

Serie:

Effizientes Projektieren mit zenon

Teil 3: Effizientes Symbolhandling

Inhalt

Effizientes Symbolhandling	2
Einleitung.....	3
Symbolbibliotheken	4
Effizientes Handling	5
Arbeiten mit der Symbolbibliothek	6
Symbole und Variablen/Funktionen.....	7
Verknüpfte Symbole – Variablen ersetzen	8
1.) Verknüpfungsregel	9
2.) Zusammengesetzte Namen	9
Erweiterungen spielend einfach.....	11
Mehrere Bilder.....	12
Last but not least: Eigenschaften freigeben	12
Wie funktioniert das im Detail?.....	12
Fazit	14

Effizientes Symbolhandling



In diesem White-Paper befassen wir uns näher mit „Effizientem Symbolhandling“. Dabei zeigen wir Ihnen, wie Sie mit sehr wenigen Mausklicks Symbole so universell gestalten können, dass Sie sich richtig viel Arbeit ersparen. Schlüsselbegriffe wie zum Beispiel eingebettete Symbole, verknüpfte Symbole, Symbol im Symbol, „Verknüpfungsregel“ versus „Verknüpfung ersetzen“ usw. werden ausführlich behandelt.

Einleitung

Zuerst möchten wir jedoch die wichtigsten Aussagen der letzten beiden Beiträge dieser Serie wiederholen: Starten wir mit der zenon Philosophie: „Parametrieren anstelle von programmieren“. Der Grund dafür ist schnell erklärt: Projekte werden schneller erstellt, sind einfacher zu handhaben bei der Inbetriebnahme, und zu guter Letzt gestalten sich Service und Wartung auch nach vielen Jahren noch kinderleicht. Die Vorteile liegen auf der Hand. Wir sind der festen Überzeugung, dass unsere Kunden damit in allen Bereichen eines Projektlebenszyklus effizient und erfolgreich sind. Der praktische Teil unseres ersten Artikels befasste sich mit „global/zentral anstelle von lokal“: Globale, also projektübergreifende Einstellungen sowie zentrale Einstellungen erlauben Ihnen, Dinge sehr rasch zu ändern. Sie sind damit in der Lage, mit nur wenigen Mausklicks das Design und das Verhalten von ganzen Projekten zu ändern. Mühsames Suchen und Ersetzen bleibt erspart. Dieser zentrale Ansatz wird uns auch in diesem Artikel einige Male verfolgen und Sie werden sehen, dass wir ihn inzwischen noch weiter verbessert haben.

Der zweite Artikel der Serie befasste sich mit „Objektorientierter Parametrierung“. Dabei wurde gezeigt, wie man mit Strukturvariablen, Arrays und den Hilfsmitteln der Vererbung und Überschreibung einen sehr flexiblen und effizienten Variablenhaushalt führen kann.

Nun aber zum eigentlichen Thema: den Symbolen. Was ist eigentlich ein Symbol in zenon? Ein Symbol kann in vielen Varianten auftreten: Es kann ein grafisches Element – zum Beispiel ein Rechteck oder eine Linie – oder ein Zahlenwert sein. Auch ein Universalregler kann zum Symbol werden, indem er mithilfe des Kontextmenüs oder der Symbolleiste „Anordnen“ umgewandelt wurde. Um ein Symbol anzulegen, ist es am sinnvollsten, mehrere Elemente aneinander zu fügen. Dann ist ein Symbol eine Gruppe von Elementen, die eine Einheit bilden und sich durch gemeinsame Eigenschaften auszeichnen.

Zunächst ist ein Symbol also nur eine Gruppe von Elementen, die man zum Beispiel gemeinsam verschieben oder deren Größe man ändern kann. Prinzipiell gibt es zwei Arten von Symbolen: eingebettete Symbole und verknüpfte Symbole. Eingebettete Symbole können Sie direkt im Bild erzeugen. Sie sind individuell und haben keinerlei Bezug zur Symbolbibliothek. Anders die verknüpften Symbole: Diese werden nicht im Bild, sondern in der Symbolbibliothek

gespeichert. Im Bild wird nur ein Verweis auf das Symbol in der Bibliothek gespeichert. Der Vorteil dabei: Das Symbol kann beliebig oft verwendet werden. In einem Bild, in mehreren Bildern, oder sogar in mehreren Projekten.

Symbolbibliotheken

Die Symbolbibliothek ist nichts anderes als ein Speicherort für Symbole. zeron Nutzer wissen, dass es davon zwei im Produkt gibt: Die Globale Symbolbibliothek und die Projektsymbolbibliothek.

Die Globale Bibliothek (Abbildung 1) dient dazu, Symbole bereitzustellen, die projektübergreifend genutzt werden. Die Symbole werden darin nach Gruppen verwaltet. Physikalisch steht jede Gruppe für eine Datei, die im Datenverzeichnis des jeweiligen zeron Editors als .SYM Datei abgelegt ist. Dadurch ergeben sich bestimmte Einschränkungen: Bei jeder zeron Version gibt es ein neues Datenverzeichnis, was bedeutet, dass die Symboldateien manuell kopiert werden müssen. Außerdem sind die Symbole nicht Teil des Projekts und werden daher bei der Projektsicherung nicht berücksichtigt.

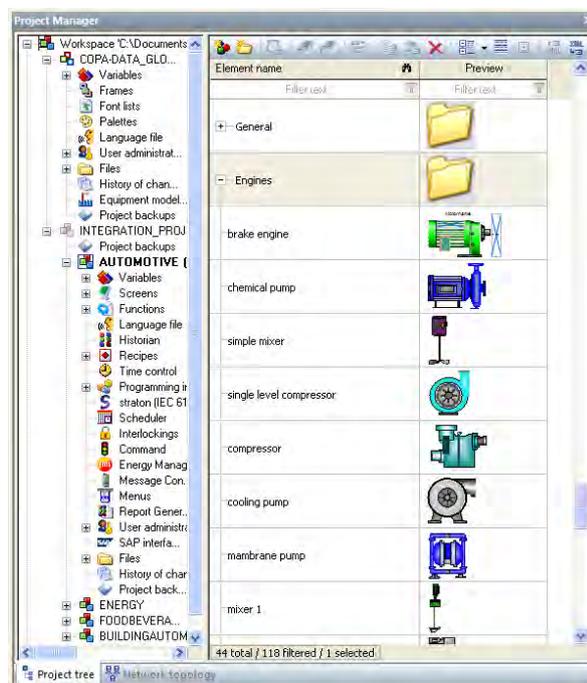


Abbildung 1: Symbolverwaltung auf Basis von Gruppen: die Globale Symbolbibliothek in zeron

Anders die Projektsymbolbibliothek: Die darin abgespeicherten Symbole sind Bestandteil des Projekts und werden somit gesichert. Dafür können sie jedoch nicht projektübergreifend genutzt werden. In der Projektsymbolbibliothek gibt es keine Gruppen wie in der Globalen Symbolbibliothek. Jedoch können anhand der beiden Eigenschaften „Kategorie“ und „Beschreibung“ ebenfalls Gruppierungen vorgenommen werden: Einfach einen Rechtsklick auf den Spaltenkopf Kategorie oder Beschreibung ausführen und die Funktion „Nach dieser Spalte gruppieren“ wählen.

Effizientes Handling

Zu Beginn des Textes haben wir erklärt, wie man ein Symbol erzeugt: Einfach ein oder mehrere Bildelemente heranziehen und „Symbol erzeugen“ ausführen. Was aber, wenn man später das Symbol um ein Element erweitern will? – Kein Problem für jeden Anwender. Dazu gibt es die Funktion „In existierendes eingebettetes Symbol einfügen“. Damit werden die selektierten Elemente dem Symbol hinzugefügt. Wichtiges Detail: Die Anordnung der Elemente bleibt erhalten. Liegen die neuen Elemente außerhalb der bisherigen Symbolgrenzen, wird das Symbol entsprechend größer.

Eine weitere sehr wichtige Funktion in Bezug auf das Symbolhandling ist die Funktion „Einzelbearbeitungsmodus Symbol/Elemente“. Damit schalten Sie alle eingebetteten Symbole im Bild in den sogenannten Einzelbearbeitungsmodus um. In diesem Modus können Sie einzelne Elemente im Symbol auswählen und deren Eigenschaften ändern, zum Beispiel eine Farbe anpassen oder die Position eines Elements im Symbol ändern. Der Clou dabei: Das Symbol wird nicht aufgelöst, sondern bleibt mit all seinen Eigenschaften erhalten. Wenn Sie den Einzelbearbeitungsmodus deaktivieren, können Sie das Symbol wieder als Ganzes bearbeiten.

Noch ein Tipp zum Bearbeiten: Oft kommt es vor, dass sich mehrere Elemente übereinander befinden und nur das oberste Element sichtbar ist. Sie müssen in jedem jedoch nicht alle Elemente auf die Seite schieben, um die darunter liegenden zu erreichen. Klicken Sie einfach bei gedrückter ALT-Taste auf das oberste Element. So wird das Element darunter selektiert. Beim nächsten Klick noch eines darunter usw. So kann man sich durch alle übereinanderliegenden Elemente durchklicken und mittels Eigenschaftenfenster oder Tastaturbedienung alle Eigenschaften auch von darunterliegenden Elementen verändern.

Arbeiten mit der Symbolbibliothek

Angenommen, wir möchten ein erzeugtes Symbol nun nicht nur einmal, sondern mehrfach verwenden. Was ist zu tun? Dazu muss das Symbol in eine der beiden Symbolbibliotheken eingefügt werden. Das passiert mittels Kontextmenü durch Rechtsklick auf ein Symbol: „Einfügen in Symbolbibliothek“. Das eingebettete Symbol bleibt im Bild erhalten. Wenn man es nicht mehr benötigt, kann es gelöscht werden. Nun ist das Symbol in der Bibliothek und auch dort bearbeitbar. Mittels Doppelklick wird das Symbol im Symboleditor geöffnet und kann dort geändert oder erweitert werden. Der Symboleditor verfügt über dieselben Eigenschaften wie der Bildeditor. Das Symbol kann aber über die Eigenschaften auch in seiner Größe geändert werden. Durch Drag & Drop wird das Symbol von der Bibliothek in das Bild gezogen und dort als verknüpftes Symbol eingefügt. Der Vorteil dabei: Wird das Symbol in der Bibliothek geändert, wird es gleichzeitig an allen Stellen mit geändert. Auch hier ein Tipp: Über das Kontextmenü kann man ein verknüpftes Symbol gleich zum Bearbeiten im Symboleditor öffnen bzw. über den Menüpunkt „Verknüpfte Elemente“ kann man direkt zum Symbol in der Bibliothek springen.

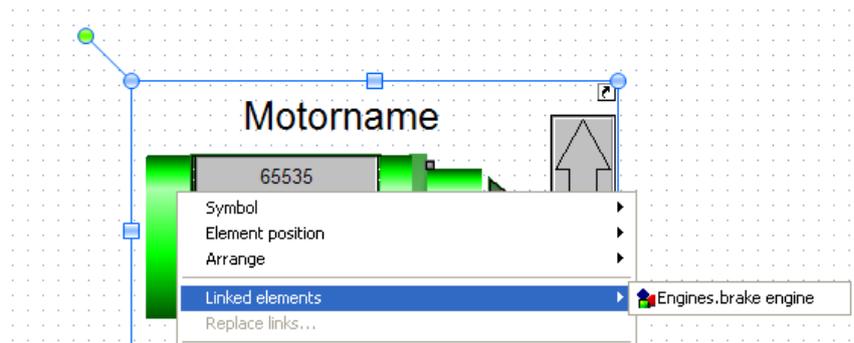


Abbildung 2: Das Symbol zeigt dessen Quelle in der Symbolbibliothek an, zu der man per Menüpunkt „Verknüpfte Symbole“ direkt springen kann.

Seit der zenon Version 6.50 gibt es noch eine weitere Verbesserung dieses Symbolhandlings. Symbole können nicht nur in ein Bild, sondern nun auch in ein anderes Symbol verknüpft eingefügt werden.

Zum Beispiel: Sie haben ein Symbol, das einen Rohrleitungsstrang darstellt. Dieser Strang beinhaltet fünf Ventile. Sie könnten nun natürlich jedes Ventil einzeln zeichnen. Wesentlich effizienter ist jedoch die folgende Vorgehensweise: Sie zeichnen das Ventil einmal und speichern es als Ventil in der Symbolbibliothek ab. Nun zeichnen Sie Ihren Rohrleitungsstrang und fügen

überall dort, wo Sie das Ventil benötigen, das Symbol „Ventil“ per Drag & Drop ein. Wenn Sie das Ventil ändern möchten, zum Beispiel in Größe, Farbe oder Verhalten, müssen Sie das nur an einer einzigen Stelle manchen.

Natürlich können Sie auch mehrere Ebenen so gestalten, also ein Symbol, das ein Symbol enthält, das wiederum ein Symbol enthält usw. Zur leichteren Erreichbarkeit können Sie mit dem Befehl „Verknüpfte Elemente“ immer eine Ebene weiter hinein springen.

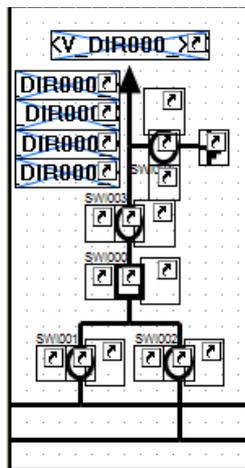


Abbildung 3: Abzweig mit vielen verknüpften Symbolen, die an zentraler Stelle verändert werden können.

Symbole und Variablen/Funktionen

Selbstverständlich können Elemente im Symbol Variablen enthalten. Auch das Symbol selbst kann mit Variablen verknüpft sein, um etwa das Symbol sichtbar/unsichtbar zu schalten oder es zu drehen. Zusätzlich können Bedienelemente wie ein Button oder ein Kombielement mit einer verknüpften Funktion im Symbol enthalten sein. Diese Variablen/Funktionen kann man sehr einfach ändern.

Zuerst für das eingebettete Symbol: Durch einen Rechtsklick auf das Symbol öffnet sich das Kontextmenü mit dem Befehl „Verknüpfungen ersetzen“. Darin werden Ihnen alle Variablen/Funktionen zum Ändern angeboten. Mit Hilfe von Wildcards können Sie die Mehrfachersetzungen durchführen. Auch einzelne Ersetzungen sind möglich. Durch Klick auf OK werden diese Einstellungen fix im Symbol gespeichert.

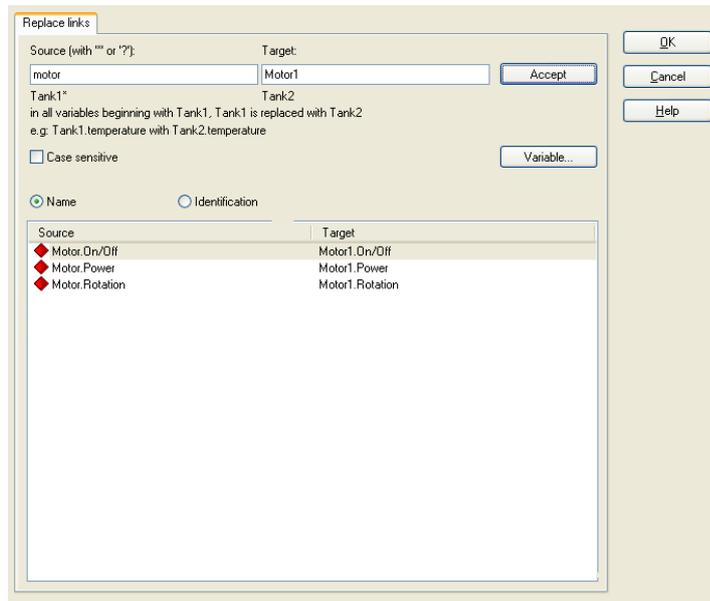


Abbildung 4: Per Kontextmenü können Verknüpfungen für eingebettete Symbole ersetzt werden.

Verknüpfte Symbole – Variablen ersetzen

Bei den verknüpften Symbolen ist es etwas anders: Bei diesen Symbolen ist im Bild selbst kein Symbol gespeichert, sondern es besteht nur ein Verweis auf das Symbol. Es können daher keine Variablen/Funktionen direkt ersetzt werden. Ein Ersetzen ist aber dennoch möglich – auf zwei verschiedene Arten: Mit Hilfe einer Verknüpfungsregel oder mit Hilfe von Automatischer Benennung (Zusammengesetzte Namen).

Zum besseren Verständnis erkläre ich beide Varianten anhand eines Beispiels:

Zuerst benötigen wir eine Variable. Am besten eignet sich dazu eine Strukturvariable, beispielsweise die Struktur Motor. Diese enthält ein Strukturelement Leistung und ein zweites Element Drehzahl. Daraus bilden wir eine Strukturvariable Motor1: Diese besteht dann aus den Variablen Motor1.Leistung und Motor1.Drehzahl. Diese beiden Variablen verbinden wir nun z.B. mit einem Bargraf und einem Zahlenwert im Symbol in der Bibliothek. Im Anschluss erzeugen wir die Strukturvariablen Motor2, Motor 3 usw. Hinweis: Das kann man natürlich auch als Array anlegen.

1.) Verknüpfungsregel

Wenn wir das Symbol nun in das Bild ziehen, öffnet sich der Dialog „Verknüpfungsregel bearbeiten“ (Abbildung 5), der dem Dialog „Verknüpfung ersetzen“ (siehe Abbildung 4) sehr ähnlich ist. In diesem Dialog können wir nun eine oder mehrere Verknüpfungsregeln mittels Variablenzuweisung ändern. In unserem Fall wollen wir Motor1 durch Motor2 ersetzen. Wir geben also bei der Quelle Motor1* und beim Ziel Motor2 (ohne Asterisk Zeichen) an. Tipp: Mehrere Regeln werden durch Strichpunkt voneinander getrennt: Quelle: Motor1*; Pumpe1* Ziel: Motor2; Pumpe2

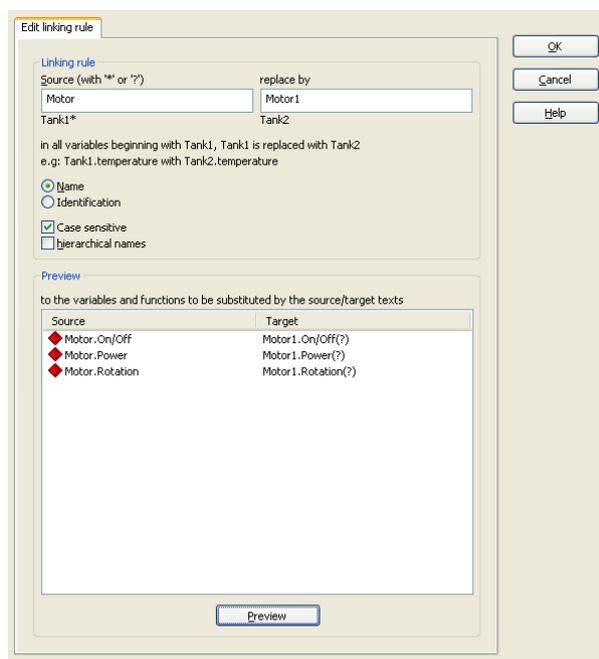


Abbildung 5: Über das Kontextmenü „Verknüpfungsregel bearbeiten“ können Sie eine oder mehrere Regeln für verknüpfte Symbole ändern.

In der zenon Runtime werden dann im Symbol die Variablen Motor2.Leistung und Motor2.Drehzahl angezeigt. Hinweis: Die Regel kann man über das Eigenschaftfenster auch nachträglich bearbeiten.

2.) Zusammengesetzte Namen

Bei dieser Variante wenden wir einen „Trick“ an: Der Variablenname im Symbol setzt sich aus dem Variablennamen im Element und dem Symbolnamen zusammen. Damit die Zusammensetzung der Namen funktioniert, muss man vorher den Variablennamen im Element bearbeiten. Dazu ist es nötig, den

Bargraf und den Zahlenwert zu selektieren und den Variablennamen manuell per Texteingabe zu verändern. In unserem Beispiel ist der Bezeichner Motor1 zu entfernen: Motor1.Leistung wird so zu .Leistung und Motor1.Drehzahl wird zu .Drehzahl. Beim Hineinziehen des Symbols in das Bild ist der Dialog „Verknüpfungsregel bearbeiten“ zu ignorieren. In den Eigenschaften müssen Sie die Eigenschaft „Zusammengesetzte Namen“ aktivieren. Wenn Sie nun noch das Symbol in Motor1, Motor2 usw. umbenennen, werden die Variablen automatisch richtig zugeordnet. Symbolname Motor1 + Variablenname .Leistung ergibt automatisch: Motor1.Leistung.

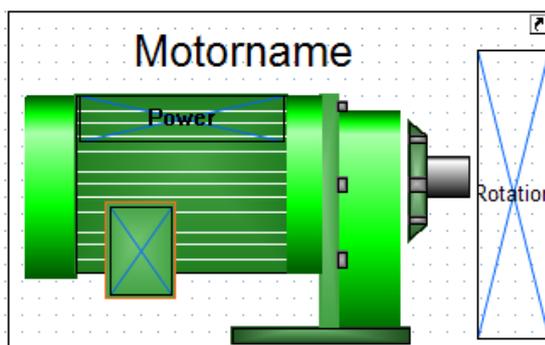


Abbildung 6: Symbol im Bild, ohne Zuordnung der Variablen

Linking rule

Source (e.g: TANK1...):

Target (e.g: TANK2):

via variable name Case sensitive

Hierarchical names

Preview:

Abbildung 7: Mit der Aktivierung von „Zusammengesetzte Namen (Hierarchical names)“ werden die Variablen automatisch verknüpft.

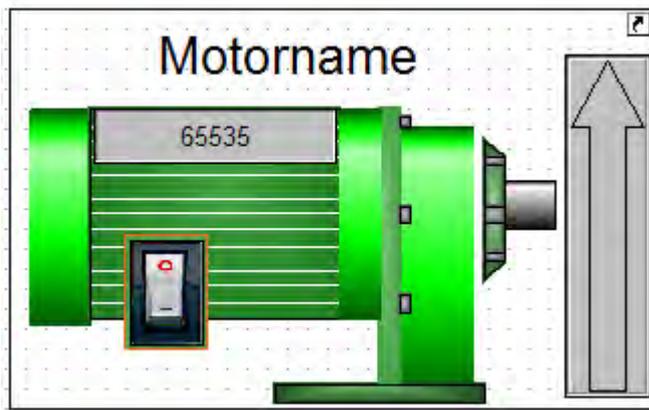


Abbildung 8: Symbol im Bild, mit Zuordnung der Variablen

Im Ergebnis unterscheiden sich die beiden Varianten nicht. Bei der ersten Variante, der Verknüpfungsregel, haben Sie allerdings den Vorteil, mehrere Variablen/Funktionen auf einmal zu ersetzen.

Erweiterungen spielend einfach

Richtig spannend wird das Thema, wenn sie das Symbol erweitern müssen. Nehmen wir an, zu Drehzahl und Leistung kommt eine weitere Anzeige hinzu: Die Stromaufnahme. Ein effizient arbeitender Projektierer lacht sich bei so einer Herausforderung ins Fäustchen – jetzt kann er alle Trümpfe ausspielen. So geht's:

1. Strukturdatentyp um das Strukturelement Stromaufnahme erweitern.
2. Mit einem Mausklick alle neuen Variablen aktivieren (diese sind standardmäßig inaktiv).
3. Das Symbol bearbeiten und dort einen neuen Zahlenwert mit der Stromaufnahme hinein zeichnen.
4. Das Symbol speichern.

Fertig, das ist alles! Egal, ob Sie zwei oder 1.000 Motoren haben: Alle werden nun über einen Zahlenwert mit Stromaufnahme verfügen. Dadurch, dass beim Symbol nur eine Verknüpfungsregel abgespeichert ist, trifft diese Regel auch für die neue Variable Motor1.Stromaufnahme zu und folglich wird in der Runtime

diese Variable auch ersetzt. Dasselbe gilt auch für die zusammengesetzten Namen.

Mehrere Bilder

Wenn Ihre 1.000 Motoren nicht in einem Bild Platz haben, ist das kein Grund zur Verzweiflung. Natürlich müssen Sie deshalb nicht 1.000 Bilder zeichnen. Ein einziges Bild genügt. Bei der Bildumschaltung können Sie wiederum problemlos den „Verknüpfung ersetzen“-Dialog verwenden. Beispiel: Wenn Sie Motor1 und Motor2 im Bild haben, ersetzen Sie bei der Bildumschaltung Motor1 durch Motor3 und Motor2 durch Motor4.

Last but not least: Eigenschaften freigeben

Sozusagen als Höhepunkt zum Thema Symbolhandling gibt es auch bei den Symbolen die Möglichkeit, Eigenschaften eines verknüpften Symbols im Bild individuell anzupassen. Das funktioniert ähnlich wie das Überschreiben von Eigenschaften bei Variablen, die vom Datentyp abgeleitet sind (siehe Teil 2 dieser Themenserie).

Wie funktioniert das im Detail?

Bleiben wir bei unserem Beispiel: Angenommen, wir möchten unserem Motorsymbol eine Überschrift geben, damit der Anwender sofort erkennt, um welchen Motor es sich handelt. Dazu legen wir einen statischen Text an und vergeben als Text: Motor. Alle Motoren im Bild tragen nun die Überschrift Motor. Das Symbol ist mit der Bibliothek verlinkt. Daher können wir den Text nicht individuell ändern. Damit dies funktioniert, müssen wir die Eigenschaft Text zuerst freigeben. Auch das funktioniert per Drag & Drop: Einfach die Eigenschaft aufnehmen und in den Bereich unter das Symbol ziehen, wo alle freigegebenen Eigenschaften verwaltet werden.

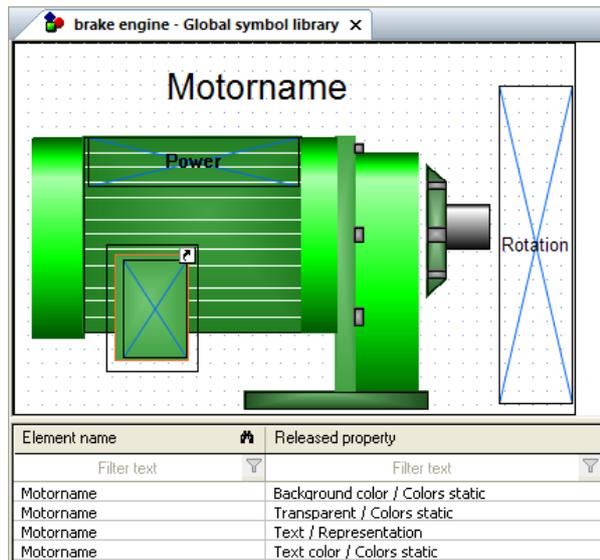


Abbildung 9: Symbol in der Globalen Symbolbibliothek mit freigegebenen Eigenschaften

Wenn wir nun das Symbol in das Bild ziehen bzw. einen Doppelklick auf ein Symbol im Bild ausführen, erscheint ein Assistent, der uns erlaubt, die Eigenschaft dynamisch zu ändern. Wir können also mühelos die Motoren in Motor1, Motor2 usw. benennen. Natürlich können diese freigegebenen Eigenschaften auch über das Eigenschaftfenster geändert werden. Darin erscheinen sie speziell gekennzeichnet als weitere Eigenschaften des Symbols.

Ebenso ist es möglich mit dem Syntax %n bzw %l Variablennamen und Variablenkennung automatisch zuzuordnen. Weitere Information hierzu in der Hilfe unter: Handbuch -> Bilder -> Bildelemente -> Combi-Element -> Darstellung von Variablenname und Variablenkennung.

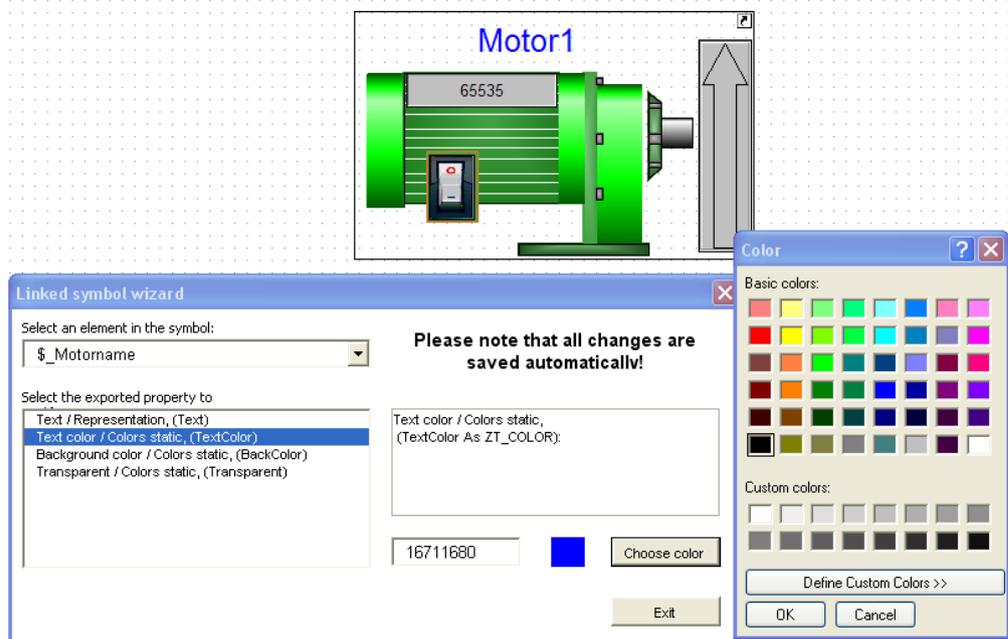


Abbildung 10: Mithilfe des Assistenten kann jede beliebige Symboleigenschaft, zum Beispiel die Farbe (wie im Bild), geändert werden.

Fazit

Die zenon Symbolverwaltung ist ein mächtiges Werkzeug, um grafische Objekte effizient zu gestalten und zu verwalten. Die Vorteile der verwendeten Mechanismen zeigen sich vor allem, wenn etwas geändert oder angepasst werden sollte. Spätestens dann entpuppt sich die zenon Philosophie „Parametrieren anstelle von programmieren“ wieder als echter Mehrwert und erlaubt dem Bediener schnelle und einfache Umsetzung.

Viel Spaß beim Projektieren!



© 2012 Ing. Punzenberger COPA-DATA GmbH

Dieser Artikel ist in einer älteren Version im IU Magazin No. 19 erschienen.

All rights reserved.

This document may not be reproduced or photocopied in any form (electronically or mechanically) without a prior permission in writing from Ing. Punzenberger COPA-DATA GmbH. The technical data contained herein has been provided solely for informational purposes and is not legally binding. Subject to change, technical or otherwise. zenon®, zenon Analyzer®, zenon Supervisor®, zenon Operator®, zenon Logic® and straton® are trademarks registered by Ing. Punzenberger COPA-DATA GmbH. All other brands or product names are trademarks or registered trademarks of the respective owner and have not been specifically earmarked. We thank our partners for their friendly support and the pictures they provided.