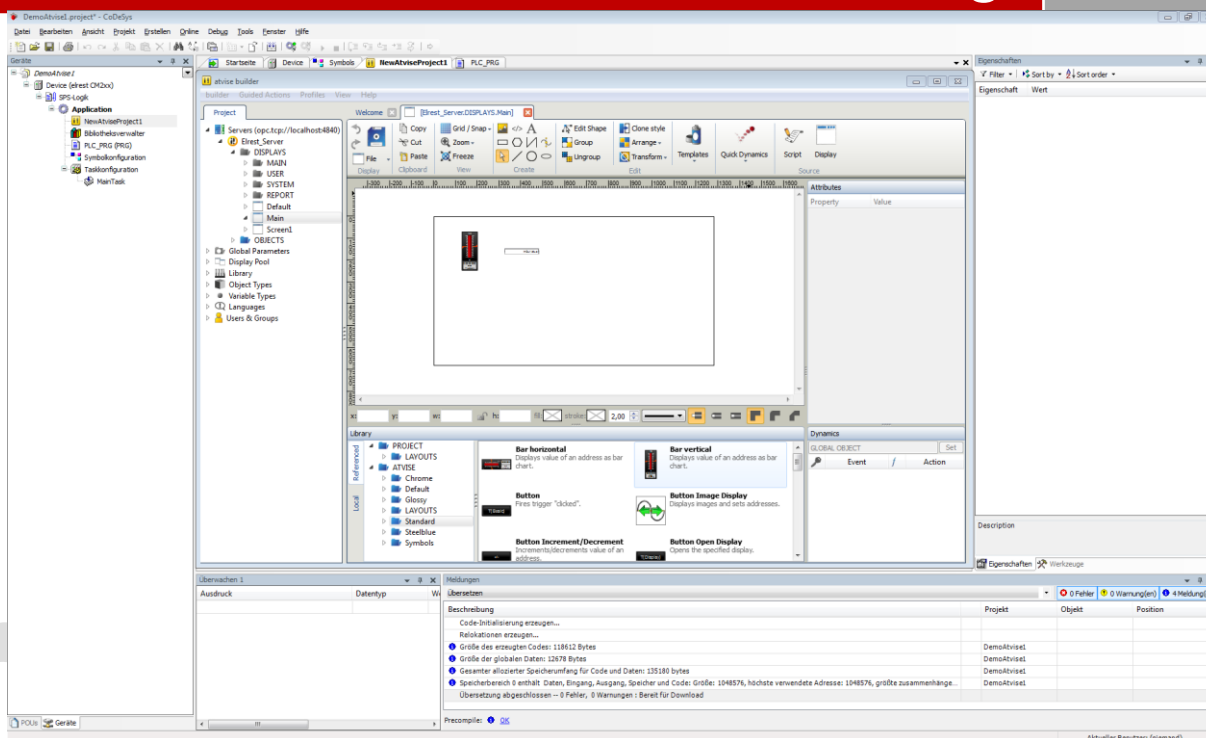


# Handbuch CoDeSys V3.5 SP3

## Installation und Handhabung



elrest Automationssysteme GmbH  
Leibnizstraße 10  
73230 Kirchheim unter Teck  
Germany  
Telefon: +49 (0) 7021 / 92025-0  
www.elrest.de

**elrest®**

# Inhaltsverzeichnis

1	Allgemein.....	7
1.1	Vorwort.....	7
1.2	Hard- und Softwareanforderungen der Softwareprodukte.....	7
1.3	Softwarelizenz.....	7
1.4	Copyright.....	8
1.5	Updates.....	9
1.6	Handbücher.....	9
1.7	Serviceleistungen.....	10
1.8	Lieferumfang.....	10
1.9	Bestimmungsgemäße Verwendung.....	10
1.10	Gewährleistung.....	11
1.11	Sicherheitsrichtlinien und Schutzmaßnahmen.....	11
2	CODESYS V3.....	13
2.1	CODESYS - Marktstandard in IEC 61131-3.....	13
2.2	CODESYS Target Visualisierung.....	13
2.3	CODESYS SoftMotion.....	14
3	Installation.....	16
3.1.1	CoDeSys V3 Installation.....	16
3.1.2	CoDeSys Package Manager Installation.....	16
4	Basiskomponenten.....	18
4.1	Basiskonzepte.....	18
4.2	Unterstützte Programmiersprachen.....	19
4.3	Profile und Plug-Ins.....	20
4.4	Was ist ein Programm.....	21
4.5	Gerät.....	22
4.6	Applikation.....	26
4.7	Taskkonfiguration.....	27
4.8	Kommunikation.....	29
4.8.1	Steuerungskonfiguration.....	29
4.8.2	Adressierung und Routing.....	30
4.8.3	Data Server.....	32
4.8.4	Netzwerkvariablen.....	32
4.9	Code-Generierung und Online Change.....	33
4.10	Boot application (boot project).....	36
4.11	SPS-Operationssteuerung über Systemvariablen.....	36

4.12	Script Sprache.....	38
4.13	Bibliotheksverwaltung .....	39
4.13.1	Installation und Einbinden im Projekt.....	39
4.13.2	Referenzierte Bibliotheken .....	40
4.13.3	Bibliotheksplatzhalter, Bibliotheksprofile .....	40
4.13.4	Eindeutiger Zugriff auf Bibliotheksmodule oder –variablen.....	41
4.13.5	CODESYS V2.3 Bibliotheken.....	42
4.13.6	Externe und interne Bibliotheken bzw. Bibliotheksbausteine, spätes Binden .....	42
4.13.7	Lizenzierte Bibliotheken .....	43
4.14	Alarmverwaltung.....	44
4.15	Package Manager .....	45
5	Geräte Schnittstellen Konfigurationen .....	46
5.1	CANbus Konfigurationseditor.....	46
5.2	EtherCAT Konfigurationseditor .....	46
5.3	Modbus Konfigurationseditor .....	46
6	Target Visualisierung .....	47
6.1	Übersicht .....	47
6.1.1	Client Seite .....	47
6.1.2	Server Seite.....	48
6.2	Projektverwaltung.....	48
6.2.1	Bilder integrieren (GlobalImagePool).....	48
6.3	Tools und Editoren der Visualisierung .....	48
6.3.1	Erzeugen einer Visualisierung .....	51
6.3.2	Visualisierungsbefehle .....	52
6.3.2.1	Visualisierungselement einfügen.....	52
6.3.2.2	Elementliste .....	53
6.3.2.3	Tastaturbedienung aktivieren .....	53
6.3.2.4	Ausrichtung.....	53
6.3.2.5	Gruppieren.....	53
6.3.2.6	Hintergrund.....	54
6.3.2.7	Einfügen von Elementen zur Alarmquittierung.....	54
6.3.2.8	Frame-Parameter in <geöffnete Visualisierung> aktualisieren.....	55
6.3.3	Visualisierungseditor .....	55
6.3.4	Visualisierungsobjekt.....	57
6.3.4.1	Eingabedialoge .....	58
6.3.4.2	Programmatischer Zugriff auf Eingabedialoge.....	63

6.3.4.3	Ereignisse und Eingabeaktionen bemerken .....	67
6.3.4.4	Tastatur-Ereignisse bemerken.....	70
6.3.5	Visualisierungselemente .....	72
6.3.5.1	Benutzereingabe beim Editieren eines Elements .....	72
	Was kann mit einem Visualisierungselement gemacht werden? .....	72
	Text und Sprache in Visualisierungen .....	74
	Eingabekonfiguration .....	77
	Farbverlaufeditor .....	91
	Zugriffsrechte.....	92
6.3.5.2	Basis .....	93
	Rechteck, abgerundetes Rechteck und Ellipse.....	93
	Linie .....	102
	Polygon, Linienzug oder Bézierkurve .....	111
	Kreis Sektor .....	121
	Bild .....	131
	Frame .....	141
6.3.5.3	Allgemeine Steuerelemente .....	153
	Text .....	153
	Combobox Integer .....	155
	Combobox Array .....	157
	Registersteuerelement .....	161
	Schaltfläche .....	163
	GroupBox .....	173
	Tabelle .....	175
	Textfeld.....	188
	Scrollbalken .....	195
	Schieberegler .....	199
	SpinControl .....	203
	Unsichtbare Eingabe .....	207
	Ladebalken .....	211
	Radiobutton .....	213
	Checkbox .....	215
6.3.5.4	Alarmmanager.....	217
	Alarmtabelle .....	217
	Alarmbanner .....	223
	Auswahl Alarmgruppe.....	226

Auswahl Alarmklasse .....	227
6.3.5.5 Zeigerinstrumente/Potentiometer/Histogramm .....	228
Balkenanzeige, Balkenanzeige frei konfigurierbar .....	228
Zeigerinstrument 90° .....	232
Zeigerinstrument 180° .....	238
Zeigerinstrument 360° .....	243
Zeigerinstrument Frei Konfigurierbar .....	248
Potentiometer .....	256
Potentiometer Frei Konfigurierbar .....	261
Histogramm .....	265
6.3.5.6 Lampen/Schalter/Bilder .....	272
Bildwechsler .....	272
Lampe.....	276
Hebelschalter, Energieschalter, Drückschalter, Drückschalter LED, Kippschalter .....	279
Drehschalter .....	281
6.3.5.7 Spezielle Steuerelemente .....	285
Trace und seine Dialoge.....	285
ActiveX-Element .....	285
Wartesymbol Würfel .....	289
Wartesymbol Blume .....	290
Texteditor .....	292
Path3D.....	298
Kurvenscheiben-Anzeiger .....	306
6.3.6 Optionen .....	307
6.3.6.1 Visualisierungen.....	307
6.3.6.2 Visualisierungsstile .....	309
6.3.6.3 Visu-Benutzerverwaltung.....	310
6.3.7 Projekteinstellungen .....	311
6.3.8 Visualisierungsprofile .....	313
6.3.9 Visualisierungsbibliotheken .....	314
7 Bibliotheken.....	315
7.1 Richtlinien für das Erstellen von Bibliotheken .....	315
7.2 Standardbibliothek.....	315
7.3 Util Bibliothek.....	315
7.4 Busspezifische Bibliotheken.....	315
7.5 Systembibliotheken .....	315

7.6	(CAA-Bibliotheken).....	315
8	Projektierung .....	316
8.1	Erstellen einer ersten Visualisierungsaufgabe .....	316
8.1.1	Bildschirmformat festlegen .....	316
8.1.2	Neue Visualisierungs Maske anlegen .....	316
8.1.3	Wechseln von Masken .....	316
8.1.4	Wechseln von Masken in einem eigenen Fenster .....	317
8.1.5	Eingabefelder .....	317
8.2	Globale Parameter .....	317
8.2.1	Eigene Elemente erstellen .....	318
9	Anhang .....	319
9.1	TODO's .....	319
9.2	Grundlagen.....	319
9.3	Dateiformate.....	320
9.4	Wichtigste Linux Befehle.....	320
10	Support.....	321
	Hotline .....	321
	Training und Workshops.....	321
11	Historie .....	321
	Datum .....	321
	Name .....	321
	Kapitel.....	321
	Änderung.....	321

# 1 Allgemein

## 1.1 Vorwort

Dieses Handbuch enthält Texte, Abbildungen und Erläuterungen zur korrekten Installation und Bedienung der visio Baugruppen. Vor der Installation und dem Einsatz der Geräte muss dieses Handbuch gelesen und beachtet werden.

Bei Fragen zur Installation, Anwendung und Bedienung wenden Sie sich bitte an die elrest-Kunden-Hotline:

Tel.:07021/92025-33

Fax:07021/92025-59

E-Mail: hotline@elrest.de

oder an Ihre zuständige Vertretung.

Dieses Handbuch wird vorbehaltlich etwaiger Änderungen herausgegeben. Änderungen können ohne Hinweis vorgenommen werden.

## 1.2 Hard- und Softwareanforderungen der Softwareprodukte

- Microsoft® Windows XP oder Windows 7
- IBM® kompatibler PC mit Pentium- Prozessor
- RAM- Speicher von mindestens 1 GB Arbeitsspeicher
- Festplatte mit mindestens 5 GB freiem Speicherplatz
- Maus oder Trackball, Tastatur
- Graphikkarte mit Mindestauflösung von 1024x768 bei 24-bit Farbtiefe (True Color)

## 1.3 Softwarelizenz

Das Softwareprodukt ist urheberrechtlich geschützt. Sie erwerben eine nicht ausschließliche Benutzungslizenz für das Softwareprodukt. Das Softwareprodukt umfasst Computersoftware, sowie möglicherweise dazugehörige Medien, gedruckte Materialien und eine "online" oder elektronische Dokumentation incl. Zubehör ("SOFTWARE-PRODUKT").

Indem Sie das SOFTWARE-PRODUKT installieren, kopieren oder anderweitig verwenden erklären Sie sich einverstanden, durch die Bestimmungen dieses EULAs (enduser license agreement) gebunden zu sein. Falls Sie den Bestimmungen dieses EULAs nicht zustimmen, sind Sie nicht berechtigt, das SOFTWARE-PRODUKT zu installieren oder zu verwenden.

Das SOFTWARE-PRODUKT wird sowohl durch Urheberrechtsgesetze und internationale Urheberrechtsverträge als auch durch andere Gesetze und Vereinbarungen über geistiges Eigentum geschützt. Das SOFTWARE-PRODUKT wird lizenziert, aber nicht verkauft.

Der Lizenzgeber gewährt Ihnen eine nicht ausschließliche Lizenz zur unentgeltlichen Nutzung der Software, sofern

Ihre Nutzung der Software dem ausschließlichen Zweck der Bewertung dient, ob eine entgeltliche Lizenz zur regulären Nutzung der Software erworben werden soll. Die Bewertung ist für Wirtschaftsunternehmen auf neunzig (90) Tage begrenzt; die Bewertung durch andere Nutzer unterliegt dieser Begrenzung auf neunzig (90) Tage nicht. Behörden (mit Ausnahme von öffentlichen Bibliotheken) gelten nicht als Ausbildungsstätten oder religiöse oder gemeinnützige Einrichtungen im Sinne dieses Vertrages. Soweit Sie die Software unentgeltlich nutzen, haben Sie kein Anrecht auf ausgedruckte Dokumentation, Unterstützungsleistungen oder telefonische Beratung.

Sie dürfen:

die Software auf einem einzelnen Computer nutzen;

die Software auf einem zweiten Computer nutzen, wenn jederzeit nur eine (1) Kopie der Software genutzt wird;

die Software auf einem Netzwerk benutzen, vorausgesetzt dass für jede Person, die über das Netzwerk Zugriff auf die Software hat, eine lizenzierte Kopie der Software erworben wurde;

die Software für Archivierungszwecke kopieren, sofern jede Kopie alle Schutzrechtsvermerke der Originalversion enthält;

Alle Programme stehen ohne Schutzstecker (Dongle) als Demo-Version zur Verfügung. Für den Einsatz einzelner oder aller Programme benötigen Sie den entsprechenden Schutzstecker (Dongle).

## 1.4 Copyright

Copyright © 2013 elrest Automationssysteme GmbH Corporation (wird in weiterer Folge "elrest" genannt). sind alle Rechte vorbehalten.

Alle Teile der Software und der Dokumentation unterliegen dem Urheberrecht. Die in diesem Handbuch beschriebene Software darf ausschließlich im Rahmen der Lizenzbedingungen genutzt werden.

Kein Teil der Dokumentation und Software darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder einem anderen Verfahren) ohne schriftliche Genehmigung der Firma elrest Automationssysteme GmbH reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden. Hiervon sind die in den Paragraphen 53 und 54 UrhG ausdrücklich genannten Ausnahmefälle nicht berührt.

Es wurden alle Anstrengungen unternommen, die Richtigkeit und Vollständigkeit der Angaben in dieser Dokumentation zu gewährleisten. Trotzdem können Fehler nicht ausgeschlossen werden. Die Firma elrest Automationssysteme GmbH kann keine juristische Verantwortung noch irgendeine Haftung übernehmen für Schäden, die durch die Benutzung von Informationen aus diesem Handbuch oder durch die Nutzung des in dieser Dokumentation beschriebenen Programms entstehen.

Die in diesem Handbuch erwähnten Produktnamen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Herstellerfirmen und werden hiermit anerkannt.

Die in diesem Dokument enthaltenen Informationen können ohne Vorankündigung geändert werden und stellen keine Verpflichtung seitens elrest Automationssysteme GmbH dar.



## 1.5 Updates

Wenn das SOFTWARE-PRODUKT als Update gekennzeichnet ist, müssen Sie über die entsprechende Lizenz für ein Produkt, das von elrest als für das Update geeignet anerkannt wird, verfügen, um das SOFTWARE-PRODUKT verwenden zu dürfen. Ein als Update bezeichnetes SOFTWARE-PRODUKT ersetzt und/oder ergänzt das Produkt, das die Basis für das Update bildet.

Sie sind verpflichtet, das Update-Produkt nur in Übereinstimmung mit den Bestimmungen dieses EULAs zu verwenden.

Wenn das SOFTWARE-PRODUKT ein Komponenten-Update eines Pakets von Software-Programmen ist, das für Sie als einzelnes Produkt lizenziert wurde, sind Sie nur berechtigt, das SOFTWARE-PRODUKT als Bestandteil dieses einzelnen Produktpakets zu verwenden und zu übertragen.

Das SOFTWARE-PRODUKT darf nicht getrennt vom Paket auf einem anderen Computer verwendet werden.

Haben Sie Fragen zu diesem Vertrag, wenden Sie sich bitte an elrest Automationssysteme GmbH, Leibnizstr. 10, 73230 Kirchheim,

Tel: ++49 / (0) 7021 / 92025 - 0, Fax: - 29, Email: [Vertrieb@elrest.de](mailto:Vertrieb@elrest.de), Internet: <http://www.elrest.de>

## 1.6 Handbücher



Handbücher der einzelnen Programmteile und Tools im PDF Datenformat.



Programm zum Öffnen und Drucken der Handbücher und anderer Dateien im PDF-Datenformat.

## 1.7 Serviceleistungen

elrest bietet Ihnen je nach Vereinbarung Serviceleistungen in Verbindung mit dem SOFTWARE-PRODUKT ("Serviceleistungen").

Die Serviceleistungen können entsprechend den elrest Bestimmungen und Programmen, die im Benutzerhandbuch, der Online-Dokumentation und/oder anderen von elrest zur Verfügung gestellten Materialien beschrieben sind, genutzt werden.

Jeder ergänzende Softwarecode, der Ihnen als Teil der Serviceleistungen zur Verfügung gestellt wird, wird als Bestandteil des SOFTWAREPRODUKTS betrachtet und unterliegt den Bestimmungen und Bedingungen dieses EULAs. elrest ist berechtigt, die technischen Daten, die Sie elrest als Teil der Serviceleistungen zur Verfügung stellen, für geschäftliche Zwecke, einschließlich der Produktunterstützung und -entwicklung, zu verwenden.

elrest verpflichtet sich, solche technischen Daten ausschließlich anonym im Sinne des Datenschutzes zu verwenden.

## 1.8 Lieferumfang

Im Lieferumfang von eStudio sind enthalten:

- CODESYS DVD-ROM
- Beschreibung CODESYS Schnelleinstieg

Zusätzlich werden je nach Vereinbarung individuelle Demosets incl. entsprechender Hard- und Software vom Vertrieb zur Verfügung gestellt.

## 1.9 Bestimmungsgemäße Verwendung

Die Software eStudio ist geeignet für die Verwendung in den Bereich der Regelungs-, Steuerungs- und Automatisierungstechnik. Die mögliche Anwendung reicht von privaten und gewerblichen Anwendungen bis zu industriellen Anwendungen. In allen Anwendungen und speziell bei Verwendung von induktiven Lasten (Motoren und Relais usw.) muss sichergestellt werden, dass auftretende Spannungsspitzen die maximalen Eingangsspannungen der Ein- und Ausgänge nicht überschreiten. Falls erforderlich, müssen externe schützende Schaltungsteile installiert werden.

## 1.10 Gewährleistung

Ein Gewährleistungsanspruch setzt eine fachgerechte Montage und Inbetriebnahme nach der für das Gerät gültigen Montage-, Inbetriebnahme- und Bedienungsanleitung voraus. Die erforderlichen Montage-, Inbetriebnahme- und Wartungsarbeiten dürfen nur von sachkundigen und autorisierten Personen durchgeführt werden. Sehen Sie hierzu unsere EULA Bestimmungen.

### Hersteller

elrest Automationssysteme GmbH  
D-73230 Kirchheim unter Teck  
Leibnizstraße 10  
Telefon: +49 (0) 7021/92025-0  
Fax: +49 (0) 7021/92025-29

### Handelsmarke



### Ursprungsland

Federal Republic of Germany

## 1.11 Sicherheitsrichtlinien und Schutzmaßnahmen

Dieses Handbuch wurde für geschultes und kompetentes Personal erstellt. Die Qualifizierung wird durch die "europäischen Richtlinien für Maschinen, Niederspannungen und EMV" definiert. Der Anschluss und die Montage der visio Baugruppen darf bei Spannungen größer der Schutzkleinspannung nur durch eine Elektrofachkraft erfolgen.

Die nationalen Vorschriften und jeweils gültigen Sicherheitsbestimmungen sind zu beachten. Eingriffe und Veränderungen an den Geräten führen zum Erlöschen des Garantieanspruches.

In diesem Handbuch werden zur Hervorhebung von bestimmten Informationen verschiedene Symbole verwendet. Hiermit erhält das Bedienpersonal notwendige Hinweise zu den Sicherheits- und Schutzmaßnahmen. Bei jedem Auftreten der Symbole muss der zugehörige Hinweis gelesen werden.



Bezeichnet eine unmittelbar drohende Gefahr, die zu einem Personen- oder Sachschaden führen kann.



Bezeichnet eine möglicherweise auftretende Gefahr, die zu einem Personen- oder Sachschaden führen kann



Bezeichnet Hinweise, damit die Handhabung einfacher wird.



(\* Kommentar zu CoDeSys Code Zeilen \*)

```
a := a+1;
```

ST





## Shell

elrest Automationssysteme GmbH übernimmt unter keinen Umständen die Haftung oder Verantwortung für Schäden, die aus einer unsachgemäßen Installation oder Anwendung der Geräte oder des Zubehörs entstanden ist.

Alle Beispiele und Abbildungen in diesem Handbuch dienen nur als Hilfe zum Verstehen des Textes. Für die Richtigkeit der dargestellten Bedienvorgänge kann keine Gewährleistung übernommen werden. elrest Automationssysteme GmbH übernimmt keine Verantwortung für eine Produktanwendung, die sich auf die dargestellten Beispiele (z.B. in eStudio Demo) bezieht.

Aufgrund der großen Anzahl von verschiedenen Anwendungsmöglichkeiten dieser Geräte müssen Sie die Anpassung für Ihren speziellen Anwendungsfall selbst vornehmen.

Wenn Schaltungskomponenten ausfallen sollten, müssen entsprechende Sicherheitseinrichtungen dafür sorgen, dass die angeschlossene Peripherie angehalten wird.

Versuchen Sie nicht, die visio Baugruppen selbst zu reparieren oder elektrische Teile auszutauschen. Wenden Sie sich hierfür ausschließlich an die elrest Service Abteilung. Kontakt können Sie über die elrest-Hotline aufnehmen.

Beachten Sie bei Installation und Einsatz der visio Baugruppen die lokalen und nationalen Normen und Vorschriften.

## 2 CODESYS V3

elrest ist mehr als 10 Jahren Partner der Fa. 3S Software ([www.3s-software.com](http://www.3s-software.com)).

CODESYS ist ein SPS Programmierool mit verschiedenen Sprachelementen. CODESYS ist mittlerweile das führende SPS Entwicklungssystem. Das bedeutet, dass Tausende von Anwendern täglich CODESYS nutzen.

elrest setzt auf den Standard CODESYS.

### 2.1 CODESYS - Marktstandard in IEC 61131-3

CODESYS (Controller Development System) ist eine SPS-Programmiersoftware, mit der Sie Steuerungen, Embedded und PC basierte Geräte nach dem internationalen Standard IEC 61131-3 programmieren können. Es besteht aus dem Programmiersystem CODESYS sowie dem Laufzeitsystem CODESYS Control. Mit dem Laufzeitsystem wird Ihr Gerät zu einer Steuerung, die ganz einfach mit CODESYS programmiert werden kann. Integrierte native Code-Generatoren (Compiler) sorgen dafür, dass der Programmcode mit optimaler Geschwindigkeit abgearbeitet wird.

Über 250 namhafte Gerätehersteller aus unterschiedlichen Branchen setzen das Software-Tool CODESYS als Programmierool für ihre Automatisierungsgeräte ein. Damit nutzen Tausende von Endanwendern aus dem Maschinen- und Anlagenbau und weiteren Industriezweigen CODESYS bei ihrer täglichen Arbeit. Das sind mehr als bei jedem anderen hardwareunabhängigen IEC 61131-3 Entwicklungssystem. Damit ist CODESYS de facto Marktstandard.

Und das Beste daran: Sie können sich CODESYS nach der [Registrierung](#) kostenlos [herunterladen](#)! Oder Sie sehen sich das [Quick Start Projekt \(Tutorial\)](#) bzw. die [Product Tour](#) an, um einen ersten Eindruck von CODESYS zu bekommen.

### 2.2 CODESYS Target Visualisierung

#### [Web-Visualisierung](#)

Optional generiert CODESYS aus den Visualisierungsinformationen eine XML-Beschreibung, die zusammen mit einem Java-Applet auf der Steuerung abgelegt und via TCP/IP auf einem Browser dargestellt werden kann. Somit stehen die Visualisierungsdaten auf den unterschiedlichsten Plattformen online zur Verfügung.

#### **Target-Visualisierung**

Für Steuerungen mit eingebautem Display können die Visualisierungsinformationen aus dem Programmiersystem in IEC 61131-3 Code umgewandelt und mit dem Codegenerator für das entsprechende Zielsystem übersetzt werden. Diese Lösung kann auf beliebige mit CODESYS programmierbare Geräte portiert werden.



## 2.3 CODESYS SoftMotion

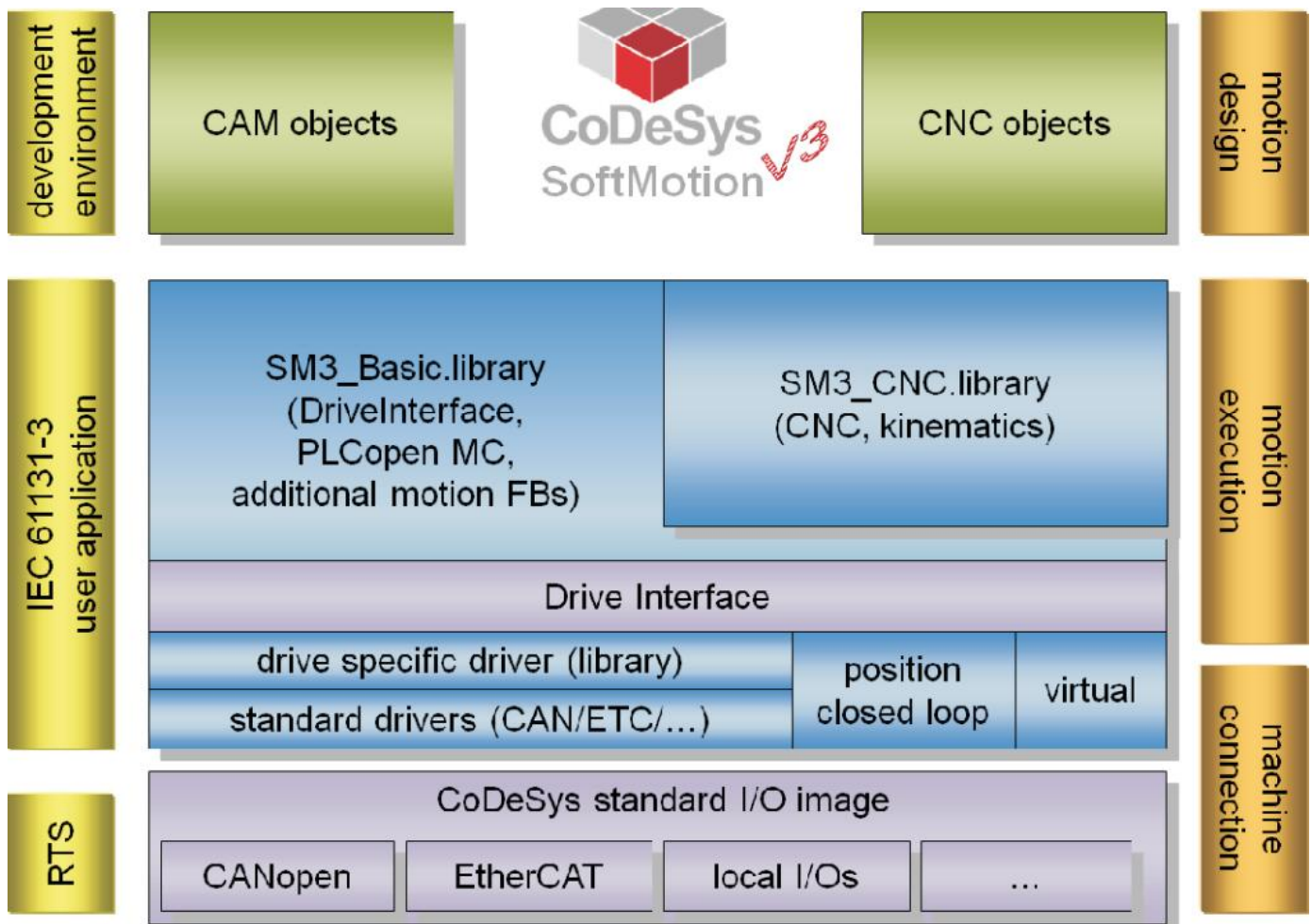
Motion-Funktionalität integriert in das CODESYS Programmier- und Laufzeitsystem – das ist CODESYS Soft-Motion. Die Einsatzbereiche reichen von einachsigen Verfahrbewegungen bis hin zu CNC-Bahninterpolationen. Klassische Lösungen realisieren die Bewegungsaufgaben mit monolithischen, hardwareabhängigen Steuerungssystemen. Im Gegensatz dazu liefert CODESYS SoftMotion die Bewegungsfunktionalität in Form eines Baukastens im SPS-Programmiersystem. Daraus können Sie anhand der IEC-Sprachmittel beliebig komplexe Lösungen erstellen. Die Achsen werden in abstrahierter Form als Datenstruktur zur Verfügung gestellt – unabhängig von Busanbindung und Antriebshersteller. Die konkrete Anbindung an den Feldbus erfolgt durch eine Treiberschnittstelle. Die Konfiguration der Antriebe erfolgt im Rahmen der Geräte-Konfiguration im Projektbaum. Aufbauend auf diese Konzeption werden verschiedene Modelle der Bewegungssteuerung als Baukasten angeboten:

- Ein- bzw. Mehrachs-Verfahrbewegungen mit PLCopen Motion Bausteinen
- Elektronische Kurvenscheibe
- Elektronisches Getriebe
- 3D CNC-Steuerung

CODESYS SoftMotion beinhaltet

- Eine Bausteinbibliothek mit sämtlichen Funktionsblöcken zur Bewegungssteuerung nach der Festlegung der IEC 61131-3-Nutzerorganisation PLCopen
- Eine CNC-Bausteinbibliothek, die Ihnen vom Interpolator bis zur Bahnvorverarbeitung alles bietet, um eine Bewegungssteuerung aufzubauen. Das Design der Bibliothek erlaubt dabei die problemlose Kombination aller Bausteine, sowie die Verteilung der Applikation in verschiedene Tasks.
- Treiber für gängige Antriebsschnittstellen, wie CAN, Sercos oder Analog-Ansteuerung  
Anpassungen an die gängigsten Antriebssysteme, z.B. von Beckhoff, Control Techniques, Copley, Danaher, Infranor, JAT, KEB, Metronix, Nanotec, Parker, Schneider Electric, Stöber und weitere
- Grafische Editoren zur Planung und Eingabe von Bewegungen:
  - Grafischer 3D DIN 66025 Editor (G-Code)
  - Kurvenscheibeneditor
- Komplexe Visualisierungselemente für sämtliche Bibliotheksfunktionen, um schnell Bedien- und Testmasken erzeugen zu können

CODESYS SoftMotion ist portabel auf unterschiedliche CODESYS-Plattformen mit 32-Bit CPU und Fließkomma-Unterstützung. Mit [CODESYS Control RTE](#) kann CODESYS SoftMotion auch auf jedem Standard-PC betrieben werden.



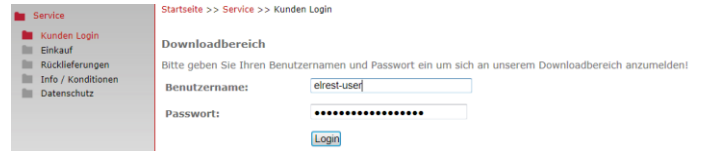
## 3 Installation

### 3.1.1 CoDeSys V3 Installation

Der einfachste Weg, installieren Sie einfach von der [elrest Homepage](#).

Nach Service → Kunden Login  
kommen Sie auf den öffentlichen Downloadbereich. Das  
Passwort erhalten Sie erstmalig und unmittelbar nach  
einer Anfrage an

[support@elrest-gmbh.de](mailto:support@elrest-gmbh.de)



	Dateiname	Geändert
■ Kunden Login		
■ Öffentlich		
■ Einkauf		
■ Rücklieferungen		
■ Info / Konditionen		
■ Datenschutz		
	■ Bilder	12-10-11 08:18
	■ Company	17-05-10 15:49
	■ FAQ	02-09-11 11:40
	■ Hardware	19-07-10 10:46
	■ Software	15-06-12 10:06
	■ kundenuploads	30-11-11 14:38
	■ produktuebersicht	18-05-10 16:30

Wechseln Sie nach: **Öffentlich** → **Software** → **CODESYS** → **V3** → **CODESYS\_Install**

Wählen Sie die höchste Versionsnummer aus. Kopieren  
Sie die Datei auf Ihr lokales Laufwerk und installieren Sie  
indem Sie bspw.

**Setup\_CODESYSV35SP3.exe**

aufrufen.



Die Datei hat eine Größe von ca. 0,5 GB.

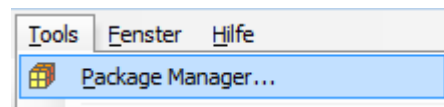
Sowohl der Download, als auch die Installation dauert einige Zeit.


### 3.1.2 CoDeSys Package Manager Installation

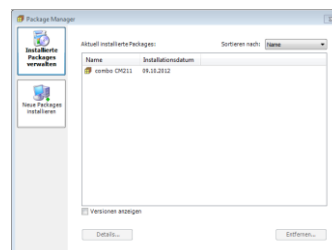
Wechseln Sie nach: **Öffentlich** → **Software** → **CODESYS** → **V3** → **Package Manager**

Hier sind für jede Geräteausführung die \*.package Dateien abgelegt.

Innerhalb von CODESYS starten Sie den Package  
Manager.



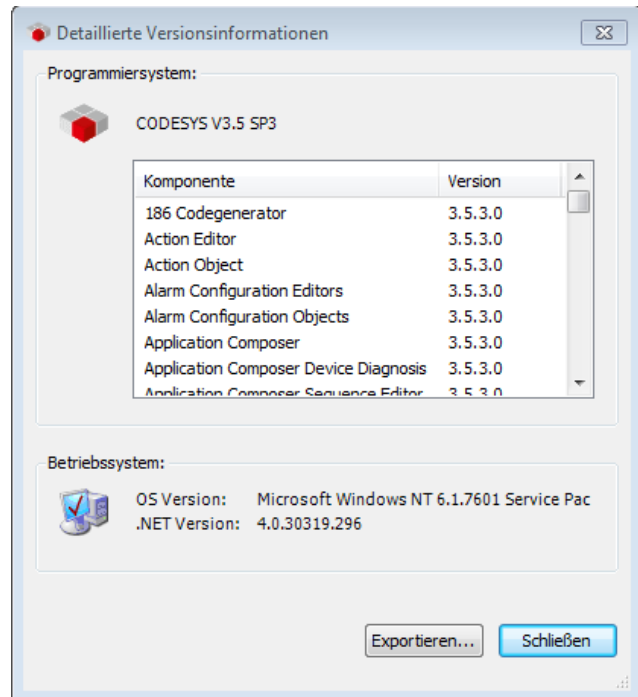
Mit dem  können Sie die gewünschten Geräte  
installieren.







Nun am Ziel angekommen, die Installation ist abgeschlossen. Sie können mit der Programmierung des ersten Gerätes beginnen.



## 4 Basiskomponenten

### 4.1 Basiskonzepte

Gemäß der Norm IEC 61131-3 unterstützt es alle Standard-Programmiersprachen, erlaubt aber auch das Einbinden von C-Routinen. In Kombination mit dem CODESYS Laufzeitsystem ermöglicht es die Programmierung mehrerer Steuerungen innerhalb eines Projektes.

Beachten Sie die folgenden Basiskonzepte, die das Programmieren mit CODESYS bestimmen:

- **Objektorientierung:** Der Gedanke der Objektorientierung äußert sich nicht nur durch die Verfügbarkeit von entsprechenden Programmierelementen und Programmfunktionen, sondern auch in der Struktur und der Versionsverwaltung von CODESYS und in der Projektorganisation. "Multi-device" und "Multi-Application" Gebrauch eines CODESYS Projektes ist möglich durch instanzierbare und gemeinsam nutzbare Programmeinheiten. Verschiedene Applikationen können auf einem Gerät betrieben werden. Parametrierbare und programmierbare Geräte können innerhalb desselben Projektes angesprochen werden.
- **Komponentenbasierte Struktur des Programmiersystems:** Welche Funktionalität das Programmiersystem dem Anwender bietet (Editoren, Menüs etc.) hängt von den gerade verwendeten Komponenten ab. Dies wiederum ist definiert im sogenannten Profil. Es gibt Systemkomponenten, die unverzichtbar sind, sowie optionale Komponenten. Zusätzlich zu den Standard 3S-Smart Software Solutions GmbH Plug-ins können kundenspezifische Plug-ins verwendet werden, die mit Hilfe des Automation Platform Toolkit erstellt werden.
- **Versionsverwaltung:** Verschiedene Versionen der CODESYS Komponenten können parallel installiert sein und dann kann mit der jeweils gewünschten gearbeitet werden. Dies umfaßt auch den gerätespezifischen Gebrauch verschiedener Compiler-Versionen. Einzelne Funktionalitäten können hinzugefügt werden ohne einen Update der gesamten Version zu erfordern.
- **Projektorganisation:** Diese ist ebenfalls durch den Gedanken der Objektorientierung geprägt: Ein CODESYS Projekt enthält sowohl ein Steuerungsprogramm, zusammengesetzt aus verschiedenen Programmierobjekten, als auch die Definitionen der "Ressourcen" die benötigt werden, um Instanzen dieses Programms, als "Applikations"-Objekte gehandhabt, auf einem bestimmten Zielsystem (Gerät, Modul, Steuerung) zu betreiben. Somit gibt es zwei Haupttypen von Objekten in einem Projekt:
- **Programmierobjekte (POUs):**  
Dies sind Programme, Funktionen, Funktionsblöcke, Methoden, Schnittstellen, Aktionen, Datentyp-Definitionen usw.  
Programmierobjekte, die projektweit, d.h. für alle im Projekt definierten Applikationen, instanziiert werden können sollen, müssen im POU-Fenster verwaltet werden.  
Das Instanzieren erfolgt dadurch, dass ein Programm-POU von der Task einer Applikation aufgerufen wird.  
Programmierobjekte, die nur im Geräte-Fenster verwaltet werden, d.h. direkt einer Applikation zugeordnet sind, können nicht von anderen Applikationen instanziiert werden.
- **Ressourcen-Objekte (Gerätebaum):**  
Dies sind Geräte-Objekte, Applikationen, Taskkonfigurationen, Rezepturverwalter etc., und werden NUR im Geräte-Fenster, also im "Gerätebaum", verwaltet. Für das Anlegen der Geräteobjekte im Gerätebaum gelten bestimmte Regeln, nach denen die zu steuernde Hardware-Umgebung abgebildet werden muss. Durch die hierarchische Anordnung von Applikations- und Geräte-Objekten werden die Gültigkeitsbereiche von Objekten wie Bibliotheken und GVLs etc. definiert.

- Codegenerierung durch integrierte Compiler und Verwendung von Maschinen-Code ermöglicht kurze Ausführungszeiten.
- Datentransfer zum Steuerungsgerät: Der Datentransfer zwischen CODESYS und der Steuerung erfolgt über einen Gateway (Komponente) und ein Laufzeitsystem. Eine komplette Online-Funktionalität für das Überwachen des Programms auf der Steuerung ist verfügbar.
- Vordefinierte Feature-Sets dienen dazu, um zwischen einer "Standard" Benutzeroberfläche mit einem reduzierten Funktionalitätsumfang und weniger Komplexität und einer "Professional" Umgebung, die alle Funktionalitäten umfaßt, wählen zu können. Beim ersten Öffnen des Programmiersystems nach der Erstinstallation wird bereits dazu aufgefordert. Aber auch danach kann nochmals umgestellt werden und auch eine benutzerdefinierte Anpassung des aktuell verwendeten Feature Sets ist möglich. Zu den Unterschieden zwischen Standard und Professional Feature-Set siehe die Beschreibung der Feature Optionen.

## 4.2 Unterstützte Programmiersprachen

Alle in der IEC-Norm IEC 61131 aufgeführten Programmiersprachen werden mittels speziell angepasster Editoren unterstützt:

- FUP/KOP/AWL-Editor für Funktionspläne (FUP), Kontaktpläne (KOP) und Anweisungslisten (AWL)
- AS-Editor für Ablaufsprache (AS)
- ST-Editor für Strukturierten Text
- Zusätzlich wird ein Editor zur Verfügung gestellt, der das Programmieren in der Sprache CFC (Continuos Function Chart) unterstützt:
- CFC-Editor für Continuous Function Chart
- CFC ist eine Erweiterung der standardmäßigen IEC-Sprachen.

## 4.3 Profile und Plug-Ins

Wenn "CODESYS without Profile" über das Startmenü gewählt wird, wird man aufgefordert, ein "Profil" zu wählen. (Wenn direkt mit "CODESYS V3" aus dem Startmenü oder über das Desktop-Symbol gestartet wird, wird automatisch das Standard-Profil verwendet.).

Ein Profil definiert den Umfang der in der Benutzeroberfläche verfügbaren Komponenten (**plug-ins**).

Jedes Plug-In ist mit einem Namen, einer Versionsnummer und Versionseinschränkungen (constraints) gekennzeichnet.

Neben den von 3S-Smart Software Solutions GmbH zur Verfügung gestellten Komponenten können auch kundenspezifische Plug-Ins integriert sein.

Beispiele für essentielle Komponenten sind 'Online Manager' oder 'Runtime System'. Diese müssen auf jeden Fall im System enthalten sein. Andere von 3S-Smart Software Solutions GmbH bereitgestellte Plug-Ins, wie verschiedenen Editoren, Code-Generatoren, Kommunikationstreiber etc. können optional verwendet werden.

Fa. elrest kann Plug-Ins für hersteller- bzw. kundenspezifische Komponenten erstellen und verwendet hierzu das original 3S Toolkit (**Automation Platform Toolkit**).

Für die Anzeige der Komponenten eines Profils und für Änderungen am Profil benötigt man den **Installation and Profile Manager** (IPM.exe). Er bietet einen Dialog zum Installieren von Plug-Ins auf dem lokalen System und einen weiteren Dialog für die Konfiguration von Profilen auf Basis der installierten Plug-Ins.

Profil-Konfigurationen werden in Dateien mit Erweiterung ".profile" gespeichert.

Beachten Sie bitte folgendes zu den Versionen von CODESYS bzw. seinen Komponenten:

- Mehrere Versionen können gleichzeitig installiert sein.
- Unterschiedliche Versionen des Laufzeitsystems und Programmiersystems können zusammen verwendet werden.
- Jeder Service-Mitarbeiter hat auf seinem Rechner alle Versionen verfügbar.
- Jedes Projekt weiß, mit welcher Version es erstellt wurde. Wenn es in einer anderen Version geöffnet wird, wird der Anwender auf mögliche bzw. nötige Aktualisierungen bzgl. Speicherformat, Bibliotheksversionen etc. hingewiesen.
- Individuelle Funktionen können hinzugefügt werden, ohne dass eine Aktualisierung aller CODESYS-Komponenten nötig wird.

## 4.4 Was ist ein Programm

Als POUs werden Programmierereinheiten (Objekte) bezeichnet, aus denen sich ein Steuerungsprogramm zusammensetzt und die zum Programm gehören.

**POU = Program Organization Unit = Programmier-Organisationseinheit.**

POUs, die im "POUs"-Fenster verwaltet werden, sind nicht gerätespezifisch, sondern projektweit verwendbar und können ggfs. für die Verwendung auf einem im Projekt beschriebenen Gerät in der gerätespezifischen Applikation "instanziiert" werden. Zu diesem Zweck werden Programm-POUs über die eine Task der jeweiligen Applikation aufgerufen.

POUs, die NUR im "Geräte"-Fenster verwaltet werden, d.h. dort im Gerätebaum direkt unter einer Applikation eingefügt sind, können nur von Applikationen verwendet werden, die unterhalb dieser Applikation eingerückt bereitstehen ("Kind"-Applikationen). Siehe weitere Informationen bei der Beschreibung des Gerätebaums und des Applikations-Objekts.

**Aber:** "POU" wird im Objekt hinzufügen-Dialog auch als Name einer bestimmten Unterkategorie dieser Objekte verwendet und bezeichnet an dieser Stelle nur Programme, Funktionsbausteine und Funktionen!

Somit ist ein Program Organization Unit Objekt generell eine Programmierereinheit, ein Objekt, das nicht-gerätespezifisch im "POUs"-Fenster oder gerätespezifisch im "Geräte"-Fenster verwaltet wird und in einem Editorfenster angezeigt und bearbeitet werden kann. Ein POU Objekt kann ein Programm, eine Funktion, ein Funktionsbaustein (Funktionsblock), oder auch eine Methode, Aktion, Interface, DUT (Data Unit Type, Datentyp) oder eine externe Datei beliebigen Formats sein.

Beachten Sie die Möglichkeit, bestimmte Eigenschaften (wie z.B. Vorgaben für den Übersetzungsvorgang etc.) für jedes POU Objekt getrennt zu definieren.

Um ein POU Objekt im Projekt einzufügen, verwenden Sie den Befehl 'Objekt hinzufügen'.

Die folgenden Typen von **POU Objekten** können im Standardfall verwendet werden:

- Aktion
- Applikation
- DUT (Datentyp)
- Externe Datei
- Globale Variablenliste
- Bildersammlung
- „Schnittstelle“
- Bibliotheksverwalter
- Methode
- Eigenschaft
- POU
- Textliste

Neben den POU-Objekten werden Geräteobjekte für das Ausführen des Programms auf dem Zielsystem benötigt (Applikation, Taskkonfiguration etc.) Diese werden im "Geräte"-Fenster verwaltet.

## 4.5 Gerät

Im Geräte-Fenster ("**Gerätebaum**") kann die Hardware abgebildet werden, auf der die Applikation(en) laufen soll(en).

Jedes **Geräte**-Objekt repräsentiert eine bestimmte (Zielsystem) Hardware. Beispiele: Steuerungsgerät, Feldbusknoten, Buskoppler, Antrieb, E/A-Modul, Monitor.


Jedes Gerät ist durch seine **Gerätebeschreibung** definiert und muss auf dem lokalen System installiert sein, um für das Einfügen im Gerätebaum verfügbar zu sein. Die Gerätebeschreibungsdatei definiert die Eigenschaften eines Geräts hinsichtlich Konfigurierbarkeit, Programmierbarkeit und möglichen Verbindungen zu anderen Geräten.

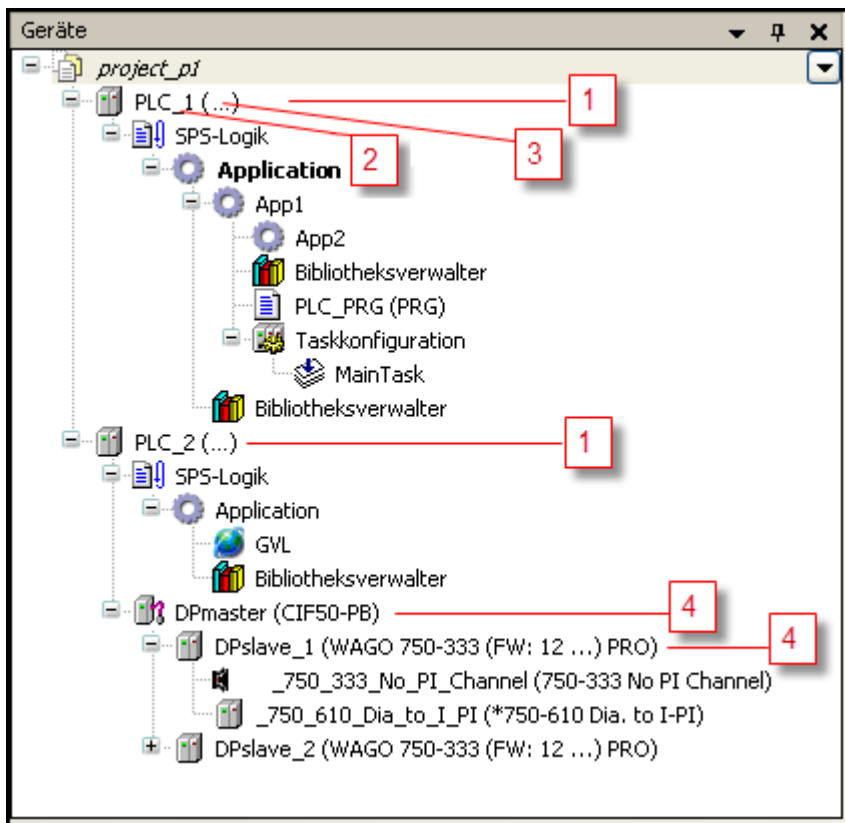
Im Gerätebaum werden jedoch nicht nur die reinen Geräte-Objekte verwaltet, sondern alle Objekte, die benötigt werden, um eine „Applikation“ (also ein Steuerungsprogramm) auf einem Gerät (Steuerung, SPS) auszuführen; also auch die 'Applikation'-Objekte, sowie "Taskkonfiguration" und "Task". Aber auch reine Programmierobjekte wie einzelne POU's, Globale Variablenlisten und Bibliotheksverwalter können hier - anstatt als projektweit instanzierbare Einheiten im POU's-Fenster - NUR im Gerätebaum verwaltet werden und sind dann eben nur für "ihre" Applikation bzw. deren "Kind-Applikationen" verwendbar.

Beachten Sie die Möglichkeit die aktive Applikation auf einem "**Simulationsgerät**" laufen zu lassen, das standardmäßig innerhalb des Programmiersystems automatisch zur Verfügung steht (zur Zeit verfügbar für CODESYS Control Win V3 Zielsystem. So wird kein echtes Zielgerät benötigt um das Online-Verhalten der Applikation zu testen. Sobald Sie in den Simulationsmodus wechseln, wird der Eintrag CODESYS Control Win V3 im Gerätebaum in Kursivdruck dargestellt; die Applikation kann eingeloggt werden.

Man beachte auch die Möglichkeit, "**Online Konfigurationsmodus**" zum Gerät zu verbinden, ohne vorher eine reelle Applikation dorthin laden zu müssen. Dies ist nützlich für die Erstinbetriebnahme eines E/A-Systems, weil die E/As in der Steuerungskonfiguration angesprochen und getestet werden können bevor ein reelles Applikationsprogramm erstellt und geladen werden wird.



Generelles:







- Der Wurzelknoten des Baums ist immer ein symbolischer Knoteneintrag:  <Projektname>.
- Die Konfigurationsbäume der "Steuerungskonfiguration" und "Taskkonfiguration", die in CODESYS in separaten Fenstern gehandhabt wurden, sind im Gerätebaum integriert. Nur die Konfiguration der diversen Geräte- bzw. Task-Parameter wird in separaten Editor-Dialogen durchgeführt. Siehe hierzu: „Taskkonfiguration“, ↘ „Steuerungskonfiguration“  
So wird die Struktur der zu steuernden Hardware-Umgebung im Gerätebaum durch die entsprechende Anordnung von Objekten abgebildet, und es ist möglich, ein heterogenes, hierarchisches System von mehrfach vernetzten Steuerungen und zugrundeliegenden Feldbussen aufzusetzen.
- Sehen Sie unten die Regeln für das Anordnen von Objekten im Gerätebaum.

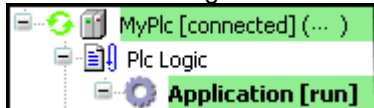


- 1 programmierbares Gerät (mit Applikationen)
- 2 symbolischer Geräteiname
- 3 Geräteiname, definiert in der Gerätebeschreibung
- 4 rein parametrierbares Gerät

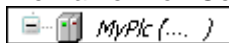
Abb. 33: Gerätebaum, Beispiel

- Ein Geräte-Eintrag im Gerätebaum zeigt jeweils das Symbol, den im Baum editierbaren symbolischen Gerätenamen und dahinter den Gerätetyp (=Gerätename, wie in der Gerätebeschreibung definiert)
- Es gibt programmierbare Geräte und nur parametrierbare Geräte. Der Gerätetyp bestimmt die mögliche Einfügeposition im Gerätebaum und die Auswahl der Objekte, die unterhalb des Geräts eingefügt werden können. Programmierbare Geräte erhalten automatisch einen zusätzlichen "SPS Logik" Knoten unterhalb des Geräte-Objekts. Unter diesem Knoten können dann die Objekte, die für das Programmieren des Geräts benötigt werden, eingefügt werden (Applikationen, GVLs, Textlisten etc.).
- Innerhalb eines Projekts können ein oder mehrere programmierbare Geräte - unabhängig von Hersteller oder Typ - konfiguriert werden ("Multi-Ressource", "Multi-Device", Netzwerke).
- Die Konfiguration eines Geräts bezüglich Kommunikation, Parameter, I/O-Abbild wird im Geräte-Editor Dialog durchgeführt, der mit einem Doppelklick auf den Geräte-Eintrag geöffnet wird (für eine Beschreibung siehe „Geräte-Editor“).
- Im Online-Modus zeigt ein Symbol vor einem Geräte-Eintrag den Status des Gerätes an:
- : SPS ist verbunden, Applikation läuft, Gerät ist in Betrieb, Daten werden ausgetauscht. Die SPS-Einstellung "E/As aktualisieren im Stop:" kann aktiviert oder deaktiviert sein.
- : SPS ist verbunden und im Stop (STOP), und SPS-Einstellung "E/As aktualisieren im Stop" ist deaktiviert.

- : SPS ist verbunden, die aktive Applikation läuft, Diagnoseinformation ist verfügbar.
- : Gerät tauscht keine Daten aus, Busfehler, keine Konfiguration, oder Simulationsmodus.
- : Gerät läuft im Demo-Modus für 30 Minuten. Nach dieser Zeit läuft der Demo-Modus aus und der Feldbus beendet den Datenaustausch.
- : Gerät ist konfiguriert, aber nicht voll betriebsbereit. Es werden keine Daten ausgetauscht. Beispielsweise CANopen-Geräte während des Hochstartens und im präoperativen Zustand.
- : Redundanz-Modus aktiv. Der Feldbus-Master sendet gerade keine Daten, weil ein anderer Master aktiv ist.
- : Die Gerätebeschreibung konnte im Geräte-Repository nicht gefunden werden.
- Die Namen aller gerade verbundenen Geräte und Applikationen werden grün hinterlegt:




- Die Namen von Geräten, die im Simulationsmodus laufen, werden in Kursivschrift dargestellt:



### Installieren von Geräten auf dem lokalen System

- Installation und De-Installation von Geräten kann im „Geräte-Repository...“ Dialog durchgeführt werden. Die Installation basiert auf Gerätebeschreibungdateien CODESYS \*.devdesc.xml. Es können aber auch busspezifische Beschreibungdateien wie \*.gsd-Dateien (Profibus), \*.eds-Dateien (CANbus, ASI) etc., über den Geräte-Repository Dialog installiert werden. Die spezifischen CODESYS V3 Konfigurationsdateien \*.cfg können verwendet werden, wenn eine entsprechende Info-Datei \*.info.cfg vorliegt, die das jeweilige Set an Konfigurationsdateien beschreibt.

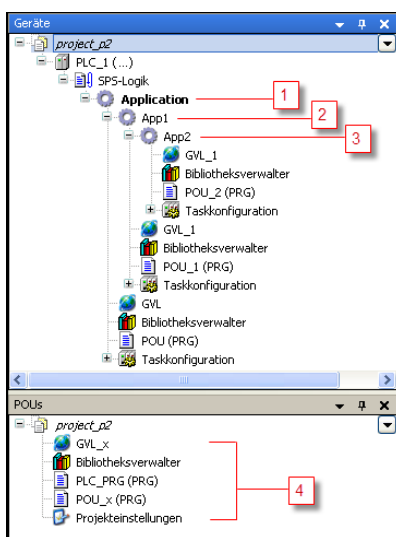
### Anordnen und Konfigurieren von Objekten im Gerätebaum, Vorgehen und Regeln:

- Um ein Objekt einzufügen, verwenden Sie den Befehl „Objekt hinzufügen“, um ein "Gerät" einzufügen am besten gleich den Befehl ↘ Gerät einfügen (bzw. anhängen). Welche Objekttypen eingefügt werden können, hängt von der gerade ausgewählten Position im Baum ab. Beispiel: Module für einen DP Profibus Slave können nicht eingefügt werden, ohne das entsprechende Slave-Objekt vorher eingefügt zu haben; Applikationen können nicht unterhalb von nicht-programmierbaren Geräten eingefügt werden. Weiterhin stehen immer nur solche Geräte zum Einfügen zur Auswahl, die korrekt auf dem lokalen System installiert wurden und zur gerade ausgewählten Position im Baum passen.
- Eine Veränderung der Anordnung ist möglich über die Standardbefehle der Kategorie ↘ Zwischenablage (Ausschneiden, Kopieren, Einfügen, Löschen) oder durch Ziehen des ausgewählten Objekts mit der Maus bei gedrückter Maustaste (für Kopieren plus [ Strg ] ). Beachten Sie zum ↘ „Einfügen“-Befehl: Falls das ausgeschnittene Objekt unterhalb oder oberhalb des gerade selektierten Objekts eingefügt werden könnte, öffnet der 'Einfügeposition auswählen'-Dialog, wo sie entsprechend entscheiden. Wenn Objekte mit der 'Kopieren und Einfügen'-Funktion hinzugefügt werden erhält das neue Objekt den gleichen Namen gefolgt von einer hochzählenden Nummer.
- Ein bereits im Gerätebaum eingefügtes Gerät kann durch eine andere Version desselben Geräts oder durch ein Gerät anderen Typs ersetzt werden (Geräte-Update). Ein eingerückt unterhalb des Geräts vorhandener Konfigurationsbaum wird dabei soweit als möglich erhalten.
- Auf der Ebene direkt unterhalb des Wurzelknotens  <Projektname> können nur "Gerät"-Objekte eingehängt werden. Wenn ein anderer Objekttyp im 'Objekt hinzufügen'-Dialog ausgewählt wird, wie z.B. "Textliste", wird dieses automatisch im POU-Fenster eingefügt! Wenn gerade kein Eintrag selektiert ist, z.B. wenn Sie mit der Maus in den leeren Bereich des Geräte-Fensters klicken, ist dies gleichbedeutend mit dem Selektieren des Wurzelknotens.
- Ein Gerät wird als Knoten im Baum eingefügt. Wenn solche in der Gerätebeschreibung definiert sind,



werden automatisch die Unterknoten mit eingefügt. Ein Unterknoten kann auch wiederum ein programmierbares Gerät sein.

- Unterhalb eines "Gerät"-Objekts können weitere Geräte eingefügt werden, die auf dem lokalen System installiert sind und im 'Objekt hinzufügen'- bzw. 'Gerät einfügen'-Dialog angeboten werden. Die Reihenfolge der Geräteobjekte innerhalb des Baums von oben nach unten: Pro Ebene werden immer zuerst die programmierbaren Geräte (SPS Logik) angeordnet, dann die restlichen Typen, jeweils alphabetisch sortiert.
- Eine „Applikation“ kann nur unterhalb eines "SPS Logik- Knotens (symbolischer Knoten für programmierbare Geräte) eingefügt werden. Pro Gerät müssen alle Applikationen eindeutig benannt sein. Unterhalb jeder Applikation können dann die weiteren für die Programmierung benötigten Objekte eingehängt werden, wie z.B. Taskkonfiguration, DUT, GVL, Visualisierungen. Eine Taskkonfiguration muss unterhalb jeder Applikation eingefügt werden und darin die entsprechenden Programmaufrufe (POU-"Instanzen" aus dem Geräte-Fenster bzw. applikationsspezifische POU's) definiert werden. Wenn mehrere Applikationen direkt unterhalb eines Geräts hängen, ist beim „I/O Abbild“ des Geräts festzulegen, die Variablen welcher dieser Applikationen aktuell für die Kommunikation mit dem Zielsystem verwendet werden sollen.
- Hierarchische Anordnung von Applikationen, Gültigkeitsbereiche:  
Eine Applikation kann auch unterhalb einer anderen Applikation eingehängt werden: Nennen wir das entstehende Konstrukt "Vaterapplikation" - "Kindapplikation". In diesem Fall gilt folgendes: Die Kindapplikation kann auf die Objekte der Vaterapplikation zugreifen, nicht aber umgekehrt! Hintergrund: Eine Kindapplikation soll immer entfernt bzw. ausgetauscht werden können, ohne die Vaterapplikation zu beeinträchtigen. Sehen Sie dazu folgende Abbildung: 'Application' ist Vaterapplikation für 'App1', diese wiederum für 'App2':



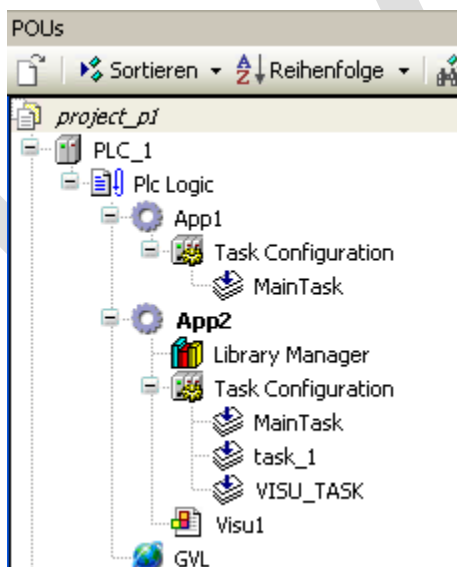
- 1 kein Zugriff auf App1, App2 und deren POU's, globale Variablen und Bibliotheken
- 2 Zugriff auf Application und deren GVLs, POU's und Bibliotheken
- 3 Zugriff auf Application und App1 und deren GVLs, POU's und Bibliotheken
- 4 kann von allen Applikationen instanziiert werden

### Netzwerk-Scan der aktuellen Hardware

Als Erleichterung für das Anlegen der Steuerungskonfiguration im Gerätebaum unterstützen die Geräte-Editoren im Standardfall die Möglichkeit, die aktuelle Hardware abzutasten und in einem Dialog darzustellen. Der Anwender kann die gewünschten Module von dort direkt in den Gerätebaum übernehmen. Siehe dazu Befehl „Geräte

## 4.6 Applikation

- Eine "Applikation" umfasst mehrere Objekte, die für das Ausführen einer bestimmten Instanz des Steuerungsprogramms auf einem bestimmten Gerät (Steuerung, SPS) nötig sind. Dazu können u.a. "unabhängige" Objekte, die im POU-Fenster verwaltet werden, instanziiert und einem Gerät im Gerätebaum zugewiesen werden. Dies entspricht der Denkweise der objektorientierten Programmierung. Aber auch rein applikationsspezifische POU's können von der Applikation verwendet werden.
- Eine Applikation wird durch ein "Applikation"-Objekt im Gerätebaum unterhalb eines SPS-Logik Knotens (d.h. programmierbares Gerät) eingefügt werden kann. Unterhalb einer Applikation können die Objekte, die das "Ressourcen-Set" der Applikation ausmachen, eingefügt werden.
- Alle Applikationen unterhalb desselben Geräts müssen eindeutig benannt sein.
- Vor dem Einfügen eines Applikationsobjekts über den Befehl Objekt hinzufügen muss das entsprechende programmierbare Gerät im Gerätebaum ausgewählt werden, ansonsten steht der Objekttyp 'Applikation' im 'Objekt hinzufügen'-Dialog nicht zur Auswahl. Vergeben Sie in diesem Dialog dann einen Namen für die neue Applikation und bestätigen Sie mit [ Open ]. Das Applikationsobjekt wird daraufhin im Gerätefenster unterhalb des Geräte-Objekts eingefügt.
- Ein essentielles Objekt für jede Applikation ist die „Taskkonfiguration“ zur Kontrolle der Ausführung eines Programms (POU-Instanzen oder applikationsspezifische POU's). Zusätzlich können der Applikation Ressourcen-Objekte wie Globale Variablenlisten, Bibliotheken etc. direkt im Geräte-Fenster zugewiesen sein, die dann im Gegensatz zu den im POU's-Fenster verwalteten nur von dieser Applikation und ihren "Kindern" verwendet werden können, siehe „Anordnen und Konfigurieren von Objekten im Gerätebaum, Vorgehen und Regeln.“ zu den Regeln.
- Innerhalb eines Projekts können mehrere Applikationen für dasselbe Gerät definiert sein, d.h. sie können auf der selben Steuerung laufen ("multi-application"). Dabei ist eine hierarchische Anordnung möglich ("Vater"- und "Kind"-Applikationen). In diesem Fall kann ein "Kind" die dem "Vater" zugewiesenen Objekte verwenden, nicht aber umgekehrt, siehe „Anordnen und Konfigurieren von Objekten im Gerätebaum, Vorgehen und Regeln.“ zu den Regeln.
- Wenn mehrere Applikationen direkt unter einem Gerät hängen, kann für das I/O-Mapping dieses Geräts festgelegt werden, welche der Applikationen berücksichtigt werden soll.



- Beispiel: Zwei Applikationen für ein Gerät
- Beachten Sie, dass die Applikation, mit der Sie im Online-Betrieb arbeiten wollen, als aktive Applikation

gesetzt sein muss; siehe "App2" in der oben gezeigten Abbildung.

- Eine Applikation kann im Objekteigenschaften-Dialog („Optionen Applikationserzeugung“) mit speziellen Eigenschaften versehen werden. Beispielsweise kann dynamische Speicherallozierung aktiviert werden. Beachten Sie auch die Möglichkeit, Information über den Applikationsinhalt mit auf die SPS zu laden. Dies ermöglicht es, die Applikation auf der Steuerung mit der gerade im Programmiersystem aktiv zu vergleichen.
- Wenn Sie dabei sind, mit einer Applikation auf das Zielgerät einzuloggen (SPS oder Simulation), wird geprüft, welche Applikationen aktuell auf der SPS liegen und ob die Applikationsparameter auf der Steuerung mit denen in der Projektkonfiguration übereinstimmen. Entsprechende Meldungen werden Nichtübereinstimmungen anzeigen und mögliche Wege, wie in dem Fall weiterverfahren werden kann; auch werden Sie die Möglichkeit erhalten, Applikationen auf der SPS zu löschen. Für weitere Details sehen Sie bitte die Beschreibung des Befehls „Einloggen“.
- Beachten Sie den Unterdialog 'Applikation' des Geräte-Editors („Applikationen“), in dem Sie sehen, welche Applikationen gerade auf einem Gerät vorhanden sind, und in dem man diese vom Zielsystem löschen kann. Hier können auch Applikationen angezeigt werden, die nicht durch ein separates Objekt im Gerätebaum vertreten sind, wie z.B. die "<Applikation>\_\_symbols.app", die eine ggfs. für die Applikation <Applikation> erzeugte (siehe „Symbolkonfiguration“, die ggfs. für die Applikation <Applikation> erzeugt wurde).

## 4.7 Taskkonfiguration

Die Taskkonfiguration definiert eine oder mehrere Tasks zur Kontrolle und Ausführung des Applikationsprogramms auf der Steuerung.

Sie ist ein essentielles Ressourcen-Objekt für eine Applikation und muss im Gerätebaum unterhalb einer solchen eingefügt sein. Eine Task kann sowohl ein applikationszugeordnetes Programm aufrufen (POU, die im Projekt nur im Gerätebaum unterhalb der Applikation vorliegt), oder ein Programm, das im POU-Fenster verwaltet wird.

Im letzteren Fall ist dies eine "Instanzierung" einer projektweit verfügbaren POU.

Eine Taskkonfiguration kann im Task Editor, bearbeitet werden, wobei die verfügbaren Optionen vom Zielsystem abhängen.

Im Online-Betrieb bietet der Task Editor eine Monitoring-Ansicht mit Informationen über Zyklen, Zykluszeiten und Task-Status.

Als zusätzliche Funktionalität der Taskkonfiguration, wenn es vom Gerät unterstützt wird, ermöglicht die "Überwachung (Profiling)" eine dynamische Analyse der Bausteine, die von einer Task kontrolliert werden. Es gibt Auskunft über die Ausführungszeiten, die Anzahl der Bausteinaufrufe und die nicht durchlaufenen Code-Zeilen.

### **Wichtige Hinweise für Multitasking-Systeme:**

Auf manchen System ist echtes preemptives Multitasking realisiert (z.B. CODESYS). In diesem Falle ist folgendes zu beachten:

Wie in CODESYS V3 teilen sich alle Tasks ein Prozessabbild. Grund: Ein eigenes Prozessabbild pro Task würde zu Lasten der Performance gehen. Das Prozessabbild kann jedoch somit immer nur zu einer Task konsistent sein. Beim Erstellen eines Projekts muss man deshalb als Anwender selbst dafür sorgen, dass bei Konflikten die Input-Daten in einen sicheren Bereich kopiert werden, bei Ausgängen gilt dasselbe. Möglichkeiten, die Konsistenz- und Synchronisierungsprobleme zu lösen, bieten beispielsweise die Bausteine der SysSem-Bibliothek.

Auch beim Zugriff auf andere globale Objekte (globale Variablen, Bausteine) können in Multitasking-Systemen Konsistenzprobleme auftreten, wenn die Objekte die Datenbreite des Prozessors überschreiten (Strukturen oder Arrays, die eine logische Einheit bilden). Zur Lösung stehen hier wiederum z.B. die Bausteine der SysSem-Bibliothek zur Verfügung.

in Arbeit

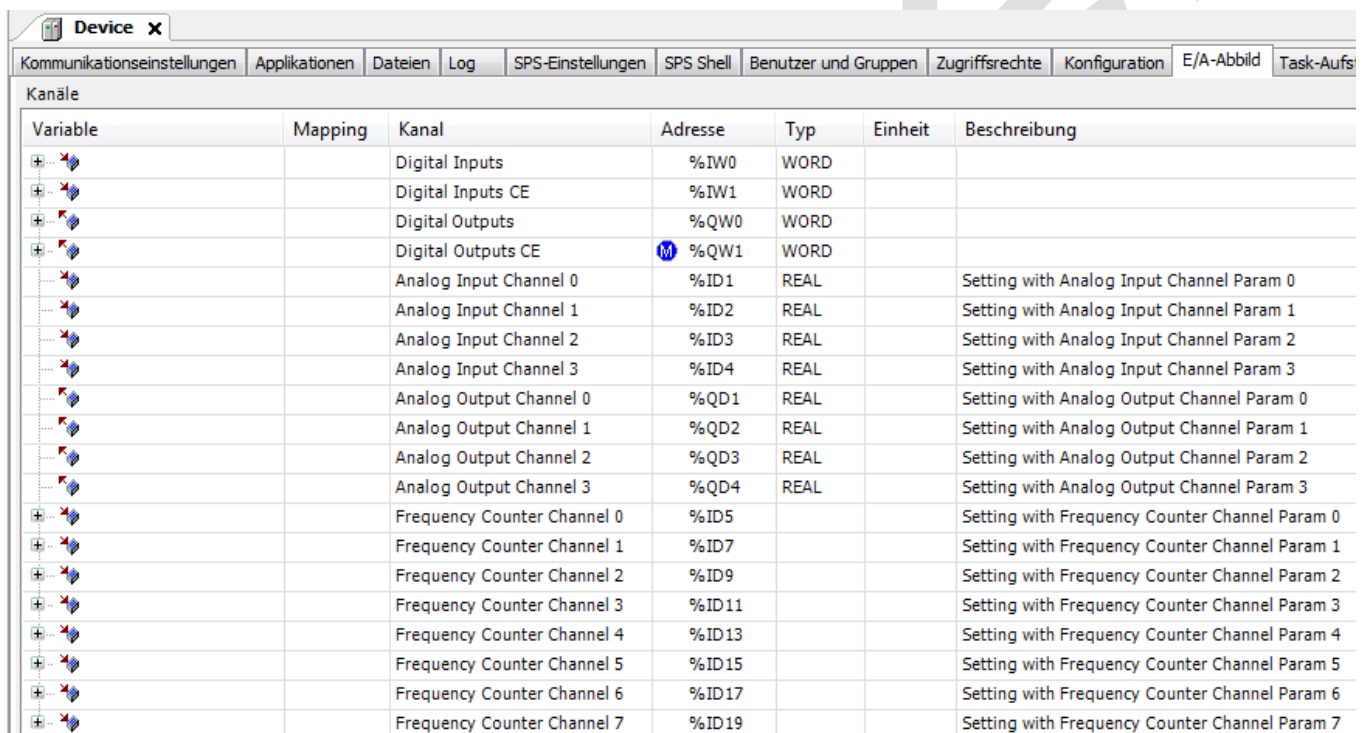
## 4.8 Kommunikation

### 4.8.1 Steuerungskonfiguration

Die "Steuerungskonfiguration" bildet die Ziel-Hardware im Programmiersystem ab, um die Ein- und Ausgänge und die Parameter der Steuerung und der Feldbusgeräte der Applikation zugänglich zu machen. Außerdem ermöglicht sie das Anzeigen der verfügbaren Geräte-Parameter.

Der Baum der "**Steuerungskonfiguration**", der in CODESYS V3 **im Gerätebaum integriert**, in dem auch die anderen Objekte arrangiert sind, die nötig sind, um eine Applikation auf einem Zielsystem laufen zu lassen. Das Abbilden der aktuellen Hardware-Konfiguration im Gerätebaum wird in den Standard-Geräte-Editoren durch eine Scan-Funktionalität erleichtert.

Die Zuweisung der Ein- und Ausgänge der Steuerung zu Projektvariablen geschieht entweder mit Hilfe der "AT Deklaration" im Deklarationseditor oder im I/O-Abbild Dialog des Geräte-Editors, der die Dialoge zur Konfiguration eines Geräts bereitstellt. Wenn unterhalb eines auf oberer Ebene des Gerätebaums positionierten Geräts mehrere Applikationen vorliegen, kann im I/O-Abbild (Mapping) Dialog dieses Geräts festgelegt werden, welche Applikation für das Mapping herangezogen werden soll. Wenn eine Steuerung neu in den Gerätebaum eingefügt wird, wird standardmäßig gleich die automatisch mit eingefügte Applikation als "Mapping-Applikation" eingetragen.



Variable	Mapping	Kanal	Adresse	Typ	Einheit	Beschreibung
		Digital Inputs	%IW0	WORD		
		Digital Inputs CE	%IW1	WORD		
		Digital Outputs	%QW0	WORD		
		Digital Outputs CE	%QW1	WORD		
		Analog Input Channel 0	%ID1	REAL		Setting with Analog Input Channel Param 0
		Analog Input Channel 1	%ID2	REAL		Setting with Analog Input Channel Param 1
		Analog Input Channel 2	%ID3	REAL		Setting with Analog Input Channel Param 2
		Analog Input Channel 3	%ID4	REAL		Setting with Analog Input Channel Param 3
		Analog Output Channel 0	%QD1	REAL		Setting with Analog Output Channel Param 0
		Analog Output Channel 1	%QD2	REAL		Setting with Analog Output Channel Param 1
		Analog Output Channel 2	%QD3	REAL		Setting with Analog Output Channel Param 2
		Analog Output Channel 3	%QD4	REAL		Setting with Analog Output Channel Param 3
		Frequency Counter Channel 0	%ID5			Setting with Frequency Counter Channel Param 0
		Frequency Counter Channel 1	%ID7			Setting with Frequency Counter Channel Param 1
		Frequency Counter Channel 2	%ID9			Setting with Frequency Counter Channel Param 2
		Frequency Counter Channel 3	%ID11			Setting with Frequency Counter Channel Param 3
		Frequency Counter Channel 4	%ID13			Setting with Frequency Counter Channel Param 4
		Frequency Counter Channel 5	%ID15			Setting with Frequency Counter Channel Param 5
		Frequency Counter Channel 6	%ID17			Setting with Frequency Counter Channel Param 6
		Frequency Counter Channel 7	%ID19			Setting with Frequency Counter Channel Param 7

Im E/A-Abbild können die hardwareseitig zu Verfügung stehende Ein- und Ausgänge konfiguriert und mit globalen Namen versehen werden.

Wir unterscheiden folgende Möglichkeiten:

#### ▲ Digitale Ein- und Ausgänge

#### ▲ Analoge Ein- und Ausgänge

Diese müssen gewöhnlich bezüglich Ihres Sensortyps konfiguriert werden.

#### ▲ Frequenz Ein- und Ausgänge

Diese müssen gewöhnlich bezüglich Ihres Sensortyps konfiguriert werden.

Als Impulszähler werden auch Frequenzeingänge verstanden.

Frequenzausgänge werden als PID-Regler Ausgänge (PWM-Mode) oder Schrittmotorenansteuerungen verwendet.

Im Dialog  
Device→Konfiguration  
können die notwendigen  
Einstellungen vorgenommen  
werden.

Parameter	Typ	Wert	Standardwert
[-] Analog Input Channel Param 1			
[-] Sensortype	Enumeration of BYTE	0V - +10Volt	
[-] NumberOfAverageElements	BYTE	8	8
[-] Analog Input Channel Param 2			
[-] Sensortype	Enumeration of BYTE	0V - +10Volt	
[-] NumberOfAverageElements	BYTE	8	8
[-] Analog Input Channel Param 3			
[-] Sensortype	Enumeration of BYTE	0V - +10Volt	
[-] NumberOfAverageElements	BYTE	8	8
[-] Analog Output Channel Param 0	Enumeration of BYTE	0V - +10V	None
[-] Analog Output Channel Param 1	Enumeration of BYTE	0V - +10V	None
[-] Analog Output Channel Param 2	Enumeration of BYTE	0V - +10V	None
[-] Analog Output Channel Param 3	Enumeration of BYTE	0V - +10V	None
[-] Frequency Counter Channel Param 0			
[-] FrequencyMode	Enumeration of BYTE	FIN_OFF	FIN_OFF
[-] BoardMode	Enumeration of BYTE	Motherboard	Motherboard
[-] Frequency Counter Channel Param 1			
[-] FrequencyMode	Enumeration of BYTE	FIN_OFF	FIN_OFF
[-] BoardMode	Enumeration of BYTE	Motherboard	Motherboard

#### 4.8.2 Adressierung und Routing

Die Adressierung bildet die Topologie des Steuerungsnetzwerks auf eindeutige Adressen ab. Eine **Knotenadresse** ist hierarchisch zusammengesetzt:

Für jede Netzwerkverbindung bestimmt der zugehörige Blocktreiber eine lokale Adresse, welche den Knoten innerhalb des lokalen Netzwerks eindeutig identifiziert. Für die gesamte Knotenadresse wird der lokalen Adresse der vom Vater vergebene Subnetz Index des lokalen Netzwerks vorangestellt. Diesem wird sodann die Knotenadresse des Vaters vorangestellt.

Dabei wird die Länge des Subnetzindices (in Bit) durch das Gerät bestimmt, wohingegen die Länge der lokalen Adresse durch den Netzwerktyp festgelegt ist.

Ein Knoten ohne Hauptnetzwerk ist ein Top-level-Knoten mit Adresse 0. Ein Knoten mit Hauptnetzwerk, welches keinen Vater enthält, ist ebenfalls ein Top-level-Knoten; er erhält die lokale Adresse des Hauptnetzwerks.

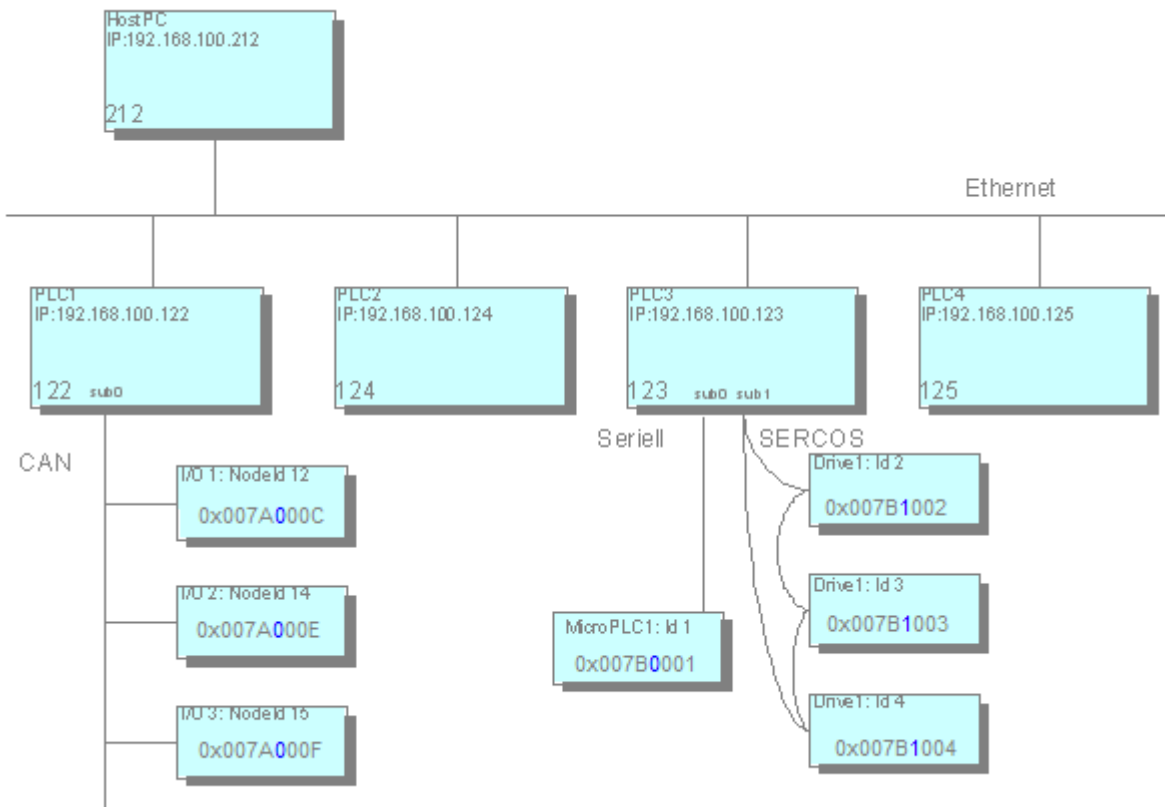


Abb. 36: Beispiel - Hauptnetzwerk und Subnetzwerke

Im Beispiel sind die Adressen der Knoten in hexadezimaler Schreibweise dargestellt. Die ersten 4 Ziffern repräsentieren die Adresse des jeweiligen Vaters im Hauptnetz, z.B. 0x007A=122 für PLC1. Das nächste Byte (in blauer Schrift) ist für den Subnetzindex reserviert und auf diesen folgt die lokale Adresse, z.B. C=12 für KnotenId 12.

Dank der Strukturierung der Adressen kann der Routing Algorithmus relativ schlank gehalten werden. So sind beispielsweise Routing Tabellen unnötig. Information wird nur lokal abgefragt: über die eigene Adresse und über die Adresse des Vaterknoten. Darauf basierend kann ein Knoten die Datenpakete korrekt bearbeiten:

Falls die Zieladresse mit der Adresse des aktuellen Knoten übereinstimmt, so ist dieser als Empfänger bestimmt.

Falls die Zieladresse mit der Adresse des aktuellen Knotens beginnt, so ist das Datenpaket entweder direkt für ein Kind oder für einen Nachkömmling des Knoten bestimmt und muss weitergeleitet werden.

Andernfalls ist der Empfänger kein Nachkömmling des aktuellen Knoten und das Datenpaket muss an seinen eigenen Vater weitergeleitet werden.

Relative Adressierung ist eine Besonderheit: Relative Adressen enthalten nicht die Knotennummer des Empfängers, sondern beschreiben direkt den Pfad vom Sender zum Empfänger. Das Prinzip ist ähnlich zum relativen Pfad im Dateisystem: Die Adresse besteht aus der Anzahl der Schritte, die das Datenpaket nach oben transportiert werden muss, d.h. zum entsprechenden Vater, und aus dem anschließenden Pfad hinab zum Zielknoten.

Der Vorteil der relativen Adressierung liegt darin, dass zwei Knoten desselben Teilbaums die Kommunikation fortsetzen können, wenn der gesamte Teilbaum an eine andere Stelle im Gesamtnetzwerk verschoben wird. Während die absolute Knotenadressierung aufgrund dieser Verschiebung geändert werden muss, ist die relative Adressierung noch immer gültig.

#### Adressbestimmung

Die Kenntnis der eigenen Adresse erfordert von einem Knoten entweder die Kenntnis über die Adresse des Vaterknoten oder über die Tatsache ein Top-level-Knoten zu sein. Zu diesem Zweck versendet der Knoten während des Hochfahrens an alle Netzwerkteilnehmer eine Nachricht zur Adressbestimmung. So lange er keine

Antwort auf diese Nachricht erhält, sieht sich der Knoten als Top-level-Knoten, fährt jedoch fort mit dem Versuch einen möglichen Vater zu entdecken. Ein Vaterknoten antwortet durch Bekanntgabe seiner Adresse. Damit wird der Knoten selbst seine Adresse vervollständigen und sie den Subnetzen bekanntgeben.

Eine Adressbestimmung kann während des Hochfahrens durchgeführt werden oder aber auf Anfrage des zur Programmierung benutzten PCs. Noch nicht implementiert: Ist die Adresse eines Knotens einmal eingefroren, so kann eine Adressbestimmung nicht länger durchgeführt werden.

### 4.8.3 Data Server

Ein "Data Server" kann einer Applikation hinzugefügt werden, um entfernte Datenquellen nutzen zu können. "Entfernte Datenquelle" in diesem Zusammenhang bedeutet, dass Variablen ("data items"), die in anderen Geräten definiert und verwendet werden, auch in der lokalen Applikation genutzt werden können.

Im Gegensatz zum Datenaustausch über Netzwerkvariablen (broadcasting) baut der Data Server Punkt-zu-Punkt-Verbindungen auf. In Abhängigkeit vom Zugriffs-Flag der auszutauschenden Datenverbindungen werden diese in der Datenquelle bzw. in der aktuellen Applikation jedesmal aktualisiert, wenn sich ihr Wert auf der jeweils anderen Seite ändert.

Die Benutzung eines Data Servers ist die schnellere Alternative zur Bereitstellung von Daten mittels einer Symbolkonfiguration. Derzeit kann über den Data Server noch nicht auf Daten zugegriffen werden, die von einem OPC-Server bereitgestellt werden. In diesem Fall ist die Implementation einer Symbolkonfiguration noch die geeignete Vorgehensweise.

### 4.8.4 Netzwerkvariablen

Netzwerkvariablen müssen in starren Variablenlisten sowohl im Sender als auch Empfänger (in einem oder mehreren Projekten) definiert werden. Ihre Werte werden via **Broadcasting** im Netz versendet. Beachten Sie, dass sich dies grundsätzlich vom Datenaustausch mittels eines „Data Server“ unterscheidet, der definierte Punkt-zu-Punkt-Verbindungen zwischen lokaler Applikation und entfernten Datenquellen verwendet.



Datenaustausch über Netzwerkvariablen ist nicht möglich, wenn es nicht vom Gerät (Zielsystem) unterstützt wird, wenn eine **Firewall** die Kommunikation blockiert, oder wenn ein anderer Client/andere Applikation denselben **UDP Port** benutzt, der in den Eigenschaften der Netzwerkvariablenliste eingestellt ist.



Nur **Arrays** deren Grenzen mit einem Literal oder einer Konstante definiert sind, werden zur Empfänger-Applikation übertragen! Konstante Ausdrücke sind für diesen Zweck nicht erlaubt! Beispiel: "arrVar : ARRAY[0..g\_iArraySize-1] OF INT ;" wird nicht übertragen: "arrVar : ARRAY[0..10] OF INT ;" wird übertragen.

Netzwerkvariablen werden gehandhabt in....

Globale Variablenliste - GVL (**GVL**) in einem sendenden Gerät (**Sender**) und in einer entsprechenden ...

Globalen Netzwerkvariablenliste (**GNVL**) in einem empfangenden Gerät (**Receiver**). GNVLs werden im „Netzwerkvariablenlisten Editor“ angezeigt.

Die zueinander gehörigen GVL und GNVLs müssen dieselben Variablendeklarationen enthalten.



Die maximal erlaubte Größe einer Netzwerkvariable ist 255 Bytes. Die mögliche Anzahl von Netzwerkvariablen ist unbegrenzt

Eine Globale Variablenliste (GVL), die Netzwerkvariablen definieren soll, muss spezielle Eigenschaften haben. Dies sind Protokoll- und Übertragungsparameter, gemäß denen die Variablenwerte innerhalb des Netzwerks versendet



werden und von allen Geräten mit einer entsprechenden GNVL empfangen werden können.



Falls die austauschenden Geräte als Sender UND Empfänger fungieren sollen, müssen die Variablenlistenkennungen der Variablenlisten (definiert im Eigenschaften-Dialog einer GVL, siehe „Netzwerkvariablen“) eindeutig sein, um fehlerhaftes Verhalten zu vermeiden!

Beachten Sie, dass die Übertragung von Netzwerkvariablen immer **nur in eine Richtung** erfolgt, vom Sender (GVL) zum Empfänger (GNVL)! Allerdings kann dadurch dass jedes Gerät sowohl GVLs als auch GNVLs handhaben kann, erreicht werden, dass ein Gerät sowohl als Sender als auch als Empfänger agiert. Eine Sender GVL kann innerhalb desselben oder von einem anderen Projekt bereitgestellt werden. Beim Anlegen einer GNVL kann die Sender GVL entweder aus der Auswahlliste der verfügbaren GVLs innerhalb des Netzwerks genommen werden oder sie kann aus einer **Exportdatei** eingelesen werden, die vorher aus der GVL erzeugt wurde (beispielsweise über den „Verknüpfung mit Datei“ Dialog der GVL). Eine solche Exportdatei ist auf jeden Fall nötig, wenn die Sender GVL in einem anderen Projekt definiert ist.

Als Voraussetzung für den Austausch von Netzwerkvariablen müssen die entsprechenden **Netzwerkbibliotheken** installiert sein. Dies erfolgt für die Standard-Netzwerkfunktionalitäten, wie z.B. für UDP, automatisch sobald die Netzwerkeigenschaften einer GVL gesetzt werden.

## 4.9 Code-Generierung und Online Change

Maschinen-Code wird erst generiert, wenn die „Applikation“ auf das Zielgerät (SPS oder Simulationsgerät) geladen wird. Bei jedem Download wird das Übersetzungsprotokoll, welches Code und Identifikation der geladenen Applikation enthält, als Datei "**<projectname>.<devicename>.<application ID>.compileinfo**" im gleichen Verzeichnis wie das Projekt gespeichert. Das Übersetzungsprotokoll wird gelöscht bei Ausführung des Kommandos „Bereinigen“ bzw. „Alles bereinigen“.

Beachten Sie, dass kein Maschinen-Code erzeugt wird, wenn das Projekt mit den Befehlen der Kategorie Erstellen übersetzt wird. Dieser Übersetzungsprozess dient dazu, das Projekt auf Programmierfehler zu überprüfen. Diese werden im Meldungsfenster (Meldungskategorie 'Übersetzen') ausgegeben. Während der Code-Erzeugung können weitere, spezielle Fehler ausgegeben werden, die nur vom Codegenerator oder erst bei der Speichervergabe festgestellt werden.



Online Change verändert das laufende Applikationsprogramm und bewirkt keinen Neustart.

Stellen Sie sicher, dass der neue Applikations-Code dennoch das gewünschte Verhalten des gesteuerten Systems bewirkt.

Abhängig von der gesteuerten Anlage können Schäden an Anlage und Werkstücken entstehen oder Gesundheit und Leben von Personen gefährdet sein.



1. Wenn ein Online Change durchgeführt wird, verhält sich der Programm-Code eventuell anders als nach einer Komplettinitialisierung, weil die Maschine ihren Status beibehält.

2. Zeigervariablen (Pointer) behalten ihren Wert aus dem letzten Zyklus. Wenn ein Zeiger auf eine Variable zeigt, die durch den Online Change ihre Größe verändert hat, wird der Wert nicht mehr korrekt geliefert. Stellen Sie sicher, dass Zeigervariablen in jedem Zyklus erneut zugewiesen werden

Wenn die Applikation, die gerade auf der Steuerung läuft, seit dem letzten Download im Programmiersystem verändert wurde, werden beim Online Change **nur die veränderten Objekte** des Projekts auf die Steuerung geladen, während das Programm dort weiter läuft.

Möglichkeit 1, einen Online Change durchzuführen

Sobald Sie versuchen, mit einem veränderten (wird mit Hilfe der Compile-Info geprüft, die im Projektverzeichnis

während des letzten Downloads abgelegt wurde) Applikationsprogramm erneut einzuloggen, werden Sie gefragt, ob ein Online Change, ein Download oder ein Einloggen ohne Änderungen durchgeführt werden soll:

▲Abb. 51: Login-Dialog

„Mit Online Change einloggen“: Diese Option ist standardmäßig aktiviert. Wenn also der Dialog mit [ OK ] verlassen wird, so werden die geänderten Objekte geladen und unmittelbar in der Online-Ansicht (Monitoring) des jeweiligen Objekts angezeigt.

„Mit Download einloggen“: Diese Option muss aktiviert sein, wenn das Projekt komplett geladen und initialisiert werden soll.

„Ohne Änderung einloggen“: Diese Option muss aktiviert sein, wenn das Programm auf der Steuerung unverändert weiterlaufen und die neuen Änderungen nicht geladen werden sollen. Danach kann explizit ein Download („Laden“) durchgeführt werden, der das gesamte Projekt neu lädt, oder beim nächsten Einloggen wird wieder gefragt, ob ein Online Change durchgeführt werden soll.

Die Schaltfläche „Details“ im Login-Dialog führt zu Informationen (Projektname, Letzte Änderung, IDE Version, Autor, Beschreibung) über die aktuelle Applikation im IDE (integrated development environment = Programmiersystem) im Vergleich zu derjenigen, die gerade auf der Steuerung (SPS) liegt:

▲Abb. 52: Applikationsinformation, Dialog

Wenn gerade andere Applikationen auf der SPS liegen, erscheinen weitere Dialogboxen zur Handhabung der Situation. Eine Beschreibung hierzu findet sich in: „Einloggen“.

Wenn der Online Change schwerwiegende Änderungen im Download-Code bewirkt, wie beispielsweise durch Verschiebungen von Zeigeradressen, oder nötige Neuzuweisungen von Schnittstellenreferenzen, erscheint eine andere Meldungsbbox nach dem Bestätigen des Online-Change-Dialogs mit „OK“ vor dem Download. Er informiert über die Effekte, die berücksichtigt werden müssen und ermöglicht die Online Change Operation abzubrechen:

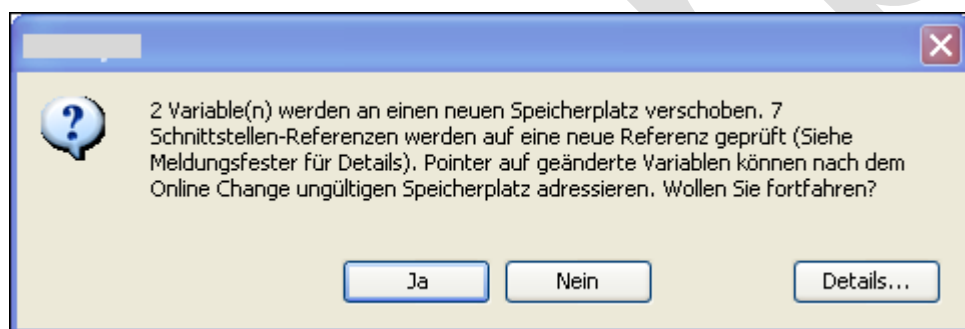


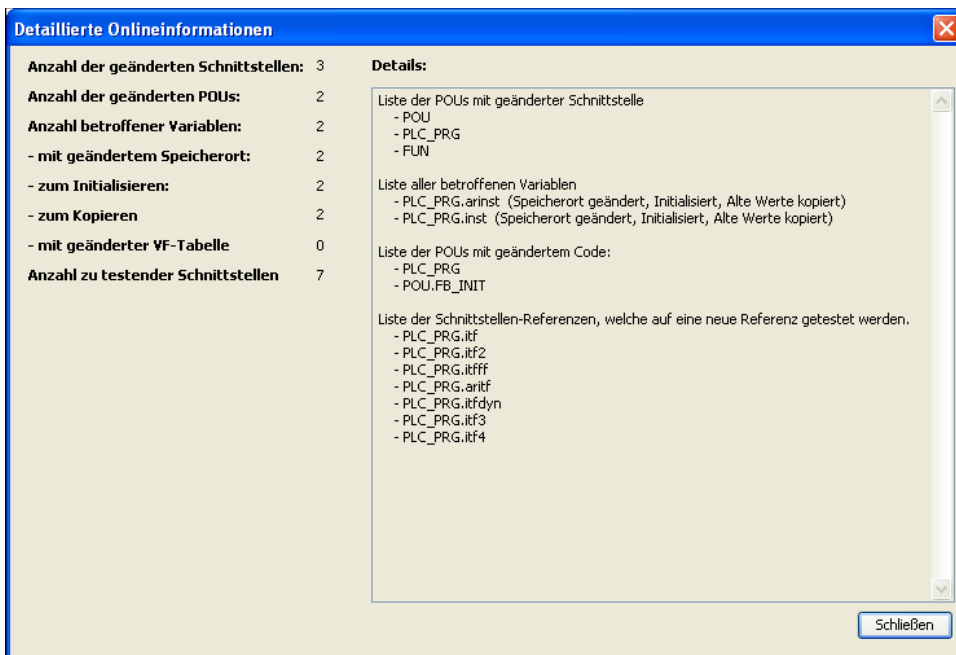
Abb. 53: Online Change Meldungsdialo im Falle von schwerwiegenden Änderungen



Ab V3.5 SP1 ist nach dem Entfernen impliziter Überwachungsfunktionen, wie beispielsweise CheckBounds, aus dem Projekt kein Online Change mehr möglich, nur ein Download.

Eine entsprechende Meldung wird ausgegeben.

- Über die Schaltfläche „Details“ im Meldungsdialo kann detaillierte Information aufgerufen werden, z.B. die Anzahl und eine Auflistung der geänderten Schnittstellen, betroffenen Variablen etc.



- Abb. 54: Detaillierte Online Change Informationen
- Möglichkeit 2, einen Online Change durchzuführen
- Mit dem Befehl „Online Change“ können Sie einen Online Change auf einer laufenden Applikation explizit herbeiführen.
- Beachten Sie, dass Online Change eines geänderten Projekts für eine Applikation nicht mehr möglich ist, nachdem ein Bereinigen durchgeführt wurde („Alles bereinigen“, „Bereinigen“). In diesem Fall wird die Information über die seit dem letzten Download geänderten Objekte gelöscht. Somit kann nur noch das gesamte Projekt neu geladen werden.

## 4.10 Boot application (boot project)

Aktuell: Bei jedem erfolgreichen Download wird die aktive Applikation automatisch in einer Datei "application.app" im Zielsystemverzeichnis gespeichert und ist somit als Bootapplikation verfügbar. Befehl "Bootapplikation erzeugen" erlaubt auch im Offline-Betrieb ein Speichern der Bootapplikation in eine Datei.

Eine Bootapplikation ist das Projekt, das automatisch gestartet wird, wenn die Steuerung gestartet wird. Dazu muss das Projekt auf der SPS in einer Datei <Projektname>.app vorhanden sein. Diese Datei kann "offline" oder "online" mit Befehl „Bootapplikation erzeugen“ (Befehlskategorie 'Online') erzeugt werden.

## 4.11 SPS-Operationssteuerung über Systemvariablen

Wenn der gesteuerte Prozess auf der Maschine in einen kritischen Zustand gelangt, sollte die Applikation in der Lage sein, gefährliche Befehle die vom Programmiersystem aus gegeben werden, zu unterdrücken; beispielsweise: Online Change, Haltepunkte setzen, Schreiben und Forcen von Variablen. Zu diesem Zweck bietet die Component Manager Bibliothek cm.library ab V3.4.3.0 einen Funktionsbaustein "PlcOperationControl" mit entsprechenden Property-Variablen.



Es hängt vom Gerät ab, ob die Verwendung der PlcOperationControl Systemvariablen unterstützt wird!

Wenn eine Instanz des Funktionsbausteins in der Applikation deklariert ist, können bestimmte Operationen auf der SPS deaktiviert bzw. wieder aktiviert werden, indem einer oder allen der folgenden Properties der Wert TRUE bzw. FALSE zugewiesen wird. Standardwert ist FALSE, TRUE deaktiviert den jeweiligen Laufzeitsystemdienst:

- xDisableApplicationOnlineChange: TRUE unterbindet einen Online Change
- xDisableApplicationDownload: TRUE unterbindet einen Download
- xDisableApplicationStop: TRUE unterbindet einen Applikationsstopp
- xDisableApplicationBP: TRUE unterbindet das Setzen eines Haltepunkts
- xDisableApplicationWrite: TRUE unterbindet das Schreiben von Variablen (wird auch über PLCHandler/IecVarAccess unterbunden!)
- xDisableApplicationForce: TRUE unterbindet das Forcen von Variablen
- xDisableApplicationReset: TRUE unterbindet einen Reset der Applikation (nicht "Reset Ursprung!")
- xDisableAll: TRUE unterbindet alle durch die oben genannten Property-Variablen betroffenen Operationen



ST

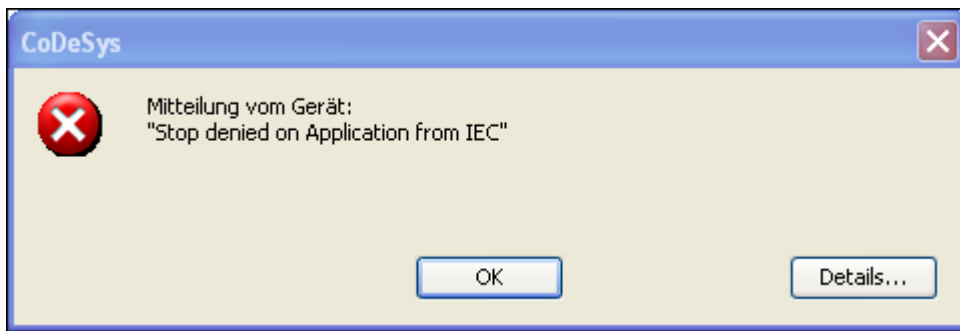
(\* Deklaration \*)

```
PlcOpCtrl_Inst : PlcOperationControl;
```

(\* Zugriff auf die Property \*)

```
PlcOpCtrl_Inst.xDisableApplicationStop := TRUE;
```

Wenn ein Laufzeitsystemdienst angefordert wird, aber gerade deaktiviert ist, wird der Anwender durch eine entsprechende Meldungsbox darauf aufmerksam gemacht. So kann er mit passenden Gegenmaßnahmen auf dem Prozess/der Maschine reagieren.



Meldungsbox für den Fall `PlcOperationControll.xDisableApplicationStop:=TRUE`



Es liegt in der Verantwortung des Applikationsprogrammierers, dass die Dienste in bestimmten sicheren Maschinenzuständen aktiviert und in kritischen deaktiviert sind! Um eine Rückfalllösung zu behalten, können "Reset Ursprung" und Löschen der Applikation nicht unterbunden werden!

## 4.12 Script Sprache

Die Script-Sprache, welche in CODESYS implementiert ist, stellt ein leistungsfähiges Werkzeug zum Automatisieren von Abläufen in CODESYS dar. Dabei können einzelne Befehle oder komplexe Programmabläufe direkt aus der CODESYS Benutzeroberfläche oder aus der Windows Kommandozeile gestartet werden. Die CODESYS Script-Sprache ist eine modulare Sprache, welche auf IronPython aufsetzt. Dazu bindet das CODESYS Plugin 'ScriptEngine' den IronPython-Interpreter mit in die CODESYS Programmierumgebung ein. Diese Implementierung ermöglicht die umfangreichen Framework-Bibliotheken von Python zu benutzen, welche neben anderen Funktionen auch den Dateizugriff in Netzwerken beinhaltet.

### Die Script Sprache IronPython

Python ist eine dynamische Sprache, welche zu vielen Zwecken eingesetzt wird. Der Schwerpunkt der Sprache liegt auf einfachem und ausdrucksstarkem Code. Sie erlaubt dem Entwickler maximale Flexibilität. IronPython ist eine Implementierung von Python für .NET und erlaubt den vollen Zugriff auf .NET Framework und Klassen. Die Implementierung des IronPython Interpreter basiert auf Python Version 2.7.

Es gibt eine Vielzahl kostenloser Handbücher und Hilfeseiten im Internet. Die folgenden Links geben Ihnen eine Einführung als auch detaillierte Informationen über IronPython.

<http://wiki.python.org/moin/BeginnersGuide>: Nützliche Handbücher um IronPython zu lernen

<http://wiki.python.org/moin/GermanLanguage>: Sammlung von Hyperlinks zu deutschen Hilfeseiten

<http://docs.python.org/release/2.7/>: Die offizielle Dokumentation zu Python 2.7

<http://ironpython.codeplex.com/>: Die IronPython Homepage

## 4.13 Bibliotheksverwaltung

Bibliotheken sind Sammlungen wiederverwendbarer Objekte wie:

POUs wie Funktionsbausteine oder Funktionen

Interfaces und ihre Methoden und Properties

Datentypen wie Enumerationen, Strukturen, Aliases, Unions

Globale Variablen, Konstanten, Parameterlisten

Textlisten, Bildersammlungen, Visualisierungen, Visualisierungselemente

Externe Dateien (z.B. Dokumentationen)

Kurvenscheibentabellen

Durch das Einbinden einer Bibliothek in ein Projekt, können die bereitgestellten Module genauso im Projekt verwendet werden wie die anderen Bausteine und Variablen, die direkt im Projekt definiert sind.

Die Verwaltung der Bibliotheken in einem Projekt erfolgt im Bibliotheksverwalter, die vorherige Installation auf dem System über den Bibliotheks-Repository-Dialog.

Die Standard-Erweiterung für eine Bibliotheksdatei ist **".library"**, im Gegensatz zu **".lib"**, was für die Dateien in CODESYS V2.3 und früheren Versionen verwendet wurden. Weitere Bibliothekstypen sind: Bibliotheken, die in der Form geschützt sind, dass ihr Quellcode nicht mehr zugänglich ist (**\*.compiled-library**), Interface-Bibliotheken (**\*\_Itfs.library**) und Container-Bibliotheken (**\*\_Cnt.library**). Siehe hierzu auch: „Richtlinien für das Erstellen von Bibliotheken“.



Für das Erstellen, Verschlüsseln und Schützen von Bibliotheken sollten bestimmte Richtlinien befolgt werden um Konsistenzprobleme zu vermeiden. Sehen Sie unbedingt „Richtlinien für das Erstellen von Bibliotheken“.

Die Projektfunktionen für globales und lokales „Suchen & Ersetzen“ können auch für nicht verschlüsselte Bibliotheken verwendet werden.



Der Umfang der verfügbaren Funktionalität für die Bibliotheksverwaltung hängt vom aktuell verwendeten „Optionen, Features“ Feature-Set.

### 4.13.1 Installation und Einbinden im Projekt

Bibliotheken können auf dem lokalen System in verschiedenen "Repositories" (Verzeichnisse, Speicherorte) verwaltet werden. Bevor eine Bibliothek in ein Projekt eingebunden werden kann, muss sie auf dem lokalen System in ein solches Repository installiert werden. Dies erfolgt im Bibliotheks-Repository Dialog in CODESYS.

Als Voraussetzung für die Installation müssen die „Projektinformationen“ eines Bibliotheksprojekts einen Titel, eine Versionsinfo und den Herstellernamen (Firma) erhalten. Optional kann eine Kategoriebezeichnung angegeben sein, die später im Bibliotheksverwalter zur Sortierung dienen kann.

Wenn keine Kategoriezuweisung in den Projektinformationen vorliegt, gehört die Bibliothek automatisch zu Kategorie "Sonstige". Weitere **Kategorien** können in einem oder mehreren XML-Datei(en) **\*.libcat.xml** definiert sein, die im Projektinformationen-Dialog geladen werden können, um eine Kategorie auszuwählen. Wenn Sie Kategorien definieren wollen, sehen Sie bitte zunächst die Richtlinien zum Erstellen von Bibliotheken „Richtlinien für das Erstellen von Bibliotheken“.

Der „Bibliotheksverwalter“ wird verwendet, um Bibliotheken in ein Projekt einzubinden. In einem "Standardprojekt" ist ein Bibliotheksverwalter-Objekt automatisch dem Standard-Gerät zugewiesen. Es kann aber auch explizit im Gerätebaum (unterhalb eines Geräts oder einer Applikation) oder im POUs-Fenster hinzugefügt werden. Dies erfolgt wie für andere Objekte über den „Objekt hinzufügen“ Dialog. Bibliotheken, die in einer Bibliothek eingebunden sind,

werden standardmäßig im Bibliotheksverwalter ebenfalls dargestellt, "versteckte Bibliotheken" sind allerdings auch möglich, siehe die Richtlinien zum Erstellen von Bibliotheken: „Richtlinien für das Erstellen von Bibliotheken“.

Liegt die Bibliothek nicht nur in verschlüsselter Form vor, sondern ist auch die ".library\*" Datei vorhanden, so können die im Bibliotheksverwalter aufgeführten POU's der Bibliothek durch einen Doppelklick auf den jeweiligen Eintrag geöffnet werden.

Wenn eine Bibliotheksfunktion im Projekt angesprochen wird, werden die Bibliotheken und Repositories in der **Reihenfolge** durchsucht, in der sie im Bibliotheks-Repository Dialog aufgeführt sind. Sehen Sie Eindeutiger Zugriff.

#### 4.13.2 Referenzierte Bibliotheken

Eine Bibliothek kann andere Bibliotheken einbinden (referenzierte Bibliotheken), wobei die Verschachtelung beliebig tief sein kann. Wenn eine solche "Vater"-Bibliothek dann selbst in ein Projekt eingebunden wird, stehen die in ihr referenzierten Bibliotheken dort ebenfalls zur Verfügung. Achtung: Allerdings nur diejenigen, die in der "Vater"-Bibliothek im globalen Bibliotheksverwalter (POUs-Fenster) eingebunden sind!

Vor dem Anlegen von Bibliotheksreferenzen sehen Sie bitte die detaillierteren Angaben in den Richtlinien: „Richtlinien für das Erstellen von Bibliotheken“ ! An dieser Stelle hier nur einige Grundlagen:

Bibliothekreferenzen sollten über Bibliotheks-**Platzhalter** definiert werden, um Probleme zu vermeiden, die durch Versionsabhängigkeiten und die Notwendigkeit herstellerspezifische Bibliotheken zu verwenden entstehen können.

In den Eigenschaften jeder referenzierten Bibliothek kann festgelegt werden, wie sie sich später, wenn sie über die "Vater"-Bibliothek in ein Projekt eingebunden wird, verhalten soll; beispielsweise ob sie im Bibliotheksverwalter "verborgen" werden soll, oder ob - im Fall einer Container-Bibliothek - der Zugriffspfad den Namensraum nicht enthalten soll.

Zum eindeutigen Zugriff auf Bausteine oder Variablen referenzierter Bibliotheken sehen Sie bitte auch die Richtlinien: „Richtlinien für das Erstellen von Bibliotheken“ oder unten: „Eindeutiger Zugriff auf Bibliotheksmodule oder -variablen“.

#### 4.13.3 Bibliotheksplatzhalter, Bibliotheksprofile

Mehrere Versionen einer Bibliothek können gleichzeitig auf dem System installiert sein.

Mehrere Versionen einer Bibliothek können gleichzeitig im Projekt eingebunden sein. Folgendermaßen wird eindeutig geregelt, auf welche Version eine Applikation in diesem Fall zugreift:

Wenn **mehrere Versionen auf derselben Ebene** im Bibliotheksverwalter vorliegen, hängt es von der Definition in den „Eigenschaften...“ ab, welche Version verwendet wird (eine bestimmte oder immer die neueste).

Wenn **mehrere Versionen auf verschiedenen Ebenen** vorliegen (was im Falle von referenzierten Bibliotheken ( „Referenzierte Bibliotheken“) der Fall sein kann), wird ein eindeutiger Zugriff auf Bibliotheksbausteine durch die Angabe des entsprechenden **Namensraums** erreicht; siehe folgender Abschnitt.

Insbesondere wenn andere Bibliotheken in einer Bibliothek referenziert werden, aber auch um ein Projekt für die Verwendung auf anderen Zielgeräten kompatibel zu machen, wird dringend empfohlen **Platzhalter** zu verwenden. Damit können Probleme aufgrund von Versionsabhängigkeiten oder der Notwendigkeit herstellerspezifische Bibliotheken zu verwenden, vermieden werden! Siehe „Bibliotheksplatzhalter, Bibliotheksprofile“



#### 4.13.4 Eindeutiger Zugriff auf Bibliotheksmodule oder –variablen

Grundsätzlich gilt folgendes: Wenn im Projekt mehrere Bausteine gleichen Namens verfügbar sind, muss der Zugriff auf eine Bausteinkomponente eindeutig sein, ansonsten treten Compiler-Fehler auf. Dies gilt sowohl für projektlokale Bausteine wie auch für solche, die in eingebundenen und in diesen referenzierten Bibliotheken bereitstehen. Die Eindeutigkeit wird in solchen Fällen durch Hinzufügen des Namensraums vor dem Bausteinnamen erreicht.

Sehen Sie bitte für genauere Informationen zur sinnvollen Definition eines Namensraums, sowie auch zur Möglichkeit, seine Voranstellung bei Zugriff auf Bibliothekselemente über das Property "LanguageModelAttribute" in den Projektinformationen zu erzwingen: „Richtlinien für das Erstellen von Bibliotheken“

Beispiele: In den folgenden Beispielen sei der "Namensraum" der Bibliothek Lib1 in den Bibliothekseigenschaften mit "Lib1" eingetragen. Sehen Sie in der rechten Spalte die Verwendung der Namensräume für eindeutige Zugriffe auf Variable var 1, die in Baustein module1 und in Baustein POU1 definiert ist..

Variable var1 ist an allen Stellen (1) bis (5) im Projekt definiert:	Eindeutiger Zugriff auf var1 mit Hilfe der entsprechenden Namensraum-Angaben...
(1) In Bibliothek Lib1 im globalen Bibliotheksverwalter im POU-S-Fenster	"Lib1.module1.var1"
(2) In Bibliothek Lib1 im Bibliotheksverwalter unterhalb von Applikation App1 von Gerät Dev1 im Geräte-Fenster	"Dev1.App1.Lib1.module1.var1"
(3) In Bibliothek Lib1, die in Bibliothek F_Lib eingebunden (referenziert) ist, im globalen Bibliotheksverwalter im POU-S-Fenster	Standardmäßig: (Option "...sichtbar machen..." ist beim Einfügen von Lib1 in F_Lib in den Bibliothekseigenschaften von Lib1 deaktiviert): "F_Lib.Lib1.module1.var1"  Wenn Option "...sichtbar machen..." aktiviert wäre, würde module1 wie eine Komponente einer top-level eingebundenen Bibliothek behandelt. Dann wäre normalerweise auch ein Zugriff über "Lib1.module1.var1" möglich, oder falls die Bibliothek nicht die Eigenschaft "qualified-access-only" hat (siehe unten), über "module1.var1"). Im vorliegenden Beispiel würde dies allerdings zu einem Compiler-Fehler führen, da der Aufruf dann nicht mehr eindeutig wäre, siehe (1) und (4).
(4) Im Objekt module1 das im POU-S-Fenster definiert ist	"module1.var1"
(5) Im Objekt POU1 das im POU-S-Fenster definiert ist	"POU1.var1"

#### 4.13.5 CODESYS V2.3 Bibliotheken

Bibliotheksprojekte, die in CODESYS V2.3 (\*.lib) und früheren Versionen erstellt wurden, werden weiterhin unterstützt:

Ein "altes" Bibliotheksprojekt (\*.lib) kann in CODESYS V3 geöffnet und dabei in eine "CODESYS Bibliothek" (\*.library) konvertiert werden.

Wenn ein altes Projekt geöffnet wird, das alte Bibliotheken referenziert, kann gewählt werden, ob diese Referenzen beibehalten, durch andere ersetzt oder gelöscht werden sollen. Sollen sie beibehalten werden, werden die betreffenden Bibliotheken in das neue Format konvertiert und dabei automatisch in das System-Bibliotheks-Repository installiert. Falls sie die nötigen Projektinformationen nicht enthalten, können diese unmittelbar nachgetragen werden. Das Muster (Mapping), nach dem eine alte Bibliothek einmal beim Konvertieren eines alten Projektes gehandhabt wurde, kann in den Projekt-Optionen gespeichert werden, so dass dieselbe Bibliothek bei künftigen Projektkonvertierungen nicht jedesmal erneut explizit behandelt werden muss.

Siehe unter „Projekt öffnen...“ für eine Beschreibung zum Vorgehen bei Konvertierung von Projekten und Bibliotheken.

#### 4.13.6 Externe und interne Bibliotheken bzw. Bibliotheksbausteine, spätes Binden

Eine "**externe Bibliothek**" im Unterschied zu einer internen Bibliothek (CODESYS Bibliotheksprojekt) ist eine Bibliotheksdatei, die außerhalb von CODESYS in einer anderen Programmiersprache, wie beispielsweise C, programmiert wird. Sie muss auf dem Zielsystem vorliegen und wird erst gebunden, wenn die Applikation dort läuft. Das Platzhalterkonzept („Bibliotheksplatzhalter, Bibliotheksprofile“) sollte für die Handhabung der Zielsystemabhängigkeiten verwendet werden.

Um die Erstellung einer externen Bibliothek vorzubereiten, erlaubt CODESYS **die entsprechenden C-Frame und C-Schnittstellen-Dateien** (m4-Dateiformat) aus dem aktuellen CODESYS Bibliotheksprojekt zu generieren. Die Verwendung dieser Funktionalität garantiert die Konsistenz zwischen der CODESYS Bibliothek und ihrer externen Implementierung. Sehen Sie hierzu den entsprechenden Befehl „Erzeuge Laufzeitsystemdateien...“.

Wie in CODESYS V2.3 ist es auch möglich, eine CODESYS Bibliothek als externe Bibliothek später zu binden, d.h. erst wenn die Applikation auf dem Laufzeitsystem betrieben wird. Zusätzlich ist es nun sogar möglich, das "späte Binden" für jeden Bibliotheksbaustein einzeln zu definieren. Zu diesem Zweck kann die Eigenschaft in den Objekteigenschaften (" Externe Implementierung") eines oder aller Bausteine aktiviert werden.

#### 4.13.7 Lizenzierte Bibliotheken

Bibliotheken können durch eine Lizenz geschützt sein. Die Lizenzverwaltung erfolgt im „Lizenz-Manager“.

Lizenzierte Bibliotheken können jederzeit im Bibliotheks-Repository installiert werden, allerdings muss für die Verwendung ihrer Bausteine in einem Projekt ein Dongle mit einer gültigen Lizenz auf dem Rechner aufgesteckt sein. Wenn keine Lizenz vorliegt, wird eine entsprechende Meldung ausgegeben sobald eine lizenzierte Bibliothek im Bibliotheksverwalter eingefügt wird. Die Bibliothek kann dann trotzdem eingefügt werden, aber sie wird mit einem roten Kreuz gekennzeichnet und das Projekt kann nicht mehr fehlerfrei übersetzt werden, wenn es einen Baustein aus der Bibliothek verwendet. Im Details-Dialog der Bibliothek - sowohl im Bibliotheks-Repository als auch im Bibliotheksverwalter - findet man Informationen zur benötigten Lizenz.

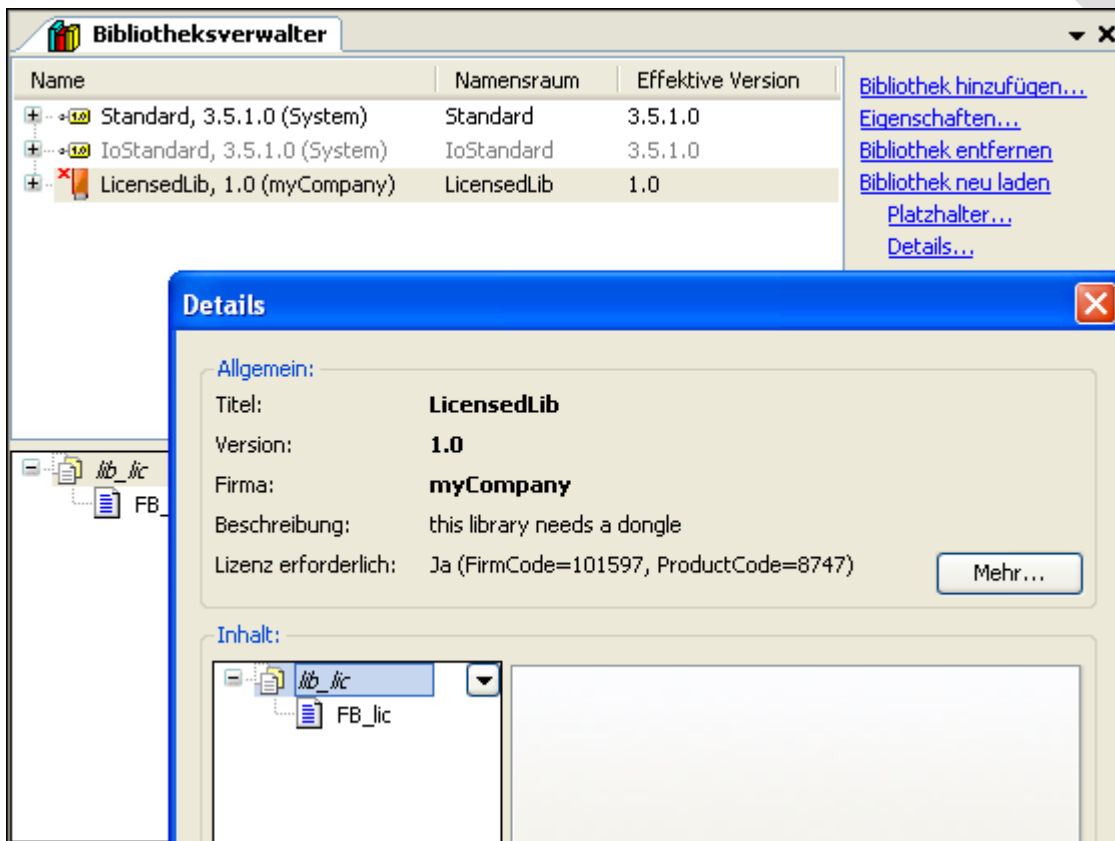


Abb.: Lizenzierte Bibliothek im Bibliotheksverwalter, momentan keine Lizenz verfügbar

## 4.14 Alarmverwaltung

Die Alarmverwaltung kann als Teil einer Applikation verwendet werden, um kritische Prozesszustände festzustellen, aufzuzeichnen und mit Hilfe von Visualisierungselementen darzustellen. Über die Applikation bzw. Benutzereingaben in der Visualisierung können Alarme wahrgenommen werden und es kann entsprechend mit Quittierung und Folgeaktionen darauf reagiert werden. Spezielle Visualisierungselemente wie auch eine Bibliothek stehen dafür zur Verfügung.

Ein entsprechender Alarm Configuration Baum muss im Projekt unterhalb der jeweiligen Applikation eingerichtet werden.

Aufsetzen der Alarmkonfiguration für eine Applikation

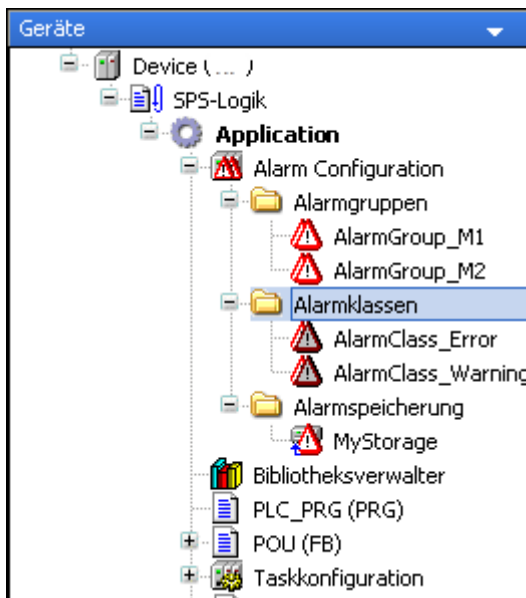


Abb. 58: Alarmkonfiguration im Gerätebaum, Beispiel

Fügen Sie zunächst über den Befehl Objekt hinzufügen ein Alarm Configuration Objekt unterhalb der Applikation im Gerätebaum ein. Die passende Bibliothek AlarmManager.library wird automatisch im Bibliotheksverwalter eingetragen.

Fügen Sie die folgenden Objekte unterhalb des "Alarm Configuration" Knotens ein: Alarm class, Alarm group, Alarm storage.

Definieren Sie die gewünschten Alarmgruppen

Konfigurieren Sie die erforderlichen Alarmklassen

Konfigurieren Sie die Alarmspeicherung

Programmieren Sie die Alarmverwaltung wie gewünscht in den Applikations-POUs

Erstellen Sie eine Visualisierung mit Hilfe des Alarntabelle- bzw. Alarmbanner-Elements

## 4.15 Package Manager

CODESYS bietet mit dem Package Manager eine Funktionalität, mit welcher die Standard-Installation von CODESYS um zusätzliche Features und Konfigurationseinstellungen erweitert werden kann. Das Konzept des Package Managers ähnelt der von typischen Windows-Installations-Mechanismen.

Die Hauptfunktionen des CODESYS Package Management sind:

- Installation von Package-Dateien in CODESYS oder einer anderen Automation Plattform Anwendung. Unterschiedliche Versionen des gleichen Package können nebeneinander installiert werden.
- Download von Package-Dateien von einem webbasierenden Server mit direkter Installation. Diese Funktion ist derzeit noch nicht implementiert.
- Deinstallation von bereits installierten Packages.
- Auflisten aller bereits installierter Packages.
- Der Package Manager selber kann über das CODESYS Menü oder als alleinstehende Anwendung mit Kommandozeile aufgerufen werden.

Ein Package ist eine ZIP-Datei mit der Erweiterung .package. Folgende Objekte können Teil eines Packages sein:

- Plug-Ins
- Bibliotheken
- Gerätebeschreibungen
- Lieferantenbeschreibungen
- Profile
- Informationsprofile
- Profileänderungen
- Dateien
- Erweiterungen zur Menükonfiguration
- Erweiterungen zur Konfiguration der Werkzeugleiste
- Erweiterungen zur Konfiguration der Tastenkürzel
- Erweiterungen zur Konfiguration der Ansichten
- Module zur Onlinehilfe
- Komplette Menükonfigurationen für spezielle Profile
- Komplette Werkzeugleisten-Konfigurationen für spezielle Profile
- Komplette Tastatur-Konfigurationen für spezielle Profile
- Optionen
- Bibliotheksprofile

## **5 Geräte Schnittstellen Konfigurationen**

### **5.1 CANbus Konfigurationseditor**




### **5.2 EtherCAT Konfigurationseditor**

### **5.3 Modbus Konfigurationseditor**

## 6 Target Visualisierung

### 6.1 Übersicht

Die Visualisierung auf den elrest Geräten:

-  **visio**
-  **combo**
-  **robusto**

erfolgt in einer reinen WEB-Technologie.

Der wesentliche Vorteil liegt in der Verwendung aller gängigen PAD's (iPAD, Samsung,... ) und Phones (iPhone, Android, Blackberry, u.v.m) als Bedien- und Visualisierungsgerät.








Dabei können Regelungs- und Steuerungsaufgaben in CoDeSys V3 umgesetzt werden.

Der Einsatzbereich erscheint nahezu unbegrenzt.



#### 6.1.1 Client Seite

Die Client Seite ist ein Webbrowser der SVG, JavaScript und asynchronous http requests (XMLHttpRequest) unterstützt. Folgende Webbrowser werden unterstützt:


Name	Vendor	Logo	V	OK	Plattform	OK	Kommentare
Firefox	Mozilla		3.x 2.x 1.5+	OK OK OK	Linux Windows Android	OK OK NT	
Chrome	Google		1.x	OK	Windows	OK	
Internet Explorer	Microsoft		8 7+ 6 SP2	OK OK OK	Windows	OK	
Safari	Apple		4.x 3.x	OK OK	Windows Mac OS-X	OK OK	
Opera	Opera		10.x 9.5+ 8.5	OK OK (part)	Windows Win. Mobile	OK OK	
Blackberry	RIM		6	OK	Blackberry 6	NT	
Linux	QT		2.6.x	OK	Linux	OK	

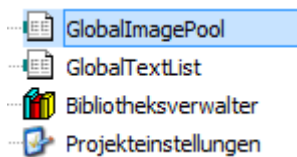
## 6.1.2 Server Seite

Die Server Seite wird unterstützt durch:

- visio-PMC4xx
- combo-CM211
- robusto-RMC4xx

## 6.2 Projektverwaltung

Auf dem Tabreiter  können verschiedene projektglobale Ressourcen verwaltet werden:



### 6.2.1 Bilder integrieren (GlobalImagePool)



Unterstützte Bildformate sind BMP, EMF, GIF, ICO, JPG, PNG, SVG and TIV.

Beachten Sie, dass eine Steuerung nicht immer alle Formate unterstützt.

## 6.3 Tools und Editoren der Visualisierung

Eine CODESYS V3 Visualisierung basiert auf einem modularen Visualisierungskonzept und stellt die folgenden Tools/Editoren zur Verfügung.

- „Visualisierungseditor“: Hier wird ermöglicht Visualisierungen/Masken/Panels in einem IEC 61131-3-Tool mit wenigen Klicks zu erstellen.
- „Visualisierungselemente“: Fertige Elemente stehen in einer Vielzahl einsatzfähig zur Verfügung.
- „Visualisierungsprofile“: Ermöglicht es aus Visualisierungsbibliotheken ein Profil zu definieren. Jedes Visualisierungsprojekt, das wenigstens eine Visualisierung enthält, muss auf einer basieren.
- „Visualisierungsbibliotheken“: Hier können Elemente zusammengefasst werden um sie für den Einsatz in anderen Projekten zur Verfügung zu stellen.
- „Visualisierungsstile“: Hier wird die Elementdarstellung definiert um ein bestimmtes Look-and-Feel der Elemente zu erhalten.
- „Visualisierungselemente-Repository“: Hier werden Elemente und Profile auf dem lokalen System verwaltet.
- Visualisierungsstilbibliotheken: Hier werden die verschiedenen Stile zusammengefasst.
- „Visualisierungsstile-Repository“: Hier werden die Stilbibliotheken verwaltet.
- „Projekteinstellungen“: Einstellungen zu Visualisierungsprojekten
- „Visualisierungsmanager mit Clients“: Hier werden die verschiedenen Visualisierungsclients verwaltet, die flexibel und mit geringem Aufwand für die unterschiedlichsten kundenspezifischen Anwendungen eingesetzt werden.



Visualisierungen mit CODESYS V3 ermöglichen Ihnen:

1. Visualisierungen durch Referenzen zu verschachteln und/oder zwischen verschiedenen Visualisierungen umzuschalten: „Frame“
2. Schnittstellen für Parametertransfer zu verwenden um komplexe Visualisierungen zu instanzieren: „Grundlegendes zum Schnittstelleneditor“
3. Unterstützung der Mehrsprachigkeit durch Textlisteneditoren: „Text und Sprache in Visualisierungen“
4. eine Benutzerverwaltung für Visualisierungen zu installieren: „Benutzerverwaltung“

### Allgemeiner Mechanismus

Entfernte Visualisierungsclients sind nichts anderes als Interpreter von Zeichnkommandos. Unabhängig von ihrem Typ (Web, Target, HMI) erhält jeder Client die gleichen Instruktionen, so dass die daraus resultierenden Visualisierungen gleich sind.

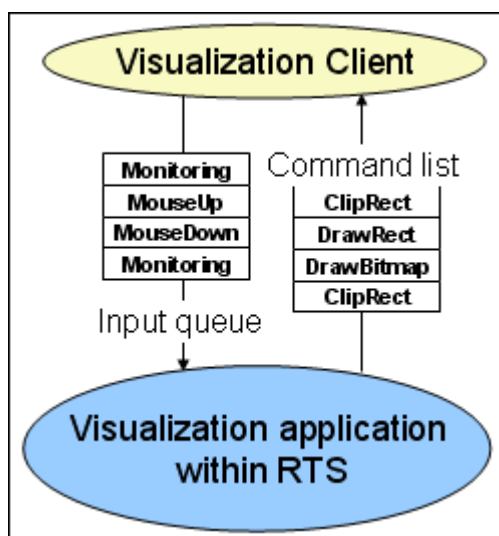


Abb. Allgemeiner Mechanismus

Als Konsequenz muss die verbundene Visualisierungsapplikation auf die SPS runter geladen werden, was bei kleinen SPSen zu Schwierigkeiten führen kann. Um dieses Problem zu vermeiden, kann ein Datenserver in das Laufzeitsystem integriert werden, der die notwendigen Daten für die Visualisierung anbietet.

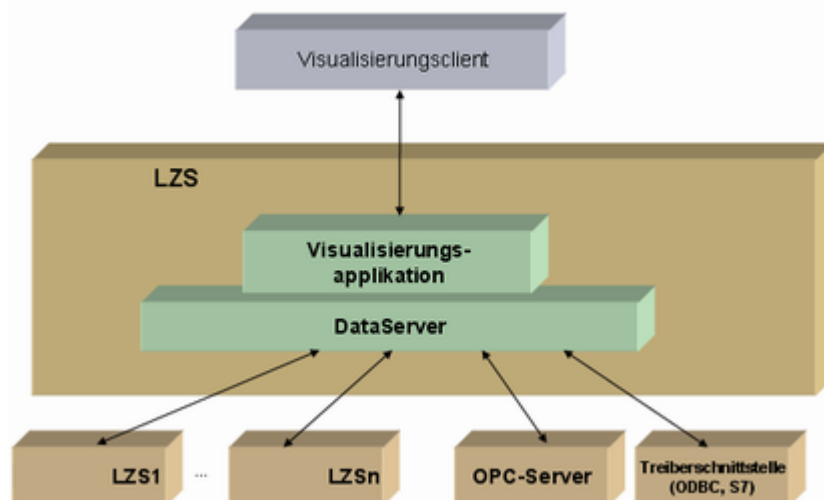
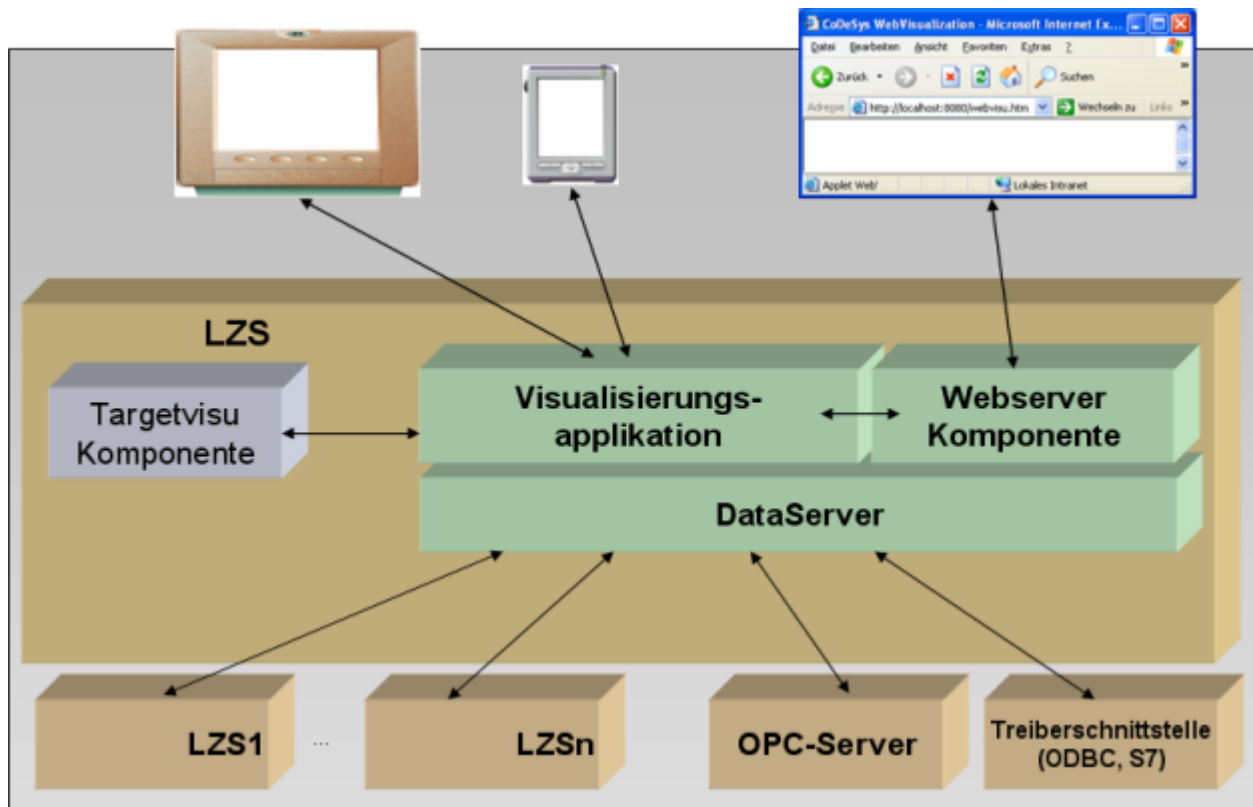


Abb. 2: Daten-Server, der die Visualisierungsdaten bereitstellt

Eine Visualisierungsapplikation, d.h. die Visualisierungselemente werden in IEC-Code im Visualisierungeditor des Programmiersystems angelegt. Falls die Visualisierung auf dem Gerät laufen soll, wird Visualisierungscode erzeugt und dorthin geladen.



Verschiedene Visualisierungsclients:

„Diagnosevisualisierung“: Dieser Client läuft nur im Programmiersystem entsprechend der reinen Online-Visualisierung in CODESYS V2.3.

„CODESYS WebVisu“: Er nutzt zur Verbindung mit der zugehörigen Applikation einen Web-Server.

„CODESYS TargetVisu“: Hier wird Steuerung und Visualisierung auf einem Gerät kombiniert.

„Grundlegendes zu CODESYS HMI“: Hier wird die Visualisierung auf einem externen Gerät dargestellt.

„CODESYS HMI(Target)“

Externe (entfernte) Daten-Server können in Visualisierungen verwendet werden. Zu diesem Zweck muss dieser in der lokalen Applikat

### 6.3.1 Erzeugen einer Visualisierung

Ein Visualisierungsobjekt kann im Gerätebaum unterhalb einer Applikation oder im POU-Pool durch Verwendung des Befehls Projekt→Objekt hinzufügen (auch im Kontextmenü) eingefügt werden. Das Fenster eines neuen Visualisierungsobjekts ist betitelt mit <visu object name> und zeigt zunächst einen leeren Bereich an.

Wird das erste Visualisierungsobjekt im Projekt hinzugefügt, dann werden die zugehörigen Visualisierungsbibliotheken automatisch vom Bibliotheksmanager entsprechend dem aktuell aktiven „Visualisierungsprofile“ auch hinzugefügt. Eine Visualisierungsbibliothek ist immer als ein spezieller Typ einer **„Platzhalterbibliothek“** mit der Auswirkung eingerichtet, dass die exakte Version der verwendeten Bibliothek nicht aufgelöst wird, bis die Bibliothek im Projekt eingefügt wird. Nur dann definiert das aktuell aktive Profil, welche Version tatsächlich benötigt wird. Bitte beachten Sie, dass dieser Bibliothekstyp sich von den Geräte spezifischen Platzhalterbibliotheken unterscheidet. Dort werden die Platzhalter in der Gerätebeschreibung aufgelöst. Die Basisbibliotheken, die standardmäßig eingefügt werden, wenn Visualisierungsobjekt eingefügt wird, sind VisuElem, VisuElemMeter, VisuElemWinControls, VisuElemTrace, VisuInputs. Sie fügen zusätzlich weitere Bibliotheken ein.

Eine Visualisierung kann als Objekt im globalen Objekte-Pool, d.h. im POU-Fenster angelegt werden, oder direkt einer Applikation zugeordnet im Gerätefenster. Sobald die erste Visualisierung unter einer Applikation hinzugefügt wurde, wird auch ein „Visualisierungsmanager mit Clients“ hinzugefügt.

Eine Startvisualisierung, also eine Visualisierung, die nach einem Login auf der SPS als erstes geöffnet wird, muss im Gerätefenster unterhalb des zugehörigen Applikationsobjekts eingefügt werden.



Im Allgemeinen muss **keine** Visualisierungsbibliothek händisch eingefügt oder explizit im Projekt verwendet werden.

Visualisierung, Dialog oder Nummernfeld/Tastatur?

Jede Visualisierung hat „Eigenschaften eines Visualisierungsobjekts“ nach seiner bestimmungsgemäßen Verwendung: „*Visualisierung*“, „*Nummernfeld/Tastatur*“, „*Dialog*“) oder Anzeigengröße. Beachten Sie in diesem Zusammenhang, dass eine Visualisierung explizit für den Einsatz als Benutzereingabedialog in anderen Visualisierungen erstellt und konfiguriert werden kann („Eingabekonfiguration“). Implizit ist auch standardmäßig Nummernfeld oder Tastatur verfügbar. Die Verwendung von Tastatur und Dialog kann in den Eigenschaften „*Eingabekonfiguration*“ eines Visualisierungselements definiert werden.



Die Visualisierung arbeitet mit einem integrierten Laufzeitsystem, deshalb erscheinen während der Arbeit im Editor gegebenenfalls Start- und Download-Meldungen.

## 6.3.2 Visualisierungsbefehle

Die Visualisierungsbefehle werden vom Plug-in 'Visual Editor' für die Menükategorie „Visualisierungskommandos“ zur Verfügung gestellt, die im Dialog Tools →Anpassen zu finden sind. Sie ermöglichen es ein Visualisierungsobjekt im Visualisierungseditor zu editieren.

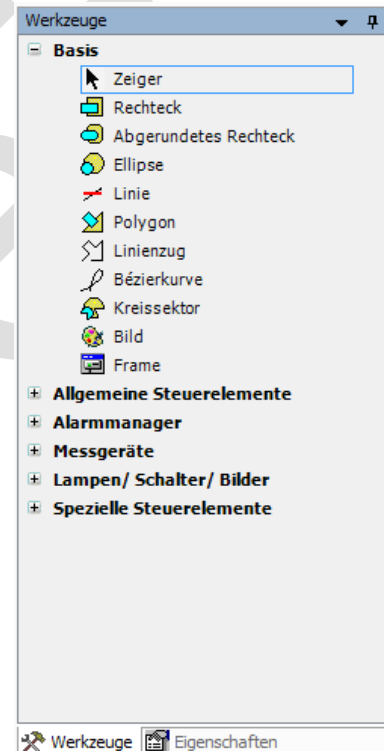
Die meisten sind standardmäßig Teil des Menüs „Visualisierung“ und deswegen auch im Kontextmenü des Visualisierungseditors vorhanden. Falls nötig, ist der Dialog Tools →Anpassen zu öffnen, um die Menükonfiguration der Kategorie „Visualisierungskommandos“ anzusehen oder zu ändern.

### 6.3.2.1 Visualisierungselement einfügen

Dieser Befehl (Kategorie „Visualisierungskommandos“) steht im Menü „Visualisierung“ und im Kontextmenü zur Verfügung. Er wird verwendet um ein Element in der aktuellen Visualisierung einzufügen. Dies entspricht dem Einfügen per Drag & Drop der Toolbox („Werkzeuge“).

Der Befehl öffnet ein Untermenü, in dem aus den aktuell verfügbaren Elementen das gewünschte ausgewählt werden.

Das Element wird daraufhin in der linken oberen Ecke des Editor-Fensters eingefügt.

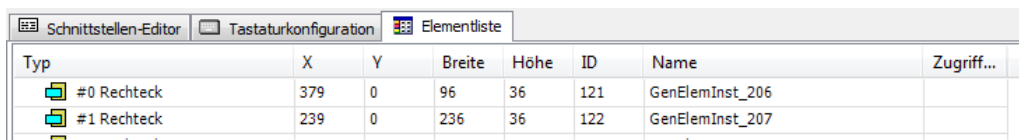


### 6.3.2.2 Elementliste

Symbol: 

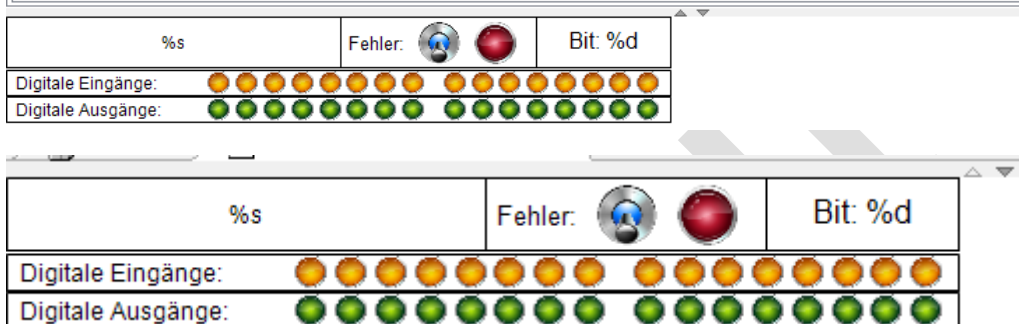
Dieser Befehl (Kategorie „*Visualisierungskommandos*“) öffnet den „Elementliste“ für die aktuelle Visualisierung. Er wird in einer Registerblattansicht im oberen Teil des Visualisierungseditors angezeigt.

Wenn aktive, können die Einstellungen eingeblendet werden.



Typ	X	Y	Breite	Höhe	ID	Name	Zugriff...
#0 Rechteck	379	0	96	36	121	GenElemInst_206	
#1 Rechteck	239	0	236	36	122	GenElemInst_207	

ausgeblendet:



### 6.3.2.3 Tastaturbedienung aktivieren

Symbol: 

Dieser Befehl (Kategorie „*Visualisierungskommandos*“) ist in der Menüleiste für eine integrierte Visualisierung ( „*Diagnosevisualisierung*“) verfügbar. Er aktiviert bzw. deaktiviert die „Tastaturbedienung im Online-Modus“ im Online-Modus einer Visualisierung.

Wenn die Tastaturbedienung aktiviert ist, können Eingaben auf Elemente und die Auswahl der Elemente über bestimmte Tastenkombinationen vorgenommen werden. In diesem Fall werden andere über Tastatur gegebene Befehle nicht ausgeführt, solange der Visualisierungseditor aktiv und im Online-Modus ist.

### 6.3.2.4 Ausrichtung

### 6.3.2.5 Gruppieren

### 6.3.2.6 Hintergrund

Symbol: 

Dieser Befehl (Kategorie „Visualisierungskommandos“) öffnet den Dialog 'Hintergrund', in dem eine Farbe und/oder eine Abbildung für den Visualisierungshintergrund ausgewählt werden kann. Aktivieren Sie die gewünschte(n) Option(en):

„Bitmap“: Um ein Hintergrundbild zu definieren, muss hier der Pfad einer Bilddatei, die in einer Bildersammlung im Projekt verfügbar ist, eingetragen werden. Geben Sie dazu den Namen der Bildersammlung und die ID der Bilddatei - getrennt durch einen Punkt "." - ein: <bildersammlung>.<ID> (z.B. "Images\_1.drive\_icon", "Images\_1.43").

„Farbe“: Um die Hintergrundfarbe der Visualisierung zu definieren, wählen Sie aus der Farbauswahlliste die gewünschte Farbe aus.

### 6.3.2.7 Einfügen von Elementen zur Alarmquittierung

Dieser Befehl (Kategorie „Visualisierungskommandos“) steht standardmäßig im Menü „Visualisierung“ und im Kontextmenü bereit, wenn eine „Alarmtabelle“ im Visualisierungseditor ausgewählt ist. Er öffnet einen Assistenten für das Einfügen von Kontrollelementen unterhalb der Tabelle:



Abb. 6: Assistent für Alarmtabelle-Zusatzelemente

Wählen Sie in „Zu verwendender Elementtyp“, ob die Kontrollelemente jeweils als „Schaltfläche“ oder nur als einfaches „Rechteck“ dargestellt werden sollen.

„Aktion“: Wählen Sie Aktionen aus, die über Eingabe auf die Schaltfläche bzw. das Rechteck ausgeführt werden sollen. Wenn die Konfiguration der Alarmtabelle (Eigenschaftskategorie "Kontrollvariablen") bereits eine Variable für eine Aktion angegeben ist, dann wird diese hier in Spalte „Variable“ angezeigt. Wenn noch keine solche Variable definiert ist, wird automatisch eine visualisierungslokale Variable angelegt (siehe "bFreezeScrlPos" in der oben gezeigten Abbildung). (siehe "bFreezeScrlPos" in der Abbildung oben).

Nach Schließen des Dialogs mit „OK“ wird im Visualisierungseditor für die gewählten Aktionen jeweils ein Element eingefügt. Die zugewiesene Variable wird jeweils in Kategorie *Eingabekonfiguration* → *Umschalten* der Elementeigenschaften eingetragen. Somit startet ein Mausklick auf das Element im Online-Betrieb die entsprechende Aktion:

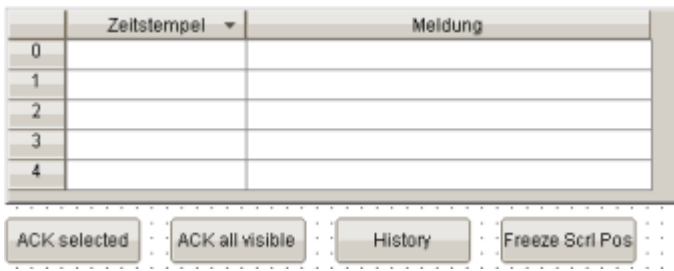


Abb. 7: Schaltflächenelemente unterhalb der Alarmtabelle

### 6.3.2.8 Frame-Parameter in <geöffnete Visualisierung> aktualisieren

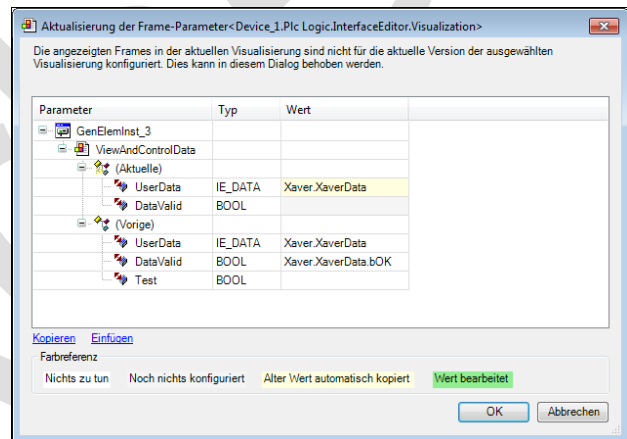
Der Befehl ist im angepassten Menü nur vorhanden und sichtbar, wenn der Visualisierungseditor geöffnet ist.

Gewöhnlich ist der Befehl dann notwendig, wenn es Compile-Fehler gibt, die Frame-Parameter betreffen. Dann öffnet der folgende Dialog um die Frame-Parameter aktualisieren zu können. Typischerweise kommt die parametrisierte Visualisierung aus einer Bibliothek, wurde dort aktualisiert und ein automatischer Update der Parameter im Projekt fand noch nicht statt.

Abb. Aktualisierung der Frame-Parameter <Name der Visualisierung>

Die Visualisierung, die neu konfiguriert werden kann, wird in Klammern <...> im Titel des Dialogs angezeigt.

Wie der Dialog zu verwenden ist, ist in „Aktualisierung der Frame-Parameter“ beschrieben.



### 6.3.3 Visualisierungseditor

Der Visualisierungseditor ist über das Plug-in 'Visualization Editor' verfügbar.

Eine Visualisierung kann mit Hilfe folgender Editoren und Tools erstellt werden:

- Der Editor selbst, betitelt mit <Name der Visualisierung>[<Objektpfad>], der die gemalte/editierte/programmierte Visualisierung enthält.
- Die „*Toolbox*“, die alle aktuell verfügbaren Visualisierungselemente enthält.
- Die Ansicht „*Eigenschaften*“, die alle Eigenschaften des Visualisierungselements enthält, das aktuell im Editor selektiert ist.
- Die Befehle des Menüs „*Visualisierung*“ und denjenigen des Kontextmenüs im Editor
- Der „*Schnittstelleneditor*“ um Platzhalter zu definieren, falls die Visualisierung in eine andere eingefügt werden soll.
- Die „*Tastaturkonfiguration*“ um Tasten oder Tastenkombinationen Folgeaktionen zuzuweisen. Zu beachten ist dabei jedenfalls, dass das Gerät, das die Visualisierung verwendet, die jeweiligen Tasten unterstützt.

- Die „*Elementliste*“, die einen Überblick über alle Elemente und deren Position der aktuellen Visualisierung zur Verfügung stellt und Selektion, Mehrfachselektion und Löschen erlaubt.

Wenn „*Schnittstelleneditor*“, „*Tastaturkonfiguration*“ oder „*Elementliste*“ über den entsprechenden Befehl (standardmäßig im Menü „*Visualisierung*“) aktiviert wird, wird der Visualisierungseditor zweigeteilt und enthält im oberen Teil die entsprechenden Registerblätter.

CODESYS V2.3-Visualisierungen können importiert werden.

Die Visualisierungselemente können über eine direkte Verwendung von CODESYS-Projektvariablen animiert werden oder in Form von Ausdrücken, das heißt kombiniert mit Operatoren und Konstanten. Dies erlaubt zum Beispiel die Variablen für eine Verwendung in der Visualisierung zu skalieren.

Innerhalb einer Visualisierung sind beliebige Ausdrücke und auch Funktionsaufrufe gültig.

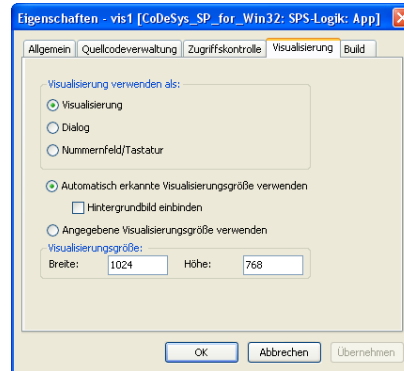
- Werkzeuge
- Elementeigenschaften
- Schnittstelleneditor
- Tastaturkonfiguration
- Elementliste



### 6.3.4 Visualisierungsobjekt

Der Dialog „*Eigenschaften*“ öffnet mit einem Klick auf den Befehl „*Eigenschaften*“ zu finden im Menü „*Ansicht*“, wenn ein Visualisierungsobjekt angewählt ist, oder im Kontextmenü eines selektierten Visualisierungsobjekts.

Abb. Dialog 'Eigenschaften', Kategorie 'Visualisierung'



Für das aktuell angewählte Visualisierungsobjekt werden folgende Eigenschaften angezeigt und können geändert werden.

„*Visualisierung verwenden als*“:

„*Visualisierung*“: Die Visualisierung wird als standardmäßiges Visualisierungsobjekt verwendet.

„*Dialog*“: Die Visualisierung stellt einen Dialog dar und ist insofern automatisch in anderen Visualisierungen verfügbar, als dass sie in *Eingabekonfiguration* → *Dialog öffnen* des Visualisierungselements zur Auswahl angeboten wird. Siehe „*Eingabedialoge*“ und „*Programmatischer Zugriff auf Eingabedialoge*“ für weitere Informationen.

„*Nummernfeld/Tastatur*“: Die Visualisierung stellt ein Nummernfeld oder eine Tastatur dar und ist insofern automatisch in anderen Visualisierungen verfügbar, als dass sie in der „*Eingabekonfiguration*“ in Kategorie „*Variable schreiben*“ (Bearbeitungstyp) zur Auswahl steht. Man beachte, dass die **Schnittstelle** einer solchen Nummernfeld/Tastatur-Visualisierung genauso aussehen muss wie die des CODESYS Standardnummernfelds/-tastatur, das durch die **VisDialogs.library** bereit gestellt wird.

„*Automatisch ermittelte Visualisierungsgröße verwenden*“: Die Größe der Visualisierung wird ermittelt, bei der alle gerade enthaltenen Visualisierungselemente und das Hintergrundbild sichtbar sind. Wenn die Option „*Hintergrundbild einbinden*“ jedoch deaktiviert ist, wird das Hintergrundbild - falls es den Bereich, der alle Visualisierungselemente einschließt, überragt - abgeschnitten.

„*Angegebene Visualisierungsgröße verwenden*“: Die Visualisierungsgröße wird durch die angezeigten Werte definiert (Breite, Höhe), wobei nicht berücksichtigt wird, ob alle Visualisierungselemente und das Hintergrundbild in diesen Bereich passen, d.h. vollständig sichtbar sind.

„*Visualisierungsgröße*“: Aktuelle „*Breite*“ and „*Höhe*“ des Visualisierungsobjekts (Anzahl Pixel)

### 6.3.4.1 Eingabedialoge

Eine Visualisierung kann zum Zweck der späteren Verwendung als Eingabedialog entsprechend erstellt und in ihren „Eigenschaften eines Visualisierungsobjekts“ als „*Dialog*“ definiert werden. Dialogvisualisierungen können dann in anderen Visualisierungen verwendet werden, um dem Benutzer eine Eingabemaske bereitzustellen.

Dialogvisualisierungen werden beispielsweise bei der Konfiguration einer „*OpenDialog*“, „*CloseDialog*“ oder „*OnDialogClosed*“ der Eigenschaft „*Eingabekonfiguration*“ eines Visualisierungselements angeboten.

Eingangs- und Ein-/Ausgangsparameter der Dialogvisualisierung müssen im „Grundlegendes zum Schnittstelleneditor“ definiert werden und müssen entsprechend von einem anderen Visualisierungselement bedient werden, das für das Öffnen des Dialogs konfiguriert wird ("Dialog öffnen"-Konfiguration). Das heißt, die Parameter können in der Konfiguration eines anderen Elements verwendet werden.



Tippsymbol

Die Eingangsparameter müssen für die „Diagnosevisualisierung“, also für die im Programmiersystem laufende Visualisierung, skalaren Typs sein. Für die Verwendung in der Target- oder Web-Visualisierung können auch andere Datentypen (auch POUs) verwendet werden.

Neben selbst erstellten Dialogvisualisierungen können **Standarddialogvisualisierungen** verwendet werden, wenn `VisuDialogs` im Projekt eingebunden wird. Aktuell sind dies Visualisierungen eines Login-Dialogs und eines Dialogs zum Öffnen und Speichern einer Datei (`VisuDialogs.Login`, `VisuDialogs.FileOpenSave`).

#### Beispiel für das Erstellen und Verwenden von eigenen Dialogvisualisierungen

Bei einem Mausklick auf eine Schaltfläche soll sich ein Dialog öffnen, in dem Zugangsdaten, in diesem Fall ein Benutzerlevel und ein Passwort, geändert werden können. Dazu erstellen wir eine Visualisierung namens `ChangeUserLevel`, die diesen Dialog darstellt, und eine weitere namens `Visualization`, die die Schaltfläche darstellt.

#### 1. Erstellen der Dialog-Visualisierung "ChangeUserLevel":

Erstellen Sie ein Visualisierungsobjekt namens `ChangeUserLevel`, das aussieht wie in der Abbildung unten gezeigt. In seinen Objekteigenschaften definieren Sie, dass es als `Dialog` verwendet werden soll. Im aktuellen Beispiel handelt es sich um einen Dialog zum Eingeben eines Benutzerlevels und eines Passworts, der nur über die OK-Schaltfläche geschlossen werden kann, wenn der angegebene Level und das Passwort zusammenpassen.

Das Bild zeigt ein Dialogfenster mit dem Titel "%s". Es enthält zwei Radio-Buttons für "Level" (0 bis 7) und ein Textfeld für "Password" mit dem Platzhalter "%s". Unten befinden sich "OK" und "Cancel" Buttons.

Abb. 21: ChangeUserLevel Dialogvisualisierung

Der Dialog setzt sich aus folgenden Elementen zusammen:

Element	Elementeigenschaften
Bild	Statische ID: <code>Login</code> ; Bild eines leeren Dialogs mit grauem Hintergrund und einer leeren blauen Titelleiste. Die Bilddatei kann über eine Bildersammlung bereitgestellt werden, wo sie mit ID "Login" abgelegt ist.
Rechteck (für die Anzeige des Dialognamens in der Titelleiste)	Texte: Text: <code>%s</code> ; wird dann ersetzt mit dem Wert des Parameters <code>stTitle</code> , siehe im folgenden;  Textvariablen: Text Variable: <code>stTitle</code> ; diese Variable ist als Eingabeparameter in der Schnittstelle der Visualisierung definiert und ihr Wert kann im Konfigurationsdialog des Visualisierungselements angegeben werden, das für das Öffnen des Dialogs verwendet wird (im vorliegenden Beispiel ist dies die "Change User Level" Schaltfläche, siehe unten) .
Radiobutton (für die Eingabe des Benutzer-Levels)	Anzahl Spalten: 4  Anordnung der Radiobuttons: <code>Von links nach rechts</code>  Radiobuttoneinstellungen: Definieren Sie die 8 Bereiche <code>[0] to [7]</code> (jeweils: Text: <code>&lt;n&gt;</code> , wobei <code>n 0 - 7</code> ist)  Variable: <code>iLevel</code> (Ein/Ausgang des Elements)
Textfeld (für die Passwort-Eingabe)	Texte: Text: <code>%s</code>  Textvariablen: Text Variable: <code>stPwd</code> (Ein/Ausgang des Elements)  Eingabekonfiguration: <code>OnMouseDown: Variable schreiben: Bearbeitungstyp: Edit, Textausgabevariable verwenden</code>
Text ("Level")	Texte: Text: <code>Level:</code>
Text ("Password")	Texte: Text: <code>Password:</code>
Schaltfläche (OK; Die Farbe zeigt an, ob Level und Passwort zusammenpassen; Ein Mausklick auf die Schaltfläche ist nur effektiv, wenn dies der Fall ist und schließt den Dialog)	Texte: Text: <code>OK</code>  Farben: Farbe und Alarmfarbe müssen unterschiedlich sein!  Farbvariablen: Farbe umschalten: <code>stPwd &lt;&gt; MUX(iLevel, stLevel0, stLevel1, stLevel2, stLevel3, stLevel4, stLevel5, stLevel6, stLevel7); (wenn Passwort und Level</code>

nicht zusammenpassen, ändert sich die Farbe nach Grau)

Zustandsvariablen: Eingaben deaktivieren: `stPwd <> MUX(iLevel, stLevel0, stLevel1, stLevel2, stLevel3, stLevel4, stLevel5, stLevel6, stLevel7)`; (wenn Passwort und Level nicht zusammenpassen, bleibt eine Eingabe auf die Schaltfläche ohne Wirkung)

Eingabekonfiguration: OnMouseDown: Dialog schließen: `ChangeUserLevel`, Ergebnis: OK; (zu beachten: die Eingaben auf den Dialog werden nicht auf die jeweiligen Variablen geschrieben, wenn die OK-Schaltfläche betätigt wird)

Schaltfläche, Cancel

Farben: Farbe = Farbe der OK-Schaltfläche

Texte: Text: Cancel

Eingabekonfiguration: OnMouseDown: Dialog schließen: `ChangeUserLevel`, Ergebnis: Cancel

Öffnen Sie den „Grundlegendes zum Schnittstelleneditor“ der Visualisierung und deklarieren folgende Variablen (die Werte der Eingangsvariablen können später in der "Dialog öffnen"-Eingabekonfiguration definiert werden, siehe „[Open Dialog](#)“ Eingabekonfiguration).

```
VAR_INPUT
  stTitle: STRING;
  stLevel0: STRING;
  stLevel1: STRING;
  stLevel2: STRING;
  stLevel3: STRING;
  stLevel4: STRING;
  stLevel5: STRING;
  stLevel6: STRING;
  stLevel7: STRING;
END_VAR
VAR_IN_OUT
  iLevel: INT;
  stPwd: STRING;
END_VAR
```

## 2. Erstellen einer Visualisierung "Visualization"

Nun wird eine Visualisierung `Visualization` mit einer Schaltfläche zum Öffnen des oben angelegten "ChangeUserLevel"-Dialogs und einer Anzeige des aktuell eingestellten Levels erstellt:

Im vorliegenden Beispiel werden die folgenden Elemente verwendet:

Element	Elementeigenschaften
Schaltfläche (zum Öffnen des "Change	<i>Texte</i> → <i>Text</i> : Change User Level

User Level" Dialogs

Eingabekonfiguration: OnMouseDown: Dialog öffnen: siehe Abbildung unten

Textfeld

*Texte* → *Text*: %s (Dialogüberschrift);

*Textvariablen* → *Text-Variable*: PLC\_PRG.iLevel (Zuweisung der Level-Eingabe zur Programmvariablen)

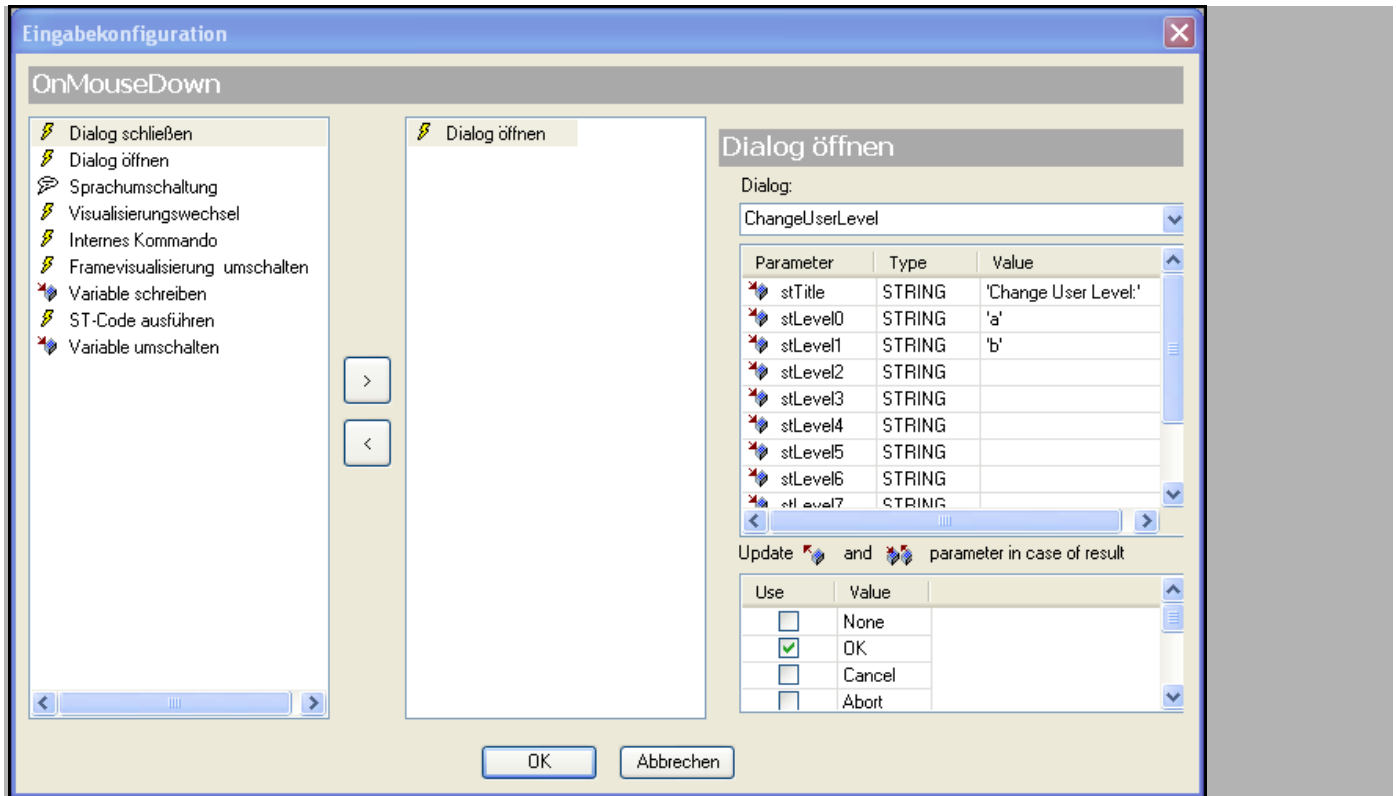


Abb. 22: Eingabekonfiguration der Schaltfläche, die die selbst-definierte Dialog-Visualisierung ChangeUserLevel öffnen soll

Hier muss der Dialog angegeben werden, der mit einem Mausklick auf die Schaltfläche geöffnet werden soll. Dazu wählen Sie "ChangeUserLevel" aus der Auswahlliste unterhalb von „Dialog“. Die Eingangs- (VAR\_INPUT) und Ein-/Ausgangsparameter (VAR\_IN\_OUT) der Dialog-Visualisierung (Schnittstelle von "ChangeUserLevel") werden aufgelistet und Sie können definieren, welche Werte durch den ChangeUserLevel-Dialog immer beim Öffnen gelesen werden sollen. Beachten Sie, dass für das aktuelle Beispiel - wie in der Abbildung oben zu sehen - nur zwei Level-Passwort-Kombinationen konfiguriert sind. Wenn der Anwender Level 0 eingibt, muss er "a" als Passwort eintippen, um die OK-Schaltfläche aktiviert zu bekommen. Wenn er Level 1 wählt, muss er "b" als Passwort verwenden. Wird einer der anderen Levels ausgewählt, muss momentan kein Passwort eingegeben werden.

Definieren Sie weiterhin, bei welcher Aktion im ChangeUserLevel-Dialog dieser Dialog geschlossen und seine Ausgänge und Ein-/Ausgänge geschrieben werden sollen. In unserem Beispiel setzen Sie einen Haken für die „Verwendung“ von „OK“. So wird später bei einem Mausklick auf die OK-Schaltfläche der ChangeUserLevel-Dialog geschlossen und die Variablen iLevel und stPwd (VAR\_IN\_OUT) werden mit den Werten aus der Benutzereingabe geschrieben.

Siehe für weitere Informationen zum "Dialog öffnen"-Konfigurationsdialog: [„Dialog öffnen“](#).

### 3. Online-Betrieb von "Visualization"

Wenn Sie nun die Visualisierung "Visualization" im Online-Modus laden und starten (möglich als Target-Visualisierung, Web-Visualisierung oder "Diagnose-Visualisierung"), dann kann der ChangeUserLevel-Dialog mit einem Mausklick auf die bezeichnete Schaltfläche in "Visualization" geöffnet werden:

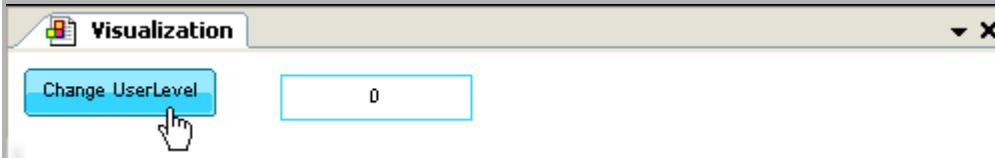


Abb. 23: Starte "Visualization", klicke auf Schaltfläche "Change UserLevel"

Der Dialog öffnet sich. Wählen Sie einen Level, z.B. "1" (wird in `iLevel` geschrieben) und geben Sie das passende Passwort "b" ein (wird in `stPwd` geschrieben). Diese Werte werden von der OK-Schaltfläche verglichen; solange sie nicht zusammenpassen, ist die Schaltfläche grau und kann nicht betätigt werden:

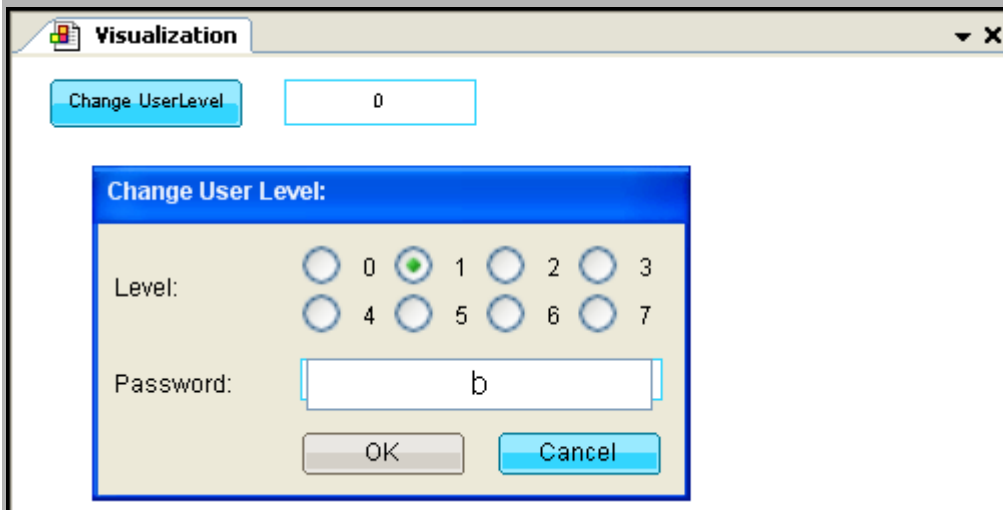


Abb. 24: Eingaben im Dialog "Change User Level"

Sie haben passende Werte eingegeben, somit wechselt OK nach blau und Sie können mit einem Mausklick darauf den Dialog schließen.

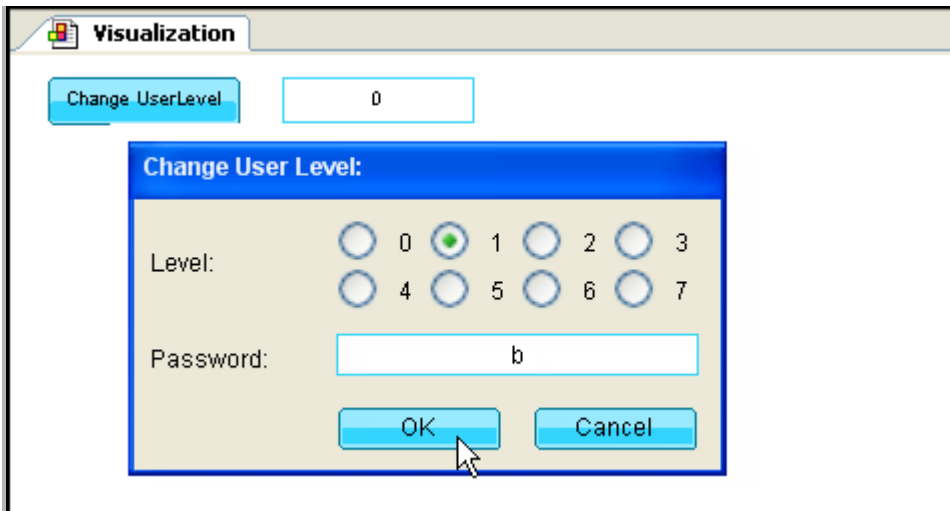


Abb. 25: OK zum Schließen des Dialogs verfügbar

Nun werden die Werte "tatsächlich" geschrieben (Beachten Sie, dass sie bisher nur auf dem Stack gehalten wurden!) und der geänderte Level ("1") wird angezeigt.

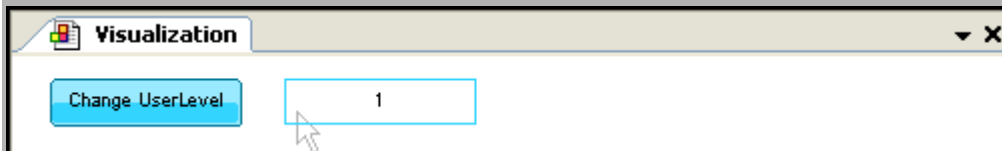


Abb. 26: Der neue Wert für den Benutzerlevel wird angezeigt

### 6.3.4.2 Programmatischer Zugriff auf Eingabedialoge

Programmatischer Zugriff auf einen „Eingabedialoge“ erstellt in einer CODESYS Visualisierung ist möglich:

Die in der **Schnittstelle** („Grundlegendes zum Schnittstelleneditor“) einer Dialogvisualisierung definierten Parameter werden automatisch in eine Struktur **<Dialogname>\_VISU\_STRUCT**. ("Dialogname" = Name der Visualisierung) geschrieben. Diese Struktur kann dann von der Applikation verwendet werden, z.B. über eine Funktion, deren Aufruf im Falle einer Benutzereingabe auf ein Visualisierungselement erfolgt (Konfiguration des Visualisierungselements). So kann das Öffnen eines Dialogs in einer Visualisierung und die Reaktion auf Benutzereingaben in diesem Dialog programmiert werden.

#### Dialogmanager

Alle Visualisierungen, die als „*Dialog*“ konfiguriert sind, werden automatisch instanziiert und vom internen Dialogmanager verwaltet. Dieser Manager kann über den ebenfalls intern arbeitenden Visualisierungsmanager angesprochen werden, indem die Methode `GetDialogManager()` aufgerufen wird. Der Dialogmanager stellt einige Methoden bereit, über die man die Applikation einen Visualisierungsdialog handhaben kann.

Die verfügbaren Methoden sind beschrieben in „Methoden für den Dialogmanager“.

Beispiel für die Verwendung des Login-Dialogs aus der Bibliothek `VisuDialog.library`

In diesem Beispiel wird ein Schaltflächenelement in einer Visualisierung so konfiguriert, dass bei einem Mausklick auf die Schaltfläche ein Login-Dialog geöffnet wird, in dem ein Benutzername und ein Passwort eingegeben werden können. Für diesen Fall wird ein Standard-Login-Dialog verwendet, der von der `VisuDialog.library` bereitgestellt wird. Aber auch ein selbsterstellter Dialog könnte in gleicher Weise gehandhabt werden:

Login

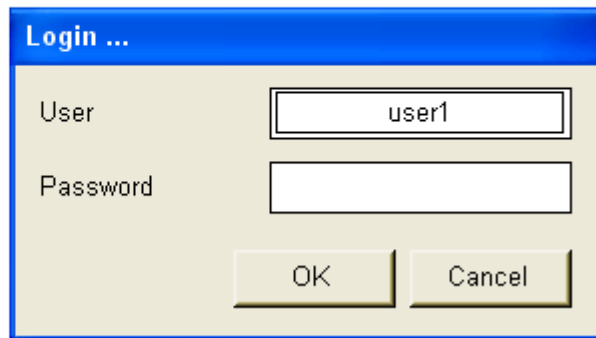


Abb. 27: Login-Dialog

Konfiguration der "Login" Schaltfläche:

Ein Schaltflächenelement wird in eine Visualisierung eingefügt. Es wird mit der Text-Eigenschaft "Login" und den folgenden zwei Eingabe-Eigenschaften konfiguriert:

1. OnMouseDown / ST-Code ausführen: Funktionsaufruf 'OpenLoginDialog(pClientData)'. Variable 'pClientData' muss an eine entsprechende Funktion 'OpenLoginDialog()' übergeben werden, die im Projekt vorhanden ist.

▲Abb. 28: Funktionsaufruf OpenLoginDialog()

2. OnDialogClosed / Dialog schließen / ST-Code ausführen: Funktionsaufruf 'OnLoginDialogClosed(pClientData)'

Die entsprechenden Funktionen 'OpenLoginDialog()' und 'OnLoginDialogClosed()' sind als Bausteine im Projekt vorhanden. Siehe im folgenden eine mögliche Implementierung:



## Implementierung der Funktion OpenLoginDialog()

Funktion `OpenLoginDialog()` wird aufgerufen wenn der Dialog geöffnet wird, z.B. bei einer `OnMouseDown`-Aktion auf das Schaltflächenelement.



ST

```
(* Kommentar zu CoDeSys Code Zeilen *)
FUNCTION OpenLoginDialog : bool
VAR_INPUT
    pClientData : POINTER TO VisuStructClientData;
END_VAR

VAR
    dialogMan : IDialogManager;
    loginDialog : IVisualisationDialog;
    pLoginInfo : POINTER TO Login_VISU_STRUCT; // Login_VISU_STRUCT enthält die Parameter, die in der Schnittstelle der
    Visualisierung "Login" definiert sind.
    result : Visu_DialogResult;
    stTitle : STRING := 'Login ...';
    stPasswordLabelText: STRING;
    stUserLabelText: STRING;
    stUsername: STRING;
END_VAR

dialogMan := g_VisuManager.GetDialogManager(); // Der Dialogmanager wird durch den implizit verfügbaren Visumanager
bereitgestellt
IF dialogMan <> 0 AND pClientData <> 0 THEN
    loginDialog := dialogMan.GetDialog('VisuDialogs.Login'); // Der zu öffnende Dialog wird angegeben
    IF loginDialog <> 0 THEN
        pLoginInfo := dialogMan.GetClientInterface(loginDialog, pClientData);
        IF pLoginInfo <> 0 THEN //im folgenden werden die Parameter des Login-Dialogs aus Login_VISU_STRUCT gelesen
            pLoginInfo^.stTitle := stTitle;
            pLoginInfo^.stPasswordLabelText := stPasswordLabelText;
            pLoginInfo^.stUserLabelText := stUserLabelText;
            dialogMan.OpenDialog(loginDialog, pClientData, TRUE, 0);
        END_IF
    END_IF
END_IF
```

## Implementierung der Funktion OnLoginDialogClosed()

OnLoginDialogClosed() definiert die Reaktion auf ein Schließen des Dialogs.



ST

(\* Kommentar zu CoDeSys Code Zeilen \*)

```
FUNCTION OnLoginDialogClosed : bool
VAR_INPUT
    pClientData : POINTER TO VisuStructClientData;
END_VAR

VAR
    dialogMan : IDialogManager;
    loginDialog : IVisualisationDialog;
    pLoginInfo : POINTER TO Login_VISU_STRUCT;
    result : Visu_DialogResult;
    stPassword: STRING;
    stUsername: STRING;
END_VAR

dialogMan := g_VisuManager.GetDialogManager(); // Der DialogManager wird vom implizit verfügbaren VisuManager bereitgestellt
IF dialogMan <> 0 AND pVisuClient <> 0 THEN
    loginDialog := dialogMan.GetDialog('VisuDialogs.Login'); // erkennt den Login-Dialog
    IF loginDialog <> 0 THEN
        result := loginDialog.GetResult(); // erkennt das Ergebnis (OK, Cancel) des Dialogs
        IF result = Visu_DialogResult.OK THEN
            loginDialog.SetResult(Visu_DialogResult.None); // Zurücksetzen auf Default (none)
            pLoginInfo := dialogMan.GetClientInterface(loginDialog, pVisuClient); // Struktur Login_VISU_STRUCT wird gelesen
            // im folgenden können die Strukturparameter gesetzt werden
            IF pLoginInfo <> 0 THEN
                stPassword := pLoginInfo^.stPassword; pLoginInfo^.stPassword := ''; // Passwort wird zurückgesetzt
                stUsername := pLoginInfo^.stUsername;
            END_IF
        ELSIF result = Visu_DialogResult.Cancel THEN
            loginDialog.SetResult(Visu_DialogResult.None);
            // Reaktion auf 'Abbrechen'
        ELSE
            //nichts zu tun
        END_IF
    END_IF
END_IF
END_IF
```

## Methoden für den Dialogmanager

**GetDialog(STRING stName):** Liefert den Dialog, der gerade durch ein Ereignis betroffen ist, als IVisualisationDialog:

METHOD GetDialog

Parameters:

Name	Type	Comment
GetDialog	VAR_OUTPUT	
stName	VAR_INPUT	

**GetClientInterface():** Liefert einen Zeiger auf die Dialogstruktur

METHOD GetClientInterface

Parameters:

Name	Type	Comment
GetClientInterface	VAR_OUTPUT	
dialog	VAR_INPUT	
pClient	VAR_INPUT	

**OpenDialog():** Öffnet den Dialog für den Client

METHOD OpenDialog

Parameters:

Name	Type	Comment
OpenDialog	VAR_OUTPUT	
dialog	VAR_INPUT	
pClient	VAR_INPUT	
bModal	VAR_INPUT	
pRect	VAR_INPUT	

**CloseDialog():** Schließt den Dialog für den Client

METHOD CloseDialog

Parameters:

Name	Type	Comment
CloseDialog	VAR_OUTPUT	
dialog	VAR_INPUT	
pClient	VAR_INPUT	

### 6.3.4.3 Ereignisse und Eingabeaktionen bemerken

Inhalt dieser Seite:

„Schließen eines Edit-Controls (Schreiben einer Variable) bemerken“

„Tastatur-Ereignisse bemerken“

Mit Hilfe von Methoden, die von den „Visualisierungsbibliotheken“ bereitgestellt werden, kann die Applikation bestimmte Ereignisse (Events), die durch Benutzereingaben in einer Visualisierung ausgelöst werden, wahrnehmen.

#### Schließen eines Edit-Controls (Schreiben einer Variable) bemerken

Die Applikation kann feststellen, wann ein "Edit Control" in einer zugehörigen Visualisierung geschlossen wird, d.h. wenn ein **Variablenwert durch eine Benutzereingabe in ein Eingabefeld** geschrieben wird. Dies kann nützlich sein, wenn eine weitere Aktion nach dem Schließen eines Edit-Controls ausgeführt werden soll.

Um über das Schließen informiert zu werden, kann eine Funktionsblock-Instanz (die die Schnittstelle VisuElems.IEditBoxInputHandler aus VisuElemBase.library implementiert) der globalen Instanz VisuElems.Visu\_Globals.g\_VisuEventManager zugewiesen werden, wobei die Methode SetEditBoxEventHandler dieser Instanz verwendet wird, wie im Beispiel unten gezeigt.

VisuElems.IEditBoxInputHandler benötigt die Methode VariableWritten, die aufgerufen wird, nachdem durch eine Benutzereingabe in einer Visualisierung ein Variablenwert geschrieben wurde.

Beispiel

Es gibt eine Visualisierung mit den folgenden Elementen:

1. Zwei Rechtecke, jedes mit einem Edit Control Feld (Eingabe "Variable schreiben"), in das der Benutzer einen Wert für Variable i oder j eingeben kann.

▲Abb. 29: Eingabe-Elemente

2. Ein weiteres Rechteck, in dem ein Text (Text Variable stInfo) mit einigen Informationen angezeigt wird, wenn eines der Eingabefelder in den oberen beiden Rechtecken geschlossen wird, d.h. wenn ein Wert geschrieben wird.

▲Abb. 30: Textausgabe-Element

Die Applikation enthält die folgenden Bausteine:

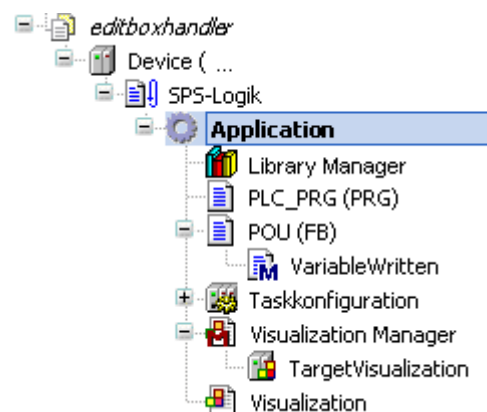


Abb. 31: Applikation mit Bausteinen



ST

```
PROGRAM PLC_PRG
VAR_INPUT
    i:INT; (* Variable, deren Wert der Benutzer über die Visualisierung schreiben kann *)
    j:REAL; (* Variable, deren Wert der Benutzer über die Visualisierung schreiben kann *)

    stInfo : STRING; (* Information zur Benutzereingabe über das Edit Control Feld; Text wird über die Methode VariableWritten
zusammengesetzt; Ergebnis wird im unteren Rechteck der Visualisierung dargestellt *) *)
END_VAR
VAR
    inst : POU;
    bFirst : BOOL := TRUE;
END_VAR

IF bFirst THEN
    bFirst := FALSE;
    VisuElems.Visu_Globals.g_VisuEventManager.SetEditBoxEventHandler(inst);
    (* Aufruf der Methode VariableWritten *)
END_IF

Function block POU
FUNCTION_BLOCK POU IMPLEMENTS VisuElems.IEditBoxInputHandler
```

(keine weiteren Deklarationen, kein Implementierungscode)

Methode 'VariableWritten', gehört zu POU:

METHOD VariableWritten : BOOL

(\* gibt Informationen, wenn ein Edit Control in der Visualisierung geschlossen wird, d.h. wenn eine Variable durch eine Benutzereingabe in einem der oberen Rechtecke geschrieben wird \*)

VAR\_INPUT

pVar : POINTER TO BYTE;

varType : VisuElems.Visu\_Types;

iMaxSize : INT;

pClient : POINTER TO VisuElems.VisuStructClientData;

END\_VAR

// String stInfo, der im unteren Rechteck dargestellt werden soll, wird hier zusammengesetzt

PLC\_PRG.stInfo := 'Variable written; type: ';

PLC\_PRG.stInfo := CONCAT(PLC\_PRG.stInfo, INT\_TO\_STRING(varType));

PLC\_PRG.stInfo := CONCAT(PLC\_PRG.stInfo, ' adr: ');

PLC\_PRG.stInfo := CONCAT(PLC\_PRG.stInfo, DWORD\_TO\_STRING(pVar));

PLC\_PRG.stInfo := CONCAT(PLC\_PRG.stInfo, ' by: ');

PLC\_PRG.stInfo := CONCAT(PLC\_PRG.stInfo, SEL(pClient^.globaldata.clienttype =  
VisuElems.Visu\_ClientType.Targetvisualization,'other visu', 'targetvisu'));

### 6.3.4.4 Tastatur-Ereignisse bemerken

Die Applikation kann Tastenereignisse (Tastatur-Event) in einer zugehörigen Visualisierung feststellen, d.h. sie bemerkt, wenn der Anwender eine **Taste drückt und wieder loslässt** wenn die Visualisierung aktiv ist.



Beachten Sie die Konfigurationsmöglichkeiten für die Tastaturbedienung von Visualisierungen

Um über solche Ereignisse informiert zu werden, kann ein Funktionsbaustein (der die Schnittstelle VisuElems.IVisuEventManager aus VisuElemBase.library implementiert) der globalen Instanz VisuElems.Visu\_Globals.g\_VisuEventManager zugewiesen werden und die Methode SetKeyEventHandler dieser Instanz wie im folgenden Beispiel verwendet werden.

#### Beispiel

Es gibt eine Visualisierung mit einem Visualisierungselement, das den Wert PLC\_PRG.stInfo als Text anzeigt.

Das heißt, im Online-Betrieb wird in diesem Element zur gerade ausgelösten Tastenaktion die Information "KeyEvent up: <dwKey>, key: <key id>, modifier: <dwModifiers>, by: <pClient^.globaldata.clienttype.....>" ausgegeben.

Beispiel: "KeyEvent up: TRUE, key: 75, modifier: 0, by: targetvisu".

Eigenschaft	Wert
Elementname	GenElemInst_2
Type of element	VISU_ST_ROUNDRECT
Position	
Center	
Colors	
Elementlook	
Texts	
Text	%s
Tooltip	
Text properties	
Absolute movement	
Relative movement	
Text variables	
Text Variable	PLC_PRG.stInfo
Tooltip Variable	
Dynamic texts	
Font variables	

Abb. 32: Textkonfiguration des Visualisierungselements

Die Applikation enthält die folgenden Bausteine:

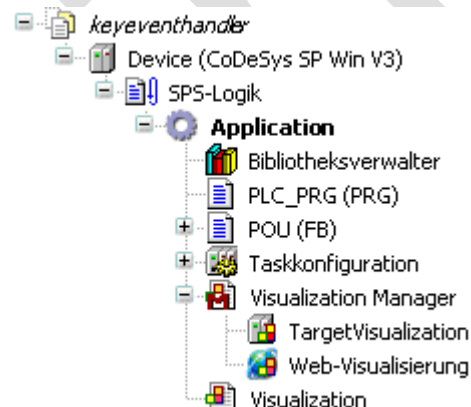


Abb. 33: Applikation mit POU's



ST

```
PROGRAM PLC_PRG
VAR_INPUT
    stInfo : STRING;
END_VAR
VAR
    inst : POU;
    bFirst : BOOL := TRUE;
END_VAR

IF bFirst THEN
    bFirst := FALSE;
    VisuElems.Visu_Globals.g_VisuEventManager.SetKeyEventHandler(inst);
END_IF

Dies führt dazu gemeldet zu bekommen, wenn ein Tastenereignis vorliegt.
Funktionsblock POU
FUNCTION_BLOCK POU IMPLEMENTS VisuElems.IKeyEventHandler
Keine weiteren Deklarationen, kein Implementierungs-Code

Die Schnittstelle VisuElems.IVisuEventManager benötigt eine Methode mit folgender Signatur, die mit aller verfügbaren
Information aufgerufen wird:
Method HandleKeyEvent, assigned to POU
/// Diese Methode wird aufgerufen, nachdem ein Tastatur-Event ausgelöst wurde.
/// RETURN:
/// TRUE - Wenn der EventHandler dieses Ereignis bearbeitet hat und es nicht von nicht von anderer Stelle behandelt werden sollte
/// FALSE - Wenn das Ereignis nicht von diesem Handler bearbeitet wird
METHOD HandleKeyEvent : BOOL
VAR_INPUT
    /// Event-Typ. Der Wert ist TRUE, wenn ein key-up Ereignis stattfand.
    bKeyUpEvent : BOOL;
    /// Key code
    dwKey : DWORD;
    /// Modifier. Mögliche Werte:
    /// VISU_KEYMOD_SHIFT : DWORD := 1;
    /// VISU_KEYMOD_ALT : DWORD := 2;
    /// VISU_KEYMOD_CTRL : DWORD := 4;
    dwModifiers : DWORD;
    /// Zeiger auf die Struktur des Clients wo das Event ausgelöst wurde
    pClient : POINTER TO VisuStructClientData;
END_VAR
VAR
END_VAR

PLC_PRG.stInfo := 'KeyEvent up: ';
PLC_PRG.stInfo := CONCAT(PLC_PRG.stInfo, BOOL_TO_STRING(bKeyUpEvent));
PLC_PRG.stInfo := CONCAT(PLC_PRG.stInfo, ' key: ');
PLC_PRG.stInfo := CONCAT(PLC_PRG.stInfo, DWORD_TO_STRING(dwKey));
PLC_PRG.stInfo := CONCAT(PLC_PRG.stInfo, ' modifier: ');
```

```

PLC_PRG.stInfo := CONCAT(PLC_PRG.stInfo, DWORD_TO_STRING(dwModifiers));
PLC_PRG.stInfo := CONCAT(PLC_PRG.stInfo, ' by: ');
PLC_PRG.stInfo := CONCAT(PLC_PRG.stInfo, SEL(pClient^.globaldata.clienttype =
VisuElems.Visu_ClientType.Targetvisualization, 'other visu', 'targetvisu'));

```

## 6.3.5 Visualisierungselemente

### 6.3.5.1 Benutzereingabe beim Editieren eines Elements

Was kann mit einem Visualisierungselement gemacht werden?

Im Visualisierungseditor können verschiedene Benutzereingaben (Mausbefehle oder Menübefehle) abhängig von der Position des Cursors im Editor und dem Zustand des Elements vorgenommen werden. Erlaubt die Cursorposition Benutzereingaben, dann ändert er seine Form um das dem Benutzer zu signalisieren.

Ein Element kann im Zustand "nicht selektiert" oder "selektiert" sein. Ein selektiertes Element wird visualisiert mit zusätzlichen kleinen Quadraten, die seine Position und Größe repräsentieren. Verwendete Befehle wirken auf dieses Element.

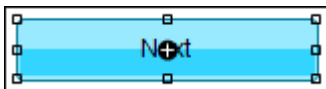





Abb. 34: Selektierter Button

#### Befehle

Abhängig vom Element und der Selektion im Editor stehen verschiedene Befehle zur Verfügung, die nicht ausgegraut sind. Detaillierte Information über Befehle ist unter [„Visualisierungsbefehle“](#) zu finden.

#### Editieren eines Visualisierungselements

Aktion im Editor	Auswirkung auf Eigenschaften
Ist ein Element fokussiert und ist der Cursor ein Handsymbol, wird es mit Klick auf das Element ausgewählt. Dann sind die Positions- und Größenrechtecke des Elements auf der Umrisslinie sichtbar.	Die Eigenschaften der aktuell selektierten Elements wird sofort angezeigt und kann in „ <i>Eigenschaften</i> “ editiert werden.
Ist ein Element fokussiert und wird der Cursor  , dann kann mit einem Klick auf das Element der Zeileneditor geöffnet und dort ein Name/Text eingegeben werden. Mit <a href="#">[ Eingabe ]</a> wird der Zeileneditor geschlossen.	„ <i>Text</i> “ unter Kategorie „ <i>Texte</i> “ enthält den eingegebenen Text.
Mit Fokus auf das Zentrum eines selektierten Elements ändert sich die Cursorform in  und das Zentrum kann mit gedrückter Maustaste auf eine beliebige X-/Y-Position gezogen werden.	„ <i>X</i> “ und „ <i>Y</i> “ unter Kategorie „ <i>Zentrum</i> “ werden aktualisiert.
Mit Fokus auf das selektierte Element ändert sich die Cursorform in  und das Element kann mit gedrückter Maustaste auf eine beliebige X-/Y-Position gezogen werden.	„ <i>X</i> “ und „ <i>Y</i> “ unter Kategorie „ <i>Position</i> “ werden aktualisiert.



Ist das Element selektiert und der Fokus auf eines der kleinen Quadrate auf der Umrisslinie, dann ändert sich die Cursorform und das Quadrat kann auf eine neue Größe gezogen werden. Wenn die Größe mit der Maus geändert wird, wird das Zentrum zurückgesetzt.

Wenn zusätzlich die [\[ Umschalttaste \]](#)-Taste gedrückt wird, wird die Breite und Höhe proportional zueinander geändert.

Wenn zusätzlich die [\[ Strg \]](#)-Taste gedrückt wird, wird die Breite und Höhe symmetrisch zum Zentrum geändert.

Wenn zusätzlich die [\[ Umschalttaste \]](#)-Taste und die [\[ Strg \]](#)-Taste gedrückt wird, wird die Größenänderung proportional und symmetrisch zum Zentrum ausgeführt.

Abhängig vom Elementtyp wird dessen Position und Größe auf unterschiedliche Weise unter Kategorie „[Position](#)“ gespeichert:

„X“, „Y“

„X“, „Y“, „Breite“, „Höhe“

Kategorie „[Punkte \[0\] .. \[<n>\]](#)“: „X“, „Y“

## Mehrfachauswahl

Mehrfachauswahl ist, wenn


[\[ Umschalttaste \]](#) gedrückt wird während Elemente per Mausklick ausgewählt oder abgewählt werden.

ein Rechteck mit gedrückter Maustaste um mehrere Elemente gezogen wird (Drag-and-drop).

↘ [„Alles selektieren“](#) verwendet wird.

Deselektion ist möglich mit ↘ [„Alles deselektieren“](#).

Alternativ können Elemente in der ↘ [„Elementliste“](#) mit Hilfe von [\[ Umschalttaste \]](#) und [\[ Strg \]](#) selektiert werden.

	Sind mehrere Elemente selektiert, dann wirken Änderungen in „ <a href="#">Eigenschaften</a> “ auf alle selektierten Elemente.
---	---

## Verlagern der Selektion mit Tastatur

Ist ein Element selektiert, kann die Selektion zum nächsten Element gemäß der Reihenfolge beim Einfügen (Z-Reihenfolge) verlagert werden, indem [\[ Tabulator \]](#) gedrückt wird. Wird [\[ Umschalttaste \]](#) + [\[ Tabulator \]](#) gedrückt, wird die Selektion in umgekehrter Reihenfolge verlagert.

## Elemente gruppieren

Mehrfach ausgewählte Elemente können mit dem Befehl ↘ [„Gruppieren“](#) gruppiert werden. Die selektierten Elemente sind in einer Gruppe festgesetzt, deren Unterelemente dann über ↘ [„Elementliste“](#) selektiert werden können. Mit ↘ [„Gruppierung aufheben“](#) wird die Gruppe wieder in seine Elemente zerlegt.

## Ändern der Reihenfolge

Um festzulegen, wo die Elemente auf der Z-Achse des Visualisierungsbereichs, das ist im Vordergrund, im Hintergrund oder irgendwo dazwischen, positioniert sein sollen, stehen Befehle unter [Visualisierung → Reihenfolge](#) zur Verfügung.

## Ausrichtung der Elemente

Um zwei oder mehr Elemente auszurichten, stehen Befehle unter [Visualisierung → Ausrichtung](#) zur Verfügung.

## Text und Sprache in Visualisierungen

Text kann einem Visualisierungselement durch eine geeignete Definition im Eigenschafteneditor oder durch direkte Eingabe im Visualisierungseditor zugewiesen werden. Für letzteres kann am selektierten Element mit der [ [Leertaste](#) ] ein Eingabefeld geöffnet werden. Siehe dazu auch [„Benutzereingabe beim Editieren eines Elements“](#).

**Textlisten** werden verwendet, um solche Texte zu verwalten. Eine Textliste ist eine POU, die Text-Strings enthält, die durch die Kombination von Textlisten-Name und einer listeninternen Text ID innerhalb des Projekts eindeutig referenziert werden können. Verschiedene Sprachversionen (Landessprache) eines Texts können innerhalb einer Textliste definiert sein und somit ist jeder Text zusätzlich noch durch ein Sprachenkürzel gekennzeichnet. Zumindest die Standard-Sprache muss für jeden Text definiert sein.

Siehe unter Kapitel [Text List](#) nach detaillierten Informationen zur Handhabung von Textlisten.



Verzeichnisse, die Textlisten für die Verwendung in der Visualisierung bereitstellen, sind in den [Projekteinstellungen, Kategorie Visualisierung](#) definiert.

### Textarten

In den Texteigenschaften eines Visualisierungselements müssen der Textlistenname und die Text-ID eingetragen werden (String-Format), um zu definieren, welcher Text im Element im Online-Modus angezeigt wird. Zusätzlich kann die Darstellung des Textes durch die Verwendung von [„Textformatierung“](#) beeinflusst werden. Da die Text-ID auch über eine Variable angegeben werden kann, wird ein dynamisches Umschalten von Texten möglich.

Wenn die aktuell verwendete Textliste mehrere Landessprachenversionen eines Textes definiert, kann die in der Visualisierung zu verwendende Sprache mit Hilfe eines Visualisierungselements umgeschaltet werden (Sprachumschaltung), das entsprechend konfiguriert ist. Dazu muss es mit einer [„Eingabekonfiguration“](#) versehen sein, die eine Mausektion zur "Sprachumschaltung" definiert.

Somit sind eine dynamische Anzeige und Sprachumschaltung für Visualisierungstexte möglich.

Grundsätzlich gibt es zwei Typen von Text in einer Visualisierung:

#### 1. Statische Texte:

Statischer Text kann im Online-Modus nicht verändert werden (nur die Landessprache wird, wenn in der Textliste entsprechende Sprachversionen definiert sind, immer automatisch derjenigen angepasst, die gerade in der Visualisierung verwendet wird). Statischer Text wird in der Eigenschaft **„Texte“** konfiguriert und wird in einer automatisch angelegten Textliste 'GlobalTextList' verwaltet.

#### 2. Dynamische Texte

Dynamischer Text kann im Online Modus durch eine Benutzereingabe oder über eine Variable der Applikation verändert werden. Er wird in der Eigenschaft **„Dynamische Texte“** konfiguriert. Dabei muss der Name der dynamischen Textliste und die Text ID des gewünschten Texts angegeben werden.

## Textformatierung

Neben der Eingabe des reinen Textes in der Elementkonfiguration können auch Formatierungsangaben eingefügt werden, um die Textanzeige im Online-Betrieb zu gestalten. Eine Formatierungsangabe beginnt immer mit einem "%", gefolgt von einem Zeichen, das die Art der Formatierung bestimmt. Sie kann einzeln oder in Kombination mit

dem eigentlichen Text verwendet werden.

Wenn in der Textangabe ein "%s" enthalten ist, wird dieses im Online-Modus durch den Wert der Variable ersetzt, die im Eigenschaften-Feld „*Textvariable*“ angegeben ist. Falls Sie den Instanznamen einer Variablen, die einem Visualisierungsbaustein als Eingangsparameter übergeben wurde, anzeigen lassen wollen, so ist vom dem Pragmaattribut 'parameterstringof' Gebrauch zu machen.

Beachten Sie, dass jede Formatierungsangabe verwendet werden kann, die der Standard-C-Bibliotheksfunktion 'sprintf' entspricht, wenn sie zum Datentyp der verwendeten Variable passt.

#### Beispiele von Formatierungsangaben und ihren Argumenten:

Zeichen nach "%"	Argument/Ausgabe als
d, i	Dezimalzahl
b	Binärzahl
o	Vorzeichenlose Oktalzahl (ohne vorangestellte Null)
x	Vorzeichenlose Hexadezimalzahl (ohne vorangestelltes "0x")
u	Vorzeichenlose Dezimalzahl
c	Einzelnes Zeichen
s	Zeichenfolge: Dieses wird im Online-Betrieb durch den Wert der Variablen ersetzt, die im Eigenschaften-Feld <i>Textvariablen</i> → <i>Textvariable</i> angegeben ist.
f	REAL-Werte; Syntax: % <Ausrichtung><Mindestbreite><Genauigkeit>

Ausrichtung wird durch Minuszeichen (linksbündig) bzw. Pluszeichen (rechtsbündig, Default) definiert; Genauigkeit definiert die Anzahl der Stellen hinter dem Komma (Default: 6). Beispiel siehe unten.

#### Beispiel

Eintrag im Eigenschafteneditor in „*Text*“:

```
Fill level %2.5f mm
```

Eintrag in „*Textvariable*“ (REAL-Variable zur Angabe des Füllstandes):

```
plc_prg.fvar1
```

-> Ausgabe im Online-Betrieb:

```
Fill level 32.89999 mm
```



Wenn ein Prozentzeichen % in Kombination mit einer der oben beschriebenen Formatierungsangaben angezeigt werden soll, muss "%" eingegeben werden. Beispiel: Geben Sie "Rate in %: %s" ein, um im Online-Betrieb folgendes zu erhalten: "Rate in %: 12" (wenn die Textvariable gerade "12" liefert).

Die Schriftart, die im Online-Betrieb verwendet wird, kann auch im Eigenschafteneditor verwendet werden. Siehe dazu die die Eigenschaften „[Texteigenschaften](#)“ und „[Fontvariablen](#).“

Die horizontale und vertikale Ausrichtung eines Textes ist im Eigenschafteneditor der Elemente definiert.

### Unterstützung des Unicode-Formats

Diese kann aktiviert werden. Siehe dazu [Text List](#).

### Ausgabe der Systemzeit

Eine Kombination aus "%t" und nachfolgenden speziellen Platzhaltern innerhalb von eckigen Klammern wird im Online-Modus durch die aktuelle Systemzeit ersetzt. Die Platzhalter definieren das Anzeigeformat, siehe untenstehende Tabelle.



Import von V2.3 Projekten: Die in CODESYS V2.3 verwendete Zeitformatierung %t wird automatisch in das neue %t[] Format konvertiert, wenn ein altes Projekt importiert wird, jedoch werden die folgenden Platzhalter nicht mehr unterstützt: %U, %W, %z, %Z.

### Gültige Platzhalter:

ddd	Name des Wochentags, abgekürzt, z.B. "Wed"
dddd	Name des Wochentags, z.B. "Wednesday"
dddd	Wochentag als Zahl (0 – 6; Sonntag ist 0)
MMM	Name des Monats, abgekürzt, z.B. "Feb"
MMMM	Name des Monats, z.B. "February"
d	Tag im Monat als Zahl (1 – 31), z.B. "8"
dd	Tag im Monat als Zahl (01 – 31), z.B. "08"
M	Monat als Zahl (1 – 12), z.B. "4"
MM	Monat als Zahl (01 – 12), z.B. "04"
jjj	Tag im Jahr als Zahl (001-366), z.B. "067"
y	Jahr ohne Angabe des Jahrhunderts (0-99), z.B. "9"
yy	Jahr ohne Angabe des Jahrhunderts (00-99), z.B. "09"
yyy	Jahr mit Angabe des Jahrhunderts, z.B. "2009"
HH	Stunde, 24-Stunden-Format (01-24), z.B. "16"
hh	Stunde, 12-Stunden-Format (01-12), z.B. "8" für 16 Uhr
m	Minuten (0-59), ohne vorangestellte Null, z.B. "6"
mm	Minuten (00-59), mit vorangestellter Null, z.B. "06"
s	Sekunden (0-59), ohne vorangestellte Null, z.B. "6"
ss	Secunden (00-59), mit vorangestellter Null, z.B. "06"

- ms      Milliseconds (0-999), without preceded null, for example "322"
- t        Kennzeichner für die Anzeige im 12-Stunden-Format: A (Stunden <12) bzw. P (Stunden >12), z.B. "A" wenn es 9 Uhr morgens ist
- tt      Kennzeichner für die Anzeige im 12-Stunden-Format: AM (Stunden <12) bzw. PM (Stunden >12), z.B. "AM" wenn es 9 Uhr morgens ist
- ''      Zeichenfolgen, die einen der oben aufgeführten Platzhalter enthalten, müssen in einzelne Hochkommas gefaßt werden; alle anderen Texte innerhalb des Format-Strings können ohne Hochkommas stehen; z.B. 'update', weil ein "d" und ein "t" enthalten sind

### Beispiel

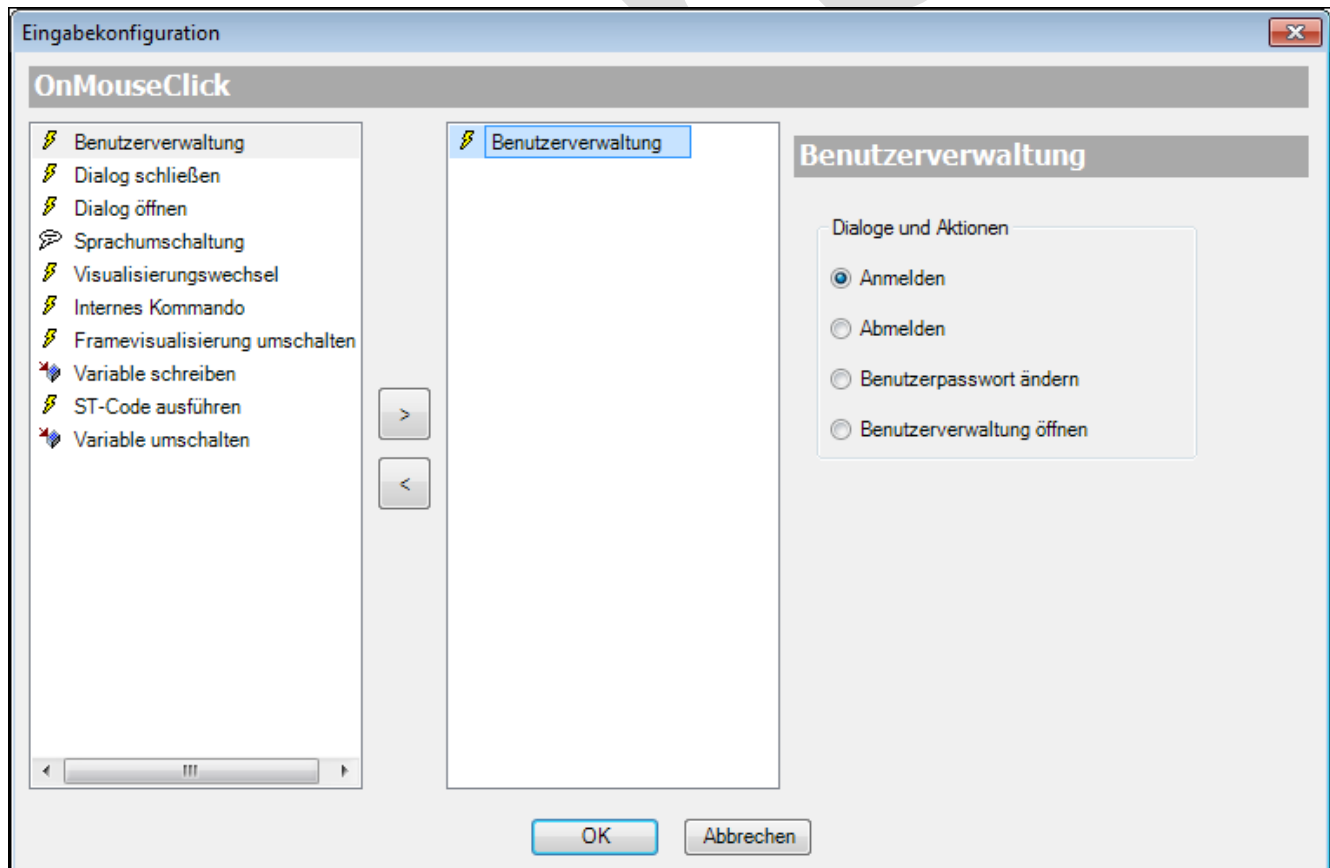
```
%t['Last update:' ddd MMM dd.MM.yy 'at' HH:mm:ss]
```

-> Anzeige im Online-Modus: Last update: Wed Aug 28.08.02 at 16:32:45

### Eingabekonfiguration

Eine oder mehrere der im Folgenden beschriebenen Nachfolgeaktionen kann im Dialog „[Eingabekonfiguration](#)“ für ein bestimmtes Ereignis (z.B. OnMouseClicked) des Elements definiert werden. Der Dialog öffnet auf einen Mausklick in das Wertefeld dieses Ereignisses im „[Eigenschaften](#)“-Fenster. Der Name des Ereignisses, für die diese Nachfolgeaktionen selektiert werden, ist als Titel in der grauen Zeile sichtbar.

### Benutzerverwaltung



▼ Abb. 35: Eingabekonfiguration Benutzerverwaltung

Hier können Sie definieren, dass Dialoge oder Aktionen auf ein selektiertes Ereignis folgen. Damit ist es Visualisierungsbenutzern im Online-Modus erlaubt, eine der aufgelisteten Benutzerschnittstellen/Aktionen so zu aktivieren wie sie in der Bibliothek VisuUserManagement zur Verfügung gestellt werden. Die Dialoge werden in Reiter [„Visualisierungen“](#) aufgelistet, wenn sie verfügbar sind.

*„Dialog and actions“*

*„Login“*: Dialog VUM\_Login wird sich öffnen.

*„Logout“*: Ein Abmeldeverfahren wird ausgeführt.

*„Change user password“*: Dialog VUM\_ChangePassword wird sich öffnen.

*„Open user configuration“*: Dialog VUM\_UserManagement wird sich öffnen.

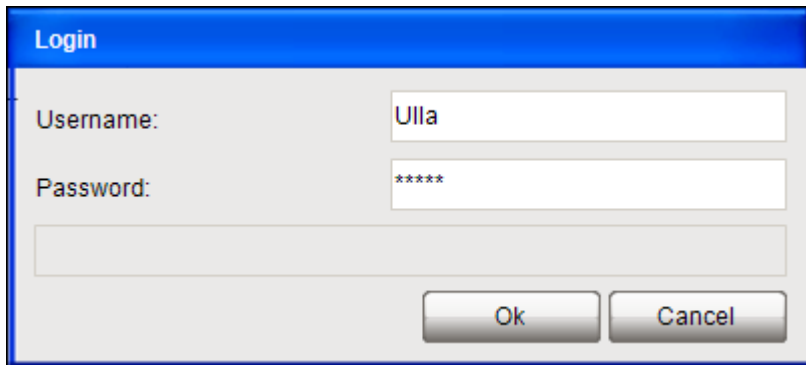


Abb. 36: VUM\_Login

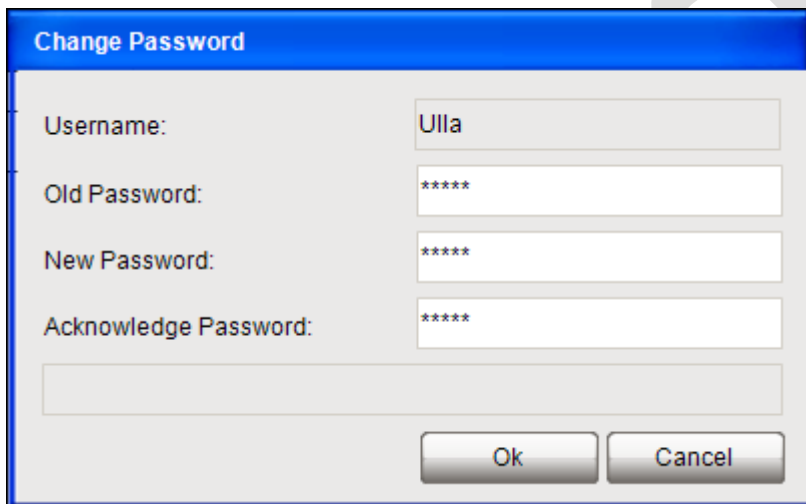


Abb. 37: VUM\_ChangePasword

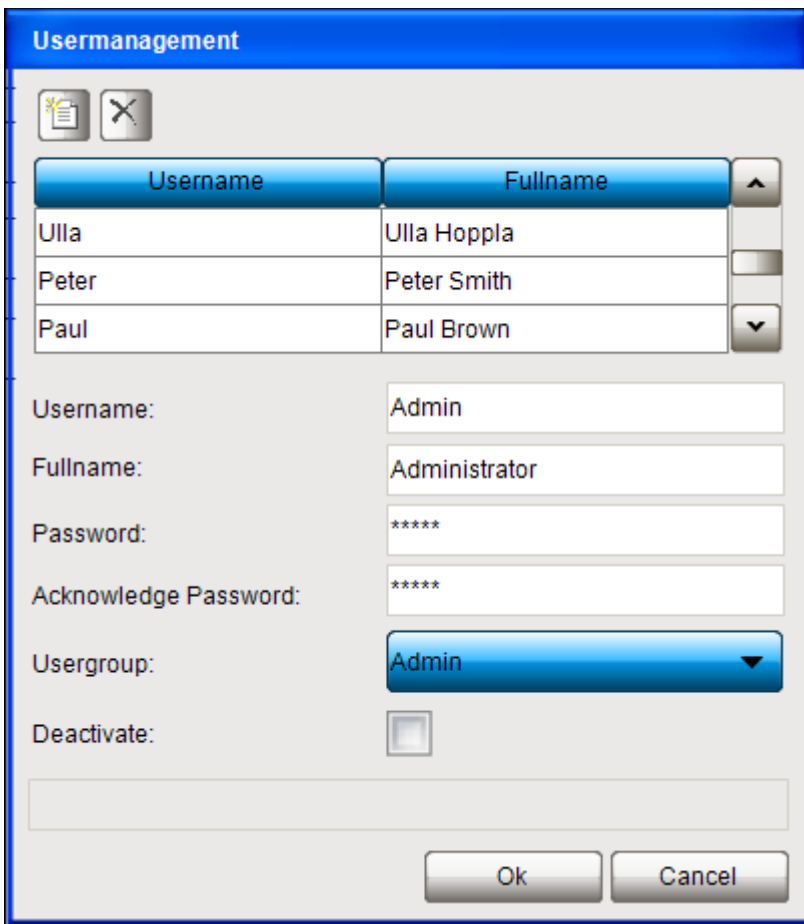
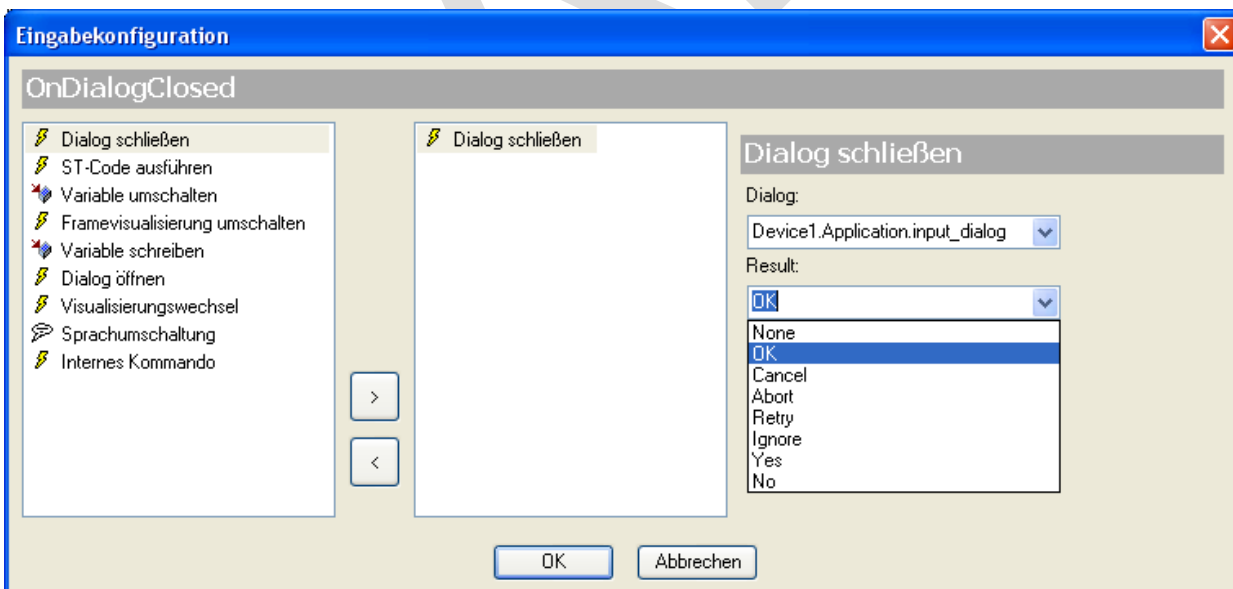


Abb. 38: VUM\_UserManagement

### Dialog schließen




▼ Abb. 39: Dialog 'Eingabekfiguration', 'Dialog schließen' selektiert

Hier kann festgelegt werden, dass auf das Ereignis hin der angegebene Dialog mit dem angegebenen Resultat geschlossen wird. Wählen Sie den gewünschten Dialog von der Auswahlliste, die alle momentan verfügbaren [„Eingabedialoge“](#) anbietet.

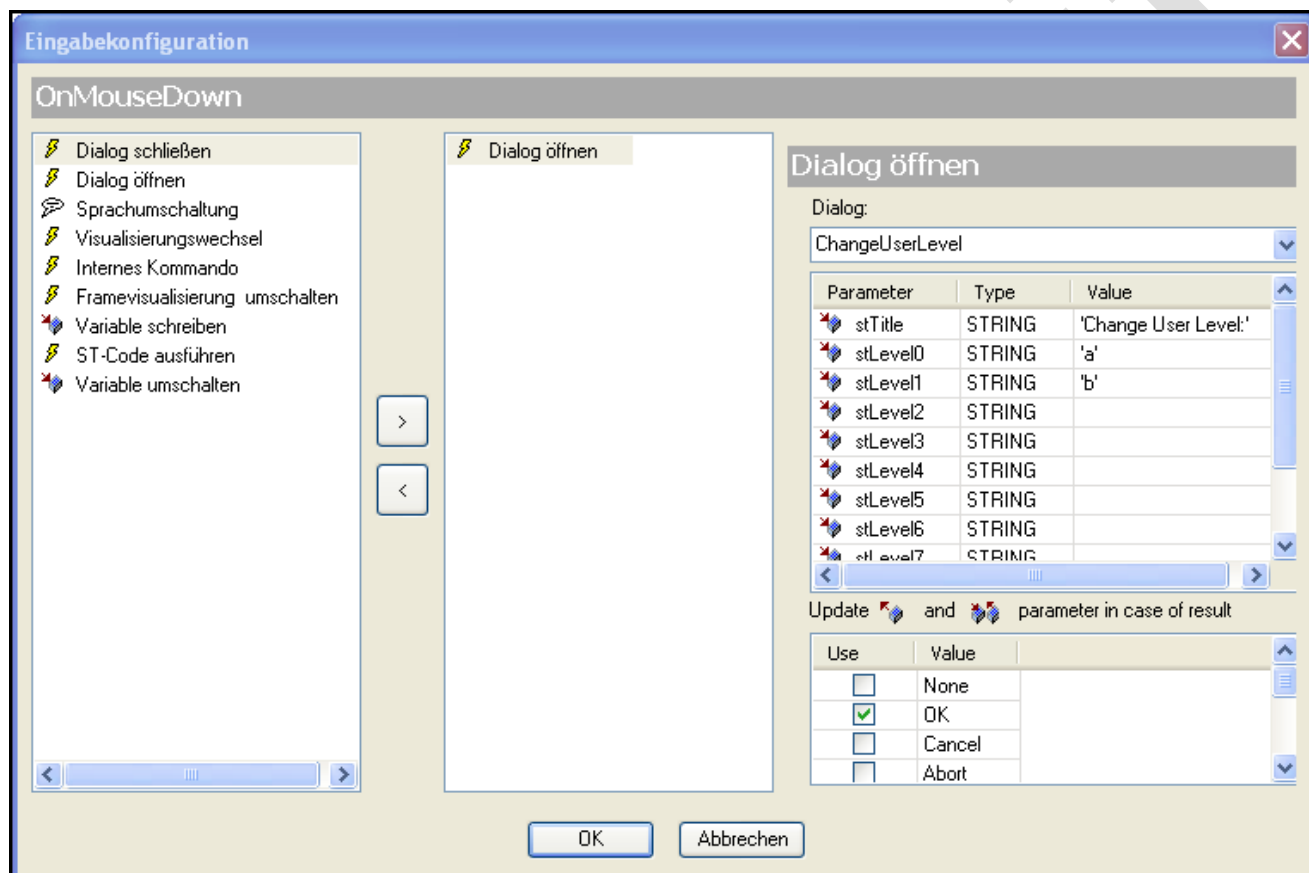
„Dialog“: Auswählen eines Dialogs aus der Auswahlliste.

„**Ergebnis**“: Die Liste bietet die Standardoptionen, die in Dialogen verwendet werden, an, in denen eine Benutzerreaktion gefordert wird: OK, Cancel, Abort, Retry, Ignore, Yes, No.

	Beachten Sie, dass das Resultat, das vom zuletzt geschlossenen Dialog stammt, abgerufen werden kann und dass eine entsprechende Reaktion darauf in einem beliebigen Element derselben Visualisierung konfiguriert werden kann. Verwenden Sie dazu die Konfigurationsoption „ <b>OnDialogClosed</b> “.
---	---

Siehe [„Eingabedialoge“](#) für mehr Information zu den Standard- und benutzerdefinierten Eingabedialogen in Visualisierungen.

## Dialog öffnen



▼ Abb. 40: Dialog 'Eingabekonfiguration', 'Dialog öffnen' selektiert

Hier können Sie definieren, dass auf die Mausaktion hin ein Dialog geöffnet werden soll, der durch eine an andere (Standard- oder vom Benutzer erstellte) Visualisierung repräsentiert wird. Die Auswahlliste bietet alle Visualisierungen an, die in ihren [„Eigenschaften eines Visualisierungsobjekts“](#) den Verwendungszweck "Dialog" eingetragen haben. Somit kann ein selbst-erstellter Dialog als Benutzereingabemaske in einer Visualisierung verwendet werden.



Zumindest das Standardobjekt "VisuDialogs.Login" und "VisuDialogs.FileOpenSave" ist verfügbar, wenn **VisuDialogs.library** im Projekt eingebunden ist.

Für die ausgewählte Dialogvisualisierung werden die Eingangs- (VAR\_INPUT) und Ein-/Ausgangsparameter (VAR\_IN\_OUT) wie in [„Grundlegendes zum Schnittstelleneditor“](#) definiert dargestellt („**Parameter**“, „**Typ**“, „**Wert**“).

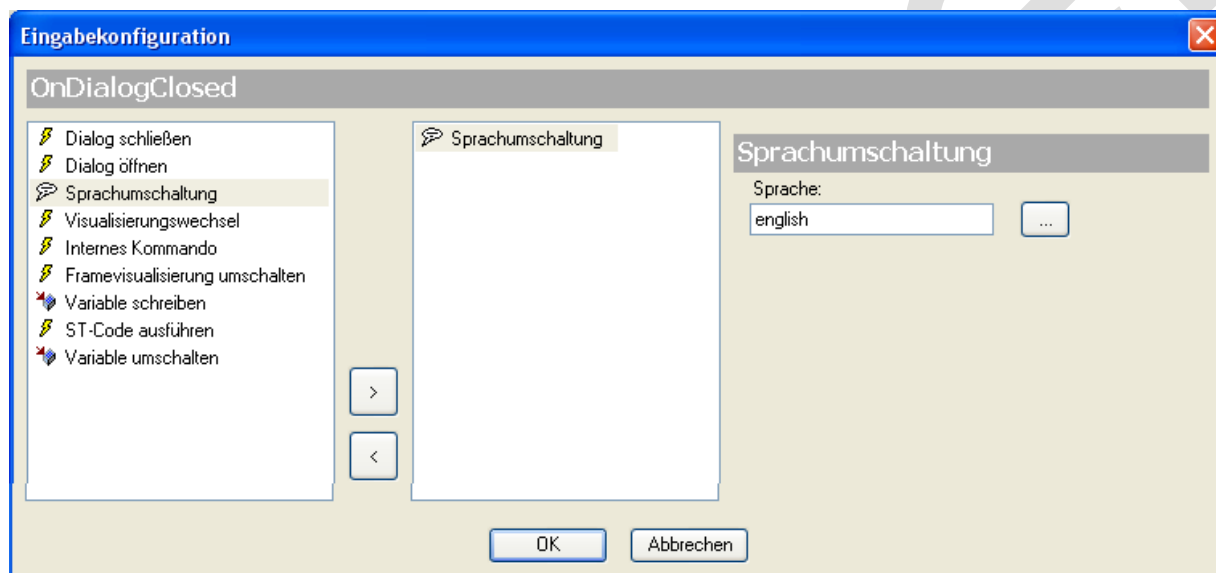
Die Standardwerte (=0) können durch visualisierungsspezifische Werte ersetzt werden, die immer dann in die Dialogvisualisierung geschrieben werden, wenn der Dialog geöffnet wird.




Außerdem muss angegeben werden, auf welches „*Ergebnis*“ der Dialogvisualisierung hin ihre Ausgangs- und Ein-/Ausgangsparameter geschrieben werden sollen: Setzen Sie einen Haken in der „*Verwenden*“-Spalte des gewünschten Ergebnis„*Werts*“. Mögliche Werte: Keinen (None), OK, Abbrechen (Cancel), Komplet abbrechen (Abort), Erneut versuchen (Retry), Ignorieren (Ignore), Ja (Yes), Nein (No).

	Im Unterschied zu früheren Versionen der Visualisierung ist es für den Standardgebrauch einer Dialogvisualisierung nicht nötig, eine „ <i>OnDialogClosed</i> “-Aktion zu definieren, um einen Dialog geschlossen zu bekommen. Jedoch steht die Aktion weiterhin als Möglichkeit für spezielle Lösungen zur Verfügung.
	<b>HINWEIS!</b> Man beachte, dass die Ausgangs- und EinAusgangsparameter einer Dialog-Visualisierung nicht geschrieben werden, bevor der Dialog geschlossen wird! Bis dahin werden die Werte nur auf dem Stack gespeichert, d.h. werden nicht als Referenzen sondern als Kopien gehandhabt.

## Sprachumschaltung



▼ Abb. 41: 'Eingabekonfiguration' , 'Sprachumschaltung' selektiert

Hier kann eine Landessprache angegeben werden, die auf die Mausaktion hin für die Darstellung der Visualisierungstexte verwendet wird. Verwenden Sie dabei das Sprachkürzel, das in der/den zugehörigen Textlisten verwendet wird. Alternativ können Sie auf die Taste  klicken, um die gewünschte Sprache im Eingabeassistenten auszuwählen.

## Visualisierungswechsel

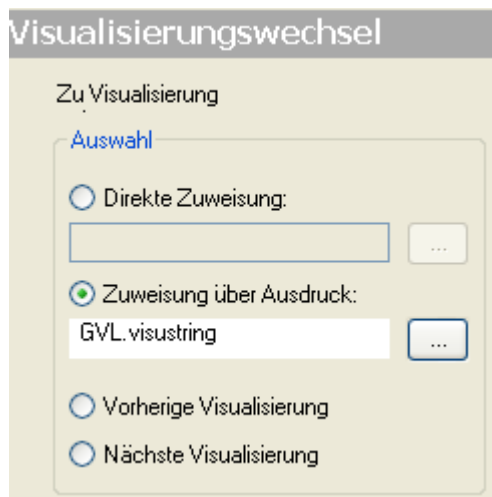


Abb. 42: 'Eingabekonfiguration', 'Visualisierungswechsel' selektiert

Wählen Sie eine der folgenden Optionen zur Festlegung, welche Visualisierung auf die Mausaktion im Online-Modus hin angezeigt werden soll (Auswahl):

„**Direkte Zuweisung**“: Die Visualisierung kann direkt eingegeben werden. Der gesamte Pfad <Gerätename>. <Applikationsname>.<Visualisierungsname> muss eingetragen werden. Dazu wird am besten die Eingabehilfe verwendet, die über Schaltfläche  geöffnet werden kann. Beispiel: "MyPlc.Appl1.Visu\_xy"

„**Zuweisung über Ausdruck**“: Hier kann eine Variable vom Typ STRING angegeben werden, die von der Applikation verwendet wird und die den Namen der Visualisierung liefert. Beispiel: "visu\_stringvar: STRING := 'MyPlc.Appl1.Visu\_xy' "

Die Abfolge, in der Visualisierungen nacheinander über Benutzereingaben angezeigt wurden, wird intern gespeichert. Mit den folgenden beiden Optionen können diese Informationen genutzt werden:

„**Vorherige Visualisierung**“: Die zuvor angezeigte Visualisierung wird wieder angezeigt. Wenn zuvor noch keine aufgerufen war, wird weiterhin die aktuelle gezeigt.

„**Nächste Visualisierung**“: Die Visualisierung, die in der aufgezeichneten Abfolge der Visualisierungswechsel hinter der aktuellen steht, wird angezeigt. Dies ist also nur möglich, wenn aus dieser vorher bereits ein Visualisierungswechsel über 'Vorherige Visualisierung' vorgenommen wurde.

### Internes Kommando

Ausführen von Programmen, Drucken, und Kommandos des Rezeptverwalters.

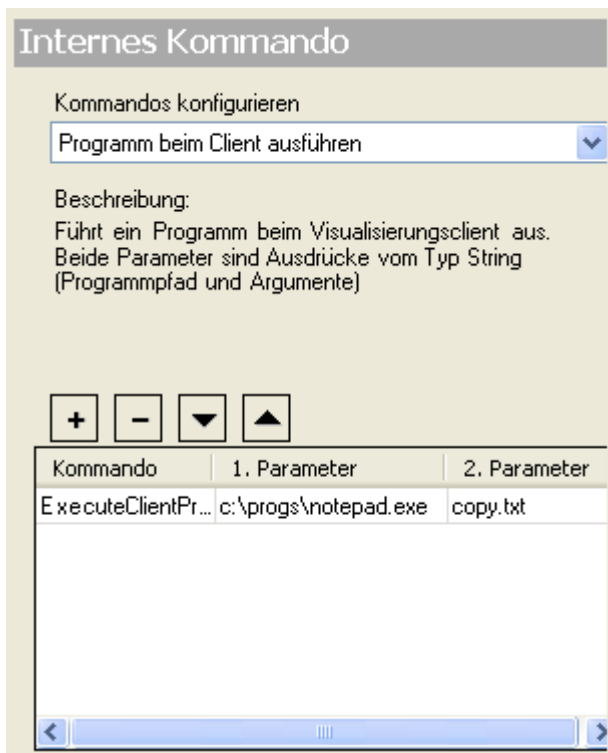






Abb. 43: 'Eingabekonfiguration', 'Internes Kommando' selektiert

Hier können ein oder mehrere Kommandos definiert werden, die auf die Mausaktion hin ausgeführt werden sollen.

Wählen sie aus der Liste „*Kommandos konfigurieren*“ einen Befehl aus und betätigen Sie die  Schaltfläche, um ihn in der Tabelle im unteren Teil des Dialogs einzufügen. Diese Tabelle enthält alle aktuell für die aktuelle Eingabekonfiguration ausgewählten Befehle.

Eine kurze Beschreibung zum oben in der Liste ausgewählten Befehl wird immer im Mittelteil des Dialogs angezeigt. Ergänzen Sie dementsprechend die Parameter des Befehls unten in den Tabellenspalten „*1st Parameter*“ und „*2nd Parameter*“. (Spalte **Kommando** zeigt den internen Kommandonamen.) Sehen Sie hierzu auch weiter unten auf dieser Hilfeseite die Beschreibung der einzelnen Kommandos.

Mit der Schaltfläche  kann der gerade ausgewählte Eintrag von der Liste entfernt werden. Die konfigurierten Befehle werden später, wenn die Benutzereingabe auf das Visualisierungselement erfolgt, entsprechend Ihrer Anordnung in der Tabelle von oben nach unten nacheinander ausgeführt. Um die Reihenfolge zu verändern, können die Einträge mit den Schaltflächen  und  verschoben werden.

Die folgenden Kommandos stehen zur Auswahl:

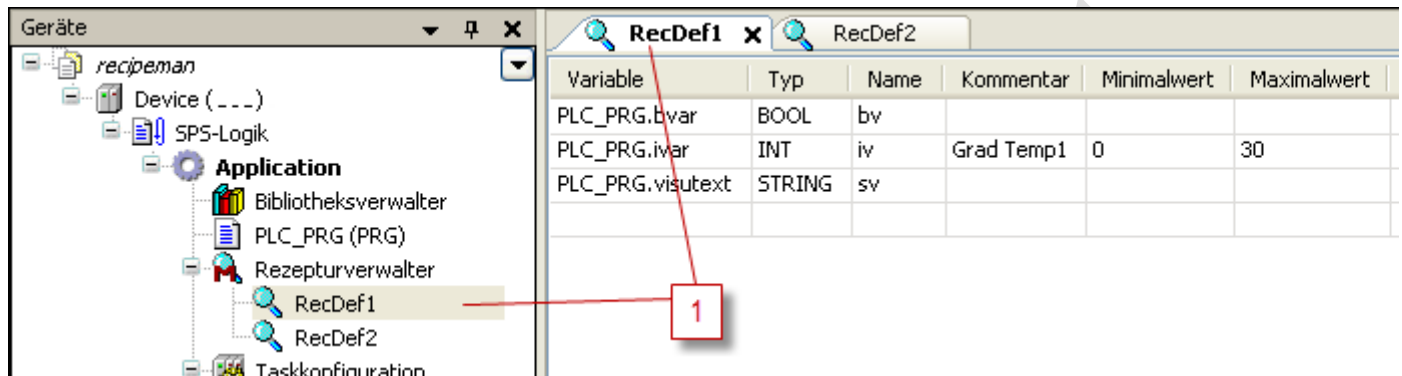
- |                                      |  |
|--------------------------------------|--|
| Programm auf der Steuerung ausführen | Das angegebene Programm (*.exe) wird auf der Steuerung bzw. beim Visualisierungs-Client ausgeführt.  |
| Programm beim Client ausführen       | 1. Parameter: Zeichenfolge (string), Pfad der Programmdatei (Beispiel: "C:\programs\notepad.exe")<br>2. Parameter: Zeichenfolge (string), Argumente für das auszuführende Programm, z.B. Name einer Datei, die das Programm öffnen soll (Beispiel: "copyfile.txt") |
| Drucken                              | Der Standarddialog „ <i>Drucken</i> “ wird geöffnet, wo Einstellungen zum Einstellen des Druckebereichs und der Druckerparameter etc. vorgenommen werden können. Darüber kann die aktuelle Visualisierung ausgedruckt werden.                                      |

Dieser Befehl wird nur innerhalb der Target-Visualisierung unter Windows unterstützt.

Für die Kommandos zur Rezepturverwaltung ist folgendes zu beachten:

Die **Rezepturdefinition** ist ein Objekt unterhalb des Rezepturverwalters im Gerätebaum des Projekts. Sie besteht aus einer Liste von Variablen und den für diese Variablen vorhandenen Rezepturen (d.h. Variablenwerten). Sie kann im Editorfenster des Rezepturverwalters geöffnet und bearbeitet werden.

Der **Rezepturname** ist der Name einer in der Rezepturdefinition beschriebenen Rezeptur für die jeweilige Liste von Variablen. Die Rezepturnamen sind dort als Tabellenüberschriften eingetragen.



1 Rezepturdefinition Name

2 Rezepturnamen

Abb. 44: Rezepturdefinition, Rezepturname

**Rezeptur erstellen** Eine neue Rezeptur wird in der angegebenen Rezepturdefinition angelegt.

1. Parameter: Name der Rezepturdefinition
2. Parameter: Name der neuen Rezeptur

**Rezeptur lesen** Die aktuellen Werte der Variablen der angegebenen Rezepturdefinition werden von der Steuerung gelesen und in die angegebene Rezeptur geschrieben. Dabei werden die Werte implizit gespeichert (in einer Datei auf der Steuerung) und gleichzeitig in der Tabelle der Rezepturdefinition im CODESYS Rezepturverwalter angezeigt. In anderen Worten: Die in CODESYS verwaltete Rezeptur wird mit den Werten aus der Steuerung aktualisiert.

1. Parameter: Name der Rezepturdefinition
2. Parameter: Name der Rezeptur, für die die Werte von der Steuerung gelesen und als neue Rezepturwerte gespeichert werden sollen.

**Rezeptur schreiben** Die Werte der angegebenen Rezeptur, wie sie im Rezepturverwalter vorliegen, werden auf die entsprechenden Variablen in der Steuerung geschrieben.

1. Parameter: Name der Rezepturdefinition
2. Parameter: Name der Rezeptur aus dieser Rezepturdefinition, deren Werte auf die Variablen in der Steuerung geschrieben werden sollen.

**Rezeptur speichern** Die Werte der gegebenen Rezeptur werden in eine Datei mit Erweiterung **\*.txtrecipe** gespeichert, deren Name definiert werden muss. Dazu öffnet der Standarddialog zum Speichern einer Datei.

**ACHTUNG:** Die implizit verwendeten Rezeptur-Dateien, die als Zwischenablage beim Lesen und Schreiben von Rezepturen nötig sind, dürfen dabei nicht überschrieben werden. D.h. der Name für

die neue Datei muss anders lauten als "<Rezepturname>.<Rezepturdefinitionsname>.txtrecipe" !

1. Parameter: Name der Rezepturdefinition
2. Parameter: Name der Rezeptur, die in der angegebenen Datei gespeichert werden soll.

**Rezeptur laden** Die Rezeptur, die in einer Datei gespeichert wurde (s.o. "Rezeptur speichern"), kann aus dieser Datei wieder geladen werden. Zur Auswahl der Datei wird der Standard-Dateiauswahl-Dialog geöffnet. Der Filter ist automatisch auf Dateierweiterung "\*.txtrecipe" gestellt. Nach dem Laden wird die Anzeige der betreffenden Rezeptur im CODESYS Rezepturverwalter aktualisiert.

1. Parameter: Name der Rezepturdefinition
2. Parameter: Name der Rezeptur, für die die Werte aus der angegebenen Datei wieder in den Rezepturverwalter geladen werden sollen.

**Rezeptur löschen** Die angegebene Rezeptur wird aus der angegebenen Rezepturdefinition gelöscht.

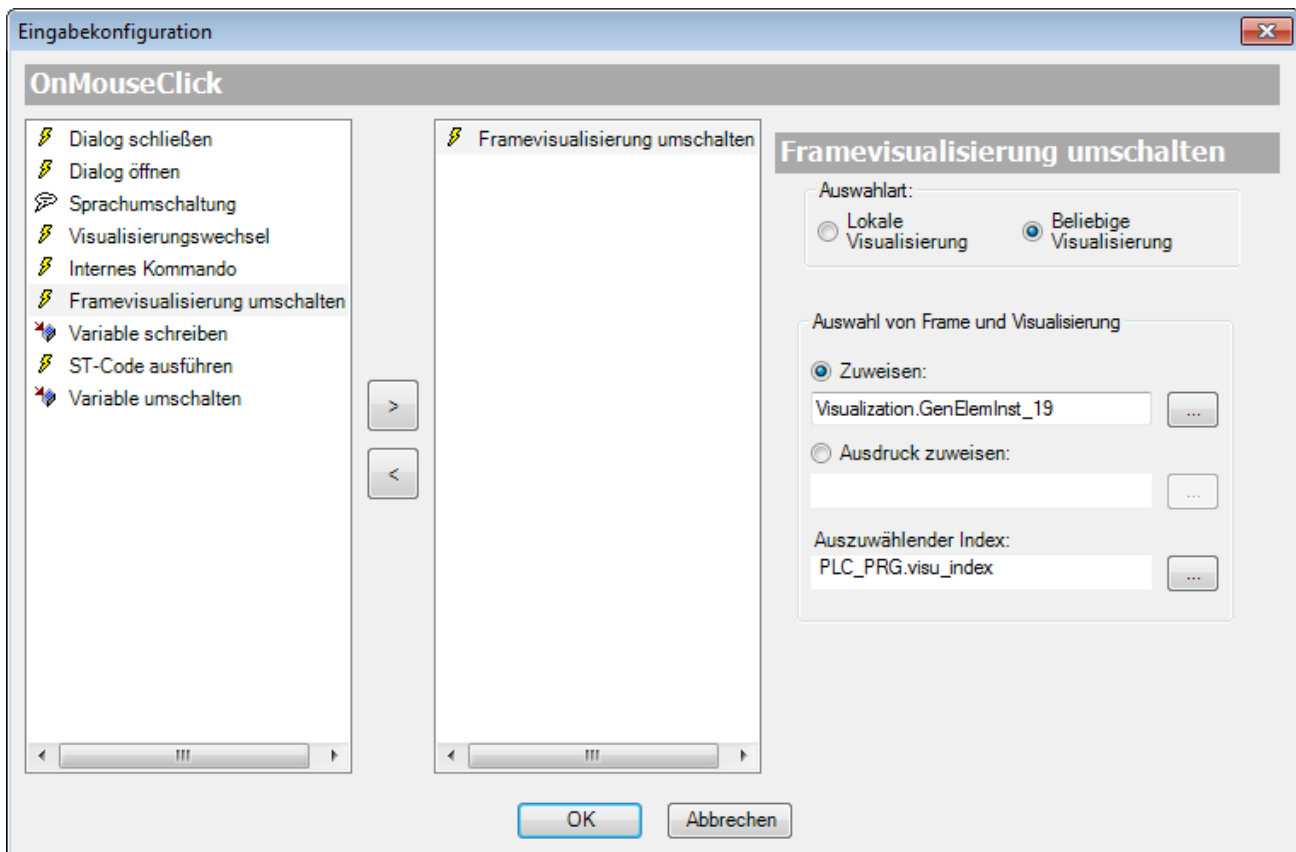
1. Parameter: Name der Rezepturdefinition
2. Parameter: Name der Rezeptur

Jede Eingabe per Maus, der im Eingabe-Konfigurationsdialog eine Aktion zugewiesen ist, zeigt im Eigenschaften-Fenster die Eigenschaft "Konfiguriert" an.

### Framevisualisierung umschalten

**Voraussetzungen:** In einer Visualisierung im Projekt gibt es Frame-Elemente, denen über den Befehl [„Frame-Auswahl“](#) und Dialog [„Konfiguration der Framevisualisierungen“](#) verschiedene Visualisierungen zugeordnet wurden. Diese Visualisierungen erhalten innerhalb des Frames eine Indizierung 0, 1, 2, usw. und im Standardfall wird die erste der zugeordneten Visualisierungen (Index "0") online im Frame angezeigt.

[„Framevisualisierung umschalten“](#): Im vorliegenden Dialog können Sie nun konfigurieren, dass bei Ausführen der Mausaktion auf das aktuelle Visu-Element in einem bestimmten Frame-Element eine bestimmte der ihm zugeordneten Visualisierungen angezeigt wird. Mit dem aktuellen Element kann also ein "Umschalter" für die Anzeige von Visualisierungen in einem Frame programmiert werden.



▼ Abb. 45: "Eingabekonfiguration", 'Framevisualisierung umschalten' selektiert

„Auswahlart“: Die Auswahl des betroffenen Frame-Elementes ...

kann sich auf die aktuelle Visualisierung beschränken, was einen einfacheren Konfigurationsdialog, allerdings ohne dynamische Konfigurationsmöglichkeit, bedeutet -> Option „Lokale Visualisierung“.

kann für alle im Projekt verfügbaren Visualisierungen erweitert werden, die auch dynamische Definitionen des Frames und der angezeigten Visualisierung erlaubt -> Option „Beliebige Visualisierung“.

„Lokale Visualisierung“: Nur Frames der aktuellen Visualisierung können angesprochen werden und stehen somit hier im Feld 'Auswahl des Frames' zur Auswahl (s.u.). Dies ist ein einfacher Dialog zur schnellen direkten Konfiguration in der lokalen Visualisierung, bietet allerdings nicht die Möglichkeit, die gewünschte Visualisierung über eine Projektvariablen anzugeben. Wenn dies nötig ist, wählen Sie Option 'Beliebige Visualisierung', siehe unten.

„Auswahl des Frames“ zeigt die lokalen Frame-Elemente und eingerückt darunter jeweils die zugeordneten Visualisierungen.

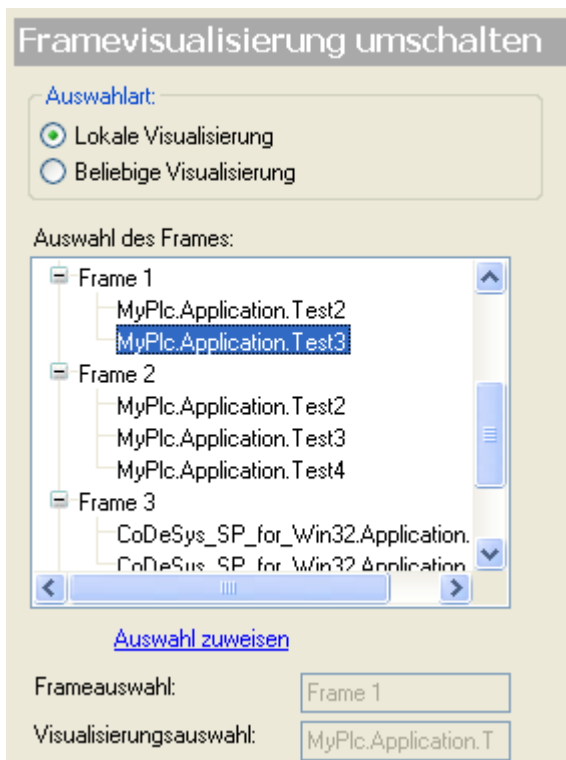


Abb. 46: Auswahl des Frames, lokal

Wählen Sie unterhalb des betroffenen Frames die gewünschte Visualisierung, die auf die Mausaktion hin angezeigt werden soll. Klicken Sie auf 'Auswahl zuweisen', um die Einstellung zu speichern. Die aktuelle Auswahl wird daraufhin in den Feldern Frameauswahl und Visualisierungsauswahl angezeigt.

**„Beliebige Visualisierung“:** Alle Frames des Projekts und ihre zugeordneten Visualisierungen stehen zur Auswahl. In diesem Fall können Frame und Visualisierung auch über Projektvariablen, also dynamisch angegeben werden. Die 'Auswahl von Frame und Visualisierung' wird dann über folgende Felder und Optionen vorgenommen:

Das gewünschte Frame-Element kann direkt oder über einen Ausdruck angegeben werden:

**„Direkte Zuweisung“:** Wenn diese Option aktiviert ist, muss der komplette Pfad des betroffenen Frame-Elementes direkt eingegeben werden. Über Schaltfläche [...] kann die Eingabehilfe dazu verwendet werden. Syntax: <Gerätename>.<Applikationsname>.<Visualisierungsname>.<Frame-Elementname>. Sehen Sie ein Beispiel in der folgenden Abbildung.



Abb. 47: Auswahl des Frames, direkt

**„Zuweisung über Ausdruck“:** **Zuweisung über Ausdruck:** Diese Option kann alternativ aktiviert werden, um eine Projektvariable vom Typ STRING für die Angabe des Frame-Elements zu verwenden. Über Schaltfläche [...] kann die Eingabehilfe geöffnet werden. Die Variable muss dann den kompletten Pfad des Frame-Elements liefern. Syntax: <Gerätename>.<Applikationsname>.<Visualisierungsname>.<Frame-Elementname>. Beispiel: frame\_var: STRING:='MyPlc.Application.Test1.Frame1';

**Die gewünschte Visualisierung,** die bei Mausaktion im ausgewählten Frame angezeigt werden soll, wird über ihren Index angegeben. Dieser Index ist eine mit 0 beginnende aufsteigende, ganzzahlige Nummerierung der Visualisierungen, die einem Frame im Dialog 'Konfiguration der Framevisualisierungen' zugewiesen wurden. Somit wird initial standardmäßig immer die erste Visualisierung in dieser Liste angezeigt.

**Auswählender Index:** Geben Sie hier den Index der gewünschten Visualisierung direkt oder über eine in der Applikation verwendete Projektvariable an. Die Schaltfläche [...] öffnet die Eingabehilfe zur Auswahl einer Variablen.

### Variable schreiben

Eingabefeld, Numpad, Keypad

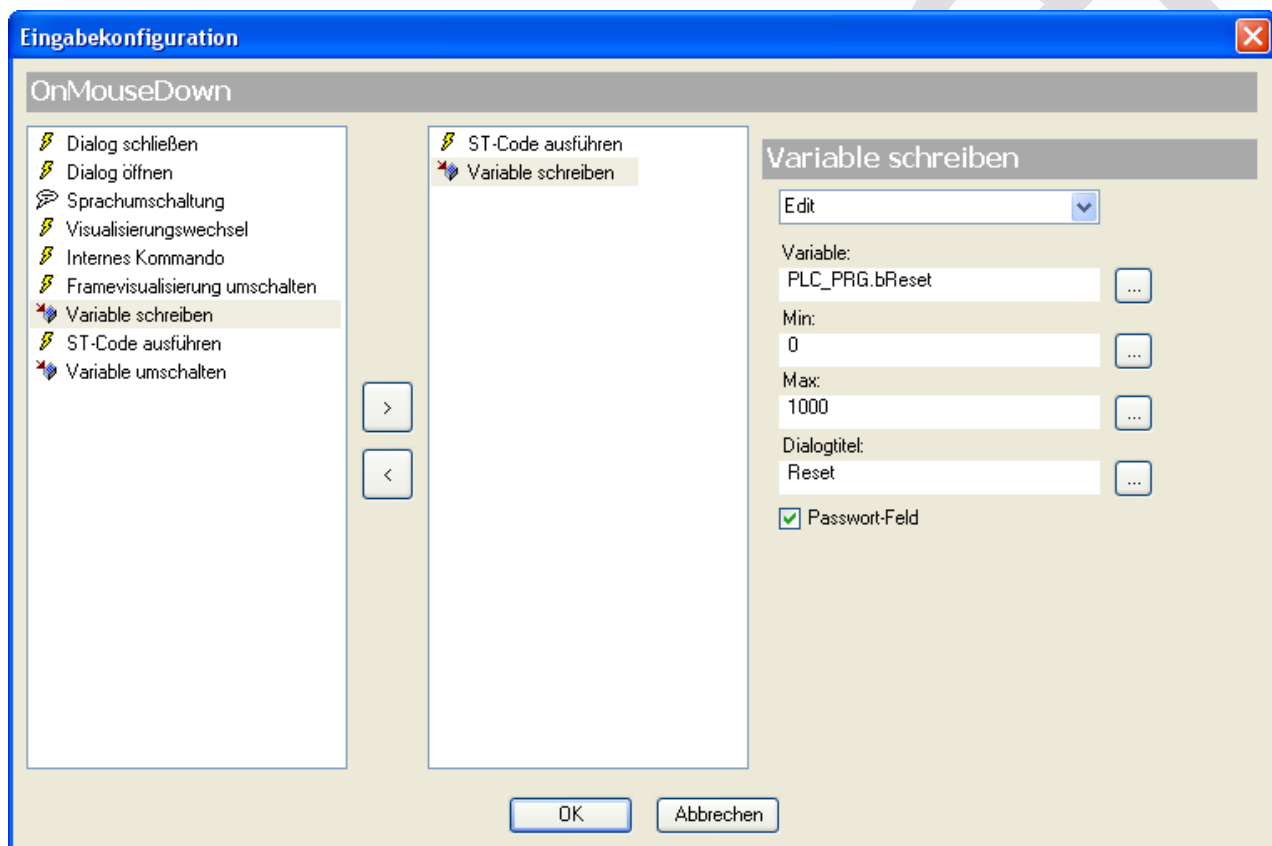


Abb. 48: 'Eingabekonfiguration', 'Variable schreiben' selektiert

Wenn für ein Visualisierungselement eine Eingabekonfiguration des Typs 'Variable schreiben' gibt, dann wird - sobald die entsprechende Mausaktion ausgeführt wird - das Element eine Möglichkeit bieten, einen Wert einzugeben. Der Wert kann dann als eine Zeichenfolge oder über ein Numpad oder Keypad eingegeben werden und wird nach Beendigung der Eingabe auf die Projektvariable geschrieben, die hier im Eingabekonfigurationsdialog angegeben ist. Der Wert wird entsprechend des Datentyps der Projektvariable als Text oder als numerischer Wert interpretiert.

Verwenden des Konfigurationsdialogs:

Wählen Sie von der Auswahlliste im rechten Teil des Dialogs einen der folgenden Eingabetypen:

**„Edit“:** Die Mausaktion wird ein Eingabefeld öffnen, in das ein Text oder ein numerischer Wert eingetippt werden

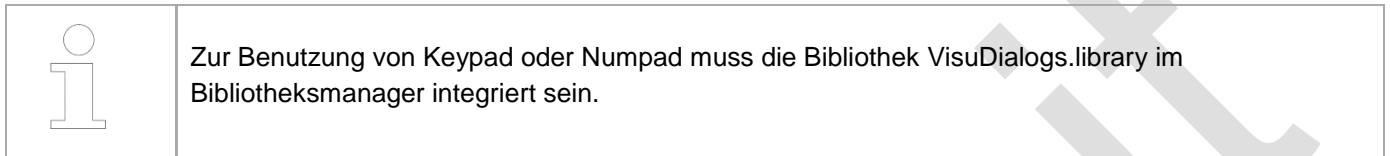


kann.

„*VisuDialogs.Keypad*“: Die Mausaktion wird ein simuliertes Keypad (Tastaturfeld) öffnen. Sie können durch Mausklicks auf die entsprechenden Tastenfelder eine Zeichenfolge eingeben.


„*VisuDialogs.Numpad*“: Die Mausaktion wird ein simuliertes Numpad (Nummerntastaturfeld) öffnen. Sie können durch Mausklicks auf die entsprechenden Tastenfelder eine numerische Zeichenfolge eingeben.

„*<Visualisierungsname>*“: Namen von verfügbaren Visualisierungen, die in ihren Eigenschaften ([↘ „Eigenschaften eines Visualisierungsobjekts“](#)) als "Nummernfeld"/"Tastatur" definiert sind. Wenn eine davon als Bearbeitungstyp ausgewählt wird, wird die Mausaktion das entsprechende visualisierte Nummernfeld bzw. die Tastatur öffnen.



„*Zu bearbeitende Variable*“: Mit einer der folgenden Optionen wird die Variable festgelegt, in die der vom Benutzer eingetippte Wert geschrieben werden soll:

„*TextausgabevARIABLE*“: Der Wert wird in die Variable geschrieben, die in den Variablen der Eigenschaften des Visualisierungselements angegeben ist.

„*Andere Variable verwenden*“: Angabe einer Projektvariable. Hierzu kann über die Schaltfläche  die Eingabehilfe geöffnet werden.

Für den Wert, der in die angegebene Variable geschrieben werden soll, kann eine Unter- und Obergrenze definiert werden. Dazu trägt man entweder direkt Werte in die „*Min.*“ und „*Max.*“ Felder ein, oder entsprechende Projektvariablen.

„*Dialog title*“: Hier kann als Text oder über eine Textvariable angegeben werden, was in der Titelleiste des Dialogs angezeigt werden soll.

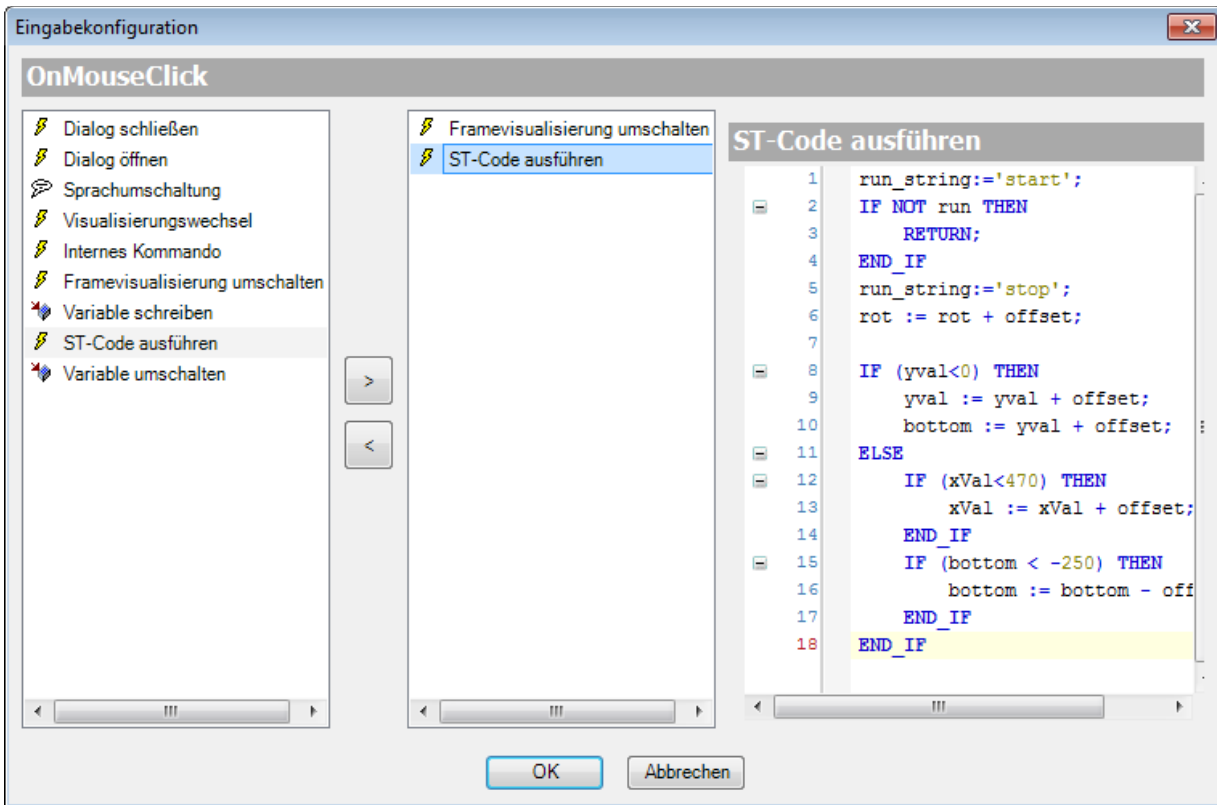
Falls Sie wünschen, dass Ihre Eingabe versteckt dargestellt werden soll durch "\*\*\*\*\*" (wie im Fall von Passwörtern) so aktivieren Sie das Kontrollkästchen vor dem „*Passwort-Feld*“.

Beispiel

Sehen Sie als Beispiel die obige Dialogabbildung und nehmen Sie an, es handele sich um den Konfigurationsdialog für ein Rechteck-Element. Wenn im Online-Modus die Maustaste auf diesem Rechteck gedrückt wird, erscheint ein Nummerntastaturfeld mit dem Titel, der von der Variablen PLC\_PRG.MyTitle geliefert wird.

Durch Mausklicks auf 1, 2 und 3 kann beispielsweise der Wert "123" eingegeben werden. Dieser Wert wird im oberen Teil des Dialogs angezeigt, ebenso wie die Minimal- und Maximalwerte für die Eingabe, die durch PLC\_PRG.minVal und PLC\_PRG.maxVal gegeben werden. Sobald die Eingabe mit einem Mausklick auf die Taste OK bestätigt wird, wird "123" auf die Variable PLC\_PRG.ivar geschrieben. Wenn "ivar" als STRING-Variable definiert ist, erhält es den Wert '123', wenn ivar eine numerische Variable ist, erhält sie den Wert 123.

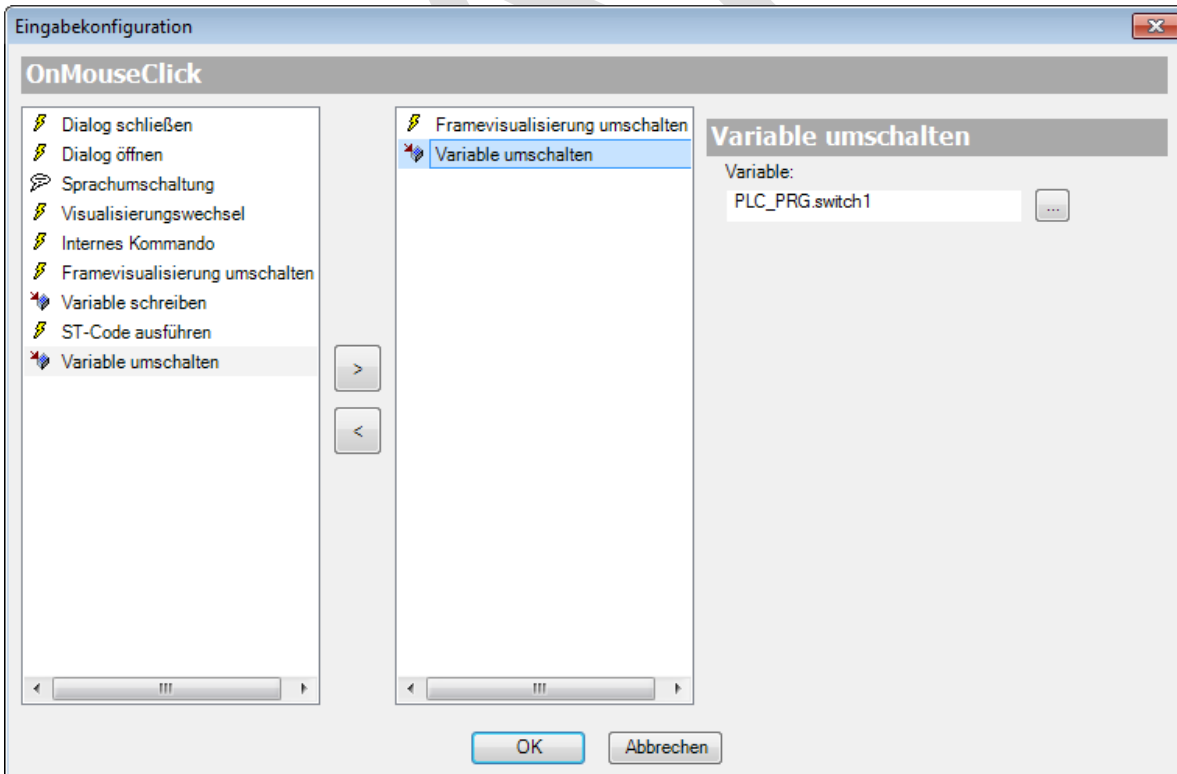
**ST-Code ausführen**




▼ Abb. 49: 'Eingabekonfiguration', 'St-Code ausführen' selektiert

Im Eingabefeld kann Code in Strukturiertem Text eingegeben werden, der auf die Mausaktion hin ausgeführt werden soll.

### Variable umschalten



▼ Abb. 50: 'Eingabekonfiguration', 'Variable umschalten' selektiert

Hier kann eine boolesche Variable eingegeben werden, die bei wiederholter Mausektion zwischen TRUE und FALSE wechseln soll. Beispiel: PLC\_PRG.switch1 Mit Klick auf  wird der „Eingabeassistent“ helfen.

### Farbverlaufeditor

Ein Klick in das Wertefeld von „Farbverlaufauswahl“ im „Eigenschaften“ Fenster öffnet den „Farbverlaufeditor“.

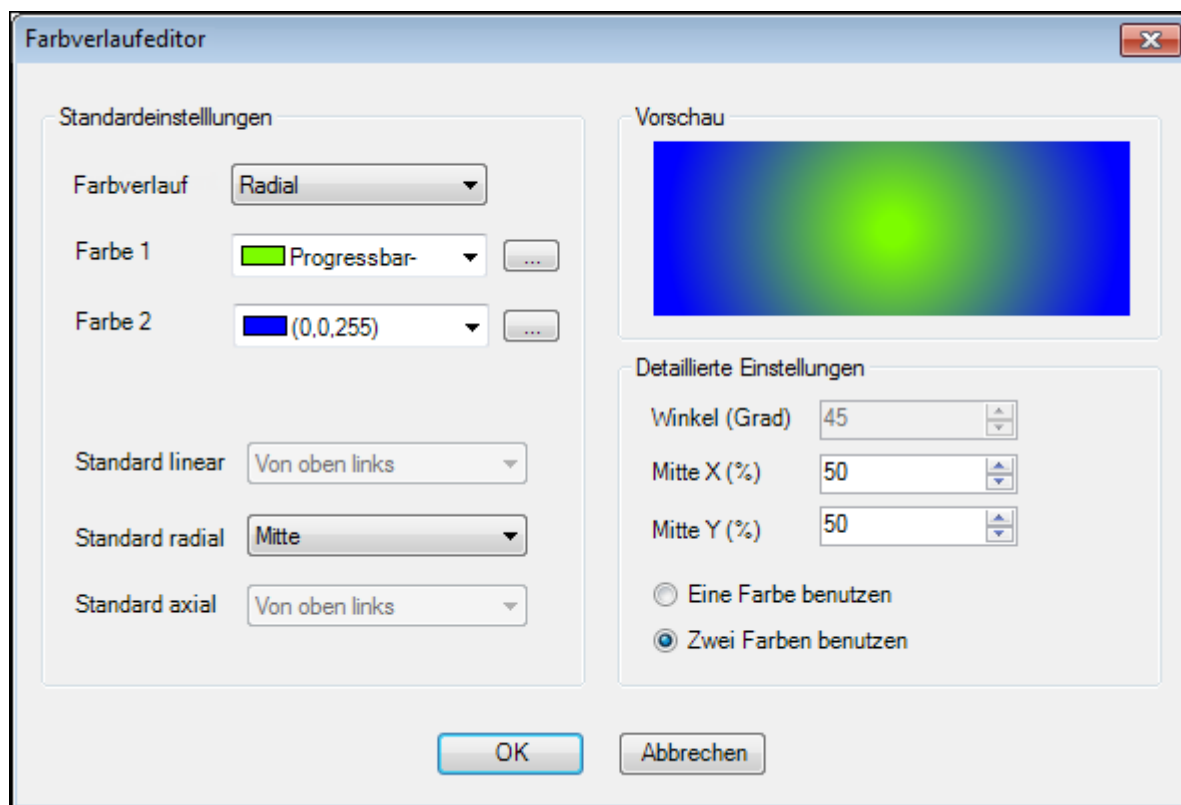




Abb. 51: Farbverlaufeditor

#### Standardeinstellungen

„Farbverlauf“: Linear, Radial, Axial.

„Farbe 1“: Erste Farbe des Gradienten. Sie kann aus der Combobox oder aus dem Farbdialog, zu öffnen mit , gewählt werden.

„Farbe 2“: Zweite Farbe des Gradienten. Sie kann aus der Combobox oder aus dem Farbdialog, zu öffnen mit , gewählt werden.

„Standard linear“: Vordefinierte Einstellungen des Farbverlaufstypes „Linear“ (Winkel des Gradienten): „Horizontal“, „Vertikal“, „Von oben links“, „Von oben rechts“, „Horizontale Farbe umgeschaltet“, „Vertikale Farbe umgeschaltet“, „Von unten links“, „Von unten rechts“.

„Standard radial“: Vordefinierte Einstellungen des Farbverlaufstypes „Radial“ (Position der Mitte): „Mitte“, „Oben links“, „Oben rechts“, „Unten links“, „Unten rechts“.

„Standard axial“: Vordefinierte Einstellungen des Farbverlaufstypes „Axial“ (Winkel des Verlaufs): „Horizontal“, „Vertikal“, „Von oben rechts“, „Von oben links“.

#### Detaillierte Einstellungen

„Winkel (Grad)“: Nur für Farbverlaufstyp „Linear“ und „Axial“ verfügbar.

„Mitte X (%)“: Nur für Farbverlaufstyp „Radial“ verfügbar. X-Position des Mittelpunktes (0 - 100%)

„Mitte Y (%)“: Nur für Farbverlauftyp „Radial“ verfügbar. Y-Position des Mittelpunktes (0 - 100%)

„Eine Farbe benutzen“: Farbverlauftyp zwischen „Farbe 1“ und der gleichen Farbe mit anderer Helligkeit. Die „Helligkeit“ kann von 0 (schwarz) bis 100 (weiß) eingestellt werden.

„Zwei Farben benutzen“: Farbverlauftyp zwischen den beiden ausgewählten Farben „Farbe 1“ und „Farbe 2“.

## Zugriffsrechte

Ein Klick in das Feld „Wert“ von „Zugriffsrechte“ in der Ansicht „Eigenschaften“ eines Elements öffnet den folgenden Dialog.

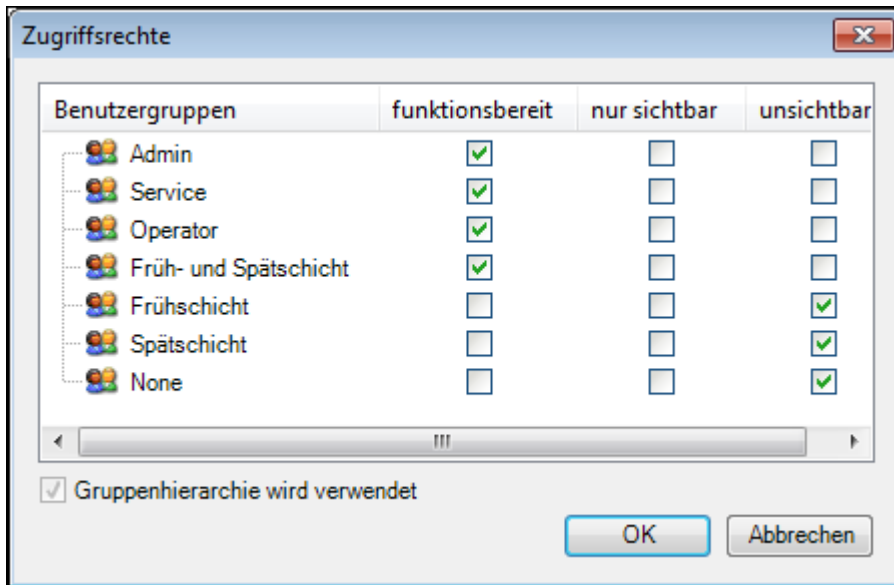


Abb. 52: Zugriffsrechte

„Benutzergruppen“: Die Gruppen, die in [VisualizationManager](#) → [Benutzermanagement](#) → [Gruppen](#) konfiguriert wurden, sind hier aufgelistet. Markieren Sie eine Checkbox in der Zeile der Gruppe um an diese Zugriffsrechte zu vergeben.

„funktionsbereit“: Ist die Checkbox markiert, bietet das Element volle Funktionalität.

„nur sichtbar“: Das Element bietet keine Funktionalität, ist aber sichtbar.

„unsichtbar“: Das Element wird nicht angezeigt.

„Gruppenhierarchie wird verwendet“: Ist in [VisualizationManager](#) → [Benutzermanagement](#) → [Einstellungen](#) die Gruppenhierarchie auf Verwenden gesetzt, dann wird dies hier angezeigt und die deaktivierte Checkbox markiert.

### Defaultgruppe „None“

„None“ ist Defaultgruppe und kann nicht gelöscht werden. Ist kein Benutzer eingeloggt, dann verhalten sich die Visualisierungselemente entsprechend der Konfiguration von „None“.



Werden die Zugriffsrechte eines Elements beschränkt, ist es empfehlenswert, „None“ mit den geringsten Rechten zu belegen

## 6.3.5.2 Basis

### Rechteck, abgerundetes Rechteck und Ellipse

▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Auszuwählen ist einer aus: Rechteck abgerundetes Rechteck Ellipse

Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.



Eigenschaft	Beschreibung Wert
„ <i>X</i> “	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„ <i>Y</i> “	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„ <i>Breite</i> “	Horizontale Breite des Elements.
„ <i>Höhe</i> “	Vertikale Höhe des Elements.

Zentrum

Mit Bearbeiten des Wertes wird gleichzeitig das zugehörige Element  im Editor verschoben.

Eigenschaft	Beschreibung Wert
„ <i>X</i> “	X-Position des Rotationspunkts des Elements
„ <i>Y</i> “	Y-Position des Rotationspunkts des Elements

Farben

Eine Farbe wird mittels Hexzahl definiert, die sich aus Rot/Grün/Blau (RGB) Anteilen zusammensetzt. Für jede dieser drei Farben sind 256 (0-255) Werte verfügbar. Hier sind die Einstellungen statisch. Die Farbe kann aus einer Auswahlliste oder dem Farbauswahldialog, der über Button  öffnet, gewählt werden. Beispiel:  0; 128; 64

Eigenschaft	Beschreibung Wert
-------------	-------------------

- „Normalzustand“ Auszuwählen ist eine Rahmen- und Füllfarbe für Normalzustand. Im Fall, dass die Variable definiert in *Farbvariablen* → *Farbumschlag* FALSE ist, ist das Element in Normalzustand.
- „Rahmenfarbe“
- „Füllfarbe“
- „Alarmzustand“ Auszuwählen ist eine Rahmen- und Füllfarbe, mit der das Element im Alarmzustand dargestellt wird. Dies ist der Fall, wenn die Variable in *Farbvariablen* → *Farbumschlag* TRUE ist.
- „Rahmenfarbe“
- „Füllfarbe“
- „Farbverlauf verwenden“ Standardmäßig ist die Checkbox deaktiviert. Falls sie aktiviert wird, wird das zugehörige Element mit Farbverlauf gezeichnet.
- „Farbverlaufsauswahl“ Dialog [↗](#) „Farbverlaufeditor“ öffnet.

## Aussehen

Eigenschaft	Beschreibung Wert
„Linienstärke“	Stärke der Rahmenlinie in Pixel. 0 kodiert das Gleiche wie 1 und setzt die Linienstärke auf 1 Pixel. Wird kein Rahmen gewünscht, muss man „Linienart“ auf unsichtbar setzen.
„Füllart“	Auszuwählen ist eines dieser Füllarten angewendet in „Füllfarbe“ Ausgefüllt: Die Füllfarbe ist sichtbar Unsichtbar: Die Füllfarbe ist unsichtbar
„Linienart“	Auszuwählen ist eine der Linienarten angewendet bei der Umrisslinie: Durchgezogen Striche Punkte Strich Punkt Strich Punkt Punkt Unsichtbar: Umrisslinie ist nicht sichtbar.

## Texte

Diese Eigenschaften werden für eine statische Definition der Elementbeschriftung verwendet.

Property	Value description
„Text“	Der angegebene Text beschriftet das Element. Ein textformatierender String ist möglich.
„Tooltip“	Der angegebenen Text erscheint als Tooltip des Elements, wenn die Visualisierung im Online-Modus ist.


## Texteigenschaften


Diese Eigenschaften werden für eine statische Definition der Schriftart verwendet.

Eigenschaft	Beschreibung Wert
„Horizontale Ausrichtung“	Definieren der horizontalen Ausrichtung des Textes durch Auswahl aus: Links Zentriert Rechts
„Vertikale Ausrichtung“	Definieren der vertikalen Ausrichtung des Textes durch Auswahl aus: Oben Zentriert Unten
„Schriftart“	Auswahl aus vordefinierten Schriftarten: Standard Überschrift Große Titel Anmerkung Mit Klick auf <input type="button" value="..."/> öffnet sich der Dialog für benutzerdefinierte Eigenschaften der Schriftart.
„Farbe Schriftart“	Auszuwählen ist eine Schriftartfarbe für das Element. Entweder aus der Auswahlliste oder durch Setzen im sich öffnenden Dialog, wenn <input type="button" value="..."/> geklickt wird.

#### Absolute Bewegung

Das Element kann bewegt werden, indem die X- und Y-Position (Pixel) der linken oberen Ecke des Elements durch eine Integer-Variablen verändert werden. Hier werden absolute Koordinatenwerte verwendet.

Eigenschaft	Beschreibung Wert
„Bewegung“	„X“: Hier ist eine Integer-Variablen einzugeben, deren Wert die aktuelle X-Position der linken oberen Ecke des Elements in Pixel definiert; kann verwendet werden, um das Element in X-Richtung zu verschieben (ein positiver Wert verschiebt das Element von links nach rechts).
„Y“	„Y“: Eingabe einer Integer-Variablen, die die aktuelle Y-Position der linken oberen Ecke des Elements in Pixel definiert; kann verwendet werden, um das Element in Y-Richtung zu verschieben (ein positiver Wert verschiebt das Element von oben nach unten).
„Rotation“	Eingabe einer Integer-Variablen, deren Wert den Winkel (Winkelgrade) definiert, um den das Element um den Rotationspunkt „Zentrum“  rotiert. positiver Wert = mathematisch positiv = im Uhrzeigersinn Zu beachten ist, dass das Element, im Gegensatz zum Verhalten bei „Innere Rotation“, selber nicht rotiert.
„Skalierung“	Eingabe einer Integer-Variablen, deren Wert den aktuellen Skalenwert (Prozent) definiert. Die Elementgröße wird linear entsprechend dieses Wertes verändert. Implizit wird der Wert durch 1000

dividiert, so dass es nicht nötig ist, REAL-Variablen zu verwenden um das Element zu verkleinern. Die Skalierung bezieht sich immer auf den Rotationspunkt: „Zentrum“ .

## Relative Bewegung

Das Element kann relative zu seiner fixen Position definiert in „Position“ verschoben werden. Die obere linke und untere rechte Ecke des Elements werden dabei um die in den Integer-Variablen angegebenen Werte (Pixel) in X- bzw. Y-Richtung verschoben. Im Gegensatz zur absoluten Bewegung wird eine relative Position definiert, also der Abstand zur ursprünglichen Position. Das Element kann dadurch in der Form verändert werden.

Siehe „Absolute Bewegung“ für das Verschieben des Elements als Ganzes.

Eigenschaft	Beschreibung Wert
„Bewegung links oben“	„X“: Integer-Variable, deren Wert die Anzahl der Pixel angibt, um die die linke obere Ecke in X-Richtung verschoben wird.
„X“	„Y“: Integer-Variable, deren Wert die Anzahl der Pixel angibt, um die die linke obere Ecke in Y-Richtung verschoben wird.
„Y“	
„Bewegung rechts unten“	„X“: „X“: Integer-Variable, deren Wert die Anzahl der Pixel angibt, um die die rechte untere Ecke in X-Richtung verschoben wird.
„X“	„Y“: „X“: „X“: Integer-Variable, deren Wert die Anzahl der Pixel angibt, um die die rechte untere Ecke in Y-Richtung verschoben wird.
„Y“	

## Textvariablen

Dies sind dynamische Definitionen. Siehe auch „Dynamische Texte“ bei Verwendung von Textlisten. Für statische Definitionen siehe „Texte“

Eigenschaft	Beschreibung Wert
„Textvariable“	Die zugewiesene Variable definiert den Elementtext ("Textausgabevariable"). Unter <a href="#">„Text und Sprache in Visualisierungen“</a> ist zu finden, wie ein Text zu formatieren ist.
„Tooltipvariable“	Die zugewiesene STRING-Variable enthält den Tooltip-Text für das Element.

## Dynamische Texte

Diese Parameter dienen der Definition von dynamischen Texten, die aus Textlisten stammen (dies erlaubt Sprachumschaltung). Siehe [Textliste](#) für nähere Informationen zur Verwendung von Textlisten.

Eigenschaft	Beschreibung Wert
„Textliste“	Name der Textliste (String) wie im POU- oder Geräte-Fenster verwendet; z.B. errorlist
„Textindex“	Index (ID) des Textes wie in der Textliste definiert (String); Direkte Angabe der ID, z.B. 'M1' oder String-Variable, z.B. error_22.
„Tooltipindex“	Index (ID) des Tooltip-Textes wie in der Textliste definiert (String); Direkte Angabe der ID, z.B. 'T1' oder String-Variable, z.B. error_Tooltip_22.

## Schriftartvariablen

Diese werden verwendet bei dynamischen Schriftdefinitionen des Elementtextes über Projektvariablen. Statischen



Definitionen werden unter [Texteigenschaften](#) → [Schriftart](#) konfiguriert.

Eigenschaft	Beschreibung Wert
„ <a href="#">Schriftname</a> “	Hier ist eine Variable Typ STRING anzugeben, die den Schriftnamen enthält, der für die Beschriftung des Elements verwendet werden soll. (Namensangabe wie im Standard-Schrifttypen-Dialog). Beispiel: <code>plc_prg.font_var font_var := Arial;</code>
„ <a href="#">Größe</a> “	Eingabe einer Variablen Typ INT, die die Größe des Elementtexts in Pixel enthält, so wie im Standarddialog Font. Beispiel: <code>prog1.font_height font_height:=16;</code>
„ <a href="#">Flags</a> “	Eingabe einer Variable Typ DWORD, die über die unten genannten Flag-Werte die Font-Darstellung festlegt. Eine kombinierte Definition kann erreicht werden, indem die jeweiligen Flag-Werte addiert werden und die Summe angegeben wird.  1: kursiv 2: fett 4: unterstrichen 8: gelöscht  <code>prog2.font_type: ist font_type:=6,</code> wird der Text fett und unterstrichen dargestellt.
„ <a href="#">Zeichensatz</a> “	Der Zeichensatz, der für den Font verwendet werden soll, kann über die Standard-Zeichensatz-Nummer definiert werden. Mit einer DWORD-Variablen kann diese Nummer angegeben werden (siehe auch die „ <a href="#">Skript</a> “-Definition im Standard-Font-Dialog)
„ <a href="#">Farbe</a> “	Eingabe einer DWORD-Variable, in der die Farbe des Elementtextes festgelegt wird.

#### Farbvariablen

Angewandt for dynamische Definitionen der Elementfarbe mittels einer Projektvariablen.

Eine Farbe wird mittels Hexzahl definiert, die sich aus Rot/Grün/Blau (RGB) Anteilen zusammensetzt. Es ist eine Projektvariable Typ DWORD anzugeben, deren Wert die Farbe definiert. (zum Beispiel `prog1.dwFillColor`).

Die Variable enthält den Farbwert in hexadezimalen Format. Um das gesamte DWORD auszufüllen, müssen die ersten beiden Ziffern nach "16#" mit 0 gesetzt werden: `16#00RRGGBB`

#### Farbwerte

`dwFillColor := 16#008FE03F;`

Hexzahl:

8F: rot

E0: grün

3F: blau

Eigenschaft	Beschreibung Wert
„ <a href="#">Farbumschlag</a> “	Hier ist eine boolsche Variable angeben, die das Umschalten der Elementfarbe zwischen „ <a href="#">Normalzustand</a> “ (Variable = FALSE) und „ <a href="#">Alarmzustand</a> “ (Variable = TRUE) steuert.

„Normalzustand“	Eingabe von DWORD-Variablen, die die Rahmen- und Füllfarbe des Elements bestimmen.
„Rahmenfarbe“	Sie überschreiben die Werte, die aktuell in <a href="#">Farben → Normalzustand</a> definiert sind. Die Werte in den Projektvariablen werden verwendet, wenn die Variable definiert in „ <a href="#">Farbumschlag</a> “ FALSE ist.
„Füllfarbe“	
„Alarmzustand“	Eingabe von DWORD-Variablen, die die Rahmen- und Füllfarbe des Elements im Alarmzustand bestimmen. Sie überschreiben die Werte, die aktuell in <a href="#">Farben → Alarmzustand</a> definiert sind. Die Werte in den Projektvariablen werden verwendet, wenn die Variable definiert in „ <a href="#">Farbumschlag</a> “ TRUE ist.
„Rahmenfarbe“	
„Füllfarbe“	

## Aussehensvariablen

Anzuwenden bei einer dynamischen Definition des Aussehens der Kontur und der Füllung des Elements. Statische Definitionen werden in „[Aussehen](#)“ festgelegt.

Eigenschaft	Beschreibung Wert
„Linienstärke“	Eingabe einer Integervariablen, die die Linienstärke des Elements in Pixel definiert. Dies überschreibt den fixen Wert, der in <a href="#">Aussehen → Linienstärke</a> festgelegt wurde.
„Füllart“	Eingabe einer DWORD-Variablen, die die Füllart des Elements definiert. Die Farbe, die bei „ <a href="#">Farbvariablen</a> “ angegeben ist, kann angezeigt oder ignoriert werden: Variablenwert = 0: gefüllt Variablenwert > 0 : unsichtbar, das heißt keine Füllung ist sichtbar Dies überschreibt den fixen Wert, der in <a href="#">Aussehen → Füllart</a> festgelegt wurde.
„Linienart“	Eingabe einer DWORD-Variablen, die die Kontur der Außenlinie festlegt. Folgende Werte entsprechen folgenden Linienarten: 0: Durchgezogen 1: Strich 2: Punkt 3: Strich Punkt 4: Strich Punkt Punkt 8: Unsichtbar. Das heißt, das Element wird ohne Außenlinie dargestellt.

## Zustandsvariablen

Eigenschaft	Beschreibung Wert
„Unsichtbarkeit“	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.
„Eingaben deaktivieren“	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann hat keine der Elementeingaben eine Auswirkung.

## Eingabekonfiguration

Hier kann definiert werden, welche Folgeaktionen ausgeführt werden, wenn der Benutzer im Online-Modus eine Eingabe im Element vornimmt.

Eingabeereignisse, die ein Element haben kann, sind:

„*OnDialogClosed*“

„*OnClick*“

„*MouseDown*“

„*MouseEnter*“

„*MouseLeave*“

„*MouseMove*“

„*MouseUp*“

Mit Klick auf „*Konfigurieren...*“ wird der Dialog [↘ „Eingabekonfiguration“](#) geöffnet, der erlaubt Folgeaktionen zuzuweisen. Man klickt in das Feld neben der zu konfigurierenden Eingabe. Für jede selektierte Folgeaktion wird ein Eintrag in der Eigenschaftentabelle erzeugt. Jede Eingabe kann eine beliebige zugeordnete Anzahl an Folgeaktionen haben.

Eigenschaft	Beschreibung Wert
„ <i>OnDialogClosed</i> “	Dieses Ereignis tritt ein, wenn innerhalb der Visualisierung eines der für Benutzereingaben vorher geöffneten Dialogfenster geschlossen wird.
„ <i>OnClick</i> “	Dieses Mausereignis tritt ein, wenn der Cursor auf ein Element zeigt und ein vollständiger Mausklick (drücken und freigeben der Maustaste) ausgeführt wird.
„ <i>MouseDown</i> “	Dieses Mausereignis tritt ein, wenn die Maustaste gedrückt wird während der Cursor auf ein Element zeigt. (Drücken der Maustaste)
„ <i>MouseEnter</i> “	Dieses Mausereignis tritt ein, wenn der Cursor auf das Element gezogen wird.
„ <i>MouseLeave</i> “	Dieses Mausereignis tritt ein, wenn der Cursor das Element verlässt.
„ <i>MouseMove</i> “	Dieses Mausereignis tritt ein, wenn der Cursor innerhalb des Elements bewegt wird.
„ <i>MouseUp</i> “	Diese Mausereignis liegt vor, wenn die Maustaste losgelassen wird. Das Drücken der Maustaste geschah vorher, außerhalb des Elements.

Eingabekonfiguration: Tasten

Mit „*Tasten*“ kann konfiguriert werden, dass beim Ereignis "Tasten" der Wert einer boolschen Projektvariable abhängig vom Mausverhalten gesetzt wird.


Eigenschaft	Beschreibung Wert
„ <i>Variable</i> “	Hier kann eine boolsche Variable zugewiesen werden, deren Wert TRUE wird, wenn die Maustaste gedrückt wird, während der Cursor auf das Element zeigt und die wieder FALSE wird, wenn die Maustaste wieder losgelassen wird.
„ <i>FALSE tasten</i> “	Wenn diese Option aktiviert ist, wird das oben beschriebene Tastverhalten für die eingetragene Variable umgekehrt. Das heißt, wenn die Maustaste gedrückt wird, wird der Variablenwert auf FALSE gesetzt, wenn die Taste wieder losgelassen wird, wird der Variablenwert auf TRUE gesetzt.

„Beim Betreten tasten, falls Maus gefangen“

Ist diese Option aktiviert, verhält sich der Variablenwert wie unter „[Variable](#)“ beschrieben, solange sich die Maus innerhalb des Elementbereichs befindet. Zusätzlich wird auf ein Ziehen der Maus reagiert. Wird bei gedrückter Maustaste der Elementbereich verlassen, wird die Variable FALSE und wird wieder TRUE, wenn in den Elementbereich zurückgekehrt wird. Es wird also berücksichtigt, dass, solange die Maustaste gedrückt ist, die Maus "gefangen" ist, auch wenn der Elementbereich verlassen wird.

#### Eingabekonfiguration: Umschalten

Mit „[Umschalten](#)“ kann konfiguriert werden, dass bei diesem Ereignis der Wert einer boolschen Projektvariable abhängig vom Mausverhalten gesetzt wird.

Eigenschaft	Beschreibung Wert
„ <a href="#">Variable</a> “	<p>Hier ist eine boolsche Variable zuzuweisen. Die Eingabehilfe kann über Schaltfläche  geöffnet werden.</p> <p>Der Wert der Variablen wird bei jedem Mausklick auf das Element zwischen TRUE und FALSE umgeschaltet. Wird der Cursor bei gedrückter Maustaste aus dem Element herausbewegt, dann erfolgt kein Toggeln. Damit kann die begonnene Umschalteingabe abgebrochen werden.</p>
„ <a href="#">Umschalten beim Loslassen falls Maus gefangen</a> “	<p>Ist diese Option aktiviert, verhält sich der Variablenwert wie unter „<a href="#">Variable</a>“ beschrieben, solange sich die Maus innerhalb des Elementbereichs befindet. Zusätzlich wird auf ein Ziehen der Maus reagiert. Wird bei gedrückter Maustaste der Elementbereich verlassen und dann die Maustaste losgelassen, wird die Variable umgeschaltet. Es wird also berücksichtigt, dass, solange die Maustaste gedrückt ist, die Maus "gefangen" ist, auch wenn der Elementbereich verlassen wird.</p>

#### Eingabekonfiguration: Hotkey

Mit „[Hotkey](#)“ kann eine Taste, auch mit Zusatzstufe, definiert und mit einer bestimmten Nachfolgeaktion (MouseDown, MouseUp) verknüpft werden, die bei einem Ereignis der Taste (KeyDown, KeyUp) ausgeführt werden soll. Standardmäßig wird bei KeyDown (Taste drücken) die MouseDown-Aktion ausgeführt und bei KeyUp (Taste loslassen) die MouseUp-Aktion. Das kann nützlich sein, wenn eine Visualisierung sowohl über Maus- als auch Tastatureingaben bedient werden soll, weil die Eingabeaktionen dann nur einmalig konfiguriert werden müssen. Diese Tastatenkonfiguration für ein Element wird auch im [„Tastaturkonfiguration“](#) der Visualisierung verwaltet. Änderungen werden immer zwischen diesem und dem Elementeigenschaften-Editor synchronisiert.

Eigenschaft	Beschreibung Wert
„ <a href="#">Taste</a> “	<p>Zuweisen einer Taste. Eine Auswahlliste stellt alle aktuell unterstützten Tasten bereit wie zum Beispiel [ M ]. <a href="#">„Tastaturkonfiguration“</a></p>
„ <a href="#">Eingabe(n)</a> “	<p>Definieren des Ereignisses, das ausgeführt werden soll, wenn die Taste beziehungsweise Taste und Modifizierer verwendet werden.</p> <p>Mögliche Werte, die in einer Auswahlliste bereitstehen:</p> <p>Keine Aktion</p> <p>MouseDown-Aktion: OnMouseDown-Aktion, wenn die Taste gedrückt wird.</p> <p>MouseUp-Aktion: OnMouseUp-Aktion, wenn die Taste freigelassen wird.</p> <p>MouseDown/MouseUp-Aktion: OnMouseUp- und OnMouseDown-Aktion, wenn die Taste</p>

gedrückt oder losgelassen wird.

„Umschalten“ Wenn diese Option aktiviert wird, muss die Taste mit [ *Umschalt* ] kombiniert verwendet werden

„Steuerung“ Wenn diese Option aktiviert wird, muss die Taste mit [ *Strg* ] kombiniert verwendet werden

„Alt“ Wenn diese Option aktiviert wird, muss die Taste mit [ *Alt* ] kombiniert verwendet werden

Eigenschaft	Beschreibung Wert
-------------	-------------------

„Zugriffsrechte“ Mit Klick öffnet [↘ „Zugriffsrechte“](#).

Statusmeldungen:

Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.

Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.

## Linie

Dieses Element ist eine einfache Linie.

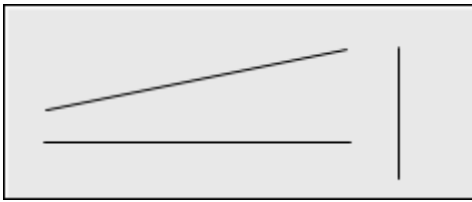


Abb. 54: Linie

### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungseditor werden folgende Benutzereingaben zur Verfügung gestellt:

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„Elementname“	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„Elementtyp“	Linie

#### Position

Eine Linie wird mit zwei Punkten definiert: Punkt [0] bis [1].



Property	Value description
„Punkte“	▪ [ <n> ]
▪ „[ <n> ]“ $n \in \{0, 1\}$	▪ „X“: Horizontale Position in Pixel.
▪ „X“	▪ „Y“: Vertikale Position in Pixel.
▪ „Y“	

#### Zentrum

Mit Bearbeiten des Wertes wird gleichzeitig das zugehörige Element  im Editor verschoben.

Eigenschaft	Beschreibung Wert
„X“	X-Position des Rotationspunkts des Elements
„Y“	Y-Position des Rotationspunkts des Elements

## Farben

Eine Farbe wird mittels Hexzahl definiert, die sich aus Rot/Grün/Blau (RGB) Anteilen zusammensetzt. Für jede dieser drei Farben sind 256 (0-255) Werte verfügbar. Hier sind die Einstellungen **statisch**. Die Farbe kann aus einer Auswahlliste oder dem Farbauswahldialog, der über Button  öffnet, gewählt werden. Beispiel:  0; 128; 64

Eigenschaft	Beschreibung Wert
„ <i>Farbe</i> “	Hier ist eine Farbe, mit der das Element im Normalzustand gefüllt wird, auszuwählen. Dies ist der Fall, wenn die Variable in <i>Farbvariablen</i> → <i>Farbumschlag</i> FALSE ist.  Mit Editieren des Werts ändert das Element gleichzeitig seine Farbe.
„ <i>Alarmfarbe</i> “	Auszuwählen ist eine Farbe, mit der das Element gefüllt wird, wenn das Element in Alarmzustand ist. Dies ist der Fall, wenn die Variable in <i>Farbvariablen</i> → <i>Farbumschlag</i> TRUE ist.

## Aussehen

Eigenschaft	Beschreibung Wert
„ <i>Linienstärke</i> “	Stärke der Rahmenlinie in Pixel. 0 kodiert das Gleiche wie 1 und setzt die Linienstärke auf 1 Pixel. Soll kein Rahmen verwendet werden, muss „ <i>Linienart</i> “ auf <code>unsichtbar</code> setzen werden.
„ <i>Linienart</i> “	Auszuwählen ist eine der Linienarten angewendet bei der Umrisslinie: <ul style="list-style-type: none"><li>▪ Durchgezogen</li><li>▪ Striche</li><li>▪ Punkte</li><li>▪ Strich Punkt</li><li>▪ Strich Punkt Punkt</li><li>▪ Unsichtbar: Dann ist die Umrisslinie unsichtbar.</li></ul>

## Texte

Diese Eigenschaften werden für eine statische Definition der Elementbeschriftung verwendet.

Property	Value description
„ <i>Text</i> “	Der angegebene Text beschriftet das Element. Ein textformatierender String ist möglich.
„ <i>Tooltip</i> “	Der angegebenen Text erscheint als Tooltip des Elements, wenn die Visualisierung im Online-Modus ist.

## Texteigenschaften

Diese Eigenschaften werden für eine statische Definition der Schriftart verwendet.

Eigenschaft	Beschreibung Wert
-------------	-------------------

### „Horizontale Ausrichtung“

Definieren der horizontalen Ausrichtung des Textes durch Auswahl aus:

- Links
- Zentriert
- Rechts

### „Vertikale Ausrichtung“

Definieren der vertikalen Ausrichtung des Textes durch Auswahl aus:

- Oben
- Zentriert
- Unten

### „Schriftart“

Auswahl aus vordefinierten Schriftarten:

- Standard
- Überschrift
- Große
- Titel
- Anmerkung


Mit Klick auf  öffnet sich der Dialog für benutzerdefinierte Eigenschaften der Schriftart.

### „Farbe Schriftart“

Auszuwählen ist eine Schriftartfarbe für das Element. Entweder aus der Auswahlliste oder durch Setzen im sich öffnenden Dialog, wenn  geklickt wird.


## Absolute Bewegung


Das Element kann bewegt werden, indem die X- und Y-Position (Pixel) der linken oberen Ecke des Elements durch eine Integer-Variable verändert wird. Hier werden absolute Koordinatenwerte verwendet.

Eigenschaft	Beschreibung Wert
„Bewegung“	<ul style="list-style-type: none"><li>▪ „X“: Hier ist eine Integer-Variable einzugeben, deren Wert die aktuelle X-Position der linken oberen Ecke des Elements in Pixel definiert; kann verwendet werden, um das Element in X-Richtung zu verschieben (ein positiver Wert verschiebt das Element von links nach rechts).</li><li>▪ „Y“: Eingabe einer Integer-Variablen, die die aktuelle Y-Position der linken oberen Ecke des Elements in Pixel definiert; kann verwendet werden, um das Element in Y-Richtung zu verschieben (ein positiver Wert verschiebt das Element von oben nach unten).</li></ul>
„Rotation“	<p>Eingabe einer Integer-Variablen, deren Wert den Winkel (Winkelgrade) definiert, um den das Element um den Rotationspunkt „Zentrum“  rotiert.</p> <p>positiver Wert = mathematisch positiv = im Uhrzeigersinn</p> <p>Zu beachten ist, dass das Element, im Gegensatz zum Verhalten bei „Innere Rotation“, selber</p>



nicht rotiert.

„*Skalierung*“ Eingabe einer Integer-Variablen, deren Wert den aktuellen Skalenwert (Prozent) definiert. Die Elementgröße wird linear entsprechend dieses Wertes verändert. Implizit wird der Wert durch 1000 dividiert, so dass es nicht nötig ist, REAL-Variablen zu verwenden um das Element zu verkleinern. Die Skalierung bezieht sich immer auf den Rotationspunkt „*Zentrum*“ .

„*Innere Rotation*“ Die hier eingetragene Integer-Variable definiert den Winkel (Winkelgrade), um den das Element um seinen Rotationspunkt gedreht wird: „*Zentrum*“ .

positiver Werte = mathematisch positiv = im Uhrzeigersinn.

Im Gegensatz zu „*Rotation*“ rotiert das Element tatsächlich.

## Relative Bewegung

Das Element kann relative zu seiner fixen Position definiert in „*Position*“ verschoben werden. Siehe „*Absolute Bewegung*“ für das Verschieben des Elements als Ganzes.

Eigenschaft	Beschreibung Wert
„ <i>Bewegung Punkt [&lt;n&gt;]</i> “ mit $n \in \{0, 1\}$	▪ „ <i>X</i> “: Eingabe einer Variablen, deren Wert die Anzahl der Pixel angibt um die der Punkt[0]/Punkt[1] in X-Richtung verschoben wird
▪ „ <i>X</i> “	▪ „ <i>Y</i> “: Eingabe einer Variablen, deren Wert die Anzahl der Pixel angibt um die der Punkt[0]/Punkt[1] in Y-Richtung verschoben wird
▪ „ <i>Y</i> “	

## Textvariablen

Dies sind dynamische Definitionen. Siehe auch „*Dynamische Texte*“ bei Verwendung von Textlisten. Für statische Definitionen siehe „*Texte*“

Eigenschaft	Beschreibung Wert
„ <i>Textvariable</i> “	Die zugewiesene Variable definiert den Elementtext ("Textausgabevariable"). Unter <a href="#">„<i>Text und Sprache in Visualisierungen</i>“</a> ist zu finden, wie ein Text zu formatieren ist.
„ <i>Tooltipvariable</i> “	Die zugewiesene STRING-Variable enthält den Tooltip-Text für das Element.

## Dynamische Texte

Diese Parameter dienen der Definition von dynamischen Texten, die aus Textlisten stammen (dies erlaubt Sprachumschaltung). Siehe [Textliste](#) für nähere Informationen zur Verwendung von Textlisten.

Eigenschaft	Beschreibung Wert
„ <i>Textliste</i> “	Name der Textliste (String) wie im POU- oder Geräte-Fenster verwendet; z.B. <code>errorlist</code>

„*Textindex*“ Index (ID) des Textes wie in der Textliste definiert (String); Direkte Angabe der ID, z.B. 'M1' oder String-Variable, z.B. `error_22`.

„*Tooltipindex*“ Index (ID) des Tooltip-Textes wie in der Textliste definiert (String); Direkte Angabe der ID, z.B. 'T1' oder String-Variable, z.B. `error_Tooltip_22`.

## Schriftartvariablen

Diese werden verwendet bei dynamischen Schriftdefinitionen des Elementtextes über Projektvariablen. Statischen Definitionen werden unter [Texteigenschaften](#) → [Schriftart](#) konfiguriert.

Eigenschaft	Beschreibung Wert
„ <i>Schriftname</i> “	Hier ist eine Variable Typ STRING anzugeben, die den Schriftnamen enthält, der für die Beschriftung des Elements verwendet werden soll. (Namensangabe wie im Standard-Schrifttypen-Dialog). Beispiel: <code>plc_prg.font_var font_var := Arial;</code>
„ <i>Größe</i> “	Eingabe einer Variablen Typ INT, die die Größe des Elementtextes in Pixel enthält, so wie im Standarddialog Font. Beispiel: <code>prog1.font_height font_height:=16;</code>
„ <i>Flags</i> “	Eingabe einer Variable Typ DWORD, die über die unten genannten Flag-Werte die Font-Darstellung festlegt. Eine kombinierte Definition kann erreicht werden, indem die jeweiligen Flag-Werte addiert werden und die Summe angegeben wird. <ul style="list-style-type: none"><li>▪ 1: kursiv</li><li>▪ 2: fett</li><li>▪ 4: unterstrichen</li><li>▪ 8: gelöscht</li></ul> <p><code>prog2.font_type: ist font_type:=6</code>, wird der Text fett und unterstrichen dargestellt.</p>
„ <i>Zeichensatz</i> “	Der Zeichensatz, der für den Font verwendet werden soll, kann über die Standard-Zeichensatz-Nummer definiert werden. Mit einer DWORD-Variablen kann diese Nummer angegeben werden (siehe auch die „ <i>Skript</i> “-Definition im Standard-Font-Dialog)
„ <i>Farbe</i> “	Eingabe einer DWORD-Variable, in der die Farbe des Elementtextes festgelegt wird.

## Farbvariablen

Eine Farbe wird mittels Hexzahl definiert, die sich aus Rot/Grün/Blau (RGB) Anteilen zusammensetzt. Für jede dieser drei Farben sind 256 (0-255) Werte verfügbar. Eine Projektvariable Typ DWORD muss angegeben werden, deren Wert die Farbe definiert (zum Beispiel `prog1.dwFillColor`).

Die Variable enthält den Farbwert in hexadezimalen Format. Um das gesamte DWORD auszufüllen, müssen die ersten beiden Ziffern nach "16#" mit 0 gesetzt werden: `16#00RRGGBB`

## Farbwerte

```
dwFillColor := 16#008FE03F;
```

Hexzahl:

- 8F: rot
- E0: grün
- 3F: blau

Eigenschaft	Beschreibung Wert
„Farbumschlag“	Hier ist eine boolsche Variabel angeben, die das Umschalten der Elementfarbe zwischen „Farbe“ (Variable = FALSE) und „Alarmfarbe“ (Variable = TRUE) steuert.
„Farbe“	Eingabe einer DWORD-Variable, die die Farbe des Elements bestimmt. Sie überschreibt den Wert, der aktuell in <i>Farben</i> → <i>Farbe</i> definiert ist, wenn die Variable in „Farbumschlag“ FALSE ist.
„Alarmfarbe“	Eingabe einer DWORD-Variable, die die Alarmfarbe des Elements bestimmt. Sie überschreibt den Wert, der aktuell in <i>Farben</i> → <i>Alarmfarbe</i> definiert ist, wenn die Variable in „Farbumschlag“ TRUE ist.

#### Zustandsvariablen

Eigenschaft	Beschreibung Wert
„Unsichtbarkeit“	Eingabe einer boolschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.
„Eingaben deaktivieren“	Eingabe einer boolschen Variable; ist der Variablenwert TRUE, dann hat keine der Elementeingaben eine Auswirkung.

#### Linienstärkenvariable

Dynamische Definition der Dicke eines Linienelements mittels Variable

Eigenschaft	Beschreibung Wert
„Ganzzahlwert“	Eingabe einer Integervariablen, die die Linienstärke des Elements in Pixel definiert. Dies überschreibt den fixen Wert, der in <i>Aussehen</i> → <i>Linienstärke</i> festgelegt ist. Zu berücksichtigen ist, dass 0 das Gleiche wie 1 kodiert und die Linienstärke dabei auf 1 Pixel gesetzt wird.

#### Linienartvariable

Dynamische Definition der Linienart mittels Variable

Eigenschaft	Beschreibung Wert
„Integral value“	Eingabe einer Integervariablen, deren Wert die Linienart definiert

Folgende Variablenwerte werden unterstützt:

- Durchgezogen
- Striche
- Punkte
- Strich Punkt
- Strich Punkt Punkt
- Unsichtbar: Dann ist die Linie unsichtbar.

Dies überschreibt den fixen Wert, der in *Aussehen* → *Linienart* festgelegt ist.

## Eingabekonfiguration

Hier kann definiert werden, welche Folgeaktionen ausgeführt werden, wenn der Benutzer im Online-Modus eine Eingabe im Element vornimmt.

Eingabeereignisse, die ein Element haben kann, sind:

- *„OnDialogClosed“*
- *„OnClick“*
- *„MouseDown“*
- *„OnMouseEnter“*
- *„OnMouseLeave“*
- *„OnMouseMove“*
- *„OnMouseUp“*

Mit Klick auf *„Konfigurieren...“* wird der Dialog [„Eingabekonfiguration“](#) geöffnet, der erlaubt Folgeaktionen zuzuweisen. Man klickt in das Feld neben der zu konfigurierenden Eingabe. Für jede selektierte Folgeaktion wird ein Eintrag in der Eigenschaftentabelle erzeugt. Jede Eingabe kann eine beliebige zugeordnete Anzahl an Folgeaktionen haben.

Eigenschaft	Beschreibung Wert
<i>„OnDialogClosed“</i>	Dieses Ereignis tritt ein, wenn innerhalb der Visualisierung eines der für Benutzereingaben vorher geöffneten Dialogfenster geschlossen wird.
<i>„OnClick“</i>	Dieses Mausereignis tritt ein, wenn der Cursor auf ein Element zeigt und ein vollständiger Mausklick (drücken und freigeben der Maustaste) ausgeführt wird.
<i>„MouseDown“</i>	Dieses Mausereignis tritt ein, wenn die Maustaste gedrückt wird während der Cursor auf ein Element zeigt. (Drücken der Maustaste)
<i>„OnMouseEnter“</i>	Dieses Mausereignis tritt ein, wenn der Cursor auf das Element gezogen wird.
<i>„OnMouseLeave“</i>	Dieses Mausereignis tritt ein, wenn der Cursor das Element verlässt.
<i>„OnMouseMove“</i>	Dieses Mausereignis tritt ein, wenn der Cursor innerhalb des Elements bewegt wird.

„*OnMouseUp*“ Diese Mausereignis liegt vor, wenn die Maustaste losgelassen wird. Das Drücken der Maustaste geschah vorher, außerhalb des Elements.


### Eingabekonfiguration: Tasten

Mit „*Tasten*“ kann konfiguriert werden, dass beim Ereignis "Tasten" der Wert einer boolschen Projektvariable abhängig vom Mausverhalten gesetzt wird.

Eigenschaft	Beschreibung Wert
„ <i>Variable</i> “	Hier kann eine boolsche Variable zugewiesen werden, deren Wert TRUE wird, wenn die Maustaste gedrückt wird, während der Cursor auf das Element zeigt und die wieder FALSE wird, wenn die Maustaste wieder losgelassen wird.
„ <i>FALSE tasten</i> “	Wenn diese Option aktiviert ist, wird das oben beschriebene Tastverhalten für die eingetragene Variable umgekehrt. Das heißt, wenn die Maustaste gedrückt wird, wird der Variablenwert auf FALSE gesetzt, wenn die Taste wieder losgelassen wird, wird der Variablenwert auf TRUE gesetzt.
„ <i>Beim Betreten tasten, falls Maus gefangen</i> “	Ist diese Option aktiviert, verhält sich der Variablenwert wie unter „ <i>Variable</i> “ beschrieben, solange sich die Maus innerhalb des Elementbereichs befindet. Zusätzlich wird auf ein Ziehen der Maus reagiert. Wird bei gedrückter Maustaste der Elementbereich verlassen, wird die Variable FALSE und wird wieder TRUE, wenn in den Elementbereich zurückgekehrt wird. Es wird also berücksichtigt, dass, solange die Maustaste gedrückt ist, die Maus "gefangen" ist, auch wenn der Elementbereich verlassen wird.

### Eingabekonfiguration: Umschalten

Mit „*Umschalten*“ kann konfiguriert werden, dass bei diesem Ereignis der Wert einer boolschen Projektvariable abhängig vom Mausverhalten gesetzt wird.

Eigenschaft	Beschreibung Wert
„ <i>Variable</i> “	Hier ist eine boolsche Variable zuzuweisen. Die Eingabehilfe kann über Schaltfläche  geöffnet werden.  Der Wert der Variablen wird bei jedem Mausklick auf das Element zwischen TRUE und FALSE umgeschaltet. Wird der Cursor bei gedrückter Maustaste aus dem Element herausbewegt, dann erfolgt kein Toggeln. Damit kann die begonnene Umschalteingabe abgebrochen werden.
„ <i>Umschalten beim Loslassen falls Maus gefangen</i> “	Ist diese Option aktiviert, verhält sich der Variablenwert wie unter „ <i>Variable</i> “ beschrieben, solange sich die Maus innerhalb des Elementbereichs befindet. Zusätzlich wird auf ein Ziehen der Maus reagiert. Wird bei gedrückter Maustaste der Elementbereich verlassen und dann die Maustaste losgelassen, wird die Variable umgeschaltet. Es wird also berücksichtigt, dass, solange die Maustaste gedrückt ist, die Maus "gefangen" ist, auch

wenn der Elementbereich verlassen wird.

## Eingabekonfiguration: Hotkey

Mit „*Hotkey*“ kann eine Taste, auch mit Zusatzaste, definiert und mit einer bestimmten Nachfolgeaktion (MouseDown, MouseUp) verknüpft werden, die bei einem Ereignis der Taste (KeyDown, KeyUp) ausgeführt werden soll. Standardmäßig wird bei KeyDown (Taste drücken) die MouseDown-Aktion ausgeführt und bei KeyUp (Taste loslassen) die MouseUp-Aktion. Das kann nützlich sein, wenn eine Visualisierung sowohl über Maus- als auch Tastatureingaben bedient werden soll, weil die Eingabeaktionen dann nur einmalig konfiguriert werden müssen. Diese Tastenkonfiguration für ein Element wird auch im [„Tastaturkonfiguration“](#) der Visualisierung verwaltet. Änderungen werden immer zwischen diesem und dem Elementeigenschaften-Editor synchronisiert.

Eigenschaft	Beschreibung Wert
„ <i>Taste</i> “	Zuweisen einer Taste. Eine Auswahlliste stellt alle aktuell unterstützten Tasten bereit wie zum Beispiel [ <i>M</i> ]. <a href="#">„Tastaturkonfiguration“</a>
„ <i>Eingabe(n)</i> “	Definieren des Ereignisses, das ausgeführt werden soll, wenn die Taste beziehungsweise Taste und Modifizierer verwendet werden.  Mögliche Werte, die in einer Auswahlliste bereitstehen: <ul style="list-style-type: none"><li>Keine Aktion</li><li>MouseDown-Aktion: OnMouseDown-Aktion, wenn die Taste gedrückt wird.</li><li>MouseUp-Aktion: OnMouseUp-Aktion, wenn die Taste freigelassen wird.</li><li>MouseDown/MouseUp-Aktion: OnMouseUp- und OnMouseDown-Aktion, wenn die Taste gedrückt oder losgelassen wird.</li></ul>
„ <i>Umschalten</i> “	Wenn diese Option aktiviert wird, muss die Taste mit [ <i>Umschalt</i> ] kombiniert verwendet werden
„ <i>Steuerung</i> “	Wenn diese Option aktiviert wird, muss die Taste mit [ <i>Strg</i> ] kombiniert verwendet werden
„ <i>Alt</i> “	Wenn diese Option aktiviert wird, muss die Taste mit [ <i>Alt</i> ] kombiniert verwendet werden

Eigenschaft	Beschreibung Wert
„ <i>Zugriffsrechte</i> “	Mit Klick öffnet <a href="#">„Zugriffsrechte“</a> .  Statusmeldungen: <ul style="list-style-type: none"><li>Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.</li><li>Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.</li></ul>

### Polygon, Linienzug oder Bézierkurve

Polygon, Linienzug und Bézierkurve sind vom gleichen Elementtyp. Sie können nur durch Ändern der Eigenschaft „*Elementtyp*“ gegenseitig konvertiert werden.

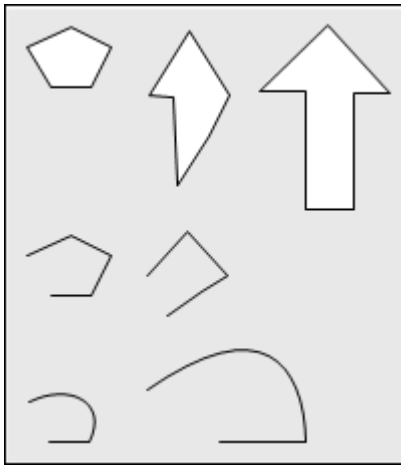



Abb. 55: Polygon, Linienzug, Bézierkurve

#### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungsektor werden folgende Benutzereingaben zur Verfügung gestellt:

Element spezifische Mauseingabe:

- Solch ein Element kann mit einer beliebigen Anzahl an Punkten, die Ausschnitt und Form des Elements bestimmen, definiert werden. Standard sind fünf Punkte: Punkte [0] bis [4]. Zusätzliche Punkte können im Editor eingefügt werden: Ist das Element selektiert und der Fokus auf einen Punkt (kleines Quadrat auf der Umrisslinie), ist die Cursor-Form . Wird „*Strg*“ gedrückt gehalten während auf den fokussierten Punkt geklickt wird, wird ein weiterer Punkt erzeugt. Ein Punkt kann gelöscht werden, wenn [ *Umschalttaste* ] + [ *Strg* ] gedrückt werden, während auf den fokussierten Punkt geklickt wird. In „*Eigenschaften*“ werden gleichzeitig die Änderungen berücksichtigt und *Position* → *Punkte* „aktualisiert“.

#### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

##### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Auszuwählen ist einer aus: <ul style="list-style-type: none"><li>▪ Polygon</li></ul>


- Linienzug
- Bézierkurve

## Position

Solch ein Element kann mit einer beliebigen Anzahl an Punkten, die Ausschnitt und Form des Elements bestimmen, definiert werden. Standard sind fünf Punkte: Punkte [0] bis [4]. Ein Punkt ist bestimmt durch X/Y-Koordinaten in Pixel.



Eigenschaft	Beschreibung Wert
„Punkte“	▪ [ <i>&lt;n&gt;</i> ]:
▪ „[ <i>&lt;n&gt;</i> ]“	▪ „X“: Horizontale Position in Pixel.
▪ „X“	▪ „Y“: Vertikale Position in Pixel.
▪ „Y“	

## Zentrum

Mit Bearbeiten des Wertes wird gleichzeitig das zugehörige Element  im Editor verschoben.

Eigenschaft	Beschreibung Wert
„X“	X-Position des Rotationspunkts des Elements
„Y“	Y-Position des Rotationspunkts des Elements

## Farben

Eine Farbe wird mittels Hexzahl definiert, die sich aus Rot/Grün/Blau (RGB) Anteilen zusammensetzt. Für jede dieser drei Farben sind 256 (0-255) Werte verfügbar. Hier sind die Einstellungen **statisch**. Die Farbe kann aus einer Auswahlliste oder dem Farbauswahldialog, der über Button  öffnet, gewählt werden. Beispiel:  0; 128; 64

Eigenschaft	Beschreibung Wert
„Normalzustand“	Auszuwählen ist eine Rahmen- und Füllfarbe für Normalzustand. Im Fall, dass die Variable definiert in <a href="#">Farbvariablen</a> → <a href="#">Farbumschlag</a> FALSE ist, ist das Element in Normalzustand.
▪ „Rahmenfarbe“	
▪ „Füllfarbe“	
„Alarmzustand“	Auszuwählen ist eine Rahmen- und Füllfarbe, mit der das Element im Alarmzustand dargestellt wird. Dies ist der Fall, wenn die Variable in <a href="#">Farbvariablen</a> → <a href="#">Farbumschlag</a> TRUE ist.
▪ „Rahmenfarbe“	



- „Füllfarbe“

„Farbverlauf verwenden“ Standardmäßig ist die Checkbox deaktiviert. Falls sie aktiviert wird, wird das zugehörige Element mit Farbverlauf gezeichnet.

„Farbverlaufsauswahl“ Dialog [↘](#) „Farbverlaufeditor“ öffnet.

## Aussehen

Eigenschaft	Beschreibung Wert
„Linienstärke“	Stärke der Rahmenlinie in Pixel. 0 kodiert das Gleiche wie 1 und setzt die Linienstärke auf 1 Pixel. Wird kein Rahmen gewünscht, muss man „Linienart“ auf unsichtbar setzen.
„Füllart“	Auszuwählen ist eines dieser Füllarten angewendet in „Füllfarbe“ <ul style="list-style-type: none"> <li>▪ Ausgefüllt: Die Füllfarbe ist sichtbar</li> <li>▪ Unsichtbar: Die Füllfarbe ist unsichtbar</li> </ul>
„Linienart“	Auszuwählen ist eine der Linienarten angewendet bei der Umrisslinie: <ul style="list-style-type: none"> <li>▪ Durchgezogen</li> <li>▪ Striche</li> <li>▪ Punkte</li> <li>▪ Strich Punkt</li> <li>▪ Strich Punkt Punkt</li> <li>▪ Unsichtbar: Umrisslinie ist nicht sichtbar.</li> </ul>

## Texte

Diese Eigenschaften werden für eine statische Definition der Elementbeschriftung verwendet.

Property	Value description
„Text“	Der angegebene Text beschriftet das Element. Ein textformatierender String ist möglich.
„Tooltip“	Der angegebenen Text erscheint als Tooltip des Elements, wenn die Visualisierung im Online-Modus ist.

## Texteigenschaften

Diese Eigenschaften werden für eine statische Definition der Schriftart verwendet.

Eigenschaft	Beschreibung Wert
„Horizontale Ausrichtung“	Definieren der horizontalen Ausrichtung des Textes durch Auswahl aus: <ul style="list-style-type: none"> <li>▪ Links</li> <li>▪ Zentriert</li> </ul>

- Rechts

*„Vertikale Ausrichtung“*

Definieren der vertikalen Ausrichtung des Textes durch Auswahl aus:

- Oben
- Zentriert
- Unten

*„Schriftart“*

Auswahl aus vordefinierten Schriftarten:

- Standard
- Überschrift
- Große
- Titel
- Anmerkung


Mit Klick auf  öffnet sich der Dialog für benutzerdefinierte Eigenschaften der Schriftart.


*„Farbe Schriftart“*

Auszuwählen ist eine Schriftartfarbe für das Element. Entweder aus der Auswahlliste oder durch Setzen im sich öffnenden Dialog, wenn  geklickt wird.


**Absolute Bewegung**

Das Element kann bewegt werden, indem die X- und Y-Position (Pixel) der linken oberen Ecke des Elements durch eine Integer-Variable verändert wird. Hier werden absolute Koordinatenwerte verwendet.

Eigenschaft	Beschreibung Wert
<i>„Bewegung“</i>	<ul style="list-style-type: none"> <li>▪ <i>„X“</i>: Hier ist eine Integer-Variable einzugeben, deren Wert die aktuelle X-Position der linken oberen Ecke des Elements in Pixel definiert; kann verwendet werden, um das Element in X-Richtung zu verschieben (ein positiver Wert verschiebt das Element von links nach rechts).</li> <li>▪ <i>„Y“</i>: Eingabe einer Integer-Variablen, die die aktuelle Y-Position der linken oberen Ecke des Elements in Pixel definiert; kann verwendet werden, um das Element in Y-Richtung zu verschieben (ein positiver Wert verschiebt das Element von oben nach unten).</li> </ul>
<i>„Rotation“</i>	<p>Eingabe einer Integer-Variablen, deren Wert den Winkel (Winkelgrade) definiert, um den das Element um den Rotationspunkt <i>„Zentrum“</i>  rotiert.</p> <p>positiver Wert = mathematisch positiv = im Uhrzeigersinn</p> <p>Zu beachten ist, dass das Element, im Gegensatz zum Verhalten bei <i>„Innere Rotation“</i>, selber nicht rotiert.</p>
<i>„Skalierung“</i>	<p>Eingabe einer Integer-Variablen, deren Wert den aktuellen Skalenwert (Prozent) definiert. Die Elementgröße wird linear entsprechend dieses Wertes verändert. Implizit wird der Wert durch</p>

1000 dividiert, so dass es nicht nötig ist, REAL-Variablen zu verwenden um das Element zu verkleinern. Die Skalierung bezieht sich immer auf den Rotationspunkt „Zentrum“ .

„Innere  
Rotation“

Die hier eingetragene Integer-Variable definiert den Winkel (Winkelgrade), um den das Element um seinen Rotationspunkt gedreht wird: „Zentrum“ .

positiver Werte = mathematisch positiv = im Uhrzeigersinn.

Im Gegensatz zu „Rotation“ rotiert das Element tatsächlich.

## Dynamische Punkte

Dynamische Definition für die Punkte, die das Element definieren.

Eigenschaft	Beschreibung Wert
„Array von Punkten“	<p>Hier ist eine Variable einzugeben, die auf einen Array der Struktur <code>VisuElems.VisuStructPoint</code> zeigt. Die <code>VisuElems.VisuStructPoint</code>-Komponenten <code>iX</code> und <code>iY</code> enthalten die X/Y-Koordinaten eines Punktes. Der Array enthält alle Punkte des Elements und die Struktur kann dynamisch durch eine Projektvariable gefüllt werden.</p> <p>Die Anzahl der Elementpunkte muss explizit in „Anzahl Punkte“ deklariert werden.</p> <p>Beispiel: <code>pPoints : POINTER TO ARRAY[0..100] OF VisuElems.VisuStructPoint;</code></p>
„Anzahl Punkte“	<p>Hier ist eine Integervariable zuzuweisen, die die Anzahl der Punkte des Elements definiert.</p> <p>Beispiel: <code>iCount : INT:=24;</code></p> <p>Das Element benötigt 24 einzelne Punkte. Diese Festlegung ist nötig, weil die Definition der einzelnen Punkte über einen Zeiger erfolgt und dies nicht erlaubt, die Anzahl der Punkte zu steuern.</p>

## Textvariablen

Dies sind dynamische Definitionen. Siehe auch „Dynamische Texte“ bei Verwendung von Textlisten. Für statische Definitionen siehe „Texte“

Eigenschaft	Beschreibung Wert
„Textvariable“	Die zugewiesene Variable definiert den Elementtext ("Textausgabevariable"). Unter <a href="#">„Text und Sprache in Visualisierungen“</a> ist zu finden, wie ein Text zu formatieren ist.
„Tooltipvariable“	Die zugewiesene STRING-Variable enthält den Tooltip-Text für das Element.

## Dynamische Texte

Diese Parameter dienen der Definition von dynamischen Texten, die aus Textlisten stammen (dies erlaubt

Sprachumschaltung). Siehe [Textliste](#) für nähere Informationen zur Verwendung von Textlisten.



Eigenschaft	Beschreibung Wert
„ <i>Textliste</i> “	Name der Textliste (String) wie im POU- oder Geräte-Fenster verwendet; z.B. <code>errorlist</code>
„ <i>Textindex</i> “	Index (ID) des Textes wie in der Textliste definiert (String); Direkte Angabe der ID, z.B. 'M1' oder String-Variable, z.B. <code>error_22</code> .
„ <i>Tooltipindex</i> “	Index (ID) des Tooltip-Textes wie in der Textliste definiert (String); Direkte Angabe der ID, z.B. 'T1' oder String-Variable, z.B. <code>error_Tooltip_22</code> .

## Schriftartvariablen

Diese werden verwendet bei dynamischen Schriftdefinitionen des Elementtextes über Projektvariablen. Statischen Definitionen werden unter [Texteigenschaften](#) → [Schriftart](#) konfiguriert.

Eigenschaft	Beschreibung Wert
„ <i>Schriftname</i> “	Hier ist eine Variable Typ STRING anzugeben, die den Schriftnamen enthält, der für die Beschriftung des Elements verwendet werden soll. (Namensangabe wie im Standard-Schrifttypen-Dialog). Beispiel: <code>plc_prg.font_var font_var := Arial;</code>
„ <i>Größe</i> “	Eingabe einer Variablen Typ INT, die die Größe des Elementtextes in Pixel enthält, so wie im Standarddialog Font. Beispiel: <code>prog1.font_height font_height:=16;</code>
„ <i>Flags</i> “	Eingabe einer Variable Typ DWORD, die über die unten genannten Flag-Werte die Font-Darstellung festlegt. Eine kombinierte Definition kann erreicht werden, indem die jeweiligen Flag-Werte addiert werden und die Summe angegeben wird. <ul style="list-style-type: none"><li>▪ 1: kursiv</li><li>▪ 2: fett</li><li>▪ 4: unterstrichen</li><li>▪ 8: gelöscht</li></ul> <p><code>prog2.font_type: ist font_type:=6</code>, wird der Text fett und unterstrichen dargestellt.</p>
„ <i>Zeichensatz</i> “	Der Zeichensatz, der für den Font verwendet werden soll, kann über die Standard-Zeichensatz-Nummer definiert werden. Mit einer DWORD-Variablen kann diese Nummer angegeben werden (siehe auch die „ <i>Skript</i> “-Definition im Standard-Font-Dialog)
„ <i>Farbe</i> “	Eingabe einer DWORD-Variable, in der die Farbe des Elementtextes festgelegt wird.

## Farben

Eine Farbe wird mittels Hexzahl definiert, die sich aus Rot/Grün/Blau (RGB) Anteilen zusammensetzt. Für jede dieser drei Farben sind 256 (0-255) Werte verfügbar. Hier sind die Einstellungen **statisch**. Die Farbe kann aus einer Auswahlliste oder dem Farbauswahldialog, der über Button  öffnet, gewählt werden. Beispiel:  0; 128; 64

Eigenschaft	Beschreibung Wert
„Normalzustand“	Auszuwählen ist eine Rahmen- und Füllfarbe für Normalzustand. Im Fall, dass die Variable definiert in <i>Farbvariablen</i> → <i>Farbumschlag</i> FALSE ist, ist das Element in Normalzustand.
▪ „Rahmenfarbe“	
▪ „Füllfarbe“	
„Alarmzustand“	Auszuwählen ist eine Rahmen- und Füllfarbe, mit der das Element im Alarmzustand dargestellt wird. Dies ist der Fall, wenn die Variable in <i>Farbvariablen</i> → <i>Farbumschlag</i> TRUE ist.
▪ „Rahmenfarbe“	
▪ „Füllfarbe“	

## Aussehensvariablen

Anzuwenden bei einer **dynamischen** Definition des Aussehens der Kontur und der Füllung des Elements. Statische Definitionen werden in „*Aussehen*“ festgelegt.

Eigenschaft	Beschreibung Wert
„Linienstärke“	Eingabe einer Integervariablen, die die Linienstärke des Elements in Pixel definiert. Dies überschreibt den fixen Wert, der in <i>Aussehen</i> → <i>Linienstärke</i> festgelegt wurde.
„Füllart“	Eingabe einer DWORD-Variablen, die die Füllart des Elements definiert. Die Farbe, die bei „ <i>Farbvariablen</i> “ angegeben ist, kann angezeigt oder ignoriert werden: <ul style="list-style-type: none"> <li>▪ Variablenwert = 0: gefüllt</li> <li>▪ Variablenwert &gt; 0 : unsichtbar, das heißt keine Füllung ist sichtbar</li> </ul> Dies überschreibt den fixen Wert, der in <i>Aussehen</i> → <i>Füllart</i> festgelegt wurde.
„Linienart“	Eingabe einer DWORD-Variablen, die die Kontur der Außenlinie festlegt. <p>Folgende Werte entsprechen folgenden Linienarten:</p> <ul style="list-style-type: none"> <li>▪ 0: Durchgezogen</li> <li>▪ 1: Strich</li> <li>▪ 2: Punkt</li> <li>▪ 3: Strich Punkt</li> <li>▪ 4: Strich Punkt Punkt</li> <li>▪ 8: Unsichtbar. Das heißt, das Element wird ohne Außenlinie dargestellt.</li> </ul>

## Zustandsvariablen

Eigenschaft	Beschreibung Wert
„Unsichtbarkeit“	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.
„Eingaben deaktivieren“	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann hat keine der Elementeingaben eine Auswirkung.

### Eingabekonfiguration

Hier kann definiert werden, welche Folgeaktionen ausgeführt werden, wenn der Benutzer im Online-Modus eine Eingabe im Element vornimmt.

Eingabeereignisse, die ein Element haben kann, sind:

- „OnDialogClosed“
- „OnClick“
- „MouseDown“
- „OnMouseEnter“
- „OnMouseLeave“
- „MouseMove“
- „MouseUp“

Mit Klick auf „Konfigurieren...“ wird der Dialog [„Eingabekonfiguration“](#) geöffnet, der erlaubt Folgeaktionen zuzuweisen. Man klickt in das Feld neben der zu konfigurierenden Eingabe. Für jede selektierte Folgeaktion wird ein Eintrag in der Eigenschaftentabelle erzeugt. Jede Eingabe kann eine beliebige zugeordnete Anzahl an Folgeaktionen haben.

Eigenschaft	Beschreibung Wert
„OnDialogClosed“	Dieses Ereignis tritt ein, wenn innerhalb der Visualisierung eines der für Benutzereingaben vorher geöffneten Dialogfenster geschlossen wird.
„OnClick“	Dieses Mausereignis tritt ein, wenn der Cursor auf ein Element zeigt und ein vollständiger Mausklick (drücken und freigeben der Maustaste) ausgeführt wird.
„MouseDown“	Dieses Mausereignis tritt ein, wenn die Maustaste gedrückt wird während der Cursor auf ein Element zeigt. (Drücken der Maustaste)
„OnMouseEnter“	Dieses Mausereignis tritt ein, wenn der Cursor auf das Element gezogen wird.
„OnMouseLeave“	Dieses Mausereignis tritt ein, wenn der Cursor das Element verlässt.
„MouseMove“	Dieses Mausereignis tritt ein, wenn der Cursor innerhalb des Elements bewegt wird.
„MouseUp“	Dieses Mausereignis liegt vor, wenn die Maustaste losgelassen wird. Das Drücken der Maustaste geschah vorher, außerhalb des Elements.


### Eingabekonfiguration: Tasten

Mit „*Tasten*“ kann konfiguriert werden, dass beim Ereignis "Tasten" der Wert einer booleschen Projektvariable abhängig vom Mausverhalten gesetzt wird.

Eigenschaft	Beschreibung Wert
„ <i>Variable</i> “	Hier kann eine boolesche Variable zugewiesen werden, deren Wert TRUE wird, wenn die Maustaste gedrückt wird, während der Cursor auf das Element zeigt und die wieder FALSE wird, wenn die Maustaste wieder losgelassen wird.
„ <i>FALSE tasten</i> “	Wenn diese Option aktiviert ist, wird das oben beschriebene Tastverhalten für die eingetragene Variable umgekehrt. Das heißt, wenn die Maustaste gedrückt wird, wird der Variablenwert auf FALSE gesetzt, wenn die Taste wieder losgelassen wird, wird der Variablenwert auf TRUE gesetzt.
„ <i>Beim Betreten tasten, falls Maus gefangen</i> “	Ist diese Option aktiviert, verhält sich der Variablenwert wie unter „ <i>Variable</i> “ beschrieben, solange sich die Maus innerhalb des Elementbereichs befindet. Zusätzlich wird auf ein Ziehen der Maus reagiert. Wird bei gedrückter Maustaste der Elementbereich verlassen, wird die Variable FALSE und wird wieder TRUE, wenn in den Elementbereich zurückgekehrt wird. Es wird also berücksichtigt, dass, solange die Maustaste gedrückt ist, die Maus "gefangen" ist, auch wenn der Elementbereich verlassen wird.

#### Eingabekonfiguration: Umschalten

Mit „*Umschalten*“ kann konfiguriert werden, dass bei diesem Ereignis der Wert einer booleschen Projektvariable abhängig vom Mausverhalten gesetzt wird.

Eigenschaft	Beschreibung Wert
„ <i>Variable</i> “	Hier ist eine boolesche Variable zuzuweisen. Die Eingabehilfe kann über Schaltfläche  geöffnet werden.  Der Wert der Variablen wird bei jedem Mausklick auf das Element zwischen TRUE und FALSE umgeschaltet. Wird der Cursor bei gedrückter Maustaste aus dem Element herausbewegt, dann erfolgt kein Toggeln. Damit kann die begonnene Umschalteingabe abgebrochen werden.
„ <i>Umschalten beim Loslassen falls Maus gefangen</i> “	Ist diese Option aktiviert, verhält sich der Variablenwert wie unter „ <i>Variable</i> “ beschrieben, solange sich die Maus innerhalb des Elementbereichs befindet. Zusätzlich wird auf ein Ziehen der Maus reagiert. Wird bei gedrückter Maustaste der Elementbereich verlassen und dann die Maustaste losgelassen, wird die Variable umgeschaltet. Es wird also berücksichtigt, dass, solange die Maustaste gedrückt ist, die Maus "gefangen" ist, auch wenn der Elementbereich verlassen wird.

#### Eingabekonfiguration: Hotkey

Mit „Hotkey“ kann eine Taste, auch mit Zusatzaste, definiert und mit einer bestimmten Nachfolgeaktion (MouseDown, MouseUp) verknüpft werden, die bei einem Ereignis der Taste (KeyDown, KeyUp) ausgeführt werden soll. Standardmäßig wird bei KeyDown (Taste drücken) die MouseDown-Aktion ausgeführt und bei KeyUp (Taste loslassen) die MouseUp-Aktion. Das kann nützlich sein, wenn eine Visualisierung sowohl über Maus- als auch Tastatureingaben bedient werden soll, weil die Eingabeaktionen dann nur einmalig konfiguriert werden müssen. Diese Tastenkonfiguration für ein Element wird auch im [„Tastaturkonfiguration“](#) der Visualisierung verwaltet. Änderungen werden immer zwischen diesem und dem Elementeigenschaften-Editor synchronisiert.

Eigenschaft	Beschreibung Wert
„Taste“	Zuweisen einer Taste. Eine Auswahlliste stellt alle aktuell unterstützten Tasten bereit wie zum Beispiel [ M ]. <a href="#">„Tastaturkonfiguration“</a>
„Eingabe(n)“	Definieren des Ereignisses, das ausgeführt werden soll, wenn die Taste beziehungsweise Taste und Modifizierer verwendet werden.  Mögliche Werte, die in einer Auswahlliste bereitstehen: <ul style="list-style-type: none"> <li>Keine Aktion</li> <li>MouseDown-Aktion: OnMouseDown-Aktion, wenn die Taste gedrückt wird.</li> <li>MouseUp-Aktion: OnMouseUp-Aktion, wenn die Taste freigelassen wird.</li> <li>MouseDown/MouseUp-Aktion: OnMouseUp- und OnMouseDown-Aktion, wenn die Taste gedrückt oder losgelassen wird.</li> </ul>
„Umschalten“	Wenn diese Option aktiviert wird, muss die Taste mit [ Umschalt ] kombiniert verwendet werden
„Steuerung“	Wenn diese Option aktiviert wird, muss die Taste mit [ Strg ] kombiniert verwendet werden
„Alt“	Wenn diese Option aktiviert wird, muss die Taste mit [ Alt ] kombiniert verwendet werden

Eigenschaft	Beschreibung Wert
„Zugriffsrechte“	Mit Klick öffnet <a href="#">„Zugriffsrechte“</a> .  Statusmeldungen: <ul style="list-style-type: none"> <li>Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.</li> <li>Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.</li> </ul>



## Kreisektor

Der Kreisektor, der standardmäßig im Visualisierungseditor eingefügt wird, ist ein Vollkreis mit der Mittelposition an dem Punkt, an dem das Element fallen gelassen wird.



Abb. 56: Kreisektor

### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungseditor werden folgende Benutzereingaben zur Verfügung gestellt:

Element spezifische Mauseingabe:

- Anfangs- und Endpunkt des Kreisektors können mit Drag-and-Drop der kleinen Quadrate auf der Kreislinie im Editor verschoben werden. Die zugehörigen Eigenschaften „Anfang“ und „Ende“ werden gleichzeitig aktualisiert.

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften


Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„Elementname“	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„Elementtyp“	Kreisektor


#### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„X“	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„Y“	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„Breite“	Horizontale Breite des Elements.
„Höhe“	Vertikale Höhe des Elements.



„Anfang“	Hier ist der Anfang der Kreissektorlinie zu definieren.
„Ende“	Definiert das Ende der Kreissektorlinie.
„Variable für Anfang“	Eingabe einer Variablen, wenn der Anfang der Kreissektorlinie dynamisch mit Variable definiert werden soll. Für Hilfe vom Eingabeassistenten ist  zu klicken.
„Variable für Ende“	Eingabe einer Variablen, eventuell auch mit Hilfe des Eingabeassistenten, falls das Ende der Kreissektorlinie dynamisch mit Variable definiert werden soll.
„Nur Kreisbogen anzeigen“	Falls die Option aktiviert ist, wird der Kreissektor ohne Radiuslinie für Anfang und Ende dargestellt,


## Zentrum

Mit Bearbeiten des Wertes wird gleichzeitig das zugehörige Element  im Editor verschoben.

Eigenschaft	Beschreibung Wert
„X“	X-Position des Rotationspunkts des Elements
„Y“	Y-Position des Rotationspunkts des Elements


## Farben

Eine Farbe wird mittels Hexzahl definiert, die sich aus Rot/Grün/Blau (RGB) Anteilen zusammensetzt. Für jede dieser drei Farben sind 256 (0-255) Werte verfügbar. Hier sind die Einstellungen **statisch**. Die Farbe kann aus einer Auswahlliste oder dem Farbauswahldialog, der über Button  öffnet, gewählt werden. Beispiel:  0; 128; 64

Eigenschaft	Beschreibung Wert
„Normalzustand“	Auszuwählen ist eine Rahmen- und Füllfarbe für Normalzustand. Im Fall, dass die Variable definiert in <a href="#">Farbvariablen</a> → <a href="#">Farbumschlag</a> FALSE ist, ist das Element in Normalzustand.
▪ „Rahmenfarbe“	
▪ „Füllfarbe“	
„Alarmzustand“	Auszuwählen ist eine Rahmen- und Füllfarbe, mit der das Element im Alarmzustand dargestellt wird. Dies ist der Fall, wenn die Variable in <a href="#">Farbvariablen</a> → <a href="#">Farbumschlag</a> TRUE
▪ „Rahmenfarbe“	ist.
▪ „Füllfarbe“	
„Farbverlauf verwenden“	Standardmäßig ist die Checkbox deaktiviert. Falls sie aktiviert wird, wird das zugehörige Element mit Farbverlauf gezeichnet.
„Farbverlaufsauswahl“	Dialog  „ <a href="#">Farbverlaufeditor</a> “ öffnet.

## Aussehen

Eigenschaft	Beschreibung Wert
„ <i>Linienstärke</i> “	Stärke der Rahmenlinie in Pixel. 0 kodiert das Gleiche wie 1 und setzt die Linienstärke auf 1 Pixel. Wird kein Rahmen gewünscht, muss man „ <i>Linienart</i> “ auf <code>unsichtbar</code> setzen.
„ <i>Füllart</i> “	Auszuwählen ist eines dieser Füllarten angewendet in „ <i>Füllfarbe</i> “ <ul style="list-style-type: none"> <li>▪ Ausgefüllt: Die Füllfarbe ist sichtbar</li> <li>▪ Unsichtbar: Die Füllfarbe ist unsichtbar</li> </ul>
„ <i>Linienart</i> “	Auszuwählen ist eine der Linienarten angewendet bei der Umrisslinie: <ul style="list-style-type: none"> <li>▪ Durchgezogen</li> <li>▪ Striche</li> <li>▪ Punkte</li> <li>▪ Strich Punkt</li> <li>▪ Strich Punkt Punkt</li> <li>▪ Unsichtbar: Umrisslinie ist nicht sichtbar.</li> </ul>

	Die „ <i>Linienstärke</i> “ und die „ <i>Linienart</i> “ wirken sich auf Windows CE basierenden Systemen nicht aus.
--	---

## Texte

Diese Eigenschaften werden für eine statische Definition der Elementbeschriftung verwendet.

Property	Value description
„ <i>Text</i> “	Der angegebene Text beschriftet das Element. Ein textformatierender String ist möglich.
„ <i>Tooltip</i> “	Der angegebenen Text erscheint als Tooltip des Elements, wenn die Visualisierung im Online-Modus ist.

## Texteigenschaften

Diese Eigenschaften werden für eine statische Definition der Schriftart verwendet.

Eigenschaft	Beschreibung Wert
„ <i>Horizontale Ausrichtung</i> “	Definieren der horizontalen Ausrichtung des Textes durch Auswahl aus: <ul style="list-style-type: none"> <li>▪ Links</li> <li>▪ Zentriert</li> <li>▪ Rechts</li> </ul>
„ <i>Vertikale</i> “	Definieren der vertikalen Ausrichtung des Textes durch Auswahl aus:

### „Ausrichtung“

- Oben
- Zentriert
- Unten

### „Schriftart“

Auswahl aus vordefinierten Schriftarten:

- Standard
- Überschrift
- Große
- Titel
- Anmerkung



Mit Klick auf  öffnet sich der Dialog für benutzerdefinierte Eigenschaften der Schriftart.

### „Farbe Schriftart“

Auszuwählen ist eine Schriftartfarbe für das Element. Entweder aus der Auswahlliste oder durch Setzen im sich öffnenden Dialog, wenn  geklickt wird.

## Absolute Bewegung

Das Element kann bewegt werden, indem die X- und Y-Position (Pixel) der linken oberen Ecke des Elements durch eine Integer-Variable verändert werden. Hier werden absolute Koordinatenwerte verwendet.

Eigenschaft	Beschreibung Wert
„Bewegung“	<ul style="list-style-type: none"><li>▪ „X“: Hier ist eine Integer-Variable einzugeben, deren Wert die aktuelle X-Position der linken oberen Ecke des Elements in Pixel definiert; kann verwendet werden, um das Element in X-Richtung zu verschieben (ein positiver Wert verschiebt das Element von links nach rechts).</li><li>▪ „Y“: Eingabe einer Integer-Variablen, die die aktuelle Y-Position der linken oberen Ecke des Elements in Pixel definiert; kann verwendet werden, um das Element in Y-Richtung zu verschieben (ein positiver Wert verschiebt das Element von oben nach unten).</li></ul>
„Skalierung“	Eingabe einer Integer-Variablen, deren Wert den aktuellen Skalenwert (Prozent) definiert. Die Elementgröße wird linear entsprechend dieses Wertes verändert. Implizit wird der Wert durch 1000 dividiert, so dass es nicht nötig ist, REAL-Variablen zu verwenden um das Element zu verkleinern. Die Skalierung bezieht sich immer auf den Rotationspunkt „Zentrum“  .
„Innere Rotation“	Die hier eingetragene Integer-Variable definiert den Winkel (Winkelgrade), um den das Element um seinen Rotationspunkt gedreht wird: „Zentrum“  .

positiver Werte = mathematisch positiv = Uhrzeigersinn.

Im Gegensatz zu „Rotation“ rotiert das Element tatsächlich.

## Textvariablen

Dies sind dynamische Definitionen. Siehe auch „[Dynamische Texte](#)“ bei Verwendung von Textlisten. Für statische Definitionen siehe „[Texte](#)“

Eigenschaft	Beschreibung Wert
„ <a href="#">Textvariable</a> “	Die zugewiesene Variable definiert den Elementtext ("Textausgabevariable"). Unter <a href="#">„Text und Sprache in Visualisierungen“</a> ist zu finden, wie ein Text zu formatieren ist.
„ <a href="#">Tooltipvariable</a> “	Die zugewiesene STRING-Variable enthält den Tooltip-Text für das Element.

### Dynamische Texte

Diese Parameter dienen der Definition von dynamischen Texten, die aus Textlisten stammen (dies erlaubt Sprachumschaltung). Siehe [Textliste](#) für nähere Informationen zur Verwendung von Textlisten.

Eigenschaft	Beschreibung Wert
„ <a href="#">Textliste</a> “	Name der Textliste (String) wie im POU- oder Geräte-Fenster verwendet; z.B. <code>errorlist</code>
„ <a href="#">Textindex</a> “	Index (ID) des Textes wie in der Textliste definiert (String); Direkte Angabe der ID, z.B. 'M1' oder String-Variable, z.B. <code>error_22</code> .
„ <a href="#">Tooltipindex</a> “	Index (ID) des Tooltip-Textes wie in der Textliste definiert (String); Direkte Angabe der ID, z.B. 'T1' oder String-Variable, z.B. <code>error_Tooltip_22</code> .

### Schriftartvariablen

Diese werden verwendet bei dynamischen Schriftdefinitionen des Elementtextes über Projektvariablen. Statischen Definitionen werden unter [Texteigenschaften](#) → [Schriftart](#) konfiguriert.

Eigenschaft	Beschreibung Wert
„ <a href="#">Schriftname</a> “	Hier ist eine Variable Typ STRING anzugeben, die den Schriftnamen enthält, der für die Beschriftung des Elements verwendet werden soll. (Namensangabe wie im Standard-Schrifttypen-Dialog). Beispiel: <code>plc_prg.font_var font_var := Arial;</code>
„ <a href="#">Größe</a> “	Eingabe einer Variablen Typ INT, die die Größe des Elementtextes in Pixel enthält, so wie im Standarddialog Font. Beispiel: <code>prog1.font_height font_height:=16;</code>
„ <a href="#">Flags</a> “	Eingabe einer Variable Typ DWORD, die über die unten genannten Flag-Werte die Font-Darstellung festlegt. Eine kombinierte Definition kann erreicht werden, indem die jeweiligen Flag-Werte addiert werden und die Summe angegeben wird. <ul style="list-style-type: none"> <li>▪ 1: kursiv</li> <li>▪ 2: fett</li> <li>▪ 4: unterstrichen</li> <li>▪ 8: gelöscht</li> </ul>

`prog2.font_type: ist font_type:=6`, wird der Text fett und unterstrichen dargestellt.

„*Zeichensatz*“ Der Zeichensatz, der für den Font verwendet werden soll, kann über die Standard-Zeichensatz-Nummer definiert werden. Mit einer DWORD-Variablen kann diese Nummer angegeben werden (siehe auch die „*Skript*“-Definition im Standard-Font-Dialog)

„*Farbe*“ Eingabe einer DWORD-Variable, in der die Farbe des Elementtextes festgelegt wird.

## Farbvariablen

Angewandt for **dynamische** Definitionen der Elementfarbe mittels einer Projektvariablen.

Eine Farbe wird mittels Hexzahl definiert, die sich aus Rot/Grün/Blau (RGB) Anteilen zusammensetzt. Es ist eine Projektvariable Typ DWORD anzugeben, deren Wert die Farbe definiert. (zum Beispiel `prog1.dwFillColor`).

Die Variable enthält den Farbwert in hexadezimalen Format. Um das gesamte DWORD auszufüllen, müssen die ersten beiden Ziffern nach "16#" mit 0 gesetzt werden: `16#00RRGGBB`

### Farbwerte

```
dwFillColor := 16#008FE03F;
```

Hexzahl:

- 8F: rot
- E0: grün
- 3F: blau

Eigenschaft	Beschreibung Wert
„ <i>Farbumschlag</i> “	Hier ist eine boolesche Variable angeben, die das Umschalten der Elementfarbe zwischen „ <i>Normalzustand</i> “ (Variable = FALSE) und „ <i>Alarmzustand</i> “ (Variable = TRUE) steuert.
„ <i>Normalzustand</i> “	Eingabe von DWORD-Variablen, die die Rahmen- und Füllfarbe des Elements bestimmen. Sie überschreiben die Werte, die aktuell in <i>Farben</i> → <i>Normalzustand</i> definiert sind. Die Werte
▪ „ <i>Rahmenfarbe</i> “	in den Projektvariablen werden verwendet, wenn die Variable definiert in „ <i>Farbumschlag</i> “ FALSE ist.
▪ „ <i>Füllfarbe</i> “	
„ <i>Alarmzustand</i> “	Eingabe von DWORD-Variablen, die die Rahmen- und Füllfarbe des Elements im Alarmzustand bestimmen. Sie überschreiben die Werte, die aktuell in <i>Farben</i> →
▪ „ <i>Rahmenfarbe</i> “	<i>Alarmzustand</i> definiert sind. Die Werte in den Projektvariablen werden verwendet, wenn die Variable definiert in „ <i>Farbumschlag</i> “ TRUE ist.
▪ „ <i>Füllfarbe</i> “	

## Aussehensvariablen

Anzuwenden bei einer **dynamischen** Definition des Aussehens der Kontur und der Füllung des Elements. Statische Definitionen werden in „*Aussehen*“ festgelegt.

Eigenschaft	Beschreibung Wert
„ <i>Linienstärke</i> “	Eingabe einer Integervariablen, die die Linienstärke des Elements in Pixel definiert. Dies überschreibt den fixen Wert, der in <i>Aussehen</i> → <i>Linienstärke</i> festgelegt wurde.
„ <i>Füllart</i> “	Eingabe einer DWORD-Variablen, die die Füllart des Elements definiert. Die Farbe, die bei „ <i>Farbvariablen</i> “ angegeben ist, kann angezeigt oder ignoriert werden: <ul style="list-style-type: none"> <li>▪ Variablenwert = 0: gefüllt</li> <li>▪ Variablenwert &gt; 0 : unsichtbar, das heißt keine Füllung ist sichtbar</li> </ul> <p>Dies überschreibt den fixen Wert, der in <i>Aussehen</i> → <i>Füllart</i> festgelegt wurde.</p>
„ <i>Linienart</i> “	Eingabe einer DWORD-Variablen, die die Kontur der Außenlinie festlegt. <p>Folgende Werte entsprechen folgenden Linienarten:</p> <ul style="list-style-type: none"> <li>▪ 0: Durchgezogen</li> <li>▪ 1: Strich</li> <li>▪ 2: Punkt</li> <li>▪ 3: Strich Punkt</li> <li>▪ 4: Strich Punkt Punkt</li> <li>▪ 8: Unsichtbar. Das heißt, das Element wird ohne Außenlinie dargestellt.</li> </ul>

### Zustandsvariablen

Eigenschaft	Beschreibung Wert
„ <i>Unsichtbarkeit</i> “	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.
„ <i>Eingaben deaktivieren</i> “	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann hat keine der Elementeingaben eine Auswirkung.

### Eingabekonfiguration

Hier kann definiert werden, welche Folgeaktionen ausgeführt werden, wenn der Benutzer im Online-Modus eine Eingabe im Element vornimmt.

Eingabeereignisse, die ein Element haben kann, sind:

- „*OnDialogClosed*“
- „*OnMouseClicked*“

- „OnMouseDown“
- „OnMouseEnter“
- „OnMouseLeave“
- „OnMouseMove“
- „OnMouseUp“

Mit Klick auf „Konfigurieren...“ wird der Dialog [„Eingabekonfiguration“](#) geöffnet, der erlaubt Folgeaktionen zuzuweisen. Man klickt in das Feld neben der zu konfigurierenden Eingabe. Für jede selektierte Folgeaktion wird ein Eintrag in der Eigenschaftentabelle erzeugt. Jede Eingabe kann eine beliebige zugeordnete Anzahl an Folgeaktionen haben.

Eigenschaft	Beschreibung Wert
„OnDialogClosed“	Dieses Ereignis tritt ein, wenn innerhalb der Visualisierung eines der für Benutzereingaben vorher geöffneten Dialogfenster geschlossen wird.
„OnMouseClicked“	Dieses Mausereignis tritt ein, wenn der Cursor auf ein Element zeigt und ein vollständiger Mausklick (drücken und freigeben der Maustaste) ausgeführt wird.
„OnMouseDown“	Dieses Mausereignis tritt ein, wenn die Maustaste gedrückt wird während der Cursor auf ein Element zeigt. (Drücken der Maustaste)
„OnMouseEnter“	Dieses Mausereignis tritt ein, wenn der Cursor auf das Element gezogen wird.
„OnMouseLeave“	Dieses Mausereignis tritt ein, wenn der Cursor das Element verlässt.
„OnMouseMove“	Dieses Mausereignis tritt ein, wenn der Cursor innerhalb des Elements bewegt wird.
„OnMouseUp“	Diese Mausereignis liegt vor, wenn die Maustaste losgelassen wird. Das Drücken der Maustaste geschah vorher, außerhalb des Elements.

### Eingabekonfiguration: Tasten

Mit „Tasten“ kann konfiguriert werden, dass beim Ereignis "Tasten" der Wert einer booleschen Projektvariable abhängig vom Mausverhalten gesetzt wird.

Eigenschaft	Beschreibung Wert
„Variable“	Hier kann eine boolesche Variable zugewiesen werden, deren Wert TRUE wird, wenn die Maustaste gedrückt wird, während der Cursor auf das Element zeigt und die wieder FALSE wird, wenn die Maustaste wieder losgelassen wird.
„FALSE tasten“	Wenn diese Option aktiviert ist, wird das oben beschriebene Tastverhalten für die eingetragene Variable umgekehrt. Das heißt, wenn die Maustaste gedrückt wird, wird der Variablenwert auf FALSE gesetzt, wenn die Taste wieder losgelassen wird, wird der Variablenwert auf TRUE gesetzt.
„Beim Betreten tasten, falls Maus“	Ist diese Option aktiviert, verhält sich der Variablenwert wie unter „Variable“ beschrieben, solange sich die Maus innerhalb des Elementbereichs befindet. Zusätzlich wird auf ein




*gefangen*

Ziehen der Maus reagiert. Wird bei gedrückter Maustaste der Elementbereich verlassen, wird die Variable FALSE und wird wieder TRUE, wenn in den Elementbereich zurückgekehrt wird. Es wird also berücksichtigt, dass, solange die Maustaste gedrückt ist, die Maus "gefangen" ist, auch wenn der Elementbereich verlassen wird.

### Eingabekonfiguration: Umschalten

Mit „*Umschalten*“ kann konfiguriert werden, dass bei diesem Ereignis der Wert einer boolschen Projektvariable abhängig vom Mausverhalten gesetzt wird.

Eigenschaft	Beschreibung Wert
<i>„Variable“</i>	<p>Hier ist eine boolsche Variable zuzuweisen. Die Eingabehilfe kann über Schaltfläche  geöffnet werden.</p> <p>Der Wert der Variablen wird bei jedem Mausklick auf das Element zwischen TRUE und FALSE umgeschaltet. Wird der Cursor bei gedrückter Maustaste aus dem Element herausbewegt, dann erfolgt kein Toggeln. Damit kann die begonnene Umschalteingabe abgebrochen werden.</p>
<i>„Umschalten beim Loslassen falls Maus gefangen“</i>	<p>Ist diese Option aktiviert, verhält sich der Variablenwert wie unter <i>„Variable“</i> beschrieben, solange sich die Maus innerhalb des Elementbereichs befindet. Zusätzlich wird auf ein Ziehen der Maus reagiert. Wird bei gedrückter Maustaste der Elementbereich verlassen und dann die Maustaste losgelassen, wird die Variable umgeschaltet. Es wird also berücksichtigt, dass, solange die Maustaste gedrückt ist, die Maus "gefangen" ist, auch wenn der Elementbereich verlassen wird.</p>

### Eingabekonfiguration: Hotkey

Mit „*Hotkey*“ kann eine Taste, auch mit Zusatzstufe, definiert und mit einer bestimmten Nachfolgeaktion (MouseDown, MouseUp) verknüpft werden, die bei einem Ereignis der Taste (KeyDown, KeyUp) ausgeführt werden soll. Standardmäßig wird bei KeyDown (Taste drücken) die MouseDown-Aktion ausgeführt und bei KeyUp (Taste loslassen) die MouseUp-Aktion. Das kann nützlich sein, wenn eine Visualisierung sowohl über Maus- als auch Tastatureingaben bedient werden soll, weil die Eingabeaktionen dann nur einmalig konfiguriert werden müssen. Diese Tastenkonfiguration für ein Element wird auch im [„Tastaturkonfiguration“](#) der Visualisierung verwaltet. Änderungen werden immer zwischen diesem und dem Elementeigenschaften-Editor synchronisiert.

Eigenschaft	Beschreibung Wert
<i>„Taste“</i>	<p>Zuweisen einer Taste. Eine Auswahlliste stellt alle aktuell unterstützten Tasten bereit wie zum Beispiel [ M ]. <a href="#">„Tastaturkonfiguration“</a></p>
<i>„Eingabe(n)“</i>	<p>Definieren des Ereignisses, das ausgeführt werden soll, wenn die Taste beziehungsweise Taste und Modifizierer verwendet werden.</p>

Mögliche Werte, die in einer Auswahlliste bereitstehen:

- Keine Aktion
- MouseDown-Aktion: OnMouseDown-Aktion, wenn die Taste gedrückt wird.
- MouseUp-Aktion: OnMouseUp-Aktion, wenn die Taste freigelassen wird.
- MouseDown/MouseUp-Aktion: OnMouseUp- und OnMouseDown-Aktion, wenn die Taste gedrückt oder losgelassen wird.

„Umschalten“ Wenn diese Option aktiviert wird, muss die Taste mit [ *Umschalt* ] kombiniert verwendet werden

„Steuerung“ Wenn diese Option aktiviert wird, muss die Taste mit [ *Strg* ] kombiniert verwendet werden

„Alt“ Wenn diese Option aktiviert wird, muss die Taste mit [ *Alt* ] kombiniert verwendet werden

Eigenschaft	Beschreibung Wert
-------------	-------------------


„Zugriffsrechte“ Mit Klick öffnet [„Zugriffsrechte“](#).

Statusmeldungen:

- Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.
- Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.

## Bild

Mit diesem Element kann in der Visualisierung ein Bild eingefügt werden. Ein Bild wird dem Bildelement zugewiesen, indem eine Bild-ID und, um eindeutigen Zugriff zu erhalten, der Name der Bildersammlung spezifiziert wird. Man kann auch dynamisch konfigurieren, indem man eine Projektvariable, deren Wert die aktuelle Bild-ID ist, für dynamisches Umschalten der Bilder zuweist.

	Verzeichnisse, die Bilder für die Verwendung in Visualisierungen bereitstellen, sind spezifiziert in <a href="#">„Projekteinstellungen“</a> Kategorie „ <i>Visualisierung</i> “.
---	--

Ein **Hintergrundbild** für die gesamte Visualisierung kann mit dem Befehl [„Hintergrund“](#) definiert werden.

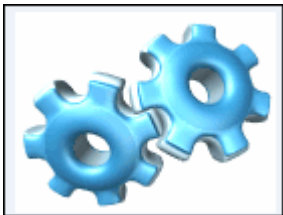


Abb. 57: Bild "Zahnkranz"

### ▲ Benutzereingabe beim Editieren eines Elements


Beim Editieren eines Elements im Visualisierungseditor werden folgende Benutzereingaben zur Verfügung gestellt:

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'


Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung	Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.	
„ <i>Elementtyp</i> “		Bild
„ <i>Statische ID</i> “	Bezeichner der Image-Datei für eine statische Definition.  Geben Sie die ID der Bilddatei ein, wie sie in der entsprechenden Bildersammlung definiert ist (STRING); der Name der Sammlung sollte dabei vorangestellt werden, um den Eintrag eindeutig zu machen (dies ist nicht nötig, wenn die Bilddatei im GlobalImagePool verwaltet wird, da diese Sammlung immer als erste durchsucht wird). Zum Beispiel: <code>imagepool2.button_image</code> . Mit Klick auf  öffnet die Eingabehilfe und listet alle verfügbaren Bildersammlungen mit ihren Bildern auf.	

Siehe [Bildersammlung](#) für weitere Informationen über Verwalten von Bilddateien in Bildersammlungen.

- „*Rahmen zeichnen*“ Wenn diese Option aktiviert ist, wird die Bilddatei mit einem Rahmen dargestellt.
- „*Abschneiden*“ Wenn diese Option zusammen mit „*Skalierungsart*“: *Unskaliert* aktiviert ist, wird im Falle eines Frame-Elements nur der Teil der Visualisierung, der in den Frame passt, dargestellt.
- „*Transparent*“ Ist die Option aktiviert, wird die Farbe, die in Eigenschaft „*Transparenzfarbe*“ angegeben ist, transparent dargestellt.
- „*Transparenzfarbe*“ Die Schaltfläche  öffnet den Farbauswahldialog zur Auswahl einer Farbe, die transparent dargestellt werden soll, wenn Option „*Transparent*“ aktiviert ist.
- „*Skalierungsart*“ Zu definieren ist hier, wie die Bilddatei auf Änderungen der Elementrahmengröße reagieren soll.
- *Isotropisch*: Das Bild behält seine Proportionen; d.h. das Verhältnis Höhe-Breite bleibt erhalten, auch wenn Höhe und Breite des Elementrahmens separat verändert werden.  
Beachten Sie folgendes, wenn Sie **die Ausrichtung der Elemente zueinander auch innerhalb eines skalierten "Frame"-Elements beibehalten** möchten. Beispielsweise haben Sie eine Lampe zentriert oberhalb eines Schalters angeordnet, und sie soll in dieser horizontal zentrierten Position bleiben, auch wenn sich die Größe des Frames verändert. Um unerwünschte horizontale oder vertikale Verschiebungen zu vermeiden, definieren Sie explizit die Horizontale bzw. Vertikale Ausrichtung mit "Zentriert" (siehe unten).
  - *Anisotropisch*: Das Bild passt sich der Größe des Elementrahmens an; d.h. Höhe und Breite können unabhängig voneinander verändert werden.
  - *Unskaliert*: Das Bild behält die Originalgröße, auch wenn sich die Größe des Elementrahmens ändert. Zu berücksichtigen ist auch, ob auch Option „*Abschneiden*“ aktiviert ist.
- Horizontale Ausrichtung Nur verfügbar, wenn die Option "Experte" für den Eigenschaften-Editor aktiviert ist und wenn die Skalierungsart des Bildes "Isotropisch" ist. Dient dazu, die horizontale Ausrichtung zu einem anderen Element (wie in der Basis-Visualisierung definiert) explizit aufrechtzuerhalten, wenn die Visualisierung innerhalb eines skalierten Frames verwendet wird. Siehe oben: Einstellung "Isotropisch"
- Links
  - Zentriert
  - Rechts

Vertikale Ausrichtung Nur verfügbar, wenn die Option "Experte" für den Eigenschaften-Editor aktiviert ist und wenn die Skalierungsart des Bildes "Isotropisch" ist. Dient dazu, die vertikale Ausrichtung zu einem anderen Element (wie in der Basis-Visualisierung definiert) explizit aufrechtzuerhalten, wenn die Visualisierung innerhalb eines skalierten Frames verwendet wird. Siehe oben: Einstellung "Isotropisch"

- Oben
- Zentriert
- Unten

## Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.



Eigenschaft	Beschreibung Wert
„X“	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„Y“	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„Breite“	Horizontale Breite des Elements.
„Höhe“	Vertikale Höhe des Elements.

## Zentrum

Mit Bearbeiten des Wertes wird gleichzeitig das zugehörige Element  im Editor verschoben.

Eigenschaft	Beschreibung Wert
„X“	X-Position des Rotationspunkts des Elements
„Y“	Y-Position des Rotationspunkts des Elements

## Farben

Eine Farbe wird mittels Hexzahl definiert, die sich aus Rot/Grün/Blau (RGB) Anteilen zusammensetzt. Für jede dieser drei Farben sind 256 (0-255) Werte verfügbar. Hier sind die Einstellungen **statisch**. Die Farbe kann aus einer Auswahlliste oder dem Farbauswahldialog, der über Button  öffnet, gewählt werden. Beispiel:  0; 128; 64

Eigenschaft	Beschreibung Wert
„Farbe“	Hier ist eine Farbe, mit der das Element im Normalzustand gefüllt wird, auszuwählen. Dies ist der Fall, wenn die Variable in <i>Farbvariablen</i> → <i>Farbumschlag</i> FALSE ist.

Mit Editieren des Werts ändert das Element gleichzeitig seine Farbe.

„*Alarmfarbe*“ Auszuwählen ist eine Farbe, mit der das Element gefüllt wird, wenn das Element in Alarmzustand ist. Dies ist der Fall, wenn die Variable in *Farbvariablen* → *Farbumschlag* TRUE ist.

## Aussehen

Eigenschaft	Beschreibung Wert
„ <i>Linienstärke</i> “	Stärke der Rahmenlinie in Pixel. 0 kodiert das Gleiche wie 1 und setzt die Linienstärke auf 1 Pixel. Soll kein Rahmen verwendet werden, muss „ <i>Linienart</i> “ auf <code>unsichtbar</code> setzen werden.
„ <i>Linienart</i> “	Auszuwählen ist eine der Linienarten angewendet bei der Umrisslinie: <ul style="list-style-type: none"><li>▪ Durchgezogen</li><li>▪ Striche</li><li>▪ Punkte</li><li>▪ Strich Punkt</li><li>▪ Strich Punkt Punkt</li><li>▪ Unsichtbar: Dann ist die Umrisslinie unsichtbar.</li></ul>

## Texte

Diese Eigenschaften werden für eine statische Definition der Elementbeschriftung verwendet.

Property	Value description
„ <i>Text</i> “	Der angegebene Text beschriftet das Element. Ein textformatierender String ist möglich.
„ <i>Tooltip</i> “	Der angegebenen Text erscheint als Tooltip des Elements, wenn die Visualisierung im Online-Modus ist.

## Texteigenschaften

Diese Eigenschaften werden für eine statische Definition der Schriftart verwendet.

Eigenschaft	Beschreibung Wert
„ <i>Horizontale Ausrichtung</i> “	Definieren der horizontalen Ausrichtung des Textes durch Auswahl aus: <ul style="list-style-type: none"><li>▪ Links</li><li>▪ Zentriert</li><li>▪ Rechts</li></ul>
„ <i>Vertikale Ausrichtung</i> “	Definieren der vertikalen Ausrichtung des Textes durch Auswahl aus: <ul style="list-style-type: none"><li>▪ Oben</li><li>▪ Zentriert</li><li>▪ Unten</li></ul>

### „Schriftart“

Auswahl aus vordefinierten Schriftarten:

- Standard
- Überschrift
- Große
- Titel
- Anmerkung

Mit Klick auf  öffnet sich der Dialog für benutzerdefinierte Eigenschaften der Schriftart.

### „Farbe Schriftart“

Auszuwählen ist eine Schriftartfarbe für das Element. Entweder aus der Auswahlliste oder durch Setzen im sich öffnenden Dialog, wenn  geklickt wird.

### Bild ID Variable

Eigenschaft	Beschreibung Wert
„Bild ID“	Hier ist eine Projektvariable, die die Bild-ID enthält, zuzuweisen. Siehe auch <a href="#">Bildersammlung</a> .

### Absolute Bewegung

Das Element kann bewegt werden, indem die X- und Y-Position (Pixel) der linken oberen Ecke des Elements durch eine Integer-Variable verändert werden. Hier werden absolute Koordinatenwerte verwendet.

Eigenschaft	Beschreibung Wert
„Bewegung“	<ul style="list-style-type: none"><li>▪ „X“: Hier ist eine Integer-Variable einzugeben, deren Wert die aktuelle X-Position der linken oberen Ecke des Elements in Pixel definiert; kann verwendet werden, um das Element in X-Richtung zu verschieben (ein positiver Wert verschiebt das Element von links nach rechts).</li><li>▪ „Y“: Eingabe einer Integer-Variablen, die die aktuelle Y-Position der linken oberen Ecke des Elements in Pixel definiert; kann verwendet werden, um das Element in Y-Richtung zu verschieben (ein positiver Wert verschiebt das Element von oben nach unten).</li></ul>


### „Rotation“

Eingabe einer Integer-Variablen, deren Wert den Winkel (Winkelgrade) definiert, um den das Element um den Rotationspunkt „Zentrum“  rotiert.

positiver Wert = mathematisch positiv = im Uhrzeigersinn

Zu beachten ist, dass das Element, im Gegensatz zum Verhalten bei „Innere Rotation“, selber nicht rotiert.

### „Skalierung“

Eingabe einer Integer-Variablen, deren Wert den aktuellen Skalenwert (Prozent) definiert. Die Elementgröße wird linear entsprechend dieses Wertes verändert. Implizit wird der Wert durch 1000 dividiert, so dass es nicht nötig ist, REAL-Variablen zu verwenden um das Element zu verkleinern. Die Skalierung bezieht sich immer auf den Rotationspunkt: „Zentrum“ .

### Relative Bewegung

Das Element kann relative zu seiner fixen Position definiert in „*Position*“ verschoben werden. Die obere linke und untere rechte Ecke des Elements werden dabei um die in den Integer-Variablen angegebenen Werte (Pixel) in X- bzw. Y-Richtung verschoben. Im Gegensatz zur absoluten Bewegung wird eine relative Position definiert, also der Abstand zur ursprünglichen Position. Das Element kann dadurch in der Form verändert werden.

Siehe „*Absolute Bewegung*“ für das Verschieben des Elements als Ganzes.

Eigenschaft	Beschreibung Wert
<p>„<i>Bewegung links oben</i>“</p> <ul style="list-style-type: none"> <li>▪ „<i>X</i>“</li> <li>▪ „<i>Y</i>“</li> </ul>	<ul style="list-style-type: none"> <li>▪ „<i>X</i>“: Integer-Variable, deren Wert die Anzahl der Pixel angibt, um die die linke obere Ecke in X-Richtung verschoben wird.</li> <li>▪ „<i>Y</i>“: Integer-Variable, deren Wert die Anzahl der Pixel angibt, um die die linke obere Ecke in Y-Richtung verschoben wird.</li> </ul>
<p>„<i>Bewegung rechts unten</i>“</p> <ul style="list-style-type: none"> <li>▪ „<i>X</i>“</li> <li>▪ „<i>Y</i>“</li> </ul>	<ul style="list-style-type: none"> <li>▪ „<i>X</i>“: „<i>X</i>“: Integer-Variable, deren Wert die Anzahl der Pixel angibt, um die die rechte untere Ecke in X-Richtung verschoben wird.</li> <li>▪ „<i>Y</i>“: „<i>X</i>“: „<i>X</i>“: Integer-Variable, deren Wert die Anzahl der Pixel angibt, um die die rechte untere Ecke in Y-Richtung verschoben wird.</li> </ul>

## Textvariablen

Dies sind dynamische Definitionen. Siehe auch „*Dynamische Texte*“ bei Verwendung von Textlisten. Für statische Definitionen siehe „*Texte*“

Eigenschaft	Beschreibung Wert
„ <i>Textvariable</i> “	Die zugewiesene Variable definiert den Elementtext ("Textausgabevariable"). Unter <a href="#">„Text und Sprache in Visualisierungen“</a> ist zu finden, wie ein Text zu formatieren ist.
„ <i>Tooltipvariable</i> “	Die zugewiesene STRING-Variable enthält den Tooltip-Text für das Element.

## Dynamische Texte

Diese Parameter dienen der Definition von dynamischen Texten, die aus Textlisten stammen (dies erlaubt Sprachumschaltung). Siehe [Textliste](#) für nähere Informationen zur Verwendung von Textlisten.

Eigenschaft	Beschreibung Wert
„ <i>Textliste</i> “	Name der Textliste (String) wie im POU- oder Geräte-Fenster verwendet; z.B. <code>errorlist</code>
„ <i>Textindex</i> “	Index (ID) des Textes wie in der Textliste definiert (String); Direkte Angabe der ID, z.B. 'M1' oder String-Variable, z.B. <code>error_22</code> .



„*Tooltipindex*“ Index (ID) des Tooltip-Textes wie in der Textliste definiert (String); Direkte Angabe der ID, z.B. 'T1' oder String-Variable, z.B. `error_Tooltip_22`.

## Schriftartvariablen

Diese werden verwendet bei dynamischen Schriftdefinitionen des Elementtextes über Projektvariablen. Statischen Definitionen werden unter [Texteigenschaften](#) → [Schriftart](#) konfiguriert.

Eigenschaft	Beschreibung Wert
„ <i>Schriftname</i> “	Hier ist eine Variable Typ STRING anzugeben, die den Schriftnamen enthält, der für die Beschriftung des Elements verwendet werden soll. (Namensangabe wie im Standard-Schrifttypen-Dialog). Beispiel: <code>plc_prg.font_var font_var := Arial;</code>
„ <i>Größe</i> “	Eingabe einer Variablen Typ INT, die die Größe des Elementtextes in Pixel enthält, so wie im Standarddialog Font. Beispiel: <code>prog1.font_height font_height:=16;</code>
„ <i>Flags</i> “	Eingabe einer Variable Typ DWORD, die über die unten genannten Flag-Werte die Font-Darstellung festlegt. Eine kombinierte Definition kann erreicht werden, indem die jeweiligen Flag-Werte addiert werden und die Summe angegeben wird. <ul style="list-style-type: none"><li>▪ 1: kursiv</li><li>▪ 2: fett</li><li>▪ 4: unterstrichen</li><li>▪ 8: gelöscht</li></ul> <p><code>prog2.font_type: ist font_type:=6</code>, wird der Text fett und unterstrichen dargestellt.</p>
„ <i>Zeichensatz</i> “	Der Zeichensatz, der für den Font verwendet werden soll, kann über die Standard-Zeichensatz-Nummer definiert werden. Mit einer DWORD-Variablen kann diese Nummer angegeben werden (siehe auch die „ <i>Skript</i> “-Definition im Standard-Font-Dialog)
„ <i>Farbe</i> “	Eingabe einer DWORD-Variable, in der die Farbe des Elementtextes festgelegt wird.

## Farbvariablen

Eine Farbe wird mittels Hexzahl definiert, die sich aus Rot/Grün/Blau (RGB) Anteilen zusammensetzt. Für jede dieser drei Farben sind 256 (0-255) Werte verfügbar. Eine Projektvariable Typ DWORD muss angegeben werden, deren Wert die Farbe definiert (zum Beispiel `prog1.dwFillColor`).

Die Variable enthält den Farbwert in hexadezimalen Format. Um das gesamte DWORD auszufüllen, müssen die ersten beiden Ziffern nach "16#" mit 0 gesetzt werden: `16#00RRGGBB`

### Farbwerte

```
dwFillColor := 16#008FE03F;
```

Hexzahl:

- 8F: rot
- E0: grün
- 3F: blau

Eigenschaft	Beschreibung Wert
„Farbumschlag“	Hier ist eine boolsche Variabel angeben, die das Umschalten der Elementfarbe zwischen „Farbe“ (Variable = FALSE) und „Alarmfarbe“ (Variable = TRUE) steuert.
„Farbe“	Eingabe einer DWORD-Variable, die die Farbe des Elements bestimmt. Sie überschreibt den Wert, der aktuell in <i>Farben</i> → <i>Farbe</i> definiert ist, wenn die Variable in „Farbumschlag“ FALSE ist.
„Alarmfarbe“	Eingabe einer DWORD-Variable, die die Alarmfarbe des Elements bestimmt. Sie überschreibt den Wert, der aktuell in <i>Farben</i> → <i>Alarmfarbe</i> definiert ist, wenn die Variable in „Farbumschlag“ TRUE ist.

### Aussehensvariable

Anzuwenden bei einer dynamischen Definition des Aussehens der Kontur des Elements. Statische Definitionen werden in „Aussehen“ festgelegt.

Eigenschaft	Beschreibung Wert
„Linienstärke“	Eingabe einer Integervariablen, die die Linienstärke des Elements in Pixel definiert. Dies überschreibt den fixen Wert, der in <i>Aussehen</i> → <i>Linienstärke</i> festgelegt wurde.
„Linienart“	Eingabe einer DWORD-Variablen, die die Kontur der Außenlinie festlegt.  Folgende Werte entsprechen folgenden Linienarten: <ul style="list-style-type: none"><li>▪ 0: Durchgezogen</li><li>▪ 1: Strich</li><li>▪ 2: Punkt</li><li>▪ 3: Strich Punkt</li><li>▪ 4: Strich Punkt Punkt</li><li>▪ 8: Unsichtbar. Das heißt, das Element wird ohne Außenlinie dargestellt.</li></ul>

### Zustandsvariablen

Eigenschaft	Beschreibung Wert
„Unsichtbarkeit“	Eingabe einer boolschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.

„Eingaben deaktivieren“ Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann hat keine der Elementeingaben eine Auswirkung.

## Eingabekonfiguration

Hier kann definiert werden, welche Folgeaktionen ausgeführt werden, wenn der Benutzer im Online-Modus eine Eingabe im Element vornimmt.

Eingabeereignisse, die ein Element haben kann, sind:

- „OnDialogClosed“
- „OnMouseClicked“
- „OnMouseDown“
- „OnMouseEnter“
- „OnMouseLeave“
- „OnMouseMove“
- „OnMouseUp“

Mit Klick auf „Konfigurieren...“ wird der Dialog [„Eingabekonfiguration“](#) geöffnet, der erlaubt Folgeaktionen zuzuweisen. Man klickt in das Feld neben der zu konfigurierenden Eingabe. Für jede selektierte Folgeaktion wird ein Eintrag in der Eigenschaftentabelle erzeugt. Jede Eingabe kann eine beliebige zugeordnete Anzahl an Folgeaktionen haben.

Eigenschaft	Beschreibung Wert
„OnDialogClosed“	Dieses Ereignis tritt ein, wenn innerhalb der Visualisierung eines der für Benutzereingaben vorher geöffneten Dialogfenster geschlossen wird.
„OnMouseClicked“	Dieses Mausereignis tritt ein, wenn der Cursor auf ein Element zeigt und ein vollständiger Mausklick (drücken und freigeben der Maustaste) ausgeführt wird.
„OnMouseDown“	Dieses Mausereignis tritt ein, wenn die Maustaste gedrückt wird während der Cursor auf ein Element zeigt. (Drücken der Maustaste)
„OnMouseEnter“	Dieses Mausereignis tritt ein, wenn der Cursor auf das Element gezogen wird.
„OnMouseLeave“	Dieses Mausereignis tritt ein, wenn der Cursor das Element verlässt.
„OnMouseMove“	Dieses Mausereignis tritt ein, wenn der Cursor innerhalb des Elements bewegt wird.
„OnMouseUp“	Diese Mausereignis liegt vor, wenn die Maustaste losgelassen wird. Das Drücken der Maustaste geschah vorher, außerhalb des Elements.

## Eingabekonfiguration: Hotkey

Mit „Hotkey“ kann eine Taste, auch mit Zusatzaste, definiert und mit einer bestimmten Nachfolgeaktion (MouseDown, MouseUp) verknüpft werden, die bei einem Ereignis der Taste (KeyDown, KeyUp) ausgeführt werden soll. Standardmäßig wird bei KeyDown (Taste drücken) die MouseDown-Aktion ausgeführt und bei KeyUp (Taste

loslassen) die MouseUp-Aktion. Das kann nützlich sein, wenn eine Visualisierung sowohl über Maus- als auch Tastatureingaben bedient werden soll, weil die Eingabeaktionen dann nur einmalig konfiguriert werden müssen. Diese Tastenkonfiguration für ein Element wird auch im [„Tastaturkonfiguration“](#) der Visualisierung verwaltet. Änderungen werden immer zwischen diesem und dem Elementeigenschaften-Editor synchronisiert.

Eigenschaft	Beschreibung Wert
„Taste“	Zuweisen einer Taste. Eine Auswahlliste stellt alle aktuell unterstützten Tasten bereit wie zum Beispiel [ M ]. <a href="#">„Tastaturkonfiguration“</a>
„Eingabe(n)“	Definieren des Ereignisses, das ausgeführt werden soll, wenn die Taste beziehungsweise Taste und Modifizierer verwendet werden.  Mögliche Werte, die in einer Auswahlliste bereitstehen: <ul style="list-style-type: none"> <li>Keine Aktion</li> <li>MouseDown-Aktion: OnMouseDown-Aktion, wenn die Taste gedrückt wird.</li> <li>MouseUp-Aktion: OnMouseUp-Aktion, wenn die Taste freigelassen wird.</li> <li>MouseDown/MouseUp-Aktion: OnMouseUp- und OnMouseDown-Aktion, wenn die Taste gedrückt oder losgelassen wird.</li> </ul>
„Umschalten“	Wenn diese Option aktiviert wird, muss die Taste mit [ Umschalt ] kombiniert verwendet werden
„Steuerung“	Wenn diese Option aktiviert wird, muss die Taste mit [ Strg ] kombiniert verwendet werden
„Alt“	Wenn diese Option aktiviert wird, muss die Taste mit [ Alt ] kombiniert verwendet werden

Eigenschaft	Beschreibung Wert
„Zugriffsrechte“	Mit Klick öffnet <a href="#">„Zugriffsrechte“</a> .  Statusmeldungen: <ul style="list-style-type: none"> <li>Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.</li> <li>Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.</li> </ul>

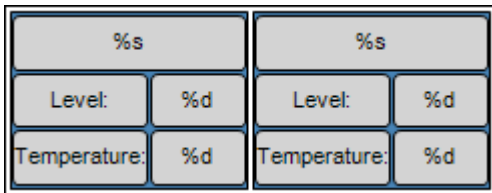
## Frame

Ein Frame-Element wird verwendet

- als [↘ „Referenz zu einer anderen Visualisierung“](#),
- zum [↘ „Anbieten eine Schnittstelle für Platzhalter“](#),
- zum [↘ „Umschalten zwischen verschiedenen Visualisierungen“](#).

Inhalt dieser Seite ist weiterhin:

[↘ „Elementeigenschaften“](#)



%s		%s	
Level:	%d	Level:	%d
Temperature:	%d	Temperature:	%d

Abb. 58: Frame

### Referenz zu einer anderen Visualisierung

Grundsätzlich kann eine Visualisierung in eine andere eingefügt und damit referenziert werden. Für diesen Zweck ist ein „*Frame*“ zu verwenden, der eine oder mehrere Referenzen zu Visualisierungen einschließen kann. Welche Visualisierung aktuell zum Einfügen zur Verfügung steht, hängt von deren Position im POU- und Gerätebaum relativ zur gerade geöffneten Visualisierung ab. Siehe dazu [↘ „Frame-Auswahl“](#).

### Anbieten eine Schnittstelle für Platzhalter

Eine Visualisierung kann eine **Schnittstelle**, zu öffnen über den Befehl [↘ „Grundlegendes zum Schnittstelleneditor“](#), anbieten, in der man die Eingabevariablen genauso wie bei einem Funktionsbaustein definieren kann. Diese Eingabeparameter funktionieren als **Platzhalter**. In einer Instanz (Referenz) der Visualisierung muss ein Platzhalter dann durch einen Wert oder Ausdruck für die spezielle Verwendung im lokalen Objekt ersetzt werden. Das Ersetzen muss in „*Eigenschaften*“ des Frames, der die Visualisierungsinstanzen einbindet, vorgenommen werden: [Referenzierte Visualisierungen → Referenzierte Visualisierungen](#). Beachten Sie dabei, dass den Eingabevariablen einer Visualisierungsinstanz, die einer bestimmten Applikation zugeordnet sind, gültige Variablen dieser Applikation zugewiesen werden müssen.

### Umschalten zwischen verschiedenen Visualisierungen


Ein Frame (so wie man Frames zum Beispiel von Internetseiten her kennt) definiert einen Teilbereich der aktuellen Visualisierung, der weitere Visualisierungen enthalten kann, zwischen denen man im Online-Modus umschaltet. Bei diesen Visualisierungen werden deren Eingaben so konfigurieren, dass sie einen Wechsel der Anzeige dieser Referenzen im Frame bewirken. Der Dialog [↘ „Eingabekonfiguration“](#) bietet dafür die Option [„Visualisierungswechsel“](#) an. Dabei ist zu berücksichtigen, dass Mausaktionen mit Hilfe der Eigenschaft

„Eingabekonfiguration“ definiert werden.

Um die Auswahl der Visualisierungen in einem Frame zu definieren ist der Dialog „*Frame-Konfiguration*“ durch Selektieren des Frames und ausführen des Befehls [↘ „Frame-Auswahl“](#) zu öffnen. Hier sind alle verfügbaren Visualisierungen bereitgestellt und man kann die gewünschte auswählen und damit in den Frame einfügen. Empfehlenswert ist es, nur Visualisierungen aus dem globalen POU-Pool zu verwenden, um zu vermeiden, dass später Probleme entstehen weil vielleicht ein Gerät umbenannt wird. Ein "gefüllter" Frame zeigt dann auch im Offline-Modus die enthaltenen Visualisierungen an.

### Umschalten zwischen Visualisierungen

Eine Visualisierung enthält drei Schaltflächen "1", "2", "3" und einen Frame, dem die Visualisierungen "Visu1", "Visu2" und "Visu3" zugewiesen sind. Die Schaltflächen werden so konfiguriert (Eingabe, OnMouse-Aktionen, Visualisierungswechsel), dass eine Benutzereingabe auf eine Schaltfläche die jeweilige Visualisierung im Frame aufruft. Somit können - im Unterschied zu CODESYS V2.3 - innerhalb eines Visualisierungselements verschiedene andere im Wechsel angezeigt werden, also eine Schalttafel visualisiert werden.

	Dieses Konzept ersetzt das Visualisierungs- und Platzhalterkonzept in CODESYS V2.3.
--	---

#### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungseditor werden folgende Benutzereingaben zur Verfügung gestellt:  
Element spezifisches Kommando:

- [↘ „Frame-Auswahl“](#)

#### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Frame
„ <i>Abschneiden</i> “	Wird diese Option zusammen mit „ <i>Skalierungsart</i> “: Unskaliert" aktiviert, dann wird nur der Teil der Visualisierung, der in den Rahmen passt, dargestellt.
„ <i>Rahmen zeichnen</i> “	Falls aktiviert, wird die referenzierte Visualisierung mit einem Rahmen angezeigt.

### „Skalierungsart“

Hier ist zu spezifizieren, wie der Frame auf Größenänderungen der Visualisierung reagieren soll:

- **Isotropisch:** Der Frame behält seine Proportionen. Damit kann das Verhältnis von Höhe und Breite beibehalten werden, sogar wenn Höhe und Breite der Visualisierung unabhängig modifiziert wurden.
- **Anisotropisch:** Der Frame richtet sich nach der Größe der Visualisierung, so dass Höhe und Breite der referenzierten Visualisierung unabhängig modifiziert werden können.
- **Unskaliert:** Die original Größe des Frames wird beibehalten ungeachtet der Visualisierungsgröße. Falls die Option „**Abschneiden**“ ebenfalls aktiviert ist, wird nur der passende Teil angezeigt.

### „Deaktivieren des Hintergrundzeichnens“

Um die Performance der Visualisierung zu optimieren, werden die nicht-animierten Elemente des Frame-Elements als Hintergrund Bitmap gezeichnet. Diese könnte dazu führen, dass die Elemente nicht in der erwarteten Reihenfolge dargestellt werden. Um dieses Verhalten zu vermeiden, kann die Funktion deaktiviert werden.

### Referenzierte Visualisierungen

Eigenschaft	Beschreibung Wert
„Referenzierte Visualisierungen“	Der Befehl „ <b>Konfiguriere</b> “ öffnet den Dialog für die <a href="#">„Frame-Auswahl“</a> . Hier werden die referenzierten Visualisierungen des Elements aufgelistet.

### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.



Eigenschaft	Beschreibung Wert
„X“	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„Y“	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„Breite“	Horizontale Breite des Elements.
„Höhe“	Vertikale Höhe des Elements.

### Zentrum

Mit Bearbeiten des Wertes wird gleichzeitig das zugehörige Element  im Editor verschoben.

Eigenschaft	Beschreibung Wert
„X“	X-Position des Rotationspunkts des Elements
„Y“	Y-Position des Rotationspunkts des Elements

## Farben

Eine Farbe wird mittels Hexzahl definiert, die sich aus Rot/Grün/Blau (RGB) Anteilen zusammensetzt. Für jede dieser drei Farben sind 256 (0-255) Werte verfügbar. Hier sind die Einstellungen **statisch**. Die Farbe kann aus einer Auswahlliste oder dem Farbauswahldialog, der über Button  öffnet, gewählt werden. Beispiel:  0; 128; 64

Eigenschaft	Beschreibung Wert
„Farbe“	Hier ist eine Farbe, mit der das Element im Normalzustand gefüllt wird, auszuwählen. Dies ist der Fall, wenn die Variable in <i>Farbvariablen</i> → <i>Farbumschlag</i> FALSE ist.  Mit Editieren des Werts ändert das Element gleichzeitig seine Farbe.
„Alarmfarbe“	Auszuwählen ist eine Farbe, mit der das Element gefüllt wird, wenn das Element in Alarmzustand ist. Dies ist der Fall, wenn die Variable in <i>Farbvariablen</i> → <i>Farbumschlag</i> TRUE ist.

## Aussehen

Eigenschaft	Beschreibung Wert
„Linienstärke“	Stärke der Rahmenlinie in Pixel. 0 kodiert das Gleiche wie 1 und setzt die Linienstärke auf 1 Pixel. Soll kein Rahmen verwendet werden, muss „Linienart“ auf <code>unsichtbar</code> setzen werden.
„Linienart“	Auszuwählen ist eine der Linienarten angewendet bei der Umrisslinie: <ul style="list-style-type: none"> <li>▪ Durchgezogen</li> <li>▪ Striche</li> <li>▪ Punkte</li> <li>▪ Strich Punkt</li> <li>▪ Strich Punkt Punkt</li> <li>▪ Unsichtbar: Dann ist die Umrisslinie unsichtbar.</li> </ul>

## Texte

Diese Eigenschaften werden für eine statische Definition der Elementbeschriftung verwendet.

Property	Value description
„Text“	Der angegebene Text beschriftet das Element. Ein textformatierender String ist möglich.
„Tooltip“	Der angegebenen Text erscheint als Tooltip des Elements, wenn die Visualisierung im Online-Modus ist.

## Texteigenschaften




Diese Eigenschaften werden für eine statische Definition der Schriftart verwendet.

Eigenschaft	Beschreibung Wert
„Horizontale Ausrichtung“	<p>Definieren der horizontalen Ausrichtung des Textes durch Auswahl aus:</p> <ul style="list-style-type: none"> <li>▪ Links</li> <li>▪ Zentriert</li> <li>▪ Rechts</li> </ul>
„Vertikale Ausrichtung“	<p>Definieren der vertikalen Ausrichtung des Textes durch Auswahl aus:</p> <ul style="list-style-type: none"> <li>▪ Oben</li> <li>▪ Zentriert</li> <li>▪ Unten</li> </ul>
„Schriftart“	<p>Auswahl aus vordefinierten Schriftarten:</p> <ul style="list-style-type: none"> <li>▪ Standard</li> <li>▪ Überschrift</li> <li>▪ Große</li> <li>▪ Titel</li> <li>▪ Anmerkung</li> </ul> <p>Mit Klick auf <input type="button" value="..."/> öffnet sich der Dialog für benutzerdefinierte Eigenschaften der Schriftart.</p>
„Farbe Schriftart“	<p>Auszuwählen ist eine Schriftartfarbe für das Element. Entweder aus der Auswahlliste oder durch Setzen im sich öffnenden Dialog, wenn <input type="button" value="..."/> geklickt wird.</p>


### Absolute Bewegung

Das Element kann bewegt werden, indem die X- und Y-Position (Pixel) der linken oberen Ecke des Elements durch eine Integer-Variable verändert werden. Hier werden absolute Koordinatenwerte verwendet.

Eigenschaft	Beschreibung Wert
„Bewegung“	<ul style="list-style-type: none"> <li>▪ „X“: Hier ist eine Integer-Variable einzugeben, deren Wert die aktuelle X-Position der linken oberen Ecke des Elements in Pixel definiert; kann verwendet werden, um das Element in X-Richtung zu verschieben (ein positiver Wert verschiebt das Element von links nach rechts).</li> <li>▪ „Y“: Eingabe einer Integer-Variablen, die die aktuelle Y-Position der linken oberen Ecke des Elements in Pixel definiert; kann verwendet werden, um das Element in Y-Richtung zu verschieben (ein positiver Wert verschiebt das Element von oben nach unten).</li> </ul>
„Rotation“	<p>Eingabe einer Integer-Variablen, deren Wert den Winkel (Winkelgrade) definiert, um den das Element um den Rotationspunkt „Zentrum“  rotiert.</p>

positiver Wert = mathematisch positiv = im Uhrzeigersinn

Zu beachten ist, dass das Element, im Gegensatz zum Verhalten bei „*Innere Rotation*“, selber nicht rotiert.

„*Skalierung*“ Eingabe einer Integer-Variablen, deren Wert den aktuellen Skalenwert (Prozent) definiert. Die Elementgröße wird linear entsprechend dieses Wertes verändert. Implizit wird der Wert durch 1000 dividiert, so dass es nicht nötig ist, REAL-Variablen zu verwenden um das Element zu verkleinern. Die Skalierung bezieht sich immer auf den Rotationspunkt: „*Zentrum*“ .

## Relative Bewegung

Das Element kann relative zu seiner fixen Position definiert in „*Position*“ verschoben werden. Die obere-linke und untere-rechte Ecke des Elements werden dabei um die in den Integer-Variablen angegebenen Werte (Pixel) in X- bzw. Y-Richtung verschoben. Im Gegensatz zur absoluten Bewegung wird eine relative Position definiert, also der Abstand zur ursprünglichen Position. Das Element kann dadurch in der Form verändert werden.

Siehe „*Absolute Bewegung*“ für das Verschieben des Elements als Ganzes.

Eigenschaft	Beschreibung Wert
„ <i>Bewegung links oben</i> “	<ul style="list-style-type: none"><li>▪ „<i>X</i>“: Integer-Variable, deren Wert die Anzahl der Pixel angibt, um die die linke obere Ecke in X-Richtung verschoben wird.</li><li>▪ „<i>Y</i>“: Integer-Variable, deren Wert die Anzahl der Pixel angibt, um die die linke obere Ecke in Y-Richtung verschoben wird.</li></ul>
▪ „ <i>X</i> “	
▪ „ <i>Y</i> “	
„ <i>Bewegung rechts unten</i> “	<ul style="list-style-type: none"><li>▪ „<i>X</i>“: „<i>X</i>“: Integer-Variable, deren Wert die Anzahl der Pixel angibt, um die die rechte untere Ecke in X-Richtung verschoben wird.</li><li>▪ „<i>Y</i>“: „<i>X</i>“: „<i>X</i>“: Integer-Variable, deren Wert die Anzahl der Pixel angibt, um die die rechte untere Ecke in Y-Richtung verschoben wird.</li></ul>
▪ „ <i>X</i> “	
▪ „ <i>Y</i> “	

## Textvariablen

Dies sind dynamische Definitionen. Siehe auch „*Dynamische Texte*“ bei Verwendung von Textlisten. Für statische Definitionen siehe „*Texte*“

Eigenschaft	Beschreibung Wert
„ <i>Textvariable</i> “	Die zugewiesene Variable definiert den Elementtext ("Textausgabevariable"). Unter <a href="#">„Text und Sprache in Visualisierungen“</a> ist zu finden, wie ein Text zu formatieren ist.
„ <i>Tooltipvariable</i> “	Die zugewiesene STRING-Variable enthält den Tooltip-Text für das Element.

## Dynamische Texte

Diese Parameter dienen der Definition von dynamischen Texten, die aus Textlisten stammen (dies erlaubt Sprachumschaltung). Siehe [Textliste](#) für nähere Informationen zur Verwendung von Textlisten.

Eigenschaft	Beschreibung Wert
„ <i>Textliste</i> “	Name der Textliste (String) wie im POU's- oder Geräte-Fenster verwendet; z.B. <code>errorlist</code>
„ <i>Textindex</i> “	Index (ID) des Textes wie in der Textliste definiert (String); Direkte Angabe der ID, z.B. 'M1' oder String-Variable, z.B. <code>error_22</code> .
„ <i>Tooltipindex</i> “	Index (ID) des Tooltip-Textes wie in der Textliste definiert (String); Direkte Angabe der ID, z.B. 'T1' oder String-Variable, z.B. <code>error_Tooltip_22</code> .

### Schriftartvariablen

Diese werden verwendet bei dynamischen Schriftdefinitionen des Elementtextes über Projektvariablen. Statischen Definitionen werden unter [Texteigenschaften](#) → [Schriftart](#) konfiguriert.

Eigenschaft	Beschreibung Wert
„ <i>Schriftname</i> “	Hier ist eine Variable Typ STRING anzugeben, die den Schriftnamen enthält, der für die Beschriftung des Elements verwendet werden soll. (Namensangabe wie im Standard-Schrifttypen-Dialog). Beispiel: <code>plc_prg.font_var font_var := Arial;</code>
„ <i>Größe</i> “	Eingabe einer Variablen Typ INT, die die Größe des Elementtextes in Pixel enthält, so wie im Standarddialog Font. Beispiel: <code>prog1.font_height font_height:=16;</code>
„ <i>Flags</i> “	Eingabe einer Variable Typ DWORD, die über die unten genannten Flag-Werte die Font-Darstellung festlegt. Eine kombinierte Definition kann erreicht werden, indem die jeweiligen Flag-Werte addiert werden und die Summe angegeben wird. <ul style="list-style-type: none"> <li>▪ 1: kursiv</li> <li>▪ 2: fett</li> <li>▪ 4: unterstrichen</li> <li>▪ 8: gelöscht</li> </ul> <p><code>prog2.font_type: ist font_type:=6, wird der Text fett und unterstrichen dargestellt.</code></p>
„ <i>Zeichensatz</i> “	Der Zeichensatz, der für den Font verwendet werden soll, kann über die Standard-Zeichensatz-Nummer definiert werden. Mit einer DWORD-Variablen kann diese Nummer angegeben werden (siehe auch die <a href="#">„Skript“</a> -Definition im Standard-Font-Dialog)
„ <i>Farbe</i> “	Eingabe einer DWORD-Variable, in der die Farbe des Elementtextes festgelegt wird.

### Farbvariablen

Eine Farbe wird mittels Hexzahl definiert, die sich aus Rot/Grün/Blau (RGB) Anteilen zusammensetzt. Für jede dieser drei Farben sind 256 (0-255) Werte verfügbar. Eine Projektvariable Typ DWORD muss angegeben werden,

deren Wert die Farbe definiert (zum Beispiel `prog1.dwFillColor`).

Die Variable enthält den Farbwert in hexadezimalen Format. Um das gesamte DWORD auszufüllen, müssen die ersten beiden Ziffern nach "16#" mit 0 gesetzt werden: 16#00RRGGBB

### Farbwerte

```
dwFillColor := 16#008FE03F;
```

Hexzahl:

- 8F: rot
- E0: grün
- 3F: blau

Eigenschaft	Beschreibung Wert
„Farbumschlag“	Hier ist eine boolesche Variable angeben, die das Umschalten der Elementfarbe zwischen „Farbe“ (Variable = FALSE) und „Alarmfarbe“ (Variable = TRUE) steuert.
„Farbe“	Eingabe einer DWORD-Variable, die die Farbe des Elements bestimmt. Sie überschreibt den Wert, der aktuell in <i>Farben</i> → <i>Farbe</i> definiert ist, wenn die Variable in „Farbumschlag“ FALSE ist.
„Alarmfarbe“	Eingabe einer DWORD-Variable, die die Alarmfarbe des Elements bestimmt. Sie überschreibt den Wert, der aktuell in <i>Farben</i> → <i>Alarmfarbe</i> definiert ist, wenn die Variable in „Farbumschlag“ TRUE ist.

### Aussehensvariable

Anzuwenden bei einer dynamischen Definition des Aussehens der Kontur des Elements. Statische Definitionen werden in „Aussehen“ festgelegt.

Eigenschaft	Beschreibung Wert
„Linienstärke“	Eingabe einer Integervariablen, die die Linienstärke des Elements in Pixel definiert. Dies überschreibt den fixen Wert, der in <i>Aussehen</i> → <i>Linienstärke</i> festgelegt wurde.
„Linienart“	Eingabe einer DWORD-Variablen, die die Kontur der Außenlinie festlegt.

Folgende Werte entsprechen folgenden Linienarten:

- 0: Durchgezogen
- 1: Strich
- 2: Punkt
- 3: Strich Punkt

- 4: Strich Punkt Punkt
- 8: Unsichtbar. Das heißt, das Element wird ohne Außenlinie dargestellt.

## Umschaltvariable

Diese Eigenschaft ermöglicht es, innerhalb eines Rahmens Visualisierungen umzuschalten.

Eigenschaft	Beschreibung Wert
„ <i>Variable</i> “	Hier ist eine Integervariable anzugeben, deren Wert die ID der anzuzeigenden Visualisierung enthält. Die ID einer Visualisierung wird durch deren Reihenfolge in der Liste der zugewiesenen Visualisierungen im Dialog „ <i>Konfiguration der Framevisualisierungen</i> “ bestimmt. Zum Beispiel ergibt der erste Eintrag in dieser Liste die ID 0, der zweite Eintrag die ID 1, usw.

## Zustandsvariablen

Eigenschaft	Beschreibung Wert
„ <i>Unsichtbarkeit</i> “	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.
„ <i>Eingaben deaktivieren</i> “	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann hat keine der Elementeingaben eine Auswirkung.

## Eingabekonfiguration

Hier kann definiert werden, welche Folgeaktionen ausgeführt werden, wenn der Benutzer im Online-Modus eine Eingabe im Element vornimmt.

Eingabeereignisse, die ein Element haben kann, sind:

- „*OnDialogClosed*“
- „*OnMouseClicked*“
- „*OnMouseDown*“
- „*OnMouseEnter*“
- „*OnMouseLeave*“
- „*OnMouseMove*“
- „*OnMouseUp*“

Mit Klick auf „*Konfigurieren...*“ wird der Dialog [„Eingabekonfiguration“](#) geöffnet, der erlaubt Folgeaktionen zuzuweisen. Man klickt in das Feld neben der zu konfigurierenden Eingabe. Für jede selektierte Folgeaktion wird ein Eintrag in der Eigenschaftentabelle erzeugt. Jede Eingabe kann eine beliebige zugeordnete Anzahl an Folgeaktionen haben.

Eigenschaft	Beschreibung Wert
-------------	-------------------

- „*OnDialogClosed*“ Dieses Ereignis tritt ein, wenn innerhalb der Visualisierung eines der für Benutzereingaben vorher geöffneten Dialogfenster geschlossen wird.
- „*OnMouseClicked*“ Dieses Mausereignis tritt ein, wenn der Cursor auf ein Element zeigt und ein vollständiger Mausklick (drücken und freigeben der Maustaste) ausgeführt wird.
- „*OnMouseDown*“ Dieses Mausereignis tritt ein, wenn die Maustaste gedrückt wird während der Cursor auf ein Element zeigt. (Drücken der Maustaste)
- „*OnMouseEnter*“ Dieses Mausereignis tritt ein, wenn der Cursor auf das Element gezogen wird.
- „*OnMouseLeave*“ Dieses Mausereignis tritt ein, wenn der Cursor das Element verlässt.
- „*OnMouseMove*“ Dieses Mausereignis tritt ein, wenn der Cursor innerhalb des Elements bewegt wird.
- „*OnMouseUp*“ Diese Mausereignis liegt vor, wenn die Maustaste losgelassen wird. Das Drücken der Maustaste geschah vorher, außerhalb des Elements.


### Eingabekonfiguration: Tasten

Mit „*Tasten*“ kann konfiguriert werden, dass beim Ereignis "Tasten" der Wert einer boolschen Projektvariable abhängig vom Mausverhalten gesetzt wird.

Eigenschaft	Beschreibung Wert
„ <i>Variable</i> “	Hier kann eine boolsche Variable zugewiesen werden, deren Wert TRUE wird, wenn die Maustaste gedrückt wird, während der Cursor auf das Element zeigt und die wieder FALSE wird, wenn die Maustaste wieder losgelassen wird.
„ <i>FALSE tasten</i> “	Wenn diese Option aktiviert ist, wird das oben beschriebene Tastverhalten für die eingetragene Variable umgekehrt. Das heißt, wenn die Maustaste gedrückt wird, wird der Variablenwert auf FALSE gesetzt, wenn die Taste wieder losgelassen wird, wird der Variablenwert auf TRUE gesetzt.
„ <i>Beim Betreten tasten, falls Maus gefangen</i> “	Ist diese Option aktiviert, verhält sich der Variablenwert wie unter „ <i>Variable</i> “ beschrieben, solange sich die Maus innerhalb des Elementbereichs befindet. Zusätzlich wird auf ein Ziehen der Maus reagiert. Wird bei gedrückter Maustaste der Elementbereich verlassen, wird die Variable FALSE und wird wieder TRUE, wenn in den Elementbereich zurückgekehrt wird. Es wird also berücksichtigt, dass, solange die Maustaste gedrückt ist, die Maus "gefangen" ist, auch wenn der Elementbereich verlassen wird.

### Eingabekonfiguration: Umschalten

Mit „*Umschalten*“ kann konfiguriert werden, dass bei diesem Ereignis der Wert einer boolschen Projektvariable abhängig vom Mausverhalten gesetzt wird.

Eigenschaft	Beschreibung Wert
„ <i>Variable</i> “	Hier ist eine boolsche Variable zuzuweisen. Die Eingabehilfe kann über Schaltfläche 

geöffnet werden.

Der Wert der Variablen wird bei jedem Mausklick auf das Element zwischen TRUE und FALSE umgeschaltet. Wird der Cursor bei gedrückter Maustaste aus dem Element herausbewegt, dann erfolgt kein Toggeln. Damit kann die begonnene Umschalteingabe abgebrochen werden.

„*Umschalten beim Loslassen falls Maus gefangen*“ Ist diese Option aktiviert, verhält sich der Variablenwert wie unter „*Variable*“ beschrieben, solange sich die Maus innerhalb des Elementbereichs befindet. Zusätzlich wird auf ein Ziehen der Maus reagiert. Wird bei gedrückter Maustaste der Elementbereich verlassen und dann die Maustaste losgelassen, wird die Variable umgeschaltet. Es wird also berücksichtigt, dass, solange die Maustaste gedrückt ist, die Maus "gefangen" ist, auch wenn der Elementbereich verlassen wird.

### Eingabekonfiguration: Hotkey

Mit „*Hotkey*“ kann eine Taste, auch mit Zusatzaste, definiert und mit einer bestimmten Nachfolgeaktion (MouseDown, MouseUp) verknüpft werden, die bei einem Ereignis der Taste (KeyDown, KeyUp) ausgeführt werden soll. Standardmäßig wird bei KeyDown (Taste drücken) die MouseDown-Aktion ausgeführt und bei KeyUp (Taste loslassen) die MouseUp-Aktion. Das kann nützlich sein, wenn eine Visualisierung sowohl über Maus- als auch Tastatureingaben bedient werden soll, weil die Eingabeaktionen dann nur einmalig konfiguriert werden müssen. Diese Tastenkonfiguration für ein Element wird auch im [„Tastaturkonfiguration“](#) der Visualisierung verwaltet. Änderungen werden immer zwischen diesem und dem Elementeigenschaften-Editor synchronisiert.

Eigenschaft	Beschreibung Wert
„ <i>Taste</i> “	Zuweisen einer Taste. Eine Auswahlliste stellt alle aktuell unterstützten Tasten bereit wie zum Beispiel [ <i>M</i> ]. <a href="#">„Tastaturkonfiguration“</a>
„ <i>Eingabe(n)</i> “	Definieren des Ereignisses, das ausgeführt werden soll, wenn die Taste beziehungsweise Taste und Modifizierer verwendet werden.  Mögliche Werte, die in einer Auswahlliste bereitstehen: <ul style="list-style-type: none"><li>Keine Aktion</li><li>MouseDown-Aktion: OnMouseDown-Aktion, wenn die Taste gedrückt wird.</li><li>MouseUp-Aktion: OnMouseUp-Aktion, wenn die Taste freigelassen wird.</li><li>MouseDown/MouseUp-Aktion: OnMouseUp- und OnMouseDown-Aktion, wenn die Taste gedrückt oder losgelassen wird.</li></ul>
„ <i>Umschalten</i> “	Wenn diese Option aktiviert wird, muss die Taste mit [ <i>Umschalt</i> ] kombiniert verwendet werden
„ <i>Steuerung</i> “	Wenn diese Option aktiviert wird, muss die Taste mit [ <i>Strg</i> ] kombiniert verwendet werden
„ <i>Alt</i> “	Wenn diese Option aktiviert wird, muss die Taste mit [ <i>Alt</i> ] kombiniert verwendet werden

Eigenschaft	Beschreibung Wert
-------------	-------------------

„Zugriffsrechte“ Mit Klick öffnet [↘ „Zugriffsrechte“](#).

Statusmeldungen:

- Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.
- Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.



### 6.3.5.3 Allgemeine Steuerelemente

#### Text

Das Element „Text“ dient dazu Beschriftungen, Überschriften und jede andere Art von Text einer Visualisierung hinzuzufügen.



Abb. 59: Text

#### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungsektor werden folgende Benutzereingaben zur Verfügung gestellt:

#### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„Elementname“	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„Elementtyp“	Text

#### Texte

Diese Eigenschaften werden für eine statische Definition der Elementbeschriftung verwendet.

Property	Value description
„Text“	Der angegebene Text beschriftet das Element. Ein textformatierender String ist möglich.

#### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„X“	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„Y“	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„Breite“	Horizontale Breite des Elements.
„Höhe“	Vertikale Höhe des Elements.

## Texteigenschaften

Diese Eigenschaften werden für eine statische Definition der Schriftart verwendet.

Eigenschaft	Beschreibung Wert
<i>„Horizontale Ausrichtung“</i>	Definieren der horizontalen Ausrichtung des Textes durch Auswahl aus: <ul style="list-style-type: none"><li>▪ Links</li><li>▪ Zentriert</li><li>▪ Rechts</li></ul>
<i>„Vertikale Ausrichtung“</i>	Definieren der vertikalen Ausrichtung des Textes durch Auswahl aus: <ul style="list-style-type: none"><li>▪ Oben</li><li>▪ Zentriert</li><li>▪ Unten</li></ul>
<i>„Schriftart“</i>	Auswahl aus vordefinierten Schriftarten: <ul style="list-style-type: none"><li>▪ Standard</li><li>▪ Überschrift</li><li>▪ Große</li><li>▪ Titel</li><li>▪ Anmerkung</li></ul> <p>Mit Klick auf <input type="text" value="..."/> öffnet sich der Dialog für benutzerdefinierte Eigenschaften der Schriftart.</p>
<i>„Farbe Schriftart“</i>	Auszuwählen ist eine Schriftartfarbe für das Element. Entweder aus der Auswahlliste oder durch Setzen im sich öffnenden Dialog, wenn <input type="text" value="..."/> geklickt wird.

## Zustandsvariablen

Property	Value description
<i>„Unsichtbarkeit“</i>	Eingabe einer boolschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.

Eigenschaft	Beschreibung Wert
<i>„Zugriffsrechte“</i>	Mit Klick öffnet <a href="#">\ „Zugriffsrechte“</a> .  Statusmeldungen: <ul style="list-style-type: none"><li>▪ Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.</li><li>▪ Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten</li></ul>

angezeigt wird.

### Combobox Integer

„Combobox Integer“ erlaubt dem Benutzer, einen Wert aus einer Drop-down-Liste auszuwählen. Der Index des selektierten Wertes wird in eine Variable geschrieben. Die Einträge können aus einer Textliste oder als Bilderliste aus der Bildersammlung herangezogen werden.

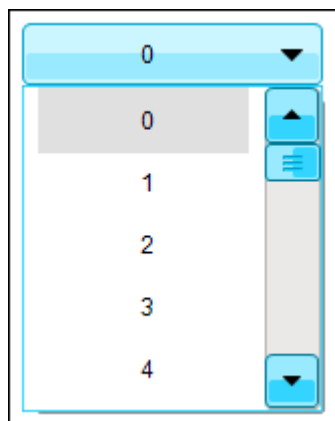


Abb. 60: Combobox Integer, Style 8

#### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungsektor werden folgende Benutzereingaben zur Verfügung gestellt:

#### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Combobox Integer

#### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„ <i>X</i> “	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach

rechts.

- „Y“ Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
- „Breite“ Horizontale Breite des Elements.
- „Höhe“ Vertikale Höhe des Elements.
- „Variable“ Anzugeben ist eine Variable Typ Aufzählung, deren Wert der Index des ausgewählten Listenelements ist.
- „Textliste“ Anzugeben ist der Name (STRING) der Textliste, die alle Einträge der Drop-down-Liste enthält und indiziert.
- „Bildersammlung“ Anzugeben ist der Name (STRING) der Bildersammlung, die alle Bilder der Drop-down-Liste enthält und indiziert.
- „Maximaler Wert“ Zu definieren ist die maximale Anzahl an Einträgen in der Drop-down-Liste der Combobox.

## Texte

Diese Eigenschaften werden für eine statische Definition der Elementbeschriftung verwendet.

Eigenschaft	Beschreibung Wert
„Tooltip“	Der angegebenen Text erscheint als Tooltip des Elements, wenn die Visualisierung im Online-Modus ist.

## Unterbereich

Eigenschaft	Beschreibung Wert
„Unterbereich verwenden“	Ist diese Option aktiviert, wird nur ein Unterbereich der zugewiesenen Listeneinträgen verwendet.
„Indexstart“	Einzugeben ist ein Integerwert, der den Eintrag indiziert, der als erstes zu verwenden ist.
„Indexende“	Einzugeben ist ein Integerwert, der den Eintrag indiziert, der als letztes zu verwenden ist.

## Zustandsvariablen

Eigenschaft	Beschreibung Wert
„Unsichtbarkeit“	Eingabe einer boolschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.
„Eingaben deaktivieren“	Eingabe einer boolschen Variable; ist der Variablenwert TRUE, dann hat keine der Elementeingaben eine Auswirkung.

Eigenschaft	Beschreibung Wert
„Zugriffsrechte“	Mit Klick öffnet <a href="#">„Zugriffsrechte“</a> .

Statusmeldungen:

- Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.
- Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.

### Combobox Array

„Combobox Array“ erlaubt dem Benutzer, eine Zeile mit Werten aus einer Drop-down-Liste auszuwählen. Der Index der selektierten Zeile wird in eine Variable geschrieben.



Abb. 61: Combobox Array, Style 8

#### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungseitor werden folgende Benutzereingaben zur Verfügung gestellt:

#### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	ComboBoxArray

#### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„X“	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„Y“	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„Breite“	Horizontale Breite des Elements.
„Höhe“	Vertikale Höhe des Elements.
„Variable“	Variable, welche den Index der ausgewählten Zeile enthält
„Daten Array“	Variable vom Typ Array mit vollständigem Pfad, welche dargestellt werden soll. Die Struktur der Variablen bestimmt die Anzahl der Spalten und Zeilen der Tabelle.

## Spalten

Das Element Combobox Array visualisiert eine Array- oder Strukturvariable in Tabellenansicht. Der Index von Arrayelementen bzw. von Komponenten einer Struktur wird in einer Spalte bzw. einer Zeile angezeigt. Zweidimensionale Arrays oder Arrays einer Struktur werden in mehreren Spalten angezeigt. Die visualisierte Variable wird in der Eigenschaft „Daten-Array“ festgelegt. Wenn dort eine Variable zugewiesen ist, kann die Darstellung der Tabellenspalten, in welchen die jeweiligen Arrayelemente dargestellt werden, festgelegt werden. Für jede Spalte, die einem Index [<n>] zugeordnet ist, ist eine individuelle Konfiguration möglich.

Eigenschaft	Beschreibung Wert
„Spalten“	Aufgrund der Struktur der Variablen die unter Daten-Array definiert ist, wird die Anzahl der Spalten ermittelt und über den Index <n> beschrieben.
▪ [<n>]	<pre>StringTabelle : ARRAY [0..2, 0..4] OF STRING := ['BMW', 'Audi', 'Mercedes', 'VW', 'Fiat', '150', '150', '150', '150', '100', 'blau', 'grau', 'silber', 'blau', 'rot']; es werden drei Spalten gebildet [0], [1] und [2].</pre>
„Max. Array-Index“	Optionale numerische Variable, über die ein maximaler Array-Index festgelegt werden kann.
„Zeilenhöhe“	Höhe der Zeilen in Pixel.
„Scrollbar Größe“	Breite der vertikalen Scrollbar in Pixel.

## Spalten: Spalte [<n>]

Eigenschaft	Beschreibung Wert
„Breite“	Breite der Spalte in Pixel
„Bildspalte“	Mit dieser Option können in dieser Spalte Bilder angezeigt werden. Wenn sie aktiviert ist, werden Bilder aus der globalen Bildersammlung oder von benutzerdefinierten Bildersammlungen in dieser Spalte der Tabelle angezeigt. Dabei legen die Werte der

Zellen einer Tabelle die ID des Bildes in der Bildersammlung fest.

„Bildkonfiguration“

„Füllmodus“

- „Füllmodus“
  - „Transparenz“
  - „Transparenzfarbe“
- Zelle ausfüllen: Das Bild wird auf die Zellengröße angepasst ohne Berücksichtigung des Breiten-Höhen-Verhältnisses.
  - Zentriert: Das Bild wird unter Beibehaltung der Proportionen innerhalb der Zelle zentriert. Selbst wenn die Höhe bzw. die Breite einzeln angepasst wird, bleibt das Breiten-Höhen-Verhältnis bestehen.

Wenn „Transparenz“ aktiviert ist, wird die unter „Transparenzfarbe“ eingestellte Farbe transparent dargestellt.

„Transparenzfarbe“: Auswahl einer Farbe aus der Combobox oder mit Hilfe des Farbauswahldialogs, welcher über die Schaltfläche  geöffnet werden kann. Die Farbe wird transparent dargestellt wenn die entsprechende Option „Transparenz“ aktiviert ist.

„Textausrichtung in Spalte“

Ausrichtung der Spalten:

- Links
- Zentriert
- Rechts

## Texte

Diese Eigenschaften werden für eine statische Definition der Elementbeschriftung verwendet.

Eigenschaft	Beschreibung Wert
„Tooltip“	Der angegebenen Text erscheint als Tooltip des Elements, wenn die Visualisierung im Online-Modus ist.

## Zustandsvariablen

Eigenschaft	Beschreibung Wert
„Unsichtbarkeit“	Eingabe einer boolschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.
„Eingaben deaktivieren“	Eingabe einer boolschen Variable; ist der Variablenwert TRUE, dann hat keine der Elementeingaben eine Auswirkung.

Eigenschaft	Beschreibung Wert
„Zugriffsrechte“	Mit Klick öffnet <a href="#">„Zugriffsrechte“</a> .

Statusmeldungen:

- Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.
- Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.

in Arbeit



## Registersteuerelement

Ein Registersteuerelement ermöglicht mehrere Formulare innerhalb eines Fensters. Reiter werden als Navigationswidget zum Umschalten zwischen den Formularen verwendet.

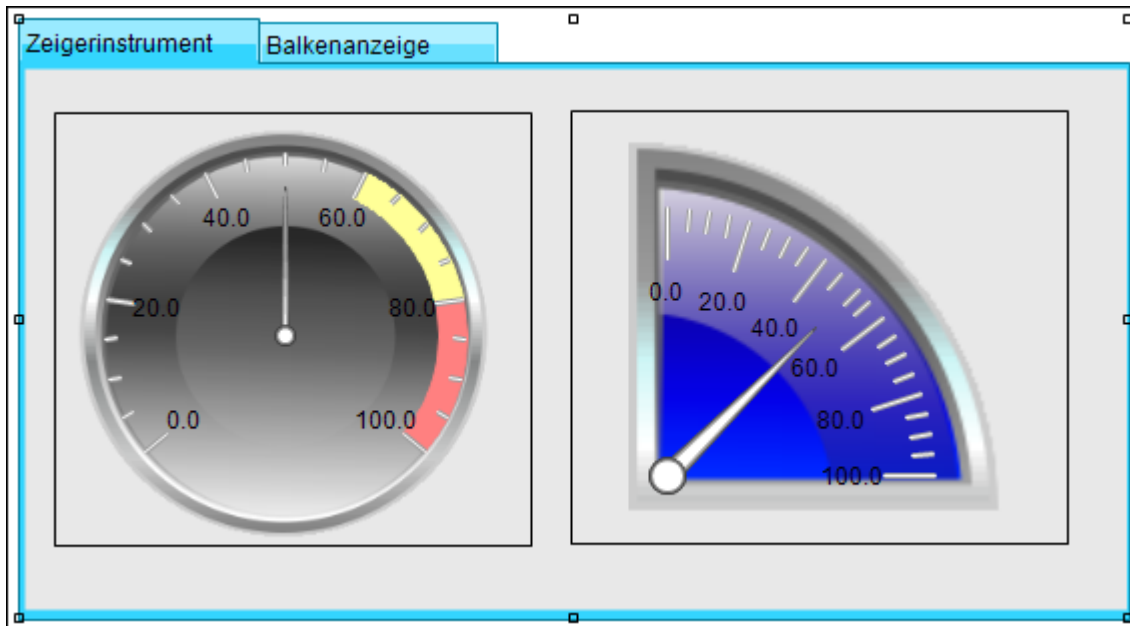


Abb. 62: Registersteuerelement, Stil 8

Wenn aufgrund der zu geringen Breite des Elements nicht alle Reiter angezeigt werden können, werden an dem Element oben rechts zwei Navigationspfeile eingeblendet.

### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungseditor werden folgende Benutzereingaben zur Verfügung gestellt:

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Registersteuerelement
„ <i>Registerbreite</i> “	Definiert die Breite des Reiters in Pixel.
Skalierungsart	Hier ist zu spezifizieren, wie das Fenster auf Änderungen der Größe zu reagieren hat:

- **Isotropisch:** das referenzierte Element behält seine Proportionen; somit bleibt, auch wenn Höhe und Breite unabhängig geändert wurden, das Verhältnis von Höhe und Breite der Visualisierung unverändert.
- **Anisotropisch:** das Registersteuerelement folgt der Größe, so dass Höhe und Breite der referenzierten Visualisierung unabhängig geändert werden können.
- **Unskaliert:** Die original Größe der Visualisierung bleibt erhalten, unabhängig von der Größe des Registersteuerelements.

*„Deaktivieren des Hintergrundzeichnens“*

Um die Performance der Visualisierung zu optimieren, werden die nicht-animierten Elemente des Frame-Elements als Hintergrund Bitmap gezeichnet. Diese könnte dazu führen, dass die Elemente nicht in der erwarteten Reihenfolge dargestellt werden. Um dieses Verhalten zu vermeiden, kann die Funktion deaktiviert werden.

**Referenzierte Visualisierungen**

Eigenschaft	Beschreibung Wert
<i>„Referenzierte Visualisierungen“</i>	Der Befehl <i>„Konfiguriere“</i> öffnet den Dialog für die <a href="#">„Frame-Auswahl“</a> .  Hier werden die referenzierten Visualisierungen des Elements aufgelistet.

**Position**

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
<i>„X“</i>	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
<i>„Y“</i>	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
<i>„Breite“</i>	Horizontale Breite des Elements.
<i>„Höhe“</i>	Vertikale Höhe des Elements.

**Umschaltvariable**

Diese Eigenschaft ermöglicht es, innerhalb eines Rahmens Visualisierungen umzuschalten.

Eigenschaft	Beschreibung Wert
<i>„Variable“</i>	Hier ist eine Integervariable anzugeben, deren Wert die ID der anzuzeigenden Visualisierung enthält. Die ID einer Visualisierung wird durch deren Reihenfolge in der Liste der zugewiesenen

Visualisierungen im Dialog „*Konfiguration der Framevisualisierungen*“ bestimmt. Zum Beispiel ergibt der erste Eintrag in dieser Liste die ID 0, der zweite Eintrag die ID 1, usw.

#### Zustandsvariablen

Eigenschaft	Beschreibung Wert
„ <i>Unsichtbarkeit</i> “	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.
„ <i>Eingaben deaktivieren</i> “	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann hat keine der Elementeingaben eine Auswirkung.

Eigenschaft	Beschreibung Wert
„ <i>Zugriffsrechte</i> “	Mit Klick öffnet <a href="#">„Zugriffsrechte“</a> .  Statusmeldungen: <ul style="list-style-type: none"> <li>▪ Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.</li> <li>▪ Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.</li> </ul>

#### Schaltfläche

Eine „*Schaltfläche*“ ermöglicht dem Benutzer, eine dem Steuerelement zugeordnete Funktion auszulösen. Optisch Anpassung ist durch Zuweisen eines Bildes möglich oder auch durch Konfigurieren einer Relieffansicht.

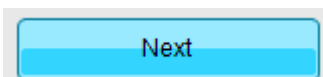


Abb. 63: Schaltfläche, Style 8

#### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungseditor werden folgende Benutzereingaben zur Verfügung gestellt:

#### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.

**Position**

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.



Eigenschaft	Beschreibung Wert
„X“	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„Y“	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„Breite“	Horizontale Breite des Elements.
„Höhe“	Vertikale Höhe des Elements.


**Zentrum**

Mit Bearbeiten des Wertes wird gleichzeitig das zugehörige Element  im Editor verschoben.

Eigenschaft	Beschreibung Wert
„X“	X-Position des Rotationspunkts des Elements
„Y“	Y-Position des Rotationspunkts des Elements

**Farben**

Eine Farbe wird mittels Hexzahl definiert, die sich aus Rot/Grün/Blau (RGB) Anteilen zusammensetzt. Für jede dieser drei Farben sind 256 (0-255) Werte verfügbar. Hier sind die Einstellungen **statisch**. Die Farbe kann aus einer Auswahlliste oder dem Farbauswahldialog, der über Button  öffnet, gewählt werden. Beispiel:  0; 128; 64

Eigenschaft	Beschreibung Wert
„Farbe“	Hier ist eine Farbe, mit der das Element im Normalzustand gefüllt wird, auszuwählen. Dies ist der Fall, wenn die Variable in <i>Farbvariablen</i> → <i>Farbumschlag</i> FALSE ist.  Mit Editieren des Werts ändert das Element gleichzeitig seine Farbe.
„Alarmfarbe“	Auszuwählen ist eine Farbe, mit der das Element gefüllt wird, wenn das Element in Alarmzustand ist. Dies ist der Fall, wenn die Variable in <i>Farbvariablen</i> → <i>Farbumschlag</i> TRUE ist.
„Farbverlauf verwenden“	Standardmäßig ist die Checkbox deaktiviert. Falls sie aktiviert wird, wird das zugehörige Element mit Farbverlauf gezeichnet.
„Farbverlaufsauswahl“	Dialog  „Farbverlaufeditor“ öffnet.
„Schaltflächenhöhe“	Angabe der Höhe in Pixel, die die Relieffansicht des Schaltflächenelements bestimmt:



## Bild

Eigenschaft	Beschreibung Wert
„Statische ID“	<p>Hier ist statisch der Identifikator der Bilddatei anzugeben, mit der die Schaltfläche darzustellen ist. Die ID (STRING) ist in der Bildersammlung definiert. Der Name der Sammlung sollte dabei vorangestellt werden, um den Eintrag eindeutig zu machen, außer die Bilddatei ist im GlobalImagePool verwaltet, da diese Sammlung immer als erste durchsucht wird. Mit Klick auf <input type="text" value="..."/> öffnet die Eingabehilfe und listet alle verfügbaren Bildersammlungen mit ihren Bildern auf.</p> <p>Beispiel: imagepool2.Schaltfläche_image</p>
„Skalierungsart“	<p>Hier ist anzugeben, wie die Bilddatei auf Änderungen der Schaltflächengröße reagieren soll:</p> <ul style="list-style-type: none"><li>▪ <b>Isotropisch:</b> Das Bild behält seine Proportionen; d.h. das Verhältnis Höhe-Breite bleibt erhalten, auch wenn Höhe und Breite der Schaltfläche separat verändert werden.</li><li>▪ <b>Anisotropisch:</b> Das Bild behält seine Proportionen; d.h. das Verhältnis Höhe-Breite bleibt erhalten, auch wenn Höhe und Breite der Schaltfläche separat verändert werden.</li><li>▪ <b>Unskaliert:</b> Das Bild behält die Originalgröße, auch wenn sich die Größe der Schaltfläche.</li></ul>
Transparenz	<p>Ist diese Option aktiviert, wird das Bitmap mit der in „<i>Transparenzfarbe</i>“ gesetzten Farbe, transparent dargestellt.</p>
Transparenzfarbe	<p>Angabe einer Transparenzfarbe anzuwenden beim Bitmap, falls die Option „<i>Transparent</i>“ aktiviert ist. Mit Klick auf <input type="text" value="..."/> öffnet ein Farbauswahldialog.</p>
Horizontale Ausrichtung	<p>Zu definieren ist die horizontale Ausrichtung des Bitmaps:</p> <ul style="list-style-type: none"><li>▪ Links</li><li>▪ Zentriert</li><li>▪ Rechts</li></ul>
Vertikale Ausrichtung	<p>Zu definieren ist der vertikalen Ausrichtung des Bitmaps:</p> <ul style="list-style-type: none"><li>▪ Oben</li><li>▪ Zentriert</li><li>▪ Unten</li></ul>

## Texte

Diese Eigenschaften werden für eine statische Definition der Elementbeschriftung verwendet.

Property	Value description
„Text“	Der angegebene Text beschriftet das Element. Ein textformatierender String ist möglich.
„Tooltip“	Der angegebenen Text erscheint als Tooltip des Elements, wenn die Visualisierung im Online-Modus ist.


### Texteigenschaften

Diese Eigenschaften werden für eine statische Definition der Schriftart verwendet.

Eigenschaft	Beschreibung Wert
„Schriftart“	<p>Auswahl aus vordefinierten Schriftarten:</p> <ul style="list-style-type: none"> <li>▪ Standard</li> <li>▪ Überschrift</li> <li>▪ Große</li> <li>▪ Titel</li> <li>▪ Anmerkung</li> </ul> <p>Mit Klick auf <input type="button" value="..."/> öffnet sich der Dialog für benutzerdefinierte Eigenschaften der Schriftart.</p>
„Farbe“	Auszuwählen ist eine Schriftartfarbe für das Element. Entweder aus der Auswahlliste oder durch
„Schriftart“	Setzen im sich öffnenden Dialog, wenn <input type="button" value="..."/> geklickt wird.

### Absolute Bewegung

Das Element kann bewegt werden, indem die X- und Y-Position (Pixel) der linken oberen Ecke des Elements durch eine Integer-Variable verändert wird. Hier werden absolute Koordinatenwerte verwendet.

Eigenschaft	Beschreibung Wert
„Bewegung“	<ul style="list-style-type: none"> <li>▪ „X“: Hier ist eine Integer-Variable einzugeben, deren Wert die aktuelle X-Position der linken oberen Ecke des Elements in Pixel definiert; kann verwendet werden, um das Element in X-Richtung zu verschieben (ein positiver Wert verschiebt das Element von links nach rechts).</li> <li>▪ „Y“: Eingabe einer Integer-Variablen, die die aktuelle Y-Position der linken oberen Ecke des Elements in Pixel definiert; kann verwendet werden, um das Element in Y-Richtung zu verschieben (ein positiver Wert verschiebt das Element von oben nach unten).</li> </ul>
„Rotation“	Eingabe einer Integer-Variablen, deren Wert den Winkel (Winkelgrade) definiert, um den das Element um den Rotationspunkt „Zentrum“  rotiert.

positiver Wert = mathematisch positiv = im Uhrzeigersinn

Zu beachten ist, dass das Element, im Gegensatz zum Verhalten bei „*Innere Rotation*“, selber nicht rotiert.

„*Skalierung*“ Eingabe einer Integer-Variablen, deren Wert den aktuellen Skalenwert (Prozent) definiert. Die Elementgröße wird linear entsprechend dieses Wertes verändert. Implizit wird der Wert durch 1000 dividiert, so dass es nicht nötig ist, REAL-Variablen zu verwenden um das Element zu verkleinern. Die Skalierung bezieht sich immer auf den Rotationspunkt „*Zentrum*“  $\oplus$ .

„*Innere Rotation*“ Die hier eingetragene Integer-Variable definiert den Winkel (Winkelgrade), um den das Element um seinen Rotationspunkt gedreht wird: „*Zentrum*“  $\oplus$ .

positiver Werte = mathematisch positiv = im Uhrzeigersinn.

Im Gegensatz zu „*Rotation*“ rotiert das Element tatsächlich.

## Relative Bewegung

Das Element kann relative zu seiner fixen Position definiert in „*Position*“ verschoben werden. Die obere linke und untere rechte Ecke des Elements werden dabei um die in den Integer-Variablen angegebenen Werte (Pixel) in X- bzw. Y-Richtung verschoben. Im Gegensatz zur absoluten Bewegung wird eine relative Position definiert, also der Abstand zur ursprünglichen Position. Das Element kann dadurch in der Form verändert werden.

Siehe „*Absolute Bewegung*“ für das Verschieben des Elements als Ganzes.

Eigenschaft	Beschreibung Wert
„ <i>Bewegung links oben</i> “	<ul style="list-style-type: none"><li>▪ „<i>X</i>“: Integer-Variable, deren Wert die Anzahl der Pixel angibt, um die die linke obere Ecke in X-Richtung verschoben wird.</li><li>▪ „<i>Y</i>“: Integer-Variable, deren Wert die Anzahl der Pixel angibt, um die die linke obere Ecke in Y-Richtung verschoben wird.</li></ul>
„ <i>Bewegung rechts unten</i> “	<ul style="list-style-type: none"><li>▪ „<i>X</i>“: „<i>X</i>“: Integer-Variable, deren Wert die Anzahl der Pixel angibt, um die die rechte untere Ecke in X-Richtung verschoben wird.</li><li>▪ „<i>Y</i>“: „<i>X</i>“: „<i>X</i>“: Integer-Variable, deren Wert die Anzahl der Pixel angibt, um die die rechte untere Ecke in Y-Richtung verschoben wird.</li></ul>

## Textvariablen

Dies sind dynamische Definitionen. Siehe auch „*Dynamische Texte*“ bei Verwendung von Textlisten. Für statische Definitionen siehe „*Texte*“

Eigenschaft	Beschreibung Wert
„ <i>Textvariable</i> “	Die zugewiesene Variable definiert den Elementtext ("Textausgabevariable"). Unter <a href="#">„Text und Sprache in Visualisierungen“</a> ist zu finden, wie ein Text zu formatieren ist.

„*Tooltipvariable*“ Die zugewiesene STRING-Variable enthält den Tooltip-Text für das Element.

### Dynamische Texte

Diese Parameter dienen der Definition von dynamischen Texten, die aus Textlisten stammen (dies erlaubt Sprachumschaltung). Siehe [Textliste](#) für nähere Informationen zur Verwendung von Textlisten.

Eigenschaft	Beschreibung Wert
„ <i>Textliste</i> “	Name der Textliste (String) wie im POU- oder Geräte-Fenster verwendet; z.B. <code>errorlist</code>
„ <i>Textindex</i> “	Index (ID) des Textes wie in der Textliste definiert (String); Direkte Angabe der ID, z.B. 'M1' oder String-Variable, z.B. <code>error_22</code> .
„ <i>Tooltipindex</i> “	Index (ID) des Tooltip-Textes wie in der Textliste definiert (String); Direkte Angabe der ID, z.B. 'T1' oder String-Variable, z.B. <code>error_Tooltip_22</code> .

### Schriftartvariablen

Diese werden verwendet bei dynamischen Schriftdefinitionen des Elementtextes über Projektvariablen. Statischen Definitionen werden unter [Texteigenschaften](#) → [Schriftart](#) konfiguriert.

Eigenschaft	Beschreibung Wert
„ <i>Schriftname</i> “	Hier ist eine Variable Typ STRING anzugeben, die den Schriftnamen enthält, der für die Beschriftung des Elements verwendet werden soll. (Namensangabe wie im Standard-Schrifttypen-Dialog). Beispiel: <code>plc_prg.font_var font_var := Arial;</code>
„ <i>Größe</i> “	Eingabe einer Variablen Typ INT, die die Größe des Elementtextes in Pixel enthält, so wie im Standarddialog Font. Beispiel: <code>prog1.font_height font_height:=16;</code>
„ <i>Flags</i> “	Eingabe einer Variable Typ DWORD, die über die unten genannten Flag-Werte die Font-Darstellung festlegt. Eine kombinierte Definition kann erreicht werden, indem die jeweiligen Flag-Werte addiert werden und die Summe angegeben wird. <ul style="list-style-type: none"> <li>▪ 1: kursiv</li> <li>▪ 2: fett</li> <li>▪ 4: unterstrichen</li> <li>▪ 8: gelöscht</li> </ul> <p><code>prog2.font_type: ist font_type:=6, wird der Text fett und unterstrichen dargestellt.</code></p>
„ <i>Zeichensatz</i> “	Der Zeichensatz, der für den Font verwendet werden soll, kann über die Standard-Zeichensatz-Nummer definiert werden. Mit einer DWORD-Variablen kann diese Nummer angegeben werden



(siehe auch die „Skript“-Definition im Standard-Font-Dialog)

„Farbe“ Eingabe einer DWORD-Variable, in der die Farbe des Elementtextes festgelegt wird.

## Farbvariablen

Eine Farbe wird mittels Hexzahl definiert, die sich aus Rot/Grün/Blau (RGB) Anteilen zusammensetzt. Für jede dieser drei Farben sind 256 (0-255) Werte verfügbar. Eine Projektvariable Typ DWORD muss angegeben werden, deren Wert die Farbe definiert (zum Beispiel `prog1.dwFillColor`).

Die Variable enthält den Farbwert in hexadezimalen Format. Um das gesamte DWORD auszufüllen, müssen die ersten beiden Ziffern nach "16#" mit 0 gesetzt werden: 16#00RRGGBB

### Farbwerte

```
dwFillColor := 16#008FE03F;
```

Hexzahl:

- 8F: rot
- E0: grün
- 3F: blau

Eigenschaft	Beschreibung Wert
„Farbumschlag“	Hier ist eine boolesche Variable angeben, die das Umschalten der Elementfarbe zwischen „Farbe“ (Variable = FALSE) und „Alarmfarbe“ (Variable = TRUE) steuert.
„Farbe“	Eingabe einer DWORD-Variable, die die Farbe des Elements bestimmt. Sie überschreibt den Wert, der aktuell in <i>Farben</i> → <i>Farbe</i> definiert ist, wenn die Variable in „Farbumschlag“ FALSE ist.
„Alarmfarbe“	Eingabe einer DWORD-Variable, die die Alarmfarbe des Elements bestimmt. Sie überschreibt den Wert, der aktuell in <i>Farben</i> → <i>Alarmfarbe</i> definiert ist, wenn die Variable in „Farbumschlag“ TRUE ist.

## Zustandsvariablen

Eigenschaft	Beschreibung Wert
„Unsichtbarkeit“	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.
„Eingaben deaktivieren“	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann hat keine der Elementeingaben eine Auswirkung.

## Schaltflächenzustandsvariable

Eigenschaft	Beschreibung Wert
-------------	-------------------

„Boolscher Wert“ Anzugeben ist eine boolsche Variable, die die Anzeige der Schaltfläche als "gedrückt" steuert, wenn die Variable TRUE ist und als "nicht gedrückt", wenn die Variable FALSE ist.

#### Bild-ID-Variable

Eigenschaft	Beschreibung Wert
„Bild-ID“	Anzugeben ist eine Projektvariable, die die Bild-ID enthält. Siehe dazu <a href="#">Bildersammlung</a>

#### Eingabekonfiguration

Hier kann definiert werden, welche Folgeaktionen ausgeführt werden, wenn der Benutzer im Online-Modus eine Eingabe im Element vornimmt.

Eingabeereignisse, die ein Element haben kann, sind:

- „OnDialogClosed“
- „OnClick“
- „MouseDown“
- „OnMouseEnter“
- „OnMouseLeave“
- „MouseMove“
- „MouseUp“

Mit Klick auf „Konfigurieren...“ wird der Dialog [„Eingabekonfiguration“](#) geöffnet, der erlaubt Folgeaktionen zuzuweisen. Man klickt in das Feld neben der zu konfigurierenden Eingabe. Für jede selektierte Folgeaktion wird ein Eintrag in der Eigenschaftentabelle erzeugt. Jede Eingabe kann eine beliebige zugeordnete Anzahl an Folgeaktionen haben.

Eigenschaft	Beschreibung Wert
„OnDialogClosed“	Dieses Ereignis tritt ein, wenn innerhalb der Visualisierung eines der für Benutzereingaben vorher geöffneten Dialogfenster geschlossen wird.
„OnClick“	Dieses Mausereignis tritt ein, wenn der Cursor auf ein Element zeigt und ein vollständiger Mausklick (drücken und freigeben der Maustaste) ausgeführt wird.
„MouseDown“	Dieses Mausereignis tritt ein, wenn die Maustaste gedrückt wird während der Cursor auf ein Element zeigt. (Drücken der Maustaste)
„OnMouseEnter“	Dieses Mausereignis tritt ein, wenn der Cursor auf das Element gezogen wird.
„OnMouseLeave“	Dieses Mausereignis tritt ein, wenn der Cursor das Element verlässt.
„MouseMove“	Dieses Mausereignis tritt ein, wenn der Cursor innerhalb des Elements bewegt wird.
„MouseUp“	Diese Mausereignis liegt vor, wenn die Maustaste losgelassen wird. Das Drücken der Maustaste geschah vorher, außerhalb des Elements.


#### Eingabekonfiguration: Tasten

Mit „*Tasten*“ kann konfiguriert werden, dass beim Ereignis "Tasten" der Wert einer booleschen Projektvariable abhängig vom Mausverhalten gesetzt wird.

Eigenschaft	Beschreibung Wert
„ <i>Variable</i> “	Hier kann eine boolesche Variable zugewiesen werden, deren Wert TRUE wird, wenn die Maustaste gedrückt wird, während der Cursor auf das Element zeigt und die wieder FALSE wird, wenn die Maustaste wieder losgelassen wird.
„ <i>FALSE tasten</i> “	Wenn diese Option aktiviert ist, wird das oben beschriebene Tastverhalten für die eingetragene Variable umgekehrt. Das heißt, wenn die Maustaste gedrückt wird, wird der Variablenwert auf FALSE gesetzt, wenn die Taste wieder losgelassen wird, wird der Variablenwert auf TRUE gesetzt.
„ <i>Beim Betreten tasten, falls Maus gefangen</i> “	Ist diese Option aktiviert, verhält sich der Variablenwert wie unter „ <i>Variable</i> “ beschrieben, solange sich die Maus innerhalb des Elementbereichs befindet. Zusätzlich wird auf ein Ziehen der Maus reagiert. Wird bei gedrückter Maustaste der Elementbereich verlassen, wird die Variable FALSE und wird wieder TRUE, wenn in den Elementbereich zurückgekehrt wird. Es wird also berücksichtigt, dass, solange die Maustaste gedrückt ist, die Maus "gefangen" ist, auch wenn der Elementbereich verlassen wird.

#### Eingabekonfiguration: Umschalten

Mit „*Umschalten*“ kann konfiguriert werden, dass bei diesem Ereignis der Wert einer booleschen Projektvariable abhängig vom Mausverhalten gesetzt wird.

Eigenschaft	Beschreibung Wert
„ <i>Variable</i> “	Hier ist eine boolesche Variable zuzuweisen. Die Eingabehilfe kann über Schaltfläche  geöffnet werden.  Der Wert der Variablen wird bei jedem Mausklick auf das Element zwischen TRUE und FALSE umgeschaltet. Wird der Cursor bei gedrückter Maustaste aus dem Element herausbewegt, dann erfolgt kein Toggeln. Damit kann die begonnene Umschalteingabe abgebrochen werden.
„ <i>Umschalten beim Loslassen falls Maus gefangen</i> “	Ist diese Option aktiviert, verhält sich der Variablenwert wie unter „ <i>Variable</i> “ beschrieben, solange sich die Maus innerhalb des Elementbereichs befindet. Zusätzlich wird auf ein Ziehen der Maus reagiert. Wird bei gedrückter Maustaste der Elementbereich verlassen und dann die Maustaste losgelassen, wird die Variable umgeschaltet. Es wird also berücksichtigt, dass, solange die Maustaste gedrückt ist, die Maus "gefangen" ist, auch wenn der Elementbereich verlassen wird.

#### Eingabekonfiguration: Hotkey

Mit „Hotkey“ kann eine Taste, auch mit Zusatzaste, definiert und mit einer bestimmten Nachfolgeaktion (MouseDown, MouseUp) verknüpft werden, die bei einem Ereignis der Taste (KeyDown, KeyUp) ausgeführt werden soll. Standardmäßig wird bei KeyDown (Taste drücken) die MouseDown-Aktion ausgeführt und bei KeyUp (Taste loslassen) die MouseUp-Aktion. Das kann nützlich sein, wenn eine Visualisierung sowohl über Maus- als auch Tastatureingaben bedient werden soll, weil die Eingabeaktionen dann nur einmalig konfiguriert werden müssen. Diese Tastenkonfiguration für ein Element wird auch im [„Tastaturkonfiguration“](#) der Visualisierung verwaltet. Änderungen werden immer zwischen diesem und dem Elementeigenschaften-Editor synchronisiert.

Eigenschaft	Beschreibung Wert
„Taste“	Zuweisen einer Taste. Eine Auswahlliste stellt alle aktuell unterstützten Tasten bereit wie zum Beispiel [ M ]. <a href="#">„Tastaturkonfiguration“</a>
„Eingabe(n)“	Definieren des Ereignisses, das ausgeführt werden soll, wenn die Taste beziehungsweise Taste und Modifizierer verwendet werden.  Mögliche Werte, die in einer Auswahlliste bereitstehen: <ul style="list-style-type: none"> <li>▪ Keine Aktion</li> <li>▪ MouseDown-Aktion: OnMouseDown-Aktion, wenn die Taste gedrückt wird.</li> <li>▪ MouseUp-Aktion: OnMouseUp-Aktion, wenn die Taste freigelassen wird.</li> <li>▪ MouseDown/MouseUp-Aktion: OnMouseUp- und OnMouseDown-Aktion, wenn die Taste gedrückt oder losgelassen wird.</li> </ul>
„Umschalten“	Wenn diese Option aktiviert wird, muss die Taste mit [ Umschalt ] kombiniert verwendet werden
„Steuerung“	Wenn diese Option aktiviert wird, muss die Taste mit [ Strg ] kombiniert verwendet werden
„Alt“	Wenn diese Option aktiviert wird, muss die Taste mit [ Alt ] kombiniert verwendet werden

## Groupbox

Elemente können mit Drag-and-Drop in eine Groupbox gezogen werden um eine optische Einheit zu bilden. Sie können mehrmals verschachtelt werden.

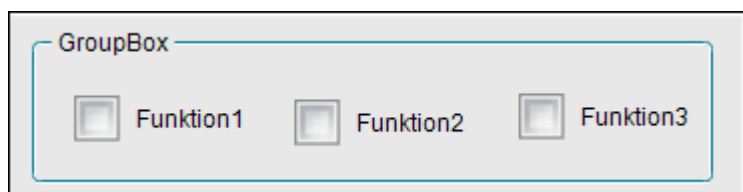


Abb. 64: Groupbox

### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungseditor werden folgende Benutzereingaben zur Verfügung gestellt:

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Groupbox

#### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„ <i>X</i> “	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„ <i>Y</i> “	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„ <i>Breite</i> “	Horizontale Breite des Elements.
„ <i>Höhe</i> “	Vertikale Höhe des Elements.

#### Texte

Diese Eigenschaften werden für eine statische Definition der Elementbeschriftung verwendet.

Property	Value description
„Text“	Der angegebene Text beschriftet das Element. Ein textformatierender String ist möglich.
„Tooltip“	Der angegebenen Text erscheint als Tooltip des Elements, wenn die Visualisierung im Online-Modus ist.

## Texteigenschaften

Diese Eigenschaften werden für eine statische Definition der Schriftart verwendet.

Eigenschaft	Beschreibung Wert
„Schriftart“	<p>Auswahl aus vordefinierten Schriftarten:</p> <ul style="list-style-type: none"> <li>▪ Standard</li> <li>▪ Überschrift</li> <li>▪ Große</li> <li>▪ Titel</li> <li>▪ Anmerkung</li> </ul> <p>Mit Klick auf <input type="text" value="..."/> öffnet sich der Dialog für benutzerdefinierte Eigenschaften der Schriftart.</p>
„Farbe Schriftart“	<p>Auszuwählen ist eine Schriftartfarbe für das Element. Entweder aus der Auswahlliste oder durch Setzen im sich öffnenden Dialog, wenn <input type="text" value="..."/> geklickt wird.</p>

## Zustandsvariablen

Property	Value description
„Unsichtbarkeit“	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.

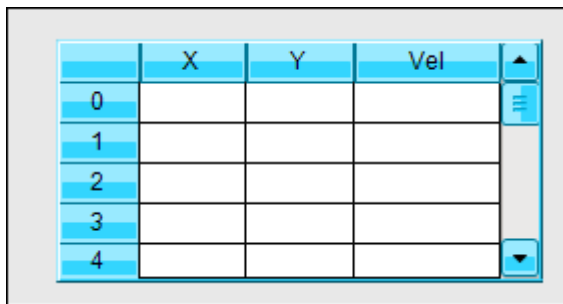
Eigenschaft	Beschreibung Wert
„Zugriffsrechte“	<p>Mit Klick öffnet <a href="#">„Zugriffsrechte“</a>.</p> <p>Statusmeldungen:</p> <ul style="list-style-type: none"> <li>▪ Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.</li> <li>▪ Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.</li> </ul>

## Table

Content of this page:

- [„Elementeigenschaften“](#)
- [„Zuweisung der Daten zur Tabelle“](#)
- [„Konfiguration mit Eigenschaften und die daraus entstehende Tabelle“](#)

A table can be inserted into a visualization to display one- or two-dimensional arrays, structures or local variables of a POU.



	X	Y	Vel	
0				
1				
2				
3				
4				

Abb. 65: Table without assigned data array, Style 8

### ▲ User input when editing an element


When editing an element in the visualization editor, the following user inputs are available:

Element-specific mouse input:

- The column width can be changed directly by dragging the separator line between two columns to the desired position while the cursor has the form  $\leftrightarrow$ .
- A table column can be **duplicated** by pressing and holding the `[ Strg ]` key and clicking on the header of the respective column.
- A table column can be **deleted** by pressing and holding the `[ Strg ]` and `[ Umschalttaste ]` keys and clicking on the header of the respective column.

### ▲ User input when editing the 'Properties'

Properties define the type of a visualization element:

	<p><b>HINWEIS!</b></p> <p>Wenn Sie die Größe der Variable, auf der die Tabelle basiert (Array-Grenzen, Anzahl der Strukturkomponenten etc.) im IEC-Code verändert haben, aber das betroffene Tabellenelement in der Visualisierung noch nicht aktualisiert haben, wird eine entsprechende Compiler-Fehlermeldung ausgegeben. Eine Möglichkeit ist dann, die Eigenschaftenkonfiguration zu öffnen und die Tabelle z.B. durch Drücken der <code>[ Eingabe ]</code>-Taste im Wertefeld der Eigenschaft "Datenarray". Eine viel komfortablere Möglichkeit ist jedoch einen Doppelklick auf die Fehlermeldung auszuführen. Dies</p>
---	--

	öffnet eine Meldungsbox, in der Sie „Ja“ oder „Nein“ zum unmittelbaren Aktualisieren der Tabellenkonfiguration sagen können.
--	--

## Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
-------------	-------------------


„*Elementname*“ Der Elementname kann geändert werden. Standardname ist "GenElemInst\_x", "x" steht für eine fortlaufende Nummerierung.

„*Elementtyp*“ Tabelle

Eigenschaft	Beschreibung Wert
-------------	-------------------

„*Datenarray*“ Hier wird die zu visualisierende Variable mit vollständigem Pfad spezifiziert. Die Struktur der Variablen bestimmt die Anzahl der Spalten und Reihen der Tabelle. Wenn die Anzahl der Array-Elemente der Variable variiert, verwenden Sie „*Max. Array-Index*“. Um die Tabelle zu aktualisieren, falls sich die Größe der Array- oder Strukturvariablen geändert hat, setzen Sie den Cursor in das Datenarray-Wertefeld und drücken [ *Eingabe* ].

„*Max. Array-Index*“ Zuzuweisen ist eine numerische Variable, die „*dynamisch*“ den maximalen Arrayindex festlegt.

	<p><b>HINWEIS!</b></p> <p>Wird der „<i>Datenarray</i>“-Wert geändert, also eine neue Variable zugewiesen, werden die weiteren Tabelleneigenschaften auf ihren Standardwert zurückgesetzt.</p>
---	---

## Spalten

Das Tabellenelement visualisiert eine Variable in Tabellenansicht. Der Index des Arrays bzw. die Komponente der Struktur wird in einer Spalte bzw. Zeile dargestellt, bei zweidimensionalen Arrays bzw. einem Array von Strukturen wird in mehreren Spalten visualisiert. Hier wird das Aussehen der Tabellenspalten definiert, in welchen die Werte der einzelnen Arrayelemente/Strukturkomponenten/Variablen angezeigt werden. Für jede zu einem bestimmten Index gehörenden Spalte ist eine eigene Konfiguration möglich.

Eigenschaft	Beschreibung Wert
-------------	-------------------

„*Spalte*“ Gemäß der Variablenstruktur, definiert in „*Datenarray*“, wird die Anzahl an Spalten identifiziert und in „*<n>*“ indiziert.

- [*<n>*]

var\_data\_array: ARRAY [2..8] OF INT: **eine Spalte** Column [0] **wird identifiziert.**

var\_data\_array: ARRAY[1..3][1..10] OF INT: **drei Spalten** [0], [1] **und** [2] **werden identifiziert**



„Zeilenkopf anzeigen“	Checkbox, um die Anzeige der Beschriftung der Zeilen in Form von zugeordneten Zeilenindizes ein- bzw. auszuschalten
„Spaltenkopf anzeigen“	Checkbox, um die Anzeige der Beschriftung der Spalten in Form von zugeordneten Zeilenindizes ein- bzw. auszuschalten
„Zeilenhöhe“	Höhe der Tabellenzeilen in Pixel
„Breite Zeilenkopf“	Breite der Spalte, die den Zeilenkopf enthält, in Pixel
„Größe des Scrollbars“	Breite des vertikalen bzw. Höhe des horizontalen Scrollbalkens in Pixel

**Spalten: Spalte [<n>]**

Eigenschaft	Beschreibung Wert
„Spaltenüberschrift“	Standardmäßig wird der Name des Array/der Struktur mit dem zur Spalte gehörenden Index/Strukturkomponente als Überschrift genommen. Die Spaltenbeschriftung kann hier geändert werden, indem ein neuer Titel eingetragen wird.
„Breite“	Spaltenbreite in Pixel
„Bildspalte“	Wird diese Option aktiviert, können in der Spalte Bilder angezeigt werden. Es werden Bilder der globalen Bildersammlung oder von benutzerdefinierten Bildersammlungen verwendet. Dabei stehen in den Zellen der Tabelle die Bild-IDs, wie in der Bildersammlung definiert.
„Bildkonfiguration“	<p>„Füllmodus“</p> <ul style="list-style-type: none"> <li>▪ „Füllmodus“</li> <li>▪ „Transparenz“</li> <li>▪ „Transparenzfarbe“</li> </ul> <p>▪ Zelle ausfüllen: Das Bild passt sich an die Größe der Zelle ohne Berücksichtigung des Höhen-Breiten-Verhältnisses an.</p> <p>▪ Zentriert: Das Bild wird in der Zelle zentriert und behält seine Proportionen; d.h. das Verhältnis Höhe-Breite bleibt erhalten.</p> <p>Ist „Transparenz“ aktiviert, wird die Farbe, die in „Transparenzfarbe“ angegeben ist, transparent dargestellt.</p> <p>„Transparenzfarbe“: Einzugeben ist eine Farbe. Sie kann auch mit Hilfe von , das einen Farbauswahldialog öffnet, ausgewählt werden. Sie wird transparent dargestellt, wenn „Transparenz“ aktiviert ist.</p>
„Textausrichtung der Überschrift“	<p>Ausrichtung des Spaltenkopfs:</p> <ul style="list-style-type: none"> <li>▪ Links</li> <li>▪ Zentriert</li> <li>▪ Rechts</li> </ul>
„Vorlage verwenden“	Bei Aktivierung der Option wird in jede Zelle dieser Tabellenspalte ein weiteres

Visualisierungselement (vom Typ „*Rechteck*“, „*Abgerundetes Rechteck*“ oder „*Ellipse*“) eingefügt. Die Eigenschaftenliste wird automatisch um die Eigenschaften dieses Elements unter „*Vorlage*“ erweitert.

„*Textausrichtung der Überschrift aus der Vorlage*“

Die Aktivierung dieser Option hat nur bei gleichzeitiger Aktivierung von „*Vorlage verwenden*“ eine Auswirkung.

Bei Aktivierung werden die in der eingefügten Vorlage getroffenen Einstellungen für Schrift(-Größe) und Ausrichtung auch auf die Spaltenüberschrift angewendet.

„*Vorlage*“

Dieser Eintrag ist nur vorhanden, wenn „*Vorlage verwenden*“ aktiviert ist.

Unter „*Vorlage*“ werden sukzessive die Eigenschaften aller Elemente, die der Spalte zugewiesen wurden, aufgelistet und können dort, wie unter [„Rechteck, abgerundetes Rechteck und Ellipse“](#) beschrieben, editiert werden.

## Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„ <i>X</i> “	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„ <i>Y</i> “	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„ <i>Breite</i> “	Horizontale Breite des Elements.
„ <i>Höhe</i> “	Vertikale Höhe des Elements.

## Texteigenschaften

Diese Eigenschaften werden für eine statische Definition der Schriftart verwendet.

Eigenschaft	Beschreibung Wert
„ <i>Horizontale Ausrichtung</i> “	Definieren der horizontalen Ausrichtung des Textes durch Auswahl aus: <ul style="list-style-type: none"> <li>▪ Links</li> <li>▪ Zentriert</li> <li>▪ Rechts</li> </ul>
„ <i>Vertikale Ausrichtung</i> “	Definieren der vertikalen Ausrichtung des Textes durch Auswahl aus: <ul style="list-style-type: none"> <li>▪ Oben</li> <li>▪ Zentriert</li> </ul>

- Unten

#### „Schriftart“

Auswahl aus vordefinierten Schriftarten:

- Standard
- Überschrift
- Größe
- Titel
- Anmerkung

Mit Klick auf  öffnet sich der Dialog für benutzerdefinierte Eigenschaften der Schriftart.

#### „Farbe Schriftart“

Auszuwählen ist eine Schriftartfarbe für das Element. Entweder aus der Auswahlliste oder durch Setzen im sich öffnenden Dialog, wenn  geklickt wird.

### Dynamische Texte

Diese Parameter dienen der Definition von dynamischen Texten, die aus Textlisten stammen (dies erlaubt Sprachumschaltung). Siehe [Textliste](#) für nähere Informationen zur Verwendung von Textlisten.

Eigenschaft	Beschreibung Wert
„Textliste“	Name der Textliste (String) wie im POU- oder Geräte-Fenster verwendet; z.B. <code>errorlist</code>
„Textindex“	Index (ID) des Textes wie in der Textliste definiert (String); Direkte Angabe der ID, z.B. 'M1' oder String-Variable, z.B. <code>error_22</code> .
„Tooltipindex“	Index (ID) des Tooltip-Textes wie in der Textliste definiert (String); Direkte Angabe der ID, z.B. 'T1' oder String-Variable, z.B. <code>error_Tooltip_22</code> .

### Schriftartvariablen

Diese werden verwendet bei dynamischen Schriftdefinitionen des Elementtextes über Projektvariablen. Statischen Definitionen werden unter [Texteigenschaften](#) → [Schriftart](#) konfiguriert.

Eigenschaft	Beschreibung Wert
„Schriftname“	Hier ist eine Variable Typ STRING anzugeben, die den Schriftnamen enthält, der für die Beschriftung des Elements verwendet werden soll. (Namensangabe wie im Standard-Schrifttypen-Dialog). Beispiel: <code>plc_prg.font_var font_var := Arial;</code>
„Größe“	Eingabe einer Variablen Typ INT, die die Größe des Elementtextes in Pixel enthält, so wie im Standarddialog Font. Beispiel: <code>prog1.font_height font_height:=16;</code>
„Flags“	Eingabe einer Variable Typ DWORD, die über die unten genannten Flag-Werte die Font-Darstellung festlegt. Eine kombinierte Definition kann erreicht werden, indem die jeweiligen Flag-Werte addiert werden und die Summe angegeben wird.

- 1: kursiv

- 2: fett
- 4: unterstrichen
- 8: gelöscht

`prog2.font_type: ist font_type:=6`, wird der Text fett und unterstrichen dargestellt.

„*Zeichensatz*“ Der Zeichensatz, der für den Font verwendet werden soll, kann über die Standard-Zeichensatz-Nummer definiert werden. Mit einer DWORD-Variablen kann diese Nummer angegeben werden (siehe auch die „*Skript*“-Definition im Standard-Font-Dialog)

„*Farbe*“ Eingabe einer DWORD-Variablen, in der die Farbe des Elementtextes festgelegt wird.

#### Zustandsvariablen

Eigenschaft	Beschreibung Wert
„ <i>Unsichtbarkeit</i> “	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.
„ <i>Eingaben deaktivieren</i> “	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann hat keine der Elementeingaben eine Auswirkung.

#### Selektion

Eigenschaft	Beschreibung Wert
„ <i>Selektionsfarbe</i> “	Füllfarbe für selektierte Tabellenzellen.
„ <i>Selektionstyp</i> “	Definiert, wie beim einem Klick auf eine Tabellenzelle die Auswahl erfolgt: <ul style="list-style-type: none"> <li>▪ <code>Keine Auswahl</code>: Keine Auswahl</li> <li>▪ <code>Zellenauswahl</code>: nur die angeklickte Zelle ist ausgewählt.</li> <li>▪ <code>Zeilenauswahl</code>: Zeilenauswahl, das heißt die Zeile, in der sich die angeklickte Zelle befindet, ist selektiert.</li> <li>▪ <code>Spaltenauswahl</code>: Spaltenauswahl, das heißt die Spalte, in der sich die angeklickte Zelle befindet, ist selektiert.</li> <li>▪ <code>Zeilen- und Spaltenauswahl</code>: Zeilen- und Spaltenauswahl, das heißt die Zeile und Spalte, in der sich die angeklickte Zelle befindet, ist selektiert.</li> </ul>
„ <i>Rahmen um ausgewählte Zellen</i> “	Ist diese Option aktiviert, wird um die selektierten Zellen ein Rahmen gezeichnet.
„ <i>Variable für Spaltenauswahl</i> “	Variable Typ INT, die den Array-Index der <b>Spalte</b> der selektierten Zelle enthält. Im Fall, dass der Datenarray auf eine Struktur zeigt, werden die Strukturkomponenten beginnend mit 0 indiziert. Zu beachten ist, dass dieser Index nur dann die korrekte Position im Array repräsentiert, wenn keine Spalten von der Anzeige in der Tabelle ausgenommen wurden.
„ <i>Variable für</i> “	Variable Typ INT, die den Array-Index der <b>Zeile</b> der selektierten Zelle enthält.

## Zeilenauswahl

„Variable für Gültigkeit der Spaltenauswahl“ Boolesche Variable, die dann TRUE ist, wenn „Variable für Spaltenauswahl“ einen gültigen Wert enthält.

„Variable für Gültigkeit der Zeilenauswahl“ Boolesche Variable, die dann TRUE ist, wenn „Variable für Zeilenauswahl“ einen gültigen Wert enthält.

Eigenschaft	Beschreibung Wert
„Zugriffsrechte“	Mit Klick öffnet <a href="#">„Zugriffsrechte“</a> .  Statusmeldungen: <ul style="list-style-type: none"><li>▪ Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.</li><li>▪ Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.</li></ul>

## Zuweisung der Daten zur Tabelle

### Beispiel

Zunächst ist die in Tabellenansicht zu visualisierende Variable zu spezifizieren. Dazu wird in „Datenarray“ eine Variable mit vollständigem Pfad eingetragen. Die Struktur der Variable bestimmt die Anzahl an Zeilen und Spalten der Tabelle.

Die **Spalten** repräsentieren im Fall eines zweidimensionalen Arrays den zweiten Index; im Fall einer Struktur repräsentieren sie die einzelnen Komponenten der Struktur, im Fall einer POU die lokalen Variablen. So wird beispielsweise ein eindimensionales Array von Elementen eines einfachen Basistyps mit Hilfe einer einzigen Spalte repräsentiert, eine Struktur wird durch eine einzige Zeile repräsentiert, wohingegen ein Array von Strukturen in einer Tabelle dargestellt wird, deren Anzahl an Zeilen der Anzahl an Array Elementen entspricht und deren Anzahl an Spalten der Anzahl an Strukturkomponenten entspricht.

Dann ist eine Konfiguration der Tabelle mit Hilfe ihrer weitem Eigenschaften möglich.

### Zugewiesene Daten und die daraus entstehende Tabelle

Deklaration der Variablen	Entstehende Tabelle
---------------------------	---------------------

```
arrDim1 :
ARRAY[2..
5] OF
INT;
```

	arrDim1[INDEX]
2	
3	
4	
5	

Beachten Sie, dass die Indizierung in der Zeilenbeschriftung mit den Indizes der ersten Dimension des Arrays übereinstimmt; infolgedessen startet die Nummerierung der Zeilen in diesem Beispiel mit einer "2"!

```
arrDim2 :
ARRAY[0..
2, 0..3]
OF INT;
```

	arrDim2[0,INDEX]	arrDim2[1,INDEX]	arrDim2[2,INDEX]
0			
1			
2			
3			

```
TYPE
MyStruct
:
```

	varStruct.iNo	varStruct.bBool	varStruct.sText
0			

```
STRUCT
```

```
iNo: INT;
```

```
bBool :
BOOL;
```

```
sText:STR
ING;
```

```
END_STRUC
T
```

```
END_TYPE
```

```
varStruct
:
MyStruct;
```

```
arrStruct
:
ARRAY[0..
3] OF
MyStruct;
```

	arrStruct[INDEX].iNo	arrStruct[INDEX].bBool	arrStruct[INDEX].sText
0			
1			
2			
3			

```
varFB:
TON;
```

	varFB.IN	varFB.PT	varFB.Q	varFB.ET	varFB.M
0					

```
arrFB :
ARRAY[0..
3] OF
TON;
```

	arrFB[INDEX].IN	arrFB[INDEX].PT	arrFB[INDEX].Q	arrFB[INDEX].ET	al
0					
1					
2					
3					

### Konfiguration mit Eigenschaften und die daraus entstehende Tabelle

Neben einigen grundlegenden Eigenschaften wie z. B. Größe, Position oder Linienbreite der Tabelle können eine Vielzahl von spezifischen Eigenschaften für die Tabelle eingestellt werden. Eine Auswahl dieser Eigenschaften soll anhand einer schrittweisen Weiterführung des **Beispiels arrStruct** näher erläutert werden, die Auswirkungen sind jeweils in der rechten Spalte der unten stehenden Tabelle aufgezeigt.

#### Beispiel

Geänderte Eigenschaften	Daraus entstehende Tabelle		
Zunächst ändern wir die standardmäßigen Einträge in der Spaltenbeschriftung in aussagekräftigere Titel ab, passen die Breite der Spalten an und legen die Ausrichtung der Überschrift fest; z.B. für die erste Spalte	Bestellung in Stück	auf Lager?	
	0		
	1		
	2		
	3		

Columns	
Column	
[0]	<input checked="" type="checkbox"/>
Column header	Bestellung in Stück
Column width	150
Text in headline	LEFT

Es gibt zwei Möglichkeiten, die Breite der Spalten zu ändern: 1. In den Elementeeigenschaften: Eintragen der gewünschten Breite in *Spalte → Breite* oder 2. Direkt im Tabellenelement: Mausklick auf den Trennstrich zwischen zwei Spalten in der Titelzeile und mit der Maus an die gewünschte Position ziehen.

Wir setzen die Höhe der Spalten auf 20 Pixel fest und schalten die Beschriftung der Zeilen aus, so dass nun die erste Spalte, die die Arrayindizes beinhaltet, fehlt. Das Bild rechts zeigt zudem die verbreiterten Bildlaufleisten („Größe des Skrollbars“), welche zur Tabelle hinzugefügt werden, sobald diese auf eine Größe unter der Summe der Zeilenhöhen bzw. der

Bestellung in Stück	auf Lager?	Art

Spaltenbreiten geschrumpft wird.

Columns	
+ Column	
Row header	<input type="checkbox"/>
Column header	<input checked="" type="checkbox"/>
Row height	20
Row header width	15
Scrollbar size	40

Um die Bildlaufleisten wieder verschwinden zu lassen, passen wir die Größe der Tabelle an: ihre Höhe ergibt sich aus der Zeilenhöhe multipliziert mit der Anzahl an Zeilen (+1 im Fall von eingeschalteter Spaltenbeschriftung); ihre Breite ist die Summe aus den einzelnen Spaltenbreiten und der Breite der Zeilenbeschriftung (sofern diese aktiviert ist). So muss zu diesen Zahlen noch ein kleiner Überhang (von bspw. 1) addiert werden.

Position	
X	21
Y	221
Width	486
Height	101
Text properties	
HorizontalAlignment	HCENTER
VerticalAlignment	VCENTER
Font	<input checked="" type="checkbox"/> Tahoma; 12px

Schriftart, -farbe und -größe können innerhalb der Dialogbox ausgewählt werden, welche nach einem Klick auf die Schaltfläche  in „Schriftart“ öffnet. Beachten Sie, dass die Vorgabe für die horizontale Textausrichtung („Textausrichtung der Überschrift“) nur für die Zeilenbeschriftung gilt; die Ausrichtung der Texte in der einzelnen Tabellenzellen folgt in den hierfür getroffenen Einstellungen in den Spalteneigenschaften.

Die Tabellenfelder selbst können zeilenweise editiert werden: Aktivieren Sie zu diesem Zweck für die ausgewählte Spalte das Kontrollkästchen „Vorlage verwenden“, woraufhin innerhalb der Spalteneigenschaften ein weiterer Punkt „Vorlage“

Die (online) Darstellung der Tabelle zeigt nun ein verändertes Aussehen:






	Bestellung in Stück	auf Lager?	
0	250	TRUE	
1	480	FALSE	
2	1500	TRUE	
3	2500	TRUE	

Beachten Sie, dass die Ausrichtung der Einträge in den Tabellenzellen entsprechend der Einstellung für die einzelnen Spalten erfolgt.

In unten stehendem Beispiel wurde für die erste und dritte Spalte ein Template benutzt, wobei das Template der ersten Spalte die links aufgezeigten Eigenschaften besitzt.



eingefügt und aufgeklappt wird:

Use template	<input checked="" type="checkbox"/>
Use text alignment in headline	<input type="checkbox"/>
<b>Template</b>	
Type of element	VISU_ST_RECTANGLE
<b>Colors</b>	
<b>Normalstate</b>	
Framecolor	 Coral
Fillcolor	 Bisque
<b>Alarmstate</b>	
Framecolor	 OrangeRed
Fillcolor	 LightSalmon
<b>Elementlook</b>	
Linewidth	1
Fillattributes	BS_SOLID
Frameattributes	PS_DASH
<b>Texts</b>	
Text	%s
Tooltip	
<b>Text properties</b>	
HorizontalAlignment	RIGHT
VerticalAlignment	VCENTER
Font	 Copperplate Gothic Light
<b>Text variables</b>	
Text Variable	PLC_PRG.arrStruct[INDEX].

	Bestellung in Stück	auf Lager?	
0	250	TRUE	D
1	480	FALSE	D
2	1500	TRUE	D
3	2500	TRUE	A

Drei Typen von Vorlagen stehen zur Auswahl: standardmäßig ist ein Rechteck eingetragen, das jedoch durch Klicken in das entsprechende Feld in ein Rechteck mit abgerundeten Ecken oder einer Ellipse geändert werden kann. Die zugehörigen Konfigurationsmöglichkeiten zeigen die Elementeigenschaften der ausgewählten Vorlage: wie gewöhnlich können Sie die Füll- und Rahmenfarben im Normal- und Alarmzustand wählen, die auch mit Hilfe einer Variablen getoggelt werden können. Die Ausrichtung der Einträge in den Zellen der Spalte wird nun (mit Ausnahme der Überschrift) durch die Einstellung im Template bestimmt.

Hinweis: Anstelle der standardmäßig eingetragenen Array-Komponente kann in „*Textvariable*“ auch jede andere Variable des Projekts eingetragen werden. Somit ist es insbesondere möglich, die Array-Elemente auch in veränderter Reihenfolge in der

Tabelle anzeigen zu lassen.

Ein Vertauschen der Spalten kann auch in der Tabelle selbst vorgenommen werden. Klicken Sie hierzu innerhalb der Titelzeile in die Mitte einer Spalte und ziehen die Spalte mit gedrückter Maustaste auf ihre neue Position.

Der Alarmzustand kann nicht nur durch einen Variablenwert ausgelöst werden, sondern auch durch Selektieren einer Tabellenzelle durch einen Mausklick im Online Modus. Das Verhalten der selektierten Region wird im Abschnitt „*Selektion*“ festgelegt:

Selection	
Selection color	HotPink
Apply to columns	
Selection type	SEL_ROW
Frame around selected cells	
Variable for selection X	PLC_PRG.Select
Variable for selection Y	PLC_PRG.Select
Variable for valid selection X	PLC_PRG.IsSel
Variable for valid selection Y	PLC_PRG.IsSel

Die hier festgesetzte Hintergrundfarbe für die selektierten Elemente gilt für alle selektierten Zellen, für welche keine Alarmfarbe (innerhalb eines Templates) vorgeschrieben wurde. So wird in unserem Beispiel eine selektierte Zelle der zweiten Spalte in HotPink eingefärbt, wohingegen eine selektierte Zelle der ersten Spalte die Alarm Farbe des zugehörigen Templates (LightSalmon) erhält.





Wenn Sie den Schalter „*Anwenden*“ drücken, so werden die Alarmfarben der speziellen Vorlage durch die in „*Selektionsfarbe*“ getroffenen Auswahl ersetzt. Folglich erhalten daraufhin die entsprechenden Felder im Template der ersten Spalte unserer Beispiels folgende Einträge:

Noch einmal das Table Element im Online Modus: Nach Anklicken der Zelle mit dem Eintrag "FALSE" gelten alle Zellen in der zweiten Zeile als selektiert. Nur die selektierte Zelle in der zweiten Spalte trägt die angegebene Auswahlfarbe, nachdem die Farben des Alarmzustands in den anderen Spalten durch die Templates vorgeschrieben werden:

	Bestellung in Stück	auf Lager?	
0	250	TRUE	D
1	480	FALSE	D
2	1500	TRUE	D
3	2500	TRUE	A

Nach Anwenden der Auswahlfarbe auch für die Templates erhält die Selektion ein verändertes Aussehen:

	Bestellung in Stück	auf Lager?	
0	250	TRUE	D
1	480	FALSE	D
2	1500	TRUE	D
3	2500	TRUE	A

Template		
Type of element	VISU_ST_RECTANGLE	
Colors		
Normalstate		
Framecolor		Coral
Fillcolor		Bisque
Alarmstate		
Framecolor		OrangeRed
Fillcolor		HotPink

„Auswahlart“ bestimmt, welche Elemente durch Anklicken einer einzelnen Tabellenzelle selektiert werden sollen: nachdem wir „Zeilenauswahl“ gewählt haben, werden alle Zellen, die zur selben Zeile gehören wie die angeklickte Zelle, als selektiert angesehen und umrahmt.

Wurden keine Spalten von der Darstellung in der Tabelle ausgeschlossen, so repräsentiert die Position der selektierten Zelle innerhalb der Tabelle die zum Eintrag gehörenden Indices des visualisierten Arrays. Diese Indizes können dann in Variablen gespeichert werden, die hierfür in den letzten drei Zeilen des Abschnitts „Selektion“ eingetragen werden. Die dritte und vierte Variable sind von booleschem Typ und werden auf TRUE gesetzt, sofern die Selektion gültig ist, was die Zeile bzw. Spalte anbelangt.

Die verknüpften Variablen erhalten als Wert den zur ausgewählten Zelle gehörenden Arrayindex:

CoDeSys_SP_Win_V3.Application.PLC_PRG		
Ausdruck	Datentyp	Wert
arrDim1	ARRAY [2..5] OF INT	
arrDim2	ARRAY [0..2, 0..3] O...	
varStruct	MyStruct	
arrStruct	ARRAY [0..3] OF My...	
varFB	TON	
arrFB	ARRAY [0..3] OF TON	
iCounter	INT	277
iRow	INT	0
iCol	INT	7
SelectedPosX	INT	2
SelectedPosY	INT	1
IsSelected	BOOL	TRUE

## Textfeld

Mit diesem Element ist es möglich Text anzuzeigen, der direkt in den Eigenschaften oder variabel in „*Textvariable*“ zugewiesen wird. Im Gegensatz zum Rechteck kann der Rahmen mit Schatten dargestellt werden.

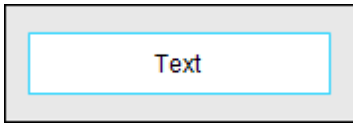


Abb. 66: Textfeld

### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungsektor werden folgende Benutzereingaben zur Verfügung gestellt:

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.



Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Textfeld

#### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„ <i>X</i> “	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„ <i>Y</i> “	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„ <i>Breite</i> “	Horizontale Breite des Elements.
„ <i>Höhe</i> “	Vertikale Höhe des Elements.

#### Farben

Eine Farbe wird mittels Hexzahl definiert, die sich aus Rot/Grün/Blau (RGB) Anteilen zusammensetzt. Für jede dieser drei Farben sind 256 (0-255) Werte verfügbar. Hier sind die Einstellungen **statisch**. Die Farbe kann aus einer Auswahlliste oder dem Farbauswahldialog, der über Button  öffnet, gewählt werden. Beispiel:  0; 128; 64

Eigenschaft	Beschreibung Wert
„Normalzustand“	Auszuwählen ist eine Rahmen- und Füllfarbe für Normalzustand. Im Fall, dass die Variable definiert in <i>Farbvariablen</i> → <i>Farbumschlag</i> FALSE ist, ist das Element in Normalzustand.
<ul style="list-style-type: none"> <li>▪ „Rahmenfarbe“</li> <li>▪ „Füllfarbe“</li> </ul>	
„Alarmzustand“	Auszuwählen ist eine Rahmen- und Füllfarbe, mit der das Element im Alarmzustand dargestellt wird. Dies ist der Fall, wenn die Variable in <i>Farbvariablen</i> → <i>Farbumschlag</i> TRUE ist.
<ul style="list-style-type: none"> <li>▪ „Rahmenfarbe“</li> <li>▪ „Füllfarbe“</li> </ul>	

## Aussehen

Eigenschaft	Beschreibung Wert
„Linienstärke“	Stärke der Rahmenlinie in Pixel. 0 kodiert das Gleiche wie 1 und setzt die Linienstärke auf 1 Pixel. Wird kein Rahmen gewünscht, muss man „Linienart“ auf unsichtbar setzen.
„Füllart“	Auszuwählen ist eines dieser Füllarten angewendet in „Füllfarbe“ <ul style="list-style-type: none"> <li>▪ Ausgefüllt: Die Füllfarbe ist sichtbar</li> <li>▪ Unsichtbar: Die Füllfarbe ist unsichtbar</li> </ul>
„Linienart“	Auszuwählen ist eine der Linienarten angewendet bei der Umrisslinie: <ul style="list-style-type: none"> <li>▪ Durchgezogen</li> <li>▪ Striche</li> <li>▪ Punkte</li> <li>▪ Strich Punkt</li> <li>▪ Strich Punkt Punkt</li> <li>▪ Unsichtbar: Umrisslinie ist nicht sichtbar.</li> </ul>
„Schattenart“	Art des Schattens der Umrisslinie <ul style="list-style-type: none"> <li>▪ Versenkt</li> <li>▪ Kein Schatten</li> <li>▪ Erhaben</li> <li>▪ Aus Stil</li> </ul>

## Texte

Diese Eigenschaften werden für eine statische Definition der Elementbeschriftung verwendet.

Property	Value description
<i>„Text“</i>	Der angegebene Text beschriftet das Element. Ein textformatierender String ist möglich.
<i>„Tooltip“</i>	Der angegebenen Text erscheint als Tooltip des Elements, wenn die Visualisierung im Online-Modus ist.

## Texteigenschaften

Diese Eigenschaften werden für eine statische Definition der Schriftart verwendet.

Eigenschaft	Beschreibung Wert
<i>„Horizontale Ausrichtung“</i>	<p>Definieren der horizontalen Ausrichtung des Textes durch Auswahl aus:</p> <ul style="list-style-type: none"> <li>▪ Links</li> <li>▪ Zentriert</li> <li>▪ Rechts</li> </ul>
<i>„Vertikale Ausrichtung“</i>	<p>Definieren der vertikalen Ausrichtung des Textes durch Auswahl aus:</p> <ul style="list-style-type: none"> <li>▪ Oben</li> <li>▪ Zentriert</li> <li>▪ Unten</li> </ul>
<i>„Schriftart“</i>	<p>Auswahl aus vordefinierten Schriftarten:</p> <ul style="list-style-type: none"> <li>▪ Standard</li> <li>▪ Überschrift</li> <li>▪ Große</li> <li>▪ Titel</li> <li>▪ Anmerkung</li> </ul> <p>Mit Klick auf <input type="button" value="..."/> öffnet sich der Dialog für benutzerdefinierte Eigenschaften der Schriftart.</p>
<i>„Farbe Schriftart“</i>	<p>Auszuwählen ist eine Schriftartfarbe für das Element. Entweder aus der Auswahlliste oder durch Setzen im sich öffnenden Dialog, wenn <input type="button" value="..."/> geklickt wird.</p>

## Textvariablen

Dies sind dynamische Definitionen. Siehe auch *„Dynamische Texte“* bei Verwendung von Textlisten. Für statische Definitionen siehe *„Texte“*

Eigenschaft	Beschreibung Wert
<i>„Textvariable“</i>	Die zugewiesene Variable definiert den Elementtext ("Textausgabevariable"). Unter <a href="#">„Text und Sprache in Visualisierungen“</a> ist zu finden, wie ein Text zu formatieren ist.
<i>„Tooltipvariable“</i>	Die zugewiesene STRING-Variable enthält den Tooltip-Text für das Element.

## Farbvariablen

Angewandt for **dynamische** Definitionen der Elementfarbe mittels einer Projektvariablen.

Eine Farbe wird mittels Hexzahl definiert, die sich aus Rot/Grün/Blau (RGB) Anteilen zusammensetzt. Es ist eine Projektvariable Typ DWORD anzugeben, deren Wert die Farbe definiert. (zum Beispiel `progl.dwFillColor`).

Die Variable enthält den Farbwert in hexadezimalen Format. Um das gesamte DWORD auszufüllen, müssen die ersten beiden Ziffern nach "16#" mit 0 gesetzt werden: `16#00RRGGBB`

### Farbwerte

```
dwFillColor := 16#008FE03F;
```

Hexzahl:

- 8F: rot
- E0: grün
- 3F: blau

Eigenschaft	Beschreibung Wert
„ <i>Farbumschlag</i> “	Hier ist eine boolesche Variable angeben, die das Umschalten der Elementfarbe zwischen „ <i>Normalzustand</i> “ (Variable = FALSE) und „ <i>Alarmzustand</i> “ (Variable = TRUE) steuert.
„ <i>Normalzustand</i> “	Eingabe von DWORD-Variablen, die die Rahmen- und Füllfarbe des Elements bestimmen. Sie überschreiben die Werte, die aktuell in <i>Farben</i> → <i>Normalzustand</i> definiert sind. Die Werte in den Projektvariablen werden verwendet, wenn die Variable definiert in „ <i>Farbumschlag</i> “ FALSE ist.
▪ „ <i>Rahmenfarbe</i> “	
▪ „ <i>Füllfarbe</i> “	
„ <i>Alarmzustand</i> “	Eingabe von DWORD-Variablen, die die Rahmen- und Füllfarbe des Elements im Alarmzustand bestimmen. Sie überschreiben die Werte, die aktuell in <i>Farben</i> → <i>Alarmzustand</i> definiert sind. Die Werte in den Projektvariablen werden verwendet, wenn die Variable definiert in „ <i>Farbumschlag</i> “ TRUE ist.
▪ „ <i>Rahmenfarbe</i> “	
▪ „ <i>Füllfarbe</i> “	

### Zustandsvariablen

Eigenschaft	Beschreibung Wert
„ <i>Unsichtbarkeit</i> “	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.
„ <i>Eingaben deaktivieren</i> “	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann hat keine der Elementeingaben eine Auswirkung.

## Eingabekonfiguration

Hier kann definiert werden, welche Folgeaktionen ausgeführt werden, wenn der Benutzer im Online-Modus eine Eingabe im Element vornimmt.

Eingabeereignisse, die ein Element haben kann, sind:

- *„OnDialogClosed“*
- *„OnClick“*
- *„MouseDown“*
- *„OnMouseEnter“*
- *„OnMouseLeave“*
- *„MouseMove“*
- *„MouseUp“*

Mit Klick auf „*Konfigurieren...*“ wird der Dialog [„Eingabekonfiguration“](#) geöffnet, der erlaubt Folgeaktionen zuzuweisen. Man klickt in das Feld neben der zu konfigurierenden Eingabe. Für jede selektierte Folgeaktion wird ein Eintrag in der Eigenschaftentabelle erzeugt. Jede Eingabe kann eine beliebige zugeordnete Anzahl an Folgeaktionen haben.

Eigenschaft	Beschreibung Wert
<i>„OnDialogClosed“</i>	Dieses Ereignis tritt ein, wenn innerhalb der Visualisierung eines der für Benutzereingaben vorher geöffneten Dialogfenster geschlossen wird.
<i>„OnClick“</i>	Dieses Mausereignis tritt ein, wenn der Cursor auf ein Element zeigt und ein vollständiger Mausklick (drücken und freigeben der Maustaste) ausgeführt wird.
<i>„MouseDown“</i>	Dieses Mausereignis tritt ein, wenn die Maustaste gedrückt wird während der Cursor auf ein Element zeigt. (Drücken der Maustaste)
<i>„OnMouseEnter“</i>	Dieses Mausereignis tritt ein, wenn der Cursor auf das Element gezogen wird.
<i>„OnMouseLeave“</i>	Dieses Mausereignis tritt ein, wenn der Cursor das Element verlässt.
<i>„MouseMove“</i>	Dieses Mausereignis tritt ein, wenn der Cursor innerhalb des Elements bewegt wird.
<i>„MouseUp“</i>	Diese Mausereignis liegt vor, wenn die Maustaste losgelassen wird. Das Drücken der Maustaste geschah vorher, außerhalb des Elements.

### Eingabekonfiguration: Tasten

Mit „*Tasten*“ kann konfiguriert werden, dass beim Ereignis "Tasten" der Wert einer boolschen Projektvariable abhängig vom Mausverhalten gesetzt wird.

Eigenschaft	Beschreibung Wert
<i>„Variable“</i>	Hier kann eine boolsche Variable zugewiesen werden, deren Wert TRUE wird, wenn die




Maustaste gedrückt wird, während der Cursor auf das Element zeigt und die wieder FALSE wird, wenn die Maustaste wieder losgelassen wird.

„*FALSE tasten*“ Wenn diese Option aktiviert ist, wird das oben beschriebene Tastverhalten für die eingetragene Variable umgekehrt. Das heißt, wenn die Maustaste gedrückt wird, wird der Variablenwert auf FALSE gesetzt, wenn die Taste wieder losgelassen wird, wird der Variablenwert auf TRUE gesetzt.

„*Beim Betreten tasten, falls Maus gefangen*“ Ist diese Option aktiviert, verhält sich der Variablenwert wie unter „*Variable*“ beschrieben, solange sich die Maus innerhalb des Elementbereichs befindet. Zusätzlich wird auf ein Ziehen der Maus reagiert. Wird bei gedrückter Maustaste der Elementbereich verlassen, wird die Variable FALSE und wird wieder TRUE, wenn in den Elementbereich zurückgekehrt wird. Es wird also berücksichtigt, dass, solange die Maustaste gedrückt ist, die Maus "gefangen" ist, auch wenn der Elementbereich verlassen wird.

### Eingabekonfiguration: Umschalten

Mit „*Umschalten*“ kann konfiguriert werden, dass bei diesem Ereignis der Wert einer booleschen Projektvariable abhängig vom Mausverhalten gesetzt wird.

Eigenschaft	Beschreibung Wert
„ <i>Variable</i> “	<p>Hier ist eine boolesche Variable zuzuweisen. Die Eingabehilfe kann über Schaltfläche  geöffnet werden.</p> <p>Der Wert der Variablen wird bei jedem Mausklick auf das Element zwischen TRUE und FALSE umgeschaltet. Wird der Cursor bei gedrückter Maustaste aus dem Element herausbewegt, dann erfolgt kein Toggeln. Damit kann die begonnene Umschalteingabe abgebrochen werden.</p>
„ <i>Umschalten beim Loslassen falls Maus gefangen</i> “	<p>Ist diese Option aktiviert, verhält sich der Variablenwert wie unter „<i>Variable</i>“ beschrieben, solange sich die Maus innerhalb des Elementbereichs befindet. Zusätzlich wird auf ein Ziehen der Maus reagiert. Wird bei gedrückter Maustaste der Elementbereich verlassen und dann die Maustaste losgelassen, wird die Variable umgeschaltet. Es wird also berücksichtigt, dass, solange die Maustaste gedrückt ist, die Maus "gefangen" ist, auch wenn der Elementbereich verlassen wird.</p>

### Eingabekonfiguration: Hotkey

Mit „*Hotkey*“ kann eine Taste, auch mit Zusatzstufe, definiert und mit einer bestimmten Nachfolgeaktion (MouseDown, MouseUp) verknüpft werden, die bei einem Ereignis der Taste (KeyDown, KeyUp) ausgeführt werden soll. Standardmäßig wird bei KeyDown (Taste drücken) die MouseDown-Aktion ausgeführt und bei KeyUp (Taste loslassen) die MouseUp-Aktion. Das kann nützlich sein, wenn eine Visualisierung sowohl über Maus- als auch Tastatureingaben bedient werden soll, weil die Eingabeaktionen dann nur einmalig konfiguriert werden müssen. Diese Tastenkonfiguration für ein Element wird auch im [„Tastaturkonfiguration“](#) der Visualisierung verwaltet.

Änderungen werden immer zwischen diesem und dem Elementeigenschaften-Editor synchronisiert.

Eigenschaft	Beschreibung Wert
„Taste“	Zuweisen einer Taste. Eine Auswahlliste stellt alle aktuell unterstützten Tasten bereit wie zum Beispiel [ M ]. <a href="#">↘ „Tastaturkonfiguration“</a>
„Eingabe(n)“	Definieren des Ereignisses, das ausgeführt werden soll, wenn die Taste beziehungsweise Taste und Modifizierer verwendet werden.  Mögliche Werte, die in einer Auswahlliste bereitstehen: <ul style="list-style-type: none"><li>Keine Aktion</li><li>MouseDown-Aktion: OnMouseDown-Aktion, wenn die Taste gedrückt wird.</li><li>MouseUp-Aktion: OnMouseUp-Aktion, wenn die Taste freigelassen wird.</li><li>MouseDown/MouseUp-Aktion: OnMouseUp- und OnMouseDown-Aktion, wenn die Taste gedrückt oder losgelassen wird.</li></ul>
„Umschalten“	Wenn diese Option aktiviert wird, muss die Taste mit [ Umschalt ] kombiniert verwendet werden
„Steuerung“	Wenn diese Option aktiviert wird, muss die Taste mit [ Strg ] kombiniert verwendet werden
„Alt“	Wenn diese Option aktiviert wird, muss die Taste mit [ Alt ] kombiniert verwendet werden

Eigenschaft	Beschreibung Wert
„Zugriffsrechte“	Mit Klick öffnet <a href="#">↘ „Zugriffsrechte“</a> .  Statusmeldungen: <ul style="list-style-type: none"><li>Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.</li><li>Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.</li></ul>

## Scrollbalken

Diese Element erzeugt eine Bildlaufleiste, deren Minimal- und Maximalwert definiert werden kann. Die Position des Reglers in der Bildlaufleiste folgt dem Wert der Variablen definiert in „Wert“.



Abb. 67: Scrollbalken, Stil 8

### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungsektor werden folgende Benutzereingaben zur Verfügung gestellt:

Element spezifische Mauseingabe:

- Standardmäßig wird der Scrollbar horizontal angezeigt. Ein eingefügtes Element kann mit Hilfe der Maus in eine andere Form gezogen werden. Sobald die Höhe die Breite übersteigt, wird die Bildlaufleiste vertikal angezeigt.

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„Elementname“	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„Elementtyp“	Scrollbalken
„Wert“	Hier ist eine Variable zuzuweisen, deren Wert die aktuelle Position des Reglers enthält.
„Minimaler Wert“	Definiert den kleinsten Wert des Reglerbereichs.
„Maximaler Wert“	Definiert den größten Wert des Reglerbereichs.
„Seitengröße“	Wird in den Scrollbereich geklickt, wird die Position des Reglers in Richtung der Mausposition verschoben. Um wieviel verschoben wird, ist hier anzugeben.

#### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„X“	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„Y“	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„Breite“	Horizontale Breite des Elements.
„Höhe“	Vertikale Höhe des Elements.

## Balken

Eigenschaft	Beschreibung Wert
„Ausrichtung“	Der Balken kann <code>Horizontal</code> und <code>Vertikal</code> verlaufen. Dies resultiert aus dem Verhältnis von Breite zu Höhe und kann hier nicht editiert werden. Im Visualisierungseditor wird durch drag-and-drop die Ausrichtung geändert: <a href="#">↘</a>
„Laufrichtung“	Ist die Ausrichtung horizontal, kann gewählt werden zwischen: <ul style="list-style-type: none"> <li>▪ <code>Links nach rechts</code></li> <li>▪ <code>Rechts nach Links</code></li> </ul> <p>Ist die Ausrichtung vertikal, kann gewählt werden zwischen:</p> <ul style="list-style-type: none"> <li>▪ <code>Von unten nach oben</code></li> <li>▪ <code>Von oben nach unten</code></li> </ul>

## Farbvariablen

Eine Farbe wird mittels Hexzahl definiert, die sich aus Rot/Grün/Blau (RGB) Anteilen zusammensetzt. Für jede dieser drei Farben sind 256 (0-255) Werte verfügbar. Eine Projektvariable Typ `DWORD` muss angegeben werden, deren Wert die Farbe definiert (zum Beispiel `prog1.dwFillColor`).

Die Variable enthält den Farbwert in hexadezimalen Format. Um das gesamte `DWORD` auszufüllen, müssen die ersten beiden Ziffern nach "16#" mit 0 gesetzt werden: `16#00RRGGBB`

### Farbwerte

```
dwFillColor := 16#008FE03F;
```

Hexzahl:

- `8F`: rot
- `E0`: grün
- `3F`: blau

Eigenschaft	Beschreibung Wert
-------------	-------------------

- „*Farbumschlag*“ Hier ist eine boolesche Variabel angeben, die das Umschalten der Elementfarbe zwischen „*Farbe*“ (Variable = FALSE) und „*Alarmfarbe*“ (Variable = TRUE) steuert.
- „*Farbe*“ Eingabe einer DWORD-Variable, die die Farbe des Elements bestimmt. Sie überschreibt den Wert, der aktuell in *Farben* → *Farbe* definiert ist, wenn die Variable in „*Farbumschlag*“ FALSE ist.
- „*Alarmfarbe*“ Eingabe einer DWORD-Variable, die die Alarmfarbe des Elements bestimmt. Sie überschreibt den Wert, der aktuell in *Farben* → *Alarmfarbe* definiert ist, wenn die Variable in „*Farbumschlag*“ TRUE ist.

## Texte

Diese Eigenschaften werden für eine statische Definition der Elementbeschriftung verwendet.


Property	Value description
„ <i>Text</i> “	Der angegebene Text beschriftet das Element. Ein textformatierender String ist möglich.
„ <i>Tooltip</i> “	Der angegebenen Text erscheint als Tooltip des Elements, wenn die Visualisierung im Online-Modus ist.

## Texteigenschaften

Diese Eigenschaften werden für eine statische Definition der Schriftart verwendet.

Eigenschaft	Beschreibung Wert
„ <i>Horizontale Ausrichtung</i> “	Definieren der horizontalen Ausrichtung des Textes durch Auswahl aus: <ul style="list-style-type: none"> <li>▪ Links</li> <li>▪ Zentriert</li> <li>▪ Rechts</li> </ul>
„ <i>Vertikale Ausrichtung</i> “	Definieren der vertikalen Ausrichtung des Textes durch Auswahl aus: <ul style="list-style-type: none"> <li>▪ Oben</li> <li>▪ Zentriert</li> <li>▪ Unten</li> </ul>
„ <i>Schriftart</i> “	Auswahl aus vordefinierten Schriftarten: <ul style="list-style-type: none"> <li>▪ Standard</li> <li>▪ Überschrift</li> <li>▪ Große</li> <li>▪ Titel</li> <li>▪ Anmerkung</li> </ul>

Mit Klick auf  öffnet sich der Dialog für benutzerdefinierte Eigenschaften der Schriftart.

„Farbe Schriftart“ Auszuwählen ist eine Schriftartfarbe für das Element. Entweder aus der Auswahlliste oder durch Setzen im sich öffnenden Dialog, wenn  geklickt wird.

## Textvariablen

Dies sind dynamische Definitionen. Siehe auch „*Dynamische Texte*“ bei Verwendung von Textlisten. Für statische Definitionen siehe „*Texte*“

Eigenschaft	Beschreibung Wert
„ <i>Textvariable</i> “	Die zugewiesene Variable definiert den Elementtext ("Textausgabevariable"). Unter <a href="#">„Text und Sprache in Visualisierungen“</a> ist zu finden, wie ein Text zu formatieren ist.
„ <i>Tooltipvariable</i> “	Die zugewiesene STRING-Variable enthält den Tooltip-Text für das Element.

## Dynamische Texte

Diese Parameter dienen der Definition von dynamischen Texten, die aus Textlisten stammen (dies erlaubt Sprachumschaltung). Siehe [Textliste](#) für nähere Informationen zur Verwendung von Textlisten.

Eigenschaft	Beschreibung Wert
„ <i>Textliste</i> “	Name der Textliste (String) wie im POU- oder Geräte-Fenster verwendet; z.B. <code>errorlist</code>
„ <i>Textindex</i> “	Index (ID) des Textes wie in der Textliste definiert (String); Direkte Angabe der ID, z.B. 'M1' oder String-Variable, z.B. <code>error_22</code> .
„ <i>Tooltipindex</i> “	Index (ID) des Tooltip-Textes wie in der Textliste definiert (String); Direkte Angabe der ID, z.B. 'T1' oder String-Variable, z.B. <code>error_Tooltip_22</code> .

## Schriftartvariablen

Diese werden verwendet bei dynamischen Schriftdefinitionen des Elementtextes über Projektvariablen. Statischen Definitionen werden unter [Texteigenschaften](#) → [Schriftart](#) konfiguriert.

Eigenschaft	Beschreibung Wert
„ <i>Schriftname</i> “	Hier ist eine Variable Typ STRING anzugeben, die den Schriftnamen enthält, der für die Beschriftung des Elements verwendet werden soll. (Namensangabe wie im Standard-Schrifttypen-Dialog). Beispiel: <code>plc_prg.font_var font_var := Arial;</code>
„ <i>Größe</i> “	Eingabe einer Variablen Typ INT, die die Größe des Elementtextes in Pixel enthält, so wie im Standarddialog Font. Beispiel: <code>prog1.font_height font_height:=16;</code>
„ <i>Flags</i> “	Eingabe einer Variable Typ DWORD, die über die unten genannten Flag-Werte die Font-Darstellung festlegt. Eine kombinierte Definition kann erreicht werden, indem die jeweiligen Flag-Werte addiert werden und die Summe angegeben wird.

- 1: kursiv

- 2: fett
- 4: unterstrichen
- 8: gelöscht

`prog2.font_type: ist font_type:=6`, wird der Text fett und unterstrichen dargestellt.

„*Zeichensatz*“ Der Zeichensatz, der für den Font verwendet werden soll, kann über die Standard-Zeichensatz-Nummer definiert werden. Mit einer DWORD-Variablen kann diese Nummer angegeben werden (siehe auch die „*Skript*“-Definition im Standard-Font-Dialog)

„*Farbe*“ Eingabe einer DWORD-Variablen, in der die Farbe des Elementtextes festgelegt wird.

#### Zustandsvariablen

Eigenschaft	Beschreibung Wert
„ <i>Unsichtbarkeit</i> “	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.
„ <i>Eingaben deaktivieren</i> “	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann hat keine der Elementeingaben eine Auswirkung.

Eigenschaft	Beschreibung Wert
-------------	-------------------

„*Zugriffsrechte*“ Mit Klick öffnet [„Zugriffsrechte“](#).

Statusmeldungen:

- Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.
- Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.

#### Schieberegler

Der Balken des Schiebereglers kann mit Hilfe der Maus verschoben werden, wodurch eine dem Schieberegler zugewiesene Variable ihren Wert innerhalb der definierten Grenzen verändert.

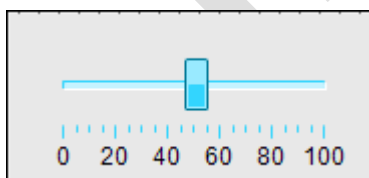


Abb. 68: Schieberegler, Stil 8

#### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungseditor werden folgende Benutzereingaben zur Verfügung gestellt:

Elementspezifische Mauseingabe:

- Standard ist, dass der Schieberegler horizontal dargestellt wird. Wird ein Schieberegler eingefügt, kann er mit Hilfe der Maus in eine andere Form gezogen werden. Sobald seine Höhe die Breite übersteigt, wird der Schieberegler vertikal dargestellt und die zugehörige Eigenschaft *Balken* → *Ausrichtung* wird aktualisiert auf Vertikal.

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Schieberegler

#### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„ <i>X</i> “	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„ <i>Y</i> “	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„ <i>Breite</i> “	Horizontale Breite des Elements.
„ <i>Höhe</i> “	Vertikale Höhe des Elements.
„ <i>Variable</i> “	Hier ist eine numerische Variable anzugeben, die die Position des Reglers enthalten wird.

#### Skala

Eigenschaft	Beschreibung Wert
„ <i>Skala anzeigen</i> “	Standardmäßig ist diese Checkbox aktiviert. Deaktivieren bewirkt ein Ausblenden der Skala.
„ <i>Skalenanfang</i> “	Wert, der den Skalenbereich von unten begrenzt.
„ <i>Skalende</i> “	Wert, der den Skalenbereich von oben begrenzt.
„ <i>Hauptskala</i> “	Abstand zwischen zwei Skalenstrichen der Grobskala.



- „*Unterskala*“                    Abstand zwischen zwei Strichen der Feinskala, die auf 0 gesetzt werden kann, falls eine weitere Unterteilung der Grobskala nicht gewünscht ist.
- „*Skalenformat (C-syntax)*“            Wird ein Skalenformat wie zum Beispiel "%d s" konfiguriert, dann wird zusätzlich eine Skalenbeschriftung angezeigt.

**Balken**

Eigenschaft	Beschreibung Wert
„ <i>Diagrammart</i> “	<p>Hier kann abhängig von der Ausrichtung des Diagramms eine Diagrammart gewählt werden.</p> <p>Ist die Ausrichtung veritkal, ist zwischen folgendem zu wählen:</p> <ul style="list-style-type: none"> <li>▪ Links: wenn sich der Regler links der Skala befinden soll.</li> <li>▪ Rechts: wenn sich der Regler rechts der Skala befinden soll.</li> <li>▪ Links und rechts: wenn sich der Regler zwischen zwei Skalen befinden soll.</li> </ul> <p>Ist die Ausrichtung horizontal, ist zwischen folgendem zu wählen:</p> <ul style="list-style-type: none"> <li>▪ Oben: wenn sich der Regler über der Skala befinden soll.</li> <li>▪ Unten: wenn sich der Regler unter der Skala befinden soll.</li> <li>▪ Oben und unten: wenn sich der Regler zwischen zwei Skalen befinden soll.</li> </ul>
„ <i>Ausrichtung</i> “	<p>Der Regler kann in „<i>horizontaler</i>“ oder „<i>verikaler</i>“ Ausrichtung verlaufen. Die Ausrichtung ist das Ergebnis des Verhältnisses von Breite zu Höhe und kann in „<i>Eigenschaften</i>“ nicht editiert werden, sondern wird im Visualisierungseditor mit Drag-and-drop editiert.</p>
„ <i>Laufrichtung</i> “	<p>Ist die Ausrichtung horizontal, ist zwischen folgendem zu wählen:</p> <ul style="list-style-type: none"> <li>▪ Links nach rechts: „<i>Skalenanfang</i>“ ist links.</li> <li>▪ Rechts nach links: „<i>Skalenanfang</i>“ ist rechts.</li> </ul> <p>Ist die Ausrichtung vertikal, ist zwischen folgendem zu wählen:</p> <ul style="list-style-type: none"> <li>▪ Unten nach oben „<i>Skalenanfang</i>“ ist unten.</li> <li>▪ Oben nach unten: „<i>Skalenanfang</i>“ ist oben.</li> </ul>

**Zustandsvariablen**

Eigenschaft	Beschreibung Wert
„ <i>Unsichtbarkeit</i> “	<p>Eingabe einer boolschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.</p>
„ <i>Eingaben deaktivieren</i> “	<p>Eingabe einer boolschen Variable; ist der Variablenwert TRUE, dann hat keine der Elementeingaben eine Auswirkung.</p>

Eigenschaft	Beschreibung Wert
-------------	-------------------

„Zugriffsrechte“ Mit Klick öffnet [↘ „Zugriffsrechte“](#).

Statusmeldungen:

- Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.
- Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.

## SpinControl

SpinControl erlaubt es, den Wert einer Variablen per Mausklick auf die kleinen Pfeiltasten an der rechten Seite des Elements zu inkrementieren oder zu dekrementieren.



Abb. 69: SpinControl, Stil 8

### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungsektor werden folgende Benutzereingaben zur Verfügung gestellt:

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Spin-Kontrolle

#### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„ <i>X</i> “	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„ <i>Y</i> “	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„ <i>Breite</i> “	Horizontale Breite des Elements.
„ <i>Höhe</i> “	Vertikale Höhe des Elements.
„ <i>Variable</i> “	Hier ist eine numerische Variable anzugeben, deren Wert nach einer Benutzereingabe ansteigt oder abnimmt.
„ <i>Zahlenformat</i> “	Eingabe einer Formatierungssequenz mit der der Variablenwert angezeigt wird wie zum Beispiel %i für eine Integer-Variable.
„ <i>Intervall</i> “	Definition des Intervalls (Schrittweite), um den der Variablenwert nach einer Benutzereingabe

inkrementiert oder dekrementiert wird.

## Farbvariablen

Eigenschaft	Beschreibung Wert
<i>„Farbumschlag“</i>	Angabe einer booleschen Variablen, die die Farbe des Elements umschaltet. Element- und Alarmfarbe können im Visualisierungsstil definiert werden.

## Zustandsvariablen

Eigenschaft	Beschreibung Wert
<i>„Unsichtbarkeit“</i>	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.
<i>„Eingaben deaktivieren“</i>	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann hat keine der Elementeingaben eine Auswirkung.

## Eingabekonfiguration

Hier kann definiert werden, welche Folgeaktionen ausgeführt werden, wenn der Benutzer im Online-Modus eine Eingabe im Element vornimmt.

Eingabeereignisse, die ein Element haben kann, sind:

- *„OnDialogClosed“*
- *„OnMouseClicked“*
- *„OnMouseDown“*
- *„OnMouseEnter“*
- *„OnMouseLeave“*
- *„OnMouseMove“*
- *„OnMouseUp“*

Mit Klick auf *„Konfigurieren...“* wird der Dialog [„Eingabekonfiguration“](#) geöffnet, der erlaubt Folgeaktionen zuzuweisen. Man klickt in das Feld neben der zu konfigurierenden Eingabe. Für jede selektierte Folgeaktion wird ein Eintrag in der Eigenschaftentabelle erzeugt. Jede Eingabe kann eine beliebige zugeordnete Anzahl an Folgeaktionen haben.

Eigenschaft	Beschreibung Wert
<i>„OnDialogClosed“</i>	Dieses Ereignis tritt ein, wenn innerhalb der Visualisierung eines der für Benutzereingaben vorher geöffneten Dialogfenster geschlossen wird.
<i>„OnMouseClicked“</i>	Dieses Mausereignis tritt ein, wenn der Cursor auf ein Element zeigt und ein vollständiger Mausklick (drücken und freigeben der Maustaste) ausgeführt wird.
<i>„OnMouseDown“</i>	Dieses Mausereignis tritt ein, wenn die Maustaste gedrückt wird während der Cursor auf ein Element zeigt. (Drücken der Maustaste)

- „*OnMouseEnter*“ Dieses Mausereignis tritt ein, wenn der Cursor auf das Element gezogen wird.
- „*OnMouseLeave*“ Dieses Mausereignis tritt ein, wenn der Cursor das Element verlässt.
- „*OnMouseMove*“ Dieses Mausereignis tritt ein, wenn der Cursor innerhalb des Elements bewegt wird.
- „*OnMouseUp*“ Diese Mausereignis liegt vor, wenn die Maustaste losgelassen wird. Das Drücken der Maustaste geschah vorher, außerhalb des Elements.

### Eingabekonfiguration: Hotkey

Mit „*Hotkey*“ kann eine Taste, auch mit Zusatz Taste, definiert und mit einer bestimmten Nachfolgeaktion (MouseDown, MouseUp) verknüpft werden, die bei einem Ereignis der Taste (KeyDown, KeyUp) ausgeführt werden soll. Standardmäßig wird bei KeyDown (Taste drücken) die MouseDown-Aktion ausgeführt und bei KeyUp (Taste loslassen) die MouseUp-Aktion. Das kann nützlich sein, wenn eine Visualisierung sowohl über Maus- als auch Tastatureingaben bedient werden soll, weil die Eingabeaktionen dann nur einmalig konfiguriert werden müssen. Diese Tastenkonfiguration für ein Element wird auch im [„Tastaturkonfiguration“](#) der Visualisierung verwaltet. Änderungen werden immer zwischen diesem und dem Elementeigenschaften-Editor synchronisiert.

Eigenschaft	Beschreibung Wert
„ <i>Taste</i> “	Zuweisen einer Taste. Eine Auswahlliste stellt alle aktuell unterstützten Tasten bereit wie zum Beispiel [ M ]. <a href="#">„Tastaturkonfiguration“</a>
„ <i>Eingabe(n)</i> “	Definieren des Ereignisses, das ausgeführt werden soll, wenn die Taste beziehungsweise Taste und Modifizierer verwendet werden.  Mögliche Werte, die in einer Auswahlliste bereitstehen: <ul style="list-style-type: none"> <li>▪ Keine Aktion</li> <li>▪ MouseDown-Aktion: OnMouseDown-Aktion, wenn die Taste gedrückt wird.</li> <li>▪ MouseUp-Aktion: OnMouseUp-Aktion, wenn die Taste freigelassen wird.</li> <li>▪ MouseDown/MouseUp-Aktion: OnMouseUp- und OnMouseDown-Aktion, wenn die Taste gedrückt oder losgelassen wird.</li> </ul>
„ <i>Umschalten</i> “	Wenn diese Option aktiviert wird, muss die Taste mit [ <i>Umschalt</i> ] kombiniert verwendet werden
„ <i>Steuerung</i> “	Wenn diese Option aktiviert wird, muss die Taste mit [ <i>Strg</i> ] kombiniert verwendet werden
„ <i>Alt</i> “	Wenn diese Option aktiviert wird, muss die Taste mit [ <i>Alt</i> ] kombiniert verwendet werden

Eigenschaft	Beschreibung Wert
„ <i>Zugriffsrechte</i> “	Mit Klick öffnet <a href="#">„Zugriffsrechte“</a> .  Statusmeldungen: <ul style="list-style-type: none"> <li>▪ Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.</li> </ul>

- Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.

in Arbeit

## Unsichtbare Eingabe

Das Element wird im Editor mit einer gestrichelten Linie angezeigt, aber im Online-Modus ist diese unsichtbar. Das Verhalten des Elements ist in der Eingabekonfiguration zu spezifizieren.

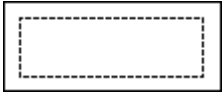


Abb. 70: Unsichtbare Eingabe

### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungeditor werden folgende Benutzereingaben zur Verfügung gestellt:

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemlnst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Unsichtbare Eingabe

#### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„ <i>X</i> “	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„ <i>Y</i> “	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„ <i>Breite</i> “	Horizontale Breite des Elements.
„ <i>Höhe</i> “	Vertikale Höhe des Elements.

#### Zustandsvariablen

Eigenschaft	Beschreibung Wert
„ <i>Eingaben deaktivieren</i> “	Zuzuweisen ist eine boolsche Variable. Ist deren Wert TRUE, dann hat keine Benutzereingabe auf das Element eine Auswirkung.

#### Eingabekonfiguration

Hier kann definiert werden, welche Folgeaktionen ausgeführt werden, wenn der Benutzer im Online-Modus eine Eingabe im Element vornimmt.

Eingabeereignisse, die ein Element haben kann, sind:

- „OnDialogClosed“
- „OnClick“
- „MouseDown“
- „OnMouseEnter“
- „OnMouseLeave“
- „MouseMove“
- „MouseUp“

Mit Klick auf „Konfigurieren...“ wird der Dialog [„Eingabekonfiguration“](#) geöffnet, der erlaubt Folgeaktionen zuzuweisen. Man klickt in das Feld neben der zu konfigurierenden Eingabe. Für jede selektierte Folgeaktion wird ein Eintrag in der Eigenschaftentabelle erzeugt. Jede Eingabe kann eine beliebige zugeordnete Anzahl an Folgeaktionen haben.

Eigenschaft	Beschreibung Wert
„OnDialogClosed“	Dieses Ereignis tritt ein, wenn innerhalb der Visualisierung eines der für Benutzereingaben vorher geöffneten Dialogfenster geschlossen wird.
„OnClick“	Dieses Mausereignis tritt ein, wenn der Cursor auf ein Element zeigt und ein vollständiger Mausklick (drücken und freigeben der Maustaste) ausgeführt wird.
„MouseDown“	Dieses Mausereignis tritt ein, wenn die Maustaste gedrückt wird während der Cursor auf ein Element zeigt. (Drücken der Maustaste)
„OnMouseEnter“	Dieses Mausereignis tritt ein, wenn der Cursor auf das Element gezogen wird.
„OnMouseLeave“	Dieses Mausereignis tritt ein, wenn der Cursor das Element verlässt.
„MouseMove“	Dieses Mausereignis tritt ein, wenn der Cursor innerhalb des Elements bewegt wird.
„MouseUp“	Diese Mausereignis liegt vor, wenn die Maustaste losgelassen wird. Das Drücken der Maustaste geschah vorher, außerhalb des Elements.

#### Eingabekonfiguration: Tasten

Mit „Tasten“ kann konfiguriert werden, dass beim Ereignis "Tasten" der Wert einer booleschen Projektvariable abhängig vom Mausverhalten gesetzt wird.

Eigenschaft	Beschreibung Wert
„Variable“	Hier kann eine boolesche Variable zugewiesen werden, deren Wert TRUE wird, wenn die Maustaste gedrückt wird, während der Cursor auf das Element zeigt und die wieder FALSE



wird, wenn die Maustaste wieder losgelassen wird.

#### „FALSE tasten“


Wenn diese Option aktiviert ist, wird das oben beschriebene Tastverhalten für die eingetragene Variable umgekehrt. Das heißt, wenn die Maustaste gedrückt wird, wird der Variablenwert auf FALSE gesetzt, wenn die Taste wieder losgelassen wird, wird der Variablenwert auf TRUE gesetzt.

#### „Beim Betreten tasten, falls Maus gefangen“

Ist diese Option aktiviert, verhält sich der Variablenwert wie unter „Variable“ beschrieben, solange sich die Maus innerhalb des Elementbereichs befindet. Zusätzlich wird auf ein Ziehen der Maus reagiert. Wird bei gedrückter Maustaste der Elementbereich verlassen, wird die Variable FALSE und wird wieder TRUE, wenn in den Elementbereich zurückgekehrt wird. Es wird also berücksichtigt, dass, solange die Maustaste gedrückt ist, die Maus "gefangen" ist, auch wenn der Elementbereich verlassen wird.

### Eingabekonfiguration: Umschalten

Mit „Umschalten“ kann konfiguriert werden, dass bei diesem Ereignis der Wert einer booleschen Projektvariable abhängig vom Mausverhalten gesetzt wird.

Eigenschaft	Beschreibung Wert
„Variable“	<p>Hier ist eine boolesche Variable zuzuweisen. Die Eingabehilfe kann über Schaltfläche  geöffnet werden.</p> <p>Der Wert der Variablen wird bei jedem Mausklick auf das Element zwischen TRUE und FALSE umgeschaltet. Wird der Cursor bei gedrückter Maustaste aus dem Element herausbewegt, dann erfolgt kein Toggeln. Damit kann die begonnene Umschalteingabe abgebrochen werden.</p>
„Umschalten beim Loslassen falls Maus gefangen“	<p>Ist diese Option aktiviert, verhält sich der Variablenwert wie unter „Variable“ beschrieben, solange sich die Maus innerhalb des Elementbereichs befindet. Zusätzlich wird auf ein Ziehen der Maus reagiert. Wird bei gedrückter Maustaste der Elementbereich verlassen und dann die Maustaste losgelassen, wird die Variable umgeschaltet. Es wird also berücksichtigt, dass, solange die Maustaste gedrückt ist, die Maus "gefangen" ist, auch wenn der Elementbereich verlassen wird.</p>

### Eingabekonfiguration: Hotkey

Mit „Hotkey“ kann eine Taste, auch mit Zusatzaste, definiert und mit einer bestimmten Nachfolgeaktion (MouseDown, MouseUp) verknüpft werden, die bei einem Ereignis der Taste (KeyDown, KeyUp) ausgeführt werden soll. Standardmäßig wird bei KeyDown (Taste drücken) die MouseDown-Aktion ausgeführt und bei KeyUp (Taste loslassen) die MouseUp-Aktion. Das kann nützlich sein, wenn eine Visualisierung sowohl über Maus- als auch Tastatureingaben bedient werden soll, weil die Eingabeaktionen dann nur einmalig konfiguriert werden müssen. Diese Tastenkonfiguration für ein Element wird auch im [„Tastaturkonfiguration“](#) der Visualisierung verwaltet. Änderungen werden immer zwischen diesem und dem Elementeigenschaften-Editor synchronisiert.

Eigenschaft	Beschreibung Wert
„Taste“	Zuweisen einer Taste. Eine Auswahlliste stellt alle aktuell unterstützten Tasten bereit wie zum Beispiel [ M ]. <a href="#">↘ „Tastaturkonfiguration“</a>
„Eingabe(n)“	Definieren des Ereignisses, das ausgeführt werden soll, wenn die Taste beziehungsweise Taste und Modifizierer verwendet werden.  Mögliche Werte, die in einer Auswahlliste bereitstehen: <ul style="list-style-type: none"> <li>Keine Aktion</li> <li>MouseDown-Aktion: OnMouseDown-Aktion, wenn die Taste gedrückt wird.</li> <li>MouseUp-Aktion: OnMouseUp-Aktion, wenn die Taste freigelassen wird.</li> <li>MouseDown/MouseUp-Aktion: OnMouseUp- und OnMouseDown-Aktion, wenn die Taste gedrückt oder losgelassen wird.</li> </ul>
„Umschalten“	Wenn diese Option aktiviert wird, muss die Taste mit [ Umschalt ] kombiniert verwendet werden
„Steuerung“	Wenn diese Option aktiviert wird, muss die Taste mit [ Strg ] kombiniert verwendet werden
„Alt“	Wenn diese Option aktiviert wird, muss die Taste mit [ Alt ] kombiniert verwendet werden

Eigenschaft	Beschreibung Wert
„Zugriffsrechte“	Mit Klick öffnet <a href="#">↘ „Zugriffsrechte“</a> .  Statusmeldungen: <ul style="list-style-type: none"> <li>Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.</li> <li>Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.</li> </ul>

## Ladebalken

Ein Ladebalken stellt den Fortschritt von etwas optisch dar. Dies erfordert eine Variable, deren Wert als Balken angezeigt wird. Die Grenzen der Variable sowie der Anzeigestil kann eingestellt werden.



Abb. 71: Ladebalken

### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungsektor werden folgende Benutzereingaben zur Verfügung gestellt:

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Ladebalken
Property	Value Description
„ <i>Variable</i> “	Zuzuweisen ist eine Variable, deren Wert als Ladebalken visualisiert wird.
„ <i>Minimaler Wert</i> “	Hier ist der kleinste mögliche Wert anzugeben.
„ <i>Maximaler Wert</i> “	Hier ist der größte mögliche Wert anzugeben.
„ <i>Stil</i> “	Auswahl eines aus folgenden Stilen: <ul style="list-style-type: none"><li>▪ Blöcke</li><li>▪ Balken</li></ul>

### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„ <i>X</i> “	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.

- „Y“ Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
- „Breite“ Horizontale Breite des Elements.
- „Höhe“ Vertikale Höhe des Elements.

## Texte

Diese Eigenschaften werden für eine statische Definition der Elementbeschriftung verwendet.

Property	Value description
----------	-------------------

- „Text“ Der angegebene Text beschriftet das Element. Ein textformatierender String ist möglich.

## Zustandsvariablen

Property	Value description
----------	-------------------

- „Unsichtbarkeit“ Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.


Eigenschaft	Beschreibung Wert
-------------	-------------------

- „Zugriffsrechte“ Mit Klick öffnet [↘ „Zugriffsrechte“](#).

Statusmeldungen:

- Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.
- Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.

## Radiobutton

Ein Radiobutton erlaubt es eine beliebige Anzahl an Optionen durch Verwendung der Schaltfläche  zu konfigurieren, um sie in einer oder mehreren Spalten anzuordnen.

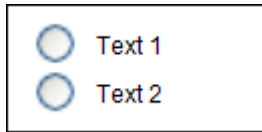


Abb. 72: Radiobutton

### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungseditor werden folgende Benutzereingaben zur Verfügung gestellt:

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Radiobutton

#### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„ <i>X</i> “	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„ <i>Y</i> “	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„ <i>Breite</i> “	Horizontale Breite des Elements.
„ <i>Höhe</i> “	Vertikale Höhe des Elements.

Eigenschaft	Beschreibung Wert
„ <i>Variable</i> “	Zuzuweisen ist eine Integer-Variable, die mit dem Index des selektierten Auswahlknopfes beschrieben wird.
„ <i>Anzahl der Spalten</i> “	Hier ist die Spaltenanzahl, mit der die Radiobuttons angeordnet werden, anzugeben.
„ <i>Anordnung der</i> “	Von links nach rechts: Die Radiobuttons werden Zeile für Zeile angeordnet, bis

„Radiobuttons“ die Anzahl an Spalten erreicht ist.

Von oben nach unten: Die Radiobuttons werden spaltenweise angeordnet, bis die Anzahl an Spalten erreicht ist.

## Zustandsvariablen

Eigenschaft	Beschreibung Wert
„Unsichtbarkeit“	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.
„Eingaben deaktivieren“	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann hat keine der Elementeingaben eine Auswirkung.

## Radiobuttoneinstellungen

Eigenschaft	Beschreibung Wert
„Radiobutton“	„Neu erstellen“: Mit Klick auf diese Schaltfläche wird ein neuer Auswahlknopf im Editor erzeugt und ein weiterer Bereich wird im Eigenschafteneditor aufgelistet.
▪ „Bereich“	Für jeden Radiobutton ist ein Bereich sichtbar, der die Einstellungen erfasst.
▪ „[<n>“	▪ [<n>] <ul style="list-style-type: none"><li>▪ „[&lt;n&gt;“: Die Nummer indiziert den Bereich. Mit Klick auf „Löschen“ wird der zugehörige Radiobutton mit seinen Einstellungen „Text“, „Tooltip“ und „Zeilenabstand in Pixel“ gelöscht.</li></ul>

## Radiobuttoneinstellungen: Radiobutton: Bereiche: [<n>]

Eigenschaft	Beschreibung Wert
„Text“	Hier ist der gewünschte Button-Name anzugeben. Standardwert ist „Radiobutton“
„Tooltip“	Hier ist ein Text, der als Tooltip auszugeben ist, einzugeben.
„Zeilenabstand in Pixel“	Hier kann der Abstand in Pixel zum oberen Button eingegeben werden.

Eigenschaft	Beschreibung Wert
-------------	-------------------

„Zugriffsrechte“ Mit Klick öffnet [„Zugriffsrechte“](#).

Statusmeldungen:

- Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.
- Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.

## Checkbox

Mit einer Checkbox kann eine boolsche Variable gesetzt und rückgesetzt werden. Ist der Haken gesetzt, wird die Variable auf TRUE gesetzt.

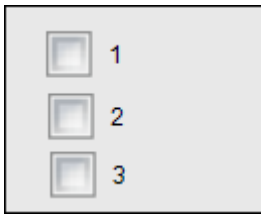


Abb. 73: Checkbox

### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungsektor werden folgende Benutzereingaben zur Verfügung gestellt:

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„Elementname“	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„Elementtyp“	Checkbox

#### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„X“	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„Y“	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„Breite“	Horizontale Breite des Elements.
„Höhe“	Vertikale Höhe des Elements.

#### Texte

Diese Eigenschaften werden für eine statische Definition der Elementbeschriftung verwendet.

Property	Value description
<i>„Text“</i>	Der angegebene Text beschriftet das Element. Ein textformatierender String ist möglich.
<i>„Tooltip“</i>	Der angegebenen Text erscheint als Tooltip des Elements, wenn die Visualisierung im Online-Modus ist.

#### Zustandsvariablen

Eigenschaft	Beschreibung Wert
<i>„Unsichtbarkeit“</i>	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.
<i>„Eingaben deaktivieren“</i>	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann hat keine der Elementeingaben eine Auswirkung.

Eigenschaft	Beschreibung Wert
<i>„Zugriffsrechte“</i>	Mit Klick öffnet <a href="#">„Zugriffsrechte“</a> .

#### Statusmeldungen:

- Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.
- Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.



## 6.3.5.4 Alarmmanager

### Alarmtabelle

Das Element ermöglicht es, Alarme einer bestimmten Alarmgruppe und Alarmklasse darzustellen. Sie kann so konfiguriert werden, dass Eigenschaften des Alarms in speziellen Spalten angezeigt werden und sie kann um eine Leiste mit zusätzlichen Kontrollelementen erweitert werden.

Es ist möglich, Alarme gemäß dem Alarmmanagement der aktuellen Applikation zu visualisieren. Im Online-Modus kann der Benutzer Aktionen wie Sortieren, Bestätigung und Anzeige der Alarmhistorie verwenden. Siehe dazu Kapitel [Alarmmanagement](#)".

	Bitmap	Time stamp	Message	Value
0		05.03.2012 10:00:41	Level of unit 2 is too high	86
1		05.03.2012 10:00:40	Level of unit 2 is over 40	86
2		05.03.2012 10:00:15	Temperature of unit 2 is too high	45

Abb. 74: Alarmtabelle, Style 8

### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungsektor werden folgende Benutzereingaben zur Verfügung gestellt:

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemlnst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Alarmtabelle
„ <i>Applikation</i> “	Soll eine ferne Alarmkonfiguration verwendet werden, ist hier der Name der Fern-Applikation anzugeben. Wenn nichts angegeben ist, sollte die Alarmkonfiguration lokal aufzufinden sein.

#### Alarmkonfiguration

Eigenschaft	Beschreibung Wert
„ <i>Alarmgruppen</i> “	Der Dialog <a href="#">„Auswahl Alarmgruppe“</a> erlaubt es zu definieren, welche Alarmgruppen in der Tabelle dargestellt werden sollen.
„ <i>Priorität von</i> “	Hier wird die kleinste Priorität definiert, bei der Alarme angezeigt werden. Der Wertebereich reicht von 0 bis 255.
„ <i>Priorität bis</i> “	Hier wird die größte Priorität definiert, bei der Alarme angezeigt werden. Der Wertebereich reicht von 0 bis 255.
„ <i>Alarmklassen</i> “	Der Dialog <a href="#">„Auswahl Alarmklasse“</a> erlaubt es zu definieren, welche Alarmklassen in der Tabelle dargestellt werden.

#### Allgemeine Tabellenkonfiguration

Eigenschaft	Beschreibung Wert
„ <i>Zeilenkopf anzeigen</i> “	Checkbox, um die Anzeige der Beschriftung der Reihen in Form des zugehörigen Spaltenindex des Arrays an- bzw. auszuschalten
„ <i>Spaltenkopf anzeigen</i> “	Checkbox, um die Anzeige der Beschriftung der Spalten definiert in „ <i>Spaltenüberschrift</i> “ an- bzw. auszuschalten
„ <i>Zeilenhöhe</i> “	Höhe der Tabellenreihe in Pixel.
„ <i>Breite Zeilenkopf</i> “	Breite des Zeilenkopfes, der die Überschrift enthält, in Pixel.
„ <i>Größe des Scrollbars</i> “	Höhe des Scrollbars, wenn er vertikal, Breite, wenn er horizontal verläuft.

### Spalten: Spalte [<n>]


Standardmäßig sind die Spalten [0] und [1] konfiguriert: Zeitstempel und Meldungstext. Mit Schaltfläche „*Neu erstellen*“ können weitere Spalten erzeugt werden, mit „*Löschen*“ kann eine einzelne gelöscht werden.

Animationen (dynamischer Text, Schriftartvariablen), Text und Tooltip werden nicht unterstützt.

Eigenschaft	Beschreibung Wert
„ <i>Spaltenüberschrift</i> “	Die Überschrift wird hier durch Eingabe eines neuen Textes gesetzt und geändert.
„ <i>Textausrichtung im Titel verwenden</i> “	Checkbox, um die Textausrichtung definiert in „ <i>Textausrichtung</i> “ auch für die Kopfzeile zu aktivieren.
„ <i>Breite</i> “	Breite der Spalte in Pixel.
„ <i>Datentyp</i> “	Hier ist zu definieren, welche Alarminformation in der zugehörigen Spalte angezeigt werden sollte: Auszuwählen ist eine aus: Symbol Zeitstempel: Datum und Zeit der letzten Statusänderung des Alarms. Zeitstempel aktiv: Datum und Zeit des letzten Aktivwerdens des Alarms. Zeitstempel inaktiv: Datum und Zeit des letzten Nichtaktivwerdens des Alarms. Zeitstempel Bestätigung: Datum und Zeit der letzten Bestätigung. Wert: aktueller Wert des Ausdrucks Meldungstext: zum beobachtet werden Priorität: Alarmpriorität Klasse: Alarmklasse Zustand: Alarmzustand
„ <i>Textausrichtung</i> “	Ausrichtung der Spaltenüberschrift Links

Zentriert

Rechts

	Im Unterschied zu CODESYS V2.3 gibt es die Eigenschaften „Zielwert“, „Typ“ und „Totzeit“ nicht mehr und <b>Sortierung</b> ist in der Alarmtabelle nicht konfigurierbar.
---	---

## Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.


Eigenschaft	Beschreibung Wert
„X“	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„Y“	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„Breite“	Horizontale Breite des Elements.
„Höhe“	Vertikale Höhe des Elements.

## Texteigenschaften

Diese Eigenschaften werden für eine statische Definition der Schriftart verwendet.

Eigenschaft	Beschreibung Wert
„Schriftart“	Auswahl aus vordefinierten Schriftarten: Standard Überschrift Große Titel Anmerkung Mit Klick auf <input type="button" value="..."/> öffnet sich der Dialog für benutzerdefinierte Eigenschaften der Schriftart.
„Farbe Schriftart“	Auszuwählen ist eine Schriftartfarbe für das Element. Entweder aus der Auswahlliste oder durch Setzen im sich öffnenden Dialog, wenn <input type="button" value="..."/> geklickt wird.

## Selektion

	Im Gegensatz zu einem Tabellenelement können diese Variablen nur gelesen und nicht beschrieben werden. Das bedeutet, dass eine Änderung des Variablenwerts im Applikationsprogramm keine Auswirkung auf die Alarmtabelle hat.
---	---

Eigenschaft	Beschreibung Wert
„Selektionsfarbe“	Füllfarbe der selektierten Zeile


„Rahmen um selektierte Zellen“	Checkbox, um einen Rahmen um den selektierten Teil an- und auszuschalten.
„Variable für selektierte Alarmgruppe“	Variable des Typs STRING, die den Namen der betroffenen Alarmgruppe enthält.
„Variable für selektierte Alarm-ID“	Variable des Typs STRING, die die Alarm-ID enthält.

### Steuervariablen

Hier können boolsche Variablen definiert werden, über die im Online-Betrieb bestimmte Aktionen auf die Tabelle ausgeführt werden können. Wenn Sie keine in der Applikation definierte Variable angeben (beispielsweise PLC\_PRG.bAcknowledgeSelected), dann wird das System automatisch eine visualisierungslokale Variable erzeugen und eintragen, beispielsweise bAckSel für „Auswahl bestätigen“.

Es gibt einen Assistenten für das Einfügen zusätzlicher Kontrollelemente unterhalb der Tabelle, eines für jede der beschriebenen vier Aktionen. Standardmäßig sind sie folgendermaßen benannt und beschriftet: „ACK selected“, „ACK all visible“, „History“, „Freeze Scrl Pos“.

Eigenschaft	Beschreibung Wert
„Auswahl bestätigen“	Standardvariable: bAckSel Wird die zugewiesene Variable TRUE, dann werden die ausgewählten Alarmer quittiert, wenn sie im sichtbaren Teil der Alarmtabelle liegen.
„Auswahl bestätigen“	Standardvariable: bAckVisible Wird die zugewiesene Variable TRUE, dann werden alle Alarmer, die gerade in der Alarmtabelle sichtbar sind, quittiert.
„Historie“	Standardvariable: bHistory Wird die zugewiesene Variable TRUE, dann wird die Alarmtabelle in den Historienmodus geschaltet. Das bedeutet, dass alle Zeitstempel des Aktivwerdens des Alarms in absteigender Reihenfolge aufgelistet werden, bis hin zur aktuellen Zeit; entsprechend dem Datentyp "Zeitstempel Aktiv". Man beachte, dass in dieser Auflistung keine Quittierung möglich ist ! Neue Ereignisse werden laufend hinzugefügt. Wenn ein Alarmspeicherungsdatei definiert ist, wird diese Historie auch dort für alle Alarmer gespeichert, für deren Klasse Alarmspeicherung aktiviert ist.
„Scroll-Position einfrieren“	Standardvariable: bFreezeScrlPos Wird die zugewiesene Variable TRUE, dann wird die aktuell im Historienmodus eingestellte Scroll-Position beibehalten, auch wenn ein neuer Alarm aktiv wird (andernfalls springt die Scroll-Position in diesem Fall zur ersten Tabellenzeile).

	Im Gegensatz zu CODESYS V2.3 die Option „Statuszeile anzeigen“ ist nicht verfügbar.
---	---

Eigenschaft	Beschreibung Wert
„Zugriffsrechte“	Mit Klick öffnet <a href="#">„Zugriffsrechte“</a> . Statusmeldungen: Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle

Gruppen bedienbar angezeigt wird.

Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.

### Assistent für das Einfügen zusätzlicher Kontrollelemente

Der Assistent für das Einfügen zusätzlicher Kontrollelemente wird mit Befehl [↘ „Einfügen von Elementen zur Alarmquittierung“](#) gestartet. Dieser Befehl ist standardmäßig im Visualisierungsmenü und im Kontextmenü verfügbar, wenn eine Alarmtabelle selektiert ist. Man beachte, dass eventuell bereits in der Applikation definierte Kontrollvariablen, die im Eigenschafteneditor der Alarmtabelle eingetragen sind, automatisch den entsprechenden Kontrollelementen zugeordnet werden. Dies bewirkt, dass die Kontrollvariablen im Eigenschafteneditor der Kontrollelemente in Sektion [Eingabekonfiguration → Umschalten](#) eingetragen werden. Wenn noch keine Applikationsvariable einer Aktion zugewiesen ist, wird das System nun automatisch eine visualisierungslokale Variable dafür anlegen und zuweisen. Die Standardnamen für die implizit erzeugten Variablen für die vier Kontrollaktionen sind: bAckSel, bAckVisible, bHistory, bFreezeScrlPos.

### Vorgehensweise bei Erstellung einer Alarmtabelle

Beispiel

1. Ziehen Sie das Element aus der Werkzeugbox ins Editorfenster. Im [„Eigenschaften“-](#)Editor des Elements muss dann definiert werden, wie die Tabelle aussehen und funktionieren soll:

	Bitmap	Time stamp	Message	Value
0		05.03.2012 10:00:41	Level of unit 2 is too high	86
1		05.03.2012 10:00:40	Level of unit 2 is over 40	86
2		05.03.2012 10:00:15	Temperature of unit 2 is too high	45

Abb. 75: Alarmtabellenelement

2. Definieren Sie in [„Eigenschaften“](#), welche Alarme visualisiert werden sollen, indem Sie die Gruppen, Klassen und Prioritäten in [„Alarmkonfiguration“](#) festlegen.
3. Definieren Sie, welche Spalten die Tabelle haben soll. Standardmäßig ist Zeitstempel und Meldungstext vordefiniert, dies kann jedoch in [↘ „Spalten: Spalte \[n\]“](#) verändert und erweitert werden.
4. Definieren Sie Position and Aussehen der Tabelle, letzteres durch Setzen der Farben und Schriftarten für die Texte, in [„Position“](#), [„Allgemeine Tabellenkonfiguration“](#) und [„Texteigenschaften“](#).
5. Definieren Sie das Aussehen von selektierten Tabellenzellen und Variablen in [„Selektion“](#)
6. Legen Sie Variablen für die Kontrolle der Alarme über die Applikation in [„Steuervariablen“](#) fest.

### Benutzereingaben einer Alarmtabelle, die sich im Online-Modus befindet

Im Online-Betrieb kann die Sortierung durch Mausklicks in die Spaltenüberschriften verändert werden. Dabei gilt, dass ein Klick in die Spaltenüberschrift die gesamte Tabelle aufsteigend nach den Daten der Spalte sortiert. Ein weiterer Klick schaltet auf absteigende Sortierung um, wiederum ein weiterer zurück zur Aufsteigenden.

Eine Selektion im Online-Modus (durch Mausklick) wirkt sich immer auf die gesamte Tabellenzeile aus. Mehrfachselektion ist nicht möglich.

### Alarmtabellevisualisierung im Online-Modus

Beispiel

	Timestamp	Message	State	Bitmap
0	11.07.2011 15:13:25	Low temperature heat right-142.80	ACKnowledge	
1	11.07.2011 15:13:25	Contacteur heat right defect	ACKnowledge	
2	11.07.2011 15:13:22	Probe heat left defect FALSE	ACKnowledge	
3	11.07.2011 15:13:10	Probe heat right defect TRUE	Active	
4	11.07.2011 15:12:58	Low temperature heat left-190.40	ACKnowledge	
5	11.07.2011 15:12:49	Heat middle OFF	Active	
6	11.07.2011 15:12:46	Heat left OFF	Active	

HeatOn	<b>Contacteur Heat</b>	HeatOn	<b>Contacteur Heat</b>	HeatOn	<b>Contacteur Heat</b>
Heating		Heating		Heating	Heat Right
<b>Cooling</b>		Cooling		<b>Cooling</b>	
Preset 20.0		Preset 20.0		Preset 20.0	
Actual -190.602		Actual 19.0		Actual -142.999	

▼ Abb. 76: Alarmvisualisierung im Online-Modus

## Alarmbanner

Das Alarmbanner ist eine vereinfachte Version der Alarmtabelle. Es visualisiert nur einen einzigen Alarm, der zu den Alarmgruppen und -klassen gehört, die in der Kategorie "Alarmkonfiguration" der Banner-Elementeigenschaften angegeben sind. Mit der Eigenschaft „*Filterkriterium*“ muss definiert werden, ob der neueste oder der wichtigste (am höchsten priorisierte) Alarm dargestellt werden soll.



Abb. 77: Alarmbanner, Style 8

### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungsektor werden folgende Benutzereingaben zur Verfügung gestellt:

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Alarmbanner
„ <i>Applikation</i> “	Soll eine ferne Alarmkonfiguration verwendet werden, ist hier der Name der Fern-Applikation anzugeben. Wenn nichts angegeben ist, sollte die Alarmkonfiguration lokal aufzufinden sein.

#### Alarmkonfiguration

Eigenschaft	Beschreibung Wert
„ <i>Alarmgruppen</i> “	Der Dialog <a href="#">↘ „Auswahl Alarmgruppe“</a> erlaubt es zu definieren, welche Alarmgruppen in der Tabelle dargestellt werden sollen.
„ <i>Priorität von</i> “	Hier wird die kleinste Priorität definiert, bei der Alarme angezeigt werden. Der Wertebereich reicht von 0 bis 255.
„ <i>Priorität bis</i> “	Hier wird die größte Priorität definiert, bei der Alarme angezeigt werden. Der Wertebereich reicht von 0 bis 255.
„ <i>Alarmklassen</i> “	Der Dialog <a href="#">↘ „Auswahl Alarmklasse“</a> erlaubt es zu definieren, welche Alarmklassen in der Tabelle dargestellt werden.

Property	Value Description
„ <i>Filterkriterium</i> “	Zu definieren ist, ob der letzte oder der wichtigste (höchste Priorität) Alarm angezeigt wird: <ul style="list-style-type: none"><li>▪ „<i>Wichtigster</i>“</li></ul>

- „*Neuester*“

## Spalten: Spalte [<n>]

Standardmäßig sind die Spalten [0] und [1] konfiguriert: Zeitstempel und Meldungstext. Mit Schaltfläche „*Neu erstellen*“ können weitere Spalten erzeugt werden, mit „*Löschen*“ kann eine einzelne gelöscht werden.

Animationen (dynamischer Text, Schriftartvariablen), Text und Tooltip werden nicht unterstützt.

Eigenschaft	Beschreibung Wert
„ <i>Breite</i> “	Breite der Spalte in Pixel.
„ <i>Datentyp</i> “	<p>Hier ist zu definieren, welche Alerminformation in der zugehörigen Spalte angezeigt werden sollte:</p> <p>Auszuwählen ist eine aus:</p> <ul style="list-style-type: none"> <li>▪ Symbol</li> <li>▪ Zeitstempel: Datum und Zeit der letzten Statusänderung des Alarms.</li> <li>▪ Zeitstempel aktiv: Datum und Zeit des letzten Aktivwerdens des Alarms.</li> <li>▪ Zeitstempel inaktiv: Datum und Zeit des letzten Nichtaktivwerdens des Alarms.</li> <li>▪ Zeitstempel Bestätigung: Datum und Zeit der letzten Bestätigung.</li> <li>▪ Wert: aktueller Wert des Ausdrucks</li> <li>▪ Meldungstext: zum beobachtet werden</li> <li>▪ Priorität: Alarmpriorität</li> <li>▪ Klasse: Alarmklasse</li> <li>▪ Zustand: Alarmzustand</li> </ul>
„ <i>Textausrichtung</i> “	<p>Ausrichtung der Spaltenüberschrift</p> <ul style="list-style-type: none"> <li>▪ Links</li> <li>▪ Zentriert</li> <li>▪ Rechts</li> </ul>

## Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
-------------	-------------------



- „X“ Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
- „Y“ Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
- „Breite“ Horizontale Breite des Elements.
- „Höhe“ Vertikale Höhe des Elements.

### Texteigenschaften

Diese Eigenschaften werden für eine statische Definition der Schriftart verwendet.

Eigenschaft	Beschreibung Wert
„Schriftart“	<p>Auswahl aus vordefinierten Schriftarten:</p> <ul style="list-style-type: none"> <li>▪ Standard</li> <li>▪ Überschrift</li> <li>▪ Große</li> <li>▪ Titel</li> <li>▪ Anmerkung</li> </ul> <p>Mit Klick auf <input type="button" value="..."/> öffnet sich der Dialog für benutzerdefinierte Eigenschaften der Schriftart.</p>
„Farbe Schriftart“	<p>Auszuwählen ist eine Schriftartfarbe für das Element. Entweder aus der Auswahlliste oder durch Setzen im sich öffnenden Dialog, wenn <input type="button" value="..."/> geklickt wird.</p>

Property	Value Description
„Bestätigungsvariable“	Hier ist eine Variable anzugeben, die den Bestätigungstext enthalten wird.

Eigenschaft	Beschreibung Wert
„Zugriffsrechte“	<p>Mit Klick öffnet <a href="#">„Zugriffsrechte“</a>.</p> <p>Statusmeldungen:</p> <ul style="list-style-type: none"> <li>▪ Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.</li> <li>▪ Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.</li> </ul>

## Auswahl Alarmgruppe

Dieser Dialog öffnet mit Klick auf die Eigenschaft „Alarmgruppe“ eines Alarmentablenelements.

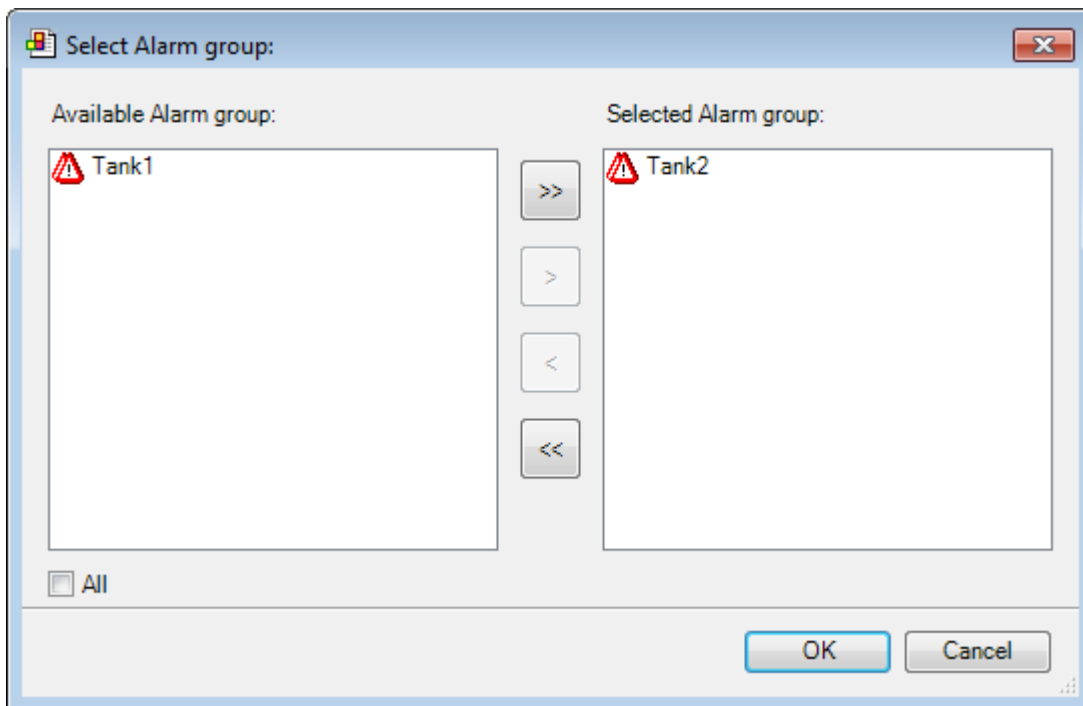


Abb. 78: Auswahl Alarmgruppe

Mit Hilfe der Pfeiltasten kann eine Alarmgruppe in die Liste oder aus der Liste „Ausgewählte Alarmgruppe“ genommen werden. Mit aktivierter Checkbox „Alle“ wird jede verfügbare Alarmgruppe selektiert.

## Auswahl Alarmklasse

Dieser Dialog öffnet mit Klick auf die Eigenschaft „Alarmklasse“ eines Alarmtabellenelements

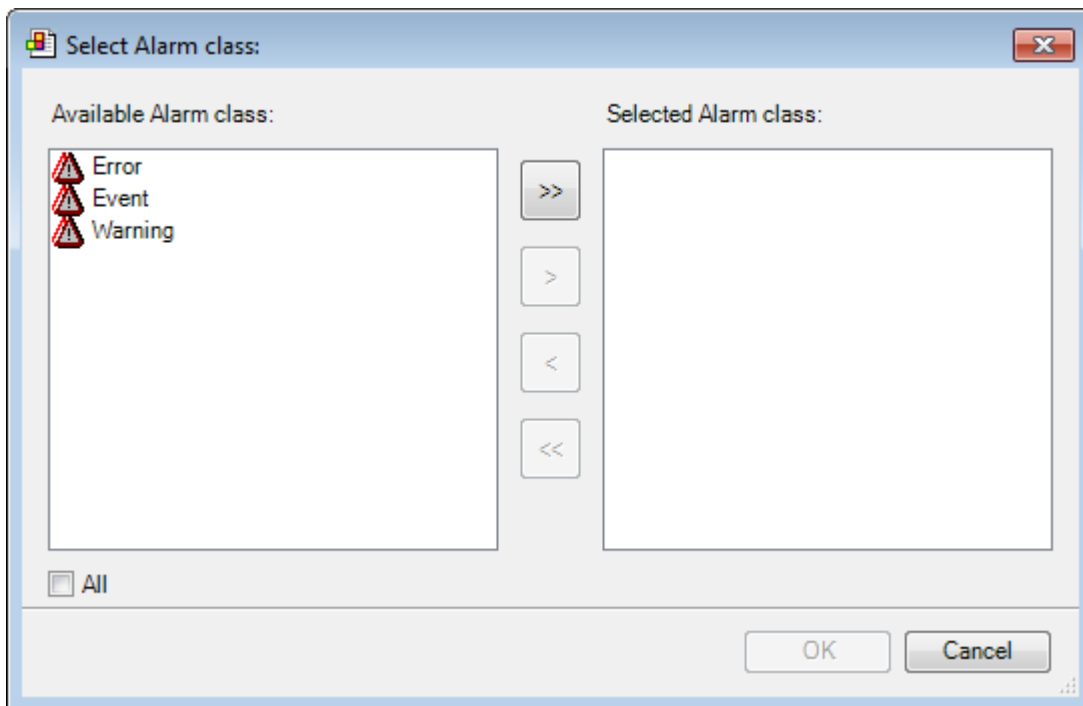


Abb. 79: Auswahl Alarmklasse

Mit Hilfe der Pfeiltasten kann eine Alarmklasse in die Liste oder aus der Liste „Ausgewählte Alarmklasse“ genommen werden. Mit aktivierter Checkbox „Alle“ wird jede verfügbare Alarmklasse selektiert.

### 6.3.5.5 Zeigerinstrumente/Potentiometer/Histogramm

Balkenanzeige, Balkenanzeige frei konfigurierbar

Hiermit kann eine Balkenanzeige der Visualisierung hinzugefügt werden. „Balkenanzeige“ hat ein vordefiniertes Design, bei dem die Hintergrundfarbe wählbar ist. Mit „Balkenanzeige frei konfigurierbar“ kann ein Bild aus einer Bildersammlung dem Element als Hintergrund zugewiesen werden. Das ist der einzige Unterschied.

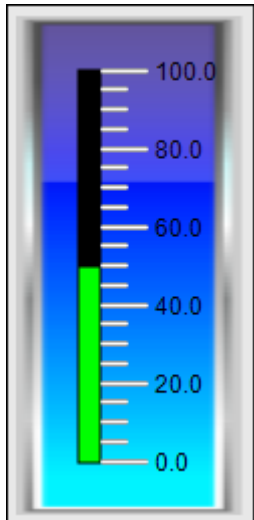


Abb. 80: Balkenanzeige

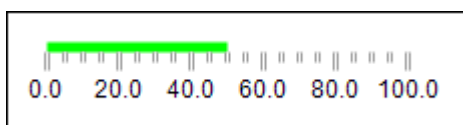


Abb. 81: Balkenanzeige frei konfigurierbar

#### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungseditor werden folgende Benutzereingaben zur Verfügung gestellt:

Element spezifische Mauseingabe

- Standard ist, dass der Balken horizontal angezeigt wird. Wurde ein Element eingefügt, so kann dessen Form mit Hilfe der Maus geändert werden. Sobald dessen Höhe die Breite überschreitet, erscheint er vertikal und *Balken* → *Ausrichtung* ist auf *Vertikal* gesetzt.

#### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

##### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„Elementname“	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.

„*Type Elementtyp*“ Balkenanzeige oder Balkenanzeige frei konfigurierbar

„*Wert*“ Zuzuweisen ist eine Variable, deren Wert als Balkenlänge dargestellt wird.

## Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„ <i>X</i> “	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„ <i>Y</i> “	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„ <i>Breite</i> “	Horizontale Breite des Elements.
„ <i>Höhe</i> “	Vertikale Höhe des Elements.

## Hintergrund

Wird keine frei konfigurierbare „*Balkenanzeige*“ verwendet, dann hat „*Hintergrund*“ folgende Auswirkung:

Eigenschaft	Beschreibung Wert
„ <i>Hintergrundbild</i> “	Auszuwählen ist eine Farbe für den Balken: <ul style="list-style-type: none"><li>▪ Gelb</li><li>▪ Rot</li><li>▪ Grün</li><li>▪ Blau</li><li>▪ Grau</li></ul>

Wird „*Balkenanzeige frei konfigurierbar*“ verwendet, dann hat „*Hintergrund*“ folgende Auswirkung:

Eigenschaft	Beschreibung Wert
„ <i>Hintergrundbild</i> “	Zuzuweisen ist ein Bild aus dem GlobalImagePool, indem der Name der Bilddatei oder dessen ID angegeben wird.

## Balken

Eigenschaft	Beschreibung Wert
„ <i>Diagrammart</i> “	Die Drop-down-Liste stellt verschiedenen Möglichkeiten zur Verfügung, wie Balken und Skale plaziert werden können: <ul style="list-style-type: none"><li>▪ Skala neben Balken</li></ul>

- Skala im Balken
- Balken in Skala
- Keine Skala.

„Ausrichtung“ Der Balken kann `Horizontal` oder `vertikal` ausgerichtet sein. Die Ausrichtung entsteht aus dem Verhältnis von Breite zu Höhe und kann hier **nicht** editiert werden. Diese Änderung erfolgt im Visualisierungseditor, indem ein Eckpunkt es Elements mit der Maus "gegriffen" und horizontal oder vertikal gezogen wird ( [↘ „Balkenanzeige, Balkenanzeige frei konfigurierbar“](#)).


„Laufrichtung“ Ist die Ausrichtung horizontal, kann gewählt werden zwischen:

- Links nach rechts
- Rechts nach Links

Ist die Ausrichtung vertikal, kann gewählt werden zwischen:

- Von unten nach oben
- Von oben nach unten

## Skala

Eigenschaft	Beschreibung Wert
„Skalenanfang“	Wert, der die Skalierungsanzeige von unten begrenzt
„Skalenende“	Wert, der die Skalierungsanzeige von oben begrenzt
„Hauptskala“	Abstand zwischen zwei Strichen der Grobskala
„Unterskala“	Abstand zwischen zwei Strichen der Feinskala. Kann auf 0 gesetzt werden, falls eine weiter Unterteilung der Grobskala nicht gewünscht ist.
„Skalenlinienstärke“	Der Wert definiert die Dicke der Skalenlinie in Pixel.
„Skalenfarbe“	Die Farbe kann über Selektionsliste bzw. über Schaltfläche  aus dem Standarddialog zur Farbauswahl eingestellt werden.
„Skala in 3D“	Ist diese Option aktiviert, wird die Skala 3-dimensionale dargestellt.
„Elementrahmen“	Ist diese Option aktivier, wird ein Rahmen um die Balkenanzeige gezogen.

## Beschriftung

Eigenschaft	Beschreibung Wert
„Einheit“	Der eingegebene Text beschriftet das Element.  String, der unterhalb des Pfeils (M) bzw. unterhalb des Mittelpunkts der Skalierung (BD) angezeigt wird (gedacht für die Eingabe der Skalierungseinheit).
„Schriftart“	Festzulegen ist die Schriftart für Einheit und Skalierung:

Auszuwählen ist:

- Standard
- Titelzeile
- Large
- Title
- Annotation

Mit Klick auf  öffnet ein Dialog für benutzerdefinierte Einstellung der Schrifteigenschaften.

#### „Skalenformat (C-Syntax)“

Benutzen Sie die C-Syntax zur Angabe der Formatierung der Skalierungsbeschriftung. So führt die Eingabe des Strings "%3.2f s" in diesem Feld zur Anzeige der Skalierungsbeschriftungen mit 3 Stellen, davon 2 Nachkommastellen, gefolgt vom Buchstaben "s"

#### „Max. Breite der Beschriftungen“

Wert, der die maximale Breite der Skalierungsbeschriftung vorgibt. In der Regel wird dieser Wert automatisch richtig festgesetzt. Benützen Sie diese Vorgabe nur, falls die automatische Anpassung nicht zum gewünschten Ergebnis führt.

#### „Höhe der Beschriftungen“

Wert, der die Höhe der Skalierungsbeschriftung vorgibt; in der Regel wird dieser Wert automatisch richtig festgesetzt. Benützen Sie diese Vorgabe nur, falls die automatische Anpassung nicht zum gewünschten Ergebnis führt.

#### „Farbe Schriftart“

Farbe für die Schrift kann über Selektionsliste bzw. über Schaltfläche  aus dem Standarddialog zur Farbauswahl eingestellt werden.

### Positionierung

Eigenschaft	Beschreibung Wert
„Horizontale Verschiebung“	Hier ist der Abstand in Pixel der Skala/des Balkens zum horizontalen Elementrahmen anzugeben.
„Vertikale Verschiebung“	Hier ist der Abstand in Pixel der Skala/des Balkens zum vertikalen Elementrahmen anzugeben.

### Farben

Eigenschaft	Beschreibung Wert
„Farbe des Graphen“	Anzugeben ist eine Farbe für den Balken.
„Balkenhintergrund“	Standard ist, dass die Hintergrundfarbe der Balkenanzeige weiß ist. Wird die Option aktiviert, wird die Hintergrundfarbe in schwarz geändert.
„Rahmenfarbe“	Farbe des Rahmens, der die Balkenanzeige umgibt, sofern das Kontrollkästchen „Elementrahmen“ aktiviert ist.
„Farbbereiche verwenden“	Bei Aktivierung dieses Kontrollkästchens werden die Farbbereiche, so wie sie unter „Farbbereiche“ definiert, angezeigt.
„Gesamte Farbe umschalten“	Ist diese Option aktiviert, ändert der gesamte Balken seine Farbe, wenn der

Variablenwert den Anfangs-/Endwert der Skala unter- oder überschreitet.

„*Farbverlauf für Balken verwenden*“

Bei Aktivierung dieser Option wird der Balken mit Farbverlauf angezeigt.

„*Farbbereichsmarkierung*“

Auswahl zwischen

- Keine Markierung
- Markierung vorwärts
- Markierung rückwärts

„*Farbbereiche*“

„*Neu erstellen*“: Mit Klick auf diesen Button wird ein neuer Farbbereich erzeugt und in Eigenschaften aufgelistet.

- „*[<n>*“

### Farben: Farbbereiche: Bereiche [<n>]

Anzahl der Farbbereiche in aufsteigender, nicht editierbarer Reihenfolge.

Nach Anklicken von  wird der zugehörige Farbbereich gelöscht und die Nummerierung neu vergeben.

Eigenschaft	Beschreibung Wert
„ <i>Bereichsanfang</i> “	Wert, der den Farbbereich von unten begrenzt
„ <i>Bereichsende</i> “	Wert, der den Farbbereich von oben begrenzt
„ <i>Farbe</i> “	Farbe, in der der Farbbereich dargestellt werden soll

Eigenschaft	Beschreibung Wert
„ <i>Zugriffsrechte</i> “	Mit Klick öffnet <a href="#">„Zugriffsrechte“</a> .

Statusmeldungen:

- Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.
- Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.

### Zeigerinstrument 90°

Mit „*Zeigerinstrument*“ kann zum Beispiel ein Drehzahlmesser der Visualisierung hinzugefügt werden. Die Nadel positioniert sich nach dem Wert der zugewiesenen Variablen. Es hat ein vorkonfiguriertes Design, in dem eine Hintergrundfarbe gesetzt werden kann.



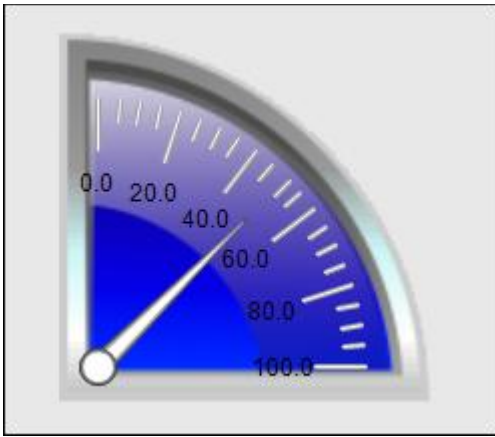


Abb. 82: Zeigerinstrument 90°

#### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungseditor werden folgende Benutzereingaben zur Verfügung gestellt:

#### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Zeigerinstrument 90°
„ <i>Wert</i> “	Zuzuweisen ist eine Variable, deren Wert als <b>Ausschlag</b> der Instrumentennadel dargestellt wird.

#### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„ <i>X</i> “	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„ <i>Y</i> “	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„ <i>Breite</i> “	Horizontale Breite des Elements.
„ <i>Höhe</i> “	Vertikale Höhe des Elements.

#### Hintergrund

Eigenschaft	Beschreibung Wert
<i>„Hintergrundbild“</i>	Auszuwählen ist eine Farbe für den Balken: <ul style="list-style-type: none"> <li>▪ Gelb</li> <li>▪ Rot</li> <li>▪ Grün</li> <li>▪ Blau</li> <li>▪ Grau</li> </ul>

## Zeiger

Eigenschaft	Beschreibung Wert
<i>„Zeigertyp“</i>	Auszuwählen ist aus: <ul style="list-style-type: none"> <li>▪ Normaler Zeiger</li> <li>▪ Dünner Zeiger</li> <li>▪ Breiter Zeiger</li> <li>▪ Dünne Nadel</li> <li>▪ Dünner 3D-Zeiger</li> <li>▪ Dünner 3D-Nadel</li> </ul>
<i>„Farbe“</i>	Farbe, mit der der Zeiger angezeigt wird.
<i>„Winkelbereich“</i>	Auswahl eines 90°-Ausschnitts: <ul style="list-style-type: none"> <li>▪ Oben rechts</li> <li>▪ Oben links</li> <li>▪ Unten links</li> <li>▪ Unten rechts</li> </ul>
<i>„Zusätzlicher Zeiger“</i>	Ist die Option aktiviert, ist ein zusätzlicher Zeiger auf der Skala gegenüber der Nadel sichtbar.

## Skala

Eigenschaft	Beschreibung Wert
<i>„Unterskalaposition“</i>	Die Unterskala kann am äußeren oder inneren Radius des Skalenrings angezeigt werden: <ul style="list-style-type: none"> <li>▪ Außen</li> <li>▪ Innen</li> </ul>
<i>„Skalentyp“</i>	Die Skala kann angezeigt werden als: <ul style="list-style-type: none"> <li>▪ Linien</li> <li>▪ Punkte</li> <li>▪ Quadrate</li> </ul>


„Skalenanfang“ „“	Wert, der die Skalierungsanzeige von unten begrenzt.
„Skalenende“	Wert, der die Skalierungsanzeige von oben begrenzt.
„Hauptskala“	Abstand zwischen zwei Strichen der Grobskala.
„Unterskala“	Abstand zwischen zwei Strichen der Feinskala; kann auf 0 gesetzt werden, falls eine weitere Unterteilung der Grobskala nicht gewünscht ist.
„Skalenlinienstärke“	Stärke der Skala in Pixel
„Skalenfarbe“	Die Farbe kann über Selektionsliste bzw. über Schaltfläche <input type="button" value="..."/> aus dem Standarddialog zur Farbauswahl eingestellt werden.
„Skala in 3D“	Ist die Option aktiviert, wird die Skala dreidimensional dargestellt.
„Skala anzeigen“	Wird keine Skalendarstellung gewünscht, ist diese Option zu deaktivieren.
„Innerer Rahmen“	Ist diese Option aktiviert, wird der Skalenring innen mit einem Rahmen versehen. Standardmäßig deaktiviert.
„Äußerer Rahmen“	Ist diese Option aktiviert, wird der Skalenring außen mit einem Rahmen versehen. Standardmäßig deaktiviert.

## Beschriftung

Eigenschaft	Beschreibung Wert
„Beschriftung“	<p>Auswählen aus</p> <ul style="list-style-type: none"> <li>▪ Außen: Skalenwerte werden auf der Außenseite der Skala platziert.</li> <li>▪ Innen: Skalenwerte werden auf der Innenseite der Skala platziert.</li> </ul>
„Einheit“	String, der unterhalb des Pfeils (M) bzw. unterhalb des Mittelpunkts der Skalierung (BD) angezeigt wird (gedacht für die Eingabe der Skalierungseinheit)
„Schriftart“	<p>Schriftart für Darstellung von Einheit und Skalenbereich</p> <p>Festzulegen ist die Schriftart durch auswählen von einem aus:</p> <ul style="list-style-type: none"> <li>▪ Standard</li> <li>▪ Überschrift</li> <li>▪ Große</li> <li>▪ Titel</li> <li>▪ Anmerkung</li> </ul> <p>Mit Klick auf <input type="button" value="..."/> öffnet ein Dialog für eine benutzerdefinierte Einstellung der Schriftart.</p>
„Skalenformat (C-Syntax)“	Hier ist die C-Syntax zur Angabe der Formatierung der Skalierungsbeschriftung zu benutzen. Z.B. führt die Eingabe des Strings "%3.2f s" in diesem Feld zur Anzeige der Skalierungsbeschriftungen mit 3 Stellen, davon 2 Nachkommastellen, gefolgt vom Buchstaben "s"

„Max. Breite der Beschriftungen“ Wert, der die maximale Breite der Skalierungsbeschriftung vorgibt; in der Regel wird dieser Wert automatisch richtig festgesetzt. Benützen Sie diese Vorgabe nur, falls die automatische Anpassung nicht zum gewünschten Ergebnis führt.

„Höhe der Beschriftungen“ Wert, der die Höhe der Skalierungsbeschriftung vorgibt; in der Regel wird dieser Wert automatisch richtig festgesetzt. Benützen Sie diese Vorgabe nur, falls die automatische Anpassung nicht zum gewünschten Ergebnis führt.

„Farbe Schriftart“ Farbe für die Schrift kann über Selektionsliste bzw. über Schaltfläche  aus dem Standarddialog zur Farbauswahl eingestellt werden.

## Positionierung


Eigenschaft	Beschreibung Wert
„Zeigerabstand“	Hier kann die Länge des Zeigers in Pixel angegeben werden.
„Beschriftungsverschiebung“	Hier kann ein Abstand in Pixel definiert werden um die Beschriftung zu positionieren.
„Einheitenverschiebung“	Hier kann ein vertikaler Abstand eingegeben werden, um den Text eingegeben in <i>Beschriftung</i> → <i>Einheit</i> zu positionieren.


## Farben

Property	Value Description
„Farbbereiche“	„Dauerhafte Farbbereiche“: Dessen Auswirkung ist nur im Online-Mode sichtbar. Ist die Option aktiviert, sind die Farbbereiche dauerhaft sichtbar. Ist die Option nicht aktiviert, ist nur der Farbbereich, auf den der Pfeil zeigt, sichtbar.
▪ „Dauerhafte Farbbereiche“	
▪ „[<n>]“	„Neu erstellen“: Mit Klick auf diesen Button wird ein weiterer Farbbereich erzeugt und angezeigt.

## Farben: Farbbereiche: Bereiche: [<n>]

Anzahl der Farbbereiche in aufsteigender, nicht editerbarer Reihenfolge.

„Löschen“: Ein Klick auf den Button  wird den Farbbereich von der Liste entfernen und die Reihenfolge neu arrangieren.

Eigenschaft	Beschreibung Wert
„Bereichsanfang“	Wert, der den Farbbereich von unten begrenzt.
„Bereichsende“	Wert, der den Farbbereich von oben begrenzt.
„Farbe“	Farbe, in der der Farbbereich dargestellt werden soll.
Eigenschaft	Beschreibung Wert
„Zugriffsrechte“	Mit Klick öffnet  „Zugriffsrechte“.

Statusmeldungen:

- Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.
- Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.

in Arbeit

## Zeigerinstrument 180°

Mit „*Zeigerinstrument*“ kann zum Beispiel ein Drehzahlmesser der Visualisierung hinzugefügt werden. Die Nadel positioniert sich nach dem Wert der zugewiesenen Variablen. Es hat ein vorkonfiguriertes Design, in dem eine Hintergrundfarbe gesetzt werden kann.

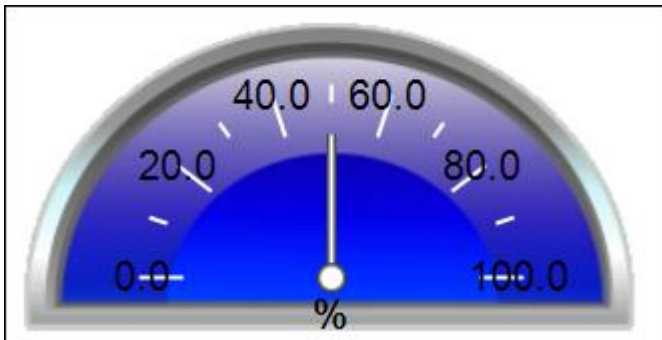


Abb. 83: Zeigerinstrument 180°

### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungseditor werden folgende Benutzereingaben zur Verfügung gestellt:

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Zeigerinstrument 180°
„ <i>Wert</i> “	Zuzuweisen ist eine Variable, deren Wert als <b>Ausschlag</b> der Instrumentennadel dargestellt wird.

#### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„ <i>X</i> “	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„ <i>Y</i> “	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„ <i>Breite</i> “	Horizontale Breite des Elements.

„Höhe“ Vertikale Höhe des Elements.

## Hintergrund

Eigenschaft	Beschreibung Wert
„Hintergrundbild“	Auszuwählen ist eine Farbe für den Balken: <ul style="list-style-type: none"><li>▪ Gelb</li><li>▪ Rot</li><li>▪ Grün</li><li>▪ Blau</li><li>▪ Grau</li></ul>

## Zeiger

Eigenschaft	Beschreibung Wert
„Zeigertyp“	Auszuwählen ist aus: <ul style="list-style-type: none"><li>▪ Normaler Zeiger</li><li>▪ Dünner Zeiger</li><li>▪ Breiter Zeiger</li><li>▪ Dünne Nadel</li><li>▪ Dünner 3D-Zeiger</li><li>▪ Dünner 3D-Nadel</li></ul>
„Farbe“	Farbe, mit der der Zeiger angezeigt wird.
„Winkelbereich“	Auswahl eines 180°-Ausschnitts: <ul style="list-style-type: none"><li>▪ Oben</li><li>▪ Unten</li><li>▪ Links</li><li>▪ Rechts</li></ul>
„Zusätzlicher Zeiger“	Ist die Option aktiviert, ist ein zusätzlicher Zeiger auf der Skala gegenüber der Nadel sichtbar.

## Skala

Eigenschaft	Beschreibung Wert
„Unterskalaposition“	Die Unterskala kann am äußeren oder inneren Radius des Skalenrings angezeigt werden: <ul style="list-style-type: none"><li>▪ Außen</li><li>▪ Innen</li></ul>
„Skalentyp“	Die Skala kann angezeigt werden als: <ul style="list-style-type: none"><li>▪ Linien</li></ul>


- Punkte
- Quadrate

„Skalenanfang“ „	Wert, der die Skalierungsanzeige von unten begrenzt.
„Skalenende“	Wert, der die Skalierungsanzeige von oben begrenzt.
„Hauptskala“	Abstand zwischen zwei Strichen der Grobskala.
„Unterskala“	Abstand zwischen zwei Strichen der Feinskala; kann auf 0 gesetzt werden, falls eine weitere Unterteilung der Grobskala nicht gewünscht ist.
„Skalenlinienstärke“	Stärke der Skala in Pixel
„Skalenfarbe“	Die Farbe kann über Selektionsliste bzw. über Schaltfläche <input type="button" value="..."/> aus dem Standarddialog zur Farbauswahl eingestellt werden.
„Skala in 3D“	Ist die Option aktiviert, wird die Skala dreidimensional dargestellt.
„Skala anzeigen“	Wird keine Skalendarstellung gewünscht, ist diese Option zu deaktivieren.
„Innerer Rahmen“	Ist diese Option aktiviert, wird der Skalenring innen mit einem Rahmen versehen. Standardmäßig deaktiviert.
„Äußerer Rahmen“	Ist diese Option aktiviert, wird der Skalenring außen mit einem Rahmen versehen. Standardmäßig deaktiviert.

## Beschriftung

Eigenschaft	Beschreibung Wert
„Beschriftung“	<p>Auswählen aus</p> <ul style="list-style-type: none"> <li>▪ Außen: Skalenwerte werden auf der Außenseite der Skala platziert.</li> <li>▪ Innen: Skalenwerte werden auf der Innenseite der Skala platziert.</li> </ul>
„Einheit“	String, der unterhalb des Pfeils (M) bzw. unterhalb des Mittelpunkts der Skalierung (BD) angezeigt wird (gedacht für die Eingabe der Skalierungseinheit)
„Schriftart“	<p>Schriftart für Darstellung von Einheit und Skalenbereich</p> <p>Festzulegen ist die Schriftart durch auswählen von einem aus:</p> <ul style="list-style-type: none"> <li>▪ Standard</li> <li>▪ Überschrift</li> <li>▪ Große</li> <li>▪ Titel</li> <li>▪ Anmerkung</li> </ul> <p>Mit Klick auf <input type="button" value="..."/> öffnet ein Dialog für eine benutzerdefinierte Einstellung der Schriftart.</p>
„Skalenformat (C-	Hier ist die C-Syntax zur Angabe der Formatierung der Skalierungsbeschriftung zu



- Syntax*“ benutzen. Z.B. führt die Eingabe des Strings "%3.2f s" in diesem Feld zur Anzeige der Skalierungsbeschriftungen mit 3 Stellen, davon 2 Nachkommastellen, gefolgt vom Buchstaben "s"
- „*Max. Breite der Beschriftungen*“ Wert, der die maximale Breite der Skalierungsbeschriftung vorgibt; in der Regel wird dieser Wert automatisch richtig festgesetzt. Benützen Sie diese Vorgabe nur, falls die automatische Anpassung nicht zum gewünschten Ergebnis führt.
- „*Höhe der Beschriftungen*“ Wert, der die Höhe der Skalierungsbeschriftung vorgibt; in der Regel wird dieser Wert automatisch richtig festgesetzt. Benützen Sie diese Vorgabe nur, falls die automatische Anpassung nicht zum gewünschten Ergebnis führt.
- „*Farbe Schriftart*“ Farbe für die Schrift kann über Selektionsliste bzw. über Schaltfläche  aus dem Standarddialog zur Farbauswahl eingestellt werden.

## Positionierung


Eigenschaft	Beschreibung Wert
„ <i>Zeigerabstand</i> “	Hier kann die Länge des Zeigers in Pixel angegeben werden.
„ <i>Beschriftungsverschiebung</i> “	Hier kann ein Abstand in Pixel definiert werden um die Beschriftung zu positionieren.
„ <i>Einheitenverschiebung</i> “	Hier kann ein vertikaler Abstand eingegeben werden, um den Text eingegeben in <i>Beschriftung</i> → <i>Einheit</i> zu positionieren.

## Farben

Property	Value Description
„ <i>Farbbereiche</i> “	„ <i>Dauerhafte Farbbereiche</i> “: Dessen Auswirkung ist nur im Online-Mode sichtbar. Ist die Option aktiviert, sind die Farbbereiche dauerhaft sichtbar. Ist die Option nicht aktiviert, ist nur der Farbbereich, auf den der Pfeil zeigt, sichtbar.
▪ „ <i>Dauerhafte Farbbereiche</i> “	
▪ „ <i>[&lt;n&gt;]</i> “	„ <i>Neu erstellen</i> “: Mit Klick auf diesen Button wird ein weiterer Farbbereich erzeugt und angezeigt.

## Farben: Farbbereiche: Bereiche: [<n>]

Anzahl der Farbbereiche in aufsteigender, nicht editierbarer Reihenfolge.

„*Löschen*“: Ein Klick auf den Button  wird den Farbbereich von der Liste entfernen und die Reihenfolge neu arrangieren.

Eigenschaft	Beschreibung Wert
„ <i>Bereichsanfang</i> “	Wert, der den Farbbereich von unten begrenzt.
„ <i>Bereichsende</i> “	Wert, der den Farbbereich von oben begrenzt.
„ <i>Farbe</i> “	Farbe, in der der Farbbereich dargestellt werden soll.

Eigenschaft	Beschreibung Wert
-------------	-------------------

„Zugriffsrechte“ Mit Klick öffnet [↘ „Zugriffsrechte“](#).

Statusmeldungen:

- Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.
- Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.

## Zeigerinstrument 360°

Mit „*Zeigerinstrument*“ kann zum Beispiel ein Drehzahlmesser der Visualisierung hinzugefügt werden. Die Nadel positioniert sich nach dem Wert der zugewiesenen Variablen. Es hat ein vorkonfiguriertes Design, in dem eine Hintergrundfarbe gesetzt werden kann.

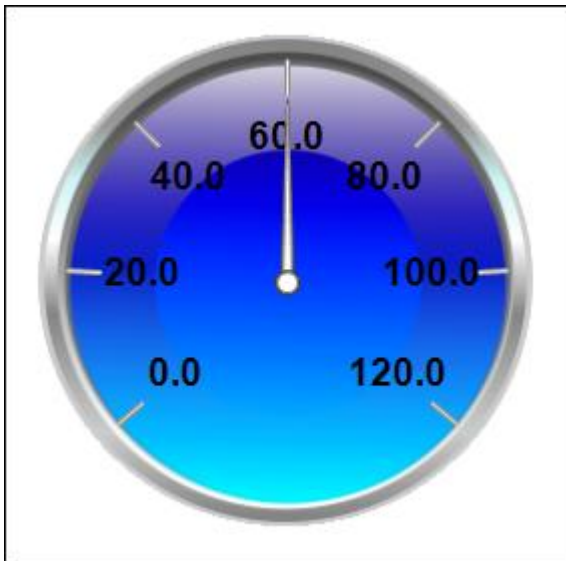


Abb. 84: Zeigerinstrument 360°

### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungseditor werden folgende Benutzereingaben zur Verfügung gestellt:

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemlnst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Zeigerinstrument 360°
„ <i>Wert</i> “	Zuzuweisen ist eine Variable, deren Wert als <b>Ausschlag</b> der Instrumentennadel dargestellt wird.

#### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
-------------	-------------------

- „X“ Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
- „Y“ Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
- „Breite“ Horizontale Breite des Elements.
- „Höhe“ Vertikale Höhe des Elements.


## Hintergrund

Eigenschaft	Beschreibung Wert
„Hintergrundbild“	Auszuwählen ist eine Farbe für den Balken: <ul style="list-style-type: none"> <li>▪ Gelb</li> <li>▪ Rot</li> <li>▪ Grün</li> <li>▪ Blau</li> <li>▪ Grau</li> </ul>

## Zeiger

Eigenschaft	Beschreibung Wert
„Zeigertyp“	Auszuwählen ist aus: <ul style="list-style-type: none"> <li>▪ Normaler Zeiger</li> <li>▪ Dünner Zeiger</li> <li>▪ Breiter Zeiger</li> <li>▪ Dünne Nadel</li> <li>▪ Dünner 3D-Zeiger</li> <li>▪ Dünner 3D-Nadel</li> </ul>
„Farbe“	Farbe, mit der der Zeiger angezeigt wird.
„Zeiger Anfang“	Der hier anzugebenen Wert wird als Winkel in Grad interpretiert, der sich vom Nullpunkt der Skala bis zum Anfang des Skalenbereichs im Gegenuhrzeigersinn aufspannt. Nullpunkt der Skala liegt bei "3 Uhr". Der Winkel hat einen Wertebereich von 0° bis 360°. Zentrum der Drehung ist die X/Y-Position.
„Zeiger Ende“	Der hier anzugebenen Wert wird als Winkel in Grad interpretiert, der sich vom Nullpunkt der Skala bis zum Ende des Skalenbereichs im Gegenuhrzeigersinn aufspannt. Nullpunkt der Skala liegt bei "3 Uhr". Eingabe eines negativen Werts ist erlaubt. Zentrum der Drehung ist die X/Y-Position. „Zeiger Ende“ muss so gesetzt werden, dass es im Uhrzeigersinn betrachtet rechts von „Zeiger Anfang“ zu liegen kommt. Der Winkel hat einen Wertebereich von 0° bis maximal 350°.
„Zusätzlicher Zeiger“	Ist die Option aktiviert, ist ein zusätzlicher Zeiger auf der Skala gegenüber der Nadel sichtbar.

## Skala

Eigenschaft	Beschreibung Wert
„Unterskalaposition“	Die Unterskala kann am äußeren oder inneren Radius des Skalenrings angezeigt werden: <ul style="list-style-type: none"> <li>▪ Außen</li> <li>▪ Innen</li> </ul>
„Skalentyp“	Die Skala kann angezeigt werden als: <ul style="list-style-type: none"> <li>▪ Linien</li> <li>▪ Punkte</li> <li>▪ Quadrate</li> </ul>
„Skalenanfang“ „	Wert, der die Skalierungsanzeige von unten begrenzt.
„Skalenende“	Wert, der die Skalierungsanzeige von oben begrenzt.
„Hauptskala“	Abstand zwischen zwei Strichen der Grobskala.
„Unterskala“	Abstand zwischen zwei Strichen der Feinskala; kann auf 0 gesetzt werden, falls eine weitere Unterteilung der Grobskala nicht gewünscht ist.
„Skalenlinienstärke“	Stärke der Skala in Pixel
„Skalenfarbe“	Die Farbe kann über Selektionsliste bzw. über Schaltfläche  aus dem Standarddialog zur Farbauswahl eingestellt werden.
„Skala in 3D“	Ist die Option aktiviert, wird die Skala dreidimensional dargestellt.
„Skala anzeigen“	Wird keine Skalendarstellung gewünscht, ist diese Option zu deaktivieren.
„Innerer Rahmen“	Ist diese Option aktiviert, wird der Skalenring innen mit einem Rahmen versehen. Standardmäßig deaktiviert.
„Äußerer Rahmen“	Ist diese Option aktiviert, wird der Skalenring außen mit einem Rahmen versehen. Standardmäßig deaktiviert.

## Beschriftung

Eigenschaft	Beschreibung Wert
„Beschriftung“	Auswählen aus <ul style="list-style-type: none"> <li>▪ Außen: Skalenwerte werden auf der Außenseite der Skala platziert.</li> <li>▪ Innen: Skalenwerte werden auf der Innenseite der Skala platziert.</li> </ul>
„Einheit“	String, der unterhalb des Pfeils (M) bzw. unterhalb des Mittelpunkts der Skalierung (BD) angezeigt wird (gedacht für die Eingabe der Skalierungseinheit)
„Schriftart“	Schriftart für Darstellung von Einheit und Skalenbereich

Festzulegen ist die Schriftart durch auswählen von einem aus:

- Standard
- Überschrift
- Große
- Titel
- Anmerkung

Mit Klick auf  öffnet ein Dialog für eine benutzerdefinierte Einstellung der Schriftart.

*„Skalenformat (C-Syntax)“*

Hier ist die C-Syntax zur Angabe der Formatierung der Skalierungsbeschriftung zu benutzen. Z.B. führt die Eingabe des Strings "%3.2f s" in diesem Feld zur Anzeige der Skalierungsbeschriftungen mit 3 Stellen, davon 2 Nachkommastellen, gefolgt vom Buchstaben "s"

*„Max. Breite der Beschriftungen“*

Wert, der die maximale Breite der Skalierungsbeschriftung vorgibt; in der Regel wird dieser Wert automatisch richtig festgesetzt. Benützen Sie diese Vorgabe nur, falls die automatische Anpassung nicht zum gewünschten Ergebnis führt.

*„Höhe der Beschriftungen“*

Wert, der die Höhe der Skalierungsbeschriftung vorgibt; in der Regel wird dieser Wert automatisch richtig festgesetzt. Benützen Sie diese Vorgabe nur, falls die automatische Anpassung nicht zum gewünschten Ergebnis führt.

*„Farbe Schriftart“*

Farbe für die Schrift kann über Selektionsliste bzw. über Schaltfläche  aus dem Standarddialog zur Farbauswahl eingestellt werden.

**Positionierung**


Eigenschaft	Beschreibung Wert
<i>„Zeigerabstand“</i>	Hier kann die Länge des Zeigers in Pixel angegeben werden.
<i>„Beschriftungsverschiebung“</i>	Hier kann ein Abstand in Pixel definiert werden um die Beschriftung zu positionieren.
<i>„Einheitenverschiebung“</i>	Hier kann ein vertikaler Abstand eingegeben werden, um den Text eingegeben in <i>Beschriftung → Einheit</i> zu positionieren.
<i>„Ursprungsverschiebung“</i>	Dieser Wert definiert den Versatz des Elements. Er kann verwendet werden um eine exakte Positionierung relativ zum Hintergrundbild zu erreichen.

**Farben**

Property	Value Description
<i>„Farbbereiche“</i>	<i>„Dauerhafte Farbbereiche“</i> : Dessen Auswirkung ist nur im Online-Mode sichtbar. Ist die Option aktiviert, sind die Farbbereiche dauerhaft sichtbar. Ist die Option nicht aktiviert, ist nur der Farbbereich, auf den der Pfeil zeigt, sichtbar.
▪ <i>„Dauerhafte Farbbereiche“</i>	
▪ <i>„[&lt;n&gt;]“</i>	<i>„Neu erstellen“</i> : Mit Klick auf diesen Button wird ein weiterer Farbbereich erzeugt und angezeigt.

## Farben: Farbbereiche: Bereiche: [<n>]

Anzahl der Farbbereiche in aufsteigender, nicht editierbarer Reihenfolge.

„Löschen“: Ein Klick auf den Button  wird den Farbbereich von der Liste entfernen und die Reihenfolge neu arrangieren.

Eigenschaft	Beschreibung Wert
„Bereichsanfang“	Wert, der den Farbbereich von unten begrenzt.
„Bereichsende“	Wert, der den Farbbereich von oben begrenzt.
„Farbe“	Farbe, in der der Farbbereich dargestellt werden soll.

Eigenschaft	Beschreibung Wert
„Zugriffsrechte“	Mit Klick öffnet <a href="#">„Zugriffsrechte“</a> .

Statusmeldungen:

- Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.
- Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.

## Zeigerinstrument Frei Konfigurierbar

Mit „*Zeigerinstrument*“ kann zum Beispiel ein Drehzahlmesser der Visualisierung hinzugefügt werden. Die Nadel positioniert sich nach dem Wert der zugewiesenen Variablen. Es hat ein vorkonfiguriertes Design, in dem eine Hintergrundfarbe gesetzt werden kann.

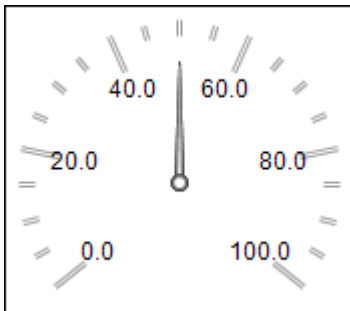


Abb. 85: Zeigerinstrument frei konfigurierbar

### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungseditor werden folgende Benutzereingaben zur Verfügung gestellt:

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Zeigerinstrument frei konfigurierbar
„ <i>Wert</i> “	Zuzuweisen ist eine Variable, deren Wert als <b>Ausschlag</b> der Instrumentennadel dargestellt wird.

#### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„ <i>X</i> “	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„ <i>Y</i> “	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„ <i>Breite</i> “	Horizontale Breite des Elements.
„ <i>Höhe</i> “	Vertikale Höhe des Elements.



## Hintergrund

Eigenschaft	Beschreibung Wert
<i>„Hintergrundbild“</i>	Zuzuweisen ist ein Bild aus dem GlobellImagePool indem entweder der Name der Bilddatei oder die Bild-ID angegeben wird.
<i>„Transparenzfarbe“</i>	Für Bilder mit transparentem Hintergrund kann eine Farbe angegeben werden.

## Zeiger

Eigenschaft	Beschreibung Wert
<i>„Zeigertyp“</i>	Auszuwählen ist aus: <ul style="list-style-type: none"><li>▪ Normaler Zeiger</li><li>▪ Dünner Zeiger</li><li>▪ Breiter Zeiger</li><li>▪ Dünne Nadel</li><li>▪ Dünner 3D-Zeiger</li><li>▪ Dünner 3D-Nadel</li></ul>
<i>„Farbe“</i>	Farbe, mit der der Zeiger angezeigt wird.
<i>„Zeiger Anfang“</i>	Der hier anzugebenen Wert wird als Winkel in Grad interpretiert, der sich vom Nullpunkt der Skala bis zum Anfang des Skalenbereichs im Gegenuhrzeigersinn aufspannt. Nullpunkt der Skala liegt bei "3 Uhr". Der Winkel hat einen Wertebereich von 0° bis 360°. Zentrum der Drehung ist die X/Y-Position.
<i>„Zeiger Ende“</i>	Der hier anzugebenen Wert wird als Winkel in Grad interpretiert, der sich vom Nullpunkt der Skala bis zum Ende des Skalenbereichs im Gegenuhrzeigersinn aufspannt. Nullpunkt der Skala liegt bei "3 Uhr". Eingabe eines negativen Werts ist erlaubt. Zentrum der Drehung ist die X/Y-Position. <i>„Zeiger Ende“</i> muss so gesetzt werden, dass es im Uhrzeigersinn betrachtet rechts von <i>„Zeiger Anfang“</i> zu liegen kommt. Der Winkel hat einen Wertebereich von 0° bis maximal 350°.
<i>„Zusätzlicher Zeiger“</i>	Ist die Option aktiviert, ist ein zusätzlicher Zeiger auf der Skala gegenüber der Nadel sichtbar.

## Skala

Eigenschaft	Beschreibung Wert
<i>„Unterskalaposition“</i>	Die Unterskala kann am äußeren oder inneren Radius des Skalenrings angezeigt werden: <ul style="list-style-type: none"><li>▪ Außen</li><li>▪ Innen</li></ul>
<i>„Skalentyp“</i>	Die Skala kann angezeigt werden als: <ul style="list-style-type: none"><li>▪ Linien</li><li>▪ Punkte</li></ul>

- Quadrate

„Skalenanfang“ „	Wert, der die Skalierungsanzeige von unten begrenzt.
„Skalenende“	Wert, der die Skalierungsanzeige von oben begrenzt.
„Hauptskala“	Abstand zwischen zwei Strichen der Grobskala.
„Unterskala“	Abstand zwischen zwei Strichen der Feinskala; kann auf 0 gesetzt werden, falls eine weitere Unterteilung der Grobskala nicht gewünscht ist.
„Skalenlinienstärke“	Stärke der Skala in Pixel
„Skalenfarbe“	Die Farbe kann über Selektionsliste bzw. über Schaltfläche <input type="button" value="..."/> aus dem Standarddialog zur Farbauswahl eingestellt werden.
„Skala in 3D“	Ist die Option aktiviert, wird die Skala dreidimensional dargestellt.
„Skala anzeigen“	Wird keine Skalendarstellung gewünscht, ist diese Option zu deaktivieren.
„Innerer Rahmen“	Ist diese Option aktiviert, wird der Skalenring innen mit einem Rahmen versehen. Standardmäßig deaktiviert.
„Äußerer Rahmen“	Ist diese Option aktiviert, wird der Skalenring außen mit einem Rahmen versehen. Standardmäßig deaktiviert.


## Beschriftung

Eigenschaft	Beschreibung Wert
„Beschriftung“	<p>Auswählen aus</p> <ul style="list-style-type: none"> <li>▪ Außen: Skalenwerte werden auf der Außenseite der Skala platziert.</li> <li>▪ Innen: Skalenwerte werden auf der Innenseite der Skala platziert.</li> </ul>
„Einheit“	String, der unterhalb des Pfeils (M) bzw. unterhalb des Mittelpunkts der Skalierung (BD) angezeigt wird (gedacht für die Eingabe der Skalierungseinheit)
„Schriftart“	<p>Schriftart für Darstellung von Einheit und Skalenbereich</p> <p>Festzulegen ist die Schriftart durch auswählen von einem aus:</p> <ul style="list-style-type: none"> <li>▪ Standard</li> <li>▪ Überschrift</li> <li>▪ Große</li> <li>▪ Titel</li> <li>▪ Anmerkung</li> </ul> <p>Mit Klick auf <input type="button" value="..."/> öffnet ein Dialog für eine benutzerdefinierte Einstellung der Schriftart.</p>
„Skalenformat (C-Syntax)“	Hier ist die C-Syntax zur Angabe der Formatierung der Skalierungsbeschriftung zu benutzen. Z.B. führt die Eingabe des Strings "%3.2f s" in diesem Feld zur Anzeige der

Skalierungsbeschriftungen mit 3 Stellen, davon 2 Nachkommastellen, gefolgt vom Buchstaben "s"

„Max. Breite der Beschriftungen“ Wert, der die maximale Breite der Skalierungsbeschriftung vorgibt; in der Regel wird dieser Wert automatisch richtig festgesetzt. Benützen Sie diese Vorgabe nur, falls die automatische Anpassung nicht zum gewünschten Ergebnis führt.

„Höhe der Beschriftungen“ Wert, der die Höhe der Skalierungsbeschriftung vorgibt; in der Regel wird dieser Wert automatisch richtig festgesetzt. Benützen Sie diese Vorgabe nur, falls die automatische Anpassung nicht zum gewünschten Ergebnis führt.

„Farbe Schriftart“ Farbe für die Schrift kann über Selektionsliste bzw. über Schaltfläche  aus dem Standarddialog zur Farbauswahl eingestellt werden.

### Positionierung


Eigenschaft	Beschreibung Wert
„Zeigerabstand“	Hier kann die Länge des Zeigers in Pixel angegeben werden.
„Skalenabstand“	Hier kann ein Abstand der Skalenstriche zum Zentrum in Pixel angegeben werden.
„Skalenlänge“	Hier kann die Länge der Skalenstriche in Pixel angegeben werden.
„Beschriftungsverschiebung“	Hier kann ein Abstand in Pixel definiert werden um die Beschriftung zu positionieren.
„Einheitenverschiebung“	Hier kann ein vertikaler Abstand eingegeben werden, um den Text eingegeben in <i>Beschriftung → Einheit</i> zu positionieren.
„Ursprungsverschiebung“	Dieser Wert definiert den Versatz des Elements. Er kann verwendet werden um eine exakte Positionierung relativ zum Hintergrundbild zu erreichen.

### Farben

Property	Value Description
„Farbbereiche“	„Dauerhafte Farbbereiche“: Dessen Auswirkung ist nur im Online-Mode sichtbar. Ist die Option aktiviert, sind die Farbbereiche dauerhaft sichtbar. Ist die Option nicht aktiviert, ist nur der Farbbereich, auf den der Pfeil zeigt, sichtbar.
▪ „Dauerhafte Farbbereiche“	
▪ „[<n>]“	„Neu erstellen“: Mit Klick auf diesen Button wird ein weiterer Farbbereich erzeugt und angezeigt.

Farben: Farbbereiche: Bereiche: [<n>]

Anzahl der Farbbereiche in aufsteigender, nicht editerbarer Reihenfolge.

„Löschen“: Ein Klick auf den Button  wird den Farbbereich von der Liste entfernen und die Reihenfolge neu arrangieren.

Eigenschaft	Beschreibung Wert
„Bereichsanfang“	Wert, der den Farbbereich von unten begrenzt.
„Bereichsende“	Wert, der den Farbbereich von oben begrenzt.
„Farbe“	Farbe, in der der Farbbereich dargestellt werden soll.

Eigenschaft	Beschreibung Wert
-------------	-------------------

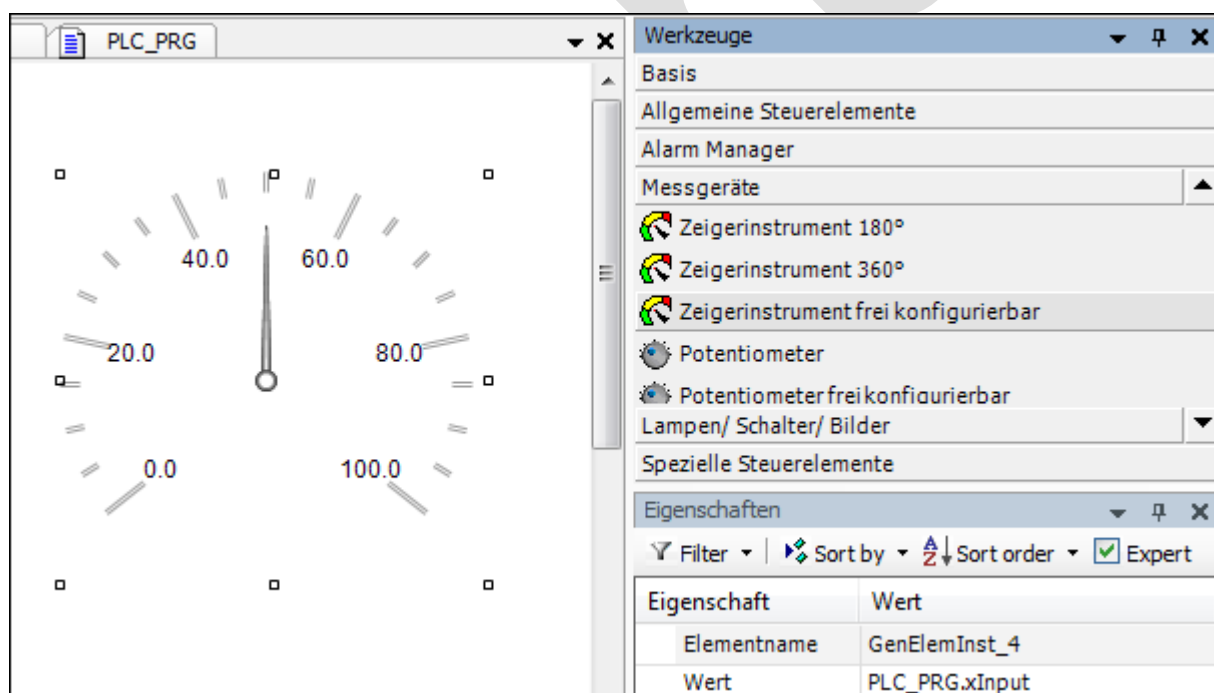
„Zugriffsrechte“	Mit Klick öffnet <a href="#">„Zugriffsrechte“</a> .
------------------	---

Statusmeldungen:


- Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.
- Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.

### Konfiguration der Eigenschaften eines frei konfigurierbaren Zeigerinstrument

Sehen Sie im Folgenden ein Beispiel für das Einfügen und die Konfiguration von „*Zeigerinstrument*“.

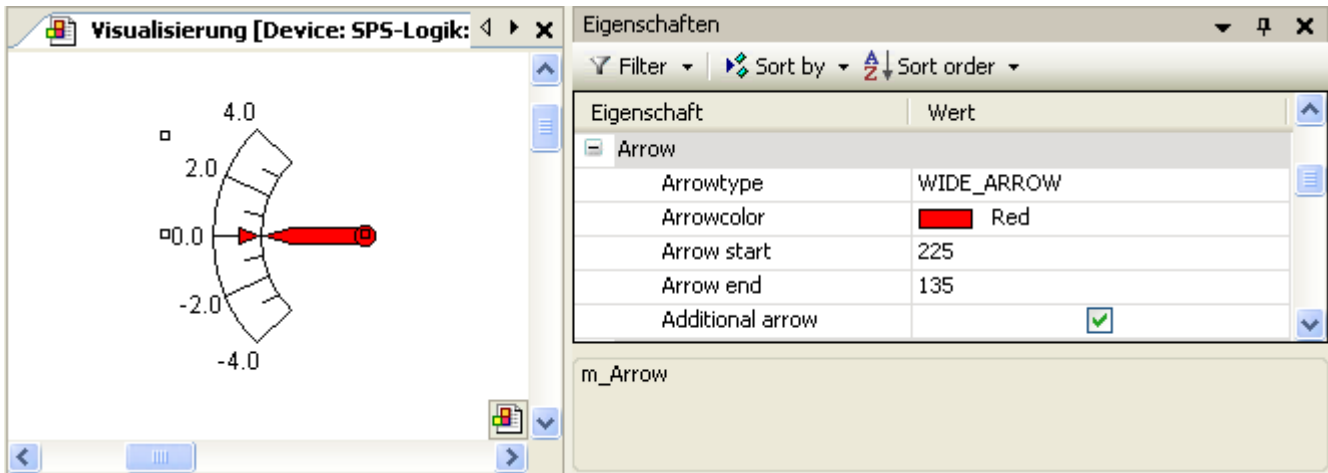


▼ Abb. 86: Beispiel - Einfügen eines Zeigerinstrument und Verknüpfung mit Eingangsvariablen

Die zu verknüpfende Eingangsvariable - im Beispiel eine Variable namens `xInput`- muss innerhalb der Elementeigenschaften des Zeigerinstrument-Elements angegeben werden; nach einem Klick in das Eingabefeld steht der Schalter  welcher betätigt werden kann um das Projekt nach der Eingangsvariable zu durchsuchen.

Achten Sie darauf die Eingangsvariable durch Angabe ihres vollständigen Pfads im Gerätebaum zu spezifizieren.

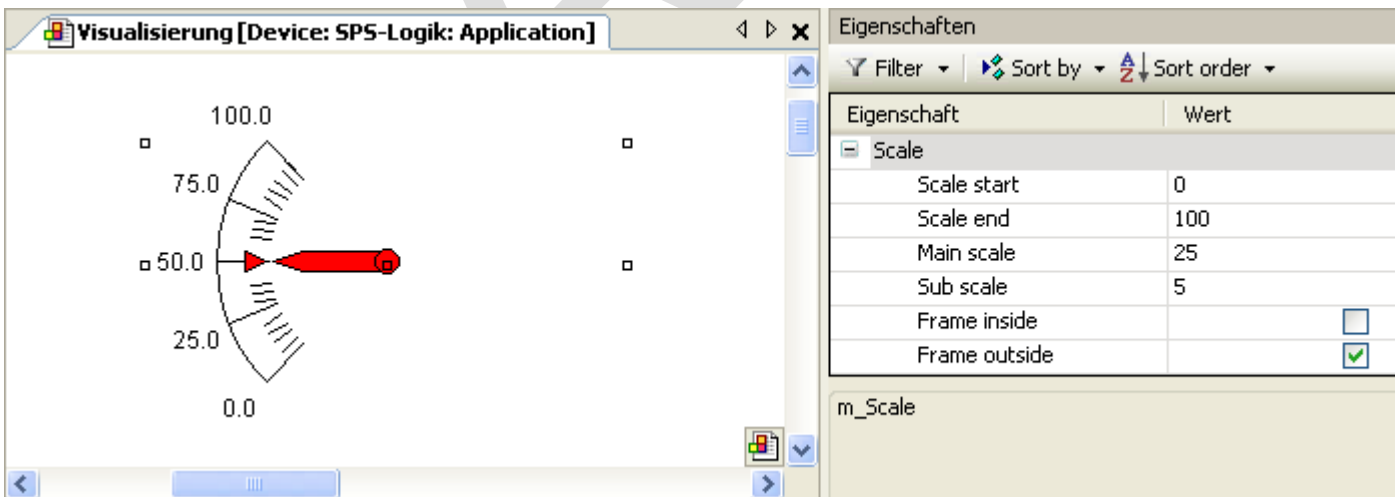
Die Orientierung und Größe der Skalenanzeige wie auch Farbe und Aussehen des Pfeil kann im Abschnitt „Zeiger“ der Elementeigenschaften festgelegt werden.



▼ Abb. 87: Beispiel - Konfiguration innerhalb des Abschnitts "Zeiger" in den Elementeigenschaften

Die Zahl im Eingabefeld „Zeiger Anfang“, „Zeiger Ende“ gibt den Winkel (in Grad) an, dem der linke/rechte Rand der Skala mit der horizontalen Gerade einschließt. Dieser Winkel ist mathematisch, d.h. entgegen dem Uhrzeigersinn orientiert, so dass der Wert im Feld „Zeiger Anfang“ immer größer sein muss als der Wert von „Zeiger Ende“. Das Paar aus Anfangs- und Endwert ist periodisch mit 360 Grad.

Im Abschnitt „Skale“ kann der Zahlenbereich für die Skala und eine Grob-/ und Feinskalierung bestimmt werden.

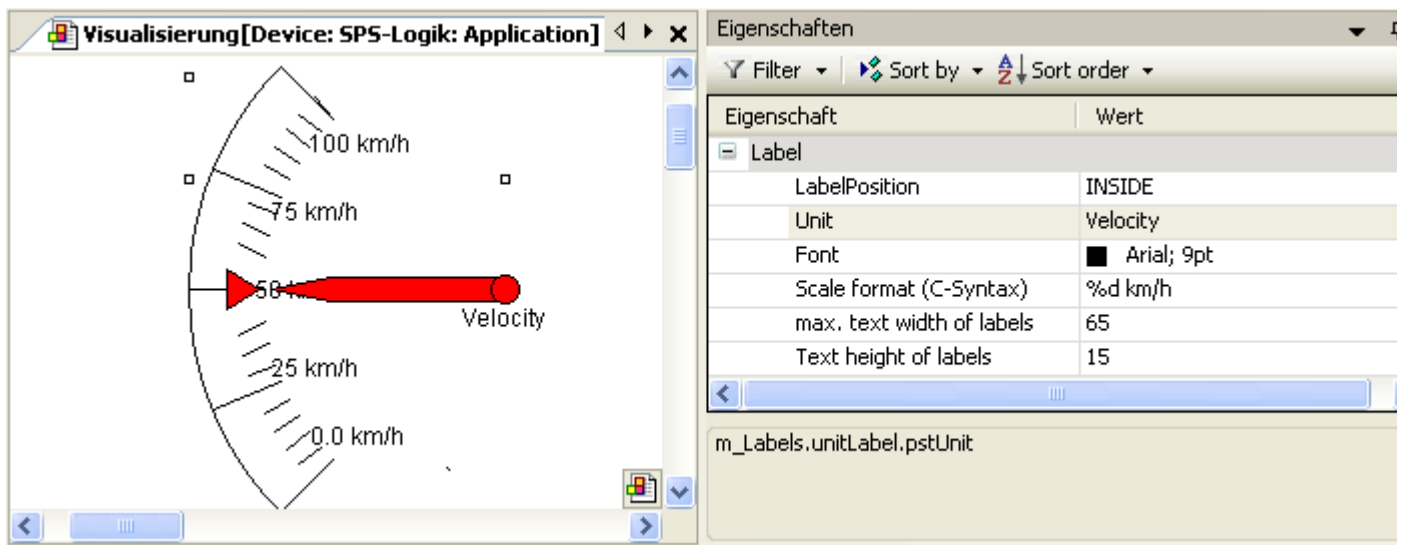


▼ Abb. 88: Beispiel - Konfiguration innerhalb des Abschnitts "Zeiger" in den Elementeigenschaften

Der Wert von „Skalenanfang“ wird am linken Rand der Skala eingetragen. Daher muss er kleiner sein als der Wert von „Skalenende“, der am rechten Rand der Skala eingefügt wird. Im Gegensatz zu der groben Skalierung kann die feinere Unterteilung („Unterskala“) weggelassen werden, indem man den Wert für ihre Abstände auf 0 setzt. In diesem Fall würden also keine feinen Skalenstriche angezeigt werden. Da das Kontrollkästchen „Innerer Rahmen“


im Beispiel deaktiviert wurde, ist der innere Kreisbogen der Skala ausgeblendet.

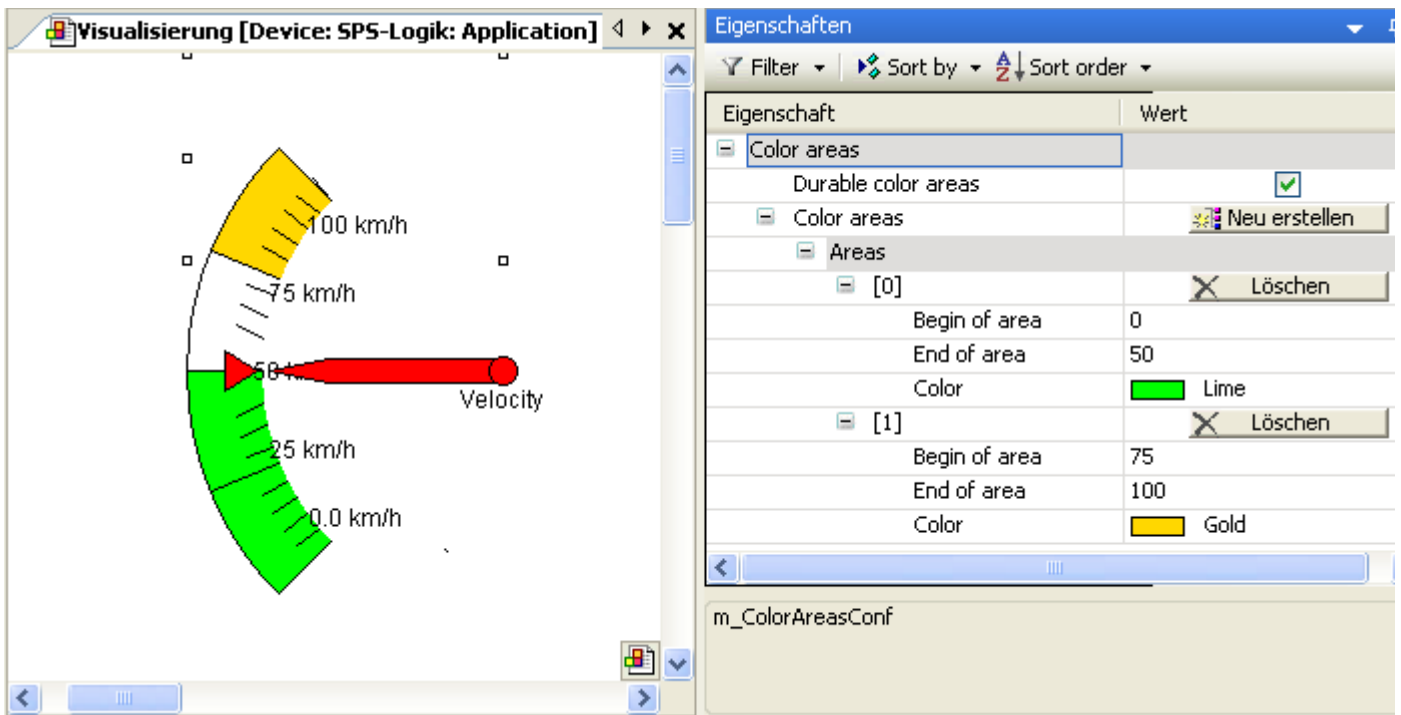
Nachdem die Skalierung festgesetzt wurde, soll nun im Abschnitt "Label" die Beschriftung der Skala formatiert werden.



▼ Abb. 89: Beispiel - Konfiguration innerhalb des Abschnitts "Label" in den Elementeigenschaften

Durch Abänderung der *Beschriftung* → *Beschriftung* von Außen in Innen verlagern wir die Anzeige der Skalierungsmarken in das Kreisinnere. Der Eintrag im Feld „*Einheit*“ erscheint unterhalb des Fußpunkt des Pfeils. Nach Auswahl einer geeigneten Schrift(farbe), passen wir die Formatierung der Skalierungsmarken an: der Zahlenwert der Skalierung ist gemäß der Syntax der Programmiersprache C zu formatieren (benutzen Sie %d für ganze Zahlen und %.Xf für Gleitkommazahlen, wobei X zu ersetzen ist durch die gewünschte Anzahl an Nachkommastellen). Die Werte in den folgenden beiden Eingabefeldern werden entsprechend Ihrer vorhergehenden Einstellungen in diesem Abschnitt eingetragen. Sie müssen diese Werte nur dann verändern, falls die automatische Anpassung nicht zum gewünschten Ergebnis führt.

Schließlich können im Abschnitt „*Farben*“ bestimmte Bereiche der Skala noch eingefärbt werden indem man „*Farbbereiche*“ anlegt mit Hilfe der Taste  **Neu erstellen**. Die Farbbereiche werden in aufsteigender Ordnung durchnummeriert. Jeder Farbbereich wird innerhalb der Elementeigenschaften mit eigenen Eingabefeldern ausgestattet.

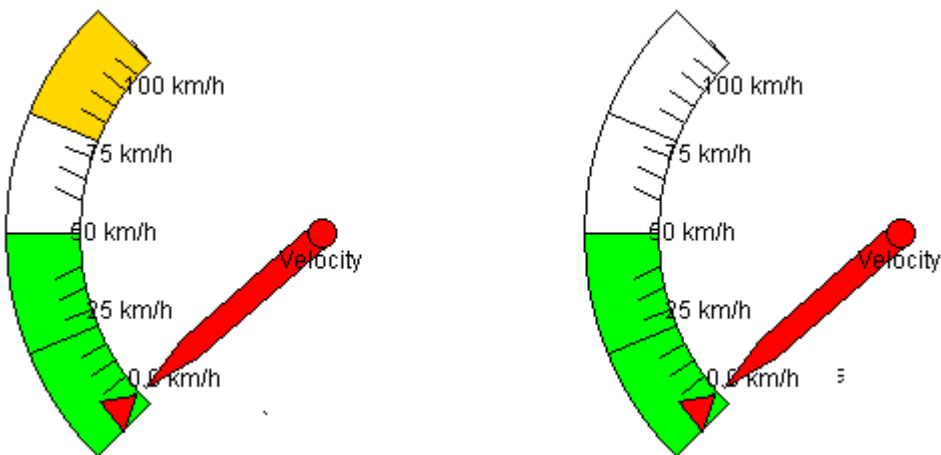


▼ Abb. 90: Beispiel - Konfiguration innerhalb des Abschnitts "Farben - Farbbereiche" in den Elementeigenschaften

Die Felder „*Bereichsanfang*“ und „*Bereichsende*“ stehen zur Spezifizierung des Unterbereichs der Skalierung zur Verfügung. Die Einfärbung kann aus einem Aufklappmenü ausgewählt werden und mit Hilfe des Schalters



Der Effekt des Kontrollkästchens „*Dauerhafte Farbbereiche*“ ist nur im Online-Modus ersichtlich:



▼ Abb. 91: Beispiel: Zeigerinstrument-Element im Online-Modus

Während das rechte Zeigerinstrument-Element nur den Farbbereich anzeigt, der den Pfeil enthält, enthält das linke Element alle angelegten Farbbereiche, da das Kontrollkästchen „*Dauerhafte Farbbereiche*“ aktiviert wurde.

## Potentiometer

Hier kann einer Visualisierung ein Potentiometer mit vordefiniertem Design hinzugefügt werden. Solch ein Bedienelement zeigt mittels Pfeil/Zeiger einen Variablenwert an, der über Benutzereingabe verschoben werden kann.



Abb. 92: Potentiometer

### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungseditor werden folgende Benutzereingaben zur Verfügung gestellt:

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Potentiometer
„ <i>Variable</i> “	Zuzuweisen ist eine numerische Variable, die die Position des Potentiometers enthält.

#### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„ <i>X</i> “	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„ <i>Y</i> “	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„ <i>Breite</i> “	Horizontale Breite des Elements.



„Höhe“ Vertikale Höhe des Elements.

## Hintergrund

Eigenschaft	Beschreibung Wert
„Hintergrundbild“	Zuzuweisen ist ein Bild aus dem GlobellImagePool indem entweder der Name der Bilddatei oder die Bild-ID angegeben wird.
„Transparenzfarbe“	Für Bilder mit transparentem Hintergrund kann eine Farbe angegeben werden.

## Zeiger

Eigenschaft	Beschreibung Wert
„Zeigertyp“	Auszuwählen ist aus: <ul style="list-style-type: none"><li>▪ Normaler Zeiger</li><li>▪ Dünner Zeiger</li><li>▪ Breiter Zeiger</li><li>▪ Dünne Nadel</li><li>▪ Dünner 3D-Zeiger</li><li>▪ Dünner 3D-Nadel</li></ul>
„Farbe“	Farbe, mit der der Zeiger angezeigt wird.
„Zeiger Anfang“	Der hier anzugebenen Wert wird als Winkel in Grad interpretiert, der sich vom Nullpunkt der Skala bis zum Anfang des Skalenbereichs im Gegenuhrzeigersinn aufspannt. Nullpunkt der Skala liegt bei "3 Uhr". Der Winkel hat einen Wertebereich von 0° bis 360°. Zentrum der Drehung ist die X/Y-Position.
„Zeiger Ende“	Der hier anzugebenen Wert wird als Winkel in Grad interpretiert, der sich vom Nullpunkt der Skala bis zum Ende des Skalenbereichs im Gegenuhrzeigersinn aufspannt. Nullpunkt der Skala liegt bei "3 Uhr". Eingabe eines negativen Werts ist erlaubt. Zentrum der Drehung ist die X/Y-Position. „Zeiger Ende“ muss so gesetzt werden, dass es im Uhrzeigersinn betrachtet rechts von „Zeiger Anfang“ zu liegen kommt. Der Winkel hat einen Wertebereich von 0° bis maximal 350°.

## Skala

Eigenschaft	Beschreibung Wert
„Unterskalaposition“	Die Unterskala kann am äußeren oder inneren Radius des Skalenrings angezeigt werden: <ul style="list-style-type: none"><li>▪ Außen</li><li>▪ Innen</li></ul>
„Skalentyp“	Die Skala kann angezeigt werden als: <ul style="list-style-type: none"><li>▪ Linien</li><li>▪ Punkte</li></ul>

- Quadrate

„Skalenanfang“ „	Wert, der die Skalierungsanzeige von unten begrenzt.
„Skalende“	Wert, der die Skalierungsanzeige von oben begrenzt.
„Hauptskala“	Abstand zwischen zwei Strichen der Grobskala.
„Unterskala“	Abstand zwischen zwei Strichen der Feinskala; kann auf 0 gesetzt werden, falls eine weitere Unterteilung der Grobskala nicht gewünscht ist.
„Skalenlinienstärke“	Stärke der Skala in Pixel
„Skalenfarbe“	Die Farbe kann über Selektionsliste bzw. über Schaltfläche <input type="button" value="..."/> aus dem Standarddialog zur Farbauswahl eingestellt werden.
„Skala in 3D“	Ist die Option aktiviert, wird die Skala dreidimensional dargestellt.
„Skala anzeigen“	Wird keine Skalendarstellung gewünscht, ist diese Option zu deaktivieren.
„Innerer Rahmen“	Ist diese Option aktiviert, wird der Skalenring innen mit einem Rahmen versehen. Standardmäßig deaktiviert.
„Äußerer Rahmen“	Ist diese Option aktiviert, wird der Skalenring außen mit einem Rahmen versehen. Standardmäßig deaktiviert.


## Beschriftung

Eigenschaft	Beschreibung Wert
„Beschriftung“	<p>Auswählen aus</p> <ul style="list-style-type: none"> <li>▪ Außen: Skalenwerte werden auf der Außenseite der Skala platziert.</li> <li>▪ Innen: Skalenwerte werden auf der Innenseite der Skala platziert.</li> </ul>
„Einheit“	String, der unterhalb des Pfeils (M) bzw. unterhalb des Mittelpunkts der Skalierung (BD) angezeigt wird (gedacht für die Eingabe der Skalierungseinheit)
„Schriftart“	<p>Schriftart für Darstellung von Einheit und Skalenbereich</p> <p>Festzulegen ist die Schriftart durch auswählen von einem aus:</p> <ul style="list-style-type: none"> <li>▪ Standard</li> <li>▪ Überschrift</li> <li>▪ Große</li> <li>▪ Titel</li> <li>▪ Anmerkung</li> </ul> <p>Mit Klick auf <input type="button" value="..."/> öffnet ein Dialog für eine benutzerdefinierte Einstellung der Schriftart.</p>
„Skalenformat (C-Syntax)“	Hier ist die C-Syntax zur Angabe der Formatierung der Skalierungsbeschriftung zu benutzen. Z.B. führt die Eingabe des Strings "%3.2f s" in diesem Feld zur Anzeige der

Skalierungsbeschriftungen mit 3 Stellen, davon 2 Nachkommastellen, gefolgt vom Buchstaben "s"

„*Max. Breite der Beschriftungen*“ Wert, der die maximale Breite der Skalierungsbeschriftung vorgibt; in der Regel wird dieser Wert automatisch richtig festgesetzt. Benützen Sie diese Vorgabe nur, falls die automatische Anpassung nicht zum gewünschten Ergebnis führt.

„*Höhe der Beschriftungen*“ Wert, der die Höhe der Skalierungsbeschriftung vorgibt; in der Regel wird dieser Wert automatisch richtig festgesetzt. Benützen Sie diese Vorgabe nur, falls die automatische Anpassung nicht zum gewünschten Ergebnis führt.

„*Farbe Schriftart*“ Farbe für die Schrift kann über Selektionsliste bzw. über Schaltfläche  aus dem Standarddialog zur Farbauswahl eingestellt werden.

## Positionierung


Eigenschaft	Beschreibung Wert
„ <i>Zeigerabstand</i> “	Hier kann die Länge des Zeigers in Pixel angegeben werden.
„ <i>Beschriftungsverschiebung</i> “	Hier kann ein Abstand in Pixel definiert werden um die Beschriftung zu positionieren.
„ <i>Einheitenverschiebung</i> “	Hier kann ein vertikaler Abstand eingegeben werden, um den Text eingegeben in <i>Beschriftung</i> → <i>Einheit</i> zu positionieren.
„ <i>Ursprungsverschiebung</i> “	Dieser Wert definiert den Versatz des Elements. Er kann verwendet werden um eine exakte Positionierung relativ zum Hintergrundbild zu erreichen.

## Farben

Property	Value Description
„ <i>Farbbereiche</i> “	„ <i>Dauerhafte Farbbereiche</i> “: Dessen Auswirkung ist nur im Online-Mode sichtbar. Ist die Option aktiviert, sind die Farbbereiche dauerhaft sichtbar. Ist die Option nicht aktiviert, ist nur der Farbbereich, auf den der Pfeil zeigt, sichtbar.
▪ „ <i>Dauerhafte Farbbereiche</i> “	
▪ „ <i>[&lt;n&gt;</i> “	„ <i>Neu erstellen</i> “: Mit Klick auf diesen Button wird ein weiterer Farbbereich erzeugt und angezeigt.

Farben: Farbbereiche: Bereiche: [<n>]

Anzahl der Farbbereiche in aufsteigender, nicht editerbarer Reihenfolge.

„*Löschen*“: Ein Klick auf den Button  wird den Farbbereich von der Liste entfernen und die Reihenfolge neu arrangieren.

Eigenschaft	Beschreibung Wert
„ <i>Bereichsanfang</i> “	Wert, der den Farbbereich von unten begrenzt.
„ <i>Bereichsende</i> “	Wert, der den Farbbereich von oben begrenzt.

„Farbe“

Farbe, in der der Farbbereich dargestellt werden soll.

Eigenschaft	Beschreibung Wert
-------------	-------------------

„Zugriffsrechte“ Mit Klick öffnet [↘ „Zugriffsrechte“](#).

Statusmeldungen:

- Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.
- Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.

## Potentiometer Frei Konfigurierbar

Hier kann einer Visualisierung ein frei konfigurierbares Potentiometer hinzugefügt werden. Solch ein Bedienelement zeigt mittels Pfeil/Zeiger einen Variablenwert an, der über Benutzereingabe verschoben werden kann.

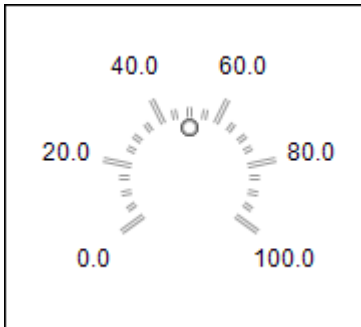


Abb. 93: Potentiometer frei konfigurierbar

### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungseditor werden folgende Benutzereingaben zur Verfügung gestellt:

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Potentiometer frei konfigurierbar
„ <i>Variable</i> “	Zuzuweisen ist eine numerische Variable, die die Position des Potentiometers enthält.

#### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„ <i>X</i> “	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„ <i>Y</i> “	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„ <i>Breite</i> “	Horizontale Breite des Elements.
„ <i>Höhe</i> “	Vertikale Höhe des Elements.

## Hintergrund

Eigenschaft	Beschreibung Wert
„ <i>Hintergrundbild</i> “	Zuzuweisen ist ein Bild aus dem GlobellImagePool indem entweder der Name der Bilddatei oder die Bild-ID angegeben wird.
„ <i>Transparenzfarbe</i> “	Für Bilder mit transparentem Hintergrund kann eine Farbe angegeben werden.

## Zeiger

Eigenschaft	Beschreibung Wert
„ <i>Zeigertyp</i> “	Auszuwählen ist aus: <ul style="list-style-type: none"><li>▪ Normaler Zeiger</li><li>▪ Dünner Zeiger</li><li>▪ Breiter Zeiger</li><li>▪ Dünne Nadel</li><li>▪ Dünner 3D-Zeiger</li><li>▪ Dünner 3D-Nadel</li></ul>
„ <i>Farbe</i> “	Farbe, mit der der Zeiger angezeigt wird.
„ <i>Zeiger Anfang</i> “	Der hier anzugebenen Wert wird als Winkel in Grad interpretiert, der sich vom Nullpunkt der Skala bis zum Anfang des Skalenbereichs im Gegenuhrzeigersinn aufspannt. Nullpunkt der Skala liegt bei "3 Uhr". Der Winkel hat einen Wertebereich von 0° bis 360°. Zentrum der Drehung ist die X/Y-Position.
„ <i>Zeiger Ende</i> “	Der hier anzugebenen Wert wird als Winkel in Grad interpretiert, der sich vom Nullpunkt der Skala bis zum Ende des Skalenbereichs im Gegenuhrzeigersinn aufspannt. Nullpunkt der Skala liegt bei "3 Uhr". Eingabe eines negativen Werts ist erlaubt. Zentrum der Drehung ist die X/Y-Position. „ <i>Zeiger Ende</i> “ muss so gesetzt werden, dass es im Uhrzeigersinn betrachtet rechts von „ <i>Zeiger Anfang</i> “ zu liegen kommt. Der Winkel hat einen Wertebereich von 0° bis maximal 350°.

## Skala

Eigenschaft	Beschreibung Wert
„ <i>Unterskalaposition</i> “	Die Unterskala kann am äußeren oder inneren Radius des Skalenrings angezeigt werden: <ul style="list-style-type: none"><li>▪ Außen</li><li>▪ Innen</li></ul>
„ <i>Skalentyp</i> “	Die Skala kann angezeigt werden als: <ul style="list-style-type: none"><li>▪ Linien</li><li>▪ Punkte</li><li>▪ Quadrate</li></ul>


„Skalenanfang“ „	Wert, der die Skalierungsanzeige von unten begrenzt.
„Skalenende“	Wert, der die Skalierungsanzeige von oben begrenzt.
„Hauptskala“	Abstand zwischen zwei Strichen der Grobskala.
„Unterskala“	Abstand zwischen zwei Strichen der Feinskala; kann auf 0 gesetzt werden, falls eine weitere Unterteilung der Grobskala nicht gewünscht ist.
„Skalenlinienstärke“	Stärke der Skala in Pixel
„Skalenfarbe“	Die Farbe kann über Selektionsliste bzw. über Schaltfläche <input type="button" value="..."/> aus dem Standarddialog zur Farbauswahl eingestellt werden.
„Skala in 3D“	Ist die Option aktiviert, wird die Skala dreidimensional dargestellt.
„Skala anzeigen“	Wird keine Skalendarstellung gewünscht, ist diese Option zu deaktivieren.
„Innerer Rahmen“	Ist diese Option aktiviert, wird der Skalenring innen mit einem Rahmen versehen. Standardmäßig deaktiviert.
„Äußerer Rahmen“	Ist diese Option aktiviert, wird der Skalenring außen mit einem Rahmen versehen. Standardmäßig deaktiviert.

## Beschriftung

Eigenschaft	Beschreibung Wert
„Beschriftung“	<p>Auswählen aus</p> <ul style="list-style-type: none"> <li>▪ Außen: Skalenwerte werden auf der Außenseite der Skala platziert.</li> <li>▪ Innen: Skalenwerte werden auf der Innenseite der Skala platziert.</li> </ul>
„Einheit“	String, der unterhalb des Pfeils (M) bzw. unterhalb des Mittelpunkts der Skalierung (BD) angezeigt wird (gedacht für die Eingabe der Skalierungseinheit)
„Schriftart“	<p>Schriftart für Darstellung von Einheit und Skalenbereich</p> <p>Festzulegen ist die Schriftart durch auswählen von einem aus:</p> <ul style="list-style-type: none"> <li>▪ Standard</li> <li>▪ Überschrift</li> <li>▪ Große</li> <li>▪ Titel</li> <li>▪ Anmerkung</li> </ul> <p>Mit Klick auf <input type="button" value="..."/> öffnet ein Dialog für eine benutzerdefinierte Einstellung der Schriftart.</p>
„Skalenformat (C-Syntax)“	Hier ist die C-Syntax zur Angabe der Formatierung der Skalierungsbeschriftung zu benutzen. Z.B. führt die Eingabe des Strings "%3.2f s" in diesem Feld zur Anzeige der Skalierungsbeschriftungen mit 3 Stellen, davon 2 Nachkommastellen, gefolgt vom Buchstaben "s"

„Max. Breite der Beschriftungen“ Wert, der die maximale Breite der Skalierungsbeschriftung vorgibt; in der Regel wird dieser Wert automatisch richtig festgesetzt. Benützen Sie diese Vorgabe nur, falls die automatische Anpassung nicht zum gewünschten Ergebnis führt.

„Höhe der Beschriftungen“ Wert, der die Höhe der Skalierungsbeschriftung vorgibt; in der Regel wird dieser Wert automatisch richtig festgesetzt. Benützen Sie diese Vorgabe nur, falls die automatische Anpassung nicht zum gewünschten Ergebnis führt.

„Farbe Schriftart“ Farbe für die Schrift kann über Selektionsliste bzw. über Schaltfläche  aus dem Standarddialog zur Farbauswahl eingestellt werden.

## Positionierung


Eigenschaft	Beschreibung Wert
„Zeigerabstand“	Hier kann die Länge des Zeigers in Pixel angegeben werden.
„Skalenabstand“	Hier kann ein Abstand der Skalenstriche zum Zentrum in Pixel angegeben werden.
„Skalenlänge“	Hier kann die Länge der Skalenstriche in Pixel angegeben werden.
„Beschriftungsverschiebung“	Hier kann ein Abstand in Pixel definiert werden um die Beschriftung zu positionieren.
„Einheitenverschiebung“	Hier kann ein vertikaler Abstand eingegeben werden, um den Text eingegeben in <i>Beschriftung</i> → <i>Einheit</i> zu positionieren.
„Ursprungsverschiebung“	Dieser Wert definiert den Versatz des Elements. Er kann verwendet werden um eine exakte Positionierung relativ zum Hintergrundbild zu erreichen.

## Farben

Property	Value Description
„Farbbereiche“	„Dauerhafte Farbbereiche“: Dessen Auswirkung ist nur im Online-Mode sichtbar. Ist die Option aktiviert, sind die Farbbereiche dauerhaft sichtbar. Ist die Option nicht aktiviert, ist nur der Farbbereich, auf den der Pfeil zeigt, sichtbar.
▪ „Dauerhafte Farbbereiche“	
▪ „[<n>]“	„Neu erstellen“: Mit Klick auf diesen Button wird ein weiterer Farbbereich erzeugt und angezeigt.

**Farben: Farbbereiche: Bereiche: [<n>]**

Anzahl der Farbbereiche in aufsteigender, nicht editierbarer Reihenfolge.

„Löschen“: Ein Klick auf den Button  wird den Farbbereich von der Liste entfernen und die Reihenfolge neu arrangieren.

Eigenschaft	Beschreibung Wert
„Bereichsanfang“	Wert, der den Farbbereich von unten begrenzt.



„Bereichsende“

Wert, der den Farbbereich von oben begrenzt.

„Farbe“

Farbe, in der der Farbbereich dargestellt werden soll.

## Histogramm

Dieses Element fügt eine Histogrammansicht der Visualisierung zu um Werte eines Arrays darzustellen. Der minimale und maximale Anzeigewert kann angegeben werden. Für bestimmte Wertebereiche können spezielle Farben definiert werden.

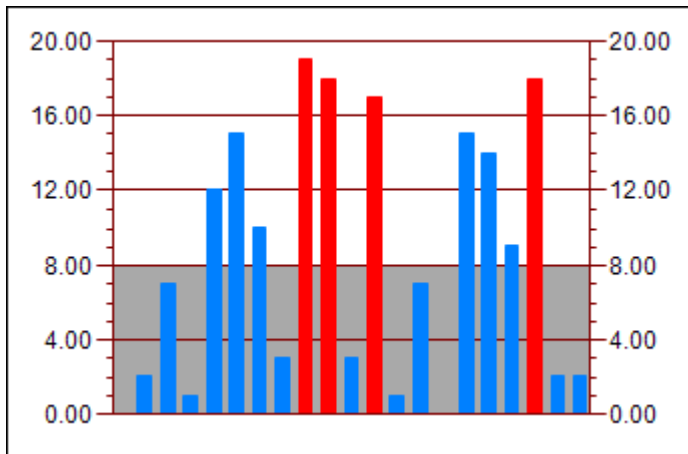


Abb. 94: Histogramm

### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungseditor werden folgende Benutzereingaben zur Verfügung gestellt:

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„Elementname“	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„Elementtyp“	Histogramm
„Datenarray“	Zuzuweisen ist ein Datenarray, der visualisiert werden soll.

#### Unterbereich des Arrays

Eigenschaft	Beschreibung Wert
„Unterbereich verwenden“	Ist diese Option aktiviert, wird nur der Unterbereich des zugewiesenen Datenarrays verwendet.
„Startindex“	Indiziert den ersten Datensatz des zugewiesenen Datenarrays.

„Endindex“

Indiziert den letzten Datensatz des zugewiesenen Datenarrays.

Eigenschaft	Beschreibung Wert
„Darstellungstyp“	<ul style="list-style-type: none"><li>▪ Balken</li><li>▪ Linien</li><li>▪ Kurve</li></ul>
„Horizontale Linien anzeigen“	Ist diese Option aktiviert, werden horizontale Linien an der Hauptskala gezeichnet.
„Relative Balkenbreite“	Dieser Wert definiert die relative Breite des Balkens bzw. der Linie in Pixel.

## Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„X“	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„Y“	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„Breite“	Horizontale Breite des Elements.
„Höhe“	Vertikale Höhe des Elements.

## Skala

Eigenschaft	Beschreibung Wert
„Skalenanfang“	Wert, der die Skalierungsanzeige von unten begrenzt
„Skalende“	Wert, der die Skalierungsanzeige von oben begrenzt
„Hauptskala“	Abstand zwischen zwei Strichen der Grobskala
„Unterskala“	Abstand zwischen zwei Strichen der Feinskala. Kann auf 0 gesetzt werden, falls eine weitere Unterteilung der Grobskala nicht gewünscht ist.
„Skalenfarbe“	Die Farbe kann über Selektionsliste bzw. über Schaltfläche <input type="button" value="..."/> aus dem Standarddialog zur Farbauswahl eingestellt werden.

## Beschriftung

Eigenschaft	Beschreibung Wert
„Einheit“	Der eingegebene Text beschriftet das Element.  String, der unterhalb des Pfeils (M) bzw. unterhalb des Mittelpunkts der Skalierung (BD) angezeigt wird (gedacht für die Eingabe der Skalierungseinheit).

### „Schriftart“

Festzulegen ist die Schriftart für Einheit und Skalierung:

Auszuwählen ist:

- Standard
- Titelzeile
- Large
- Title
- Annotation

Mit Klick auf  öffnet ein Dialog für benutzerdefinierte Einstellung der Schrifteigenschaften.

### „Skalenformat (C-Syntax)“

Benutzen Sie die C-Syntax zur Angabe der Formatierung der Skalierungsbeschriftung. So führt die Eingabe des Strings "%3.2f s" in diesem Feld zur Anzeige der Skalierungsbeschriftungen mit 3 Stellen, davon 2 Nachkommastellen, gefolgt vom Buchstaben "s"

### „Max. Breite der Beschriftungen“

Wert, der die maximale Breite der Skalierungsbeschriftung vorgibt. In der Regel wird dieser Wert automatisch richtig festgesetzt. Benützen Sie diese Vorgabe nur, falls die automatische Anpassung nicht zum gewünschten Ergebnis führt.

### „Höhe der Beschriftungen“

Wert, der die Höhe der Skalierungsbeschriftung vorgibt; in der Regel wird dieser Wert automatisch richtig festgesetzt. Benützen Sie diese Vorgabe nur, falls die automatische Anpassung nicht zum gewünschten Ergebnis führt.

### „Farbe Schriftart“

Farbe für die Schrift kann über Selektionsliste bzw. über Schaltfläche  aus dem Standarddialog zur Farbauswahl eingestellt werden.

## Farben

Eigenschaft	Beschreibung Wert
„Farbe des Graphen“	Definieren Sie eine Farbe für den Graphen.
„Alarmfarbe“	<ul style="list-style-type: none"><li>▪ „Alarmwert“: Hier ist der Schwellenwert für Alarm festzulegen.</li><li>▪ „Alarmbedingung“: Wenn der aktuelle Wert des Arrayelements die Alarmbedingung erfüllt, dann wird die Alarmbedingung gesetzt.</li></ul>
„Alarmwert“	<ul style="list-style-type: none"><li>▪ Weniger: Alarm wird ausgelöst, wenn der Wert kleiner als der Schwellenwert ist.</li></ul>
„Alarmbedingung“	<ul style="list-style-type: none"><li>▪ Mehr: Alarm wird ausgelöst, wenn der Wert größer als der Schwellenwert ist.</li></ul>
„Alarmfarbe“	<ul style="list-style-type: none"><li>▪ „Alarmfarbe“: der einzelne Balken wird mit der hier angegebenen Alarmfarbe dargestellt</li></ul>
„Farbbereiche verwenden“	Wird diese Option aktiviert, werden die spezifizierten Farbbereiche so wie unter „Farbbereiche“ definiert dargestellt.


„Farbbereiche“

„*Neu erstellen*“: Mit Klick auf diesen Button wird ein neuer Bereich erstellt und in Eigenschaften aufgelistet.

- „[<n>]“

**Farben: Farbbereiche: Bereiche: [<n>]**

Anzahl der Farbbereiche in aufsteigender, nicht editierbarer Reihenfolge.

„*Löschen*“: Ein Klick auf den Button  wird den Farbbereich von der Liste entfernen und die Reihenfolge neu arrangieren.

Eigenschaft	Beschreibung Wert
„ <i>Bereichsanfang</i> “	Wert, der den Farbbereich von unten begrenzt.
„ <i>Bereichsende</i> “	Wert, der den Farbbereich von oben begrenzt.
„ <i>Farbe</i> “	Farbe, in der der Farbbereich dargestellt werden soll.

Eigenschaft	Beschreibung Wert
„ <i>Zugriffsrechte</i> “	Mit Klick öffnet <a href="#">„Zugriffsrechte“</a> .

Statusmeldungen:

- Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.
- Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.

## Konfiguration eines Histogramms

Sehen Sie im folgenden ein Beispiel für das Einfügen und die Konfiguration eines Histogramm Elements:

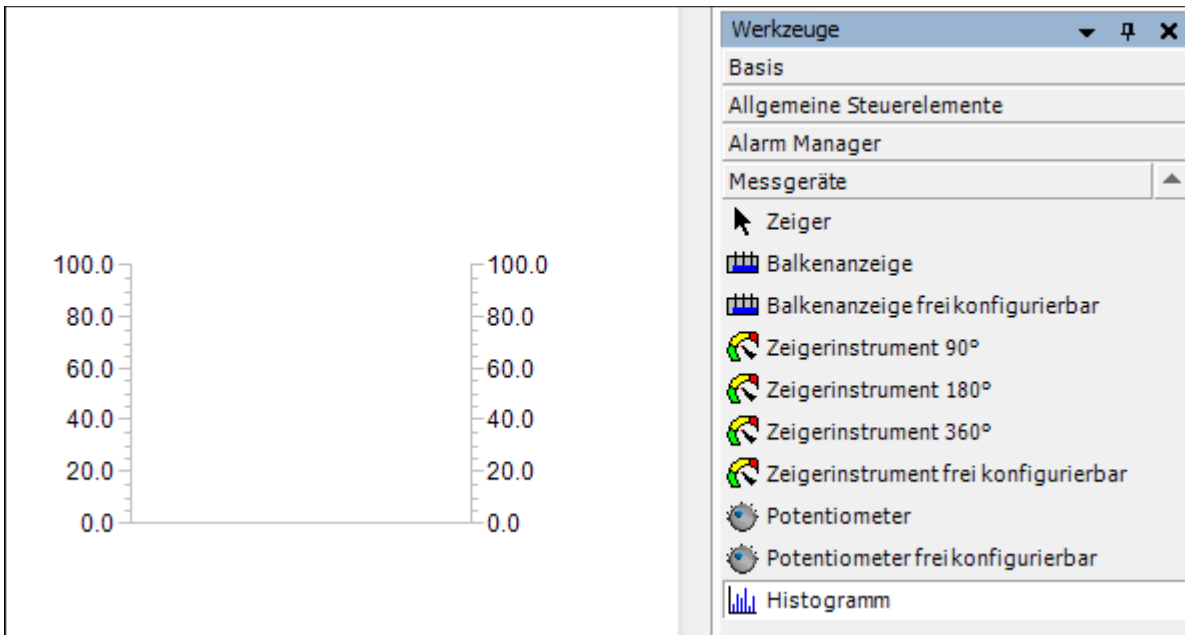


Abb. 95: Beispiel - Einfügen des Elements Histogramm

Das zugeordnete Datenarray - im Beispiel ein Array mit dem Namen `Histogram` - muss in den Elementeigenschaften des Histogramm-Elementes angegeben werden. Nach einem Klick in das Eingabefeld der Eigenschaft „*Datenarray*“ steht Ihnen der Schalter  zur Verfügung, welcher benutzt werden kann um im Projekt nach der Variable zu browsen. Achten Sie darauf, die Eingangsvariable durch Angabe ihres vollständigen Pfads im Gerätebaum zu spezifizieren.

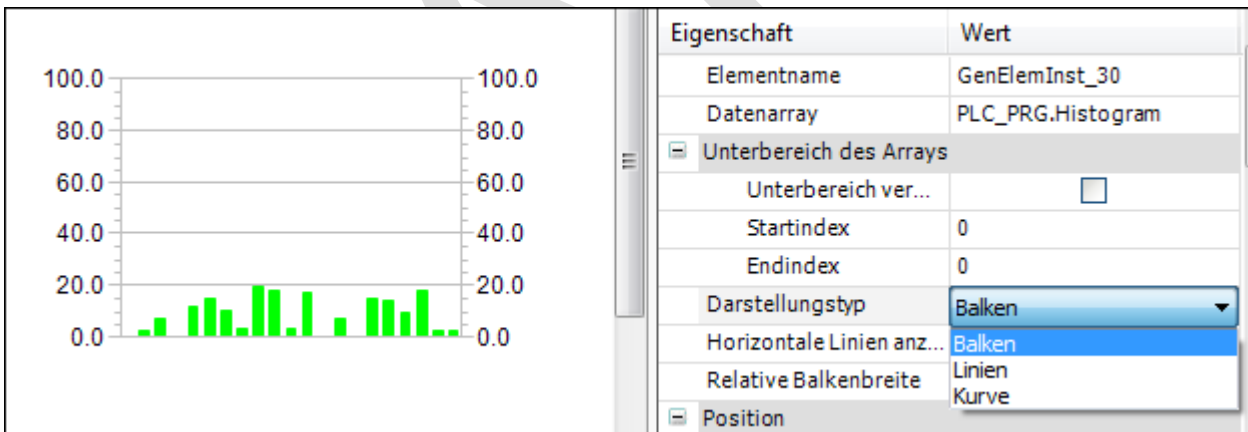
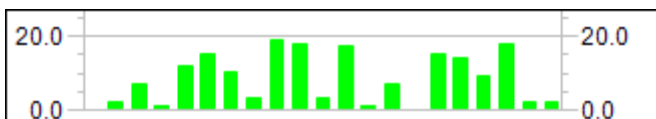


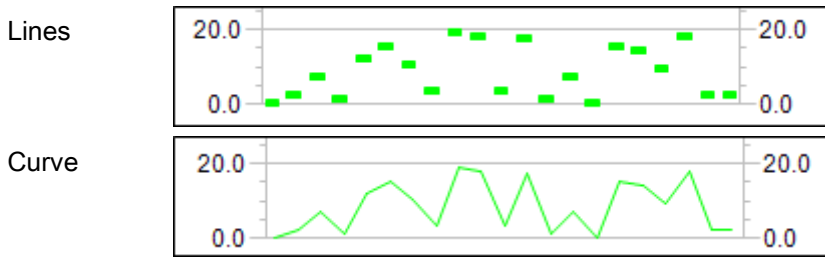
Abb. 96: Beispiel - Konfiguration des Darstellungstyps

„*Unterbereich verwenden*“: Diese Option wird benutzt, wenn nicht alle Elemente des Arrays angezeigt werden sollen, sondern nur die Elemente von Index „*Startindex*“ bis „*Endindex*“.

„*Darstellungstyp*“: Das Datenarray kann in drei unterschiedlichen Formen dargestellt werden:

Bars





„*Horizontale Linien anzeigen*“: Mit dieser Option werden horizontale Linien an der Hauptskale gezeichnet.

„*Relative Balkenbreite*“: Dieser Wert definiert die relative Breite des Balkens bzw. der Linie

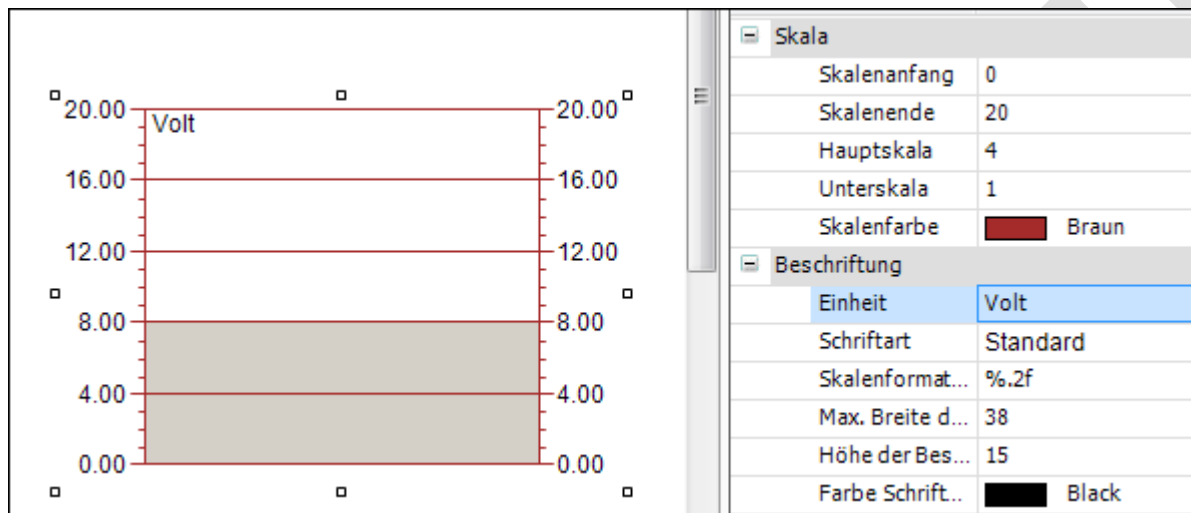


Abb. 97: Beispiel - Skalierung des Histogramm

In den Eigenschaften der „*Skala*“ kann der numerische Bereich der Skalierung und die Unterteilung durch Haupt- und Unterskala definiert werden. Der Wertebereich der Skala wird von unten durch den Wert in „*Skalenanfang*“ und von oben durch den Wert in „*Skalenende*“ begrenzt. Der Wert von „*Skalenanfang*“ muss also geringer sein als der von „*Skalenende*“. Beide Distanzangaben in „*Hauptskala*“ wie „*Unterskala*“ können auf 0 gesetzt werden um eine Anzeige der Skalierungsstriche auszuschalten: Wird der Wert der Grobskalierung (Hauptskala) auf 0 gesetzt, so werden unabhängig vom Wert der Feinskalierung keinerlei Skalierungsstriche gezeichnet. Wird die Feinskalierung (Unterskala) auf 0 gesetzt, so werden nur die Skalierungsstriche der groben Unterteilung gezeichnet. Um die Skalenfarbe zu ändern, muss auf das Feld der Skalenfarbe geklickt werden - es öffnet sich ein Dialog, in welchem die Farbe ausgewählt werden kann.

Im Abschnitt „*Beschriftung*“ kann die Darstellung der Skalierung festgelegt werden. Der Eintrag im Eingabefeld „*Einheit*“ ist zur Angabe der Einheit der Skalierung gedacht und wird links oben im Histogramm angezeigt. Nach Wahl einer geeigneten Schrift(farbe) kann das Format der Beschriftungen angepasst werden: Die Angabe der Formatierung für Zahlenwerte muss entsprechend der Syntax der Programmiersprache C vorgenommen werden (benützen Sie %d für ganzzahlige Werte und %.Xf für Gleitpunktzahlen, wobei X durch die gewünschte Anzahl an Nachkommastellen ersetzt werden muss).

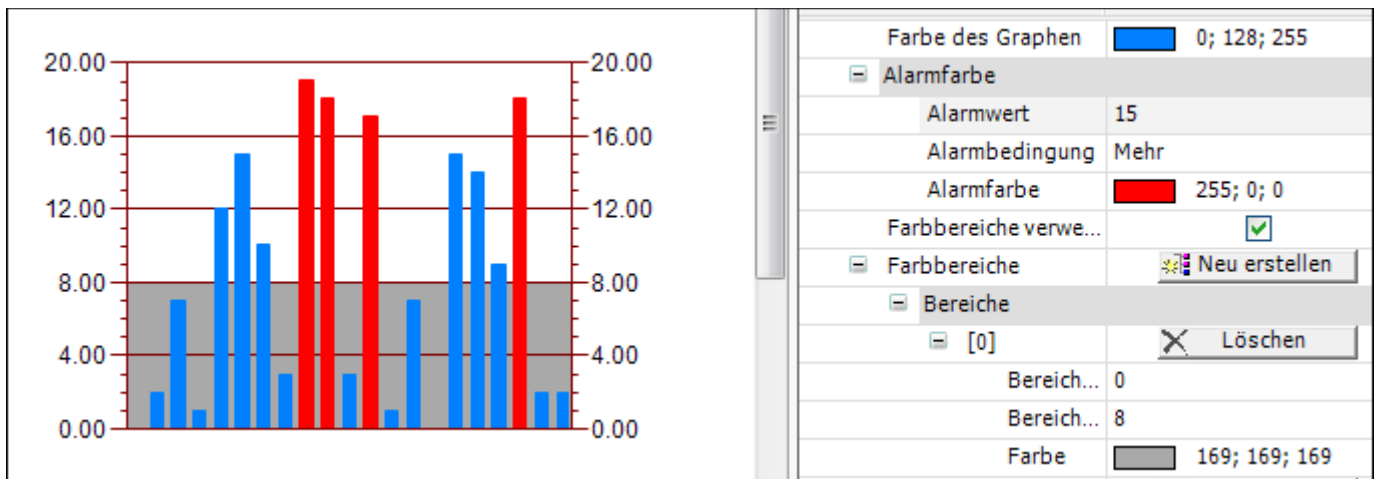




Abb. 98: Beispiel - Konfiguration der Farben des Histogramm

Schließlich kann die Einfärbung des Elements innerhalb des Abschnitts „Alarmfarbe“ gesetzt werden:

Zuerst wird die Farbe des Histogramm ausgewählt. Innerhalb des Unterbereichs „Alarmfarbe“ kann eine Alarmfarbe definiert werden, in welcher der einzelne Balken angezeigt wird, falls der aktuelle Wert des Array-Elementes die Alarmbedingung erfüllt. Diese Bedingung ergibt sich aus der Definition des Alarmwerts, welcher entweder nicht überschritten (beim Setzen der „Alarmbedingung“ auf „Mehr“) oder nicht unterschritten werden darf (beim Setzen der „Alarmbedingung“ auf „Weniger“).

Teilbereichen der Skala kann dann eine bestimmte Farbe zugewiesen werden, indem für sie mit Hilfe des Schalters  ein Farbbereich erzeugt wird. Die Farbbereiche werden aufsteigend nummeriert. Jeder Farbbereich wird innerhalb der Elementeigenschaften mit eigenen Eingabefeldern ausgestattet:

In „Bereichsanfang“ und „Bereichsende“ können die Grenzen des Unterbereichs festgelegt werden. Seine Einfärbung kann aus dem Aufklappmenü ausgewählt werden. Ein einmal angelegter Farbbereich kann durch einen Klick auf den zugehörigen Schalter  gelöscht werden.

### 6.3.5.6 Lampen/Schalter/Bilder

#### Bildwechsler

Der Bildwechsler kann verwendet werden, um zwischen zwei Bildern umzuschalten. Durch einen Mausklick auf das Bild wechselt es auf das zweite und die zugewiesene boolesche Variable wird TRUE. Entsprechend der Einstellung in „*Elementverhalten*“ wird das erste Bild wieder angezeigt, wenn entweder die Maustaste losgelassen wird oder das nächste mal ein Mausklick ausgeführt wird.

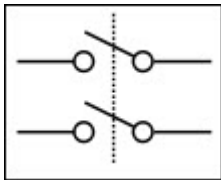


Abb. 99: Beispiel für "Bild an"

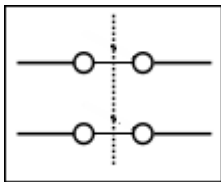


Abb. 100: Beispiel für "Bild aus"

#### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungsektor werden folgende Benutzereingaben zur Verfügung gestellt:

#### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Bildwechsler

#### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„ <i>X</i> “	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.



- „Y“ Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
- „Breite“ Horizontale Breite des Elements.
- „Höhe“ Vertikale Höhe des Elements.
- „Variable“ Boolesche Variable, deren Zustand entsprechend der Benutzereingabe wechselt. Abhängig von „Elementverhalten“ ist die Variable TRUE, solange die Maustaste gedrückt ist („Bildtaster“) oder der Wert ändert sich mit jedem Mausklick („Bildumschalter“).

## Bildeinstellungen

Um die Einträge für die Bildernamen einheitlich zu halten, sollte der Name der Bildersammlung als Präfix angegeben werden. Dies ist nicht notwendig, wenn die Bilder im GlobalImagePool verwaltet werden, da diese sowieso als erste durchsucht wird.

Eigenschaft	Beschreibung Wert
„Bild an“	Hier ist die ID einer Bilddatei anzugeben, die in der jeweiligen Bildersammlung verwendet wird. Das Bild wird für den An-Zustand verwendet. Typ STRING
„Bild aus“	Hier ist die ID einer Bilddatei anzugeben, die in der jeweiligen Bildersammlung verwendet wird. Das Bild wird für den Aus-Zustand verwendet. Typ STRING
„Skalierungsart“	<p>Hier ist zu spezifizieren, wie das Element auf die Änderung der Framegröße zu reagieren hat:</p> <ul style="list-style-type: none"> <li>▪ <b>Isotropisch:</b> Höhe und Breite des Bildes werden proportional zum Frame verändert. Beachten Sie folgendes, wenn Sie <b>die Ausrichtung der Elemente zueinander auch innerhalb eines skalierten "Frame"-Elements beibehalten</b> möchten. Beispielsweise haben Sie eine Lampe zentriert oberhalb eines Schalters angeordnet, und sie soll in dieser horizontal zentrierten Position bleiben, auch wenn sich die Größe des Frames verändert. Um unerwünschte horizontale oder vertikale Verschiebungen zu vermeiden, definieren Sie explizit die Horizontale bzw. Vertikale Ausrichtung mit "Zentriert" (siehe unten).</li> <li>▪ <b>Anisotropisch:</b> Das Bild füllt, unabhängig von seinen Proportionen, den ganzen Frame aus.</li> <li>▪ <b>Unskaliert:</b> Die Bildgröße ist festgelegt und wird nicht verändert, wenn der Frame seine Größe ändert.</li> </ul>
Horizontale Ausrichtung	<p>Nur verfügbar, wenn die Option "Experte" für den Eigenschaften-Editor aktiviert ist und wenn die Skalierungsart des Bildes "Isotropisch" ist. Dient dazu, die horizontale Ausrichtung zu einem anderen Element (wie in der Basis-Visualisierung definiert) explizit aufrechtzuerhalten, wenn die Visualisierung innerhalb eines skalierten Frames verwendet wird. Siehe oben: Einstellung "Isotropisch"</p> <ul style="list-style-type: none"> <li>▪ <b>Links</b></li> </ul>

- Zentriert
- Rechts

#### Vertikale Ausrichtung

Nur verfügbar, wenn die Option "Experte" für den Eigenschaften-Editor aktiviert ist und wenn die Skalierungsart des Bildes "Isotropisch" ist. Dient dazu, die vertikale Ausrichtung zu einem anderen Element (wie in der Basis-Visualisierung definiert) explizit aufrechtzuerhalten, wenn die Visualisierung innerhalb eines skalierten Frames verwendet wird. Siehe oben: Einstellung "Isotropisch"

- Oben
- Zentriert
- Unten

#### „Elementverhalten“

- Bildumschalter: Bild und Variable toggeln mit jedem Mausklick.
- Bildwechsler: „Bild an“ wird angezeigt und die Variable ist TRUE, solange die Maustaste gedrückt gehalten wird.

#### „FALSE Tasten“

Diese Option ist nur verfügbar, wenn bei Elementverhalten "Bildwechsler" angestellt ist.

Wenn deaktiviert (Standardeinstellung), wird die zugeordnete Variable durch das Drücken der Maustaste auf TRUE gesetzt, andernfalls wird sie auf FALSE gesetzt.

#### Texte

Diese Eigenschaften werden für eine statische Definition der Elementbeschriftung verwendet.

Eigenschaft	Beschreibung Wert
„Tooltip“	Der angegebenen Text erscheint als Tooltip des Elements, wenn die Visualisierung im Online-Modus ist.

#### Zustandsvariablen

Eigenschaft	Beschreibung Wert
„Unsichtbarkeit“	Eingabe einer boolschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.
„Eingaben deaktivieren“	Eingabe einer boolschen Variable; ist der Variablenwert TRUE, dann hat keine der Elementeingaben eine Auswirkung.

Eigenschaft	Beschreibung Wert
„Zugriffsrechte“	Mit Klick öffnet <a href="#">„Zugriffsrechte“</a> .

Statusmeldungen:

- Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das

Element für alle Gruppen bedienbar angezeigt wird.

- Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.

in Arbeit

## Lampe

Die Lampe leuchtet auf, wenn die zugewiesene Variable gesetzt wird. In „*Hintergrundbild*“ kann die Farbe der Lampe ausgewählt werden.

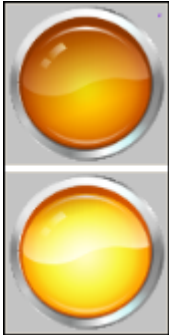


Abb. 101: Lampe

### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungsektor werden folgende Benutzereingaben zur Verfügung gestellt:

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Lampe

#### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„ <i>X</i> “	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„ <i>Y</i> “	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„ <i>Breite</i> “	Horizontale Breite des Elements.
„ <i>Höhe</i> “	Vertikale Höhe des Elements.
„ <i>Variable</i> “	Zuweisen einer boolesche Variable, deren Zustand den Zustand der Lampe schaltet. Bei TRUE wird

die Lampe eingeschaltet und leuchtend dargestellt.

## Bildeinstellungen

Eigenschaft	Beschreibung Wert
„Skalierungsart“	<p>Hier ist zu spezifizieren, wie das Element auf die Änderung der Framegröße zu reagieren hat:</p> <ul style="list-style-type: none"><li>▪ <b>Isotropisch:</b> Höhe und Breite des Bildes werden proportional zum Frame verändert. Beachten Sie folgendes, wenn Sie <b>die Ausrichtung der Elemente zueinander auch innerhalb eines skalierten "Frame"-Elements beibehalten</b> möchten. Beispielsweise haben Sie eine Lampe zentriert oberhalb eines Schalters angeordnet, und sie soll in dieser horizontal zentrierten Position bleiben, auch wenn sich die Größe des Frames verändert. Um unerwünschte horizontale oder vertikale Verschiebungen zu vermeiden, definieren Sie explizit die Horizontale bzw. Vertikale Ausrichtung mit "Zentriert" (siehe unten).</li><li>▪ <b>Anisotropisch:</b> Das Bild füllt, unabhängig von seinen Proportionen, den ganzen Frame aus.</li><li>▪ <b>Unskaliert:</b> Die Bildgröße ist festgelegt und wird nicht verändert, wenn der Frame seine Größe ändert.</li></ul>
Horizontale Ausrichtung	<p>Nur verfügbar, wenn die Option "Experte" für den Eigenschaften-Editor aktiviert ist und wenn die Skalierungsart des Bildes "Isotropisch" ist. Dient dazu, die horizontale Ausrichtung zu einem anderen Element (wie in der Basis-Visualisierung definiert) explizit aufrechtzuerhalten, wenn die Visualisierung innerhalb eines skalierten Frames verwendet wird. Siehe oben: Einstellung "Isotropisch"</p> <ul style="list-style-type: none"><li>▪ Links</li><li>▪ Zentriert</li><li>▪ Rechts</li></ul>
Vertikale Ausrichtung	<p>Nur verfügbar, wenn die Option "Experte" für den Eigenschaften-Editor aktiviert ist und wenn die Skalierungsart des Bildes "Isotropisch" ist. Dient dazu, die vertikale Ausrichtung zu einem anderen Element (wie in der Basis-Visualisierung definiert) explizit aufrechtzuerhalten, wenn die Visualisierung innerhalb eines skalierten Frames verwendet wird. Siehe oben: Einstellung "Isotropisch"</p> <ul style="list-style-type: none"><li>▪ Oben</li><li>▪ Zentriert</li><li>▪ Unten</li></ul>

## Texte

Diese Eigenschaften werden für eine statische Definition der Elementbeschriftung verwendet.

Eigenschaft	Beschreibung Wert
„ <i>Tooltip</i> “	Der angegebenen Text erscheint als Tooltip des Elements, wenn die Visualisierung im Online-Modus ist.

#### Zustandsvariablen

Eigenschaft	Beschreibung Wert
„ <i>Unsichtbarkeit</i> “	Eingabe einer boolschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.
„ <i>Eingaben deaktivieren</i> “	Eingabe einer boolschen Variable; ist der Variablenwert TRUE, dann hat keine der Elementeingaben eine Auswirkung.

#### Hintergrund

Eigenschaft	Beschreibung Wert
„ <i>Hintergrundbild</i> “	<p>Auszuwählen ist eine Farbe für die Lampe:</p> <ul style="list-style-type: none"> <li>▪ Gelb</li> <li>▪ Rot</li> <li>▪ Grün</li> <li>▪ Blau</li> <li>▪ Grau</li> </ul>

Eigenschaft	Beschreibung Wert
„ <i>Zugriffsrechte</i> “	<p>Mit Klick öffnet <a href="#">„Zugriffsrechte“</a>.</p> <p>Statusmeldungen:</p> <ul style="list-style-type: none"> <li>▪ Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.</li> <li>▪ Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.</li> </ul>

## Hebelschalter, Energieschalter, Drückschalter, Drückschalter LED, Kippschalter

Der Schalter wird verwendet um den Wert einer boolschen Variablen zu setzen



Abb. 102: Hebelschalter, Energieschalter, Drückschalter, Drückschalter LED, Kippschalter

### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungseditor werden folgende Benutzereingaben zur Verfügung gestellt:

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Typ des selektierten Elements: Hebelschalter, Energieschalter, Drückschalter, Drückschalter LED oder Kippschalter.

#### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„ <i>X</i> “	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„ <i>Y</i> “	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„ <i>Breite</i> “	Horizontale Breite des Elements.
„ <i>Höhe</i> “	Vertikale Höhe des Elements.
„ <i>Variable</i> “	Boolsche Variable, deren Wert entsprechend der Benutzereingabe wechselt. Abhängig von „ <i>Elementverhalten</i> “ ist die Variable TRUE, solange die Maustaste gedrückt ist („ <i>Bildtaster</i> “) oder der Wert ändert sich mit jedem Mausklick („ <i>Bildumschalter</i> “).

#### Bildeinstellungen

Property	Value Description
„Skalierungsart“	<p>Hier ist zu spezifizieren, wie das Element auf die Änderung der Framegröße zu reagieren hat:</p> <ul style="list-style-type: none"> <li>▪ <b>Isotropisch:</b> Höhe und Breite des Bildes werden proportional zum Frame verändert. Beachten Sie folgendes, wenn Sie <b>die Ausrichtung der Elemente zueinander auch innerhalb eines skalierten "Frame"-Elements beibehalten</b> möchten. Beispielsweise haben Sie eine Lampe zentriert oberhalb eines Schalters angeordnet, und sie soll in dieser horizontal zentrierten Position bleiben, auch wenn sich die Größe des Frames verändert. Um unerwünschte horizontale oder vertikale Verschiebungen zu vermeiden, definieren Sie explizit die Horizontale bzw. Vertikale Ausrichtung mit "Zentriert" (siehe unten).</li> <li>▪ <b>Anisotropisch:</b> Das Bild füllt, unabhängig von seinen Proportionen, den ganzen Frame aus.</li> <li>▪ <b>Unskaliert:</b> Die Bildgröße ist festgelegt und wird nicht verändert, wenn der Frame seine Größe ändert.</li> </ul>
Horizontale Ausrichtung	<p>Nur verfügbar, wenn die Option "Experte" für den Eigenschaften-Editor aktiviert ist und wenn die Skalierungsart des Bildes "Isotropisch" ist. Dient dazu, die horizontale Ausrichtung zu einem anderen Element (wie in der Basis-Visualisierung definiert) explizit aufrechtzuerhalten, wenn die Visualisierung innerhalb eines skalierten Frames verwendet wird. Siehe oben: Einstellung "Isotropisch"</p> <ul style="list-style-type: none"> <li>▪ Links</li> <li>▪ Zentriert</li> <li>▪ Rechts</li> </ul>
Vertikale Ausrichtung	<p>Nur verfügbar, wenn die Option "Experte" für den Eigenschaften-Editor aktiviert ist und wenn die Skalierungsart des Bildes "Isotropisch" ist. Dient dazu, die vertikale Ausrichtung zu einem anderen Element (wie in der Basis-Visualisierung definiert) explizit aufrechtzuerhalten, wenn die Visualisierung innerhalb eines skalierten Frames verwendet wird. Siehe oben: Einstellung "Isotropisch"</p> <ul style="list-style-type: none"> <li>▪ Oben</li> <li>▪ Zentriert</li> <li>▪ Unten</li> </ul>
„Elementverhalten“	<ul style="list-style-type: none"> <li>▪ <b>Bildumschalter:</b> Der Schalter wird als eingeschaltet angezeigt, und die Variable ist TRUE, solange die Maustaste gedrückt gehalten wird.</li> <li>▪ <b>Bildwechsler:</b> Die Anzeige des Schalters und Variable toggeln mit jedem Mausklick.</li> </ul>



## Texte

Diese Eigenschaften werden für eine statische Definition der Elementbeschriftung verwendet.

Eigenschaft	Beschreibung Wert
„ <i>Tooltip</i> “	Der angegebenen Text erscheint als Tooltip des Elements, wenn die Visualisierung im Online-Modus ist.

## Zustandsvariablen

Eigenschaft	Beschreibung Wert
„ <i>Unsichtbarkeit</i> “	Eingabe einer boolschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.
„ <i>Eingaben deaktivieren</i> “	Eingabe einer boolschen Variable; ist der Variablenwert TRUE, dann hat keine der Elementeingaben eine Auswirkung.

## Hintergrund

Eigenschaft	Beschreibung Wert
„ <i>Hintergrundbild</i> “	Auszuwählen ist eine Farbe, die den Schalter markiert: <ul style="list-style-type: none"><li>▪ Gelb</li><li>▪ Rot</li><li>▪ Grün</li><li>▪ Blau</li><li>▪ Grau</li></ul>

Eigenschaft	Beschreibung Wert
„ <i>Zugriffsrechte</i> “	Mit Klick öffnet <a href="#">„Zugriffsrechte“</a> .  Statusmeldungen: <ul style="list-style-type: none"><li>▪ Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.</li><li>▪ Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.</li></ul>

## Drehschalter

Der Drehschalter wird verwendet um den Wert einer boolschen Variablen zu setzen.



Abb. 103: Drehschalter

### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungseditor werden folgende Benutzereingaben zur Verfügung gestellt:

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Drehschalter

#### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„ <i>X</i> “	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„ <i>Y</i> “	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„ <i>Breite</i> “	Horizontale Breite des Elements.
„ <i>Höhe</i> “	Vertikale Höhe des Elements.
„ <i>Variable</i> “	Boolsche Variable, deren Wert entsprechend der Benutzereingabe wechselt. Abhängig von „ <i>Elementverhalten</i> “ ist die Variable TRUE, solange die Maustaste gedrückt ist („ <i>Bildtaster</i> “) oder der Wert ändert sich mit jedem Mausklick („ <i>Bildumschalter</i> “).

#### Bildeinstellungen

Eigenschaft	Beschreibung Wert
„ <i>Skalierungsart</i> “	Hier ist zu spezifizieren, wie das Element auf die Änderung der Framegröße zu reagieren hat: <ul style="list-style-type: none"> <li>▪ <b>Isotropisch:</b> Höhe und Breite des Bildes werden proportional zum Frame</li> </ul>

verändert.

Beachten Sie folgendes, wenn Sie **die Ausrichtung der Elemente zueinander auch innerhalb eines skalierten "Frame"-Elements beibehalten** möchten. Beispielsweise haben Sie eine Lampe zentriert oberhalb eines Schalters angeordnet, und sie soll in dieser horizontal zentrierten Position bleiben, auch wenn sich die Größe des Frames verändert. Um unerwünschte horizontale oder vertikale Verschiebungen zu vermeiden, definieren Sie explizit die Horizontale bzw. Vertikale Ausrichtung mit "Zentriert" (siehe unten).

- **Anisotropisch:** Das Bild füllt, unabhängig von seinen Proportionen, den ganzen Frame aus.
- **Unskaliert:** Die Bildgröße ist festgelegt und wird nicht verändert, wenn der Frame seine Größe ändert.

#### Horizontale Ausrichtung

Nur verfügbar, wenn die Option "Experte" für den Eigenschaften-Editor aktiviert ist und wenn die Skalierungsart des Bildes "Isotropisch" ist. Dient dazu, die horizontale Ausrichtung zu einem anderen Element (wie in der Basis-Visualisierung definiert) explizit aufrechtzuerhalten, wenn die Visualisierung innerhalb eines skalierten Frames verwendet wird. Siehe oben: Einstellung "Isotropisch"

- Links
- Zentriert
- Rechts

#### Vertikale Ausrichtung

Nur verfügbar, wenn die Option "Experte" für den Eigenschaften-Editor aktiviert ist und wenn die Skalierungsart des Bildes "Isotropisch" ist. Dient dazu, die vertikale Ausrichtung zu einem anderen Element (wie in der Basis-Visualisierung definiert) explizit aufrechtzuerhalten, wenn die Visualisierung innerhalb eines skalierten Frames verwendet wird. Siehe oben: Einstellung "Isotropisch"

- Oben
- Zentriert
- Unten

#### „Elementverhalten“

- **Bildumschalter:** Der Drehschalter wird als eingeschaltet angezeigt, und die Variable ist TRUE, solange die Maustaste gedrückt gehalten wird.
- **Bildwechsler:** Zustand des Drehschalters und Variable toggeln mit jedem Mausklick.

#### „Ausrichtung“

- **Oben:** Der Schalter ist nach oben gerichtet.
- **Seitlich:** Der Schalter ist seitlich gerichtet.

### „Farbwechsel“

Ist die Option nicht aktiviert, dann leuchtet der Schalter auf, wenn er auf eingeschaltet ist.

Ist die Option aktiviert, dann leuchtet der Schalter nicht auf, auch wenn er auf eingeschaltet ist.

## Texte

Diese Eigenschaften werden für eine statische Definition der Elementbeschriftung verwendet.

Eigenschaft	Beschreibung Wert
„ <a href="#">Tooltip</a> “	Der angegebenen Text erscheint als Tooltip des Elements, wenn die Visualisierung im Online-Modus ist.

## Zustandsvariablen

Eigenschaft	Beschreibung Wert
„ <a href="#">Unsichtbarkeit</a> “	Eingabe einer boolschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.
„ <a href="#">Eingaben deaktivieren</a> “	Eingabe einer boolschen Variable; ist der Variablenwert TRUE, dann hat keine der Elementeingaben eine Auswirkung.

## Hintergrund

Eigenschaft	Beschreibung Wert
„ <a href="#">Hintergrundbild</a> “	Auszuwählen ist eine Farbe, die den Schalter markiert: <ul style="list-style-type: none"><li>▪ Gelb</li><li>▪ Rot</li><li>▪ Grün</li><li>▪ Blau</li><li>▪ Grau</li></ul>

Eigenschaft	Beschreibung Wert
„ <a href="#">Zugriffsrechte</a> “	Mit Klick öffnet <a href="#">„Zugriffsrechte“</a> .  Statusmeldungen: <ul style="list-style-type: none"><li>▪ Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.</li><li>▪ Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.</li></ul>

### 6.3.5.7 Spezielle Steuerelemente

#### Trace und seine Dialoge

Das Element ermöglicht es, einen Trace-Graphen, der permanent Variablenwerte monitort und anzeigt, in die Visualisierung zu integrieren. Der angezeigte Trace-Graph wird in den Elementeigenschaften konfiguriert. Zusätzlich sind Steuerelemente, die die Trace-Funktionalität steuern, entweder von Hand oder mit Hilfe des [„Trace-Assistent“](#) zu programmieren.

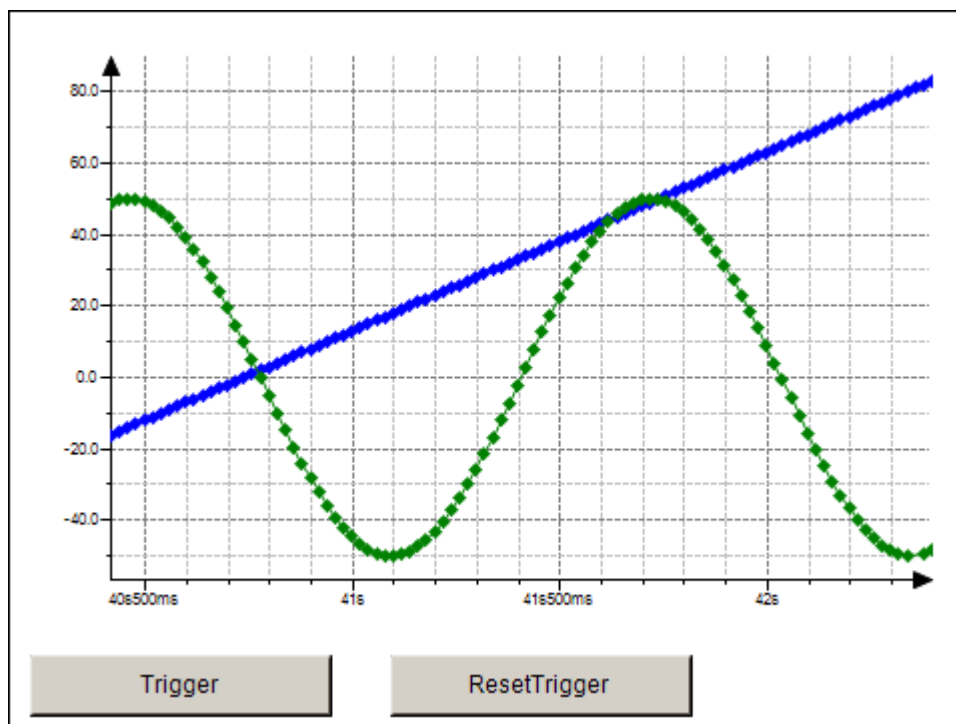


Abb. 104: Visualisierungstrace online

#### ActiveX-Element

Das ActiveX-Element erlaubt es, eine existierende ActiveX-Komponente zu implementieren



Das ActiveX Control kann innerhalb der Target-Visualisierung unter Windows verwendet werden. Auf anderen Target-Visualisierungs-Plattformen wie WinCE und Linux ist es von dem OEM abhängig, welche Controls unterstützt werden.

Methoden und Parameter können mit den Schaltflächen  **Neu erstellen** und  **Löschen** hinzugefügt und gelöscht werden.

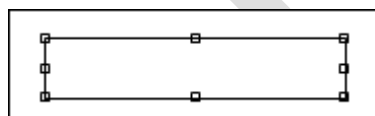


Abb. 119: ActiveX-Element

#### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungseditor werden folgende Benutzereingaben zur Verfügung gestellt:

#### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

## Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	ActiveX-Element
„ <i>Element</i> “	Hier kann eine installierte ActiveX-Komponente zugewiesen werden. <input type="text"/> ist zu verwenden, wenn eine Komponente mit Hilfe des Eingabeassistenten ausgewählt werden soll. Es ist ein STRING-Parameter einzugeben, das den Namen oder die ID der Komponenten beschreibt.

## Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„ <i>X</i> “	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„ <i>Y</i> “	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„ <i>Breite</i> “	Horizontale Breite des Elements.
„ <i>Höhe</i> “	Vertikale Höhe des Elements.

## Absolute Bewegung

Das Element kann bewegt werden, indem die X- und Y-Position (Pixel) der linken oberen Ecke des Elements durch eine Integer-Variable verändert wird. Hier werden absolute Koordinatenwerte verwendet.

Eigenschaft	Beschreibung Wert
„ <i>Bewegung</i> “	<ul style="list-style-type: none"><li>▪ „<i>X</i>“: Hier ist eine Integer-Variable einzugeben, deren Wert die aktuelle X-Position der linken oberen Ecke des Elements in Pixel definiert; kann verwendet werden, um das Element in X-Richtung zu verschieben (ein positiver Wert verschiebt das Element von links nach rechts).</li><li>▪ „<i>Y</i>“: Eingabe einer Integer-Variablen, die die aktuelle Y-Position der linken oberen Ecke des Elements in Pixel definiert; kann verwendet werden, um das Element in Y-Richtung zu verschieben (ein positiver Wert verschiebt das Element von oben nach unten).</li></ul>

## Zustandsvariablen

Property	Value description
„Unsichtbarkeit“	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.

### Initiale Aufrufe

Methodenaufrufe, die während der Initialisierung ausgeführt werden, können unter dieser Eigenschaft definiert werden. Sie werden ausschließlich im ersten Zyklus ausgeführt.

Eigenschaft	Beschreibung Wert
„Methodenaufrufe“	<ul style="list-style-type: none"> <li>▪ „Methoden“: Mit Klick auf „Neu erstellen“ wird ein weiterer Eintrag eines Methodenaufrufs erzeugt.</li> </ul>
Methoden	

### Zyklische Aufrufe

Methodenaufrufe, die in jedem Zyklus ausgeführt werden, können unter dieser Eigenschaft definiert werden. Sie werden im Aktualisierungsintervall der Visualisierung aufgerufen.

Eigenschaft	Beschreibung Wert
„Methodenaufrufe“	<ul style="list-style-type: none"> <li>▪ „Methoden“: Mit Klick auf „Neu erstellen“ wird eine weiterer Eintrag eines Methodenaufrufs erzeugt.</li> </ul>
Methoden	

### Bedingte Aufrufe

Zusätzlich kann ein bedingter Aufrufe zugewiesen werden.

Bedingte Methodenaufrufe werden während dem Aktualisierungsintervall der Visualisierung ausgeführt und können hier definiert werden. Anders als bei initialen oder zyklischen Aufrufen wird eine Aufrufbedingung unter der Eigenschaft *Methoden* → [*Nummer*] definiert. Sie werden nur bei steigender Flanke der Aufrufbedingung ausgeführt.

Eigenschaft	Beschreibung Wert
„Methodenaufrufe“	<ul style="list-style-type: none"> <li>▪ „Methoden“: Mit Klick auf „Neu erstellen“ wird eine weiterer Eintrag eines Methodenaufrufs erzeugt.</li> </ul>
Methoden	
[<Nummer>]	

### Initiale/Zyklische/Bedingte Aufrufe: Methoden[<Nummer>]

Eigenschaft	Beschreibung Wert
„[<Nummer>]“	Die Nummer indiziert einen Methodenaufruf.
„“	„Löschen“: Löscht den zugehörigen Methodenaufruf mit allen Einstellungen: „Parameter“, „Ergebnisparameter“ und alles unter „Variable“
„Method“	Der Name der aufzurufenden Methode muss hier eingegeben werden.
„Parameter“	Die Variablen der Parameter müssen hier zugewiesen werden.
<ul style="list-style-type: none"> <li>▪ „Parameter“</li> <li>▪ „Parameter“</li> <li>▪ „Parameter“</li> </ul>	<p>„Neu erstellen“: Mit jedem Klick wird ein weiterer Parametereintrag „Variable“ erzeugt.</p> <p>„Zuzuweisen“: Zuzuweisen ist ein Variablenname für den zugehörigen Parameter.</p>
„Ergebnisparameter“	Jede Methode hat einen Ergebnisparameter zur Verfügung.
▪ „Variable“	„Variable“: Zuzuweisen ist eine Variable, die das Ergebnis enthalten soll.
„Aufrufbedingung“	<p>„Aufrufbedingung“ steht nur in „Bedingte Aufrufe“ zur Verfügung.</p> <p>„Variable“: Zuzuweisen ist eine boolesche Variable. Bei steigender Flanke der Variable wird die zugewiesene Methode ein mal aufgerufen.</p>

Eigenschaft	Beschreibung Wert
„Zugriffsrechte“	<p>Mit Klick öffnet <a href="#">„Zugriffsrechte“</a>.</p> <p>Statusmeldungen:</p> <ul style="list-style-type: none"> <li>▪ Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.</li> <li>▪ Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.</li> </ul>



## Wartesymbol Würfel

Dieses animierte Element kann verwendet werden, um zu signalisieren, dass das System beschäftigt ist oder auf Daten wartet.



Abb. 120: Wartesymbol Würfel

### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Wartesymbol Würfel

### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„ <i>X</i> “	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„ <i>Y</i> “	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„ <i>Breite</i> “	Horizontale Breite des Elements.
„ <i>Höhe</i> “	Vertikale Höhe des Elements.

### Zustandsvariablen

Property	Value description
„ <i>Unsichtbarkeit</i> “	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.

Eigenschaft	Beschreibung Wert
„ <i>Zugriffsrechte</i> “	Mit Klick öffnet <a href="#">„Zugriffsrechte“</a> .

Statusmeldungen:

- Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das

Element für alle Gruppen bedienbar angezeigt wird.

- Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.

### Wartesymbol Blume

Dieses animierte Element kann verwendet werden, um zu signalisieren, dass das System beschäftigt ist oder auf Daten wartet.



Abb. 121: Wartesymbol Blume

### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Wartesymbol Blume

### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„ <i>X</i> “	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„ <i>Y</i> “	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„ <i>Breite</i> “	Horizontale Breite des Elements.
„ <i>Höhe</i> “	Vertikale Höhe des Elements.

### Farben

Eine Farbe wird mittels Hexzahl definiert, die sich aus Rot/Grün/Blau (RGB) Anteilen zusammensetzt. Für jede dieser drei Farben sind 256 (0-255) Werte verfügbar. Hier sind die Einstellungen **statisch**. Die Farbe kann aus einer

Auswahlliste oder dem Farbauswahldialog, der über Button  öffnet, gewählt werden. Beispiel:  0; 128; 64

Eigenschaft	Beschreibung Wert
„Rahmenfarbe“	Auswahl einer Rahmenfarbe
„Füllfarbe“	Auswahl einer Füllfarbe

#### Aussehen

Eigenschaft	Beschreibung Wert
„Linienstärke“	Stärke der Rahmenlinie in Pixel. 0 kodiert das Gleiche wie 1 und setzt die Linienstärke auf 1 Pixel. Wird kein Rahmen gewünscht, muss man „Linienart“ auf unsichtbar setzen.
„Füllart“	Auszuwählen ist eines dieser Füllarten angewendet in „Füllfarbe“ <ul style="list-style-type: none"> <li>▪ Ausgefüllt: Die Füllfarbe ist sichtbar</li> <li>▪ Unsichtbar: Die Füllfarbe ist unsichtbar</li> </ul>
„Linienart“	Auszuwählen ist eine der Linienarten angewendet bei der Umrisslinie: <ul style="list-style-type: none"> <li>▪ Durchgezogen</li> <li>▪ Striche</li> <li>▪ Punkte</li> <li>▪ Strich Punkt</li> <li>▪ Strich Punkt Punkt</li> <li>▪ Unsichtbar: Umrisslinie ist nicht sichtbar.</li> </ul>

Eigenschaft	Beschreibung Wert
„Symbolfarbe“	Auswahl einer Farbe, mit der das Blumensymbol angezeigt werden soll
„Symbollinienstärke“	Angabe einer Zahl, die die Stärke der Symbollinien in Pixel angibt. 0 kodiert das gleiche wie 1 und setzt die Linienstärke auf 1 Pixel.

#### Zustandsvariablen

Property	Value description
„Unsichtbarkeit“	Eingabe einer booleschen Variable; ist der Variablenwert TRUE, dann ist das Element im Online-Modus unsichtbar.

Eigenschaft	Beschreibung Wert
„Zugriffsrechte“	Mit Klick öffnet <a href="#">„Zugriffsrechte“</a> .

Statusmeldungen:

- Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.
- Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist

gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.

### Texteditor

Das Visualisierungselement „*Text Editor*“ wird verwendet zum Anzeigen und Editieren des Inhalts von Textdateien in ASCII oder Unicode, die sich auf der Steuerung befinden.

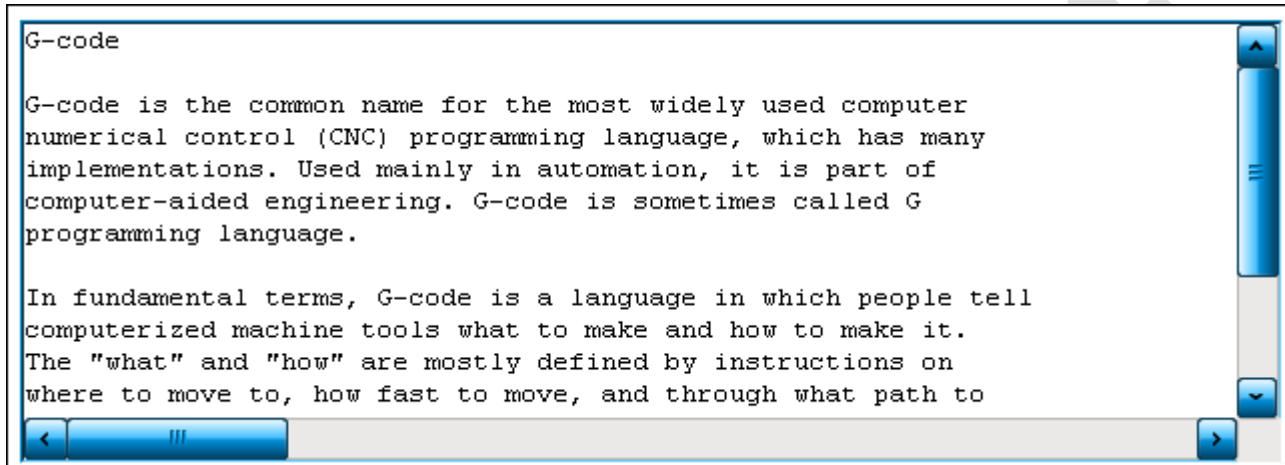


Abb. 122: Texteditor

#### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungseditor werden folgende Benutzereingaben zur Verfügung gestellt:

#### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

#### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften mit ihrer Beschreibung aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.
„ <i>Elementtyp</i> “	Text-Editor

#### Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
-------------	-------------------

- „X“ Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
- „Y“ Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
- „Breite“ Horizontale Breite des Elements.
- „Höhe“ Vertikale Höhe des Elements.

#### Schriftart

Eigenschaft	Beschreibung Wert
„Name“	Hier ist der Name einer auf dem System installierten Schriftart anzugeben, in der dann der Text im Texteditor ausgegeben wird. Dabei muss eine nicht proportionale Schriftart wie Courier New verwendet werden.
„Größe“	Definition der festen Größe des Schriftgrads, zum Beispiel 12.

#### Steuervariablen: Datei

Eigenschaft	Beschreibung Wert
„Dateiname“	Hier ist eine STRING-Variable zuzuweisen, die den Dateinamen und, falls erforderlich, den Pfad enthält. Pfad ist standardmäßig das Installationsverzeichnis der Steuerung, zum Beispiel: "..\GatewayPLC "
„Öffnen“	Hier ist eine boolsche Variablen zuzuweisen um das Öffnen der in „Dateiname“ bestimmten Datei zu steuern. Wird die Variable TRUE, öffnet sich die Datei.
„Schließen“	Zuweisen eine boolsche Variablen um das Schließen der Datei zu steuern. Wird die Variable TRUE, schließt sich die Datei.
„Speichern“	Zuweisen eine boolsche Variablen um das Speichern der Datei zu steuern. Wird die Variable TRUE, wird Datei gespeichert.
„Neu“	Zuweisen eine boolsche Variablen um das Erzeugen einer neuen Datei zu steuern. Wird die Variable TRUE, öffnet eine neue Datei mit dem in „Dateiname“ bestimmten Namen.

#### Steuervariablen: Bearbeiten

Eigenschaft	Beschreibung Wert
„Suchen nach“	Hier ist eine STRING-Variable zuzuweisen, die den Suchbegriff enthält.
„Finden“	Hier ist eine boolsche Variable zuzuweisen, die das Suchen nach dem Suchbegriff steuert. Wird die Variable TRUE, startet das Suchen.
„Nächstes Vorkommen finden“	Zuweisen einer boolsche Variable die das Suchen nach dem nächsten Vorkommen des Suchbegriffs steuert. Wird die Variabel TRUE, startet das Suchen an der aktuellen Position.

#### Steuervariablen: Caret-Position

Eigenschaft	Beschreibung Wert
„Zeile“	Hier ist eine Integer-Variable zuzuweisen, die solange „Setzen auslösen“ FALSE ist die die

aktuelle Zeilennummer enthält.

- „*Spalte*“ Hier ist eine Integer-Variable zuzuweisen, die solange „*Setzen auslösen*“ FALSE ist die die aktuelle Spaltennummer enthält.
- „*Position*“ Hier ist eine Integer-Variable zuzuweisen, die solange „*Setzen auslösen*“ FALSE ist die die aktuelle Position des Carets in fortlaufender Nummerierung enthält.
- „*Setzen auslösen*“ Hier ist eine boolsche Variable zuzuweisen, die die Caret-Position steuert. Ist die Variable FALSE, enthalten die Variablen in „*Zeile*“, „*Spalte*“ und „*Position*“ die aktuelle Position. Ist die Variable TRUE, wird das Caret an die Position gesetzt, die in „*Zeile*“ und „*Spalte*“ angegeben ist.

#### Steuervariablen: Selektion

Eigenschaft	Beschreibung Wert
„ <i>Start-Position</i> “	Hier ist eine Integer-Variable zuzuweisen, die die Startposition der Textselektion in fortlaufender Nummerierung enthält.
„ <i>Ende-Position</i> “	Zuweisen einer Integer-Variablen, die die Endposition der Textselektion in fortlaufender Nummerierung enthält.
„ <i>Start-Zeilennummer</i> “	Zuweisen einer Integer-Variablen, die die Zeilennummer am Anfang der Textselektion enthält.
„ <i>Start-Spaltenindex</i> “	Zuweisen einer Integer-Variablen, die den Spaltenindex am Anfang der Textselektion enthält.
„ <i>Ende-Zeilennummer</i> “	Zuweisen einer Integer-Variablen, die die Zeilennummer am Ende der Textselektion enthält.
„ <i>Ende-Spaltenindex</i> “	Zuweisen einer Integer-Variable, die den Spaltenindex am Ende der Textselektion enthält.
„ <i>Zu selektierende Zeile</i> “	Zuweisen einer Integer-Variablen, die eine zu selektierende Zeile bestimmt.
„ <i>Selektion auslösen</i> “	Zuweisen einer boolschen Variablen um die Textselektion zu steuern. Wird die Variable TRUE, wird die Textselektion, wie in „ <i>Zu selektierende Zeile</i> “ definiert, gesetzt.

#### Steuervariablen: Fehlerbehandlung

Eigenschaft	Beschreibung Wert
„ <i>Variable für Fehlernummer</i> “	Hier ist eine Integer-Variable zuzuweisen, die die Fehlernummer im Fehlerfall enthält. Die Nummern sind in <code>GVL_ErrorCodes</code> , Teil der Bibliothek <code>VisuElemTextEditor</code> , deklariert. Um den Fehlertext zur Fehlernummer zu erhalten, muss die Funktion <code>VisuFctTextEditorGetErrorText()</code> der Bibliothek aufgerufen werden. Siehe <a href="#">Beispiel „Fehlermeldungen“</a>

#### Steuervariablen:

- „*Variable für Inhalt geändert*“ Zuweisen einer boolschen Variable. Ist der Variablenwert TRUE, hat sich der Inhalt des Texteditors geändert.
- „*Variable für*“ Zuweisen einer boolschen Variablen. Ist der Variablenwert TRUE, hat die Datei nur

- Zugriffsmodus* Leserecht. Ist der Variablenwert FALSE, hat die Datei Schreib-/Leserecht.
- „Maximale Zeilenlänge“* Eingabe einer ganzen Zahl, die die maximale Länge einer Zeile angibt.
- „Editiermodus“* Wählen aus
  - Nur lesend: Nur-Lese-Zugriff auf die Datei. In diesem Modus wird der Editor hellgrau unterlegt dargestellt, wenn die Steuerung läuft.
  - Lesen/Schreiben: Lese-/Schreibzugriff auf die Datei.

## Neue Dateien

- „Zeichencodierung“* Wird eine neue Datei erzeugt, ist hier die Zeichencodierung festzulegen:
  - ASCII
  - Unicode (Little endian)
  - Unicode (Big endian)

- „Zeilenendzeichen“* Wird eine neue Datei erzeugt, ist hier das Zeilenendzeichen zu definieren:
  - CR/LF: üblich in Windows-Systemen
  - LF: üblich in Unix-Systemen

Wird ein bestehendes File geöffnet, wird das Zeilenendzeichen der öffnenden Datei identifiziert und automatisch verwendet.

Eigenschaft	Beschreibung Wert
-------------	-------------------

*„Zugriffsrechte“* Mit Klick öffnet [„Zugriffsrechte“](#).

Statusmeldungen:

- Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.
- Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.

## Benutzereingaben beim Navigieren im Texteditor im Online-Modus

Der Texteditor erlaubt allgemein bekannte Benutzereingaben um im Text zu navigierten während die Visualisierung läuft.

Benutzereingaben	Beschreibung
------------------	--------------

<a href="#">[ Pfeil links ]</a>	Das Caret wird um ein Zeichen nach links verschoben. Ist die aktuelle Caretposition ein Zeilenanfang, wird es ans Ende der vorigen Zeile gesetzt, falls es diese gibt.
<a href="#">[ Pfeil rechts ]</a>	Das Caret wird um ein Zeichen nach rechts verschoben. Ist die aktuelle Caretposition ein Zeilenende, wird es an den Anfang der nächsten Zeile gesetzt, falls es diese gibt.
<a href="#">[ Pfeil aufwärts ]</a>	Das Caret wird in die vorige Zeile verschoben.
<a href="#">[ Pfeil abwärts ]</a>	Das Caret wird in die nächste Zeile verschoben.
<a href="#">[ Pos1 ]</a>	Das Caret wird an den Anfang der aktuellen Zeile verschoben.
<a href="#">[ Ende ]</a>	Das Caret wird an das Ende der aktuellen Zeile verschoben.
<a href="#">[ Bild ]</a>	Die vorige Seite wird angezeigt.
<a href="#">[ Bild ↓ ]</a>	Die nächste Seite wird angezeigt.
<a href="#">[ Umschalttaste ] + [ Pfeil links ]</a>	Textselektion
<a href="#">[ Umschalttaste ] + [ Pfeil rechts ]</a>	
<a href="#">[ Entf ]</a> bei selektiertem Text	Der selektierte Text wird entfernt. Ist nichts selektiert, wird das Zeichen rechts der Caretposition gelöscht.
<a href="#">[ Eingabe ]</a>	Eine neue Zeile wird erzeugt und das Caret wird an deren Anfang gesetzt.
<a href="#">[ Strg ] + [ Tab ]</a>	Ein Tabulatorzeichen wird eingefügt. Aktuell wird es als einzelnes Blank dargestellt.
<a href="#">[ Tab ]</a>	Der Fokus wird zur nächsten Visualisierung verschoben.

## Texteditor mit Steuerelementen

Um Zugriff auf die vom Texteditor unterstützten Steuervariablen zu erhalten, ist es notwendig, ein zusätzliches Visualisierungselement der Visualisierung hinzuzufügen und es mit dem Texteditor über Steuervariablen zu verbinden.

### Steuerelement "Load"

▲Abb. 123: Visualisierung mit Texteditor und einigen Steuervariablen

Um das Visualisierungselement "Load" mit dem Texteditorelement zu verbinden, ist folgendes zu tun:

1. Deklarieren der Steuervariablen des Elements "Load" in IEC-Code in zum Beispiel PLC\_PRG:  

```
bOpen : BOOL;
```
2. Deklarieren der Variablen für „Dateiname“ des Texteditors in IEC-Code in zum Beispiel PLC\_PRG:  

```
stFileName : STRING := 'MyText.txt'
```
3. Hinzufügen eines Rechteckelements zur Visualisierung und dessen Konfiguration:  
*Texte* → *Text*: Load



[Inputkonfiguration](#) → [Umschalten](#) → [Variable](#): PLC\_PRG.bOpen

4. Hinzufügen eines Texteditorelements zur Visualisierung und dessen Konfiguration:

[Steuervariablen](#) → [Datei](#) → [Dateinamen](#): PLC\_PRG.stFileName

[Steuervariablen](#) → [Datei](#) → [Öffnen](#): PLC\_PRG.bOpen

Alle Steuervariablen, die in den Eigenschaften des Texteditors vorhanden sind, können so angebunden werden.

## Fehlermeldungen

Um den Fehlertext der Fehlernummer, die in [Steuervariablen](#) → [Fehlerbehandlung](#) → [Variable für Fehlernummer](#) zur Verfügung steht, auszugeben, ist der Funktionsaufruf `VisuFctTextEditorGetErrorText()` in IEC-Code zu programmieren.

1. Deklarieren der erforderlichen Variablen:

```
usiErrorCode : USINT;  
stErrorMsg   : STRING(255);
```

2. Implementieren des Funktionsaufrufs:

```
stErrorMsg := VisuFctTextEditorGetErrorText(usiErrorCode);
```

3. Ausgabe des Fehletextes in der Visualisierung zum Beispiel in einem Rechteckelement mit folgenden Eigenschaften:

[Texte](#) → [Text](#): %s

[Textvariablen](#) → [Textvariable](#): POU.stErrorMsg

## Path3D

Ein „*Path3D*“ kann eine **3D-Bahn** (bestehend aus einer Anzahl von Punkten, die mit Linien verbunden sind) anzeigen. Das Element ist speziell dafür ausgelegt, eine Bewegung darzustellen, die durch ein Maschinenwerkzeug oder einen Roboter ausgeführt wird. Im Falle eines Maschinenwerkzeugs/CNC-Editors, zeigt das Path3D den idealen Bahnverlauf, aber das Element kann auch für andere Applikationen eingesetzt werden.

Neben der Bahn wird ein ähnliches Daten-Array von Punkten dargestellt, genannt **Spur**. Im Falle eines Maschinenwerkzeugs/CNC-Editors zeigt die Spur den tatsächlichen Bahnverlauf, den der TCP/Roboter abgefahren ist.

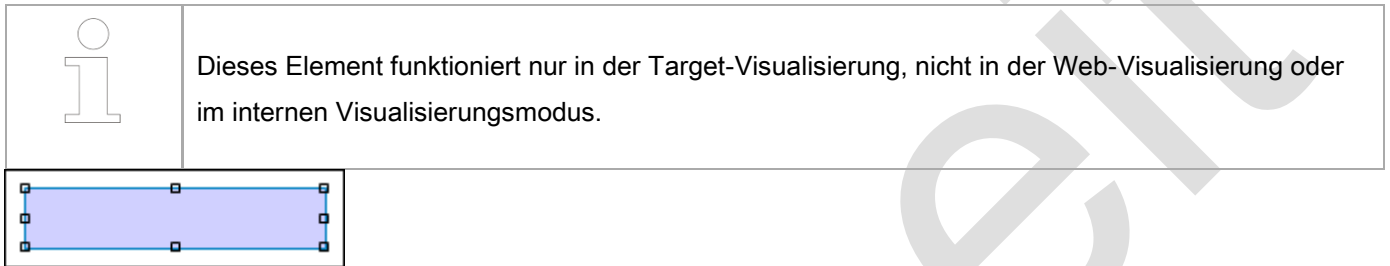
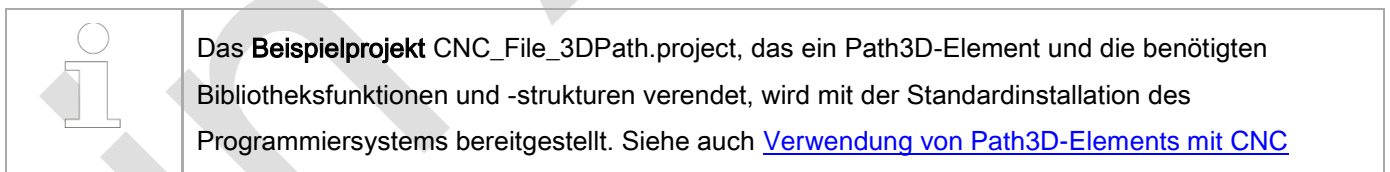


Abb. 124: Path3D

Die Datenstruktur `VisuStruct3DTrack`, Teil der Bibliothek `VisuElem3DPath`, beschreibt eine Bahn oder Spur durch eine bestimmte Anzahl von Punkten, bestimmt und gepuffert über die Applikation. Die Spur zeigt typischerweise die letzten *n* Positionen an, so dass immer nur ein bestimmter Teil von ihnen dargestellt wird.

`VisuStruct3DTrack.pProjection` ist eine Variable, die durch das Visualisierungselement gesetzt wird und Informationen über die Bahn-/Spurprojektion enthält. Sie kann von der Applikation (nur) gelesen werden. Außerdem können die Methoden `Projection.Apply` oder `.ApplyV` verwendet werden, um zu sehen, ob die transformierte Position innerhalb oder außerhalb des Visualisierungsanzeigebereichs liegt, der durch `Projection.ElementRect` definiert ist. Über die Struktur `VisuStruct3DControl` kann die Applikation die Kameraposition für den 3D-Modus kontrollieren.



### ▲ Benutzereingabe beim Editieren eines Elements

Beim Editieren eines Elements im Visualisierungseeditor werden folgende Benutzereingaben zur Verfügung gestellt:

### ▲ Benutzereingabe beim Editieren der 'Eigenschaften'

Eigenschaften definieren die Eigenart eines Visualisierungselements:

### Elementeigenschaften

Im Folgenden werden alle Elementeigenschaften und deren Beschreibungen aufgelistet.

Eigenschaft	Beschreibung Wert
„ <i>Elementname</i> “	Der Elementname kann geändert werden. Standardname ist "GenElemInst_x", "x" steht für eine fortlaufende Nummerierung.

## Position

Hier ist die Position (X/Y-Koordinaten) und Größe (Breite und Höhe) des Elements jeweils in Pixel zu definieren. Der Ursprung liegt in der oberen linken Fensterecke. Die positive X-Achse verläuft nach rechts, die positive Y-Achse verläuft nach unten. Werden die Werte editiert, wird gleichzeitig das angezeigte Element im Editor geändert.

Eigenschaft	Beschreibung Wert
„ <i>X</i> “	Definiert die horizontale Position. X=0 ist der linke Fensterrand. Die positive Achse zeigt nach rechts.
„ <i>Y</i> “	Vertikale Position. Y=0 ist der obere Fensterrand. Die positive Achse zeigt nach unten.
„ <i>Breite</i> “	Horizontale Breite des Elements.
„ <i>Höhe</i> “	Vertikale Höhe des Elements.

## Bahnbeschreibung

Die folgende Datenstruktur, definiert in der Bibliothek VisuElem3DPath, enthält eine Bahn oder eine Spur:

```
FUNCTION_BLOCK VisuStruct3DTrack
VAR_INPUT
    pData: POINTER TO ARRAY[0..0] OF VisuStruct3DPathPoint;
    udiNumberOfPointsInArray: UDINT;
    udiFirstPoint: UDINT;
    udiNumberOfPointsToDraw: UDINT;
    pProjection: POINTER TO Projection;
END_VAR
```

pData zeigt auf einen Array von VisuStruct3DPathPoint-Elementen. Dieser Array muss wenigstens udiNumberOfPointsInArray Elemente enthalten. Der Array wird von der Applikation alloziert und wird auch von dieser gesetzt.

Ein Punkt wird folgendermaßen beschrieben:

```
TYPE VisuStruct3DPathPoint :
STRUCT
    v: Vector3;
    udiSourceElementID: UDINT;
    dwAddInfo: DWORD;
END_STRUCT
END_TYPE
```

v enthält die x, y und z Position des Punktes.

udiSourceElementID wird zur Hervorhebung verwendet und verweist auf das ursprüngliche Objekt.

`dwAddInfo` markiert den Punkt als

- `Bit0=TRUE`: Es zeigt den Start- oder Endpunkt des Objekts an.
- `Bit1=TRUE`: Bei diesem Punkt sollte die Polylinie, die die Bahn anzeigt, enden. Zwischen diesem Punkt und dem nächsten wird keine Linie gezeichnet. Dies kann verwendet werden, wenn die Bahn einen Spalt enthält oder wenn eine optimierte Struktur realisiert werden soll, die für eine Projektion sorgt und nur sichtbare Teile der Bahn/Spur angezeigt werden sollen. In diesem Fall ist es möglich, dass die Bahn/Spur den sichtbaren Bereich verlässt. Dann ist die Bahn/Spur keine kohärent Linie, sondern wird unterbrochen.
- Alle weiteren Bits sind reserviert für künftige Anwendungen.

`diNumberOfPointsInArray` enthält die Größe des Arrays auf den `pDate` zeigt

`udiFirstPoint` ist die Position des ersten Punktes der Bahn oder der Spur. 0 bedeutet, dass `pData^[0]` der erste Punkt ist:

```
0 <= udiFirstPoint < udiNumberOfPointsInArray
```

`udiNumberOfPointsToDraw` definiert, wie viele Punkte gültig sind und gezeichnet werden müssen:

```
0 <= udiNumberOfPointsToDraw <= udiNumberOfPointsInArray
```

Der Datenarray ist ein Ringpuffer. Die Punkte werden einer nach dem anderen im Array hinzugefügt während `udiNumberOfPointsToDraw` um eins erhöht wird. Ist `udiNumberOfPointsToDraw` gleich `udiNumberOfPointsInArray`, überschreibt der Daten-Provider wieder den ersten Punkt. `udiNumberOfPointsInArray` wird nicht länger inkrementiert. Aber `udiFirstPoint` muss inkrementiert werden, weil der älteste Punkt der ist, der eine Position nach der zuletzt generierten liegt.

Rückmeldung der Projektion von Bahn/Spur wird übernommen von: `VisuStruct3DTrack`. Das ist eine Variable die vom Visualisierungselement gesetzt wird. Die Applikation kann dies nur lesen. Es enthält Information der Projektion: bei Verwendung der Methoden `Projection.Apply` oder `Projection.ApplyV` kann man die transformierte Position sehen, die innerhalb oder außerhalb des Visualisierungsbereichs liegt, die von `Projection.ElementRect` beschrieben wird.

Eigenschaft	Beschreibung Wert
<i>„Bahndaten“</i>	Weisen Sie eine Variable vom Typ Pointer to Array of <code>VisuStruct3DTrack</code> zu, die im IEC-Code deklariert ist; Beispiel: <code>PLC_PRG.pc.vs3dp</code> . Die Struktur steht über die <code>VisuElem3DPath.library</code> bereit.
<i>„Bahnfarbe“</i>	Bahndarstellungsfarbe, über die Auswahlliste oder die Farbdefinitionstafel zu definieren
<i>„Bahnliniendicke“</i>	Bahnliniendicke in Pixel; Beispiel: "2"
<i>„Randpunkt-Markierung“</i>	Darstellung der Punkte zwischen zwei aufeinanderfolgenden Objekten in der Bahn: <ul style="list-style-type: none"><li>▪ Endpunkte werden nicht markiert</li></ul>

- Endpunkte werden mit einem Kreis markiert
- Endpunkte werden mit einem Kreuz markiert
- Endpunkte werden mit einem Pluszeichen markiert

## Spurbeschreibung

Die Spurdaten sind genauso strukturiert wie die Bahndaten: VisuStruct3DTrack ([↘ „Bahnbeschreibung“](#)).

Eigenschaft	Beschreibung Wert
„ <i>Spurdaten(VisuStruct3DTrack)</i> “	Weisen Sie eine Variable vom Typ Pointer to Array of VisuStruct3DTrack zu, die im IEC-Code deklariert ist; Beispiel: PLC_PRG.pc.vs3dt. Die Struktur steht über die VisuElem3DPath.library bereit.
„ <i>Spurfarbe</i> “	Spurdarstellungsfarbe, über die Auswahlliste oder die Farbdefinitionstafel zu definieren
„ <i>Spurliniendicke</i> “	Spurliniendicke in Pixel; Beispiel: "2"

## Kamerasteuerung

### Steuerung und Navigation

Die Kameraposition für den 3D-Modus wird mit einer Referenz auf eine externe Datenstruktur gesteuert, die folgende Operationen erlaubt:

- Links/Rechts/Nach oben/Nach Unten verschieben
- Drehen um X/Y/Z Achse
- Rücksetzen der Ansicht auf X/Y-, Y/Z- oder Z/X-Ebene, so dass die Bahn und die Spur vollständig enthalten ist.

Diese Datenstruktur wird in der Applikation alloziert und deren Werte können in der Applikation selbst oder in einem HMI-Element gesetzt werden.

Die steuernde Datenstruktur ist wie folgt im IEC-Code definiert:

```
TYPE VisuStruct3DControl :
STRUCT
    iX, iY, iZ, iTurnX, iTurnY, iTurnZ: INT;
    xResetXY, xResetYZ, xResetZX: BOOL;
END_STRUCT
END_TYPE
```

Die Werte iX, iY, iZ, iTurnX, iTurnY und iTurnZ bestimmen die Geschwindigkeit, mit der das gesamte Bild in X/Y/Z-Richtung verschoben und die Geschwindigkeit mit der die Bahn/Spur um die X/Y/Z-Achse gedreht wird. Wert 0 bedeutet keine Änderung, Wert -32768 oder 32767 führt die Bewegung/Drehung mit maximaler Geschwindigkeit aus. Ist die boolsche Variable TRUE, dann wird die Ansicht so geändert, dass die vollständige Bahn/Spur in die

Ansicht passt und die spezifizierte Ebene komplett sichtbar ist.

Eigenschaft	Beschreibung Wert
„ <i>Steuerungsdatenstruktur(VisuStruct3DControl)</i> “	Weisen Sie eine Variable vom Typ <code>POINTER TO ARRAY OF VisuStruct3DControl</code> zu, wie sie im IEC-Code deklariert ist; Beispiel: <code>PLC_PRG.pc.vs3dc</code> . Die Struktur ist Teil der Bibliothek <b>VisuElem3DPath</b> und enthält die Information über die Kameraposition für den 3D-Modus. Dessen Werte können über die Applikation selbst oder über ein HMI-Element gesetzt werden. <b>VisuElem3DPath</b> enthält fertig benutzbare Visualisierungsframes, die eine mögliche Benutzerschnittstelle für diese Daten zur Verfügung stellt.

#### Zusätzliche Aspekte

Eigenschaft	Beschreibung Wert
„ <i>Koordinatensystem</i> “	Wenn ein Koordinatensystem angezeigt werden soll, aktivieren Sie diese Option.
„ <i>Gitter</i> “	Wenn ein Gitter angezeigt werden soll, aktivieren Sie diese Option.
„ <i>Gitterfarbe</i> “	Gitterfarbe, über die Auswahlliste oder die Farbdefinitionstafel zu definieren

#### Hervorhebung

Einzelne Teile der Bahn können hervorgehoben werden. Jeder Punkt der Spur bekommt eine zufällige ID, die im Falle eines CNC-Editors mit der Objekt-ID, auf der der Punkt liegt, verknüpft ist. Diese ID ("highlight ID") kann über die Applikation angegeben werden, um zu definieren, welches Element/Teil der Spur hervorgehoben werden soll.

Eigenschaft	Beschreibung Wert
„ <i>Hervorhebungsart</i> “	Wählen Sie eine der folgenden Hervorhebungsarten: <ul style="list-style-type: none"> <li>Nur das Element mit "highlight ID" (im Falle eines CNC-Editors verknüpft mit der Objekt-ID) wird in der Hervorhebungsfarbe dargestellt</li> <li>Alle Elemente vor dem Element mit der "highlight ID" und das Element selbst werden in der Hervorhebungsfarbe dargestellt</li> </ul>
„ <i>Variable</i> “	Projektvariable, die die ID eines Elements angibt. Beispiel: <code>PLC_PRG.iVarElementID</code> . Diese "highlight ID" wird für die Einstellung der Hevorhebungsart berücksichtigt, siehe oben. Die Variable muss in der IEC-Applikation gesetzt werden.
„ <i>Hervorhebungsfarbe</i> “	Hervorhebungsfarbe, über die Auswahlliste oder die Farbdefinitionstafel zu definieren

#### Elementdarstellung

Eigenschaft	Beschreibung Wert
„Rahmenliniendicke“	Dicke des Rahmens um das Element, in Pixel; Beispiel: "1"
„Rahmenlinienstil“	Wählen Sie einen dieser Stiltypen für die Rahmenlinie: <ul style="list-style-type: none"> <li>▪ Durchgezogen</li> <li>▪ Striche</li> <li>▪ Punkte</li> <li>▪ Strich Punkt</li> <li>▪ Strich Punkt Punkt</li> <li>▪ Unsichtbar</li> </ul>
„Hintergrund transparent“	Aktivieren Sie diese Option, wenn der Hintergrund transparent erscheinen soll.
„Hintergrundfarbe“	Hintergrundfarbe, über die Auswahlliste oder die Farbdefinitionstafel zu definieren

Eigenschaft	Beschreibung Wert
„Zugriffsrechte“	Mit Klick öffnet <a href="#">„Zugriffsrechte“</a> .  Statusmeldungen: <ul style="list-style-type: none"> <li>▪ Nicht gesetzt. Alle Rechte: Die Defaultmeldung ist gesetzt, wenn das Element für alle Gruppen bedienbar angezeigt wird.</li> <li>▪ Rechte sind vergeben: eingeschränkte Rechte: Die Meldung ist gesetzt, wenn das Element für mindestens eine Gruppen mit eingeschränktem Verhalten angezeigt wird.</li> </ul>

### SoftMotion unabhängige Verwendung

Dieses Element ist ausgelegt für eine Verwendung mit CODESYS SoftMotion CNC. Es kann aber auch unabhängig verwendet werden. In diesem Fall muss die Applikation die Bahndaten zur Verfügung stellen. Die Beispielapplikation "3D Path example", die im CODESYS Store zu haben ist, zeigt, wie dies erreicht werden kann. Wird dieses Elemente zusammen mit CODESYS SoftMotion CNC verwendet, dann helfen einige Funktionsbausteine der Bibliothek **SM3\_CNC\_Visu** die Daten der Bahn und Spur zur Verfügung zu stellen.

### Verwendung von Path3D mit CNC basierend auf SM3\_CNC\_Visu

Funktionsbausteine um Visulisierungsdaten zu erhalten

Die folgenden Bausteine werden dafür verwendet, um aus einer Bahn der CODESYS SoftMotion CNC ein Punktearray zu bekommen, welches mit dem Visu-Element Path3D dargestellt werden kann.

Folgende Eingänge der Bausteine haben überall gleiche Bedeutung

- `bExecute`: TRUE startet den Baustein
- `bAbort`: TRUE beendet ihn.
- `udiNumberOfPointsInArray`: Anzahl der Punkte, die in `pBuffer` im referenzierten Array deklariert wurden.
- `pBuffer`: Zeiger auf den angelegten Punktearray
- `poqCNCPath`: Zeiger auf `SMC_OutQueue`, die die Bahn beschreibt und aus anderen CNC-Bausteinen (z.B. `SMC_NCDecoder`) kommt.
- Der Ausgang `vs3dt` enthält die Bahnbeschreibung und wird in der Regel dem Visualisierungselement als Bahn (`vtrPath`) übergeben.

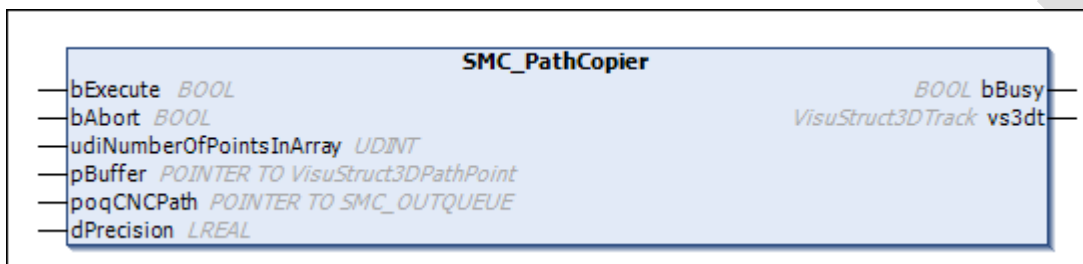


Abb. 125: SMC\_PathCopier

Der Funktionsbaustein SMC\_PathCopier ist dafür ausgelegt, eine Queue, die aktuell vom Interpolator befahren wird, darzustellen. Er reproduziert den Inhalt der Eingangswarteschlange `poqCNCPath` in die Ausgangsbahn `vs3dt` und beginnt immer beim ersten Element der Queue. Im Fall eines linearen Elements wird nur der Start- und Endpunkt zur Bahn `vs3dt` hinzugefügt. Im Falle eines gebogenen Elements werden diese Elemente interpoliert und `n` Punkte zu `vs3dt` hinzugefügt. `n` hängt von der Länge des Objekts, vom aktiven Zoomfaktor des Visuelements und dem Eingang `dPrecision` ab. SMC\_PathCopier setzt mit dem nächsten Objekt der Warteschlange fort, bis entweder ihr Ende erreicht ist oder bis die Spur `vs3dt` voll ist und kein Punkt hinzugefügt werden kann. Im letzteren Fall wartet er, bis das erste Element der Queue gelöscht wurde und folglich die entsprechenden Punkt auch gelöscht werden kann.

Der Funktionsbaustein wird typischerweise verwendet um das nächste Element, das interpoliert werden soll, anzuzeigen.

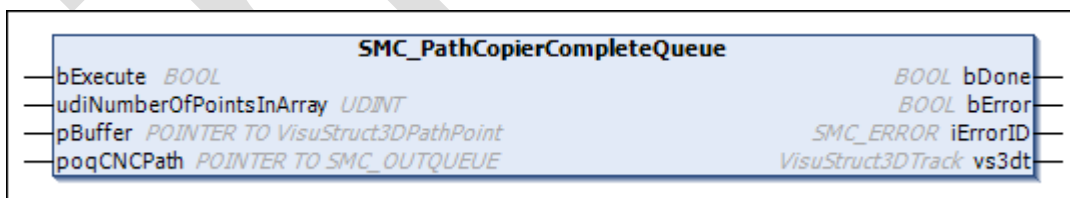


Abb. 126: SMC\_PathCopierCompleteQueue

Der Funktionsbaustein SMC\_PathCopierCompleteQueue wird verwendet, um eine Bahn, die komplett in einer Warteschlange enthalten ist, in ein Punktefeld umzuwandeln. Er kann nur angewendet werden, wenn das gesamte CNC-Programm in `poqCNCPath` passt, im Gegensatz zu SMC\_PathCopier, der verwendet wird, wenn die Warteschlange nicht vollständig in den Speicher passt.



Für jedes lineare Element wird ein Punkt verwendet. Die übrigen Punkte werden gleichmäßig auf die nichtlinearen Elemente verteilt. Fehler werden ausgegeben, wenn die Bahn nicht vollständig in der Warteschlange enthalten ist oder die Anzahl der Bahnelemente die Anzahl der Punkte überschreitet.

Der Funktionsbaustein wird verwendet um eine CNC-Bahn am präzisesten anzuzeigen. Aber er kann nur verwendet werden, wenn die Warteschlange die Bahn vollständig enthält.



Abb. 127: SMC\_PathCopierFile

Der Funktionsbaustein SMC\_PathCopierFile liest und dekodiert eine G-Code-Datei (sFileName) und erzeugt aus der so gewonnenen Bahn einen Punkte-Array. Dabei wird, wenn die Anzahl der Bahnelemente kleiner als die Anzahl der Punkte ist, wie bei SMC\_PathCopierCompleteQueue verfahren. Andernfalls werden die Punkte gleichmäßig auf die Bahn verteilt. Die Eingänge pvl und piStartPosition werden an die Funktionsbausteine [SMC\\_ReadNCFile](#) und [SMC\\_NCDecoder](#) weitergereicht; d.h. die Dokumentation dort gilt genauso für die Eingänge hier.

Der Baustein wird dazu verwendet um einen Überblick über ein vollständiges CNC-Programm zur Verfügung zu stellen. Er verwendet alle Punkte des Spurpuffers, aber im Fall eines sehr großen G-Code-Programms oder eines riesigen Zoomfaktors wird er nicht mehr präzise sein.

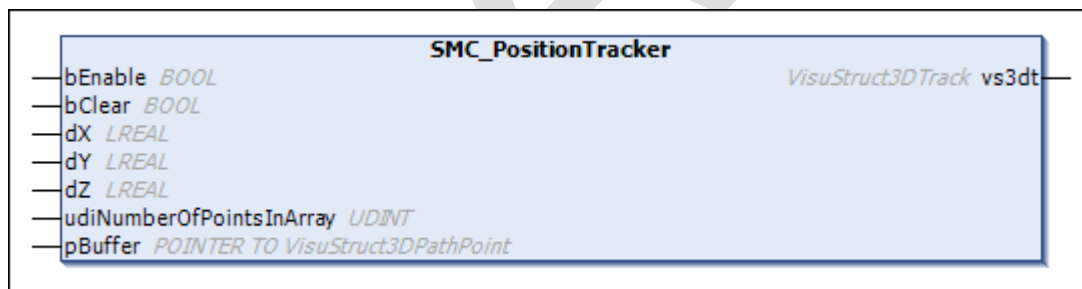


Abb. 128: SMC\_PositionTracker

Der Funktionsbaustein SMC\_PositionTracker hilft Positionen aufzuzeichnen und als Spur dem Visu-Element Path3D zur Verfügung zu stellen. Solange bEnable = TRUE ist, wird aufgezeichnet. bClear löscht den Puffer. dX, dY, dZ enthalten die Koordinate, die aufgezeichnet werden sollen.

Im Gegensatz zu den oberen Funktionsbausteinen, die Daten zur Verfügung stellen um eine ideale Kontur anzuzeigen, ist dieser Funktionsbaustein dafür ausgelegt um die tatsächliche Bahn aufzuzeichnen. Das Visuelement Path3D ist dafür ausgelegt beide zu zeigen, die ideale Bahn und die tatsächliche Spur.

### *Kurvenscheiben-Anzeiger*

Das Element Kurvenscheiben-Anzeiger, welches aus internen Gründen vorübergehend verfügbar ist, sollte nicht benutzt werden. Es wird empfohlen, die Visualisierungsvorlage des [Online Kurvenscheibeneditor](#) zu verwenden um Kurventabellen darzustellen.

in Arbeit

## 6.3.6 Optionen

### 6.3.6.1 Visualisierungen

Dieser Unterdialog des „*Optionen*“-Dialogs besteht aus den Registerblättern „*Allgemein*“ und „*Raster*“, die allgemeine Einstellungen für das Arbeiten mit Visualisierungen im Programmiersystem erlauben.

#### Allgemein

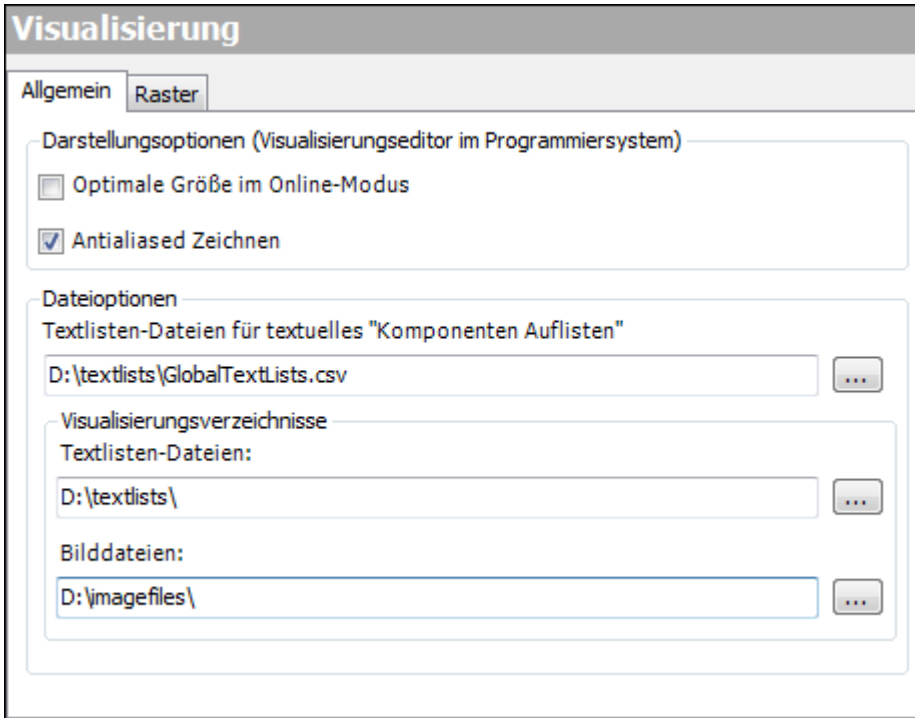
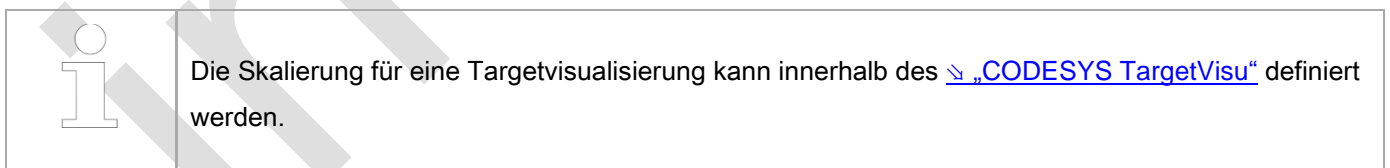


Abb. 129: Dialog für Visualisierungsoptionen, 'Allgemein'

#### „Darstellungsoptionen (Visualisierungseditor im Programmiersystem)“

„*Anpassen im Online Modus:*“: Diese Option muss aktiviert werden, wenn die Visualisierung im Online-Betrieb an die aktuelle Editorfenster-Größe angepasst werden soll, so dass immer alle Elemente sichtbar sind.



„*Antialiased Zeichnen:*“: Aktivieren Sie diese Option, wenn beim Zeichnen der Visualisierungen im Visualisierungseditor-Fenster des Programmiersystems Antialiasing verwendet werden soll (Offline oder Online).


#### „Dateioptionen“:


„*Textlistendateien für textuelles "Komponenten-Auflisten"*“: Im zugehörigen Eingabefeld kann der Name (mit Pfad) der Datei (Typ .csv) eingetragen werden, die die Vorschläge definiert, die dann von der Funktion „*Komponenten*

„*aufflisten*“ gemacht werden, wenn man in den Eigenschaften eines Visualisierungselements im Eingabefeld bei „*Texts/Text*“ eintragen will. Diese Datei kann beispielsweise aus dem Export einer globalen Textliste stammen. Die Textdatei muss als Tabelle angelegt sein, ihre Struktur muss der von Textlisten entsprechen.

„*Visualisierungsverzeichnisse*“:

Hier können Standard-Verzeichnispfade vordefiniert werden, die in „*Projekteinstellungen*“, Kategorie „*Visualisierung*“ jedoch noch projektspezifisch verändert werden können. Letztendlich bestimmend für das Projekt sind die Projekteinstellungen! [↗ „Projekteinstellungen“](#)

„*Textlistendateien*“: Öffnen Sie mit Schaltfläche  den Standarddialog zur Definition eines existierenden oder eines neuen Verzeichnisses für die in den Visualisierungen verwendeten Sprachdateien (Textlisten).

„*Bilddateien*“: Öffnen Sie mit Schaltfläche  den Standarddialog zur Definition eines existierenden oder eines neuen Verzeichnisses für die Bilddateien (Bildersammlungen), die im Projekt für die Verwendung in Visualisierungen bereitstehen sollen. Hier können mehrere Verzeichnisse in einer Strichpunkt-separierten Liste angegeben werden.

## Raster

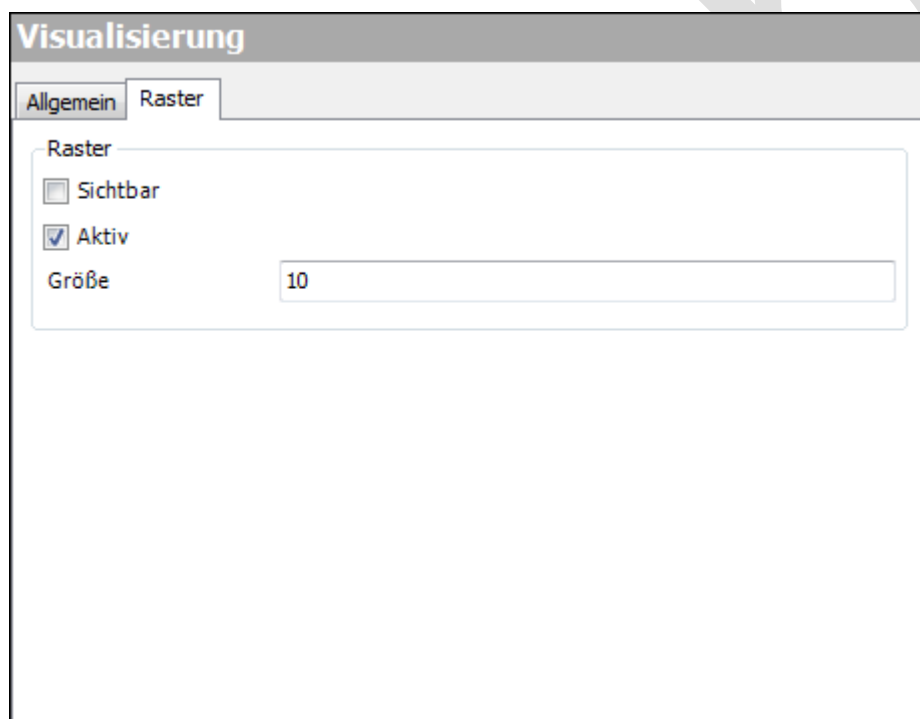


Abb. 130: Dialog für Visualisierungsoptionen, Raster

„*Sichtbar*“: Wenn diese Option aktiviert ist, wird ein Raster gemäß der unten definierten Größe als optische Orientierungshilfe im Visualisierungseditor eingeblendet.

„*Aktiv*“: Wenn diese Option aktiviert ist, können Visualisierungselemente nur auf den Punkten des unten in "Größe" definierten Rasters positioniert werden, egal ob das Raster "sichtbar" oder nur "aktiv" gesetzt ist. Beim Einfügen oder

Verschieben eines Elements bedeutet dies, dass das Zentrum des Elements nur auf einem Rasterpunkt platziert werden kann. Beim Verändern eines Elements bedeutet es, dass das kleine Rechteck an der Elementkante, das bewegt wird, um die Größe oder Form des Elements zu ändern, nur auf einen Rasterpunkt verschoben werden kann.

„*Größe*“: Vertikaler und horizontaler Abstand zwischen den Rasterpunkten in Pixel.

### 6.3.6.2 Visualisierungsstile

Dieser Unterdialog „*Visualisierungsstile*“ des Dialogs „*Optionen*“ dient der Definition des Visualisierungsstils, der in **Visualisierungen in Bibliotheken** und **globalen Visualisierungen** in Projekten verwendet werden soll.



Tippsymbol

Im Online-Modus ist nicht die aktuelle Einstellung im Optionendialog wirksam, sondern die, die im Visualisierungsmanager vorliegen. Siehe die allgemeinen Information zu den Visualisierungsstilen: [↗](#)  
„*Visualisierungsstile*“.

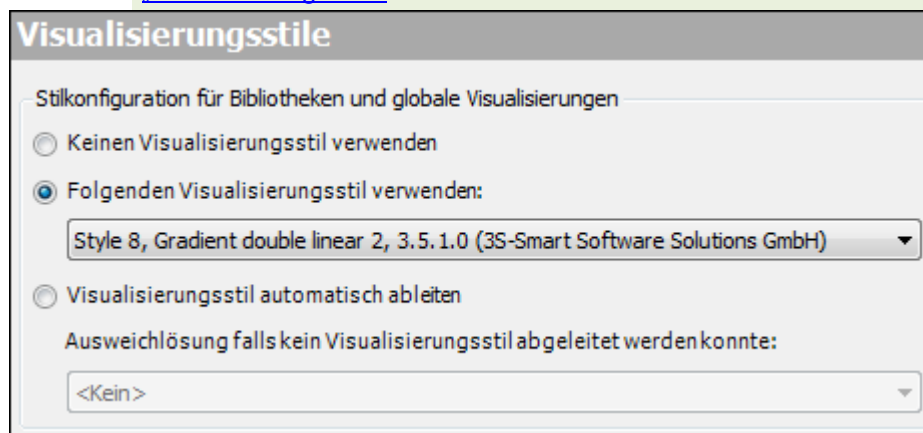


Abb. 131: Dialog 'Optionen', 'Visualisierungsstile'

„*Stilkonfiguration für Bibliotheken und globale Visualisierungen*“:

„*Keinen Visualisierungsstil verwenden*“: Wenn diese Option aktiviert ist, verhalten sich Visualisierungen in Bibliotheken oder im globalen POU's Pool so als ob kein Stil zugewiesen wurde.

„*Folgenden Visualisierungsstil verwenden*“: Wenn diese Option aktiviert ist, verhalten sich Visualisierungen in Bibliotheken oder im globalen POU's Pool gemäß dem ausgewählten Stil. Die Auswahlliste stellt alle aktuell installierten Stile bereit.

„*Visualisierungsstil automatisch ableiten*“: Wenn diese Option aktiviert ist, versuchen Visualisierungen in Bibliotheken oder im globalen POU's Pool zunächst, den im Visualisierungsmanager des aktuellen Projekts eingestellten Stil soweit als möglich anzuwenden. Wenn das nicht möglich ist, wird der in „*Ausweidlösung, falls kein Visualisierungsstil abgeleitet werden konnte*“ angegebene verarbeitet.



Tippsymbol

Die Stileinstellungen die für alle Visualisierungen der aktuellen Applikation gültig sind, müssen in [↗](#)  
„*Einstellungen*“ konfiguriert werden.

### 6.3.6.3 Visu-Benutzerverwaltung

Benutzerverwaltungskonfiguration für Bibliotheken und globale Visualisierungen

Keine Visu-Benutzerverwaltung verwenden

Folgende Benutzergruppenliste für die Visualisierung verwenden:  
(Geben Sie die Benutzergruppennamen getrennt durch Kommas in der Textbox ein.)

Admin,Service,Operator,None

Benutzerverwaltung für Visualisierung automatisch ableiten  
Ersatzfunktion, falls mehrere Visu-Benutzerverwaltungen abgeleitet werden konnten:

VisualizationManager [Device: SPS-Logik: Application]

Abb. 132: Visualisierungsbenutzerverwaltung

„Benutzerverwaltungskonfiguration für Bibliotheken und globale Visualisierungen“: Die hier ausgewählte Option wird angewendet bei Visualisierungen, die aus Bibliotheken eingebunden werden und bei globalen Visualisierungen, die sich unterhalb des POU-Baums befinden.

- „Keine Visu-Benutzerverwaltung verwenden“: Markieren sie diese Option, dann verhalten sich die betroffenen Visualisierungen als wäre keine Benutzerverwaltung konfiguriert.
- „Folgende Benutzergruppenliste für die Visualisierung verwenden“: Markieren sie diese Option, dann verwenden die betroffenen Visualisierungen die im Textfeld eingetragenen Benutzergruppen. Das Textfeld kann manuell oder mit Hilfe der Schaltfläche „Benutzergruppen für globale Visualisierungen“ aus einer konfigurierten [Benutzerverwaltung](#) heraus gefüllt werden.
- „Benutzerverwaltung für Visualisierungen automatisch ableiten“: Markieren sie diese Option, dann verwenden die betroffenen Visualisierungen die Benutzerverwaltungskonfiguration des hier ausgewählten Visualisierungsmanagers. Die Auswahlliste stellt alle tatsächlich installierten Visualisierungsmanager des Projekts zur Verfügung. Wenn dies nicht möglich ist, werden die Benutzergruppen der Option „Folgende Benutzergruppenliste für die Visualisierung verwenden“ verwendet.



Die Benutzerverwaltung einer Visualisierung **unterhalb des Gerätebaums** muss in „Benutzerverwaltung“ des zugehörigen Visualisierungsmanagers vorgenommen werden.

## 6.3.7 Projekteinstellungen

### Visualisierung

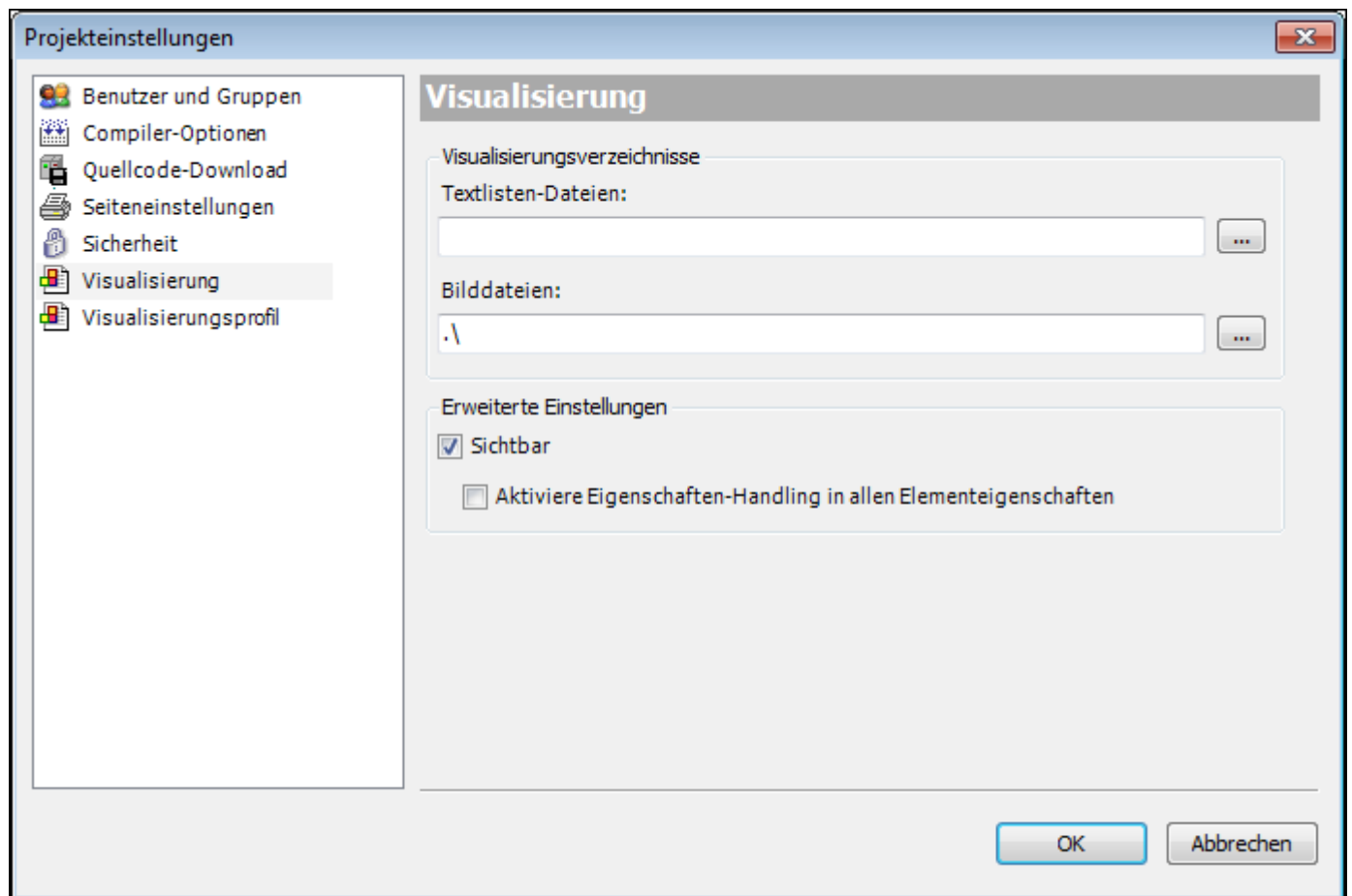




Abb. 133: Projekteinstellungen, Kategorie Visualisierung

„*Visualisierungsverzeichnisse*“: Die Standardeinstellungen für die folgenden Verzeichnisse sind durch die [Projekt-Optionen](#) vorgegeben.

„*Textlisten-Dateien*“: Öffnen Sie mit Schaltfläche  den Standarddialog zur Definition eines existierenden oder eines neuen Verzeichnisses für Sprachdateien, die im Projekt zur Text- und Sprachenkonfiguration in der Visualisierung verwendet werden sollen. Die Dateien, die in diesem Verzeichnis gefunden werden, sind dann zu dem Zeitpunkt verfügbar, wenn eine Textliste importiert werden soll, z.B. nachdem sie von einem Übersetzer extern bearbeitet wurde. Es kann hier nur **ein** Verzeichnis angegeben werden.

„*Bilddateien*“: Öffnen Sie mit Schaltfläche  den Standarddialog zur Definition eines existierenden oder eines neuen Verzeichnisses für die Bilddateien, die im Projekt für die Verwendung in Visualisierungen bereitstehen sollen. Hier können mehrere Verzeichnisse in einer Strichpunkt-separierten Liste angegeben werden.

### Visualisierungsprofil

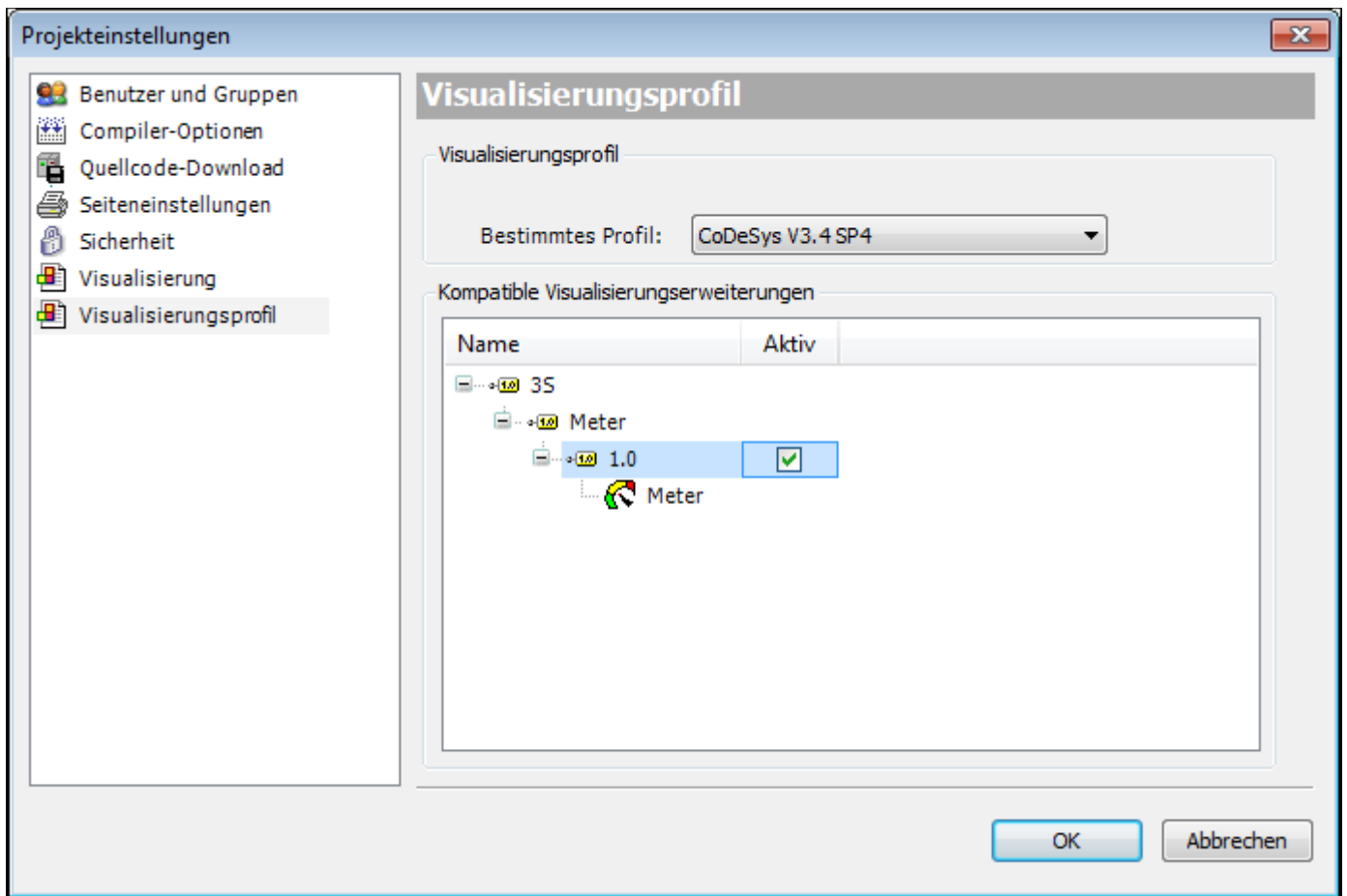


Abb. 134: Projekteinstellungen, Kategorie Visualisierungsprofil

Legen Sie hier fest, welches der aktuell auf dem System verfügbaren [„Visualisierungsprofile“](#) verwendet werden soll, wenn das Projekt geöffnet wird. Weiterhin werden alle Visualisierungserweiterungen, welche zu dem ausgewählten Profil kompatibel sind, angezeigt und können für das Projekt aktiviert werden:

*„Bestimmtes Profil“*: Das aktuell aus der Auswahlliste eingestellte Profil wird verwendet.

Wenn ein Projekt geöffnet wird, das nicht das neueste verfügbare Visualisierungsprofil verwendet, wird darauf hingewiesen und der Anwender kann im Dialog *„Projektumgebung“* eine entsprechende Aktualisierung vornehmen. Mehr Informationen sind unter [Projektumgebung](#) zu finden.

Wenn die Einstellungen auch für weitere Projekte gelten sollen, die im aktuell geöffneten Programmiersystem erstellt werden, muss die Option *„Diese Einstellung als Default für neue Projekte verwenden“* aktiviert werden.



### 6.3.8 Visualisierungsprofile

Jedes Visualisierungsprojekt, also ein Projekt, das mindestens 1 Visualisierungsobjekt enthält, **muss** auf einem Visualisierungsprofil basieren. Dieses Profil definiert folgendes:

- die Namen und Versionen der [Visualisierungsbibliotheken](#) die automatisch im Projekt eingebunden werden sobald ein Visualisierungsobjekt angelegt wird
- eine Auswahl von aus den eingebundenen Bibliotheken stammenden Visualisierungselementen, die in der [Werkzeug-Box des Visualisierungs-Editors](#) bereitstehen sollen

Die **Profilkonfiguration** wird außerhalb eines Projekts im [Visualisierungselemente-Repository](#) Editor vorgenommen. Es können mehrere Profile definiert und auf dem lokalen System abgelegt werden.

Welches **Profil standardmäßig im Projekt verwendet wird**, ist in den [„Visualisierungsprofil“](#) definiert. Dort kann auch jederzeit auf ein anderes Profil umgeschaltet werden. Beachten Sie, dass das gewählte Profil für alle Geräte und Applikationen gilt.

Wenn das Profil gewechselt wird, erscheint eine Meldung, dass dies möglicherweise ein Einloggen ohne Online Change oder Download verhindert.

Das Wechseln des Profils verursacht eine automatische Aktualisierung im Bibliotheksverwalter bezüglich der erforderlichen Bibliotheken. Diese müssen nicht händisch angepasst werden.

Ein Visualisierungsprofil muss mindestens eine **"Haupt"-Visualisierungsbibliothek** definieren, d.h. eine Bibliothek, die die Ersetzungen für die Platzhalter, die in anderen Visualisierungsbibliotheken verwendet sind, definiert. Im Falle einer Standardinstallation von 3S-Smart Software Solutions GmbH ist dies die Bibliothek **VisuElems.library**.

Wenn ein **altes Projekt** geöffnet wird, das noch nicht mit einem Visualisierungsprofil angelegt wurde, werden Sie gefragt, ob auf den neuen Profil-Mechanismus umgeschaltet werden soll oder nicht.

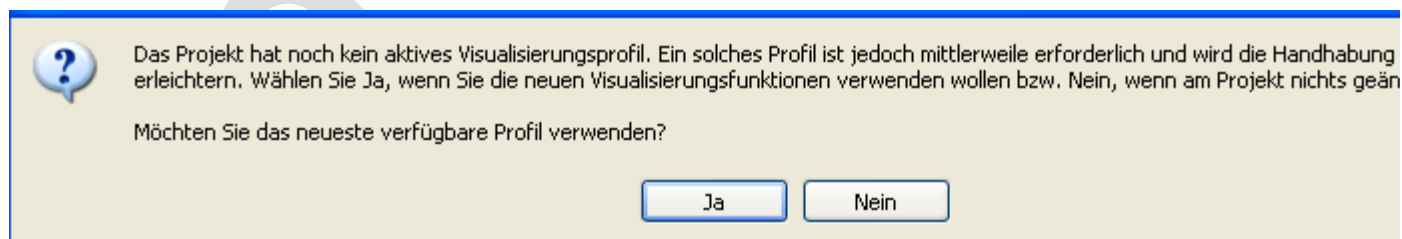


Abb. 135: Meldungsbox, falls Profil fehlt

Wenn Ja geantwortet wird, wird das ["Neueste Profil"](#) verwendet. Bei Nein wird das älteste verfügbare Profil in den Projekteinstellungen eingetragen (genannt "Kompatibilitätsprofil"), aber im Projekt werden keine weiteren Änderungen vorgenommen.

### 6.3.9 Visualisierungsbibliotheken

Da Visualisierungselemente in CODESYS V3 als Funktionsbausteine angelegt sind, werden sie als Bibliotheken bereitgestellt.

CODESYS V3 „Visualisierungsprofile“ definiert. Das Profil legt auch genau fest, welche Elemente aus diesen Bibliotheken in „*Werkzeuge*“ des Visualisierungseditors bereitstehen.

Eine Visualisierungsbibliothek ist immer als spezieller Typ einer **Platzhalterbibliothek** angelegt. Das bewirkt, dass die genaue Version der zu verwendenden Bibliothek nicht festgelegt ist, solange sie nicht in einem Projekt eingebunden wird. Erst dann bestimmt das aktuelle Visualisierungsprofil, welche Version tatsächlich benötigt wird. Beachten Sie, dass dieser Typ von Bibliothek sich von den gerätespezifischen Platzhalterbibliotheken unterscheidet, bei denen die Platzhalter aus der Gerätebeschreibung aufgelöst werden.

Basis-Bibliotheken sind standardmäßig eingebunden, sobald ein Visualisierungsobjekt in ein Standardprojekt eingefügt wird. Sie referenzieren weitere Bibliotheken, die hier nicht aufgeführt sind. Im Standardfall müssen Sie die Visualisierungsbibliotheken nicht explizit einfügen oder in Ihren Applikationen verwenden:

Basisbibliotheken

**VisuElems:** Basissatz an Visualisierungselementen

**VisuElemMeter:** Zeigerinstrument und Balkenanzeige

**VisuElemWinControls:** Tabelle, Textfeld, Scrollbalken

**VisuElemTrace:** Trace

**VisuInputs:** Handhabung der Benutzereingaben einer Visualisierung

In Visualisierungen, die einem Gerät zugewiesen sind, kann es **Geräte abhängige Einschränkungen** gemäß den Bibliotheken, die das Gerät unterstützt, geben. Diese betreffen Konfigurationsmöglichkeiten, wie zum Beispiel Liste der verfügbaren Elementen, Schriftarten, Farben, Bildformate, maximale Anzahl an Visualisierungen unterhalb des Geräts.

## **7 Bibliotheken**

**7.1 Richtlinien für das Erstellen von Bibliotheken**

**7.2 Standardbibliothek**

**7.3 Util Bibliothek**

**7.4 Busspezifische Bibliotheken**

**7.5 Systembibliotheken**

**7.6 (CAA-Bibliotheken)**

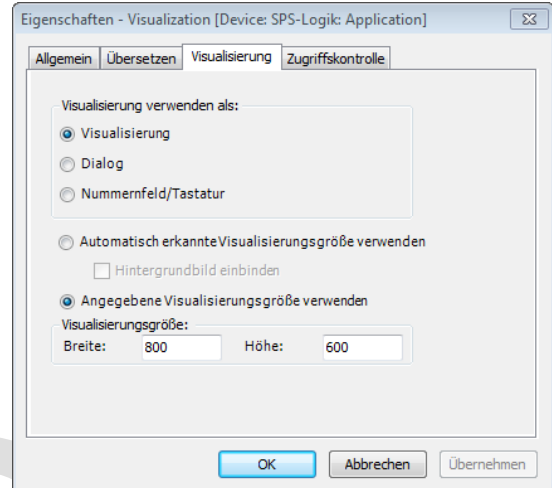
## 8 Projektierung

### 8.1 Erstellen einer ersten Visualisierungsaufgabe

#### 8.1.1 Bildschirmformat festlegen

Im Wesentlichen geht es um eine korrekte Einstellung von 4:3 (bspw. 800x600) oder 16:9 (800x480).

Wählen Sie Visualisierung → Eigenschaften und stellen die korrekten Verhältnisse ein:



#### 8.1.2 Neue Visualisierungs Maske anlegen

#### 8.1.3 Wechseln von Masken

8.1.4 *Wechseln von Masken in einem eigenen Fenster*

8.1.5 *Eingabefelder*

**8.2 Globale Parameter**

## 8.2.1 Eigene Elemente erstellen

Wenn Sie beispielsweise ein Element Pumpe erstellen wollen, gehen Sie in folgenden Schritten voran:

- Anlegen einer Struktur unter : Application → Objekt hinzufügen → DUT  
Beispielsweise : stuct\_Pumpe
- In der Struktur stuct\_Pumpe tragen Sie alle Elemente ein, mit denen Sie das Element von außen steuern möchten.



DUT

```
TYPE stuct_Pumpe :
```

```
STRUCT
```

```
    Farbumschlag: BOOL;
```

```
    Pumpenstellung: REAL;
```

```
END_STRUCT
```

```
END_TYPE
```

- In dem Visualisierung Editor können Sie mit  die Schnittstellen einblenden.



Visualization

```
VAR_IN_OUT
```

```
    stuct_Pumpe (*Visualisierung*) : stuct_Pumpe(*Programmcode*);
```

```
END_VAR
```

- Im Eigenschaften Dialog können bezugnehmen auf diese Struktur:

Eigenschaften	
Filter	Sortieren
Reihenfolge	Experte
Eigenschaft	Wert
Elementname	GenElemInst_3
Elementtyp	Lampe
Position	
X	63
Y	98
Breite	40
Höhe	43
Variable	stuct_Pumpe.Farbumschlag

und entsprechend die Elemente auswählen.

- Schon können Sie dieses Element auf beliebigen Masken einfügen und

Eigenschaften	
Filter	Sortieren
Reihenfolge	Experte
Eigenschaft	Wert
Pumpe	
stuct_Pumpe	PLC_PRG.inst_struct_Pumpe

den Eigenschaften die globale Variable zuweisen.

## 9 Anhang

### 9.1 TODO's

- Überschrift 5 Nummerierung

### 9.2 Grundlagen

FTP "File Transfer Protocol"	RFC 959 <a href="http://tools.ietf.org/html/rfc959">http://tools.ietf.org/html/rfc959</a>
HTTP/1.0 "Hypertext Transfer Protocol"	RFC 1945 <a href="http://tools.ietf.org/html/rfc1945">http://tools.ietf.org/html/rfc1945</a>
HTTPS "Hypertext Transfer Protocol Secure"	RFC 2818 <a href="http://tools.ietf.org/html/rfc2818">http://tools.ietf.org/html/rfc2818</a>
JSON "JavaScript Object Notation"	<a href="http://json.org/">http://json.org/</a>
application/ x-www-form-urlencoded	<a href="http://www.w3.org/TR/html401/interact/forms.html#h-17.13.4.1">http://www.w3.org/TR/html401/interact/forms.html#h-17.13.4.1</a>
PKCS #1 "RSA Cryptography Standard"	<a href="http://www.rsa.com/rsalabs/node.asp?id=2125">http://www.rsa.com/rsalabs/node.asp?id=2125</a>
ZLIB Compression (aka gzip)	RFC 1950 <a href="http://tools.ietf.org/html/rfc1950">http://tools.ietf.org/html/rfc1950</a>
AES "Advanced Encryption Standard"	<a href="http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf">http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf</a>
regex POSIX Regular Expressions	<a href="http://www.opengroup.org/onlinepubs/009695399/toc.htm">http://www.opengroup.org/onlinepubs/009695399/toc.htm</a>

### 9.3 Dateiformate

htm	text/html
svg	image/svg+xml
xml	text/xml
js	text/javascript
css	text/css
gif	image/gif
jpg	image/jpeg
png	image/png

### 9.4 Wichtigste Linux Befehle

telnet root root

CoDeSysControl.cfg

Shell starten:



```
. stop.sh  
. start.sh
```

Shell



## 10 Support

### Hotline

Für zusätzliche Unterstützung und Informationen, können Sie unsere Hotline zu folgenden Zeiten:

Mo-Fr: von 8.00- 12.00 und 13.00 - 16.30

Außerhalb dieser Zeiten, können Sie uns per e-mail oder fax erreichen:

Telefon: ++49 (0) 7021/92025-33  
Fax: ++49 (0) 7021/92025-29  
E-mail: hotline@elrest.de

### Training und Workshops

Wir bieten Ausbildung oder Projekt bezogene Workshops zu elrest Produkte an.

Für weitere Informationen, kontaktieren Sie bitte unsere Vertriebsabteilung:

Telefon: ++49 (0) 7021/92025-0  
Fax: ++49 (0) 7021/92025-29  
E-mail: [vertrieb@elrest.de](mailto:vertrieb@elrest.de)

## 11 Historie

Datum	Name	Kapitel	Änderung
18.05.2013	GS		erstellt

© 2013 elrest Automationssysteme GmbH. Alle Rechte vorbehalten.

Die in diesem Dokument enthaltenen Informationen können ohne Vorankündigung geändert werden und stellen keine Verpflichtung seitens elrest Automationssysteme GmbH dar. Die Software und/oder Datenbanken, die in diesem Dokument beschrieben sind, werden unter einer Lizenzvereinbarung und einer Geheimhaltungsvereinbarung zur Verfügung gestellt. Die Software und/oder Datenbanken dürfen nur nach Maßgabe der Bedingungen der Vereinbarung benutzt oder kopiert werden. Es ist rechtswidrig, die Software auf ein anderes Medium zu kopieren, soweit das nicht ausdrücklich in der Lizenz- oder Geheimhaltungsvereinbarung erlaubt wird. Ohne ausdrückliche schriftliche Erlaubnis der elrest Automationssysteme GmbH dürfen weder dieses Handbuch noch Teile davon für irgendwelche Zwecke in irgendeiner Form mit irgendwelchen Mitteln, elektronisch oder mechanisch, mittels Fotokopie oder Aufzeichnung reproduziert oder übertragen werden. Abbildungen und Beschreibungen sowie Abmessungen und technische Daten entsprechen den Gegebenheiten oder Absichten zum Zeitpunkt des Druckes dieses Prospektes. Änderungen jeder Art, insbesondere soweit sie sich aus technischem Fortschritt, wirtschaftlicher Ausföhrung oder ähnlichem ergeben, bleiben vorbehalten. Die externe Verschaltung der Geräte erfolgt in Eigenverantwortung.