Manuel de zenon

Développez efficacement vos projets avec zenon

v. 7.20

©2015 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 <CD_COMPANYNAME<. 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 18

5. Réutilisation d'éléments 19

5.1 Remplacement de variables et de fonctions 20

5.1.1 Conventions d'affectation de noms 20

5.1.2 Possibilités de remplacement 21

5.2 Symboles 26

5.2.1 Propriétés d'accès libre 26

5.2.2 Remplacement par des symboles 29

5.2.3 Bibliothèque générale de symboles 30

5.3 Types de données structure 32

5.4 Matrices de réaction 35

5.5 Projet global 35

5.6 XML 37

5.7 Assistants 38

5.8 Réutiliser des projets 39

5.8.1 Sauvegarde de projet 39

5.8.2 Enregistrer sous 40

5.8.3 Gestion multi-projets 40

6. Réseau intégré 41

7. Conseils et astuces 42

7.1 zenon Editor 42

7.2 Runtime de zenon 44

7.3 Raccourcis-clavier 46

7.4 Test de projets 52

1. Bienvenue dans l'aide de COPA-DATA

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 (mailto:documentation@copadata.com).

ASSISTANCE PROJET

Si vous vous rendez compte que vous avez besoin de licences ou de modules supplémentaires, veuillezcontacter l'équipe commerciale par e-mail : support@copadata.com (mailto: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 (mailto: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. Nous vous recommandons de lire ce chapitre, même si vous avez déjà acquis de l'expérience avec d'autres systèmes de contrôle de procédés. Vous apprendrez ici des notions fondamentales de zenon et de son fonctionnement, et découvrirez comment et pourquoi zenon vous aide à travailler plus facilement et rapidement.

UN APERÇU DE ZENON

zenon est constitué d'Editor et du Runtime.
Les projets sont créés dans Editor. L'utilisation et la surveillance s'effectuent dans le Runtime. Editor 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 :

- Simplicité d'utilisation (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 que vous pouvez paramétrer rapidement, au lieu de passer des heures à programmer vos équipements. Une fois définis, les objets peuvent être utilisés dans différents projets. Vous pouvez utiliser les fonctions d'héritage de propriétés, et modifier les propriétés en un clin d'œil.
- Excellente capacité de réutilisation (à la page 19) des objets et éléments 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 41) : 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 également constituer un réseau de projets autonomes individuels pour développer des 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 (mailto: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. Vous pouvez effectuer votre choix au début des listes de propriétés. De cette manière, vous pouvez rapidement accéder aux propriétés dont vous avez le plus fréquemment besoin.

UTILISATION GÉNÉRALE DU CLAVIER DANS EDITOR

Editor peut être facilement contrôlé grâce aux symboles et aux menus actionnables avec la souris ; vous pouvez également utiliser le clavier de façon universelle. Les raccourcis-clavier de Windows fonctionnent également dans zenon. Par exemple :

- `Ctrl+C` et `Ctrl+V` pour **`copier`** et **`coller`**.
- `Insér.` crée de nouveaux objets.
- `Suppr.` supprime les objets sélectionnés.
- Le **`pointeur de la souris`** permet de se déplacer parmi les vues.
- `Échap` termine l'action en cours.
- `Retour arrière` remonte d'un niveau dans l'arborescence des répertoires.
- `Ctrl+Z` annule les dernières actions, à l'exception des opérations sur bases de données.

Pour plus d'informations à ce sujet, reportez-vous au chapitre Raccourcis-clavier.

GLISSER ET DÉPOSER

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

- Modifier l'ordre des entrées de menus
- Lier des polices à des éléments dynamiques tels qu'une réglette universelle, une horloge, un élément combiné etc.
- Modifier l'ordre des éléments structurels pour les données de type structure.
- Modifier l'ordre d'éléments 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.

INFO-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 liée (par exemple, la 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 offre une vaste gamme de fonctions pouvant faciliter le dessin d'équipements, et vous offre également de nombreuses possibilités : du zoom sans incrément et redimensionnement en un clic, verrouillage des coins d'éléments ou distribution automatique d'éléments suivant des règles. Vous trouvez plus de détails au chapitre Images.

ACCÉDER À L'ÉLÉMENT LIÉ

Pour tous les éléments graphiques, fonctions, variables et autres éléments, vous pouvez accéder directement à l'élément lié. Par exemple, vous pouvez cliquer sur un élément d'affichage pour afficher directement la variable liée ; si vous cliquez à nouveau dessus, vous pouvez accéder directement aux types de données utilisées et même afficher l'unité employée. Le cas échéant, vous pouvez facilement annuler toutes ces étapes et revenir au point de départ en l'espace de quelques clics.

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.

ADAPTEZ 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 Active X :
  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 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 Editor. 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.

Informations

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. Ce type d'objet de driver définit le domaine du PLC concerné par l'adressage.

TYPE D'OBJET DRIVER

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

Le type d'objet de driver détermine :

- Le domaine de l'automate
- La granularité du driver dans ce domaine
- Il définit les types de données pouvant être créés dans la zone de mémoire.

Conseil : tous les types de données ne peuvent pas être créés dans un domaine. Par conséquent, suivez toujours les instructions fournies lorsque vous créez une variable :

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

ACCESSOIREMENT : GRANULARITÉ

La granularité joue un rôle particulièrement important dans la programmation d'automates à adressage numérique, tels que les Siemens S7. Tous les automates ne possèdent pas la même résolution, et tous les domaines d'un automate ne possèdent pas non plus la même résolution.

Par exemple, le domaine de blocs de données d'un automate Siemens S5 est orienté mot, tandis que son domaine de marqueurs est orienté octet. Cela signifie que la plus petite unité pouvant être adressée est soit un octet, soit un mot. Maintenant, si vous voulez écrier un bit dans l'un de ces domaines, 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 l'automate.

Les mêmes informations sont requises lors de l'adressage des domaines, appelés "offset". Vous commencez par zéro et continuez à compter en incréments par octets/mots. Si vous utilisez l'adressage automatique (à la page 18) dans zenon, la granularité est automatiquement prise en compte. La granularité ne dépend donc 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 développement. Elles ne communiquent pas avec l'automate, mais lisent une zone de la mémoire interne du driver contenant principalement des informations statistiques. Toutefois, vous pouvez également contrôler des fonctions spéciales à l'aide des variables, à l'image des numéros de téléphone de connexions modem ou des commandes de numérotation/de fin de connexion.

Vous trouverez plus d'informations au chapitre Variables des drivers.

Un fichier dBase contenant les plus importantes variables de drives est inclus sur le CD d'installation de zenon. Vous pouvez l'utiliser pour importer ces variables simplement et efficacement dans tous les drivers, 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 structure
- Éléments de structure

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 ex. BOOL, INT, USINT, UDINT, STRING, WSTRING, etc.). Ces données aident à définir la taille d'une zone ; par exemple, grâce à
BOOL : 1 bit signés
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. Celles-ci possèdent la même description que les propriétés des variables :

- Identification
- Unité
- Plage de valeurs
- Valeurs limites

Pourquoi cette "double comptabilisation" ?

La raison est simple :
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 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, nous envisageons ici une orientation objet : les données de type objet transmettent leurs propriétés à la variable. La différence ?

Les propriétés liées/dérivées de la variable peuvent être séparées/écrasées. Cela fonctionne pour chaque propriété individuellement, mais également pour toutes les propriétés à la fois. Cela signifie que vous pouvez reprendre l'unité provenant du type de données de manière centrale, mais également remplacer l'identification ou l'adresse directement au niveau de la variable. Voici comment cela marche :

- 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 d'autres fonctions via le menu contextuel :

- Lier la propriété au type de données :

  Si vous avez interrompu la liaison, vous pouvez la rétablir.

- Dissocier la propriété :

  Vous pouvez dissocier une propriété unique sans devoir modifier la valeur. Si vous modifiez ultérieurement la valeur du type de données, la variable ne sera pas modifiée par cette action. Cela vous permet de définir exactement l'élément auquel doit être appliqué le concept d'héritage.

Une option utile : vous pouvez également lier ou dissocier toutes les propriétés du type de données en un clic. Cela vous permet de basculer très rapidement vers et depuis l'état original (héritage de toutes les propriétés). En quoi est-ce que cela vous est bénéfique, en tant qu'utilisateur ?

Flexibilité accrue : si vous utilisez de nombreuses variables possédant la même limite, par ex. une alarme pour la valeur 1, vous pouvez simplement définir la limite au niveau du type de données. La limite réelle et les autres paramètres facultatifs peuvent alors être définis séparément pour chaque variable. Vous n'avez plus besoin de créer et d'actualiser la limite pour chacune des variables, ce qui augmentera très significativement l'efficacité du développement. Le type de données par défaut BOOL possède une limite prédéfinie à 0 et 1 !

Si vous n'utilisez pas une limite pour chaque variable BOOL, vous n'avez pas besoin de veiller à désactiver manuellement la limite pour chaque variable. Au lieu de cela, vous pouvez utiliser une fonction extrêmement pratique de zenon et créer votre propre type de données BOOL. Suivez les instructions qui vous sont fournies :

- Laissez le type de données standard tel quel.
- Cliquez sur le bouton `Nouveau type de données`.
- Vous verrez alors s'afficher la liste contenant les types de données déjà existants.
- Choisissez `BOOL standard` :
- Spécifiez un nom, par ex. `MyBool`.

- Toutes les propriétés du nouveau type de données sont automatiquement reprises par l'ancien type.
- Supprimez les limites dans `MyBool`.

Utilisez ce nouveau type de données `MyBool` pour créer toutes les variables BOOL qui ne doivent pas comporter de limites ; cela supprime toutes les limites inutiles. Les autres limites, que vous souhaitez conserver, ne seront pas affectées.

Vous pouvez évidemment créer d'autres types de données BOOL personnalisés, et autant que vous le souhaitez. Toutes les variables utilisant des paramètres identiques pour la plupart de leurs propriétés reçoivent un type de données spécifique. Si l'une des propriétés change, vous pouvez effectuer les modifications de manière centrale, et toutes les variables dérivées seront automatiquement mises à jour (exceptées celles dont vous avez dissocié les liaisons).

Types de données de structure - Éléments de structure - Structure

TYPES DE DONNÉES STRUCTURE

Le type de données de structure désigne un groupe de types de données dont l'ordre et l'organisation sont définis avec exactitude. Par exemple, une structure Moteur peut inclure les éléments Régime actuel (tr/min) et Alimentation.

Une structure peut également être imbriquée, et différents niveaux hiérarchiques sont possibles.

Avec les types de données Structure, vous pouvez développer votre ensemble de variables exactement tel qu'il existe dans l'automate ou dans la réalité.

ÉLÉMENT DE STRUCTURE

Un type de données de structure est en réalité une simple "coque" qui porte le nom de la structure, mais ne comporte aucune propriété propre.

Les propriétés seront uniquement attribuées après que les éléments de structure aient été ajoutés au type de données de structure. On distingue les types suivants :

- Type de données de structure liées :

  Référence à un type de données existant. Toutes les propriétés sont reprises, à l'exception du nom.

  Avantage : si l'élément structure fournit de nombreuses propriétés d'un type de données existant, celles-ci peuvent être réutilisées. Vous pouvez évidemment toujours modifier chacune des propriétés des variables, ou les dissocier du type de données.
- Type de données de structure embarqué :

  Autorise la configuration de paramètres d'adressage individuels. Ce nouveau type de données est uniquement valide dans cette structure.

Informations

Orientation objet avantageuse : si la structure est modifiée d'une manière ou d'une autre, il suffit de modifier le type de données de structure afin qu'il s'adapte automatiquement à toutes les variables dans cette structure.

Toutes les caractéristiques peuvent être modifiées :

- Les propriétés simples, telles que l'unité du type de données
- Les propriétés complexes, telles que l'ajout ou la suppression de limites
- La modification de l'ensemble de la structure, par ex. en modifiant l'ordre des éléments ou en ajoutant/en supprimant des éléments de structure uniques dans le type de données de structure.

STRUCTURE

Si plusieurs machines sont concernées, l'utilisation d'une structure est recommandée.

Par exemple : 100 pompes au lieu d'une pompe.
Il vous suffit de changer les variables de structure en structure. D'un simple clic, 100 variables deviennent 100 variables de structure.

Suivant notre exemple, nous devons créer 12 variables pour chacune de nos pompes, ce qui produit un total de 1200 variables. La méthode du paramétrage orienté objet autorise cela en quelques clics seulement. Par ailleurs, elle offre l'avantage de fournir des variables prédéfinies. Chacune des variables inclut déjà les propriétés dont elle a besoin, telles que l'unité, le calcul de valeur, les alarmes, les entrées de liste d'événements etc.

Informations

Les types de données de structure sont également adaptés à la réutilisation (à la page 32) de variables.

Exemple d'une pompe :

SITUATION INITIALE

Une pompe est constituée de deux moteurs. Chacun de ces moteurs possède des variables telles que :

- `Régime actuel (tr/min)`

- `Consommation électrique`
- `Affichage des résultats`
- etc.

Chaque moteur comporte une réglette avec les variables suivantes :

- `Régime nominal (tr/min)`
- `P-share`
- `I-share`

Construisez cette constellation comme indiqué ici :

1. Créez une structure **réglette** à partir d'un type de données pour :
   - `Régime nominal (tr/min)`
   - `P-share`
   - `I-share`
2. Créez une structure **moteur** comportant ces types de données :
   - `Régime actuel (tr/min)`
   - `Consommation électrique`
   - `Affichage des résultats`
   - etc.
3. Intégrez la **réglette** à la structure **moteur**.
4. Enfin, créez la structure de la pompe et intégrez le moteur à celle-ci.

   Puisque la pompe comporte deux moteurs, il vous suffit d'ajouter deux fois le type de données moteur à votre structure **pompe**.
5. Créez une variable reposant sur ce type de données de structure :
   - Créer une nouvelle variable
   - Choisissez le type de données de structure **pompe**

   Les éléments individuels de cette variable de structure sont appelés éléments de structure (à la page 16). Vous pouvez utiliser chacun de ces éléments à tout endroit dans zenon – sur les synoptiques, comme alarmes, dans les archives, dans les recettes, etc.

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

Exemples pratiques :

- Adresse
- Import / Export

ADRESSAGE :

Grâce aux commandes numériques, vous pouvez décider si l'adressage doit être automatique ou semi-automatique. Toutes 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 pour les éléments de structure, sur la base de la position des éléments de structure, pour une structure comme pour un ensemble de structures. 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 :

Ici, les adresses de départ uniques attribuées comportent déjà un type de données. Les autres adresses sont ensuite déterminées en fonction de ces adresses. Vous pouvez évidemment les modifier ultérieurement. Vous trouverez plus de détails 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.

IMPORTATION ET EXPORTATION AU FORMAT 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 26)	Synoptiques
Types de données structure (à la page 32)	Variables
Matrices de réaction (à la page 35)	Variables
Projet global (à la page 35)	Éléments centraux d'un projet
XML (à la page 37)	Importation et exportation de composants de projets
Assistants (à la page 38)	Importation ou personnalisation de synoptiques et d'éléments
Réutiliser des projets (à la page 39)	Projets

Informations

Pour réutiliser efficacement des éléments, lors de la création de variables, de fonctions et de synoptiques, assurez-vous de les nommer (à la page 20) 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 Conventions d'affectation de noms

Pour autoriser le remplacement sûr des variables et autres éléments, l'affectation de noms doit être systématique et standardisée, dans la mesure du possible. Ceci facilite la réutilisation des variables et autres éléments, mais également les opérations de maintenance et de développement inverse.

Différents systèmes sont disponibles pour faciliter l'affectation de noms systématique.

PAR EXEMPLE : INDUSTRIE ÉLECTRIQUE

Allemagne

- KKS (Kraftwerk-Kennzeichen-System – Système de classification des centrales électriques) ; pour plus de détails (en allemand), consultez la page http://de.wikipedia.org/wiki/Kraftwerk-Kennzeichensystem (http://de.wikipedia.org/wiki/Kraftwerk-Kennzeichensystem).
- La norme DIN 6779 (Kennzeichnungssystematik für technische Produkte und technische Produktdokumentation – Système de classification des produits techniques et documentations de produits techniques) ; pour plus de détails (en allemand), consultez la page http://de.wikipedia.org/wiki/DIN_6779 (http://de.wikipedia.org/wiki/DIN_6779).
- AKZ (Anlagenkennzeichnungssystem – Système de classification d'équipements) ; pour plus de détails (en allemand), consultez la page http://de.wikipedia.org/wiki/Anlagenkennzeichnungssystem (http://de.wikipedia.org/wiki/Anlagenkennzeichnungssystem).

International

- KKS (Système de classification des centrales électriques) ; pour plus de détails, consultez la page (en allemand) http://de.wikipedia.org/wiki/Kraftwerk-Kennzeichensystem (http://en.wikipedia.org/wiki/KKS_Power_Plant_Classification_System).

Ces standards existent dans tous les secteurs de l'industrie. Nous vous recommandons d'utiliser les conventions d'affectation de noms correspondantes.

EXEMPLE DANS SQL

Dans un projet de centrale électrique, les variables doivent être nommées conformément au système KKS. Une variable correspondante avec l'intitulé `C01_MDY10-QA001_QA07` indique :

- Équipement éolien `C01` (rang C, no. 1)
- Contrôleur de turbine éolienne `MDY10`, référence de pièce `QA001`
- Dispositif de protection de puissance `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.

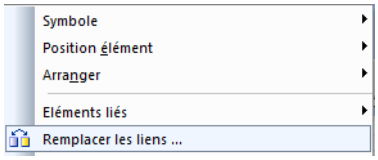
Remplacement des liens d'un synoptique

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

- Une convention d'affectation de noms claire pour les variables (à la page 20).

Celle-ci permet de copier et 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

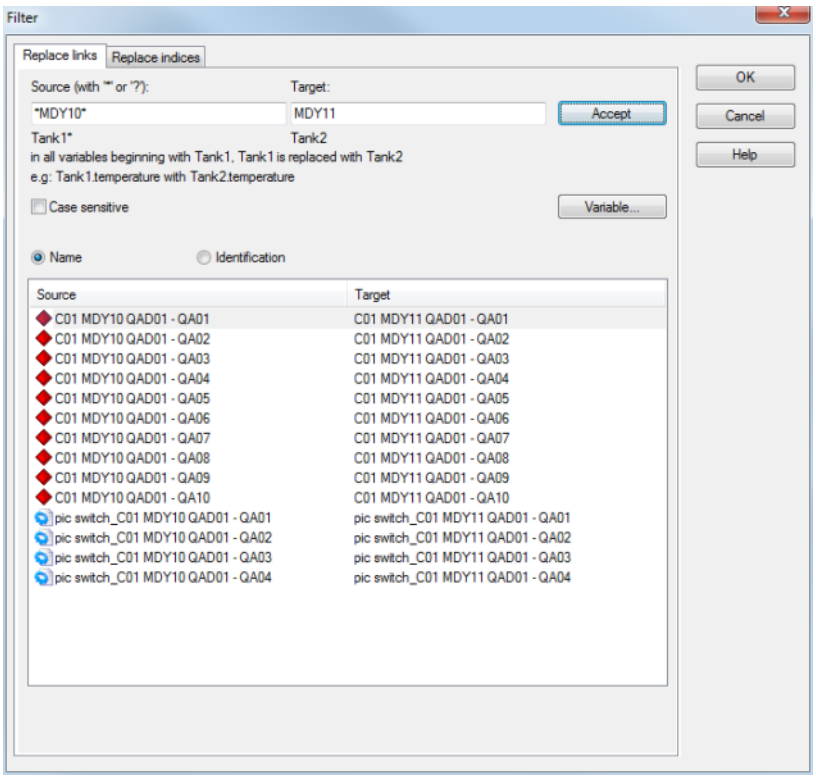
Dans notre exemple, le projet comporte plusieurs variables issues de différents éléments de l'équipement. Les éléments suivants sont présents dans le synoptique de procédé :

- 10 variables provenant du contrôleur de turbine éolienne **MDY10**, liées aux éléments dynamiques
- Des boutons d'appel de synoptique sont présents à différents emplacements dans **MDY10**

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

1. Les éléments dynamiques auxquels les variables et fonctions sont liées 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**

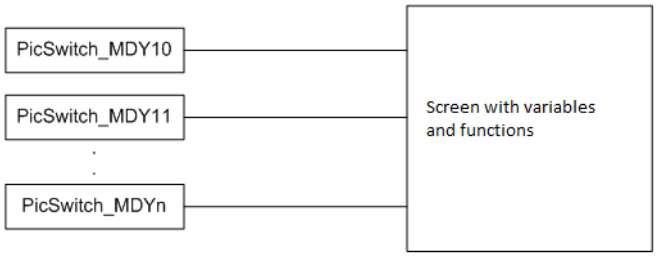
Le synoptique peut maintenant être utilisé pour le nouvel automate.

Remplacement 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, 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. Cette fonction est disponible lors de la création d'une boîte de dialogue de remplacement (à la page 22).

3. Chaque fonction contient un procédé de remplacement propre.



Une variante de ce procédé permet de n'utiliser qu'un appel de synoptique : le remplacement d'indices (à la page 24) dans les tableaux.

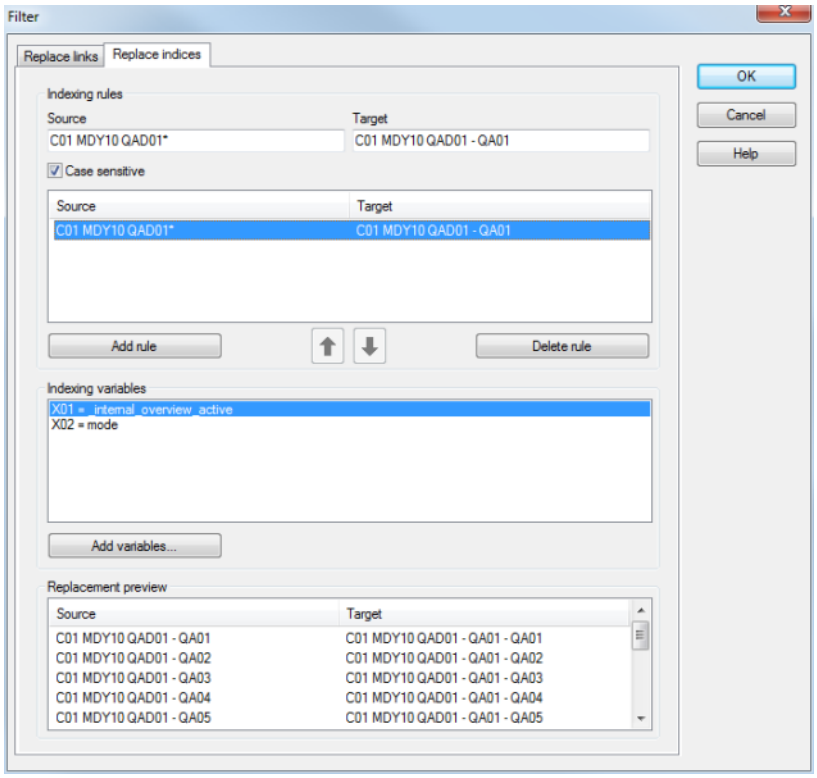
Remplacement d'indices

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 à la procédure de 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*
- Cible : C01 MDY{X01} QA001

Effet :

- Lors de l'appel de synoptique, la fonction est informée qu'elle doit utiliser la valeur de variable de l'`indice du contrôleur de turbine éolienne X01` dans les noms de variables.

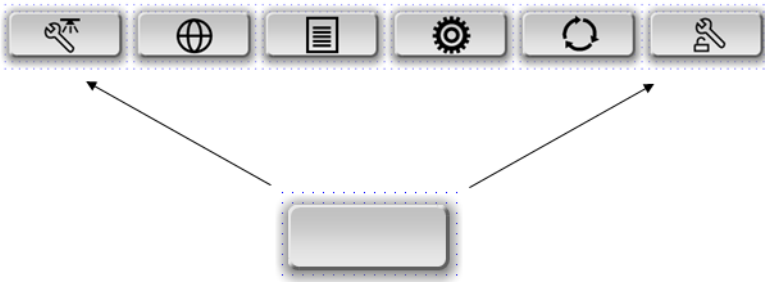
  Exemple : si cette variable comporte la valeur 12, le synoptique de procédé est affiché avec toutes les variables de contrôleur de turbine éolienne `MDY12` lors de l'appel de synoptique.

5.2 Symboles

Les symboles offrent une grande capacité de réutilisation. Ils peuvent être créés sous des formes très complexes, et autorisent l'héritage. Les symboles peuvent être incorporés ou liés. En cas de liaison, la modification à un endroit suffit pour mettre à jour tous les synoptiques utilisant ce symbole. Les symboles peuvent également être liés à d'autres symboles.

EXEMPLE D'UNE BARRE D'OUTILS

Un symbole (rectangle vide) est lié en tant que base de tous les boutons.



Les propriétés librement accessibles (à la page 26) permettent de créer chaque bouton avec différents éléments graphiques. Si la forme ou la couleur des boutons doit être modifiée, le symbole de bouton vide doit être modifié. Tous les boutons acceptent automatiquement la forme et/ou la couleur nouvelles.

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.

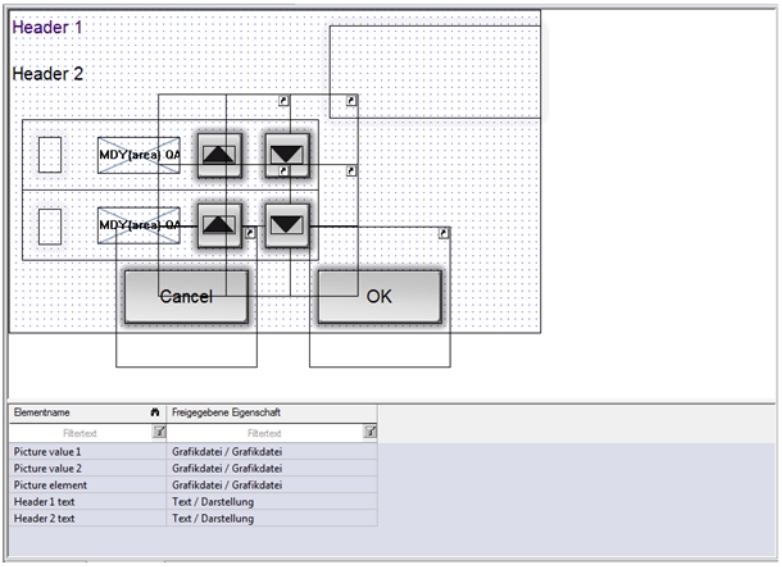
Copier et coller :

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.

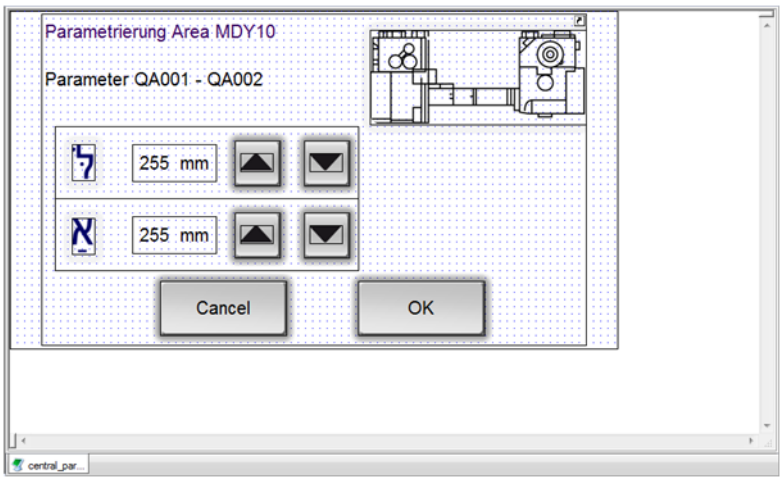
Toutefois : L'héritage des propriétés n'est pas possible ici. Si l'opérateur configurant le projet souhaite ultérieurement effectuer des modifications de manière centralisée (par rapport à la structure de base de la boîte de dialogue de configuration de paramètres), ils doivent d'abord faire glisser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

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

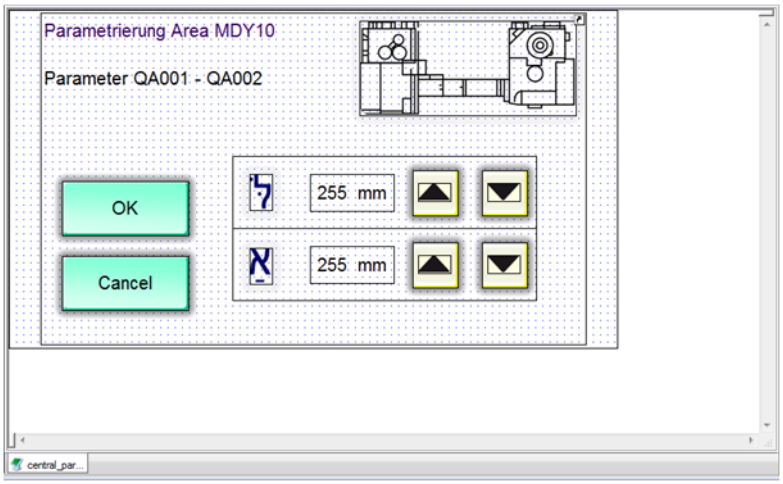
1. La boîte de dialogue de configuration des paramètres est créée avec un symbole.
2. Les éléments devant être adaptés à différents points de paramètres sont dissociés du concept d'héritage par le biais des propriétés non héritées.
3. Le symbole est lié, positionné et adapté aux synoptiques correspondants.



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 29) 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 20) 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

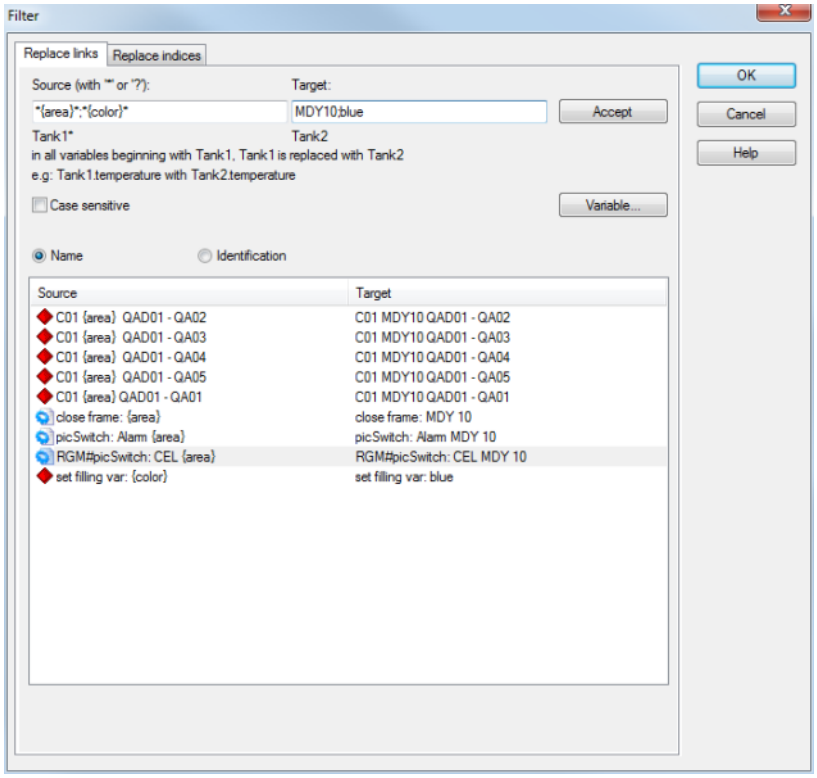
Selon l'approche choisie, différentes boîtes de dialogue sont affichées pour le remplacement des variables et fonctions.

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

La gestion des symboles s'effectue soit dans la bibliothèque de symboles du projet, soit dans le projet global ou dans la bibliothèque générale de symboles. La principale différence est la suivante :

Bibliothèque de symboles	Propriété
Bibliothèque générale de symboles	Les symboles sont disponibles dans tous les projets. Le nœud **Bibliothèque de symboles globale** se trouve dans le gestionnaire de projets, sous les projets actuellement chargés. Intitulé lors de la création d'un lien sur le synoptique : **[groupe de symboles]/[nom du symbole]** Règles : ▸ La boîte de dialogue sélection de variables affiche tous les projets de l'espace de travail pour la création de liens. ▸ Ces symboles sont enregistrés dans le dossier de programme de zenon, et sont uniquement mis à jour lors du démarrage d'Editor. Ces symboles ne sont pas enregistrés durant la sauvegarde du projet. ▸ Les verrouillages et les alias de la fonction CAL ne peuvent pas être configurés. Avertissement : Si les symboles contenant des verrouillages ou des alias sont ajoutés, ces paramètres sont supprimés.
Bibliothèque de symboles du projet global	Les symboles sont disponibles dans tous les projets de l'espace de travail. Intitulé lors de la création d'un lien sur le synoptique : **(g)[nom du symbole]** Règles : ▸ Les symboles copiés dans la bibliothèque de symboles du projet global conservent leurs liens de variables sans modification. ▸ La boîte de dialogue sélection de variables affiche tous les projets de l'espace de travail pour la création de liens. Les variables liées sont placées devant le nom du projet correspondant. ▸ Le nom de la **Variable** peut être modifié dans la fenêtre des propriétés. Ainsi, par exemple, le préfixe peut être supprimé avec l'origine du projet. ▸ Les symboles sont également sauvegardés lors de la sauvegarde d'un projet. ▸ Les verrouillages et les alias de la fonction CAL ne peuvent pas être configurés. Avertissement : Si les symboles contenant des verrouillages ou des alias sont ajoutés, ces paramètres sont supprimés.
Bibliothèque de symboles du projet	Les symboles sont uniquement disponibles dans le projet actuel. Intitulé lors de la création d'un lien sur le synoptique : **[Nom du symbole]** Règles : ▸ Les symboles sont enregistrés dans le dossier du projet. La bibliothèque de symboles du projet se trouve dans le projet actuel, dans le nœud **Synoptiques**, et est sauvegardée lors de la sauvegarde du projet. ▸ Les verrouillages et les alias de la fonction CAL peuvent être configurés. Ces propriétés sont également conservées lors de l'ajout de symboles.

Les symboles peuvent être copiés par glisser&déplacer.

- De la **bibliothèque générale de symboles** vers la **bibliothèque de symboles du projet global**
- De la **bibliothèque de symboles du projet global** vers la **bibliothèque de symboles du projet**, ou inversement

La copie directe de la **bibliothèque de symboles** vers la **bibliothèque de symboles du projet** ou la copie du projet ou du projet global vers la **bibliothèque générale de symboles n'est pas autorisée**.

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 29).

Il n'est pas toujours nécessaire de distribuer la collection entière de symboles. Une procédure d'exportation/importation de données XML (à la page 37) 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 Types de données structure

Les variables peuvent être réutilisées avec des types de données de structure. Des structures complexes peuvent ainsi être créées en utilisant les avantages de la fonction d'héritage.

EXEMPLE

Le type de données de structure Moteur comprend trois éléments :

- Etat
- Vitesse

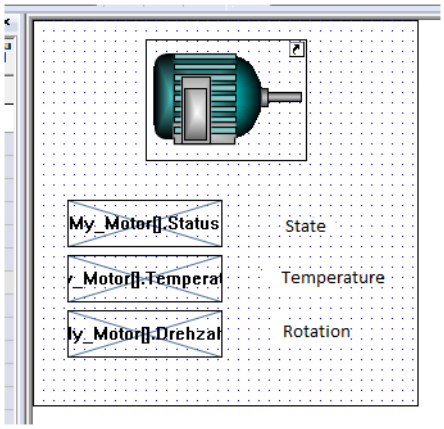
- Température

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

Les éléments diffèrent selon le type d'héritage :

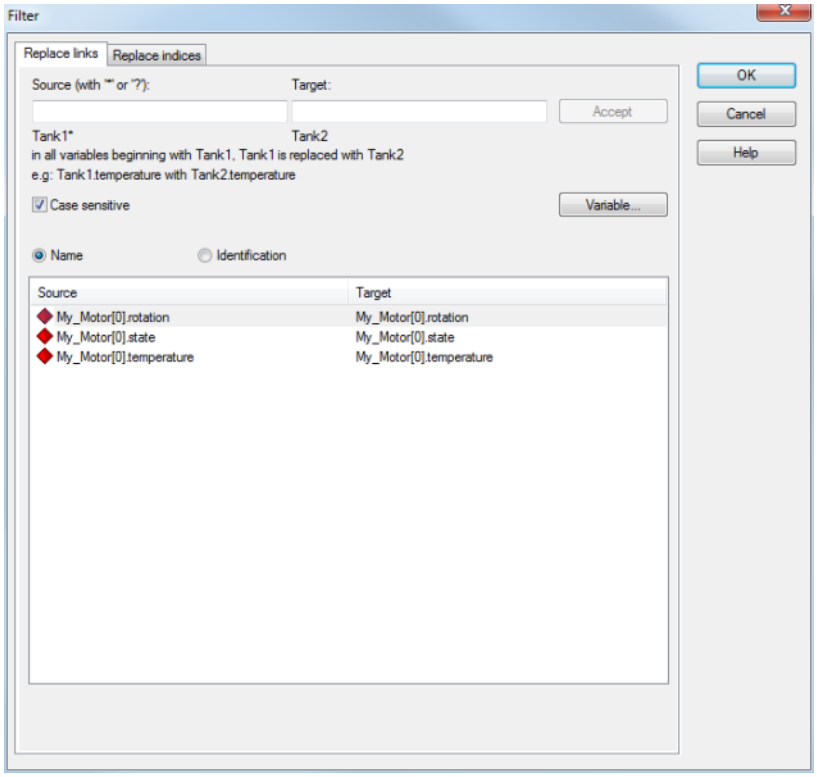
- les données **`État`** et **`Vitesse`** proviennent du type de données `INT`, mais leur héritage a été interrompu par le procédé d'incorporation.
  Ceci signifie : Que les mises à jour ultérieures du type de données `INT` n'a aucune incidence sur ces deux éléments.
- Les données "**`Température`**" proviennent également du type de données `INT` ; l'héritage est préservé.
  Ceci signifie : Que les modifications ultérieures apportées au type de données `INT` ont également une incidence sur cet élément de structure.

Si, par exemple, 20 moteurs sont maintenant configurés, des variables reposant sur le type de données "Moteur" généré automatiquement peuvent être créées. Ceci peut, par exemple, se produire lors de la création d'un tableau avec ce type de données. L'affichage graphique est déterminé par un symbole, dans cet exemple.



Lorsque ce symbole est lié à dans un synoptique de procédé, les variables de renseignement du symbole sont remplacées par un type de données de structure créé automatiquement. Il est important que la convention d'affectation de noms (à la page 20) utilisée soit bien réfléchie dès le début.

Si un synoptique de procédé est créé pour fournir une vue d'ensemble de tous les moteurs, le symbole inséré peut être facilement dupliqué par une manipulation copier-coller. La variable cible doit encore être remplacée par le numéro de séquence correspondant à chaque symbole :



Si les valeurs limite de la donnée Vitesse doivent être ajoutées ultérieurement, vous bénéficiez entre-temps de la relation d'héritage existante entre le type de données créé automatiquement et les variables créées résultantes. Pour la modification planifiée, une nouvelle valeur limite est créée pour l'élément de structure **`Vitesse`**, du type de données `Moteur`. Cette modification apportée à une position affecte la totalité des 20 variables de moteur créées.

L'opérateur configurant le projet est entièrement libre de modifier cette relation d'héritage. Une valeur limite pour la donnée **`Vitesse`** peut être modifiée pour une des 20 variables de moteur créées.

Remarque : les propriétés individuelles ou toutes les propriétés d'une variable sont déconnectées du type de données.

Informations

Vous pouvez lire des conseils relatifs à l'utilisation des types de données de structure dans la section Types de données de structure – Éléments de structure – Tableau de structure (à la page 16) et dans le manuel consacré aux variables de la section Types de données de structure.

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

Un projet global permet de définir des standards pour certains projets. Ceci permet de définir certains modules ou éléments pour des emplacements standard, 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.

- Modèles d'équipement :

  cette fonctionnalité autorise la configuration centralisée de projets, car elle autorise la division d'un projet en deux "usines". Dans ce cas, toutefois, vous devez clairement définir les éléments de configuration provenant du point de configuration central et ceux provenant du point de configuration local.

- Utilisateur :

  si un standard doit s'appliquer à certains utilisateurs globaux, ceux-ci peuvent être définis par le biais d'un projet global.

- Fichiers :

  les éléments graphiques peuvent être régis par un standard, par exemple, qui concerne les projets individuels. Ces éléments peuvent être accessibles globalement.

- Cadres :

  Ces éléments définissent la disposition du synoptique de procédé et la taille des projets à développer. Par exemple, il peut être défini clairement dans certaines directives que les nouveaux synoptiques peuvent uniquement être créés avec des cadres provenant d'un projet global. Ceci permet de créer une apparence graphique uniforme et globale.

- Polices et palettes de couleurs :

  Pour créer un affichage de texte et des couleurs uniformes, un ensemble prédéfini de listes de polices et/ou de palettes de couleurs peut être défini pour une utilisation globale.

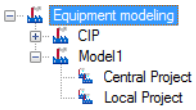
- Tables de langues :

Si l'application d'un standard entraîne la définition d'un ensemble de conditions et de déclarations à utiliser dans une configuration, 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.

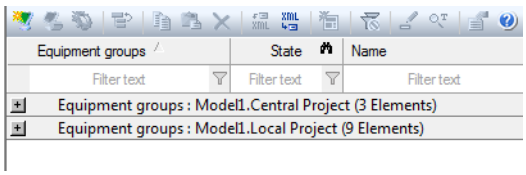
EXEMPLE

Un point de configuration central 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 39). Les équipes peuvent alors l'importer dans leur espace de travail. Une directive doit clairement stipuler que les points de configuration locaux ne doivent pas apporter de modifications à un projet global, car celles-ci seraient remplacées à chaque mise à jour provenant du point de configuration central.

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 à des fins d'assistance lors de la visualisation du projet dans Editor. Dans le détail, cela donnerait l'impression que chaque élément configuré dans Editor a été associé à un groupe d'équipements correspondant.



Les synoptiques de procédé existants sont affichés après l'attribution des équipements.

Par conséquent, un point de configuration local peut clairement identifier les synoptiques mis à jour depuis un point de configuration central, au moyen d'une procédure d'importation au format XML. Les opérateurs procédant à la configuration locale d'un projet peuvent également facilement distinguer les synoptiques qu'ils ne sont pas autorisés à modifier. 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 complète. Les modifications individuelles peuvent également être mises à jour localement, via l'exportation et l'importation de données au format XML (à la page 37).

5.6 XML

L'importation et l'exportation via l'interface XML offrent plusieurs 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 files avec différents contenus doivent être utilisés. Pour accepter le contenu de différents fichiers XML, les dépendances existantes doivent être prises en compte :

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

EXEMPLE POUR UN SYNOPTIQUE AU FORMAT XML :

La méthode la plus exhaustive consiste à exporter un synoptique, car le fichier XML ainsi créé contient non seulement le synoptique et les éléments contenus, mais également le cadre, les variables et les fonctions associés – tous les éléments visibles sur le synoptique et directement liés à celui-ci.

Sur le synoptique **A**, nous utilisons un bouton pour exécuter une fonction **B**, qui rouvre 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.

Vous devez donc :

1. D'abord, importer la fonction,
2. puis le synoptique, et ensuite
3. importer une nouvelle fois la même fonction ; dans le cas contraire, la relation avec le synoptique ne pourra pas être créée dans la fonction.

HARMONISATION DE PROJETS AU SEIN D'USINES

La fonction d'exportation/importation au format XML est également utile pour garantir l'uniformité des projets au sein d'une usine. Si une entreprise décide que certains composants d'un projet doivent être conçus de manière uniforme, 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 par le biais de l'importation au format XML

Une autre méthode consiste à effectuer l'exportation et l'importation par le biais 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 38)

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, sous la forme de données individuelles.

PERSONNALISATION À L'AIDE D'UN ASSISTANT

Un assistant peut également être utilisé pour configurer des fonctions standard, présentes au format XML. Une fonction est importée au format XML, puis adaptée aux exigences spécifiques du projet par le biais de la modification de 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. Voici quelques exemples de situations dans lesquelles l'utilisation d'assistants est recommandée :

- Création de projets de modèles :

  Un assistant permet de créer certains composants d'un nouveau projet en quelques clics. Contrairement au simple chargement d'une sauvegarde de projet (à la page 39) complète du projet de modèle, l'opérateur configurant le projet dispose d'une plus grande capacité d'individualisation.

  Par exemple, un assistant peut créer différents types de projet, selon la machine concernée.

- Création de projets prédéfinis:

  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. De cette manière, les synoptiques, fonctions, variables, etc. peuvent être configurés 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 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, paramètres 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 réutilisés 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 39)

Enregistrer sous (à la page 40)

Gestion multi-projets (à la page 40)

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 37).

Notez que les éléments existants portant le même nom sont remplacés lors de l'importation. Durant la configuration, les éléments configurés pouvant être remplacés durant l'importation doivent être

clairement identifiés. Par exemple, l'utilisation d'un suffixe correspondant au nom de l'élément est envisageable, tout comme la procédure d'utilisation de groupes d'équipements (reportez-vous à la section projet global (à la page 35)).

5.8.2 Enregistrer sous

La fonction **`Enregistrer sous`** créer une copie d'un projet, sous un nouveau nom et avec un nouveau GUID, dans la zone de travail active.

EXEMPLE

Les projets individuels associés à différents types de machines doivent être créés sur la base d'un projet de modèle. Pour cela :

1. Une sauvegarde de projet (à la page 39) du projet de modèle doit être chargée
2. Plusieurs projets portant des noms différents sont créés dans l'espace de travail actif, à l'aide de la fonction **`Enregistrer sous`** (par exemple, **`Machine 1`**, **`Machine 2`**, etc.).

Chaque projet de machine comporte une situation de projet initiale identique. En revanche, les projets ne sont plus liés à un projet de modèle. Les modifications apportées à un projet de modèle ne sont plus automatiquement acceptées dans les projets copiés. Toutefois, les modifications apportées au modèle peuvent être introduites dans les projets individuels par le biais de la fonction d'exportation/importation (à la page 37) au format XML.

Notez que les éléments existants portant le même nom sont remplacés lors de l'importation. Durant la configuration, les éléments configurés pouvant être remplacés durant l'importation doivent être clairement identifiés. Par exemple, l'utilisation d'un suffixe correspondant au nom de l'élément est envisageable, tout comme la procédure d'utilisation de groupes d'équipements (reportez-vous à la section projet global (à la page 35)).

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. De cette manière, un projet peut être modifié de manière centrale, tandis que la configuration locale est effectuée dans un autre projet.

EXEMPLE

Les éléments graphiques peuvent être réutilisés dans un projet local par le biais d'un point de configuration central, afin de créer une interface utilisateur uniforme à l'échelle globale. Pour cela, la configuration centrale crée un projet de modèle, qui constitue la base graphique utilisée par toutes les

autres équipes de configuration. Toutefois, si une modification est apportée à la conception de l'interface utilisateur à l'échelle centrale, celle-ci doit être mise en œuvre sur tous les sites 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. 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.

> **Informations**
>
> *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. La création de systèmes redondants, formant un réseau efficace, à partir de projets autonomes isolés est tout aussi simple.

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 :

- permet de créer plusieurs projets - qui peuvent comporter plusieurs utilisateurs - simultanément sur le réseau.
- Garantissez la protection de vos équipements grâce à la redondance circulaire totale de zenon.
- Installez des systèmes distribués sans la moindre difficulté.
- Disposez d'un accès distant aux postes.
- Suivez et contrôlez vos équipements via zenon Web Server.
- Pouvez consulter les données de procédé sur tous les postes en temps réel.

- Pouvez afficher les actions définies sur un poste de travail (par ex. l'acquittement d'alarmes) sur tous les autres postes.
- Pouvez procéder à la journalisation et l'archivage centralisés des actions.
- Pouvez utiliser les données de procédé instantanément 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 :

- Editor (à la page 42)
- Runtime (à la page 44)
- La phase de test (à la page 52)
- Raccourcis clavier (à la page 46)

7.1 zenon Editor

Conseils relatifs à zenon Editor :

GÉNÉRAL

- Enregistrer les vues d'Editor : Les profils Editor vous permettent d'enregistrer vos divisions de fenêtres et vos attributions de modules personnelles.
- Accédez directement au dossier des fichiers d'Editor : 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 structurés.
  - 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 Runtime de zenon

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 et/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 **Ecriture 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
Transport distant : Fermer le mode plein écran	Ctrl+Alt+Maj+F
Démarrer VSTA Editor	Alt+F10
Démarrer VBA Editor	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\[Dossier SQL]\[UID]\FILES	Ctrl+Alt+E
Ouvre l'explorateur de fichiers en affichant les fichiers du projet actuel. Par exemple : C:\Users\Public\Documents\<CD_zenon_Projects\[Projet]\INI\[Dossier]\INI	Ctrl+Alt+D
Ouvre l'explorateur de fichiers en affichant les fichiers de Runtime du projet actuel. Par exemple : C:\Users\Public\Documents\zenon_Projects\[Projet]\INI	Ctrl+Alt+R
Démarrer le Runtime ; créer préalablement les fichiers du Runtime modifiés.	F5
Créer les fichiers du Runtime modifiés	F7

PROFILS EDITOR

Commande	Raccourci-clavier
Charger le profil Editor 1	Maj+F1
Charger le profil Editor 2	Maj+F2
Charger le profil Editor 3	Maj+F3
Charger le profil Editor 4	Maj+F4
Charger le profil Editor 5	Maj+F5
Charger le profil Editor 6	Maj+F6

Charger le profil Editor 7	Maj+F7
Charger le profil Editor 8	Maj+F8
Enregistrer la vue actuelle d'Editor 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 la touche **`Ctrl`** enfoncée, puis appuyez sur la touche **`A`**.

`Ctrl++` signifie : maintenez la touche **`Ctrl`** enfoncée, puis appuyez sur la touche **`+`**.

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`

SÉLECTIONNER

Commande	Raccourci-clavier
Sélectionner plusieurs objets	Appuyez sur `Maj` ou `Ctrl`
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	Appuyez sur `Maj` durant la sélection
Sélectionner les objets masqués	1. Appuyez sur `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	`Tab.`
Sélectionner l'élément précédent dans l'ordre de création des éléments	`Maj` + `Tabulation`

POSITION

Commande	Raccourci-clavier
Déplacer l'objet sélectionné	`Cursor keys (Touches fléchées du clavier)`
Déplacer l'objet de 10 pixels à chaque pression sur une touche fléchée	`Maj` + `touches fléchées`
Déplacer l'objet horizontalement ou verticalement	Appuyez sur `Maj` durant 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	Appuyez sur Ctrl durant le déplacement
Supprime l'élément sélectionné.	Suppr
Couper l'élément sélectionné	Maj + Suppr. Ctrl+X
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 le coin au pixel près en vous servant des **touches fléchées**, ou le déplacer par incréments de 10 pixels en appuyant sur les touches fléchées+Maj. Remarque : si un angle différent de 0 a été défini pour 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	Appuyez sur Alt durant le redimensionnement.
Redimensionnement proportionnel	Appuyez sur Maj durant le redimensionnement

VUE DE DÉTAIL

Commande	Raccourci-clavier
Crée un nouvel élément pour le module concerné	Insér.
Modifie la colonne sélectionnée	F2
Copie un élément sélectionné de la liste	Ctrl+C
Insère un élément préalablement copié de la liste	Ctrl+V
Supprime un élément sélectionné de la liste	Suppr.
Fait défiler la liste de plusieurs éléments vers le haut	Page préc.
Fait défiler la liste de plusieurs éléments vers le bas	Page suiv.
Permet de parcourir la liste	Touche fléchée

RUNTIME

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

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`.