr Commits eines Branches geladen werden. Alternativ kann die ID des Commits auch manuell eingegeben werden. Ist das Repository nicht bei GitHub ist der Dialog eingeschränkt und es kann lediglich ein Branch ausgewählt werden. Es wird automatisch der neueste Commit auf diesem Branch ausgewählt.




Eine Lokalisierung stellt Namen und Beschreibungen des Moduls in einer Sprache bereit. Es können mehrere Lokalisierungen für mehrere Sprachen hinzugefügt werden, mindestens eine ist allerdings erforderlich. Das Modul muss in allen lokalisierten Sprachen nutzbar sein. Wurde ein Modul einmal mit einer Lokalisierung veröffentlicht, so muss diese Sprache auch in zukünftigen Veröffentlichungen angeboten werden.
Durch einen Klick auf "Lokalisierung hinzufügen" öffnet sich ein Dialog in welchem die Sprache der neuen Lokalisierung ausgewählt werden kann.



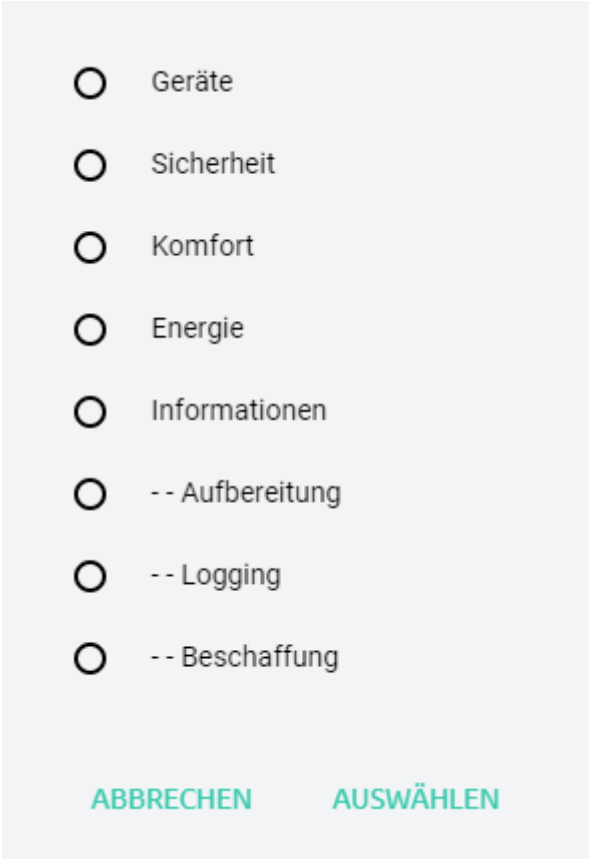

Nachdem die Auswahl mit "OK" bestätigt wurde, erscheint eine neue leere Lokalisierung der ausgewählten Sprache. Mit einem Klick auf den Pfeil rechts kann eine Lokalisierung ausgeklappt werden.




Folgende Felder müssen für eine Lokalisierung ausgefüllt werden:

| FELD | BESCHREIBUNG |
|---|---|
| Name | Der angezeigte Name des Moduls |
| Beschreibung | Eine Beschreibung des Moduls |
| Versionsinformation | Informationen was zu dieser Version neu ist |
| Link zur Dokumentation | Ein Link zur Erklärung des Moduls. Dies kann beispielsweise eine Readme im Repository oder auch ein erklärender Beitrag im Forum sein |

Durch einen Klick auf das X neben der Sprache kann eine Lokalisierung wieder entfernt werden.
Abschließend werden passende Kategorien für die Einreichung ausgewählt. Über "Kategorie hinzufügen" erscheint ein Dialog in welchem eine neue Kategorie ausgewählt werden kann und mit "Auswählen" bestätigt wird.




Kategorien können durch einen Klick auf das dazugehörige X wieder entfernt werden. Für eine Einreichung ist mindestens eine Kategorie erforderlich, es dürfen aber auch mehr Kategorien sein.
Unter "Einreichung" können noch weitere Einstellungen vorgenommen werden.

| EINSTELLUNG | BESCHREIBUNG |
|---|---|
| Aktualisierung beinhaltet keine funktionalen Änderungen | Diese Einstellung kann aktiviert werden, wenn die aktuelle Einreichung keine für Benutzer relevanten programmatischen Änderungen beinhaltet und beispielsweise nur der Stil aktualisiert wurde. Die neue Version wird Benutzern nicht angeboten. Dies kann sinnvoll sein um nicht-funktionale Änderungen einzureichen, damit das Modul wieder auf aktuellem Stand erkannt wird |
| Schließe Beta- und Testing-Kanal nach erfolgreichem Review | Ist diese Einstellung aktiviert, werden nach erfolgreicher Veröffentlichung alle niedrigeren Kanäle geschlossen. Geschlossene Kanäle sind nicht mehr im Module Store verfügbar. Benutzer, die nun auf einem geschlossenen Kanal sind, werden für Aktualisierungen auf den nächsthöheren verfügbaren Kanal verwiesen. Existieren aktuell keine Veröffentlichung auf einem niedrigeren Branch, wird diese Einstellung nicht angezeigt |

Abschließend bietet die aktuelle Vorlage drei Buttons. Durch "Verwerfen" wird die Vorlage wieder gelöscht. Existiert von dem Modul nur die Vorlage, so wird damit das gesamte Bundle gelöscht. "Speichern" speichert den aktuellen Zustand der Vorlage. Hierfür müssen nicht alle Elemente ausgefüllt sein. Ein Modul ohne Lokalisierungen und Kategorien kann also gespeichert werden, aber nicht eingereicht. Per "Einreichen" reicht man das Modul zur Bereitstellung im Module Store ein. Auf den Kanälen Testing und Beta wird das Modul direkt im Module Store angeboten. Bei einer Einreichung auf Stable durchläuft das Modul vor der Veröffentlichung einen Reviewprozess durch das Symcon-Team. Wird das Modul angenommen, wird es im Module Store veröffentlicht. Wird es abgelehnt, wird man im Entwicklerbereich und per E-Mail benachrichtigt, warum es zur Ablehnung kam.

## REVIEW

Bei dem Review eines Moduls wird auf verschiedene Punkte auf Basis unser Best Practices geprüft. Dies sind inbesondere folgende Punkte:

Greift das Modul auf lokale Dateien zu und könnte diese auslesen und manipulieren?

> Es ist in Ausnahmefällen genehmigt auf lokale Dateien zuzugreifen, insbesondere innerhalb des IP-Symcon-Ordners. Hier erfolgt allerdings eine genauere Prüfung um Missbrauch zu verhindern.

Ist für jede Lokalisierung des Modules auch eine entsprechende Übersetzung vorhanden?

Lassen sich die Instanzen des Moduls fehlerfrei erstellen, auch wenn möglicherweise nicht alle Anforderungen erfüllt sind?

> Ein Modul muss unter allen Umständen als Instanz erstellbar sein. Das Modul muss also selbstständig prüfen ob die Anforderungen erfüllt sind. Sind die Anforderungen nicht erfüllt, so soll dies geeignet in der Instanzkonfiguration dargestellt werden.

Läuft eine eventuelle Kommunikation über Splitter/IO wie vorgesehen über den Datenfluss via SendDataToChildren/SendDataToParent/ReceiveData/ForwardData?

Greift ein Modul nur auf eigene oder ihm zugewiesene Objekte zu?

> Eigene Objekte liegen unterhalb der Instanz.
> Per SelectVariable und ähnlichen Elementen können einem Modul weitere Objekte zugewiesen werden, welche im erwarteten Rahmen verwendet werden dürfen.
> Externe Objekte, welche dem Modul nicht zugewiesen wurden, dürfen nicht manipuliert werden.

Neue Objekte dürfen außerhalb der Instanz nur erstellt werden, wenn dies explizit vom Benutzer bestätigt wurde. Ein konformes Beispiel hierfür ist die Verwendung des Konfigurationselements "Configurator".

> Dies verbietet auch die Erstellung von Ereignissen um interne Funktionen zu schalten. Hierfür sind RegisterTimer bzw. MessageSink und die dazugehörigen Funktionen zu verwenden.

Abweichungen sind mit Absprache möglich. Dies sollte bei Einreichung allerdings an geeigneter Stelle erörtert werden, beispielsweise per E-Mail oder als Kommentar an der betroffenen Stelle im Code.

Codequalität oder Fehlerfreiheit wird im Rahmen des Reviews nicht überprüft.

"IPSymcon", "IPS" oder vergleichbares darf nicht Teil eines in einer Lokalisierung gewählten Namens sein.

Module, die als Bedingung mehr als die aktuelle Stable-Version erfordern, also beispielsweise nur mit der aktuellen Beta funktionieren, dürfen nicht auf dem Stable-Kanal eingereicht werden.

Einige Objekteigenschaften befinden sich auch bei moduleigenen Objekten in der Hoheit des Benutzers. Diese dürfen initial vorgegeben werden, aber ohne explizite Bestätigung des Benutzers nicht weiter manipuliert werden. Diese Eigenschaften sind:

> Der Name
> Alle visuellen Einstellungen: Objekt anzeigen, Objekt aktiv und Icon
> Die Beschreibung
> Die Position
> Alle ereignisspezifischen Eigenschaften
> Benutzerdefiniertes Profil, benutzerdefinierte Aktion und Logging von Variablen dürfen nie ohne explizite Bestätigung gesetzt werden, auch nicht initial! Aktion oder Profil können selbstverständlich über Standardaktion bzw. -profil definiert werden.

PHP-Dateien dürfen nicht den kurzen PHP-Tag (<?) verwenden und sollten stattdessen den langen PHP-Tag (<?php) verwenden.

Nach der initialen Erstellung einer Instanz liegt die Hoheit über die Eigenschaften beim

Benutzer.

Wenn das Modul hier etwas ändern möchte, so können andere Werte über die Dynamikfunktion UpdateFormField vorgegeben werden und dann vom Benutzer bestätigt werden.

Somit sollten die Funktionen IPS_SetProperty und IPS_ApplyChanges nie verwendet werden.

Wird bei einer List oder einem Tree der Parameter loadValuesFromConfiguration auf false gesetzt, so muss die Eigenschaft dennoch beim Laden im Sinne des Benutzers gefüllt werden. Der Parameter soll es lediglich ermöglichen beispielsweise die Reihenfolge anzupassen oder neue Werte mittig hinzuzufügen.

Eine Einreichung benötigt einen sprechenden Namen, eine passende Beschreibung und Versionsinformation sowie eine Dokumentation, welche die Konfiguration und Verwendung des Moduls erklärt.

Sind Konfiguratoren mit einem Splitter oder I/O verbunden, so dürfen sie nur Geräte verwalten, welche auch zu diesem Splitter bzw. I/O verbunden sind

Instanzen, welche mit dem gleichen Splitter oder I/O wie der Konfigurator verbunden sind, sollen im Konfigurator aufgelistet werden, egal ob diese physikalisch noch existieren oder nicht

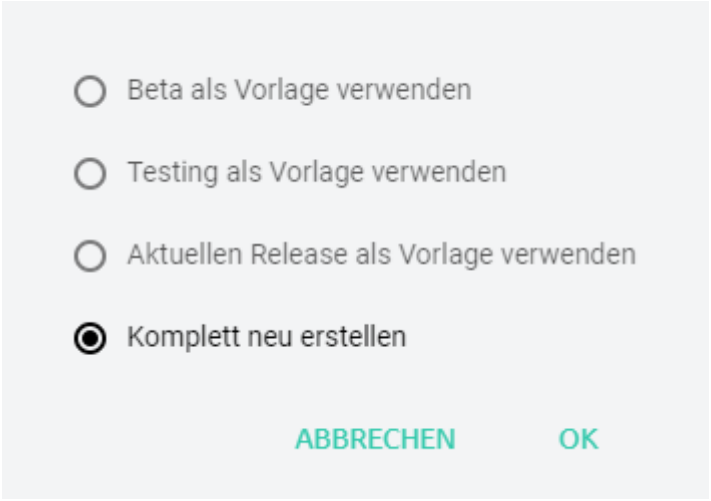
Instanzen, welche mit einem anderen Splitter oder I/O verbunden sind, dürfen nicht in einem Konfigurator mit Splitter oder I/O aufgelistet werden

Bietet ein Modul eine Discovery-Instanz an, so muss diese ohne Konfiguration funktionieren

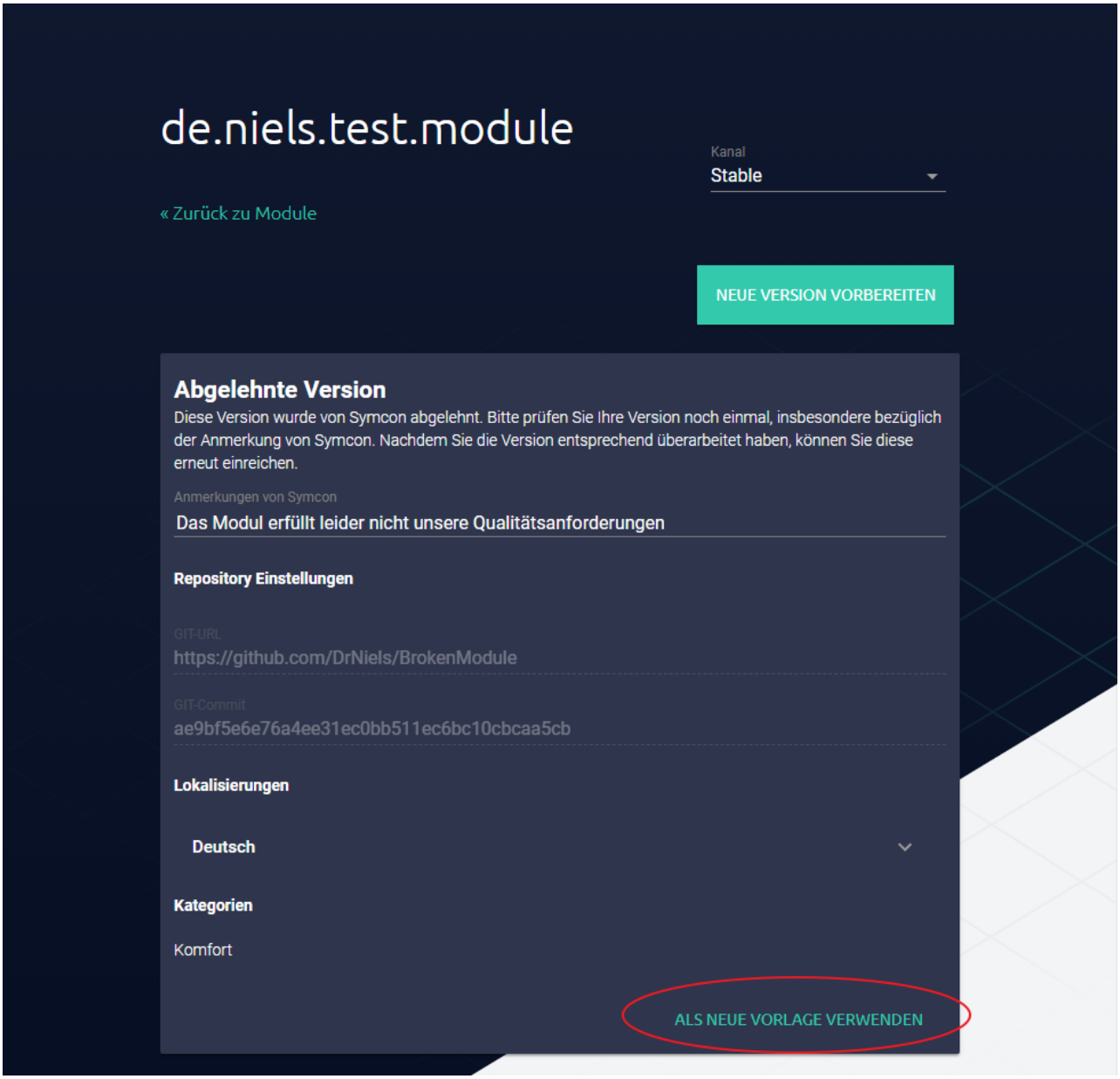
Da Logging problemlos via $this->LogMessage realisiert werden kann und diese Nachrichten korrekt mit der sendenden Instanz verknüpft sind, darf IPS_LogMessage nicht mehr zum Logging verwendet werden

## MODUL BEARBEITEN

Existiert unter "Modul bearbeiten" auf einem Kanal keine aktuelle Vorlage, so kann durch einen Klick auf "Neue Version vorbereiten" eine neue Vorlage erstellt werden. Im Dialog kann ausgewählt werden, ob man eine komplett neue Vorlage erstellt oder eine aktuelle Veröffentlichung als Vorlage verwenden möchte.



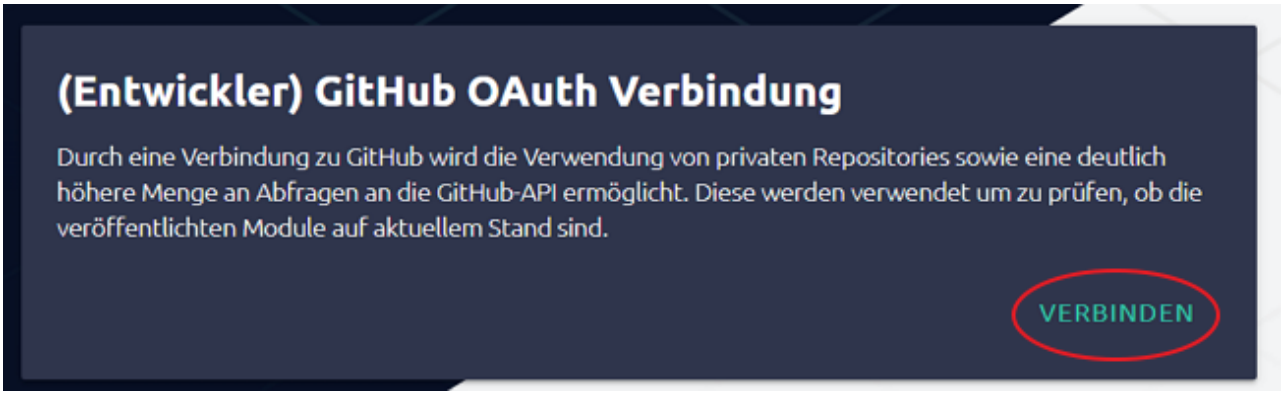

Alternativ ist bei Veröffentlichungen eine Option diese auf einen höheren Kanal zu ziehen vorhanden. Wurde eine Einreichung abgelehnt, so kann diese auch als neue Vorlage verwendet werden um erkannte Mängel auszugleichen ohne alles erneut zu konfigurieren.



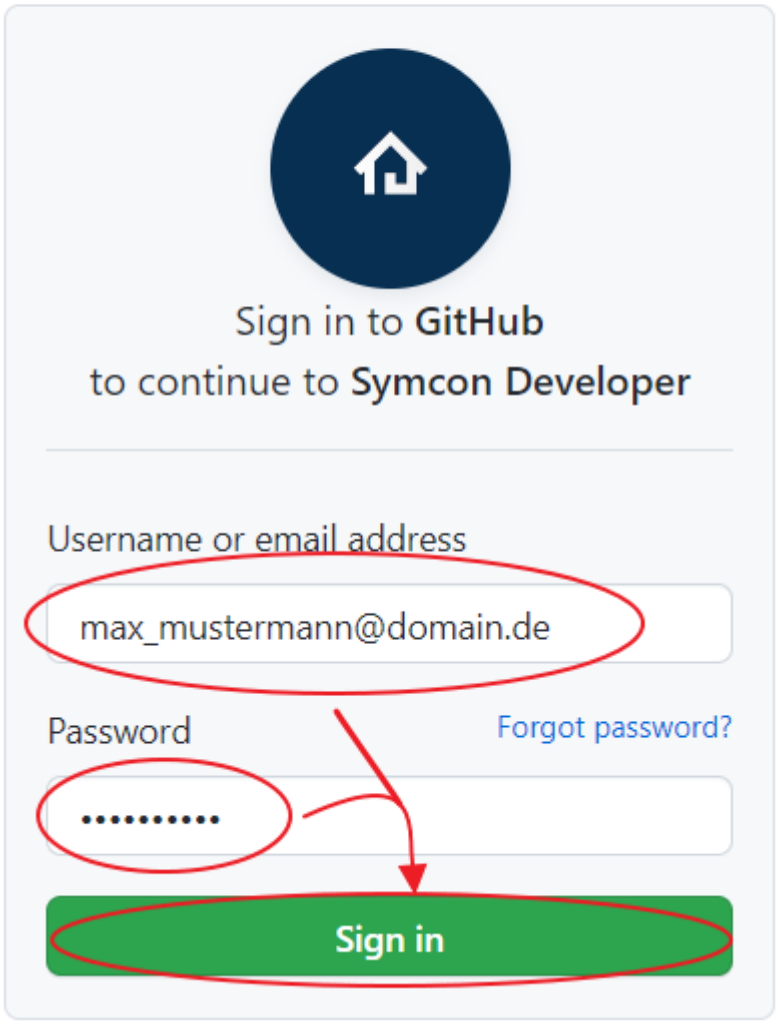

Soll ein Modul, das sich aktuell im Review befindet, zurückgezogen werden, so kann dies durch einen Klick auf "Zurückziehen" getan werden. Befindet sich ein Modul im Module Store, so kann es nicht mehr zurückgezogen werden.

## MIT GITHUB VERBINDEN

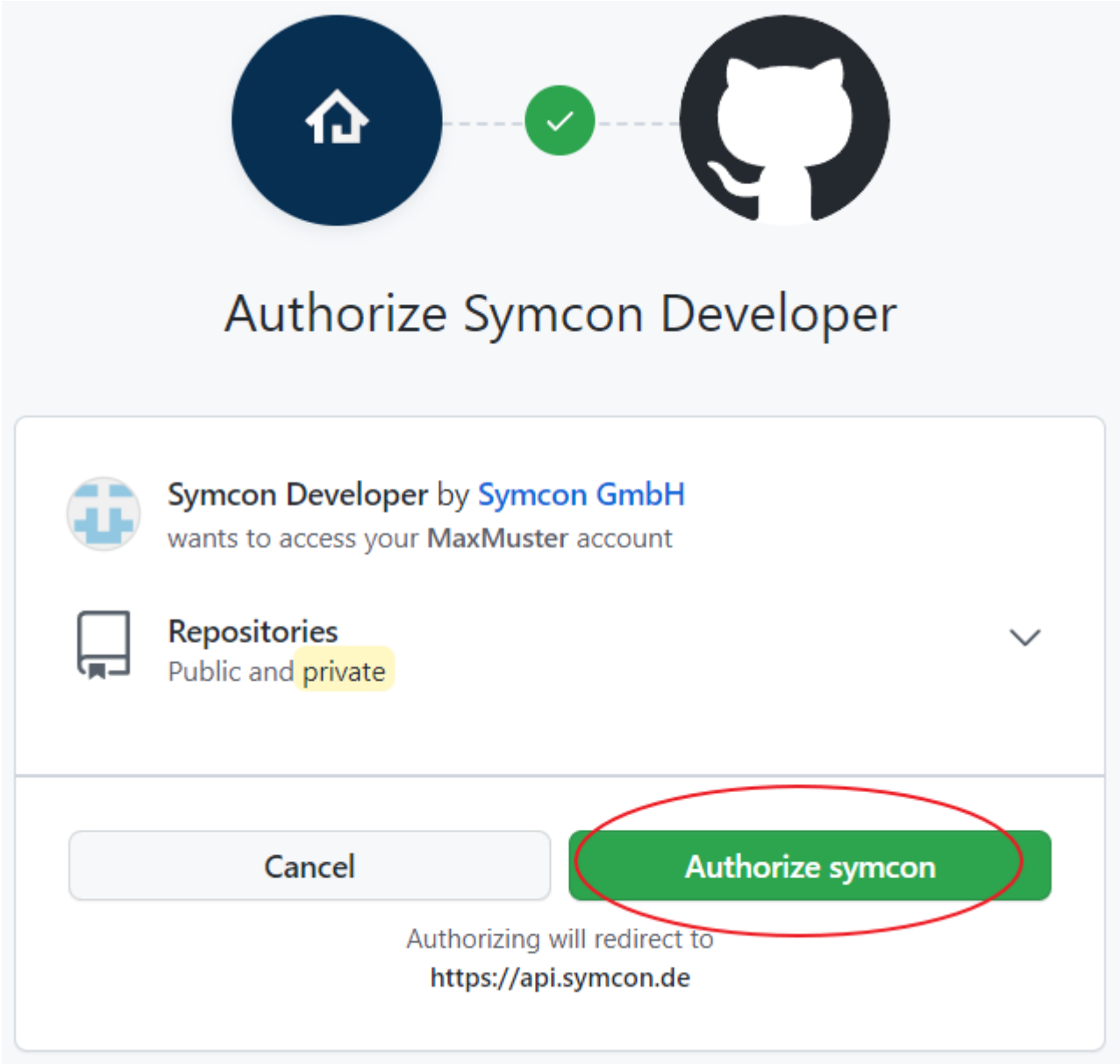
Das Konto des persönlichen Bereichs kann mit GitHub verknüpft werden. Dies ermöglicht zum einen die Verwendung von privaten Repositories sowie eine deutlich höhere Menge an Abfragen an die GitHub-API. Diese können insbesondere relevant werden, wenn die veröffentlichten Module auf aktuellen Stand geprüft werden sollen. Die Verknüpfung kann in den Einstellungen des persönlichen Bereichs durchgeführt werden. Dazu muss im Bereich "(Entwickler) GitHub OAuth Verbindung" auf "Verbinden" geklickt werden.




Sofern im Browser aktuell kein GitHub-Benutzer angemeldet ist, geschieht dies im nächsten Schritt. Hierzu müssen E-Mail-Adresse und Passwort des GitHub-Benutzerkontos eingegeben werden. Ist bereits ein Benutzer angemeldet, wird dieser Schritt automatisch übersprungen.




Abschließend muss der Zugriff durch Symcon autorisiert werden. Dies geschieht durch einen Klick auf "Authorize symcon".




Nun ist das Konto des persönlichen Bereichs mit dem GitHub-Benutzerkonto verknüpft. Um die Verbindung im Nachhinein wieder zu lösen, kann dies via Klick auf "Trennen" getan werden.

# Struktur

## BESCHREIBUNG

Die vorausgesetzte Struktur ist für eine richtige Funktionsweise innerhalb von IP-Symcon nötig. Nur wenn die Struktur eingehalten wird, funktioniert das Einbinden über Module Control.
  Die "library.json" ist das Kernstück einer jeden Modulentwicklung.
  Die "module.php" und "module.json" bilden zusammen das eigentliche Modul.
  Für die Einrichtung der Konfigurationsseite kann die form.json genutzt werden.
  Für mögliche Übersetzungen dient die locale.json.
  Der Modulordnername (hier: Modul1, Modul2) sollte den gleichen Namen haben, wie der Klassenname in der module.php.
Ordner, welche keine module.json beinhalten, werden als fehlerhaft markiert.
Ausnahme bilden die Ordner:
  libs/ (ab Version 4.2)
  docs/ (ab Version 4.2)
  imgs/ (ab Version 4.2)
  tests/ (ab Version 4.4)
  actions/ (ab Version 6.0)



Punkt-Ordner (z.B. .github, .style) werden ebenfalls ignoriert und sind nicht für die korrekte Funktionsweise des Modul nötig. Diese sind zum Beispiel wichtig, für die korrekte Funktionsweise des Repositories.

Diese Ordner werden nicht als Modul eingebunden und bieten die Möglichkeit externe Libraries, Dokumente, Bilder und Tests zur Verfügung zu stellen.

## VERZEICHNISSTRUKTUR

```
Bibliothek
|
- Modul1
| |
| - module.php
| |
| - module.json
| |
| - form.json (optional)
| |
| - locale.json (optional)
|
- Modul2
| |
| - module.php
| |
| - module.json
```

```
| |
| - form.json (optional)
| |
| - locale.json (optional)
|
- actions (optional)
| |
| - Definitionen von Aktionen
|
- libs (optional)
| |
| - beliebige Libraries
|
- docs (optional)
| |
| - beliebige Dokumente
|
- imgs (optional)
| |
| - beliebgie Mediendateien
|
- tests (optional)
| |
| - beliebige Testdateien
|
- library.json
|
- README.txt (optional)
```

# SDK (Skins)

Benötigt: IP-Symcon >= 3.0

## BESCHREIBUNG

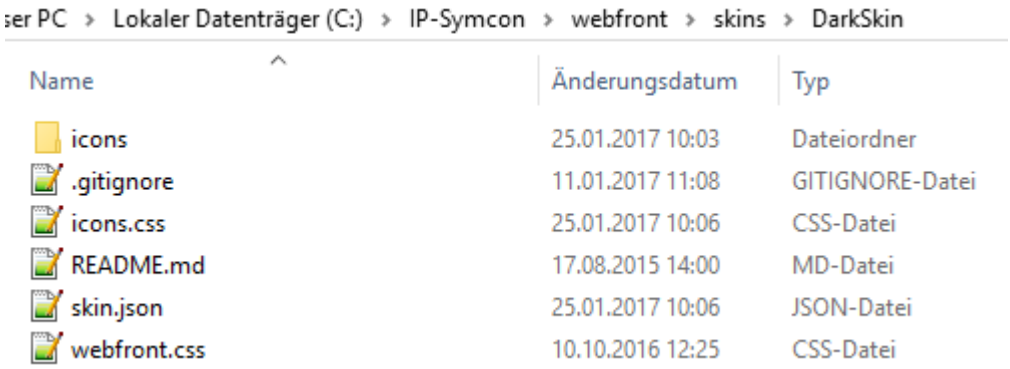
Ab der Version 3.0 können eigene Skins erstellt werden.
Das Erstellen und Austauschen von Icons ist schnell verwirklicht, sollen allerdings ganze Änderungen am Style verwirklicht werden, sind zusätzlich CSS Kenntnisse nötig.

> Dies funktioniert nur für das WebFront. Sollte ein Skin oder Anpassung der mobilen Apps gewünscht sein, kann dies über das Enterprise Programm verwirklicht werden.
> Weitere Informationen können unter Produkte für Unternehmen gefunden und über Direktkontakt ein individuelles Branding angefragt werden

## VORAUSSETZUNGEN

IP-Symcon 3.0 oder höher
CSS Kenntnisse (sofern mehr als die Icons verändert werden sollen)
Beispiel Skin herunterladen um Grundstruktur zu haben (empfohlen)

## ORDNERSTRUKTUR




```
Skinname
|
- icons (optional)
| |
| - Aircraft.png
| |
| - Bird.png
|
- img (optional)
| |
| - Background.png
```

```
|
- font (optional)
| |
| - WebFont.ttf
|
- skin.json (wird generiert)
|
- webfront.css (benötigt, wenn auch leer)
|
- icons.css (wird generiert)
|
- README.md (optional)
```

## ICONS-ORDNER

Icons beinhaltet alle Icons. Sollten gleichnamige Standard-Icons vorhanden sein, werden die Custom-Icons bevorzugt.
Enthält alle Icons im PNG Format und Größe 32x32.
Namen der Icons dürfen nur aus Buchstabe, Zahlen, Minus (-) und Unterstrich (_) bestehen. Leerzeichen, Umlaute und Sonderzeichen sind nicht erlaubt.
Die Icons müssen alle mit Großbuchstaben beginnen (z.B. Arrow.png, Right.png)
Bei Wortverkettungen ist es empfohlen, dass diese auch mit einem großen Buchstaben geschrieben werden (z.B. ArrowRight.png). Dies dient der besseren Lesbarkeit und der Konsistenz zu den IP-Symcon Icons.
Wenn ein Icon den selben Namen hat wie eines der IP-Symcon Icons, dann wird im WebFront das Custom-Icon bevorzugt. Die Groß-/Kleinschreibung ist zu beachten. (Diese Funktion ist eine elegante Art ein fremdes WebFront ohne viel Aufwand mit einem Skin auszurüsten.)

## IMAGE-ORDNER

Img beinhaltet zusätzliche Grafiken für z.B. mögliche Hintergründe.
Enthält weitere Bilder für z.B. spezielle Hintergründe

## FONT-ORDNER

Beinhaltet benötigte Fonts für den Skin.
Enthält weitere WebFonts
Die CSS-Elemente der WebFont sollen in der webfront.css hinzugefügt werden

## EINZELDATEIEN

Dateien welche im Hauptordner liegen.

### SKIN.JSON

Meta-Datei, die automatisch vom Skin-Builder erstellt wird. Die Felder Author, Version, URL, GIT und Compatible dürfen nach dem Erstellen verändert werden. URL sollte dabei die

Adresse zum Skin-Thema im Forum sein. GIT sollte die Adresse zum GitHub-Repository sein, damit das Skin-Control entsprechend Updates herunterladen kann. Das Icons Array wird automatisch erstellt und dient dem WebFront Editor als Hilfsmittel, um die Custom-Icons auswählbar zu machen. Compatible muss den Wert 1 besitzen. Dieser wird strikt überprüft, um bei Änderungen am WebFront, die den Skins inkompatibel machen, diesen deaktivieren zu können.

### WEBFRONT.CSS

CSS-Datei, die Änderungen am WebFront CSS-Style enthält. Muss vorhanden sein. Kann auch leer sein.
CSS-Änderungen müssen den o.g. Regeln entsprechen.

### ICONS.CSS

CSS-Datei, die die Links auf die benutzerdefinierten Icons enthält.
Wird automatisch vom Skin-Builder generiert und sollte nicht verändert werden.

### README.MD

Readme welche von GitHub automatisch im Browser in der Repository Übersicht angezeigt wird. Kann zur Dokumentation genutzt werden.

## SKIN BUILDER

Der Skin-Manager wird mit http//host:port/skins/ aufgerufen. (Beispiel: http://localhost:3777/skins/)
Sobald ein Skin ausgewählt wurde, werden die aktuell in der CSS-/Meta-Datei vorhandenen Icons angezeigt.
Wenn neue Icons hinzugefügt/entfernt wurden, muss ein "Build Skin!" ausgeführt werden, damit die icons.css/skin.json entsprechend neu generiert werden. Bei Änderungen an Icons selbst oder der webfront.css ist keine Aktion erforderlich. Evtl. muss der Browsercache gelöscht werden, damit Änderungen sichtbar werden.



Bis Version 4.4 war der Skin-Manager unter http//host:port/user/skins/ zu finden. (Beispiel: http://localhost:3777/user/skins/)

## SKINS BEREITSTELLEN

Skins können bei GitHub hochgeladen werden.
Der GitHub Repository-Name sollte dem Namen des Skins entsprechen.
Im Repository sollten sich nur die notwendigen Dateien befinden wie icons.css, skin.json, webfront.css und optionale Ordner wie icons, font oder img. Eine README.md und eine LICENSE Datei werden zusätzlich empfohlen. Die Dateien dürfen nicht in einem zusätzlicher Unterordner verschachtelt sein, da ansonsten der Installer nicht korrekt arbeitet.

Bei der Veröffentlichung muss die Größe (Anzahl in kB/mB), die der Skin zusätzlich beim Laden braucht, angegeben werden. (Diese wird nach dem Build-Prozess angezeigt)

## SKINS INSTALLIEREN

Dies funktioniert via Skin-Control und ist dort beschrieben.

## SKINS AUSWÄHLEN

Im WebFront Konfigurator gibt es ein Feld "Skin" unter dem Reiter Darstellung, in dem der Skin ausgewählt werden kann.

# Tools

## NETWORK-CONFIGURATION-TOOL

Mit dem Network-Configuration-Tool können alle IPS-Gateways im Netzwerk gefunden werden.
Eine Anleitung zur Nutzung kann HIER gefunden werden.
Zum Download

## RECOVERYTOOL

Das Recoverytool kann ein Image auf die SymBox schreiben.
Eine Anleitung zur Nutzung kann HIER unter dem Punkt "5.2. Wiederherstellung mithilfe des Recovery-Tools" gefunden werden.
Zum Download

## MODUL ENTWICKLUNG

Folgende Tools sind hilfreich bei der Entwicklung von Modulen für den Module Store und helfen um diese erfolgreich einzureichen.

## GUID GENERATOR

Der GUID Generator generiert eine GUID im richtigen Format für das Modul.

## MODUL GENERATOR

Der Modul Generator ist ein Wizard welcher bei dem anlegen der Grundstruktur für ein Modul hilft und alle grundlegenden Einrichtungen für ein funktionierenden Datenfluss einrichtet.

## MODUL VALIDATOR

Der Modul Validator prüft die entwickelten Dateien auf eine korrekte Syntax und ob alle benötigten Informationen enthalten sind.

## EXTERNE IDES

Externe IDEs können Funktionsdeklarationen einfügen und anzeigen.

### PHPSTORM

Eine Anleitung zur Generierung und Nutzung in PhpStorm kann HIER gefunden werden.

## VISUAL STUDIO CODE

Eine Anleitung zur Generierung und Nutzung in Visual Studio Code kann HIER gefunden werden.

# GUID Generator

{8E6496D7-F20B-CD08-B3D5-88A9BCC29D5E}

WEITERE GUID GENERIEREN

IN ZWISCHENABLAGE KOPIEREN

# Modul Generator

## BIBLIOTHEK

GUID

Autor

Name

URL

1.0

0

0

MODUL HINZUFÜGEN

MODULVORLAGE GENERIEREN

# Modul Validator

DATEI AUSWÄHLEN (OPTIONAL)

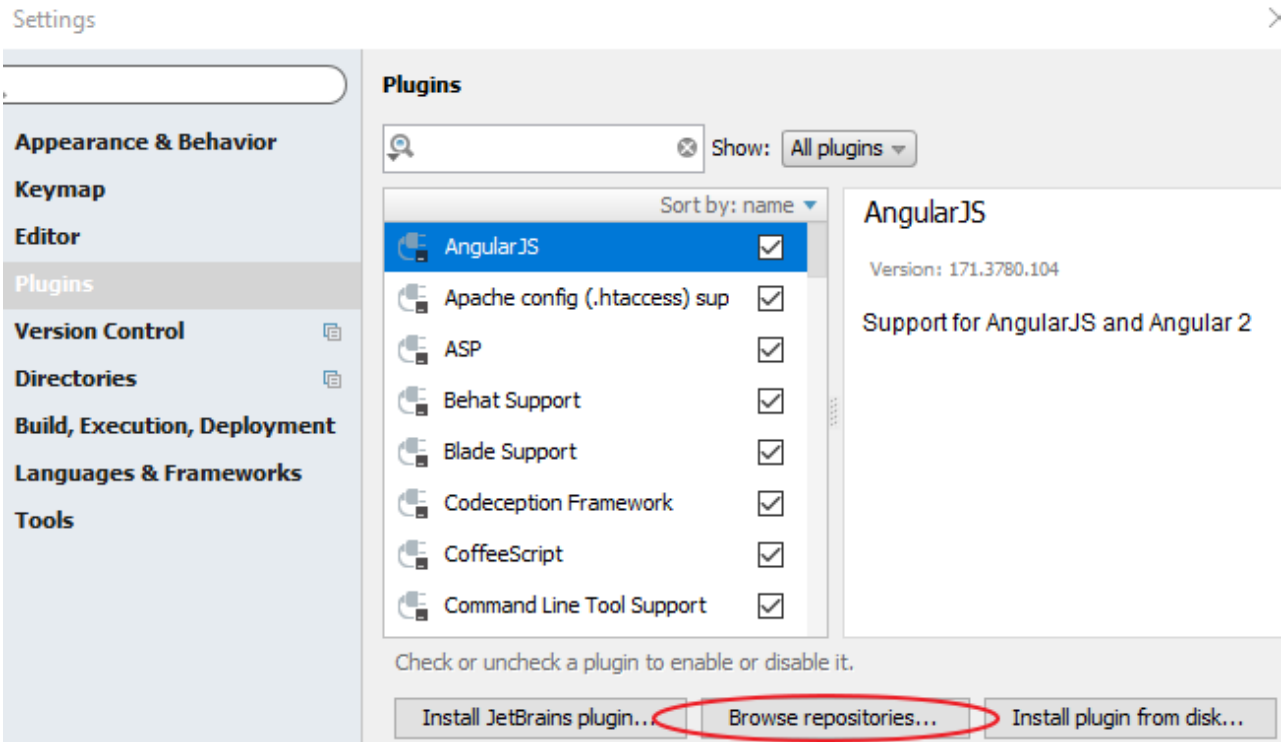
VALIDIEREN

# PhpStorm

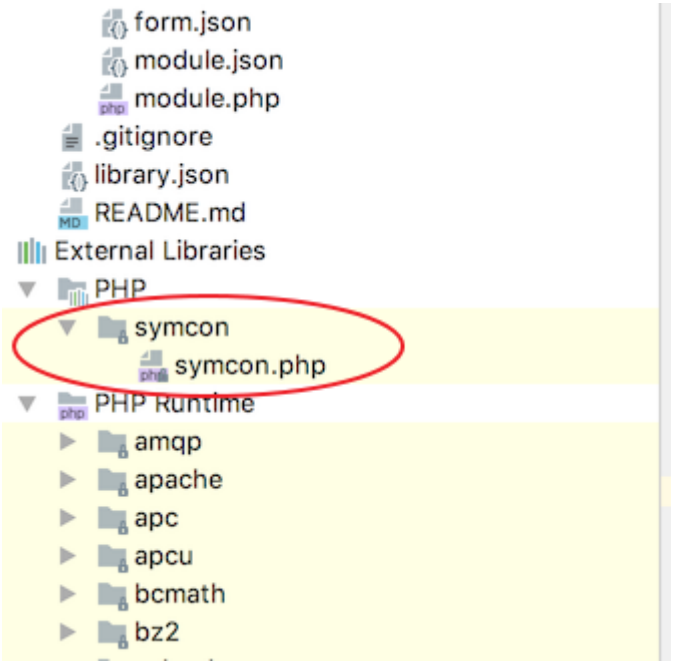
Für PhpStorm ist eine Funktionsliste via Plugin erstellt worden, um diese im Quellcode anzuzeigen.

## PLUGIN INSTALLIEREN

"File -> Settings" aufrufen
"Plugin -> Browse Repositories..."
Nach "Library for IP-Symcon" suchen und das Plugin installieren
PhpStorm neustarten




Bei korrekter Installation befindet sich im linken Projektbaum unter "External Libraries -> PHP" die Library "symcon".


form.json
module.json
module.php
.gitignore
library.json
README.md
External Libraries
PHP
symcon
symcon.php
PHP Runtime
amqp
apache
apc
apcu
bcmath
bz2

# Visual Studio Code

Für VisualStudioCode gibt es einige nützliche Extensions für die Modulentwicklung.

## EXTENSION FÜR IP-SYMCON INSTALLIEREN

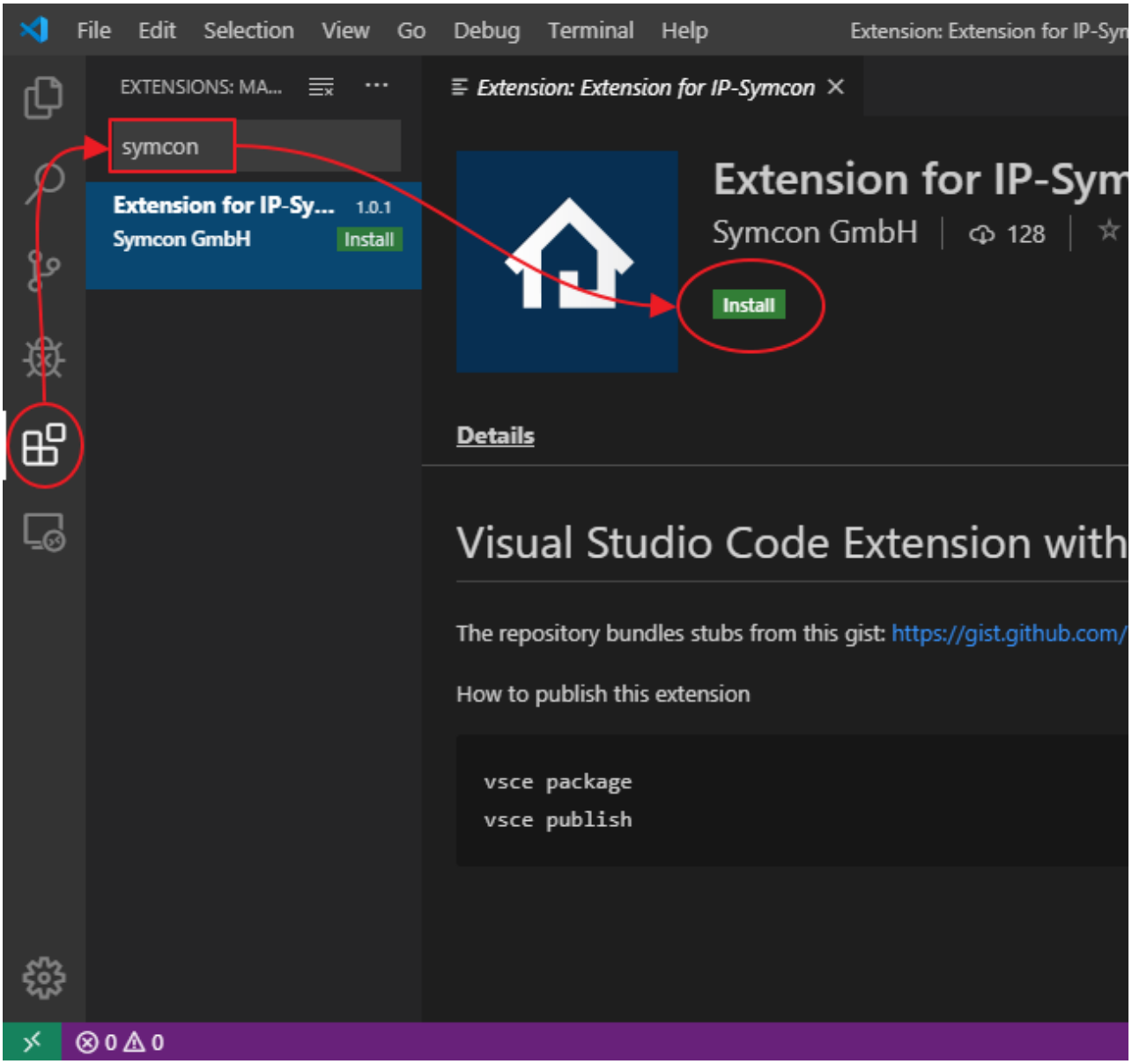
In dieser Extension ist eine Funktionsliste erstellt worden, um diese im Quellcode anzuzeigen.

### MARKETPLACE

Diesen Link zum Marketplace aufrufen und auf "Install" klicken.

### EXTENSIONBROWSER IN VSC

"Extensions" aufrufen
Nach "Symcon" suchen
Die "Extension for IP-Symcon" auswählen und durch klick auf "Install" installieren
Im Extensionsbereich sollte nun unter "Installed" -> "Extension for IP-Symcon" erscheinen




## PHP HERUNTERLADEN

Für die Extensions "CS Fixer" und "PHPUnit" wird PHP gebraucht.
Hier ist ein Link zum Download von PHP. PHP 8 oder höher wird empfohlen.
Mit einem Klick auf das Zahnrad (Manage) "Settings" aufrufen.
Hier unter dem Punkt "Extensions" den Punkt "PHP" auswählen.
Dort gibt es den Punkt "PHP > Validate: Executable Path" bei dem die "settings.json" aufgerufen werden kann.
Hinter "php.validate.executablePath" wird nun der Pfad zur php.exe angegeben.
Die "\" müssen mit einem zweiten "\" maskiert werden.

```
"php.validate.executablePath": "Path\\To\\php.exe"
```

## CS FIXER INSTALLIEREN

Benötigt werden:

Die PHAR-Datei, welche unter diesem Link heruntergeladen werden kann. Die Version 3 wird mit PHP 8 Unterstützt.
Ein .style-Ordner. Empfohlen wird dieses Repository als Submodul im eigenen Repository einzubinden
Die Extension um cs-fixer in Visual Studio Code zu nutzen, welche im Marketplace heruntergeladen oder nachfolgend direkt in Visual Studio Code eingebunden werden kann

### EXTENSIONBROWSER IN VSC

"Extensions" aufrufen
Nach "php cs fixer" suchen
Den "php cs fixer" von junstyle auswählen und durch klick auf "Install" installieren
Im Extensionsbereich sollte nun unter "Installed" -> "php cs fixer" erscheinen

### CS FIXER KONFIGURIEREN

Mit einem Klick auf das Zahnrad (Manage) "Settings" aufrufen
Unter dem Punkt "Extensions" den Punkt "PHP CS Fixer" auswählen
Den Punkt "Allow Risky" setzt man auf true
Unter dem Punkt "PHP-cs-fixer: Executable Path" wird der Pfad zur "php-cs-fixer.phar" angegeben.
Alternativ können diese Einstellungen in die "settings.json" eingeben werden.

```
"php-cs-fixer.executablePath": "Path\\To\\php-cs-fixer-v3.phar",
"php-cs-fixer.allowRisky": true
```

### CS FIXER NUTZEN

Wenn alles korrekt eingebunden und eingerichtet ist, kann man die Extention nutzen. Entweder in dem man in der zu formatierenden Datei rechtsklickt und Format Document auswählt oder den Shortcut "Shift+Alt+F" drückt.
Soll das ganze Projekt den Stil übernehmen kann über die Comando-Zeile der folgende Befehl eingegeben werden.

```
php "Path\\To\\php-cs-fixer-v3.phar" fix --config=.style/.php_cs -v
--allow-risky=yes
```

## PHP UNIT INSTALLIEREN

Benötigt werden:
Die PHAR-Datei von PHPUnit, welche unter diesem Link heruntergeladen werden kann
Ein Test, welcher ausgeführt werden soll
Die Extension um PHPUnit in Visual Studio zu nutzen, welche im Marketplace heruntergeladen oder nachfolgend direkt in Visual Studio Code eingebunden werden kann

## EXTENSIONBROWSER IN VSC

"Extensions" aufrufen
Nach "PHPUnit" suchen
Den "PHPUnit Test Explorer" auswählen und durch klick auf "Install" installieren
Im Extensionsbereich sollte nun unter "Installed" -> "PHPUnit Test Explorer" und "Test Explorer UI" erscheinen

## PHP UNIT KONFIGURIEREN

Mit einem Klick auf das Zahnrad (Manage) "Settings" aufrufen
Unter dem Punkt "Extensions" den Punkt "PHPUnit configuration" auswählen
Bei dem Punkt "Phpunit: Files" wird "{test,tests}/*/Test.php" eingegeben
Unter dem Punkt "Phpunit: PHP" wird der Pfad zur PHP.exe eingegeben.
Bei dem Punkt "Phpunit: Phpunit" wird der Pfad zur Phpunit Phar Datei angegeben
Desweiteren sollte unter "Phpunit: Args" ["--configuration", "tests/phpunit.xml"] eingegeben werden.

Alternativ können diese Einstellungen in die "settings.json" eingeben werden.

```
"phpunit.phpunit": "Path\\To\\phpunit-9.5.10.phar",
"phpunit.args": [
"--configuration", "tests/phpunit.xml"
],
"phpunit.php": "Path\\To\\php.exe",
"phpunit.files": "{test,tests}/**/*Test.php",
```

Jetzt kann PHPUnit genutzt werden. Es können allerdings noch Fehler auftauchen. So kann der folgende Fehler auftauchen:

```
"PHPUnit requires the "mbstring" extension."
```

Zum Ordner in dem die "php.exe" liegt navigieren
Die Datei "php.ini" suchen in diesem Ordner suchen. Sollte diese nicht vorhanden sein nach "php.ini-development" suchen und diese umbennenen
Die Datei "php.ini" in einem beliebigen Text-Editor öffnen
Im Abschnitt "Dynamic Extensions" nach "mbstring" suchen
Vor "extension=mbstring" das Semikolon entfernen
Datei speichern
Visual Studio Code schließen und wieder öffnen

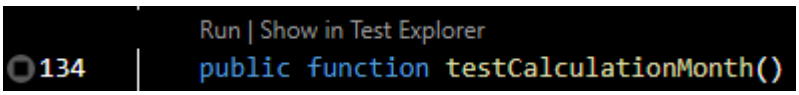
Nun kann es zu einem weiteren Fehler kommen:

```
PHP Warning: PHP Startup: Unable to load dynamic library 'mbstring'
(tried: C:\php\ext\mbstring (Das angegebene Modul wurde nicht
gefunden.), )
C:\php\ext\php_mbstring.dll (Das angegebene Modul wurde nicht
gefunden.) in
Unknown on line 0
```

Zum Ordner in dem die "php.exe" liegt navigieren
Die Datei "php.ini" suchen in diesem Ordner suchen. Sollte diese nicht vorhanden sein nach "php.ini-development" suchen und diese umbennenen
Die Datei "php.ini" in einem beliebigen Text-Editor öffnen
Im Abschnitt "Paths and Directories" nach "extension_dir" suchen
Vor "extension_dir = "ext"", welches unter "On windows:" steht, das Semikolon entfernen
Datei speichern
Visual Studio Code schließen und wieder öffnen

## PHPUNIT NUTZEN

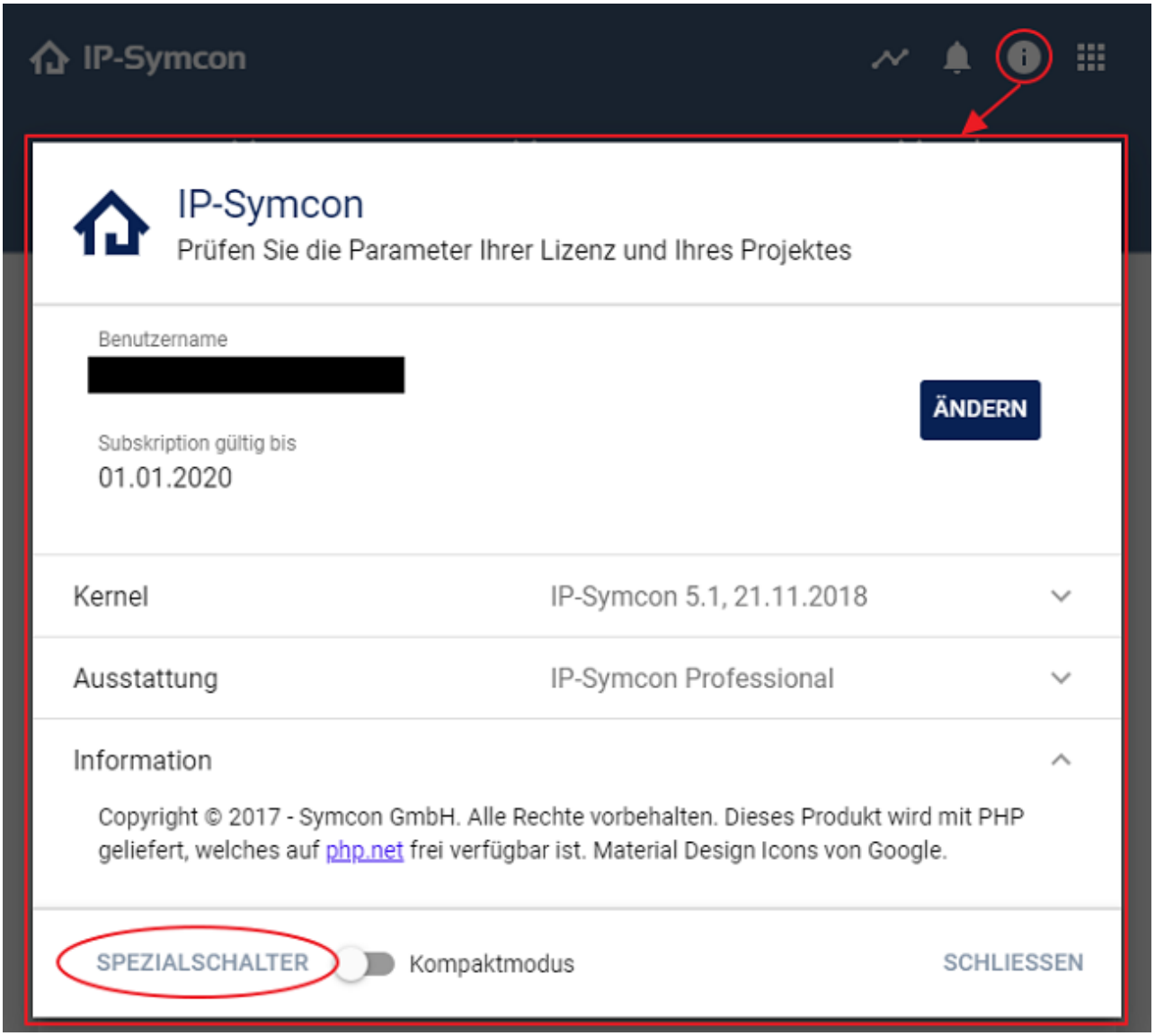
Nun sollte über der Funktion mit dem Test zwei Schaltflächen angezeigt werden. Mit einem Klick auf Run wird der Test gestartet.



# Spezialschalter

Benötigt: IP-Symcon >= 2.6
Um einige Funktionen von IP-Symcon besser kontrollieren zu können, ist es möglich, einige Schalter innerhalb der Verwaltungskonsole zu verändern.

Es ist zu beachten, dass jegliche Änderungen erst wirksam werden, sobald IP-Symcon neu gestartet wurde.




Folgende Schalter stehen zur Verfügung:

| OPTIONSNAME | SPEZIALSCHALTERFUNKTION | BESCHRE |
|---|---|---|
| ArchiveCommitInterval<br>(ab 5.0) | Zeitintervall des Archivs | **Default = 60**. Sekund<br>geloggte Werte auf d<br>geschrieben werden. |
| ArchiveRecordLimit<br>(ab 4.3) | Anzahl der genutzten Datenpunkte im Archiv | **Default = 10000**. Ma<br>die Limit Parameter d<br>Funktionen. Dies kor<br>den maximal genutzt<br>zum Rendern und Ag<br>Graphen. Achtung: H<br>verlangsamt das Ren<br>Nur für leistungsstarl<br>geeignet. |
| ArchiveRecordLimitBool<br>(ab 5.0) | Anzahl der genutzten Roh-Datenpunkte im Archiv | **Default = 500**. Maxim<br>Limit Parameter der A<br>Funktionen. Dies kor<br>den maximal genutzt<br>Datensätzen zum Re<br>Aggregieren von Gra<br>Höhere Anzahl verlar<br>Rendern von Graphe<br>leistungsstarke Syste |
| BackupCount | Anzahl der Backups | **Default = 25**. Anzahl<br>Settings, die im 'back<br>archiviert werden. Ei<br>Settings wird jeweils<br>und kurz nach Mitter<br>Zahl im Dateinamen i<br>UnixTimeStamp des<br>Wird der Wert Null (0<br>werden unbegrenzt v<br>erstellt. (Achtung,<br>Speicherplatzverbrau |
| CompatibilityRequired<br>(ab 4.0) | Kompatibilitätsfunktionen | **Default = Deaktivier<br>Aktiviert)**. Sofern **akt**<br>IP-Symcon 4.0 die<br>Kompatibilitätsfunkt<br>veraltete Funktionen<br>verfügbar. Bei Aktivie<br>Spezialschalters erhö<br>Laufzeit jedes PHP-Sl<br>wennmöglich deaktiv<br>Diese Funktion ist un<br>aufgrund der Abwärt<br>standardmäßig aktivi |
| ConnectCheck | Überprüfungsintervall vom Connect Dienst | **Default = 30 (in Minu**<br>Minuten, in welchem<br>vom Connect Dienst |
| ConnectLimit<br>(ab 5.4) | Limit der Verbindungsversuche | **Default = 25** (Maxima<br>maximale Anzahl an<br>Verbindungsversuche<br>24 Stunden des Conn<br>Alle Anfragen darübe<br>mit einer Fehlermeld<br>und ins Logfile gesch |