zenon
by COPA-DATA

Manuel de zenon

Développez efficacement vos projets avec zenon

v.8.10

© 2019 Ing. Punzenberger COPA-DATA GmbH

Tous droits réservés.

La distribution et/ou reproduction de ce document ou partie de ce document, sous n'importe quelle forme, n'est autorisée qu'avec la permission écrite de la société COPA-DATA. Les données techniques incluses ne sont fournies qu'à titre d'information et ne présentent aucun caractère légal. Document sujet aux changements, techniques ou autres.

Contenu

1 Bienvenue dans l'aide de COPA-DATA 5

2 Développez efficacement vos projets avec zenon 5

3 Simplicité d'utilisation 6

4 Orientation objet 8

4.1 Paramétrer au lieu de programmer 9

4.2 Global et central plutôt que local 11

4.3 Paramétrage orienté objet 12

4.3.1 Drivers 12

4.3.2 Type de données 14

4.3.3 Conseils relatifs à l'adressage et à l'importation/exportation 19

5 Réutilisation d'éléments 20

5.1 Remplacement de variables et de fonctions 20

5.1.1 Naming conventions 21

5.1.2 Possibilités de remplacement 22

5.2 Symbols 25

5.2.1 Propriétés d'accès libre 26

5.2.2 Remplacement par des symboles 28

5.2.3 Bibliothèque générale de symboles 29

5.3 Structure Data Type 32

5.4 Matrices de réaction 34

5.5 Projet global 34

5.6 XML 36

5.7 Assistants 37

5.8 Réutiliser des projets 38

5.8.1 Sauvegarde de projet 38

5.8.2 Save as 39

5.8.3 Gestion multi-projets 39

6 Réseau intégré : 40

7 Conseils et astuces 41

7.1 zenon Éditeur : 41

7.2 zenon Runtime 44

7.3 Raccourcis-clavier....45
7.4 Test de projets....51

1 Bienvenue dans l'aide de COPA-DATA

TUTORIELS VIDÉO DE ZENON.

Des exemples concrets de configurations de projets dans zenon sont disponibles sur notre chaîne YouTube (https://www.copadata.com/tutorial_menu). Les tutoriels sont regroupés par sujet et proposent un aperçu de l'utilisation des différents modules de zenon. Les tutoriels sont disponibles en anglais.

AIDE GÉNÉRALE

Si vous ne trouvez pas certaines informations dans ce chapitre de l'aide ou si vous souhaitez nous suggérer d'intégrer un complément d'information, veuillez nous contacter par e-mail : documentation@copadata.com.

ASSISTANCE PROJET

Si vous avez besoin d'aide dans le cadre d'un projet, n'hésitez pas à adresser un e-mail à notre service d'assistance : support@copadata.com

LICENCES ET MODULES

Si vous vous rendez compte que vous avez besoin de licences ou de modules supplémentaires, veuillezcontacter l'équipe commerciale par e-mail : E-mail sales@copadata.com.

2 Développez efficacement vos projets avec zenon

Utilisez-vous zenon pour la première fois ?
Vous trouverez ici des informations qui vous aideront à utiliser zenon efficacement. Même si vous maîtrisez déjà d'autres systèmes de contrôle de procédé, nous vous recommandons de lire ce chapitre. Vous découvrirez les concepts fondamentaux sur lesquels repose zenon, ainsi que son fonctionnement, mais aussi pourquoi et comment zenon vous aide à travailler plus facilement et plus rapidement.

UN APERÇU DE ZENON

zenon est constitué d'Editor et du Runtime.
Les projets sont créés dans l'Éditeur. La gestion et l'observation se déroulent dans le Runtime L'Éditeur et le Runtime sont compatibles à compter des versions 6.20 et ultérieures. Par exemple, un projet créé avec la version 6.50 est également compatible avec la version 6.22 du Runtime, et inversement.

zenon diffère principalement des autres systèmes par les caractéristiques suivantes :

- Utilisation simple (à la page 6) : de nombreuses actions peuvent être réalisées en quelques clics, tandis que les assistants rendent votre travail plus rapide et plus simple. Vous pouvez également travailler en effectuant des manipulations de glisser&déplacer avec la souris, ou en utilisant le clavier – à votre convenance.
- Orientation objet concrète (à la page 8): Vous travaillez continuellement avec des objets dont vous pouvez rapidement définir les paramètres, au lieu d'écrire des programmes fastidieux. Vous utilisez également les objets définis une fois dans tous les projets ; vous bénéficiez de fonction de transmission de propriétés et de la capacité de modifier ces dernières avec une rapidité inégalée.
- Excellente capacité de réutilisation (à la page 20) des objets et éléments préalablement configurés : de nombreux éléments, des variables individuelles jusqu'aux projets entiers, peuvent être réutilisés très facilement et efficacement dans zenon.
- Réseau intégré (à la page 40) : le développement distribué sur le réseau et la redondance totale ne demandent plus un effort démesuré. Vous pouvez créer des réseaux en cochant la case concernée. De cette manière, vous pouvez connecter des projets autonomes individuels sous forme de systèmes redondants efficaces.

CONSEILS ET ASTUCES

Vous trouverez des conseils utiles sur le forum de www.copadata.com. La meilleure manière d'apprendre à profiter pleinement des capacités de zenon consiste à assister aux séminaires de formation et aux ateliers de COPA-DATA, adaptés à vos priorités et vos projets. Votre distributeur peut vous fournir d'autres informations ; vous pouvez également nous écrire à l'adresse sales@copadata.com.

3 Simplicité d'utilisation

zenon facilite la configuration grâce à de nombreux petits agréments, et augmente ainsi votre productivité. En utilisant zenon, vous découvrirez que vous pourrez résoudre de nombreux problèmes à votre convenance. Par exemple :

FAVORIS

Vous pouvez définir vos propres favoris dans les propriétés des objets et les boîtes de dialogue des fonctions. Ils vous sont toujours proposés au début de la liste des propriétés, et vous permettent d'accéder rapidement aux propriétés que vous utilisez fréquemment.

UTILISATION GÉNÉRALE DU CLAVIER DANS ÉDITEUR

Éditeur peut être facilement contrôlé grâce aux symboles et aux menus actionnables avec la souris ; vous pouvez également l'utiliser uniquement à l'aide du clavier. Les raccourcis-clavier de Windows fonctionnent également dans zenon. Par exemple :

- **Ctrl+C** et **Ctrl+V** pour **copier** et **coller**.
- **Inser** crée de nouveaux objets.
- **Suppr** supprime les objets sélectionnés
- **Déplacement de la souris lorsqu'un bouton de souris est enfoncé :** Navigation dans les vues
- **Echap** annule une action
- **Retour arrière** remonte d'un niveau dans l'arborescence des répertoires.
- **Ctrl+Z** annule la dernière action, à l'exception des opérations sur bases de données.

Pour plus d'informations concernant l'utilisation au clavier, reportez-vous au chapitre Raccourcis clavier.

GLISSER&DÉPOSER

Dans Éditeur, vous pouvez travailler plus efficacement en utilisant des manipulations de type glisser-déplacer. Par exemple :

- Ordre de modification des entrées de menus
- Lier des polices à des éléments dynamiques tels que des réglettes universelles, des horloges, des éléments combinés etc.
- Modifier une séquence d'éléments structurels pour les données de type structure.
- Modifier une séquence dans un script
- Copier ou déplacer des fonctions à l'aide de différents scripts
- Copier des scripts en les déplaçant vers le nœud principal.

ASSISTANTS INTELLIGENTS POUR L'EXÉCUTION DE TÂCHES COMPLEXES

Les assistants intégrés à zenon vous aident à créer un projet de base en un clin d'œil, ou à réaliser des ajustements complexes. Nous vous recommandons d'utiliser les assistants pour la configuration de

réglettes universelles, d'éléments combinés et d'archives, plus particulièrement si vous n'êtes pas encore familiarisé avec le logiciel.

INFOS-BULLES POUR LES ÉLÉMENTS ET LES PROPRIÉTÉS

Si vous positionnez le pointeur de la souris sur un élément d'affichage (par exemple, un bouton de texte) ou une propriété comportant un lien (par exemple, une fonction liée à un bouton de texte), une info-bulle s'affiche à l'écran. Elle contient des informations relatives aux éléments liés, telles que la fonction liée et ses paramètres.

POSSIBILITÉS DE DESSIN ÉTENDUES

L'éditeur graphique propose une vaste gamme de fonctions qui facilitent la représentation visuelle des équipements, tout en proposant de nombreuses options de conception. Ces options s'étendent du zooms fluide à la capacité de redimensionner les éléments en faisant glisser les coins, ainsi qu'à la distribution automatique d'éléments conformément à des règles. Vous trouvez plus de détails au chapitre Synoptiques.

ACCÉDER À DES ÉLÉMENTS LIÉS

Pour tous les éléments graphiques, fonctions, variables et autres éléments, vous pouvez accéder directement aux éléments liés. Par exemple, d'un clic de souris, vous pouvez directement accéder à un élément lié depuis un élément d'affichage ou, depuis une variable, accéder au type de de données et même aux unités utilisées. Si nécessaire, vous pouvez également effectuer facilement toutes ces étapes à rebours jusqu'à l'élément initial, d'un clic de souris.

4 Orientation objet

COPA-DATA envisage l'automation d'un point de vue pragmatique, selon une orientation objet. C'est ainsi que fonctionne zenon ; de plus, ce programme peut lui-même être largement automatisé. zenon ne nécessite aucun langage de programmation ; vous n'aurez pas à programmer la moindre ligne de code. Au lieu de cela, vous attribuez des propriétés à des objets à l'aide de la souris. Cela fonctionne non seulement avec les objets individuels, mais également de manière centrale, pour un projet entier, et d'un objet à l'autre également.

Si la perspective de ne pas pouvoir personnaliser zenon et de vous retrouver entièrement dépendant des propriétés des objets vous inquiète, voici une bonne nouvelle : vous pouvez bien évidemment adapter la configuration de zenon individuellement. Pour cela, utilisez préférablement VBA ou l'environnement de développement VSTA (.NET). En outre, zenon propose de nombreuses interfaces permettant de communiquer avec des équipements et des logiciels, et fournit zenon Logic avec une logique SCADA intégrée.

TROIS PRINCIPES DE BASE IMPORTANTS POUR VOUS AIDER À UTILISER ZENON

Trois principes centraux simplifient et facilitent le développement et la maintenance de projets dans zenon :

1. Paramétrer au lieu de programmer (à la page 9) :

   Définissez facilement les paramètres requis, au lieu de programmer ou d'adapter des scripts.

2. Global et central plutôt que local (à la page 11) :

   Définissez des objets une fois et réutilisez-les aussi souvent que vous le souhaitez, même dans différents projets, au lieu de rédiger ou copier des scripts indéfiniment.

3. Paramétrage orienté objet : (à la page 12)

   profitez de tous les avantages de la réflexion orientée objet.

4.1 Paramétrer au lieu de programmer

Quel avantage vous offre le paramétrage de projets plutôt que leur programmation ?

Vous évitez de créer du code susceptible de contenir des erreurs et gagnez en flexibilité ; votre projet dispose de structures claires, est plus rapide et offre une plus grande capacité de réutilisation. Quatre éléments font particulièrement pencher la balance en faveur du paramétrage :

1. **Les modules prédéfinis offrent un gain de temps ;**
   zenon comporte de nombreux modules prédéfinis. Au lieu de programmer des scripts vous-même, vous choisissez le module adéquat et le configurez en sélectionnant les propriétés et les paramètres de votre choix. Cela vous permet de créer des projets exécutables en un clin d'œil.

2. **Le produit fourni par le fabricant est déjà arrivé à maturité ;**
   le concepteur se concentre sur ses projets. Le logiciel est développé et testé par le fabricant ; les concepteurs n'ont pas besoin de rédiger leur propre code. Il est donc très simple d'utiliser zenon – même pour les employés ne possédant aucune expérience du développement de logiciels.

3. **Maintenance et adaptation simples du projet :**
   zenon opère selon une logique strictement orientée objet. Associée à la philosophie consistant à paramétrer plutôt qu'à programmer, cela signifie une maintenance et une adaptation simples. Pour les modifications et le contrôle, il vous suffit de contrôler ou modifier les propriétés des objets individuels au lieu de devoir lire et adapter de longues lignes de code. Tout au long de la durée de vie du projet, cela garantit :

   - Aperçu : les projets peuvent être recréés plusieurs années plus tard, même par des employés inexpérimentés.

- Sécurité : les scripts et la programmation, susceptibles de dissimuler des erreurs, sont évités.
- Un gain de temps : la configuration, l'adaptation et la maintenance peuvent être réalisés beaucoup plus rapidement.

4. **Adaptation facile sur différentes machines :**
   zenon peut être adapté facilement et rapidement sur des machines nouvelles ou modifiées. Les adaptations de projets peuvent être effectuées en quelques clics seulement.

ADAPTATIONS ZENON

Le paramétrage évite que les erreurs contenues dans des scripts ne soient reproduites à l'infini. Il évite les laborieuses opérations de modification de scripts individuels lorsque vous souhaitez simplement adapter quelques détails. Toutefois, cela ne signifie pas que vous devez être inflexible ! zenon vous offre également de nombreuses possibilités d'étendre et d'adapter votre système.

- **VBA/VSTA :**
  Grâce au langage de scripts intégré Visual Basic for Applications et à C# et VB.NET, vous pouvez exécuter n'importe quel code dans le Runtime. Ceci autorise l'exécution de tâches d'automation, de tâches logiques et de tâches de gestion d'interfaces, à l'image des accès aux bases de données.
- **zenon Logic:**
  Cet environnement de contrôle entièrement intégré vous permet, grâce à sa logique SCADA, d'utiliser les cinq langages de programmation IEC 61131-3. Grâce à lui, vous pouvez exécuter vos calculs et résoudre tous vos problèmes logiques.
- **PCE:**
  Le moteur de contrôle de procédés intégré vous permet d'utiliser VB-Script et Java Script dans le cadre de vos tâches d'automation.
- **Contrôles ActiveX :**
  Vous pouvez intégrer des contrôles standard, tels que Flash Player, Acrobat Reader etc. à zenon. L'interface est évidemment visible, ce qui vous permet d'y inclure vos propres contrôles ActiveX.
- **Éléments WPF**
  L'intégration directe du format **Windows Presentation Foundation** (Windows Presentation Foundation de Microsoft) vous permet d'intégrer directement dans les images de zenon des éléments graphiques, tels que des boutons, des pointeurs et de nombreux autres éléments utilisant ce nouveau format graphique. Ces éléments peuvent être librement liés à des variables et des fonctions, qui vous permettent de mettre en œuvre les effets graphiques de votre choix.
- **Interface COM :**
  Vous disposez d'un accès externe à zenon via l'interface COM entièrement visible. Différents langages de programmation, tels que C++, .NET, Delphi etc. sont fournis aux fins des communications.

4.2 Global et central plutôt que local

Les propriétés peuvent être définies de nombreuses manières dans zenon, en fonction de la tâche :

- Directement sur l'objet
- De manière centrale
- De manière globale

Bien entendu, dans zenon, vous pouvez ajuster tous les paramètres directement au niveau de l'objet. Toutefois, cela n'est pas toujours la solution la plus intelligente.

DÉFINIR LES PROPRIÉTÉS DE MANIÈRE CENTRALE

Il est souvent logique de définir les paramètres une fois seulement, de manière centrale. De cette manière, les nouveaux objets proposent les propriétés souhaitées dès le départ. Les modifications sont très faciles à mettre en œuvre : la propriété souhaitée est modifiée de manière centrale, à un endroit seulement ; tous les objets concernés l'adoptent automatiquement. Cela signifie un clic seulement, au lieu d'innombrables lignes de code ou de nombreux clics de souris.

Ces paramètres centraux peuvent être localisés avec facilité depuis n'importe quelle position dans Éditeur. Tout objet recevant directement ses paramètres d'un emplacement central offre la possibilité de les afficher directement. Il vous suffit simplement de suivre une liaison pour voir immédiatement de quelle manière ces éléments sont liés.

Toutefois, si vous pouvez définir les propriétés de manière centrale pour un projet, vous pouvez également les définir globalement.

GESTION DE PROJET GLOBALE

zenon vous permet de gérer plusieurs projets simultanément – et ainsi d'économiser votre ressource la plus précieuse : les heures de travail. Par exemple, il est très facile d'utiliser les mêmes polices de caractères dans tous vos projets. Une fois définies, les propriétés telles que les polices ou les tailles de polices sont instantanément disponibles pour tous les projets d'un domaine de travail. Si vous avez besoin d'une autre police de caractères pour tous vos projets, il ne vous faudra que quelques secondes pour la modifier dans le projet global. Et si vous souhaitez fournir un projet avec une police de caractères différente ? Dans ce cas, il vous suffit de modifier la police concernée de manière centrale dans le projet.

Information

Dans le projet global, utilisez un système de numérotation différent du système employé dans les projets individuels. zenon privilégie toujours les paramètres locaux si les nombres sont identiques.

Et : Veillez à utiliser des noms uniques pour les listes de polices, et veillez à ce que l'utilisation de ces noms soit cohérente dans le projet global et les projets individuels. De cette manière, vous veillez à ce que toutes les polices soient bien retrouvées en cas de changement de langue.

APERÇU DES RELATIONS ENTRE OBJETS

Vous disposez en outre d'un aperçu de vos projets et des relations entre les objets grâce à ces éléments :

- **Liste des références croisées :**
  cette liste présente les références croisées dans un projet et vous permet de rechercher facilement l'utilisation des variables et des fonctions.
- **Assistant documentation**
  Crée automatiquement une documentation complète pour votre projet. Vous pouvez configurer exactement ce que vous souhaitez inclure, jusqu'à la moindre propriété.

4.3 Paramétrage orienté objet

Le paramétrage orienté objet constitue la philosophie fondamentale de zenon. Cette philosophie permet d'organiser le travail de manière claire, et de consacrer davantage de temps à la création et la gestion de variables.

Chaque variable dans zenon repose sur deux éléments :

- Driver (à la page 12)
- Type de données (à la page 14)

4.3.1 Drivers

Les drivers ne sont pas directement intégrés à zenon ; ils sont mis en œuvre via un type d'objet de driver. Ceci détermine la zone du contrôleur concernée par l'adressage.

TYPE D'OBJET DRIVER

Il existe de nombreux types d'objets de drivers, par exemple, les marqueurs d'automates standard, les blocs de données, les entrées, les sorties, les compteurs etc. – mais également d'autres types spéciaux, tels que les alarmes ou les variables de drivers.

Le type d'objet de driver détermine :

- La zone du contrôleur
- La granularité du driver dans cette zone
- Les types de données pouvant être créés dans cette zone

Conseil : tous les types de données ne peuvent pas être créés dans un domaine. Ici, lors de la création d'une variable, vous devez toujours :

- D'abord choisir le driver,
- Puis le type d'objet de driver et
- Enfin, le type de données.

INFORMATIONS SUPPLÉMENTAIRES : GRANULARITÉ

La granularité est particulièrement importante pour les contrôleurs à adressage numérique, tel que le modèle Siemens S7, par exemple. Tous les contrôleurs ne possèdent pas la même résolution, et toutes les zones d'un contrôleur ne possèdent pas non plus la même résolution.

Par exemple, la zone de blocs de données d'un automate Siemens S5 est orientée mot, tandis que sa zone de marqueurs est orientée octet. Cela signifie que la plus petite unité pouvant être adressée est soit un octet, soit un mot. Si vous voulez écrire un bit dans l'une de ces zones, vous devez lire au moins un octet ou un mot entier, marquer le bit souhaité, le modifier, puis écrire à nouveau l'octet ou le mot entier dans le contrôleur.

Les mêmes informations sont requises lors de l'adressage des zones, appelées "offset". Vous commencez par zéro et continuez à compter en incréments par octets ou mots. Si vous utilisez l'adressage automatique (à la page 19) dans zenon, la granularité est automatiquement prise en compte. La granularité ne dépend pas du driver, mais du type d'objet de driver.

VARIABLES DU DRIVER

Les variables de drivers sont proposées par tous les drivers, et offrent de nombreux avantages en termes de configuration de projets. Elles ne communiquent pas avec le contrôleur, mais lisent une zone d'enregistrement interne du driver contenant principalement des informations statistiques. Toutefois, des fonctions spéciales peuvent être contrôlées avec des variables ; par exemple, les numéros de téléphone pour les connexions par modem ou les commandes de numérotation ou de fin d'appel.

Vous trouverez des informations plus détaillées à ce sujet au chapitre Variables de driver de l'aide.

Un fichier dBase contenant les plus importantes variables de drivers est inclus sur le support d'installation de zenon. Vous pouvez ainsi importer des variables pour chaque driver, au lieu de les créer manuellement.

4.3.2 Type de données

Les types de donnés sont au cœur même du paramétrage orienté objet. Puisque vous ne pouvez rien modifier dans les -types d'objet de drivers, ces derniers ne peuvent pas être utilisés dans le cadre de l'orientation objet. Les types de données, en revanche, offrent de nombreuses possibilités.

On distingue trois types :

- Types de données simples
- Types de données de structure
- Éléments de structure

4.3.2.1 Types de données simples

Les types de données simples sont toujours des types de données IEC, ce qui signifie les types de données définies par IEC dans le standard 61131-3, par exemple, BOOL, INT, USINT, UDINT, STRING, WSTRING, etc.. La longueur de la plage de valeurs est définie.

Exemples :

- BOOL : *1* Bit
- INT : *16* bits non signés
- UINT : *16* bits non signés

Les types de données dans zenon autorisent non seulement l'utilisation des types de données IEC ; de nombreuses autres propriétés sont disponibles. Ils comportent la même identification que les propriétés des variables :

- Identification
- Unité
- Plage de valeurs
- Valeurs limite

Quel est l'intérêt de cette "double conservation des données" ?

La raison est aussi simple qu'elle est pratique :
À l'image des éléments dynamiques, qui reprennent le type et la taille de la police de caractères liée, la

variable reprend le calcul de la valeur, l'unité et les valeurs limites provenant du type de données sur lequel elle est fondée.

ORIENTATION OBJET

Contrairement à la liaison d'éléments dynamiques et de polices de caractères, ceci implique une orientation objet : les objets de type données transmettent leurs propriétés à la variable. La différence ?

Les propriétés liées/dérivées peuvent être séparées ou remplacées pour les variables. Cela fonctionne individuellement, pour chaque propriété, mais également pour toutes les propriétés à la fois. Cela signifie que vous pouvez reprendre l'unité issue du type de données de manière centrale, mais remplacer directement l'identification ou l'adresse pour la variable. Cela se déroule comme suit :

- Modifiez la valeur concernée de la variable lorsque la liaison est interrompue.
  Vous pouvez vérifier cela en consultant les propriétés : la coche n'est plus présente.

Vous pouvez utiliser les fonctions suivantes depuis le menu contextuel dans les propriétés du projet :

- **Dériver une propriété du type de données :**

  Si vous avez dissocié un lien, vous pouvez le rétablir.

- **Retirer le lien d'une propriété du type de données :**

  Vous pouvez dissocier une propriété individuelle sans devoir modifier la valeur. Si vous remplacez ultérieurement le type de données, vous pouvez être certain que la variable ne sera pas influencée par cette action. Vous pouvez définir si le concept d'héritage doit agir ou non.

- **Lier toutes les propriétés avec le type de données**
  Lie toutes les propriétés dans le nœud de propriétés vers le type de données.

- **Retirer le lien de toutes les propriétés du type de données**
  Déconnecte toutes les propriétés dans le nœud de propriétés du type de données.

Pour ouvrir le menu contextuel, cliquez sur une propriété dans les propriétés du projet avec le bouton droit de la souris.

Pour vous faciliter la tâche, vous pouvez également accepter ou séparer toutes les propriétés du type de données d'un clic de souris. Vous pouvez ainsi restaurer très rapidement l'état d'origine (l'ensemble des éléments dérivés) ou annuler cette modification. En quoi est-ce utile, concrètement ?

Flexibilité particulièrement élevée : si vous utilisez de nombreuses variables possédant la même valeur limite, par exemple, une alarme pour la valeur 1, définissez cela au niveau du type de données. Vous définissez le texte de valeur limite et les autres paramètres facultatifs séparément, pour chaque variable. Vous enregistrez ainsi la création et la maintenance de la valeur limite pour chaque variable individuelle, et votre rapidité de développement s'accroît considérablement. Le type de données BOOL, créé par défaut, comporte également une valeur limite à 0 et 1 !

Si vous n'avez même pas défini une valeur limite pour chaque variable bool, vous n'avez pas besoin de vous préoccuper de chaque variable et de désactiver la valeur limite. Au lieu de cela, vous pouvez

utiliser une fonction extrêmement pratique de zenon et créer votre propre type de données BOOL. Suivez ces instructions :

- Laissez simplement le type de données par défaut tel quel.
- Créer un nouveau type de données (par l'entrée **Nouveau type de données simple...** dans le gestionnaire de projet dans le menu contextuel de l'entrée **Types de données** ou par l'entrée dans la barre d'outils)
- Donnez-lui un nom, p. ex. *MyBool*
- Une liste des types de données existants s'affiche à l'écran.
- Choisissez *booléennes standard* :
- Toutes les propriétés du nouveau type de données sont automatiquement reprises de l'ancien type.
- Supprimez les valeurs limites pour *MyBool*

Vous créez ensuite toutes les variables bool qui ne doivent pas comporter de valeurs limites pour ce nouveau type de données bool *MyBool* ; les valeurs limites indésirables ont d'ores et déjà disparu. En revanche, celles que vous souhaitez utiliser sont conservées.

Vous ne pouvez bien entendu pas créer un seul type de données bool ; vous pouvez en créer autant que vous le souhaitez. Toutes les variables qui nécessitent des propriétés identiques sur des plages étendues reçoivent leur propre type de données. Si une propriété est modifiée, effectuez la modification de manière centrale ; toutes les variables dérivées seront automatiquement modifiées, à l'exception de celles pour lesquelles le lien a été modifiée.

4.3.2.2 Structure data types - structure elements - structure array

STRUCTURE DATA TYPE

Structure data type means a group of data types that are precisely defined in terms of their order and arrangement. For example, the structure motor can consist of the elements of actual speed and power consumption.

A structure can also be nested, whereby as many hierarchy levels as desired are possible.

With structure data types, the variable household can be be structured in precisely the same way as it actually exists in the controller or logically.

STRUCTURE ELEMENT

A structure data type is actually just a hull that bears the name of the structure but does not have any properties of its own.

These first come with the structure elements that are added to the structure data type. In doing so, a distinction is made between:

- Linked structure data type:

  Reference to a pre-existing data type. All properties are taken on, with the exception of the name.

  Advantage: If the structure element has many properties of a pre-existing data type, these can easily be reused. It is of course still possible to change each of the variable properties or separate them from the data type.

- Separate embedded structure data type:

  Allows individual addressing settings. This new data type is only valid within this structure.

> **Information**
>
> Advantage of object orientation: If the structure changes at any point, it is then sufficient to change the structure data type in order to automatically amend all variables that are on this structure.
>
> Changes are possible for all forms:
>
> - simple properties like the unit of the data type
> - complex properties like adding or deleting a limit value
> - Change to the complete structure, such as a change to the sequence of elements or addition or deletion of individual structure elements in a structure data type

STRUCTURE ARRAY

If it is not just one machine, but many, the structure array is used.

For example: 100 Pumps instead of 1 pump.
You simply change the structure variables to a structure array. 100 variables thus become 100 structure variables with the click of a mouse.

If we, for example, have to create 12 individual variables for each pump, we need a total of 1200 variables. The method of "setting parameters in an object-orientated manner" allows this with a few mouse clicks. In addition there is the advantage that all variables have already been preset. Each individual variable already provides all properties that it needs, such as unit, value determination, alarms, CEL entries etc.

> **Information**
>
> Structure data types are also suitable for the reuse (à la page 32) of variables.

4.3.2.2.1Example pump:

INITIAL SITUATION

A pump consists of two motors. Each of these motors has some variables like:

- *Actual speed (RPM)*
- *Power consumption*
- *Output*
- etc.

Each motor has a slider with the following variables:

- *Nominal speed (RPM)*
- *P-share*
- *I-share*

You build this constellation as directed:

1. Create a structure **slider** from one data type each for
   - *Nominal speed (RPM)*
   - *P-share*
   - *I-share*
2. Create a structure **motor** with these data types each:
   - *Actual speed (RPM)*
   - *Power consumption*
   - *Output*
   - etc.
3. Adopt the **slider** in the structure **motor** .
4. Finally, create the pump structure and integrate the motor into it.

   As we have two motors in our pump, you simply add the motor data type to your **pump** structure twice.
5. Create a variable that is based on this structure data type:
   - Create new variable
   - Choose structure data type **pump**

The individual elements of this structure variable are called Structure elements (à la page 16) .You can use each of these elements everywhere in zenon- in screens, as alarms, in archives, in recipes, etc.

4.3.3 Conseils relatifs à l'adressage et à l'importation/exportation

Conseils pratiques pour :

- Adressage
- Exporter/Importer

ADRESSAGE :

Avec les contrôleurs numériques, vous avez le choix entre un adressage automatique ou semi-automatique. Les propriétés décrites ici peuvent également être atteintes par l'intermédiaire de VBA/VSTA et être ainsi utilisées aux fins de la configuration automatique.

ADRESSAGE AUTOMATIQUE :

L'offset et, le cas échéant, les adresses d'octet et de bit sont calculés automatiquement sur la base de la position des éléments de structure ; avec les tableaux, ils sont calculés sur l'ensemble du tableau. Si vous avez créé la même structure dans l'automate et dans zenon, tout fonctionnera parfaitement et vous n'aurez pas à vous préoccuper de l'adressage.

ADRESSAGE SEMI-AUTOMATIQUE :

Vous définissez déjà des adresses de début individuelles pour le type de données. Les autres adresses sont ensuite calculées à l'aide de ces adresses. Vous pouvez également les modifier si nécessaire. Vous trouvez plus d'informations au chapitre Adressage automatique.

Pour les automates comportant un adressage à base de symboles, vous devez utiliser le même nom dans zenon et dans l'automate.

EXPORT/IMPORT XML :

Les informations telles que les types de données, les structures, les propriétés héritées etc. sont également incluses lors de l'importation ou de l'exportation. Cela signifie qu'une fois définies, les types de données de structure et les variables de structure peuvent être facilement exportées, puis importées dans d'autres projets pour être réutilisées ou adaptées, selon le besoin.

5 Réutilisation d'éléments

La première fois qu'un projet est créé, vous devez prévoir du temps pour créer les variables, fonctions, synoptiques et liens associés. Ce temps peut toutefois être considérablement réduit lors des projets ultérieurs. zenon vous offre en effet la possibilité de transférer facilement les éléments créés vers d'autres projets.

Par exemple, dans de nombreux projets, vous devez créer les composants standard de tous les projets, tels que les synoptiques d'état du système ou les synoptiques détaillés des composants matériels utilisés de manière récurrente (pompes, vannes, moteurs, etc.).

Il existe différentes manières de réutiliser les composants :

Thème	Réutilisation de :
Remplacement de variables et de fonctions (à la page 20)	Éléments
Symboles (à la page 25)	Synoptiques
Types de données de structure (à la page 32)	Variables
Matrices de réaction (à la page 34)	Variables
Projet global (à la page 34)	Éléments centraux d'un projet
XML (à la page 36)	Importation et exportation de composants de projets
Assistants (à la page 37)	Importation ou personnalisation de synoptiques et d'éléments
Réutilisation de projets (à la page 38)	Projets

Information

Pour réutiliser efficacement des éléments, lors de la création de variables, de fonctions et de synoptiques, assurez-vous de les nommer (à la page 21) de manière claire.

5.1 Remplacement de variables et de fonctions

Les variables et fonctions conservées sous forme d'éléments dynamiques peuvent être remplacés de manière automatique. Cette opération peut être effectuée de différentes manières :

- Remplacement des liens d'un synoptique (à la page 22)
- Remplacement des liens lors des appels de synoptique (à la page 23)
- Remplacement d'indices (à la page 24)
- D'autres informations concernant le remplacement de variables et de fonctions sont également disponibles dans le manuel consacré aux synoptiques de la section Remplacement des liens de variables et de fonctions.

5.1.1 Naming conventions

To be able to replace variables and other elements securely, the naming should be systematic and standardized if possible. You therefore support not only the reusability, but also maintenance and reverse engineering.

Different systems support you with systematic naming.

FOR EXAMPLE: ENERGY INDUSTRY

Germany

- KKS (Kraftwerk-Kennzeichen-System - Power Plant Classification System), for details (in German), see http://de.wikipedia.org/wiki/Kraftwerk-Kennzeichensystem (http://de.wikipedia.org/wiki/Kraftwerk-Kennzeichensystem)
- DIN 6779 (Kennzeichnungssystematik für technische Produkte und technische Produktdokumentation - Classification System for Technical Products and Technical Product Documentation), for details (in German), see http://de.wikipedia.org/wiki/DIN_6779 (http://de.wikipedia.org/wiki/DIN_6779)
- Equipment Classification System, for details (in German "Anlagenkennzeichnungssystem"), see http://de.wikipedia.org/wiki/Anlagenkennzeichnungssystem (http://de.wikipedia.org/wiki/Anlagenkennzeichnungssystem)

International

- KKS (Power Plant Classification System), for details, see http://en.wikipedia.org/wiki/KKS_Power_Plant_Classification_System (http://en.wikipedia.org/wiki/KKS_Power_Plant_Classification_System)

Such standards exist for all industries. It is recommended that their naming convention is used.

KKS EXAMPLE:

Variables are to be named in accordance with the KKS in an energy project.
A corresponding variable with the label **C01_MDY10-QA001_QA07** indicates:

- Wind energy equipment **C01** (row **C**, no. **1**)

- Wind turbine control **MDY10**
- Power part **QA001**
- Power protection **QA07**

5.1.2 Possibilités de remplacement

La fonction de remplacement peut être utilisée à différents points dans un projet :

- Remplacement des liens d'un synoptique (à la page 22) : les synoptiques sont copiés, et les liens sont remplacés dans le synoptique copié.
- Remplacement des liens lors des appels de synoptique (à la page 23) : un seul synoptique est utilisé pour différents automates ; les liens sont adaptés lors de l'appel de synoptique.
- Remplacement d'indices (à la page 24) : remplacement des variables dans un synoptique de procédé sur la base de la valeur des variables d'indices.

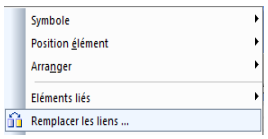
5.1.2.1 Remplacement des liens d'un synoptique :

Si des variables sont associées à un synoptique, elles peuvent facilement être remplacées par le biais d'une boîte de dialogue de remplacement. Conditions requises :

- Une convention d'affectation de noms claire pour les variables (à la page 21).

Celle-ci permet de copier&coller, puis continuer à utiliser les synoptiques créés. Le remplacement est exécuté par le biais du menu contextuel :

- Cliquez avec le bouton droit sur l'élément de synoptique.
- Cliquez sur **Remplacer les liens**.



La boîte de dialogue de remplacement est alors ouverte.

EXEMPLE

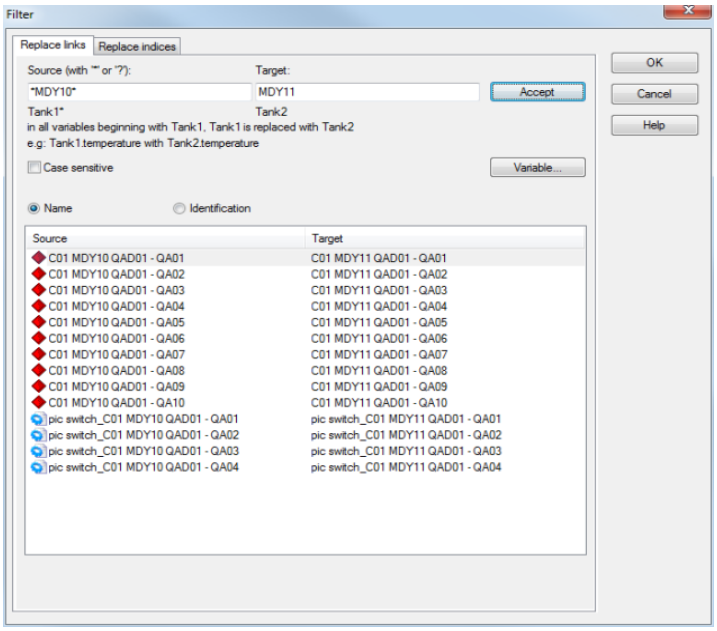
Dans notre exemple, le projet comporte plusieurs variables issues de différentes composantes de l'équipement. Les éléments suivants sont présents dans le synoptique de procédé :

- 10 variables issues du contrôleur de turbine éolienne **MDY10**, et comportent des éléments dynamiques liés

- Des boutons d'appel de synoptique présents à différents emplacements dans le contrôleur **MDY10**

En effectuant un manipulation copier&coller, la personne configurant le projet compte réutiliser le synoptique dans sa forme exacte pour le contrôleur de turbine éolienne **MDY11** ou remplacer les variables et/ou les fonctions par les variables et fonctions correspondantes provenant du nouvel automate. Pour cela :

1. Les éléments dynamiques auxquels sont liées les variables et fonctions sont affichés en surbrillance.
2. La boîte de dialogue de remplacement s'affiche



3. Dans le champ **Source**, l'identifiant d'équipement **MDY10** est saisi
(Les premier et dernier caractères ***** sont des opérateurs)
4. L'expression *MDY11* est saisie dans le champ **Cible**
5. Déclencher la procédure de remplacement en cliquant sur **Accepter**, puis fermer la boîte de dialogue de remplacement en cliquant sur **OK**

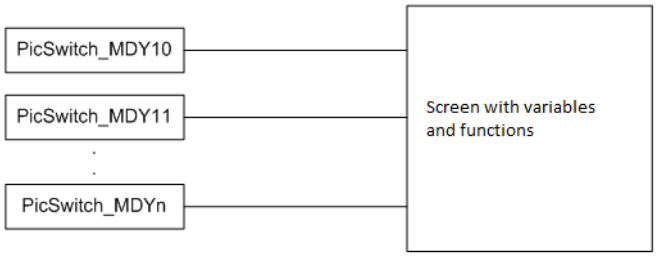
Le synoptique peut maintenant être utilisé pour le nouvel automate.

5.1.2.2 Remplacez des liens lors des appels de synoptique

Avec cette méthode, le synoptique d'origine est toujours utilisé et appelé dans le Runtime avec différentes variables et fonctions. Le synoptique contient différentes variables et fonctions, comme dans l'exemple Remplacement des liens de synoptiques (à la page 22). Le remplacement est effectué durant l'appel de synoptique. Pour cela :

1. Plusieurs fonctions d'appel de synoptique sont configurées sur ce synoptique.

2. La boîte de dialogue de remplacement (à la page 22) est proposée lors de la création de la fonction.
3. Chaque fonction contient un procédé de remplacement propre.



Une variable pour cela, qui n'apparaît que lors de l'appel de synoptique : Remplacez des indices (à la page 24) dans les tableaux.

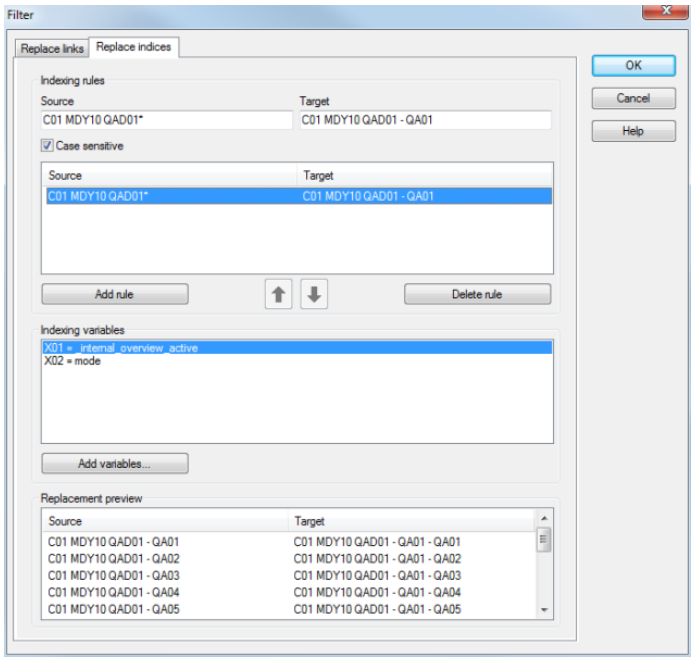
5.1.2.3 Remplacement des indices

Dans les tableaux, le remplacement de variables peut être effectué par le biais des variables d'indices.

Comme avec le remplacement des liens lors des appels de synoptiques (à la page 23), le synoptique est utilisé une fois seulement. Les variables liées sont remplacées durant la commutation. Les appels de synoptique ne sont configurés qu'une fois (contrairement au remplacement des liens de synoptiques (à la page 22)), et peuvent être réutilisés plusieurs fois par le biais d'une variable d'indice.

EXEMPLE

Un appel de synoptique est exécuté sur un synoptique de procédé contenant 10 variables pour le contrôleur de turbine éolienne **MDY10**. L'objectif de l'opérateur configurant le projet consiste à réutiliser le synoptique à l'échelle 1:1, car les contrôleurs de turbine éolienne **MDY11**, **MDY12** et **MDY13** comportent le même nombre de variables.



Règle de gestion des indices :

- Source : C01 MDY10 QA001*
- But : C01 MDY{X01} QA001

Ceci a l'effet suivant :

- Lors de l'appel de synoptique, la fonction est informée qu'elle doit utiliser la valeur de variable du **contrôleur de turbine éolienne X01** dans le nom de la variable utilisé pour l'appel de synoptique des synoptiques dans le Runtime.

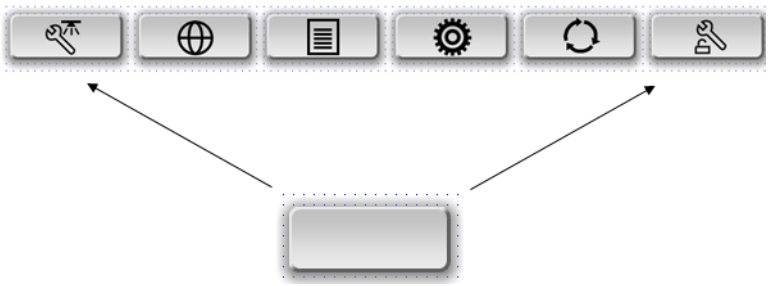
  Exemple : Si cette variable possède une valeur de 12, le synoptique de procédé contenant toutes les variables du contrôleur de turbine éolienne **MDY12** sont affichées lors de l'exécution de l'appel de synoptique.

5.2 Symbols

Symbols offer great potential for reuse. Symbols can be built up in a very complex manner and support inheritance. Symbols can be embedded or linked. With linking, modification in one place is sufficient to update all screens that use this symbol. Symbols can also be linked to other symbols.

EXAMPLE OF BUTTON BAR

A symbol (empty right corner) is linked as a basis in all buttons.



Free-access properties (à la page 26) make it possible to issue each button with different graphics. If the form or color of the buttons is changed, only the "empty button" symbol needs to be modified. All buttons automatically take on the new form and/or color.

5.2.1 Propriétés d'accès libre

Les symboles transmettent leurs propriétés aux objets auxquels ils sont liés. Toutefois, la transmission de propriétés individuelles peut être annulée. Prenons l'exemple de l'élément graphique affiché dans notre bouton. Si la transmission est annulée, cette propriété peut être définie individuellement pour chaque objet. Les modifications apportées à ce détail dans l'élément initial n'ont plus d'incidence sur les autres objets.

RÉUTILISATION DE SYNOPTIQUES

L'association de symboles et de propriétés libérées peut être une solution efficace à la réutilisation de synoptiques de procédé.

EXEMPLE

Une boîte de dialogue centrale de configuration des paramètres d'un projet est utilisée pour la configuration de différents paramètres. Cette fonctionnalité étant utilisée dans différents domaines, il est également important que certaines adaptations (telles que les couleurs d'arrière-plan, les synoptiques…) soient possibles.

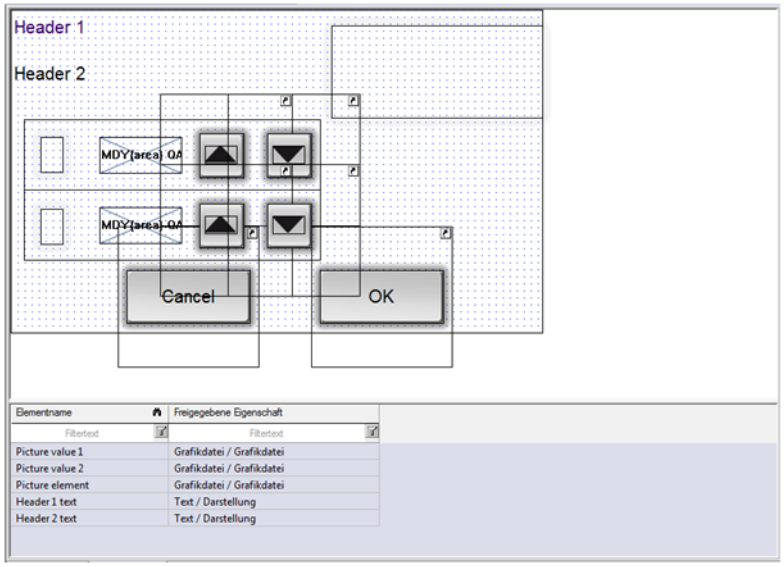
Copier et coller :

1. Créez un synoptique de procédé contenant cette boîte de dialogue de configuration des paramètres.
2. Créez une copie du synoptique en effectuant un copier-coller.
3. Les synoptiques de procédé qui ont été dupliqués peuvent être adaptés graphiquement aux besoins du projet.

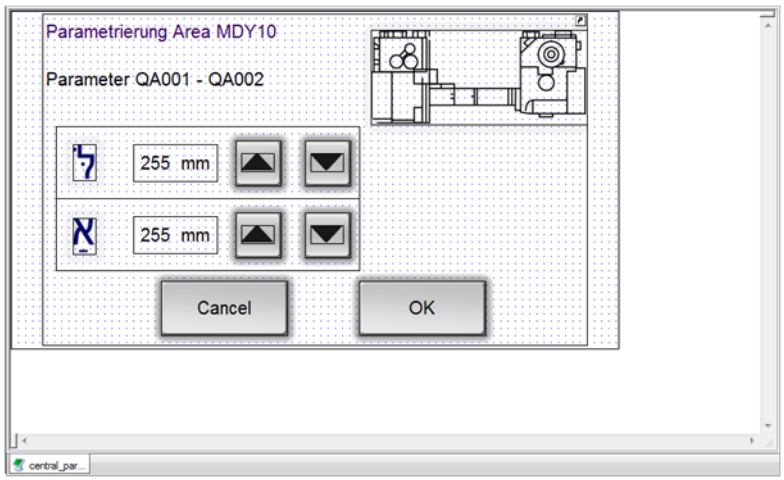
Toutefois : L'héritage des propriétés n'est pas possible ici. Si l'opérateur configurant le projet souhaite ultérieurement effectuer des modifications de manière centralisée (par rapport à la structure de base de la boîte de dialogue de configuration de paramètres), ils doivent d'abord faire glisser ces éléments individuellement vers chaque synoptique de procédé.

Approche de l'utilisation des symboles avec des propriétés non héritées :

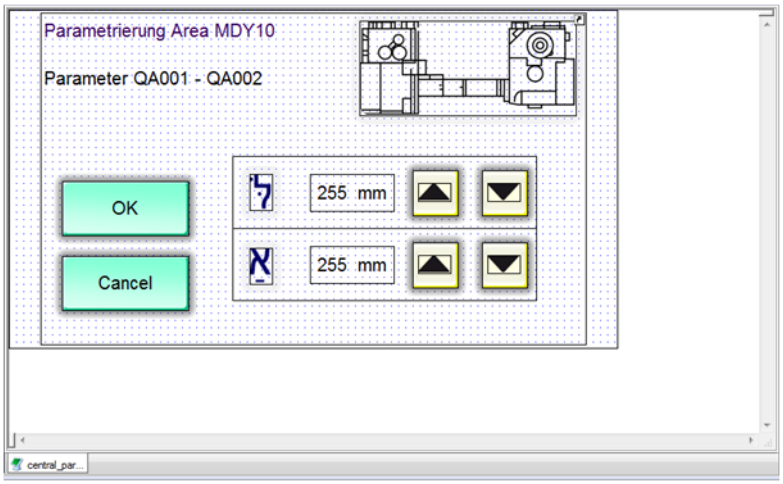
1. La boîte de dialogue de configuration des paramètres est créée avec un symbole.
2. Les éléments devant être adaptés à différents points de paramètres sont dissociés du concept d'héritage par le biais des propriétés non héritées.
3. Le symbole est lié, positionné et adapté aux synoptiques correspondants.



4. Si la personne configurant le projet décide de modifier l'apparence générale de la boîte de dialogue, elle doit effectuer cette modification de manière centrale dans le symbole. Cette modification affecte alors tous les éléments auxquels le symbole est lié.



5. Les symboles globale peuvent être adaptés individuellement, à l'aide des propriétés non transmises et de la fonction de remplacement (à la page 28) correspondante.



6. Dans ce cas, l'opérateur configurant le projet a décidé de modifier l'interface utilisateur graphique générale de la boîte de dialogue de configuration des paramètres. Cette modification est mise en œuvre de manière centrale, via le symbole de configuration des paramètres. Les propriétés individuelles des boîtes de dialogue de configuration des paramètres individuels restent inchangées ; seules les propriétés contenues dans l'héritage sont présentes.

5.2.2 Remplacement par des symboles

Le procédé décrit dans la section Remplacement de variables et de fonctions (à la page 20) peut également être effectué pour les symboles. Ici aussi, il est impératif de réfléchir à l'd'affectation de noms (à la page 21) des objets.

Lors du remplacement de symboles, il est important de distinguer deux approches différentes : un symbole issu d'une bibliothèque (de projet ou globale) peut être :

- Inséré dans un synoptique sous forme de groupe d'éléments
- Lié dans un synoptique

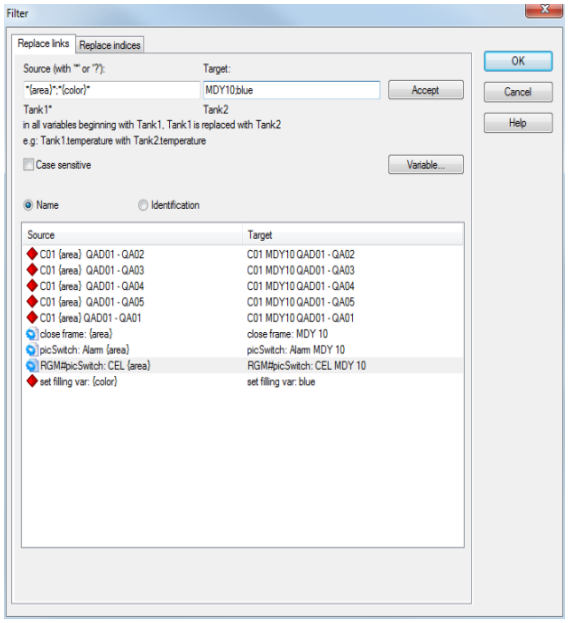
Selon l'approche choisie, différentes boîtes de dialogue sont affichées pour le remplacement des variables et fonctions.

GROUPES D'ÉLÉMENTS

Dans ce cas, les variables sont directement remplacées par le biais de la boîte de dialogue de remplacement. La boîte de dialogue offre la possibilité de sélectionner des variables, par le biais d'une boîte de dialogue de sélection de variables. L'utilisation de symboles de cette manière n'est idéalement pas adaptée à la réutilisation.

SYMBOLES

Ce procédé permet d'effectuer les opérations avec des chaînes de remplacement. Ces chaînes sont remplacées par les variables correspondantes durant la compilation. Si cela n'est pas possible par le biais de la boîte de dialogue affichée, les remplacements peuvent être effectués par le biais d'une boîte de dialogue de sélection de variables. L'opérateur configurant le projet utilise ici des chaînes de remplacement individuelles, séparées par un point-virgule (;).



Le symbole associé à cette boîte de dialogue de remplacement peut maintenant être réutilisé dans différents éléments de l'équipement. Cet exemple suppose que des variables et fonctions similaires sont utilisées dans chaque élément de l'équipement (différenciées uniquement par l'abréviation de l'équipement dans le nom), et que celles-ci peuvent être remplacées rapidement, et avec une vue d'ensemble claire.

Vous trouverez d'autres informations concernant le remplacement de symboles dans le manuel consacré aux synoptiques, dans la section Symboles.

5.2.3 Bibliothèque générale de symboles

Il existe plusieurs possibilités pour la gestion des symboles. Les symboles peuvent être gérés dans :

- La bibliothèque de symboles du projet local
- La bibliothèque de symboles du projet global
- La bibliothèque générale de symboles

Symbol Library	Property
General symbol library	Symbols are available in all projects. The **general symbol library** node is located in the project manager below the currently-loaded projects.

Symbol Library	Property
	Label when linking in the screen: **[symbol group]/[symbol name]** **Rules:** ▸ The dialog for selecting variables offers all projects of the workspace for linking. ▸ These symbols are saved in the zenon program folder and only updated when the Editor starts. These symbols are not saved during project backup. ▸ Interlockings and aliases for ALC cannot be configured. **Attention:** If symbols that contain interlockings or aliases are added, these settings are removed.
SYMBOL LIBRARY IN THE GLOBAL PROJECT	Symbols are available for all projects of the workspace. Label when linking in the screen: **Global project_[Symbol name]** **Rules:** ▸ The symbols copied into the symbol library of the global project retain their variable linking without changes. ▸ The variable dialog offers all projects of the workspace for linking. Linked variables are placed in front of the name of the respective project. ▸ The name of the **Variable** can be amended in the properties window. This way, for example, the prefix can be deleted with the project origin. ▸ The symbols are also backed up when a project is backed up. ▸ Interlockings and aliases for ALC cannot be configured. **Attention:** If symbols that contain interlockings or aliases are added, these settings are removed.
Symbol library in the project	Symbols are only available in the current project. Label when linking in the screen: **[Symbol name]** **Rules:** ▸ The symbols are saved in the project folder. The project symbol library is in the current project in the **Screens** node and is backed up together with project backup. ▸ Interlockings and aliases for ALC can be configured. These properties are also retained when symbols are added.

COPYING OF SYMBOLS BETWEEN LIBRARIES

Symbols can be copied with **Drag&Drop** .
For this, the following applies:

- Only symbols of the first level of the tree can be copied.
- Groups and folders cannot be copied.
- If there are already names of symbols in the target library, the added symbols are automatically renamed.
  Renaming is carried out by adding an underscore and an ordinal number, for example: **Calibrator_1**.
- Copying between libraries is only possible to a limited extent.

Overview of copying between symbol libraries using **Drag&Drop**:

Copying using Drag&Drop possible	General symbol library	Symbol library in the global project	Symbol library in the project
General symbol library	--	+	--
Symbol library in the global project	--	--	+
Symbol library in the project	--	+	--

Key:

- **+**: copying possible
- **--**: copying not possible

EXEMPLE

un point de configuration central compile une collection de symboles devant être utilisés par les points de configuration distribués dans différents projets individuels dans le monde entier. Par exemple, les symboles complexes affichant les parties importantes de synoptiques de procédés sont affichés. Cette collection de symboles est un dossier ou un fichier, par exemple *Global_Used_Symbols.SYM*. Si des modifications graphiques centrales sont apportées à cette collection globale de symboles, *Global_Used_Symbols.SYM* est simplement transmis à tous les points de configuration, où le remplacement est effectué. L'adaptation individuelle est effectuée par le biais des propriétés non héritées (à la page 26) et de la fonction de remplacement (à la page 28).

Il n'est pas toujours nécessaire de distribuer la collection entière de symboles. Une procédure d'exportation/importation de données XML (à la page 36) peut également être utilisée pour distribuer des symboles individuels sélectionnés.

Attention

L'utilisation de la bibliothèque de symboles globale doit être régie par des règles claires : les points de configuration locaux ne doivent pas modifier les symboles de la collection de symboles. Toute modification serait remplacée lors de la prochaine mise à jour.

5.3 Structure Data Type

Variables can be reused using structure data types. Complex structures can thus be created and the advantages of inheritance can be used.

EXAMPLE

The "motor" structure data type consists of three elements:

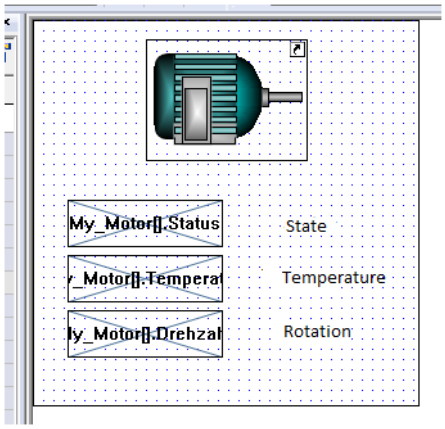
- Status
- Speed
- Temperature

motor	Structure datatype	0
status	Structure element	INT/<embed...
rotati...	Structure element	INT/<embed...
temp...	Structure element	INT

The inheritance type of the elements is different:

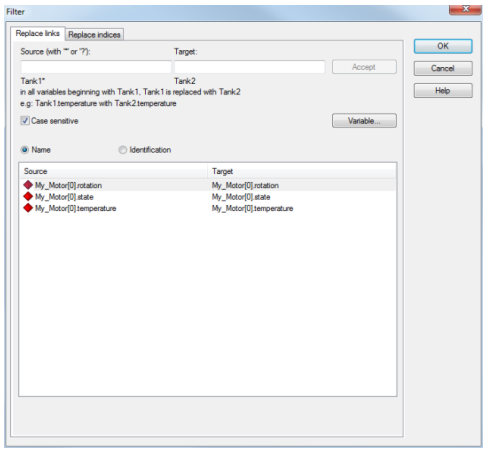
- **Status**" and "**Speed**" are derived from the *INT* data type, but their inheritance was broken off by the embedding.
  This means: Subsequent updating of the *INT* data type does not have an effect on these two elements.
- "**Temperature**" also comes from the *INT* data type, the inheritance is intact.
  This means: Subsequent changes to the *INT* data type also has an effect on the settings of this structure element.

If, for example 20 motors with a structure that is always identical are configured, variables based on the self-created "motor" data type can be created. This can, for example, happen by creating an array of this data type. The graphic display is triggered by means of a symbol in this example.



When linking the symbol in a process screen, the placeholder variables of the symbol are replaced by those of the self-created structure data type. Ensure that you have well-thought out naming (à la page 21) from the start.

If a process screen is created as an overview of all motors, the inserted symbol can be easily duplicated by copying & pasting. Only the target variable needs to be replaced with the corresponding sequence number on each symbol:



If limit values for the "speed" variables are to be added at a later time, you benefit from the existing inheritance display between the self-created data type and the variables that are created as a result. For the planned change, a new limit value is created for the "**speed**" structure element of the separate "*motor*" data type. This change made at a position has an effect on all 20 motors that have been created.

However the person configuring the project is free to undo this inheritance relationship. The limit value for "**speed**" can be changed for one of the 20 motor variables that have been created.

Note: Individual properties or all properties of a variable are disconnected from the data type.

Information

You can also read tips on the use of structure data types in the Structure data types - structure elements - structure array (à la page 16) section and in the Variables manual in the Structure data types section.

5.4 Matrices de réaction

Les matrices de réaction permettent de veiller à ce qu'une variable reste inchangée tout au long du projet. Contrairement aux valeurs limites, elles adoptent une approche centralisée : Une matrice de réaction est configurée une fois, puis attribuée à un nombre donné de variables. Toutes les variables liées à la matrice de réaction réagissent de la même manière. Cela vous offre l'avantage d'une maintenance centralisée et simple.

5.5 Projet global

Dans un projet global, des paramètres standard peuvent être définis à l'échelle du projet. Ceci permet de définir certains modules ou éléments pour tous les sites, tandis que les projets locaux définissent des paramètres individuels.

Les éléments suivants peuvent être configurés dans un projet global :

- **Gestion d'alarmes :**

  Déclaration des groupes d'alarmes, des classes d'alarmes et des zones d'alarmes à utiliser à l'échelle globale.

- **Modélisation des équipements :**

  Cet élément est compatible avec la configuration centrale du projet, une division peut être configurée sous la forme de deux "usines". Dans ce cas, les éléments de configuration du projet provenant du point de configuration central du projet et ceux provenant du point de configuration du projet local doivent être configurés.

- **Utilisateur :**

  Si une norme impose l'existence de certains utilisateurs globalement valides, ceux-ci peuvent être déterminés par l'intermédiaire du projet global.

- **Fichiers :**

  Une norme peut, par exemple, imposer l'utilisation d'éléments graphiques devant être utilisés dans des projets créés individuellement. Ces éléments peuvent être fournis à l'échelle globale.

- **Cadres :**

  Ces éléments définissent la disposition du synoptique de procédé et la taille des projets à développer. Par exemple, il est possible de clairement définir, dans un projet, une directive

selon laquelle les nouveaux synoptiques peuvent uniquement être créés à l'aide de modèles d'un projet global. Ceci permet de garantir l'uniformité des éléments graphiques à l'échelle du projet.

- **Polices et palettes de couleurs :**

  Pour promouvoir l'uniformité des textes et couleurs affichés, un ensemble prédéfini de listes de polices et/ou de palettes de couleurs peut être défini pour une utilisation globale.

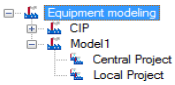
- **Tables de langues :**

  Si un ensemble déterminé de conditions et de déclarations doit être utilisé dans la configuration d'un projet, il est possible de définir ces éléments à l'aide des tables de langues définies et de réaliser, dans le même temps, les traductions correspondantes.

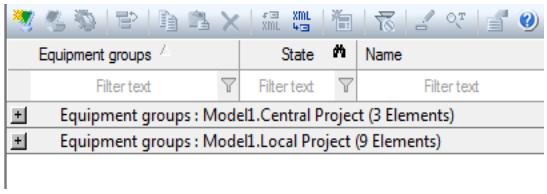
EXEMPLE

Un point de configuration central du projet définit les standards concernant les couleurs, les polices, les modèles, etc. par le biais d'un projet global, pour toutes les personnes procédant à la configuration du projet. Le projet global est distribué à toutes les équipes sous forme de sauvegarde de projet (à la page 38). Les équipes peuvent alors l'importer dans leur espace de travail. Une directive doit être utilisée pour stipuler clairement que les points de configuration du projet local ne doivent pas apporter de modifications à un projet global, car ils sont régulièrement remplacés par la saisie des mises à jour provenant du point de configuration central du projet.

EXEMPLE D'UN MODÈLE D'ÉQUIPEMENT



Un modèle d'équipement simple, qui fait uniquement une distinction entre un groupe de configuration central et local. Dans un projet individuel, cette distinction peut désormais être utilisée pour faciliter la visualisation du projet dans Éditeur. Dans le détail, cela donnerait l'impression que chaque élément configuré dans Éditeur a été attribué à un groupe d'équipements correspondant.



Les synoptiques de procédé existants sont affichés après chaque attribution d'équipements.

Le point de configuration du projet local peut ainsi clairement détecter quels synoptiques sont mis à jour par un point de configuration de projet central via l'importation au format XML. Pour la personne assurant la configuration du projet local, les synoptiques ne devant pas être modifiés peuvent ainsi être

clairement identifiés. En effet, chaque modification apportée à un synoptique global est remplacée lors de la mise à jour suivante.

Conseil : pour mettre un projet global à jour, il n'est pas toujours nécessaire de créer et d'importer une sauvegarde de projet. Les modifications individuelles peuvent également être mises à jour localement, grâce à l'exportation et l'importation au format XML (à la page 36).

5.6 XML

L'importation et l'exportation via l'interface XML offrent de nombreuses possibilités de réutilisation de composants de projets.

> **Attention**
>
> Les éléments existants portant le même nom sont remplacés durant l'importation.

DÉPENDANCES DURANT L'IMPORTATION

Si plusieurs éléments d'un projet doivent être importés, plusieurs fichiers XML avec différents contenus doivent être utilisés. Pour accepter correctement le contenu de différents fichiers XML, les dépendances existantes doivent être prises en compte :

Si, par exemple, un synoptique est importé et les variables associées n'existent pas dans le projet, les liens ne pourront pas être recréés. Pour créer correctement tous les liens, il peut être nécessaire d'importer deux fois un synoptique ou une variable.

Exemple : Importation d'un synoptique via XML

La variante la plus complète consiste à importer un synoptique. Dans ce cas, le fichier XML créé ne doit pas seulement contenir le synoptique et les éléments inclus, mais aussi le cadre, les variables liées ainsi que les fonctions.

Le fichier XML a besoin de tout le contenu qui peut être vu dans le synoptique et est donc directement lié à celui-ci.

Sur le synoptique **A**, une fonction **B** est exécutée d'une pression sur un bouton, pour ensuite ouvrir le synoptique **A** :

- La fonction **B** nécessite le synoptique **A** pour que celui-ci puisse effectivement être lié à la fonction **B**.
- Le synoptique **A** nécessite la fonction **B** pour que celle-ci puisse effectivement être liée au bouton.

Importer :

1. D'abord la fonction

2. Puis le synoptique
3. Enfin la même fonction une fois de plus
   Remarque : La référence à l'écran ne peut pas être établie autrement dans la fonction.

HARMONISER LES PROJETS AU SEIN D'UNE USINE

L'exportation/l'importation au format XML est également utile pour garantir l'harmonisation de projets à l'échelle d'une usine. Si une entreprise détermine que certaines composantes d'un projet doivent être conçues de manière uniforme à l'échelle mondiale, les éléments uniformes sont les suivants :

1. Configuration centrale
2. Exportation sous forme de fichier XML
3. Distribution sous forme de fichier XML
4. Acceptation par chaque équipe locale sous forme d'importation au format XML

Une autre méthode consiste à effectuer l'exportation et l'importation à l'aide d'un assistant spécifique.

Parce que les éléments existants sont remplacés durant l'importation au format XML, ces éléments uniformes doivent être nommés de manière claire, conformément à une convention obligatoire. Toute modification locale de ces éléments sera remplacée lors de la prochaine importation.

ASSISTANTS (À LA PAGE 37)

zenon propose un assistant pour aider l'utilisateur à accomplir les tâches d'importation. L'assistant permet à l'utilisateur de sélectionner ce qu'il souhaite importer en le guidant dans ses décisions. Une des méthodes d'utilisation de l'assistant consiste à créer une bibliothèque au format XML contenant différentes variantes des synoptiques et fonctions.

CONFIGURATION AVEC L'ASSISTANT

Un assistant peut également être utilisé pour personnaliser des fonctions standard, présentes au format XML. Une fonction est importée au format XML, puis adaptée aux exigences spécifiques du projet en modifiant certaines propriétés variables selon les objets. Par exemple, une fonction d'appel de synoptique au format XML peut être réutilisée plusieurs fois, grâce à la personnalisation des propriétés.

5.7 Assistants

Les assistants simplifient la réutilisation d'éléments et de composants du projet. Les assistants sont principalement adaptés aux tâches suivantes :

- **Création de projets de modèles :**

  Un assistant permet de créer certaines composantes d'un nouveau projet en quelques clics. Contrairement à l'importation simple d'une sauvegarde de projet (à la page 38) complète du

projet de modèle, l'opérateur configurant le projet dispose ici d'une plus grande capacité d'individualisation.

Par exemple, un assistant peut créer différents types de projet, quelle que soit la machine concernée.

- **Création de composantes prédéfinies d'un projet :**

  Les assistants peuvent également faciliter la création de certains éléments d'un projet.

  Par exemple : Création d'un élément de projet "**Moteur**".
  L'assistant crée des éléments prédéfinis dans cet élément. Il est ainsi possible de créer des synoptiques, des fonctions, des variables, etc. en quelques clics. Outre les délais de configuration réduits, vous pouvez également obtenir une apparence uniforme, car les assistants sont programmés depuis un point de configuration central et sont utilisés par les points de configuration locaux.

- **Configuration à l'aide de bases de données ou fichiers prédéfinis :**

  Les bases de données ou fichiers prédéfinis peuvent également faciliter la création ou l'extension de projets. Dans ce cas, les éléments à configurer sont décrits dans une base de données ou un fichier Excel (nom de fonction, type de fonction, paramètre de fonction, nom de variable, offset, etc. Ces éléments sont lus à l'aide de l'assistant, et les configurations correspondantes sont réalisées dans zenon.

D'autres informations concernant les assistants et leur création sont disponibles dans le manuel Assistants.

5.8 Réutiliser des projets

Les projets peuvent également être utilisés de différentes manières. Dans ce cas, les mécanismes présentés dans le chapitre consacré à la capacité de réutilisation jouent parfois un rôle significatif.

Sauvegarde de projet (à la page 38)

Enregistrer sous (à la page 39)

Gestion multi-projets (à la page 39)

5.8.1 Sauvegarde de projet

Les sauvegardes de projets permettent d'accepter des projets complets.

Par exemple, un projet peut être distribué à toutes les équipes sous forme de modèle. Ces modèles créent une sauvegarde de projet et personnalisent le projet.

> **Attention**
>
> Cette méthode n'est pas adaptée au transfert des modifications de projet, car toutes les configurations locales sont remplacées lors de leur acceptation.

Pour assurer la maintenance de projets créés de cette manière au niveau de toutes les équipes, nous vous recommandons d'utiliser la fonction d'exportation/importation de données XML (à la page 36).

Veuillez noter que les éléments existants portant le même nom sont remplacés lors de l'importation. Lors de la configuration d'un projet, les éléments configurés pouvant être remplacés doivent être clairement identifiés. Par exemple, pour l'identification nécessaire, un ajout correspondant au nom de l'élément ou à la procédure par l'intermédiaire des groupes d'équipements, reportez-vous à la section projet global (à la page 34).

5.8.2 Save as

Save as creates a copy of a project in the active working area under a new name and with a new GUID.

EXAMPLE

Individual projects for different machine types are to be created, based on a template project. To do this:

1. The project backup (à la page 38) of the template project is loaded in
2. Several differently-named projects are created in the active workspace using **Save as** (such as **Machine 1**, **Machine 2**, ...)

Each machine project has the same initial project situation. However, the projects no longer have any connection to a template project. Changes that are made in a template project are not automatically accepted into the copied projects. However changes to the template can be introduced into individualized projects by means of XML export/import (à la page 36).

Veuillez noter que les éléments existants portant le même nom sont remplacés lors de l'importation. Lors de la configuration d'un projet, les éléments configurés pouvant être remplacés doivent être clairement identifiés. Par exemple, pour l'identification nécessaire, un ajout correspondant au nom de l'élément ou à la procédure par l'intermédiaire des groupes d'équipements, reportez-vous à la section projet global (à la page 34).

5.8.3 Gestion multi-projets

Les projets peuvent être compilés dans un espace de travail de la gestion multi-projets de zenon, dans le cadre d'un projet d'intégration. Il est ainsi possible de modifier un projet de manière centrale, tandis que les configurations de projets locaux sont effectuées dans un autre projet.

EXEMPLE

Un point de configuration de projet central est utilisé pour mettre en œuvre la réutilisation d'éléments graphiques dans les projets locaux, afin de garantir une interface utilisateur uniforme. Pour cela, la configuration centrale crée un projet modèle qui constitue la base graphique utilisée par toutes les autres équipes de configuration de projet. Toutefois, si une modification est apportée à la conception de l'interface utilisateur dans la configuration centrale, celle-ci doit être mise en œuvre à tous les niveaux de configuration du projet. Cette opération peut être effectuée à l'aide de la fonction de gestion multi-projets : un projet contient les éléments devant être gérés de manière centralisée, tandis qu'un autre projet contient la configuration locale.

Procédure :

1. Un projet de modèle est initialement transmis à toutes les équipes de configuration.
2. Il est ensuite transféré vers l'espace de travail actif.
3. Localement, un deuxième projet vide, contenant les modifications spécifiques, est créé localement.
4. Deux projets sont maintenant présents dans l'espace de travail ; le projet de modèle et le projet personnalisé.
5. L'organisation multi-hiérarchique des projets est effectuée à l'étape suivante.
6. Le projet de modèle devient le projet d'intégration ; le projet individuel devient le sous-projet.

> **Information**
>
> Cette méthode n'est pas adaptée aux projets devant être exécutés sous Windows CE, car un seul projet peut être démarré à la fois sous CE.

6 Réseau intégré :

Grâce à zenon, le développement distribué sur le réseau et la redondance totale ne demandent plus un effort démesuré ; vous pouvez créer des réseaux en cochant la case concernée dans les propriétés du projet. Et voulà exactement la facilité avec laquelle vous pouvez créer des systèmes redondants, connectés en réseau de manière efficace, à partir de projets autonomes.

Si vous possédez un réseau TCP/IP Windows, zenon propose automatiquement des fonctions réseau accessibles d'un clic, ainsi qu'un modèle client/serveur ou un système à hiérarchies multiples comportant des sous-stations, des centres d'espace de travail et des centres. Celui-ci :

- crée également des projets avec d'autres utilisateur sur le réseau, au même moment
- garantit la protection de vos équipements grâce à la redondance circulaire totale de zenon.

- Créez des systèmes distribués sans la moindre difficulté.
- Accédez à distance aux postes de travail
- Suivez et contrôlez vos équipements via zenon Web Server.
- Pouvez consulter les données de procédé sur tous les postes en temps réel.
- Rendez les fonctionnalités d'un espace de travail (par exemple, l'acquittement d'alarmes) visibles dans tous les autres espaces de travail
- Assurez la journalisation et l'archivage centralisés des actions
- Utilisez instantanément les données de procédé pour les systèmes ERP, tels que SAP

zenon assurera automatiquement la synchronisation horaire requise sur tous les ordinateurs participants.

7 Conseils et astuces

Dans zenon, vous disposez de nombreux raccourcis clavier et méthodes rapides pour configurer votre projet. Nous vous présentons ici une sélection de raccourcis et méthodes.

Conseils et astuces pour :

- Éditeur (à la page 41)
- Runtime (à la page 44)
- La phase de test (à la page 51)
- Raccourcis-clavier (à la page 45)

7.1 zenon Éditeur :

Conseils relatifs à zenon Éditeur :

GÉNÉRAL

- Enregistrer les vues d'Éditeur : Les profils Éditeur vous permettent d'enregistrer vos divisions de fenêtres et vos attributions de modules personnelles.
- Accédez directement au dossier des fichiers d'Éditeur : Sélectionnez le projet, puis appuyez sur les touches **Ctrl+Alt+E**.
- Accédez directement au dossier des fichiers du Runtime : Sélectionnez le projet, puis appuyez sur les touches **Ctrl+Alt+R**.

SYNOPTIQUE/ÉDITEUR DE SYMBOLES

- Lier des éléments :
  Vous pouvez lier des fonctions, des variables, des polices, etc. à des **éléments dynamiques** en effectuant un glisser-déplacer.
- Ajouter une copie au même emplacement :
  Le raccourci-clavier **Ctrl+Maj+V** vous permet d'ajouter des éléments copiés au même emplacement que l'élément source.

Tracer un cercle :
Pour tracer un cercle, faites glisser le pointeur de l'élément **Ellipse/cercle** avec la souris, tout en maintenant la touche **Maj** enfoncée.

- Modifier une ligne de façon symétrique :
  Si vous maintenez la touche **Alt** enfoncée pendant que vous déplacez les coins extérieurs, la modification est apportée de manière symétrique.
- Redimensionnement :
  Vous pouvez redimensionner plusieurs éléments de synoptique simultanément : Pour cela :
  - Sélectionnez tous les éléments.
  - Convertissez-les en groupe d'éléments
  - Redimensionnez le groupe d'éléments à votre convenance.
  - Les éléments individuels sont redimensionnés dans la même proportion.
  - Scindez à nouveau le groupe d'éléments en éléments individuels.
- Modèles des synoptiques :
  Vous pouvez convertir tout synoptique de votre choix en modèle de synoptique.
- Manipuler les coordonnées X/Y d'un élément :

  Vous pouvez directement saisir les coordonnées X/Y d'un élément ou d'un coin d'un polygone dans une boîte de dialogue ou le déplacer à l'aide des touches fléchées.
  - Saisie directe : Double-cliquez sur la poignée de redimensionnement d'un élément ou d'un coin d'un polygone pour ouvrir une boîte de dialogue. Vous pouvez directement saisir les coordonnées X/Y de ce point.
  - Déplacement avec les touches fléchées : Placez le pointeur de la souris sur la poignée de redimensionnement d'un élément ou un coin d'un polygone. Vous pouvez maintenant déplacer directement les coordonnées X/Y de ce point, à l'aide des touches fléchées.
- Accéder à l'élément lié :
  Cliquez avec le bouton droit sur une propriété dans la fenêtre des propriétés ouvre un menu contextuel permettant d'accéder directement à l'élément lié.

CHANGEMENT DE LANGUE

- Intitulé des unités :
  Définissez le nom d'unité exact en tant qu'info-bulle. Dans l'environnement habituel, les utilisateurs réfléchissent normalement en unités nationales. Pour le dépannage, l'optimisation de l'internationalisation et les projets dotés d'une portée internationale, l'utilisation de cette info-bulle peut simplifier significativement leur travail.
  Rendez ces info-bulles traduisibles, et utilisez également des dénominations internationales pour le pays. Vous pouvez également traduire l'unité de mesure dans la langue correspondante.
- Unités de valeur en tant que caractère et non en tant que mot :
  Utilisation de l'affichage des unités pour l'affichage de valeurs. Une traduction n'est alors pas nécessaire.
  Exemple d'affichage de la température : Le terme degré(s) doit être traduit. Toutefois, le symbole **°** est identique dans toutes les langues et peut également être utilisé pour les degrés Celsius, Fahrenheit and Kelvin.
  Définissez le symbole correspondant dans les propriétés de l'unité de base.
- Conversion des valeurs :
  Ajoutez la conversion de valeur correspondante à chaque fonction de changement de langue. Dans le cas contraire, des erreurs d'interprétation peuvent se produire.

ÉDITEUR DE SYMBOLES

- Déverrouillage de la propriété :
  Vous pouvez déverrouiller une propriété d'un élément en effectuant un glisser-déplacer pour déplacer la propriété dans la fenêtre inférieure de l'éditeur de symboles. Dans ce cas, vous devez toujours cliquer sur le nom des propriétés, et non sur la valeur.
- Mode d'édition individuel de symboles :
  Le raccourci **Alt+clic** sur un élément permet d'activer l'élément situé dessous.

VARIABLES ET DRIVERS

- Substitution :
  Pour pouvoir appliquer de manière optimale la possibilité de substitution :
  - Utilisez des types de données de structure.
  - Assurez-vous de donner à vos variables un nom court et concis.
  - Tenez compte de la possibilité d'une substitution lorsque vous attribuez un nom aux variables. Ce nom doit être unique et facilement substituable.
  - Configurez d'abord un synoptique sous forme de modèle, que vous pourrez réutiliser autant de fois que nécessaire après l'avoir complété à l'aide d'une fonction.

- Créez automatiquement des variables en important :
  De nombreux drivers facilitent la création automatique de variables avec un adressage correct en autorisant l'importation depuis l'automate ou depuis un fichier. Pour cela, cliquez avec le bouton droit sur le driver, puis sélectionnez **Importer les variables à partir du driver**.

7.2 zenon Runtime

Conseils pour le Runtime de zenon :

GÉNÉRAL

- Affichage du nom du synoptique :
  Cliquez avec le bouton droit sur une zone vide de l'écran et maintenez le bouton droit enfoncé. Le nom du synoptique s'affiche.
- Liens de l'affichage :
  Cliquez avec le bouton droit sur un **élément dynamique** et maintenez le bouton droit enfoncé. Les fonctions ou les variables lié(e)s sont affichés.
- Erreur de communication :
  Affichage au moyen de symboles sur l'élément numérique :
  - Coin supérieur gauche rouge : pas de communication avec l'automate.
  - Coin supérieur gauche bleu : pas de communication avec le serveur.
- Écrire une valeur prescrite :
  Outre la valeur, vous pouvez modifier une commande.

  Seule la valeur d'une variable peut être modifiée dans le Runtime. Si la boîte de dialogue standard est utilisée, la fonction devant être exécutée peut également être modifiée à l'aide de la liste déroulante **Commande**, en *Désactiver valeur spontanée*, par exemple. Pour que la boîte de dialogue standard soit affichée, la propriété **Synoptique clavier** doit être désactivée dans la propriété **Éléments dynamiques**.
- Écrire une valeur prescrite directement, sans boîte de dialogue d'exécution :
  Pour écrire une valeur prescrite directement, dans les propriétés de l'**élément dynamique**, dans le groupe **Écrire valeur prescrite**, activez la propriété **Sans dialogue**.

  Ce paramètre peut également être défini avec la fonction **Ecrire/modifier valeur prescrite**, en activant l'option **Directement sur matériel**.

SERVEUR D'ARCHIVES

- Afficher le nom du fichier ARX dans le synoptique :
  Le nom du fichier ARX peut être affiché dans la liste d'archives, dans une colonne du synoptique révision d'archive. Pour cela, dans le fichier **zenon6.ini**, dans la section **[ARCHEDIT]**, définissez l'entrée **KURZBEZEICHNUNG=** sur *1*. La description courte est alors affichée dans la

liste. Elle correspond aux noms de fichiers ARX et fait également partie de toutes les archives d'agrégation.

UTILISATION AU CLAVIER

- Utilisation au clavier :
  Le Runtime peut également être intégralement utilisé à l'aide du clavier. Pour plus de détails, reportez-vous au chapitre Utilisation au clavier du manuel Runtime.
- Désactivation des raccourcis-clavier :
  Les raccourcis-clavier de Windows peuvent être désactivés. Pour plus de détails, reportez-vous au chapitre Désactivation des raccourcis-clavier du manuel Runtime.

7.3 Raccourcis-clavier

zenon permet d'accomplir de nombreuses actions à l'aide de raccourcis-clavier.

GÉNÉRAL

Commande	Raccourci-clavier
Ouvrir l'aide	**F1**
Ouvrir/fermer le mode plein écran	**Maj**+**F9**
Distante : Fermer le mode plein écran	**Ctrl**+**Alt**+**Maj**+F
Démarrer l'éditeur VSTA	**Alt**+**F10**
Démarrer VBA Éditeur	**Alt**+**F11**
Assistants Ouvrir la sélection	**Alt**+**F12**
Ouvre l'explorateur de fichier pour le projet actuel, en affichant le dossier SQL. Correspond à : *%ProgramData%\COPA-DATA\[SQL-Ordner]\[UID]\FILES*	**Ctrl**+**Alt**+**E**
Ouvre l'explorateur de fichiers en affichant les fichiers du projet actuel. Par exemple : *C:\Users\Public\Documents\zenon_Projects\[Project]\INI\[Rechner]\INI*	**Ctrl**+**Alt**+**D**
Ouvre l'explorateur de fichiers en affichant les	**Ctrl**+**Alt**+**R**

Commande	Raccourci-clavier
fichiers de Runtime du projet actuel. Par exemple : *C:\Users\Public\Documents\zenon_Projects\[Project]\INI*	
Démarrer le Runtime ; créer préalablement les fichiers Runtime modifiés.	**F5**
Créer les fichiers Runtime modifiés	**F7**

PROFILS EDITOR

Commande	Raccourci-clavier
Charger le profil 1 de l'Éditeur	**Maj**+**F1**
Charger le profil 2 de l'Éditeur	**Maj**+**F2**
Charger le profil 3 de l'Éditeur	**Maj**+**F3**
Charger le profil 4 de l'Éditeur	**Maj**+**F4**
Charger le profil 5 de l'Éditeur	**Maj**+**F5**
Charger le profil 6 de l'Éditeur	**Maj**+**F6**
Charger le profil 7 de l'Éditeur	**Maj**+**F7**
Charger le profil 8 de l'Éditeur	**Maj**+**F8**
Enregistrer la vue actuelle de l'Éditeur sous :	
Profil 1	**Ctrl**+**Maj**+**F1**
Profil 2	**Ctrl**+**Maj**+**F2**
Profil 3	**Ctrl**+**Maj**+**F3**
Profil 4	**Ctrl**+**Maj**+**F4**
Profil 5	**Ctrl**+**Maj**+**F5**
Profil 6	**Ctrl**+**Maj**+**F6**
Profil 7	**Ctrl**+**Maj**+**F7**
Profil 8	**Ctrl**+**Maj**+**F8**

ÉDITEUR GRAPHIQUE :

Note concernant les raccourcis : le signe plus (+) signifie que les touches sont enfoncées simultanément.
Par exemple :

Ctrl+A signifie : Maintenez enfoncée la touche **Ctrl** et appuyez ensuite sur **A**.

Ctrl++ signifie : Maintenez enfoncée la touche **Ctrl** et appuyez ensuite sur touche plus.

GÉNÉRAL

Commande	Raccourci-clavier
Fenêtre principale : faire défiler le contenu avec le pointeur Main	Maintenez la touche **Espace** enfoncée
Fermer le synoptique actuel	**Ctrl**+**F4**
Ouvrir les propriétés	**Alt**+**Entrée**

SÉLECTION

Commande	Raccourci-clavier
Sélectionner plusieurs objets	Maintenez la touche **Maj** ou **Ctrl** enfoncée.
Désélectionner un objet dans une sélection multiple	**Ctrl** + **clic de souris**
Sélection : modifier l'ordre de tri. Définir l'élément sur lequel tous les autres seront réalignés	Maintenez la touche **Maj** enfoncée pendant la sélection
Sélectionner les objets masqués	1. **Appuyez sur la touche Alt** 2. **Cliquez sur l'objet et déplacez-le**
Sélectionner tous les éléments d'un synoptique	**Ctrl**+**A**
Sélectionner l'élément suivant dans l'ordre de création des éléments	**Tabulation**
Sélectionner l'élément précédent dans l'ordre de création des éléments	**Maj** + **Tabulation**

POSITION

Commande	Raccourci-clavier
Déplacer l'objet sélectionné	**Touches fléchées**
Déplacer l'objet de 10 pixels à chaque pression sur une touche fléchée	**Maj**+**touche fléchée**
Déplacer l'objet horizontalement ou verticalement	Maintenez la touche **Maj** enfoncée pendant le déplacement
Centrer l'objet sélectionné dans la zone de travail	**H**

ACTIONS

Commande	Raccourci-clavier
Enregistrer les modifications	**Ctrl**+**S**
Coller l'élément depuis le presse-papiers	**Ctrl**+**V** **Maj** + **Insér.**
Insérer un élément du presse-papiers dans sa position originale ; l'original et la copie se superposent	**Ctrl**+**Maj**+**V**
Copie l'élément sélectionné.	**Ctrl**+**C** **Ctrl**+**Insér.**
Copier, plutôt que déplacer	Maintenez la touche **Ctrl** enfoncée pendant le déplacement .
Duplique l'élément sélectionné. Vous trouverez des informations plus détaillées à ce sujet au chapitre Duplication d'éléments.	**Ctrl**+**D**
Supprime l'élément sélectionné.	**Suppr**
Coupe l'élément sélectionné.	**Maj**+**Suppr** **Ctrl**+**X**
Ouvre la boîte de dialogue qui permet de	Ctrl+R

Commande	Raccourci-clavier
remplacer les liens dans l'élément sélectionné.	
Annuler les modifications	**Ctrl+Z** **Alt+Retour arrière**
Ajouter ou supprimer un nœud dans l'élément sélectionné. Ajouter : le pointeur de la souris devient un symbole moins (+). Supprimer : le pointeur de la souris devient un symbole moins (-). Fonctionne avec les polylignes, les polygones et les barres verticales.	**Ctrl+Maj**
Annuler le tracé des polylignes et polygones	**s**
Annuler le tracé des polylignes et polygones et supprimer la dernière section tracée	**Échap**
Déplacer les éléments sélectionnés d'un niveau vers le haut	**+**
Déplacer les éléments sélectionnés d'un niveau vers le bas	**-**
Déplacer les éléments sélectionnés au premier plan	**Ctrl++**
Déplacer les éléments sélectionnés vers l'arrière-plan	**Ctrl+-**

REDIMENSIONNEMENT

Commande	Raccourci-clavier
Modifier la taille	Déplacez le pointeur de la souris sur le coin ; le pointeur devient une flèche. Vous pouvez ensuite positionner avec une précision du pixel avec les **touches fléchées** ou par étapes de 10 pixel en maintenant **Maj** enfoncée. **Remarque :** si un angle différent de *0* a été défini pour

Commande	Raccourci-clavier
	un élément via la propriété **Angle de rotation [°]**, il ne peut pas être redimensionné à l'aide des touches fléchées.
Redimensionner l'objet autour du centre	Maintenez la touche **Alt** enfoncée pendant le redimensionnement.
Redimensionnement proportionnel	Maintenez la touche **Maj** enfoncée pendant le redimensionnement.

ZOOM

Réduire la vue	Ctrl + Maj + -
Agrandir la vue	**Ctrl** + **Maj** + **+**
Régler la vue à 100 %	**Ctrl** + **Alt** + **0**
Adapter la vue à l'espace disponible dans l'Éditeur et afficher en respectant les proportions	**Ctrl** + **0**

ZOOM À L'AIDE DE LA SOURIS

Réduire la vue	Ctrl + roulette de la souris vers le bas
Agrandir la vue	**Ctrl** + roulette de la souris vers le haut

VUE DÉTAILLÉE

Commande	Raccourci-clavier
Crée un nouvel élément pour le module concerné	**Ins**
Modifie la colonne sélectionnée	**F2**
Copie un élément sélectionné de la liste	**Ctrl**+**C**
Insère un élément préalablement copié de la liste	**Ctrl**+**V**

Commande	Raccourci-clavier
Supprime un élément sélectionné de la liste	**Suppr**
Fait défiler la liste de plusieurs éléments vers le haut	**Page haut**
Fait défiler la liste de plusieurs éléments vers le bas	**Page bas**
Permet de parcourir la liste	**Touche fléchée**

RUNTIME

Commande	Raccourci-clavier
Afficher les différentes fenêtres Windows ouvertes.	**Alt**+**Tab**
Arrêter le Runtime	**Alt**+**F4**

MENU CONTEXTUEL ÉLÉMENTS

Créer un groupe d'éléments	Créer un groupe d'éléments Le raccourci-clavier Ctrl + G peut aussi être utilisé à cette fin.
Dissocier	Dissocie un groupe d'éléments sous forme d'éléments de synoptique individuels. Le raccourci-clavier **Ctrl + Maj G** peut aussi être utilisé à cette fin.
Mode plein écran	Fermer le mode plein écran : **Maj** + **F9**

7.4 Test de projets

Conseils relatifs aux tests de projets :

- Nom de la variable dans le synoptique :
  Pour afficher un nom de variable dans le synoptique, configurez un élément combiné sans état, avec le texte par défaut : **%n**.