Manuel de zenon

Energy Edition (Produit Energie)

v. 7.60

©2017 Ing. Punzenberger COPA-DATA GmbH

Alle Rechte vorbehalten.

Die Weitergabe und Vervielfältigung dieses Dokuments ist - gleich in welcher Art und Weise - nur mit schriftlicher Genehmigung der Firma COPA-DATA gestattet. Technische Daten dienen nur der Produktbeschreibung und sind keine zugesicherten Eigenschaften im Rechtssinn. Änderungen - auch in technischer Hinsicht - vorbehalten.

Contenu

1. Bienvenue dans l'aide de COPA-DATA 5

2. Energy Edition (Produit Energie) 6

3. Coloration Automatique des Lignes (CAL) - Topologie 7

3.1 Éléments CAL 8

3.1.1 Éléments procéduraux 10

3.1.2 Lignes 22

3.1.3 Vérification du projet 29

3.2 Configuration 30

3.2.1 Configuration des sources 30

3.2.2 Configuration des verrouillages topologiques 35

3.2.3 Configuration du marqueur de synoptique 40

3.3 Fonction : Modifier la couleur de la source (CAL) 42

3.4 Alias pour les synoptiques détaillés 43

3.5 Localisation de défauts sur les réseaux électriques 46

3.5.1 Rechercher un défaut de terre 48

3.5.2 Recherche de court-circuit 54

3.6 Détection des erreurs par impédance et calcul de la répartition de charge 58

3.6.1 Localisation de défauts par impédance pour le court-circuit 58

3.6.2 Modèle topologique étendu 60

3.6.3 API 62

4. Command Processing 63

4.1 Command Processing 66

4.2 Command processing detail view toolbar and context menu 66

4.3 Engineering in the Editor 69

4.3.1 Creating a screen of the type Command Processing 71

4.3.2 Variables of the command group 89

4.3.3 Configure command input 92

4.3.4 Create menu 143

4.3.5 Create Runtime files 145

4.4 Operating during Runtime 149

4.4.1 Execution of a command 150
4.4.2 Screen type Command Processing 159
4.4.3 Reload project online 172
4.4.4 Logging in the CEL 173
4.4.5 Server change in redundant operation 173
4.4.6 Exit Runtime 173
4.4.7 Lock return variable 174

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 (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. Energy Edition (Produit Energie)

zenon Energy Edition est une suite dotée de fonctionnalités spéciales pour l'industrie de la production d'énergie et des technologies procédurielles. L'utilisateur bénéficie de fonctions faciles à déployer, qui autorisent l'ajustement individuel de l'application à l'environnement physique.

Informations concernant la licence

Ce module doit disposer d'une licence dans Editor et le Runtime.

Energy Edition propose également les fonctions suivantes :

- Gestion de commandes
- CAL (Coloration Automatique des Lignes) : déjà incluse dans la licence du module Energy Edition, fournit des propriétés de coloration de lignes de base.
- Séquences de commandes
- Transformation d'éléments topologiques
- Pack Topologie : nécessite l'ajout de licences au niveau du serveur (pas du client) et étend le module CAL avec les fonctions suivantes :
  - Alimentation multiple
  - Alimentation sécurisée
  - Verrouillages topologiques
  - Déconnexion d'éléments topologiques
  - Détection d'erreur et recherche de défaut de terre

3. Coloration Automatique des Lignes (CAL) - Topologie

La coloration topologique des lignes permet d'attribuer un effet dynamique automatique aux tubes des composants technologiques (médias) et aux lignes de distribution d'énergie (pour l'électricité). Ainsi, la coloration de réseaux topologiques contrôlée par le procédé peut être réalisée facilement.

Parce que la structure des tubes est définie dans les synoptiques avec tous ses éléments technologiques (par exemple, les réservoirs et les vannes ou les générateurs, les commutateurs et les consommateurs d'énergie), elle est émulée en tant que modèle et le flux est affiché dans le Runtime.

Pour permettre les modèles de chevauchement dans les synoptiques, la conception et la configuration intégrales sont toujours gérées globalement au niveau du projet. Vous disposez ainsi d'un modèle topologique complet, utilisé pour la simulation des états des tubes et, finalement, pour la coloration des tubes.

La topologie entière est générée automatiquement à partir de la conception graphique. Aucune autre action de développement n'est nécessaire.

Informations

L'algorithme du module CAL s'exécute une fois seulement, depuis une source commençant au niveau de chaque interrupteur.

SYNOPTIQUES DE DÉTAIL

Pour afficher des synoptiques individuels, une section partielle peut être sélectionnée dans le réseau topologique et affichée individuellement, au moyen d'alias. Un synoptique de détail (à la page 43) peut donc être affiché avec les données provenant de différents composants d'équipements, tels que des sorties ou des réseaux partiels.

Informations concernant la licence

Ces fonctions ou modules doivent disposer d'une licence pour Editor et le Runtime (autonome, serveur, veille).

Aucune licence n'est nécessaire pour le client de Runtime.

La gestion des licences est assurée par le module zenon Energy Edition.

- CAL : inclus dans la licence du module Energy Edition ; fournit des propriétés de coloration de lignes de base.
- Pack Topologie : nécessite l'ajout de licences au niveau du serveur (pas du client) et étend le module CAL avec les fonctions suivantes :
  - Alimentation multiple
  - Alimentation sécurisée
  - Verrouillages topologiques
  - Éléments topologiques Transformateur et Séparateur
  - Détection d'erreur (version 6.50 et ultérieures)

3.1 Éléments CAL

La fonction de coloration automatique des lignes (CAL) permet de colorer les lignes en fonction de l'état du procédé. L'élément combiné est utilisé comme élément de procédé. Le module de Coloration automatique des lignes permet de définir automatiquement un effet dynamique pour les tuyaux, tubes ou lignes dans le secteur des technologies (pour les médias), ainsi que dans les réseaux topologiques (pour l'électricité).

DÉVELOPPEMENT

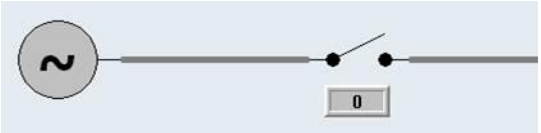
Pour la configuration, on distingue deux types d'éléments de synoptique avec différentes fonctions. Il s'agit d'une part d'éléments de procédure (à la page 10) (source, interrupteur/disjoncteur, drain, transformateur et/ou liaison) et, d'autre part, de lignes (à la page 22).

Dans ce contexte, les éléments techniques comportent une fonction et une couleur (source et transformateur). Si les éléments de procédure sont actifs, les lignes connectées adoptent la couleur de ces éléments à la source et au transformateur, ou adoptent la couleur de la ligne d'entrée de l'élément pour l'interrupteur et la liaison. Si les éléments de procédure sont inactifs, la couleur des lignes est reprise de la définition dans Editor.

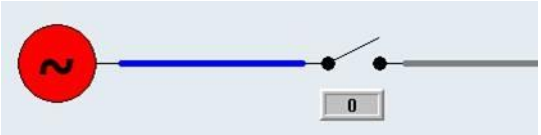
Les différentes fonctionnalités des éléments sont définies dans les propriétés de l'élément combiné.

EXEMPLE

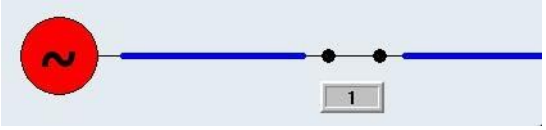
Une source a une ligne connectée. Un interrupteur est connecté à la ligne. Et une seconde ligne est connectée aussi. Si la source est active, la première ligne est colorée avec la couleur de coloration automatique des lignes définie pour la source, jusqu'à l'interrupteur. L'autre ligne n'est pas colorée avant la fermeture de l'interrupteur.



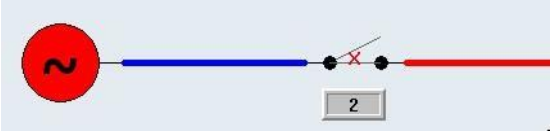
Source inactive



Source active



Interrupteur fermé



Non défini ou non valide

> **Informations**
>
> *Si l'état de l'élément de procédure est `Non défini` ou `Défaillance`, cet état est détecté automatiquement. Toutes les lignes connectées et tous les éléments suivants sont affichés dans la couleur de la source prédéfinie `non définie`, dans les deux états.*

NOMBRE DE COMMUTATEURS FERMÉS EN SÉRIE

Pour garantir le bon fonctionnement de l'algorithme CAL, il est important de tenir compte du nombre de commutateurs fermés connectés en série.

Recommandation : ne pas dépasser 256 commutateurs fermés en série entre la source et le drain.

3.1.1 Éléments procéduraux

Les éléments procéduraux sont créés dans zenon à l'aide d'un **élément combiné**. Leur état détermine la coloration de la ligne connectée.

Les paramètres suivants sont disponibles :

Propriété	Description
Le type de fonction	Cela définit le type technologique de l'élément combiné.
`Terminaison`	Pour les côtés barre omnibus. Bloque l'affichage du message d'erreur **Ligne connectée d'un côté seulement** lors de la compilation dans Editor.
`Générateur`	Un générateur se comporte généralement comme une source, mais il est considéré comme indépendant et non synchrone.
`Aucune fonction`	L'élément ne comporte aucune fonction dans le module Coloration Automatique des Lignes. **Remarque :** Le type de fonction `Aucune fonction` est la valeur par défaut.
`Lien`	Avec un lien, une ligne peut être continuée à un autre emplacement. Si un lien est affecté par une ligne, tous les autres liens avec le même numéro de lien sont également affectés par cette ligne. Ici, peu importe si les liens sont sur le même synoptique ou dans d'autres synoptiques du projet. De cette façon, des lignes indépendantes des synoptiques peuvent être définies. Il est possible d'avoir plus de 2 liens connectés au même numéro de lien dans un projet. Les liens peuvent être alimentés par plusieurs lignes en même temps, et peuvent également alimenter plusieurs lignes. En principe, il n'y a pas de différence entre les entrées et les sorties. L'information source est toujours transmise à toutes les lignes connectées. **Attention :** Deux éléments "Lien" ne peuvent pas être connectés directement à une ligne. Entre les deux, il doit exister au moins un autre élément procédural (commutateur/sectionneur ou transformateur). Un lien ne peut pas être activé ou désactivé ; il est toujours actif.
`Source`	Transmet sa couleur. Si la source est active (valeur : `1`), toutes les lignes connectées pour lesquelles l'option **Couleur définie par CAL** est définie dans les propriétés de l'élément reçoivent la couleur de la source. Les couleurs des sources sont définies dans les propriétés du projet. (par ex. cuves ou générateurs). Une source est un pôle unique auquel est associé un numéro de source unique. La source peut être commutée en fonction de l'état de sa variable principale. En règle générale, les sources sont considérées comme synchrones avec le réseau et amovibles. Pour des informations détaillées concernant la source, reportez-vous au chapitre Configuration des sources (à la page 30).
`Commutateur`	Avec ceci, les lignes peuvent être interrompues. Si l'interrupteur est `fermé/actif` (valeur : `1`), la connexion entre les deux lignes est fermée et la ligne est colorée avec la couleur définie pour la source jusqu'au prochain interrupteur. Dans ce cas, un interrupteur transmet la couleur de la source de la ligne d'entrée vers la ligne de sortie. Si l'état de l'interrupteur est `non valide` (valeur : `3`) ou `non défini` (valeur : `2`) ou si l'état de la variable principale est NON VALIDE, la ligne est colorée dans la couleur `non définie` issue de la configuration du module CAL dans les propriétés de projet. Ainsi, un interrupteur transmet le numéro de source `0` (UNDEFINED) au niveau de sa sortie (connexion `2`), au lieu du numéro de la source

	entrante. **Exemple :** se reporter à la section **Exemple d'interrupteur - Couleurs de la fonction CAL** (à la page 16). **Remarque** : Si la propriété **Commuter entrée/sortie** est active, l'entrée et la sortie de cet élément sont inversées pour la fonction CAL.
`Vanne`	Une réglette (vanne) se comporte de la même manière qu'un `commutateur`, à ceci près qu'elle est utilisée pour les conduites d'eau et de gaz. Valeur de la variable principale : ▸ Interrupteur DÉSACTIVÉ : Valeur `0` -> vanne en position fermée -> pas d'écoulement d'eau ▸ Réglette ACTIVE : valeur `1` -> vanne en position fermée -> pas d'écoulement d'eau ▸ Réglette DIF : valeur `2` -> vanne en position partiellement ouverte -> écoulement d'eau ▸ Réglette STO : valeur `3` -> défaillance de la vanne **Remarque** : Si la propriété **Commuter entrée/sortie** est active, l'entrée et la sortie de cet élément sont inversées pour la fonction CAL.
`Drain`	Ceci définit la fin de la ligne. Le drain n'influence pas la coloration ; il est uniquement utilisé pour permettre l'affichage du modèle en totalité. Si un programme externe (par exemple, VBA) voulait accéder au modèle, alors le drain serait probablement nécessaire pour pouvoir faire d'autres calculs, et donc il doit être inséré. Dans les projets du module Energy, le drain est utilisé pour représenter les consommateurs. Ces éléments sont utilisés pour calculer les verrouillages topologiques "**L'appareil ne sera pas alimenté**" du module CAL (dans le cadre de la **gestion de commande**).
`Clapet antiretour`	Le `clapet antiretour` transmet uniquement des informations dans une direction. Valeur de la variable principale : ▸ `Valeur 0:` La transmission d'informations n'est pas active (= clapet fermé) ▸ `Valeur 1 ou 2:` La transmission d'informations est possible dans une direction seulement. Dans ce cas, la couleur de la source est uniquement transmise de l'entrée à la sortie. La transmission dans la direction opposée n'est pas assurée. Ceci affecte également la transmission de la terre. ▸ `Valeur 3:` La transmission d'informations n'est pas définie. Ceci se produit par exemple en cas de défaillance du `clapet antiretour`. Dans ce cas, l'état est uniquement transmis au niveau de la sortie.

	Remarque : Si la propriété **Commuter entrée/sortie** est active, l'entrée et la sortie de cet élément sont inversées pour la fonction CAL. Le `clapet antiretour` est également pris en compte par les verrouillages topologiques (à la page 35).
`Transformateur`	Un transformateur est un drain et une source en même temps. Donc avec un transformateur, la couleur d'entrée (source d'entrée) peut être transformée en une nouvelle couleur de sortie (couleur source du transformateur). La ligne de sortie est uniquement définie comme active lorsque le transformateur comporte une ligne d'entrée active. Toutefois, la ligne de sortie ne reçoit pas la couleur de la ligne d'entrée comme avec un commutateur, mais reçoit au lieu de ça la couleur source du transformateur. Donc une source doit être définie pour chaque transformateur. Un transformateur ne peut pas être activé ou désactivé, il est toujours actif. **Remarque** : Si la propriété **Commuter entrée/sortie** est active, l'entrée et la sortie de cet élément sont inversées pour la fonction CAL. `Transformateur compatible avec une production réversible :` Pour obtenir un transformateur compatible avec une production réversible, vous devez sélectionner une source autre qu'`UNDEFINED [0]` pour l'élément **Source pour alimentation inverse**. Cela signifie que le transformateur se comporte de la même manière dans les deux directions : de l'entrée vers la sortie (en avant), mais également de la sortie vers l'entrée (en arrière). La seule différence est que la propriété **Source pour alimentation inverse** est utilisée pour la distribution ultérieure du numéro de source, et non la propriété **Source**. **Remarque :** Les états de réseau défaillants ou les configurations manquantes, tels qu'un signal provenant de l'entrée et de la sortie en même temps ou qu'un court-circuit à l'entrée et à la sortie, ne sont pas colorés de manière particulière. Cela signifie que le transformateur capable d'assurer une production réversible se comporte comme deux transformateurs commutés de manière antiparallèle, qui ne sont pas capables d'assurer une production réversible.
`Sectionneur`	Un sectionneur se comporte généralement comme un `commutateur`. Toutefois, un sectionneur dans le modèle topologique ne doit pas être activé ou désactivé lorsqu'il est en charge (verrouillage topologique "**Sectionneur en charge**" dans la gestion de commande). À l'image de l'interrupteur, la variable détermine l'état : `Marche, Arrêt, Position intermédiaire, Défaillance`. **Remarque** : Si la propriété **Commuter entrée/sortie** est active, l'entrée et la sortie de cet élément sont inversées pour la fonction CAL.

Les nombres sources indiqués pour les types de fonctions de la source et du transformateur sont transmis aux appareils (drains) par l'intermédiaire des interrupteurs fermés (sectionneurs, réglettes, etc.). Les couleurs de l'ensemble des lignes connectées et éléments procéduraux sont calculées à partir de la somme de rang supérieur des nombres sources d'alimentation.

NUMÉRO DE SOURCE ET DE LIEN

Paramètre	Description
Source	Ici une source peut être assignée à un élément. Dans la liste, on trouve toutes les sources définies dans les propriétés CAL du projet. Tous les noms de sources sont affichés. Cette propriété est uniquement active si le type de fonction sélectionné est `Source`, `Transformateur` ou `Générateur`. Pour des informations détaillées concernant la source, reportez-vous au chapitre Configuration des sources (à la page 30). **Attention :** les sources système prédéfinies (ID 0 à 9) ne doivent pas être utilisées en tant que sources personnalisées dans le projet ; exception : la terre.
Numéro de lien	Seul le numéro de liaison est saisi pour une fonction de `liaison`. Tous les numéros de lien identiques d'un projet sont corrélés. Description détaillée dans le type de fonction `Lien`. Cette propriété est accessible uniquement si le type de fonction est `Lien`.

VARIABLES D'ÉLÉMENTS PROCÉDURAUX

Pour qu'un interrupteur (ou un sectionneur, une réglette, etc.) reçoive l'état ouvert, fermé ou non valide, un type de données BOOL ou une variable entière doit être lié(e) en tant que variable principale à l'élément combiné correspondant.

Exemple :

- Driver IEC870 : Variables avec un **ID de type** compris entre `T01 et T37`
- Driver IEC850 : Variables `*/Pos/stVal[ST]`
- Driver DNP3 : Variables d'entrée

Condition préalable : le **mappage DPI/DPC** n'a pas été désactivé dans le driver.

Informations

Pour la position d'un interrupteur, seuls les deux premiers bits de la variable principale sont pris en compte.

- Le premier bit est la commutation réelle : 0 signifie ARRÊT et 1 signifie MARCHE.
- Le deuxième bit est le bit d'erreur. Aucune erreur n'est présente lorsque ce bit est égal à 0.

Pour une source, l'état "Présent" (MARCHE) / "Non présent" (ARRÊT) doit également être évalué à l'aide de la variable principale liée. Les variables de type de données BOOL internes sont les mieux adaptées à cette fin. La source peut alors être liée au reste de la topologie à l'aide d'un interrupteur ou d'un sectionneur (ce qui est une pratique commune, dans la réalité), afin que la couleur de la source soit uniquement transmise par la position de l'interrupteur.

Note : pour la variable principale d'une source connectée au réseau par l'intermédiaire d'un interrupteur/sectionneur, telle que la terre, créez une variable de **driver interne**, configurez-la avec le paramètre **réseau de calcul** et la **valeur initiale** `1` ("Toujours présent"). Vous pouvez également directement lier une source à une variable de procédé variable (la source et l'interrupteur associé réunis) si vous n'avez pas besoin de **verrouillages topologiques** lors de la commutation de la source.

ÉTATS

- Un interrupteur et une source sont activés (fermés) si la valeur de la variable liée est `1`.
- Un interrupteur est non valide si la valeur de la variable liée est `>1` ou comporte un état `INVALID`.
  Un interrupteur non valide fournit le numéro de source `0` (non défini) à sa sortie (connexion 2) au lieu de la saisie du numéro de source. Dans la direction vers l'entrée, l'interrupteur fonctionne normalement.

  Remarque : si la variable principale possède l'état `INVALID`, l'ensemble du réseau suivant est `INVALID`, car l'état du réseau est inconnu. L'état `INVALID` est transmis par le biais des interrupteurs fermés suivants.

Attention

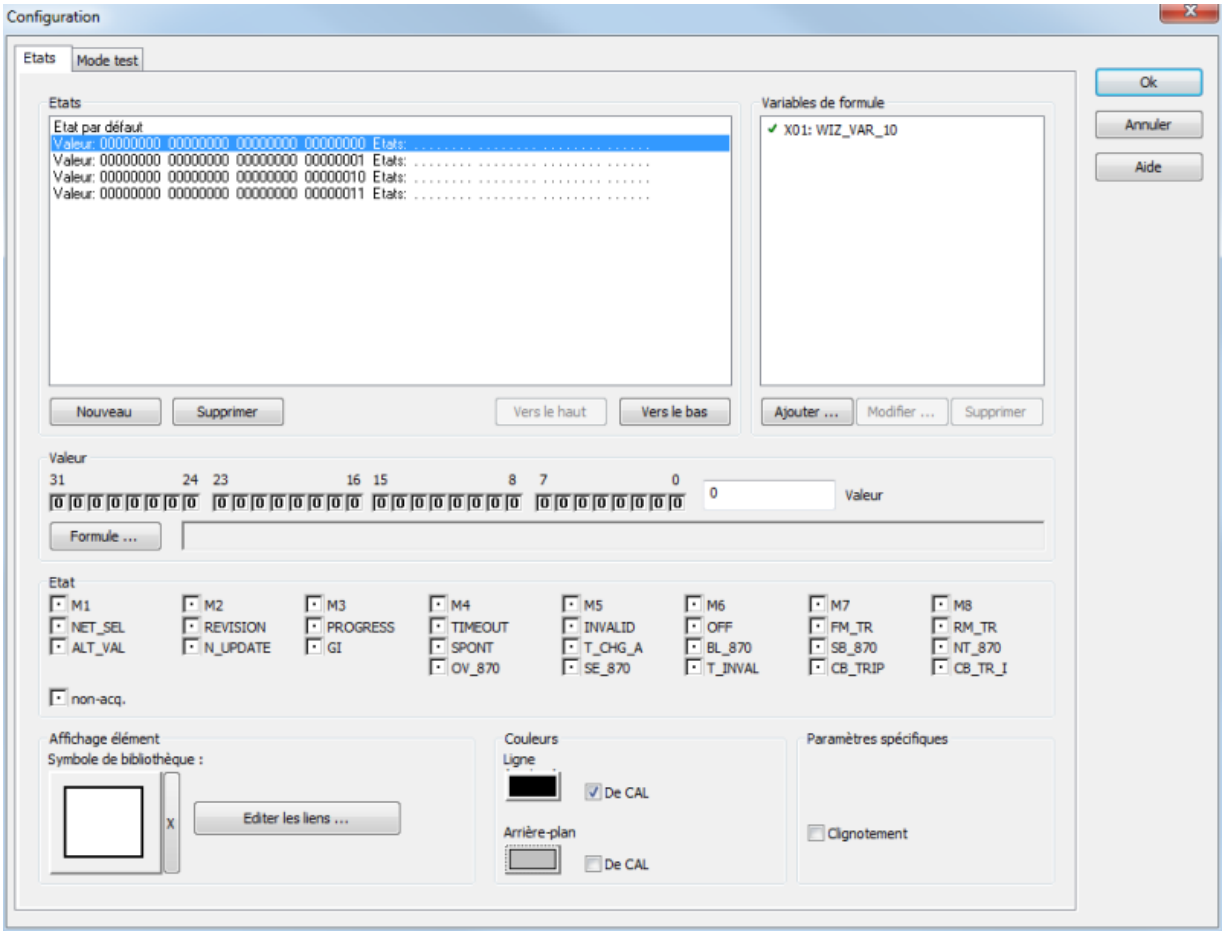
*Dans les états individuels de l'élément combiné, si la couleur et la couleur de remplissage de la fonction **CAL** sont activées, la ligne et les éléments procéduraux sont colorés dans le Runtime.*

Voir la section Exemple d'interrupteur - Couleurs de la fonction CAL.

EXEMPLE 1

Élément combiné avec l'état de valeur `00` et coloration de ligne de la fonction CAL :

1. Configuration dans Editor :
   - Élément combiné avec l'état de valeur `00`
   - Coloration de ligne de la fonction CAL active

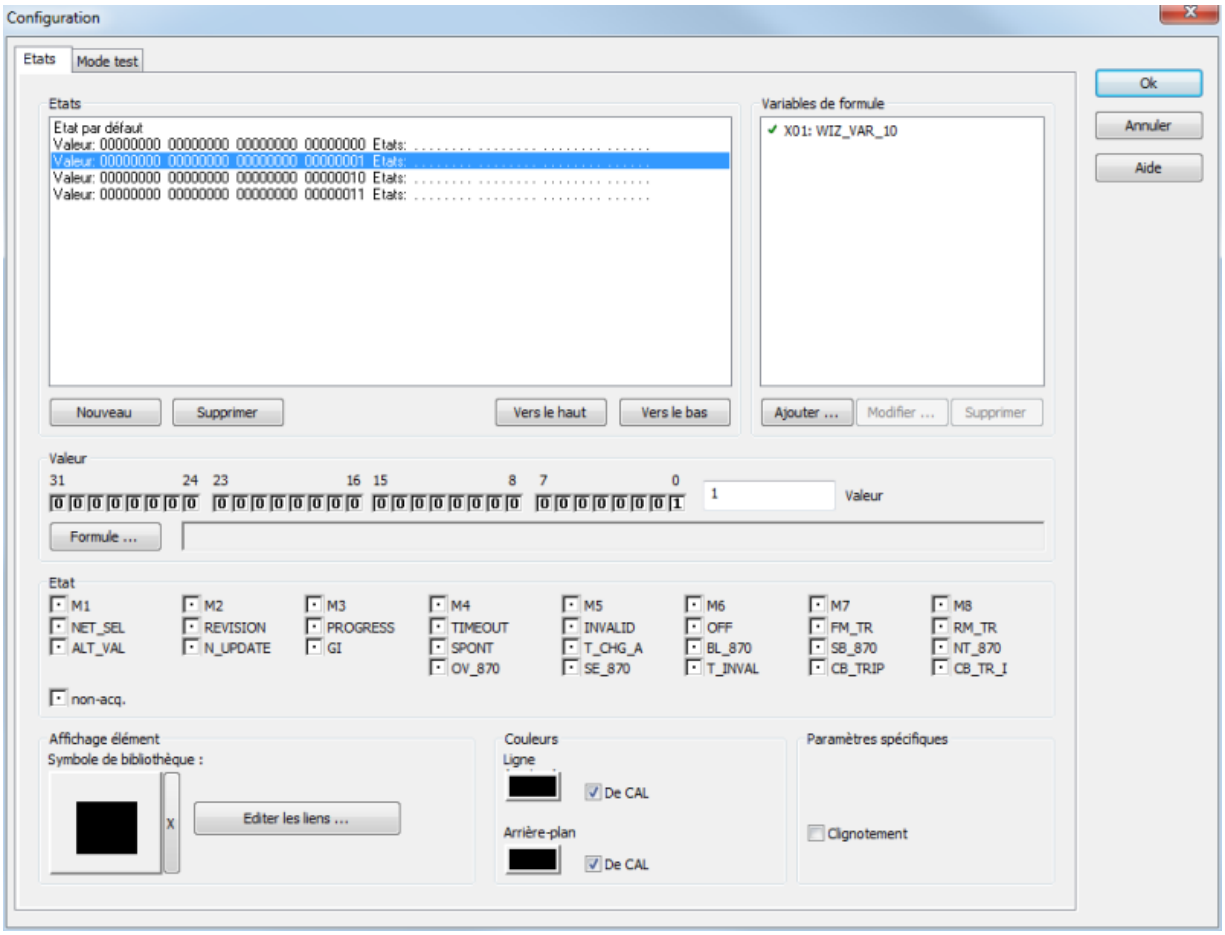


2. Résultats suivants dans le Runtime :
   - Couleur de la source : vert
   - Couleur sans tension : blanc
   - État de l'interrupteur : `désactivé/ouvert` (valeur `0`)

EXEMPLE 2

Élément combiné avec l'état de valeur `01` et coloration de ligne de la fonction CAL :

1. Configuration dans Editor
   - Élément combiné avec l'état de valeur `01`
   - Coloration de ligne de la fonction CAL active
   - Couleur de remplissage de la fonction CAL active

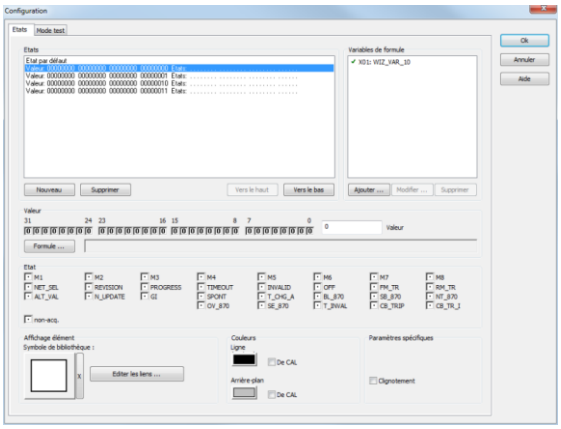


2. Résultats suivants dans le Runtime :
   - Couleur de la source : `Vert`
   - Couleur sans tension : `Blanc`
   - État de l'interrupteur : `activé/fermé` (valeur `1`)

EXEMPLE 3

Élément combiné avec l'état de valeur `00` sans coloration de ligne de la fonction CAL :

1. Configuration dans Editor :
   - Élément combiné avec l'état de valeur `00`
   - Coloration de ligne de la fonction CAL inactive



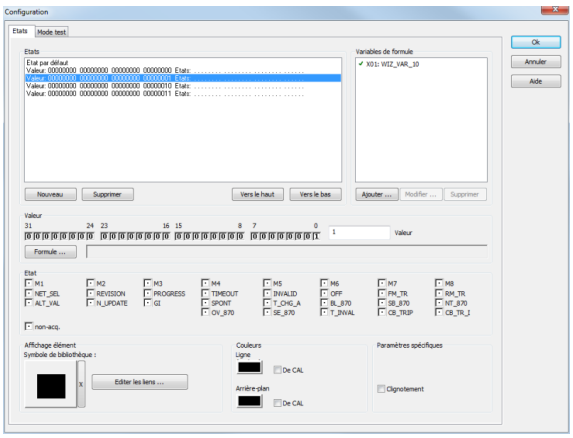
2. Résultats suivants dans le Runtime :
   - Couleur de la source : `Vert`
   - Couleur sans alimentation et couleur de construction de la ligne : `Blanc`
   - Ligne définie et couleur de remplissage de l'élément combiné : `noir`.
   - État de l'interrupteur : `désactivé/ouvert` (valeur `0`)

EXEMPLE 4

Élément combiné avec l'état de valeur `01` sans coloration de ligne de la fonction CAL :

1. Configuration dans Editor
   - Élément combiné avec l'état de valeur `01`
   - Coloration de ligne de la fonction CAL inactive

- Coloration de remplissage de la fonction CAL inactive



2. Résultats suivants dans le Runtime :
   - Couleur de la source : vert
   - Couleur sans alimentation et couleur de construction de la ligne : `Blanc`
   - Ligne définie et couleur de remplissage de l'élément combiné : `noir`.
   - État de l'interrupteur : `activé/fermé` (valeur `1`)

Points de connexion des éléments de procédure

Lors de la configuration, une ligne est connectée à un élément de procédure (élément combiné) par le chevauchement des tracés sur le synoptique aux points de connexion de l'élément combiné. Une seule ligne peut être connectée au même point de connexion en même temps. Toutes les lignes dont les origines se trouvent dans la zone définie sont connectées (topologie du graphique).

> **Attention**
>
> *Les éléments de CAL doivent uniquement être utilisés sans rotation, car :*
>
> *Le calcul du modèle topologique du CAL dans Editor repose sur la position des éléments sans rotation et sans prise en compte d'éléments dynamiques.*

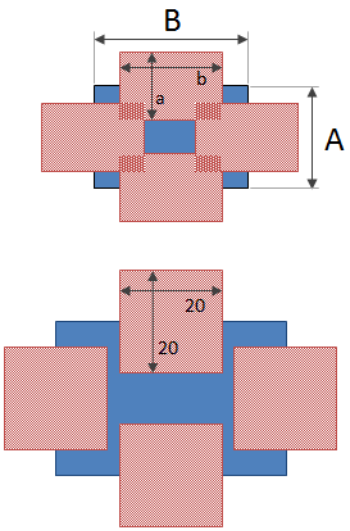
POINTS DE CONNEXION ET ZONES DE CONNEXION

- La zone de connexion d'un point de connexion se trouve au centre de chaque côté de l'élément combiné. Chaque élément combiné comporte donc quatre points de connexion.
- La taille d'une zone de connexion correspond à 2/3 de la hauteur et la largeur d'un élément combiné, mais ne dépasse pas 20 pixels.

- Chaque zone de connexion est centrée dans le coin de l'élément correspondant et s'étend symétriquement vers l'intérieur et l'extérieur, sur une distance maximale de 10 pixels.

> **Attention !**
>
> *Si l'élément combiné mesure moins de 30 pixels, les zones de connexion dans un élément se chevauchent. Les lignes pouvant se toucher peuvent être sources d'erreurs (compilation, coloration).*

Les points de connexion possibles pour les éléments combinés d'une taille supérieure et inférieure à 30 pixels sont visibles dans l'illustration.



Couleurs

- Bleu : Élément combiné
- Rouge : Zones de connexion

Dimensions :

- **A**: hauteur de l'élément combiné
- **B**: largeur de l'élément combiné
- **a**: Largeur de la zone de connexion : 2/3 de **A**, mais 20 pixels maximum.
- **b**: Longueur de la zone de connexion : 2/3 de **B**, mais 20 pixels maximum.

RÈGLES

- Si une ligne se trouve hors de la zone de connexion, aucune connexion n'est détectée et il n'y a donc pas de coloration de la ligne. Et donc il n'y aura pas non plus de coloration des lignes suivantes.

- En règle générale, tous les points de connexion décrits peuvent être utilisés avec les sources, les drains et les `Links`.
  Attention : Avec les sources et les drains, un seul point de connexion peut être utilisé à la fois. Si différents points de connexion sont utilisés simultanément, des états non définis peuvent se produire.
  Les éléments de type `Link` peuvent également utiliser plusieurs points de connexion simultanément. Les informations relatives à la couleur entrante sont transmises à toutes les lignes.
- Dans le cas des interrupteurs, sectionneurs, réglettes et transformateurs, la connexion `1` (entrée) se trouve à gauche ou sur le dessus et la connexion `2` (sortie) se trouve à droite ou en bas. Cette séquence peut être modifiée avec la propriété **Commuter entrée/sortie**.
  Attention : Aux interrupteurs et transformateurs, il convient de veiller à n'utiliser qu'une connexion d'entrée et une connexion de sortie. L'utilisation simultanée de plusieurs points de connexion d'entrée ou de sortie produit des incohérences, et n'est donc pas fiable.
- Pour tous les éléments de procédure, ce qui suit est vrai : Une seule ligne peut être connectée à un point de connexion. Les jonctions ne peuvent pas être établies directement sur un élément ; elles doivent être dessinées avec des lignes.

Commuter entrée/sortie

Si un transformateur, un sectionneur ou un interrupteur est configuré, l'entrée et la sortie peuvent être inversées. Pour cela :

1. Sélectionnez le transformateur, le sectionneur ou l'interrupteur en tant que **Type de fonction**.
2. Cochez la case **Commuter entrée/sortie**

L'entrée est alors définie en bas à droite et la sortie en haut à gauche.

PRÉSENTATION

Configuration de l'équipement	Entrée	Sortie
normal	Gauche	À droite
normal	Haut	En bas
Inversée	Droite	Gauche
Inversée	Bas	Haut

3.1.2 Lignes

Les lignes sont représentées par des éléments vectoriels Ligne, Ligne brisée et Tube.

Si l'option **Couleur définie par CAL** est cochée pour une ligne, la couleur provient de la configuration de la fonction CAL. Les lignes sont colorées automatiquement par le système selon l'état des éléments procéduraux et les paramètres CAL.

Ici, la couleur provient habituellement de la source la plus prioritaire dont le flux circule dans la ligne ou reste "vide/non alimentée", telle qu'elle est définie dans le synoptique avec des couleurs statiques ou dynamiques.

Vous définissez le type d'affichage à l'aide de listes déroulantes :

- Priorité pour l'affichage
- Affichage alimentations multiples
- Affichage alimentation sécurisée

Les options suivantes sont disponibles dans les propriétés des lignes :

Paramètre	Description
Couleur définie par CAL	Active la Coloration Automatique des Lignes pour ces éléments vectoriels. Ceci signifie : Si la source de la ligne est active et que tous les interrupteurs entre la source et la ligne sont fermés (ou si toutes les vannes sont ouvertes), la ligne est colorée en conséquence. Si la ligne est "alimentée" par une source unique, la couleur de la source définie est utilisée pour colorer la ligne. La largeur de la ligne n'est pas modifiée.
Priorité pour l'affichage	Définit si l'`alimentation multiple`, l'`alimentation sécurisée` ou les deux alimentations sont affichées. Par défaut : `Alimentation multiple`
`Alimentation sécurisée`	L'élément est affiché selon les règles de l'alimentation sécurisée. Une ligne est considérée comme dotée d'une alimentation sécurisée si elle est alimentée par deux interrupteurs ou transformateurs aux moins, avec une source située hors du système. Les sources du système ne contribuent pas à l'alimentation sécurisée, mais ne l'excluent pas non plus.
`Alimentation multiple`	L'élément est affiché selon les règles de l'alimentation multiple. Une ligne est considérée comme dotée d'une alimentation multiple si elle est alimentée par deux sources différentes au moins. Dans ce cas, peu importe qu'il s'agisse de sources système ou de sources utilisateur ; le côté depuis lequel la ligne est alimentée par les sources n'a également pas d'importance.
`Pas de priorité`	Les règles de coloration des paramètres `Alimentation multiple` et `Alimentation sécurisée` sont appliquées simultanément, si les deux critères sont satisfaits. Ceci signifie : Si une ligne ▸ comporte une `alimentation multiple` et une `alimentation sécurisée`, ▸ La priorité est définie sur `Pas de priorité` ▸ L'affichage de l'alimentation multiple est défini sur `Les deux sources les plus prioritaires` ▸ L'affichage de l'alimentation sécurisée est défini sur `Double épaisseur` La ligne est alors deux fois plus épaisse, et affichée sous forme de ligne en trait tireté de deux couleurs.

Affichage alimentation multiple	'Alimentation multiple' signifie qu'une ligne est alimentée par plusieurs sources simultanément. Ici vous pouvez définir comment les lignes alimentées de façon multiple sont affichées. Par défaut : `La source la plus prioritaire`
`La source la plus prioritaire`	La ligne prend la couleur de la source la plus prioritaire. **Remarque :** Les priorités sont données par l'ordre des sources dans la configuration CAL globale au projet.
`Les deux sources les plus prioritaires`	S'applique aux lignes alimentées par deux sources différentes ou plus. Les deux sources avec les plus hautes priorités définissent la coloration. La ligne est affichée avec les deux couleurs (en pointillé). La longueur des tirets peut être modifiée via la propriété **Longueur de pointillé fournie plusieurs fois**. Les sources système concernent les alimentations multiples, aussi bien que les sources réelles ; la configuration correspondante permet de colorer les lignes de deux couleurs.
`Couleur alternative`	La couleur définie dans la propriété **Couleur alternative** est utilisée.
Longueur de pointillé fournie plusieurs fois	Définit la longueur des tirets (en pixels) des lignes, lignes brisées ou tubes utilisée par la fonction CAL pour l'option `Les deux sources les plus prioritaires` pour la propriété **Affichage alimentation multiple**. ▸ Minimum : `0` (longueur des tirets automatique) ▸ Maximum : `32767` ▸ Par défaut : `0`
Couleur alternative	Couleur de remplacement utilisée par la fonction de CAL pour les lignes, les lignes brisées ou les tubes dotés d'alimentations multiples.
Affichage alimentation sécurisée	'Alimentation sécurisée' signifie qu'une ligne est alimentée par plusieurs chemins (en parallèle) mais d'une même source. Dans ce cas, vous pouvez définir comment cette "alimentation sécurisée" est représentée. Une ligne est toujours affichée comme dotée d'une `alimentation sécurisée` si elle est alimentée par aux moins deux interrupteurs comportant une source réelle (c'est-à-dire une source autre qu'une

	source système). Par défaut : `normal`
`double épaisseur`	A un sens pour les lignes alimentées de façon parallèle par la même source. Si c'est le cas, l'épaisseur de la ligne est deux fois supérieure à l'épaisseur configurée. (Exemple : Une ligne avec une épaisseur de 5 pixels sera affichée avec une épaisseur de 10 pixels si elle est "alimentée de façon sécurisée" par la même source. Si cette ligne est alimentée par deux sources différentes ou plus (alimentations multiples), l'épaisseur de ligne n'est pas changée ! La couleur est `toujours` définie par la source avec la priorité la plus haute !
`double luminosité`	A un sens pour les lignes alimentées de façon parallèle par la même source. La ligne est affichée avec une luminosité qui est le double de la luminosité définie dans ses propriétés. Si cette ligne est alimentée par deux sources différentes ou plus (alimentations multiples), la couleur de la ligne n'est pas changée ! Si cette ligne est alimentée de façon sécurisée depuis une même source, la luminosité de la ligne affichée est deux fois plus élevée. Formule de calcul de la double luminosité : 1. La couleur RVB est transformée selon le système HLS. 2. L (Luminance = Luminosité) est recalculée avec une valeur NouvelleLuminance égale à : 240*3/4 + L/4. 3. La valeur de la couleur est retransformée dans le système RGB avec la nouvelle luminosité. La couleur est `toujours` définie par la source avec la priorité la plus haute !
`normal`	L'élément est affiché avec la couleur de la source et l'épaisseur configurée.
Utiliser alias	Active : un alias est utilisé.
Alias	Ouvre la boîte de dialogue (à la page 43) permettant de sélectionner un modèle.

Informations

La couleur des sources et les priorités sont définies dans les propriétés du projet.

Les sources définies par l'utilisateur doivent posséder un ID supérieur à 9. Les ID inférieurs ou égaux à 9 sont réservés aux sources système.

Informations

Le calcul de la couleur d'une ligne dans le Runtime est fait selon la liste des priorités suivante :

1. Coloration Automatique des Lignes (Priorité la plus haute, outrepasse tous les autres paramètres)
2. Couleurs dynamiques
3. Couleurs statiques

Exemple

Dans l'exemple suivant, la source 0 est de couleur bleue et la source 1 de couleur rouge. Et la source 0 est la source la plus prioritaire.

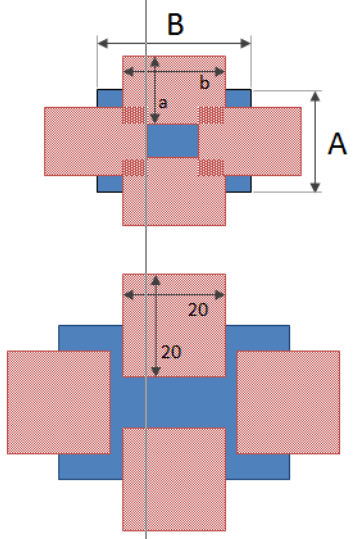
Source 0

Source 1

Il en résulte ceci pour les différentes options :

	Ligne / Ligne brisée	Tube

La source la plus prioritaire		
Les deux sources les plus prioritaires		
double épaisseur		
double luminosité		



Points de connexion des lignes

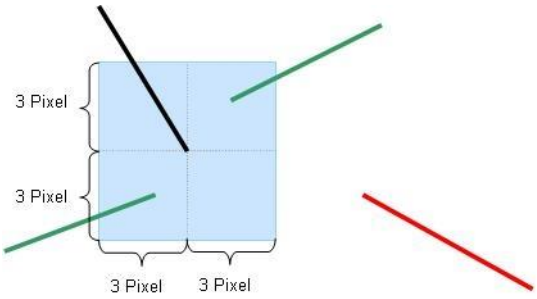
La connexion d'une ligne (ligne, ligne brisée ou tube) sur une autre ligne est faite par superposition /chevauchement du tracé dans le synoptique aux points de connexion. Les points de connexion - ou les zones de connexion - sont les extrémités de chaque ligne et jusqu'à trois pixels autour.

Exemple

Le point de départ d'une ligne a les coordonnées (point de départ x, y) : 150/100 pixels.
Ceci produit une zone de connexion (x / y) : 147 - 153 / 97 - 103 pixels.

Si la ligne commence ou se termine et d'autres lignes commencent ou se terminent dans cette zone, alors les lignes sont automatiquement connectées sans avoir à configurer autre chose. Un seul recouvrement des zones de connexion des lignes n'est pas suffisant !

Dans l'illustration suivante, les zones de connexion sont affichées graphiquement (les lignes vertes sont connectées à la ligne noire, mais pas la ligne rouge).



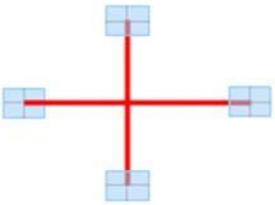
Informations

Le nombre de lignes connectées à une zone de connexion n'est pas limité.

Attention

Si une ligne est en dehors de la zone de connexion (par exemple, la ligne rouge dans l'illustration), aucune connexion n'est établie et il n'y a pas de coloration de la ligne. Et donc il n'y aura pas non plus de coloration des lignes suivantes.

Des croisements de ligne peuvent être effectués, si les fins de lignes ne sont pas dans les zones de connexion.



Attention

Les éléments de CAL doivent uniquement être utilisés sans rotation, car :

Le calcul du modèle topologique du CAL dans Editor repose sur la position des éléments sans rotation et sans prise en compte d'éléments dynamiques.

3.1.3 Vérification du projet

Développez les éléments de procédure et les lignes souhaités sur un ou plusieurs synoptiques, et enregistrez-les. Vous pouvez ensuite vérifier, via les fonctions **Créer tous les fichiers du Runtime** ou **Créer les fichiers du Runtime modifiés**, si des erreurs ou des conflits sont présents sur les synoptiques. En cas d'erreurs ou de conflits, les messages d'erreur ou les avertissements correspondants sont affichés dans la fenêtre de suivi.

Informations

Double-cliquez sur la ligne correspondante dans la fenêtre de suivi. Le synoptique comportant l'élément de synoptique erroné sera ouvert automatiquement. Si l'élément de synoptique erroné forme partie d'un symbole, le symbole correspondant est automatiquement sélectionné.

Le message d'erreur suivant peut être affiché.

- CAL : Synoptique '%s' - Deux éléments Lien comportant des numéros de lien différents sont connectés à la ligne '%s. (Un double-clic ouvre le synoptique et sélectionne la ligne.)
- CAL : Synoptique '%s' - Plus que deux points de connexion sont utilisés sur l'élément '%s'. Sur un élément, vous ne pouvez utiliser qu'un point en entrée et un point en sortie. Il doit être connecté sur un coté. (Un double-clic ouvre le synoptique et sélectionne l'élément.)

Les avertissements suivants peuvent être affichés.

- CAL : Synoptique '%s' - La ligne d'alias '%s' est connectée à une ligne sans alias. (Un double-clic ouvre le synoptique et sélectionne la ligne.)
- CAL : Synoptique '%s' - La ligne d'alias '%s' est connectée à une ligne sans alias. Il doit être connecté sur un coté. (Un double-clic ouvre le synoptique et sélectionne l'élément.)
- CAL : Synoptique '%s' - La ligne d'alias '%s' est connectée à une ligne sans alias. Il doit être connecté sur un coté. (Un double-clic ouvre le synoptique et sélectionne l'élément.)
- CAL : Synoptique '%s' - La ligne '%s' est connectée d'un côté seulement. (Un double-clic ouvre le synoptique et sélectionne la ligne.)
- CAL : Synoptique '%s' - l'élément '%s' n'est pas connecté. Il doit être connecté sur un coté. (Un double-clic ouvre le synoptique et sélectionne l'élément.)
- CAL : Synoptique '%s' - l'élément '%s' n'est connecté que d'un côté. Il doit être connecté sur un coté. (Un double-clic ouvre le synoptique et sélectionne l'élément.)

Dans les messages d'erreur ou les avertissements, les éléments correspondants sont identifiés à l'aide de la référence de l'élément. Cette référence sert également de clé de lien pour les alias de CAL.

3.2 Configuration

Pour configurer la fonction CAL :

1. Dans les propriétés du projet, sélectionnez la propriété **Configuration CAL** du groupe **Coloration Automatique de Lignes**.
2. Cliquez sur le bouton **...**
3. La boîte de dialogue de configuration s'affiche à l'écran.
4. Configurez les propriétés souhaitées pour les éléments suivants :
   - Sources (à la page 30)

     Créez de nouvelles sources dans les couleurs souhaitées pour la topologie.

     Remarque : Dans ce cas, notez que les sources système (ID 0 à 9) possèdent une signification prédéfinie ou sont réservées pour des versions futures.

     Tenez également compte des principes mentionnés dans la section Mode de coloration avec l'état UNDEFINED [Non défini] (à la page 34).)
   - Verrouillages (à la page 35)

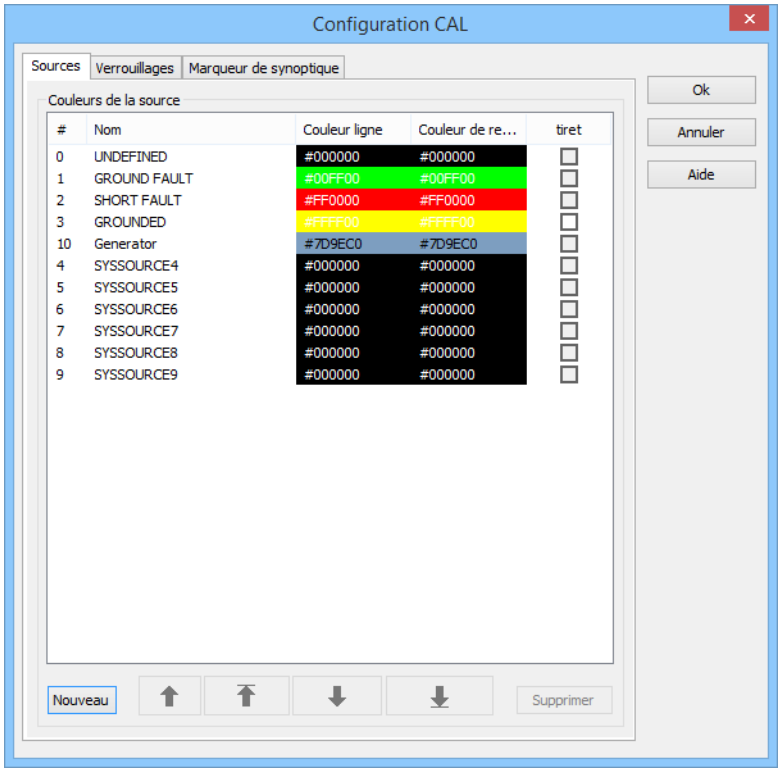
     Configurez les **verrouillages topologiques** dont le module de **gestion des commandes** doit tenir compte.
     Remarque : l'onglet est uniquement visible avec la licence de la suite Topology Package.
   - Marqueur de synoptique (à la page 40)

     Configurez la table des couleurs du marqueur de synoptique avec la fonction de **détection d'erreurs par impédance**.
     Remarque : l'onglet est uniquement visible avec la licence de la suite Topology Package.

3.2.1 Configuration des sources

La configuration des sources, c'est-à-dire leurs noms et leurs couleurs (ordre et priorité), s'effectue par rapport au projet, dans les propriétés du projet, sous Configuration CAL. Les sources possédant un identifiant compris entre 0 et 9 sont réservées pour les sources système. La fonction des sources qui possèdent déjà une fonction (par ex., mise à la terre, la couleur de la "source de terre") ne doit pas être modifiée. Les sources qui n'ont encore aucune fonctionnalité dans la version actuelle de zenon restent réservées pour des versions futures.

Les couleurs des sources avec un ID supérieur à 10 sont librement configurables pour les éléments procéduraux ; par exemple, pour la source "Générateur" ou "110 kV", etc. Ajoutez d'autres couleurs à cette fin.



#	Nom	Couleur ligne	Couleur de re...	tiret
0	UNDEFINED	#000000	#000000	☐
1	GROUND FAULT	#00FF00	#00FF00	☐
2	SHORT FAULT	#FF0000	#FF0000	☐
3	GROUNDED	#FFFF00	#FFFF00	☐
10	Generator	#7D9EC0	#7D9EC0	☐
4	SYSSOURCE4	#000000	#000000	☐
5	SYSSOURCE5	#000000	#000000	☐
6	SYSSOURCE6	#000000	#000000	☐
7	SYSSOURCE7	#000000	#000000	☐
8	SYSSOURCE8	#000000	#000000	☐
9	SYSSOURCE9	#000000	#000000	☐

COULEURS DE LA SOURCE

Paramètre	Description
Numéro	Numéro interne unique consécutif pour identifier la source. Ce numéro est donné par le système automatiquement et ne peut pas être modifié. Attention : Les ID 0 à 9 sont réservés aux sources système et ne doivent pas être utilisés de manière spécifique à l'utilisateur.
Nom	Nom logique de la source (par exemple, "Eau" ou "Terre"). Ce nom est également utilisé lors de la sélection du numéro de source pour les éléments combinés. Vous pouvez changer le nom en cliquant dessus avec le bouton gauche de la souris. Le mode édition est activé. Les modifications sont acceptées par la touche Entrée ou en sélectionnant une autre source. Remarque : Les intitulés suivants ne peuvent pas être changés selon la langue :
Couleur ligne	Couleur de ligne de la source correspondante. Cette couleur est utilisée pour colorer les lignes et les lignes brisées et définit la couleur pour l'extérieur des tubes.
Tiret	Cochez la case pour activer l'option. Si l'option est active, la ligne est dessinée en tirets. Remarque : Cette case peut uniquement être cochée pour l'option **GROUNDED**. Cette case à cocher est grisée pour toutes les autres sources.
Nouveau	Ajoute une nouvelle couleur.
Supprimer	Supprime la couleur sélectionnée.
Vers le haut (flèche)	Déplace la source sélectionnée d'une position vers le haut.
Complètement vers le haut (flèche)	Déplace la source sélectionnée vers le début de la liste.
Vers le bas (flèche)	Déplace la source sélectionnée d'une position vers le bas.
Complètement vers le bas (flèche)	Déplace la source sélectionnée vers la fin de la liste

FERMER BOÎTE DE DIALOGUE

Options	Description :
OK	Applique toutes les modifications effectuées sur tous les onglets, puis ferme la boîte de dialogue.
Annuler	Annule toutes les modifications effectuées sur tous les onglets, puis ferme la boîte de dialogue.
Aide	Ouvre l'aide en ligne.

Les couleurs peuvent être configurées directement en saisissant le code hexadécimal correspondant, ou à l'aide d'une palette de couleurs.

Pour effectuer une saisie directe :

1. Cliquez sur la description de la couleur avec le bouton gauche de la souris.

   Le champ bascule en mode Édition.
2. Saisissez le code.
3. Appuyez sur la touche `Entrée` ou sélectionnez une autre source pour appliquer la modification.

Pour effectuer une sélection via une palette de couleurs :

1. Sélectionnez la ligne de votre choix.
2. Cliquez sur le bouton **...** adjacent à la couleur.
   Remarque : Le bouton **...** est uniquement visible si l'entrée correspondant à la couleur est sélectionnée avec un clic.

   La palette de couleurs s'affiche dans un menu contextuel.
3. Sélectionnez la couleur de votre choix.

Le code hexadécimal décrit la couleur en mode RVB qui est constitué comme suit : **#RRGGBB**.

Produit	Signification
#	Identifiant indiquant qu'un code de couleur en hexadécimal est utilisé.
RR	2 chiffres représentant la valeur du rouge pour la couleur en hexadécimal. `0-255` correspond à `0-FF`.
GG	2 chiffres représentant la valeur du vert (green) pour la couleur en hexadécimal. `0-255` correspond à `0-FF`.
BB	2 chiffres représentant la valeur du bleu pour la couleur en hexadécimal. `0-255` correspond à `0-FF`.

Informations

La séquence dans la liste indique la priorité des sources, la première étant la plus prioritaire.

Pour modifier les priorités des sources uniques, vous pouvez les déplacer vers le haut ou vers le bas à l'aide des boutons fléchés.

Attention

Restrictions lors de la suppression de sources et de la réinitialisation de colorations incorrectes :

Les sources possédant un identifiant compris entre 0 et 9 sont réservées pour les sources système. Elles ne peuvent pas :

- Être supprimées
- Être réinitialisées comme une couleur incorrecte

Suppression de sources

Pour que des sources puissent être supprimées, elles doivent posséder un identifiant supérieur à 10. Seules les sources avec des numéros supérieurs peuvent être supprimées.

Réinitialisation de colorations incorrectes

SI vous réinitialisez les colorations incorrectes après avoir corrigé la source, vous ne pouvez pas utiliser de couleurs source système. Vous devez sélectionner une couleur dont l'identifiant est supérieur à 10.

Mode de coloration avec l'état UNDEFINED (Non défini)

La coloration du réseau peut être mise en œuvre de deux manières avec l'état UNDEFINED (Non défini) :

- Standard
- L'entrée est prioritaire

Ce paramètre est configuré par le biais des propriétés de projet **Coloration Automatique de Lignes/Mode pour la coloration**.

STANDARD

L'inspection du diagramme débute depuis une source et se poursuit sur l'ensemble du réseau, de telle manière que chaque interrupteur fermé (la variable d'interrupteur possède la valeur 1) dans chaque direction n'est parcouru qu'une fois, permettant d'éviter tout cycle. Chaque nœud visité (c'est-à-dire chaque segment de ligne) est coloré avec la couleur de la source. Les lignes directement connectées sont identifiées comme des nœuds.

Si la recherche découvre un interrupteur comportant une variable d'interrupteur avec l'état suivant, la couleur UNDEFINED est utilisée pour la coloration à partir de cet interrupteur :

- INVALID [valeurs aux choix]

- est non valide [valeur : 3]
- est dans une position intermédiaire [valeur : 2])

L'inspection du diagramme se poursuit maintenant sous la même forme. Chaque interrupteur est parcouru une fois dans chaque direction avec la couleur UNDEFINED. Par conséquent, chaque interrupteur peut être parcouru au maximum quatre fois par source :

1. Avec le numéro de source, vers l'avant
2. Avec le numéro de source, vers l'arrière
3. Avec le paramètre UNDEFINED, vers l'avant
4. Avec le paramètre UNDEFINED, vers l'arrière

L'ENTRÉE EST PRIORITAIRE

Avec le paramètre `L'alimentation est prioritaire`, seules les lignes dotées d'une alimentation provenant d'une source au moins, mais non alimentées par une source en particulier, sont colorées avec le paramètre UNDEFINED. Si une ligne est alimentée par une source au moins, elle ne peut plus recevoir la couleur UNDEFINED d'une autre source.

Cette recherche est une recherche en deux étapes :

- Lors de la première étape, à l'instar du paramètre `Standard`, la couleur de la source est distribuée sur le réseau depuis chaque source commutée, dans la mesure où l'interrupteur suivant est fermé. La recherche se termine si l'interrupteur est ouvert ou non valide/non défini.
- Lors de la deuxième étape, la recherche débute à chaque interrupteur non valide/non défini alimenté d'un côté, la couleur UNDEFINED étant distribuée au côté non alimenté. Cette recherche considère également que les interrupteurs non définis/non valides sont fermés, et distribue donc la couleur UNDEFINED sur le réseau jusqu'à trouver un interrupteur manifestement ouvert. En outre, une recherche se termine si un élément de ligne déjà alimenté est trouvé.

3.2.2 Configuration des verrouillages topologiques

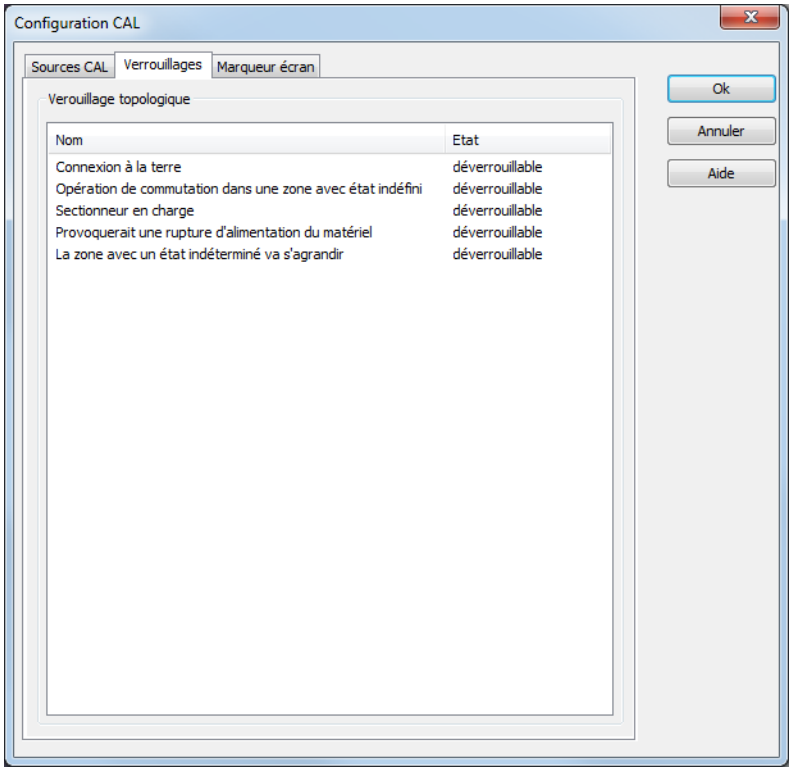
Dans le Runtime, les modules de **gestion des commandes** peuvent calculer les verrouillages résultant de l'état dynamique du réseau électrique, dont la topologie a été intégralement configurée à l'aide du module **CAL**. À l'aide de la topologie du réseau configurée avec le module CAL et des états actuels des sources (`MARCHE`/`ARRÊT`), interrupteurs, sectionneurs, etc., il est possible de détecter automatiquement l'**entrée de commande** qui entraînera l'exécution d'une commande (par exemple, "Tension à la terre"), puis d'interdire l'exécution de la commande.

Les verrouillages topologiques du module **CAL** pour la fonction d'**envoi de commande** sont configurés de manière centrale, pour le projet correspondant. Dans ce cas, une décision est également prise pour

déterminer si un utilisateur peut déverrouiller un verrouillage (à condition d'avoir le **niveau d'autorisation de déverrouillage** correspondant pour l'action).

Informations

*Cette boîte de dialogue est uniquement disponible si vous possédez les licences des modules Energy Edition et Coloration Automatique des Lignes (**Automatic Line Coloring**).*



Le paramétrage fait ici s'applique de façon globale sur le modèle topologique complet. Les conditions suivantes sont disponibles.

Paramètre	Description
Tension à la masse	Le verrouillage est actif, si un `disjoncteur` / `sectionneur` doit être fermé et qu'une "terre" est connectée à son premier connecteur et son autre connecteur est connecté ou indéfini.
Action de commutation dans une zone comportant un état non défini.	Le verrouillage est actif si un `disjoncteur` / `sectionneur` doit être fermé et ses deux connecteurs sont "non définis" ou "perturbés".
Sectionneur en charge	Le verrouillage est actif si certaines conditions d'activation ou de `désactivation` sont satisfaites. Conditions : voir la section Sectionneur en charge – Conditions de verrouillage (à la page 38).
Provoquerait une rupture d'alimentation du matériel	Le verrouillage est actif si un `interrupteur`/`sectionneur` doit être ouvert et si l'alimentation d'un appareil sous tension, alimenté par une tension provenant d'une `source` (c'est-à-dire un `drain`), est ensuite interrompue.
La zone avec un état indéterminé va s'agrandir	Le verrouillage est actif si un `disjoncteur` / `sectionneur` doit être fermé et un de ses connecteurs est "non défini" ou "perturbé" alors que l'autre ne l'est pas. Il est également signalé si la `commande` a été configurée avec une **direction de commutation** `Aucune`.

Si vous cliquez sur la colonne **État** pour l'un de ces verrouillages, vous obtenez une liste déroulante avec trois choix :

Paramètre	Description
Ne pas vérifier	La condition sélectionnée ne sera pas prise en compte dans ce projet (pour le modèle topologique).
Déverrouillable	La condition sélectionnée est considérée dans ce projet. Si la condition s'applique, l'utilisateur peut la déverrouiller avec la fonction de gestion de commande (dans un synoptique de type **Gestion de commande**). Cette action de déverrouillage est consignée dans la **liste d'événements (CEL)**.
Non déverrouillable	La condition sélectionnée est considérée dans ce projet. L'utilisateur ne peut pas la déverrouiller.

EXCEPTIONS CONCERNANT LE VERROUILLAGE TOPOLOGIQUE

Le verrouillage topologique n'est pas effectué si :

- La variable d'un interrupteur comporte l'état Révision
  Ou
- La variable est corrigée manuellement ou définie sur **Valeur de remplacement** et, dans le même temps, définie sur la même valeur de variable que la valeur initiale ; en d'autres termes, si l'interrupteur :

- Est défini sur OFF (Désactivé), puis corrigé manuellement sur OFF (Désactivé) ou s'il est remplacé.
- Est défini sur ON (Activé), puis corrigé manuellement sur ON (Activé), ou s'il est remplacé.

Sectionneur en charge - Conditions de verrouillage

Pour le verrouillage topologique **Sectionneur en charge**, un sectionneur peut être commuté si l'une des conditions suivantes est satisfaite pour les segments de ligne connectant les sectionneurs :

LORS DE L'ACTIVATION DU SECTIONNEUR :

Une vérification est effectuée pour déterminer si, avant l'activation de l'état MARCHE, la topologie se trouve dans l'un des états suivants :

- Les deux sections de ligne sont alimentées/reliées à la terre par la même source ;
- Une section de ligne ne reçoit aucune tension, l'autre section de ligne est reliée à la terre ;
- Une section de ligne n'est pas en charge.

LORS DE LA DÉSACTIVATION DU SECTIONNEUR :

Une vérification est effectuée pour déterminer si, avant l'activation de l'état ARRÊT, la topologie se trouve dans l'un des états suivants :

- Les deux sections de ligne sont alimentées par la même source ;
- Si une section de ligne ne reçoit plus de tension, l'autre section de ligne est reliée à la terre ;
- Une section de ligne n'est plus en charge.

Informations

Signification de "pas en charge"

L'état `pas en charge` signifie :

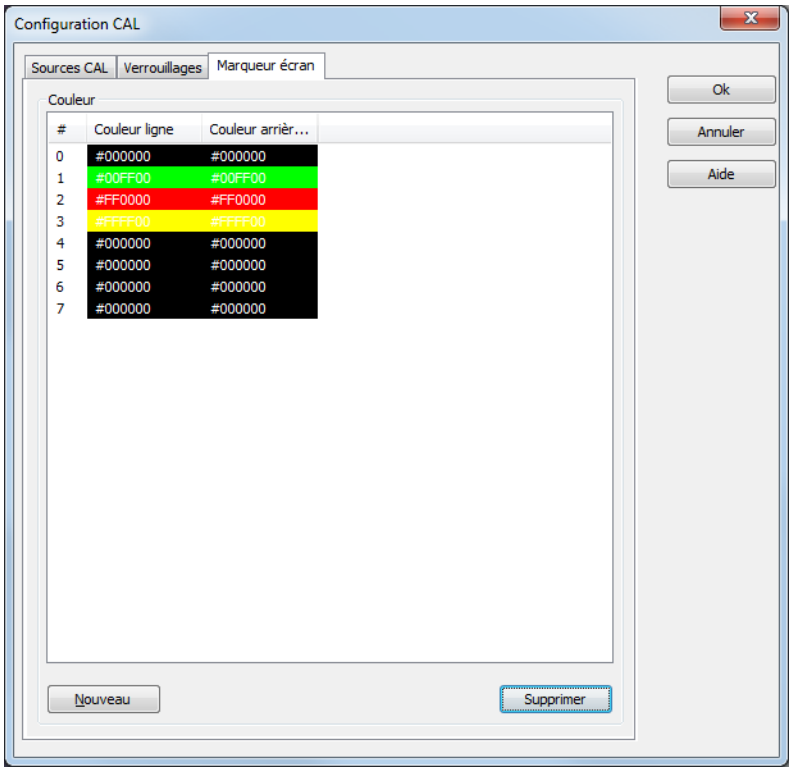
- Soit :
  Tous les interrupteurs et sectionneurs connectés à la section de ligne sont ouverts.
- Ou :
  Les interrupteurs et sectionneurs connectés à la section de ligne sont fermés, mais sont uniquement connectés à une section supplémentaire qui n'est pas en charge.

Par ailleurs, toutes les conditions suivantes doivent être satisfaites pour que l'état `pas en charge` soit vrai :

- L'ensemble des sources et consommateurs connectés au segment de ligne est désactivé.
- Aucun transformateur ne peut être connecté au segment de ligne.
- Il ne doit pas s'agir d'une ligne connectée uniquement à ce sectionneur (ligne ouverte).

3.2.3 Configuration du marqueur de synoptique

Ici, vous pouvez configurer la table des couleurs associée au marqueur de couleur pour la détection d'erreurs par impédance et le calcul de la répartition de charge (à la page 62). Voir également : **AddMarker**.



Paramètre	Description
Numéro	Numéro de série interne unique, garantissant une attribution claire. Ce numéro est donné par le système automatiquement et ne peut pas être modifié.
Couleur ligne	Couleur de ligne du marqueur de synoptique.
Couleur de remplissage	Couleur de remplissage du marqueur de synoptique.
Nouveau	Ajoute une nouvelle couleur.
Supprimer	Supprime la couleur sélectionnée. Remarque : Seule la dernière couleur de la liste peut être supprimée. Les couleurs standard ne peuvent pas être supprimées.

Les couleurs peuvent être configurées directement en saisissant le code hexadécimal correspondant, ou à l'aide d'une palette de couleurs.

Pour effectuer une saisie directe :

1. Cliquez sur la description de la couleur avec le bouton gauche de la souris.

   Le champ bascule en mode Édition.
2. Saisissez le code.
3. Appuyez sur la touche `Entrée` ou sélectionnez une autre source pour appliquer la modification.

Pour effectuer une sélection via une palette de couleurs :

1. Sélectionnez la ligne de votre choix.
2. Cliquez sur le bouton **...** adjacent à la couleur.
   Remarque : Le bouton **...** est uniquement visible si l'entrée correspondant à la couleur est sélectionnée avec un clic.

   La palette de couleurs s'affiche dans un menu contextuel.
3. Sélectionnez la couleur de votre choix.

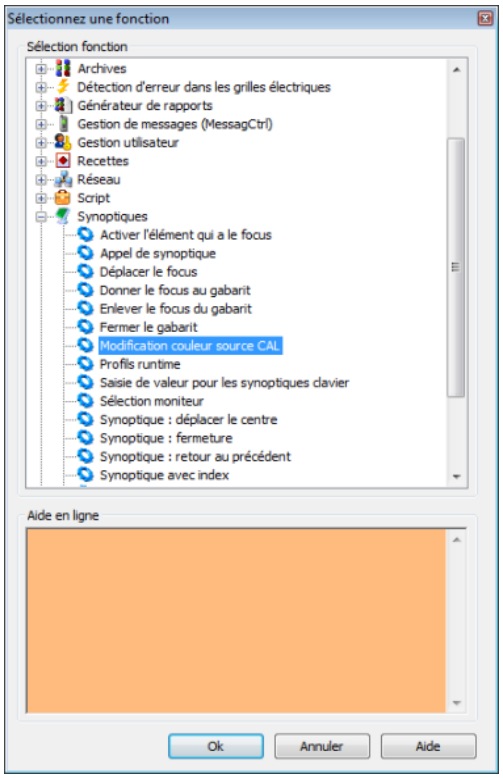
Le code hexadécimal décrit la couleur en mode RVB qui est constitué comme suit : **#RRGGBB**.

Produit	Signification
#	Identifiant indiquant qu'un code de couleur en hexadécimal est utilisé.
RR	2 chiffres représentant la valeur du rouge pour la couleur en hexadécimal. `0-255` correspond à `0-FF`.
GG	2 chiffres représentant la valeur du vert (green) pour la couleur en hexadécimal. `0-255` correspond à `0-FF`.
BB	2 chiffres représentant la valeur du bleu pour la couleur en hexadécimal. `0-255` correspond à `0-FF`.

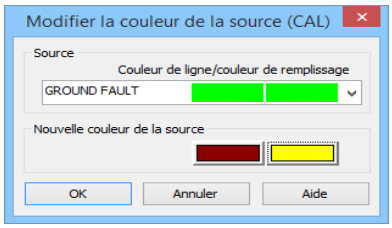
3.3 Fonction : Modifier la couleur de la source (CAL)

L'avant-plan et l'arrière-plan d'une source CAL peuvent être temporairement modifiés pour la coloration dans le Runtime à l'aide de la fonction **Modifier la couleur de la source (CAL)**. La modification reste active jusqu'à la fermeture ou l'actualisation du Runtime ou la nouvelle exécution de la fonction. Pour créer la fonction :

- Sélectionnez Nouvelle fonction
- Accédez au nœud Synoptiques.
- Sélectionnez **Modifier la couleur de la source (CAL)**.



- La boîte de dialogue de définition des couleurs de ligne et de remplissage s'affiche.
- Définissez la couleur souhaitée.



Propriété	Fonction
Source	Liste déroulante permettant de sélectionner la source et afficher les couleurs actuellement attribuées. Ces couleurs ne peuvent pas être modifiées ici.
Nouvelle couleur pour la source	Cliquez sur la couleur pour ouvrir une boîte de dialogue permettant de sélectionner une couleur.

3.4 Alias pour les synoptiques détaillés

Pour afficher des synoptiques individuels, une section partielle peut être sélectionnée dans le réseau topologique et affichée individuellement, au moyen d'alias. Les éléments du synoptique dans le synoptique de détail ne sont pas inclus dans le modèle topologique, reçoivent toutefois leurs couleurs CAL du modèle. Ces éléments sont liés à un alias des éléments du synoptique dans le synoptique global.

Attention

Les alias sont uniquement valides dans un projet.

Cela signifie, pour les symboles contenant des liens vers des alias :

*Si le symbole est ajouté à la **bibliothèque générale de symboles** ou à la **bibliothèque de projet global**, puis édité, toutes les informations concernant les alias de la fonction CAL sont perdues (aucune demande de confirmation n'est affichée).*

CRÉER UN ALIAS

Un alias peut être créé pour les éléments suivants :

- Ligne
- Ligne brisée
- Tube
- Élément combiné

Attention

Un alias de la fonction CAL ne peut pas être créé si le nom du synoptique sélectionné contient un point (**.**).

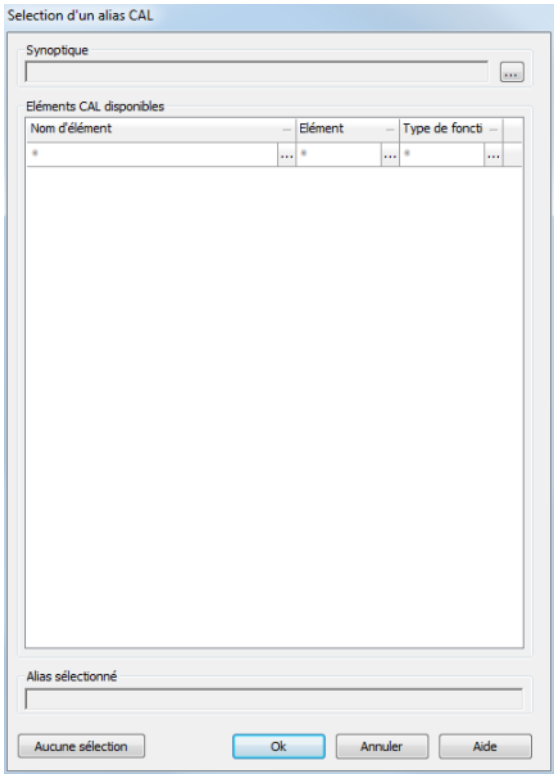
Solution : Remplacez le point dans le nom du synoptique par un caractère différent, tel qu'un tiret, par exemple (_).

Pour créer un élément source en tant qu'alias :

- Activez-le dans les propriétés **Utiliser alias** de l'élément.

  Pour cela, le module CAL doit comporter une licence et la propriété **Couleur définie par CAL** doit être active.

- Cliquez sur le bouton ... de la propriété **Alias**.
- La boîte de dialogue de sélection d'éléments s'affiche à l'écran.



Paramètre	Description
Synoptique	Cliquez sur le bouton ... pour ouvrir une boîte de dialogue permettant de sélectionner un synoptique.
Éléments CAL disponibles	Affiche les éléments appartenant à un synoptique avec le nom de l'élément, le type d'élément et le type de fonction. Cliquez sur un élément pour sélectionner un alias. **Filtre** Les éléments peuvent être triés en fonction de toutes les colonnes. Lors de la configuration d'un filtre, les options fournies par tous les autres filtres sont limitées aux valeurs pouvant être associées de manière logique. ▸ **Nom** : saisie d'une expression de recherche avec des opérateurs (*). Les 12 dernières expressions de recherche sont proposées dans la liste jusqu'à la fermeture d'Editor. ▸ **Élément** : Sélectionnez dans la liste déroulante. ▸ **Le type de fonction** : Sélectionnez dans la liste déroulante. Cliquez sur **...** pour ouvrir la liste des recherches enregistrées ou la liste déroulante. Si un filtre est actif, cliquez sur **X** pour le supprimer.
Alias sélectionné	Affiche l'élément sélectionné dans le champ des **éléments CAL disponibles**.
Aucune sélection	Supprime l'élément sélectionné.
OK	Enregistre la sélection et ferme la boîte de dialogue.
Annuler	Annule toutes les modifications et ferme la boîte de dialogue.
Aide	Ouvre l'aide en ligne.

Informations

Lors de la sélection d'un élément pour un nouvel alias, seuls les éléments et les synoptiques issus du même projet que celui pour lequel l'alias est défini peuvent être sélectionnés. Les éléments issus de sous-projets ou de projets parallèles ne sont pas disponibles.

REMPLACEMENT DES NOMS D'ALIAS

Les alias peuvent être remplacés lors d'appels de synoptiques avec la fonction Remplacer les liens. Un synoptique de détail peut donc être affiché avec les données provenant de différents composants d'équipements, tels que des lignes ou des réseaux partiels. Les noms d'alias sont remplacés de la même manière que les variables et les fonctions. Il est également possible de remplacer les éléments utilisés dans les symboles. La boîte de dialogue ouverte pour la cible est la même que pour la propriété **Alias**.

Remarque : la substitution à l'aide de variables d'indices n'est pas autorisée.

3.5 Localisation de défauts sur les réseaux électriques

La détection d'erreurs permet de repérer les sections de réseau sujettes à des défauts de terre ou des courts-circuits au moyen de couleurs spéciales dans le module CAL. Les points de départ de la détection sont appelés détecteurs de défaut de terre ou de court-circuit (par ex., un équipement de protection) et sont attribués à un disjoncteur. Les détecteurs de défaut de terre et de court-circuit se trouvent toujours à la sortie d'un élément disjoncteur. Pour cette raison, les variables correspondantes (contenant les rapports de l'équipement de protection) sont liées aux éléments CAL `Interrupteur`.

Les rapports provenant des équipements de protection sont représentés par une coloration spéciale, avec les couleurs de source ID 1 et 2, toutefois uniquement si le rapport est reçu pendant que les lignes sont actives. Dans le même temps, les rapports sont définis sur les mêmes variables aux fins de l'affichage. Les affichages d'erreurs graphiques peuvent ainsi être présentés sur l'écran, par exemple, avec d'autres éléments combinés qui sont uniquement visibles si un affichage est actif.

L'affichage doit être réinitialisé manuellement (acquitté) lorsque les équipements protecteurs ont rétracté les rapports.

> **Informations**
>
> *Cette fonction est uniquement disponible si vous possédez les licences des modules Energy Edition et Coloration Automatique des Lignes.*

DÉTECTION D'ERREURS

La détection d'erreurs s'exécute localement sur chaque ordinateur présent sur le réseau zenon. Chaque client présent sur le réseau possède un modèle spécifique indépendant et peut donc rechercher les défauts de terre et les courts-circuits dans différentes sections de la topologie.

La détection d'erreurs sur le réseau électrique comporte plusieurs phases :

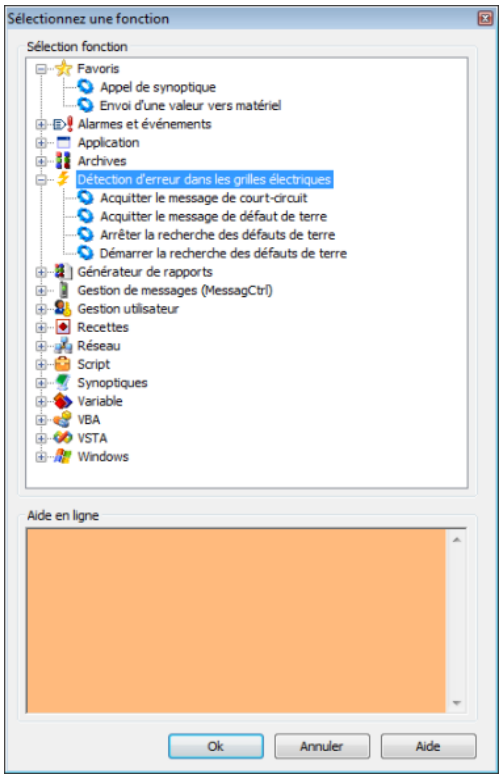
- Rechercher un défaut de terre (à la page 48)
- Recherche de court-circuit (à la page 54)

Pour configurer la détection d'erreur :

- Vous avez besoin d'une licence pour le module CAL et le module Energy Edition de zenon.
- Configurez les synoptiques adéquats.
- Configurez (à la page 10) le module CAL en fonction des éléments combinés avec la fonction `interrupteur`
- Configurez (à la page 22) les lignes afin qu'elles soient colorées par le module CAL.

Le Runtime comporte des fonctions spéciales pour la détection d'erreurs :

- Démarrer la recherche des défauts de terre (à la page 51)
- Acquitter (à la page 52) un message de défaut de terre (à la page 52)
- Arrêter la recherche des défauts de terre (à la page 53)
- Acquitter le message de court-circuit (à la page 57)



COLORATIONS

Les erreurs peuvent être affichés avec une coloration spéciale des lignes dans le module CAL si les notifications sont reçues lorsque les lignes sont actives. Dans le Runtime, la couleur attribuée par le module CAL change automatiquement dès que l'état de la ligne change. Les colorations configurées peuvent être modifiées dans le Runtime via la fonction Modifier la couleur de la fonction CAL (à la page 42).

Les messages sont traités dans leur ordre d'arrivée. En cas de conflit :

- Les couleurs d'affichage des erreurs sont prioritaires.
- Les messages de court-circuit sont prioritaires sur les messages de défaut de terre.

3.5.1 Rechercher un défaut de terre

La recherche de défaut de terre permet de signaler les sections du réseau présentant potentiellement un défaut de terre en les colorant. La couleur est définie par la configuration des couleurs source du module CAL (à la page 30) pour la source `GROUND FAULT`, ID 1 (DÉFAUT DE TERRE). Dans le même temps, les notifications sont définies sur les variables supplémentaires aux fins de l'**affichage** graphique.

Les sections du réseau susceptibles de présenter un défaut de terre sont dérivées des rapports de défaut de terre fournis par les équipements de détection de défauts de terre (indicateurs de défaut de terre ou équipements de protection enregistrant les défauts de terre). Les considérations suivantes s'appliquent aux défauts de terre :

- Chaque équipement peut présenter un à trois messages de défaut de terre.
- Les rapports de défaut de terre peuvent être gérés de deux manières : par le traitement de messages permanents ou le traitement de messages provisoires.
- Pour les équipements de détection de défauts de terre directionnels, la direction peut être devant ou derrière, en fonction du déclenchement.
  - Devant : D'abord, la notification de la direction arrive en **amont** et/ou en **aval**, puis via le **déclencheur**.
  - Derrière : Le **déclencheur** arrive avant la solution.

> **Informations**
>
> *Si cette connexion est établie avec succès, un composant réseau présentant potentiellement un défaut de terre n'est alors plus considéré comme défaillant.*

DÉVELOPPEMENT

Pour configurer une recherche de défaut de terre :

1. Attribuez l'élément combiné représentant l'élément de commutation à l'interrupteur (à la page 49) de l'élément **Type de fonction**.
2. Définissez le mode de recherche de défaut de terre (à la page 49), le déclencheur de défaut de terre (à la page 51) et l'affichage du défaut de terre (à la page 50).
3. Créez les fonctions de début de recherche de défaut de terre (à la page 51), d'acquittement du rapport de défaut de terre (à la page 52) et de fin de recherche de défaut de terre (à la page 53).

> **Informations**
>
> *Pour pouvoir également limiter la détection de défauts de terre sur les réseaux hétérogènes, seul un emplacement comportant des défauts de terre est analysé sur chaque chemin, en commençant par une source.*

Mode de recherche pour les défauts de terre

La fonction de recherche de court-circuit peut soit :

- Colorer la portion du réseau comportant potentiellement un court-circuit,
  ou
- L'ensemble du réseau sur lequel a été détecté le court-circuit.

Le mode de coloration est défini via la propriété **Mode de recherche pour les défauts de terre**.

Pour configurer la propriété :

- Accédez au nœud **Coloration Automatique de Lignes** dans les propriétés.
- Sélectionnez le mode de votre choix dans la liste déroulante de la propriété **Mode de recherche pour les défauts de terre**.
  - Colorer la partie du réseau : colore uniquement les parties du réseau comportant potentiellement un court-circuit.
  - Colorer l'ensemble du réseau : colore l'ensemble du réseau lié sur lequel a été détecté le court-circuit.

Ce paramètre peut être modifié dans le Runtime via le modèle d'objet API de zenon. Dans ce cas, la recherche de court-circuit est recalculée à nouveau.

Type de détection de défaut de terre

La direction et le type de traitement de message pour l'élément combiné sont déterminés par le paramètre **Type**. Pour la configuration de projets :

1. Accédez au nœud **Coloration Automatique de Lignes** dans les propriétés de l'élément combiné.
2. Ouvrez le nœud **Détection des défauts de terre**.
3. Sélectionnez le type souhaité, ainsi que la direction et le type de traitement de message, dans la propriété **Type**.
   - Sens de défilement :
     Indique si le front montant de l'alarme de déclenchement est prioritaire ou non sur le front montant d'une direction.
   - Devant : L'état de la direction actuelle est utilisé pour le front montant de l'alarme de déclenchement.
   - Derrière : après un front montant de l'alarme de déclenchement, le premier front montant d'une direction est attendu ; s'il ne se produit pas dans les deux secondes suivantes, l'équipement de détection de défaut de terre est considéré comme non directionnel.
   - Traitement de l'information :
     États dans lesquels les informations peuvent être traitées.
   - Aucun : interrupteur normal ; les informations ne sont pas traitées.

- Traitement des messages permanents : les nouveaux messages reçus sont considérés comme un nouveau déclenchement de défaut de terre.
- Traitement des messages transitoires : les messages reçus durant une recherche (à la page 51) de courant sont supprimés.

Remarque : La distinction entre le traitement des messages permanents et le traitement des messages transitoires a uniquement trait au traitement du message, et non au type de message. Il n'est donc pas nécessaire que le traitement des messages transitoires concerne un bit transitoire.

Attention

Pour supprimer les défauts de terre intermittents, les messages de défaut de terre reçus à intervalles de moins de 2 secondes sont ignorés.

Affichage des défauts de terre

La variable liée à la propriété **Afficher** est une variable de sortie pour la détection d'erreurs, et affiche l'état enregistré de l'équipement de détection de défaut de terre. Ceci est indispensable, car tous les messages sont enregistrés au niveau interne jusqu'à leur acquittement, c'est-à-dire qu'ils ne sont pas forcément conformes à l'état actuel des variables de message.

Chaque fois qu'un enregistrement est réalisé, une valeur prescrite est transmise à cette variable. Dans ce cas, les valeurs sont les suivantes :

Valeur	Signification
0	Pas de défaut de terre
1	Défaut de terre devant
2	Défaut de terre derrière
3	Défaut de terre non directionnel
4	État d'erreur -> Les deux directions ont été activées

Informations

Pour réduire les problèmes liés au fonctionnement du réseau, la variable liée ici doit être une variable liée.

Déclenchement de défauts de terre

L'alarme de détection de défaut de terre est définie par la variable **Déclenchement**. Elle peut contenir des informations concernant la présence d'un défaut de terre et la direction du défaut de terre selon l'équipement de détection de défaut de terre. Dans ce cas, la distinction suivante est faite :

- Alarmes de défaut de terre non directionnel
- Alarmes de défaut de terre directionnel avec une alarme de déclenchement
- Alarmes de défaut de terre directionnel avec une alarme de déclenchement

Pour configurer la variable pour l'élément **Déclenchement** :

1. Accédez au nœud **Coloration Automatique de Lignes** dans les propriétés de l'élément combiné.
2. Ouvrez le nœud **Détection des défauts de terre**.
   a) Pour les alarmes de défaut de terre non directionnelles :

      Cliquez sur le bouton ... de la propriété **Déclenchement**.

      Sélectionnez la variable que vous souhaitez importer dans la boîte de dialogue qui s'affiche à l'écran.

      Les propriétés de la direction restent vides.

   b) Pour les alarmes de défaut de terre directionnel avec une alarme de déclenchement :

      Liez la variable à **Déclenchement** et ajoutez la direction adéquate :

      En avant : liez une variable à la propriété **Avancer**.

      En arrière : liez une variable à la propriété **Reculer**.

   c) Pour les alarmes de défaut de terre directionnel sans une alarme de déclenchement :

      Liez la variable à la direction correspondante :

      En avant : liez une variable à la propriété **Avancer**.

      En arrière : liez une variable à la propriété **Reculer**.

      La propriété **Déclenchement** reste vide.

Remarque : si vous adressez un équipement d'identification directionnel avec la propriété **Avancer** dans les deux directions, cette configuration est considérée comme erronée et est ignorée.

Démarrer la recherche des défauts de terre

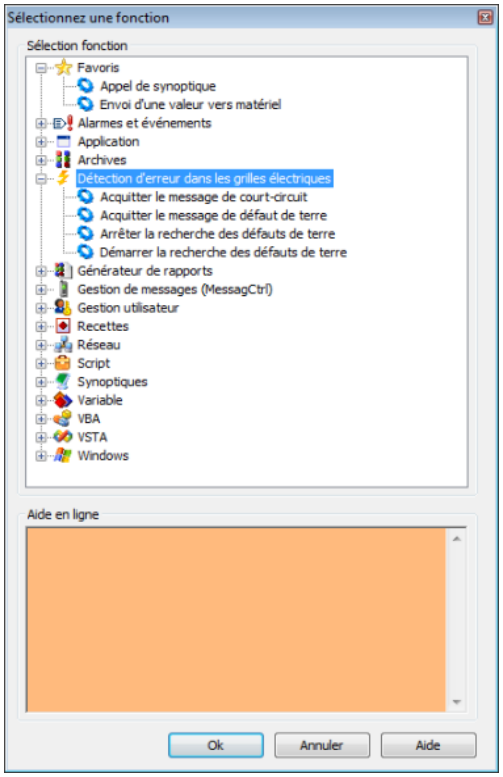
La fonction **Démarrer la recherche des défauts de terre** permet de localiser les défauts de terre et a deux effets dans le Runtime :

1. Les messages provenant de tous les équipements de détection de défaut de terre configurés avec l'option Traitement des messages fluctuants sont ignorés.

2. L'algorithme de recherche est modifié : les actions de commutation peuvent uniquement restreindre davantage la section affectée par un défaut de terre. Les nouveaux messages reçus n'augmentent donc pas l'étendue de la section potentiellement affectée par un défaut de terre.

Pour configurer la fonction **Démarrer la recherche des défauts de terre** :

- Créez une nouvelle fonction.
- Accédez au nœud de détection d'erreur du réseau électrique.
- Sélectionnez la fonction **Démarrer la recherche des défauts de terre**.



- Liez la fonction à un bouton.

Acquitter le message de défaut de terre

La fonction **Acquitter le message de défaut de terre** permet d'acquitter un message de défaut de terre enregistré au niveau interne provenant d'un équipement de détection de défaut de terre. De cette manière, l'état de défaut de terre verrouillé au niveau interne est réinitialisé si l'état est toujours en attente, ou défini comme étant acquitté. Un message de défaut de terre enregistré sera uniquement supprimé au niveau interne s'il a été acquitté et n'est plus en attente.

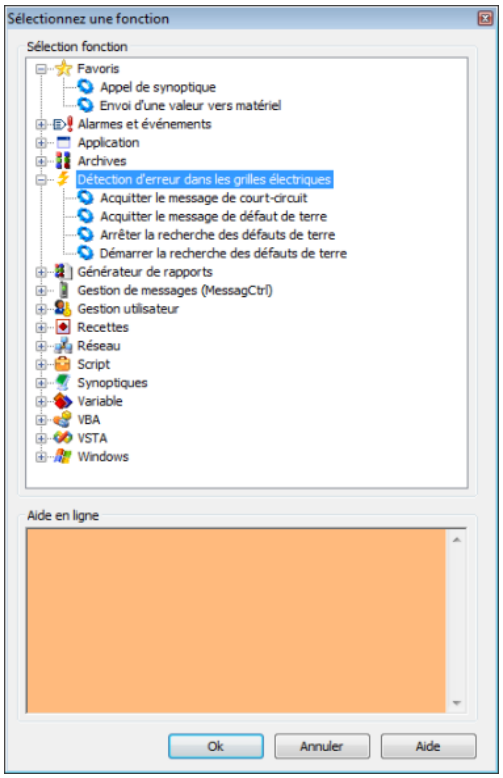
Règles de l'acquittement :

- si une variable correspondant à une variable de déclenchement ou de direction d'un équipement de détection de défaut de terre est liée, ce message de défaut de terre spécial est acquitté.

- Si aucune variable n'a été liée, tous les messages de défaut de terre sont acquittés.
- L'acquittement peut également être effectué via le modèle d'objet de l'API de zenon.

Pour configurer la fonction **Acquitter le message de défaut de terre** :

- Créez une nouvelle fonction.
- Accédez au nœud de détection d'erreur du réseau électrique.
- Sélectionnez la fonction **Acquitter le message de défaut de terre**.



- La boîte de dialogue de sélection de variable s'affiche à l'écran.
- Liez la variable souhaitée à la fonction.
- Liez la fonction à un bouton.

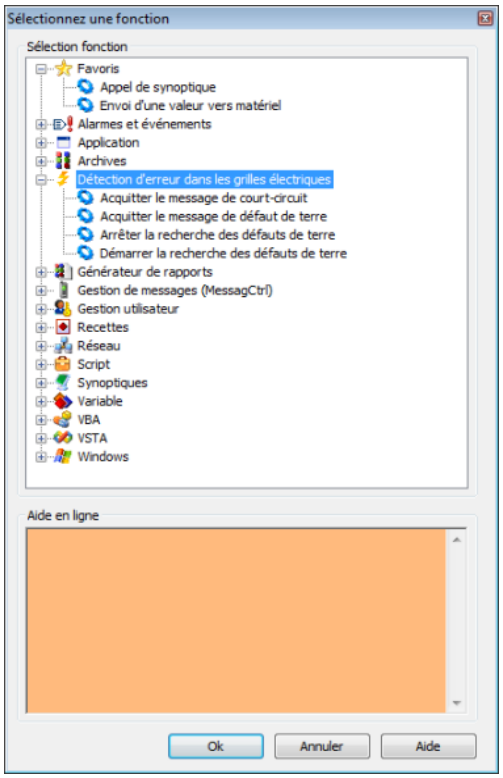
Arrêter la recherche des défauts de terre

Pour arrêter la recherche de défaut de terre, exécutez la fonction **Arrêter la recherche des défauts de terre** dans le Runtime.

Pour configurer la fonction :

- Créez une nouvelle fonction.
- Accédez au nœud de détection d'erreur du réseau électrique.

- Sélectionnez la fonction **Arrêter la recherche des défauts de terre**.



- Liez la fonction à un bouton.

3.5.2 Recherche de court-circuit

La recherche de court-circuit permet de signaler les sections du réseau présentant potentiellement un court-circuit en les colorant. La couleur est définie par la configuration des couleurs source du module CAL pour la source `DÉFAUT DE COURT-CIRCUIT`.

Les composants du réseau potentiellement sujets aux courts-circuits sont déterminés en fonction des rapports de court-circuit. Un équipement d'identification de court-circuit (indicateur de court-circuit, équipement de protection) peut comporter un à trois messages de court-circuit. Pour les équipements de détection de court-circuit directionnel, la direction peut être en avant ou en arrière, en fonction du déclenchement. Si cette connexion est établie avec succès, un composant réseau présentant potentiellement un court-circuit n'est alors plus considéré comme défaillant.

DÉVELOPPEMENT

Pour configurer la recherche de court-circuit :

1. Attribuez l'élément combiné représentant l'élément de commutation à l'interrupteur (à la page 55) de l'élément **Type de fonction**.

2. Définissez l'affichage des défauts de terre (à la page 55) et le déclenchement de la détection de défauts de terre (à la page 56).
3. Configurez la fonction d'acquittement des messages de défaut de terre (à la page 57).

Détection de défauts de terre

La direction et le type de traitement de message pour l'élément combiné sont déterminés par le paramètre **Type**. Pour la configuration de projets :

1. Accédez au nœud **Coloration Automatique de Lignes** dans les propriétés de l'élément combiné.
2. Ouvrez le nœud **Détection cout-circuit**.
3. Sélectionnez le type souhaité dans la propriété **Type**.
   - `Direction:`
     Indique si le front montant de l'alarme de déclenchement est prioritaire ou non sur le front montant d'une direction.
   - `Devant :`
     L'état de la direction actuelle est utilisé pour le front montant de l'alarme de déclenchement.
   - `Derrière :`
     après un front montant de l'alarme de déclenchement, le premier front montant d'une direction est attendu ; s'il ne se produit pas dans les deux secondes suivantes, l'équipement de détection de court-circuit est considéré comme non directionnel.
   - `Traitement de l'information :`
     États dans lesquels les informations peuvent être traitées.
   - `aucune :`
     interrupteur normal ; les informations ne sont pas traitées.
   - `Traitement des messages permanents :`
     Les nouveaux messages reçus sont considérés comme un nouveau déclenchement de défaut de terre.

Affichage des défauts de terre

La variable liée à la propriété **Afficher** est une variable de sortie pour la détection d'erreurs, et affiche l'état enregistré de l'équipement de détection de défaut de terre. Ceci est indispensable, car tous les messages sont enregistrés au niveau interne jusqu'à leur acquittement, c'est-à-dire qu'ils ne sont pas forcément conformes à l'état actuel des variables de message.

Chaque fois qu'un enregistrement est réalisé, une valeur prescrite est transmise à cette variable. Dans ce cas, les valeurs sont les suivantes :

Valeur	Signification
0	Pas de court-circuit
1	Court-circuit devant
2	Court-circuit derrière
3	Court-circuit non directionnel

Déclenchement de la détection de défaut de terre

La variable du message provenant de l'équipement d'identification de court-circuit est définie par la variable **Déclenchement**. Il peut contenir des informations concernant la présence d'un court-circuit et la direction du court-circuit selon l'équipement de détection de défaut de terre. Dans ce cas, la distinction suivante est faite : Dans ce cas, la distinction suivante est faite :

- Détecteur de court-circuit non directionnel
- Détecteur de court-circuit directionnel avec alarme de déclenchement
- Détecteur de court-circuit directionnel sans alarme de déclenchement

Pour configurer les variables pour chaque élément :

1. Accédez au nœud **Coloration Automatique de Lignes** dans les propriétés de l'élément combiné.
2. Ouvrez le nœud **Détection cout-circuit**.
   a) Pour les détecteurs de court-circuit non directionnel

      Cliquez sur le bouton ... de la propriété **Déclenchement**.

      Sélectionnez la variable que vous souhaitez importer dans la boîte de dialogue qui s'affiche à l'écran.

      Les propriétés de la direction restent vides.

   b) Pour les détecteurs de court-circuit directionnel avec alarme de déclenchement

      Liez la variable à **Déclenchement** et ajoutez la direction adéquate :

      En avant : liez une variable à la propriété **Avancer**.

      En arrière : liez une variable à la propriété **Reculer**.

   c) Pour les détecteurs de court-circuit directionnel sans alarme de déclenchement

      Liez la variable à la direction correspondante :

      En avant : liez une variable à la propriété **Avancer**.

      En arrière : liez une variable à la propriété **Reculer**.

      La propriété **Déclenchement** reste vide.

Acquitter le message de court-circuit

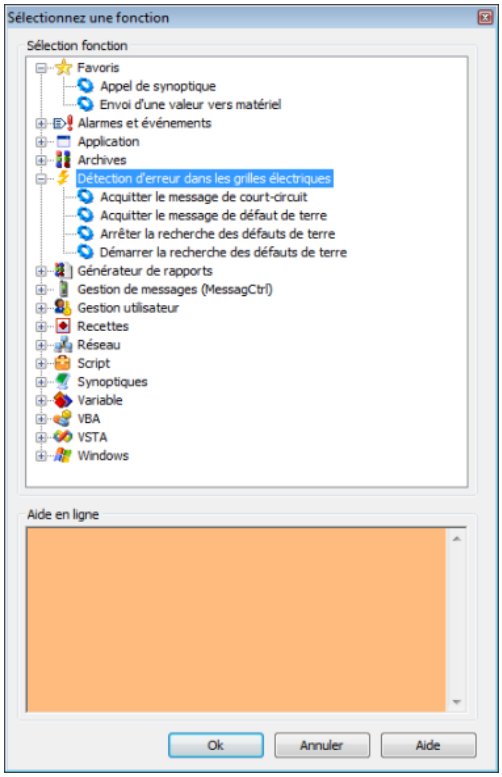
La fonction **Acquitter le message de court-circuit** permet d'acquitter un message de court-circuit enregistré au niveau interne provenant d'un équipement de détection de court-circuit. De cette manière, l'état de défaut de terre verrouillé au niveau interne est réinitialisé si l'état est toujours en attente, ou défini comme étant acquitté. Un message de court-circuit enregistré sera uniquement supprimé au niveau interne s'il a été acquitté et n'est plus en attente.

Règles de l'acquittement :

- si une variable correspondant à une variable de déclenchement ou de direction d'un équipement de détection de court-circuit est liée, ce message de court-circuit spécial est acquitté.
- Si aucune variable n'a été liée, tous les messages de court-circuit sont acquittés.
- L'acquittement peut également être effectué via le modèle d'objet de l'API de zenon.

POUR CONFIGURER LA FONCTION ACQUITTER LE MESSAGE DE COURT-CIRCUIT **:**

- Créez une nouvelle fonction.
- Accédez au nœud de détection d'erreur du réseau électrique.
- Sélectionnez la fonction **Acquitter le message de court-circuit**.



- Sélectionnez la variable que vous souhaitez importer dans la boîte de dialogue qui s'affiche à l'écran.

- Liez la fonction à un bouton.

3.6 Détection des erreurs par impédance et calcul de la répartition de charge

La détection des erreurs par impédance et le calcul de la répartition de charge étendent la fonction CAL. La fonction CAL identifie les nœuds et les rayons, tandis que ce modèle détecte également les lignes et leurs paramètres. La localisation des défauts à partir de l'équipement de protection est possible grâce à la configuration dans zenon Editor.

Le modèle fournit des propriétés et des méthodes d'évaluation externe de la localisation de défauts et de la répartition de charges, par le biais d'une API.

PROPRIÉTÉS DE LA FONCTION CAL ET DU MODÈLE TOPOLOGIQUE ÉTENDU

Les éléments **Élément combiné** et **Ligne** (ligne, ligne brisée, tube) de la CAL possèdent des propriétés spéciales dédiées à la détection d'erreurs par impédance et le calcul de la répartition de charge. Les propriétés de répartition de charge ne sont pas évaluées dans zenon, mais sont disponibles par l'intermédiaire d'algorithmes créés par les utilisateurs avec l'API de zenon.

Le modèle topologique utilisé pour la coloration est complété par un modèle topologique étendu qui inclut toutes les lignes sous forme de rayons séparés. Le modèle topologique étendu est enregistré sous le nom **ALC.xml** et peut ainsi être lu par les applications externes. **ALC.xml** contient deux sections :

- **GraphElements**:
  contient le modèle topologique étendu, sans alias
- **GraphAliases**:
  contient uniquement les alias

3.6.1 Localisation de défauts par impédance pour le court-circuit

Avec la localisation de défauts par impédance, un marqueur d'erreur est défini à l'emplacement du problème dans la topologie. Les valeurs d'impédance mesurées par les équipements de protection sont évaluées par le module **CAL**. En fonction de la topologie, les marqueurs d'erreur sont correctement positionnés sur un synoptique zenon.

En cas de court-circuit, si la réactance n'est pas égale à zéro, la procédure de localisation du court-circuit démarre :

- Court-circuit :
  Signalé par une variable liée pour la propriété **Déclenchement**

- Réactance :
  Valeur de la variable (issue du type de données REAL) liée à la propriété **Valeur de réactance de la protection**.

POSITION DU MARQUEUR

Toutes les lignes sont parcourues dans la direction correspondante. La direction résulte d'une réactance négative ou positive. La réactance correspondante de la ligne parcourue est déduite, et la recherche continue jusqu'à ce que la réactance résiduelle soit inférieure à la réactance de la ligne suivante. Un marqueur est tracé sur la ligne. La position du marqueur correspond à la réactance résiduelle.

Si aucune valeur de réactance n'est présente, aucun marqueur n'est défini en cas de notification de court-circuit. Pour que le marqueur soit tracé correctement, la zone ne doit pas être en charge au moment de la notification de court-circuit. En présence de notifications de court-circuit retardées, la réactance est uniquement évaluée en cas de réception de la notification de direction ou d'expiration du délai d'attente de 2 secondes.

La recherche est annulée en cas de détection d'un élément ouvert ou d'un autre élément CAL. Chaque composante du réseau et chaque ligne individuelle qu'il contient doit s'exécuter une fois seulement par déclencheur ; le nombre de marqueurs présents sur le réseau de lignes est donc inférieur au nombre possible.

Lors du rechargement, les marqueurs déjà existants sont dessinés au même point qu'avant le rechargement. Les modifications apportées à la configuration de la localisation de défauts sont uniquement évaluées après un autre court-circuit.

En cas d'effacement et d'acquittement d'une notification de court-circuit, tous les marqueurs de ce déclencheur de court-circuit sont supprimés.

Remarque : En fonction de l'ordre de rectification du court-circuit et du rétablissement du fonctionnement, le marqueur peut rester visible, bien que la ligne ne soit plus colorée comme un court-circuit.

Configuration dans Editor

Avec la localisation de défauts par impédance, un marqueur d'erreur est défini à l'emplacement du problème dans la topologie. La localisation est calculée à partir de l'impédance, sur la base de la topologie.

Pour configurer la localisation de défauts par impédance dans zenon Editor, effectuez les opérations suivantes :

1. Activez la localisation de défauts par impédance :
   a) Pour cela, cliquez sur le projet dans votre **Espace de travail**.
   b) Cliquez sur le groupe de propriétés **Coloration Automatique de Lignes** du projet.
   c) Activez la propriété **Localisation de défaut basée sur l'impédance**.

2. Configurez l'affichage des marqueurs d'écran avec les propriétés de projet particulières :
   a) **Taille marqueur écran**
   b) **Epaisseur de ligne du marqueur écran**
   c) **Affiche le type de marqueur écran**
3. Créez un synoptique zenon.
4. Positionnez l'**élément combiné** sur le synoptique zenon.
   La boîte de dialogue de sélection de variables s'affiche.
5. Configurez les paramètres du module CAL pour l'élément combiné :
   a) Assurez-vous que l'élément combiné a été sélectionné.
   b) Accédez au groupe de propriétés **Coloration Automatique de Lignes**.
   c) Dans la propriété **Type de fonction**, sélectionnez l'entrée `Commutation` dans la liste déroulante.
   d) Liez la propriété **Valeur de réactance de la protection** (dans la section **Localisation défaut de la protection / distribution charge** des propriétés) à une variable du type de données `REAL` avec la valeur de l'impédance mesurée.
   e) Sélectionnez le type de **Détection cout-circuit** dans la liste déroulante de la propriété **Type**.
   f) Configurez la couleur du marqueur dans la propriété **Couleur du marqueur**.

3.6.2 Modèle topologique étendu

Chaque objet possède un ID unique, grâce auquel il est référencé dans le fichier. Les attributs correspondent à un sous-ensemble des éléments de synoptique zenon ayant créé les éléments.

GRAPHELEMENT

ID	Description
Picture	Nom du synoptique
ElementID	ID d'élément de synoptique
ElementRef	Référence d'élément de synoptique
Type	Type d'élément de synoptique (voir Élément)
SourceID	Numéro de source
ReverseSourceID	Numéro de source inversé
Variable	Variable d'état
VarProtReact	Variable de réactance
MaxIType	Type d'intensité maximale
MaxIVal	Valeur constante d'intensité maximale
VarMaxI	Variable d'intensité maximale
VarCurI	Variable d'intensité instantanée
VarCalcI	Variable d'intensité calculée
VarCurP	Variable de puissance instantanée
LoadType	Type de charge
LoadVal	Valeur constante de charge
VarLoad	Variable Charge
React	Réactance
Resist	Résistance
Length	Longueur de ligne
Node1IDs	Liste de tous les ID d'éléments connectés au nœud Node1
Node2IDs	Liste de tous les ID d'éléments connectés au nœud Node2

GRAPHALIAS

ID	Description
Picture	Nom du synoptique
ElementID	ID d'élément de synoptique
ElementRef	Référence d'élément de synoptique
Type	Type d'élément de synoptique (voir Élément)

OrigElemRef	Élément de synoptique - référence à l'élément de synoptique original
OrigGraphElemID	ID des éléments originaux dans GraphElements

3.6.3 API

Dans le modèle d'objet de l'API de zenon, les objets **ALCGraphElement** et **ALCGraphAlias** sont disponibles pour le modèle. Ils contiennent les mêmes informations que le fichier XML. Ces objets sont accessibles dans le moteur CAL via les fonctions :

- **GraphElemCount()**
- **GraphAliasCount()**
- **GraphElemItem()**
- **GraphAliasItem()**

VERROUILLAGES TOPOLOGIQUES SPÉCIFIQUES À L'UTILISATEUR

Si un verrouillage topologique est activé, l'événement suivant est appelé dans le moteur CAL :

- **void CheckInterlocking(IALCEdge* pALCEdge, long nNewState, tpLockResult* LockResult, BSTR* bsText, VARIANT_BOOL* bUnlockable);**

L'interrupteur/le sectionneur à commuter et le nouvel état sont transférés. L'événement peut renseigner **LockResult**, **bUnlockable** et **bsText** pour afficher une violation de condition de verrouillage. Si le gestionnaire d'événements renvoie **tpBusy** dans **LockResult**, il est interrogé jusqu'à ce qu'il ne renvoie plus la réponse **tpBusy**, pendant `10 secondes` maximum toutefois. Le verrouillage est actif après `10 secondes`. Le texte de verrouillage et l'impossibilité du déverrouillage sont indiqués dans **bsText** et **bUnlockable**.

MARQUEUR DE SYNOPTIQUE

Des éléments de marqueur peuvent être insérés dans le synoptique via l'API de zenon. Les éléments de marqueur suivants sont disponibles pour les éléments suivants :

- Ligne
- Ligne brisée
- Tube

Ces éléments sont ajoutés ou supprimés, via les fonctions de l'API, dans **DynPictures** :

- **BSTR AddMarker(BSTR bsScreenName, long nElementID, short nPosition, short nLineColorIndex, short nFillColorIndex);**
- **VARIANT_BOOL DelMarker(BSTR bsID);**

Le GUID du marqueur, fourni par AddMarker(), identifie le marqueur de façon unique et sert à la fois de nom d'élément (avec le préfixe **$MARKER**_), ainsi que de clé la fonction de suppression, via DelMarker(). Les marqueurs insérés via l'API sont enregistrés dans le projet, en fonction du synoptique. **Attention :** l'enregistrement n'est pas rémanent ; il dure uniquement jusqu'au redémarrage du Runtime.

Les marqueurs définis ici sont affichés indépendamment du moniteur sur lequel est affiché le synoptique. Au niveau interne, les marqueurs sont traités comme des éléments de synoptique actionnables normaux : Des événements de souris sont invoqués à cette fin.

L'apparence des marqueurs est déterminée dans les paramètres du projet, dans la section **Coloration Automatique de Lignes** de la configuration du projet :

- **Affiche le type de marqueur écran** : triangle, cercle, carré, croix
- **Taille marqueur écran** : Taille en pixels :
- **Epaisseur de ligne du marqueur écran** : Largeur en pixels
- Couleur du marqueur : ce paramètre est défini par l'intermédiaire de l'indice dans la table de couleurs du marqueur (à la page 40), située dans les propriétés des éléments de synoptique, dans le groupe **Coloration Automatique de Lignes**

4. Command Processing

Command processing serves primarily for the secured switching of variables in energy technology. 'Secured' means that there is a check whether the switching operation is allowed, according to the configured interlocking condition and the dynamically updated topology (current physical state of the topological network). The configuration of the topology and the topological commands is done via the ALC (Automatic Line Coloring) (à la page 7) module.

Note: You can find step-by-step instructions for the creation of a configuration of simple **command processing** in the Project configuration in the Editor (à la page 69) chapter.

> **Informations concernant la licence**
>
> *Included in the license for Energy Edition.*
>
> *The functions documented in this manual are only available if zenon Energy Edition is licensed.*

Command groups always contain a set of defined actions, which are usually adjusted to a specific data point (a specific device) . For example, different command groups can be defined individually and centrally for different topological elements (switch/disconnector etc.) .

A data point for the command processing always consists of 2 physical variables:

- Response variable:
  The response variable is defined centrally for the whole command group. It represents the status of the topological element, for example whether the switch is open or closed.
- Command variable:
  A defined command variable is assigned to every action inside a command group. The driver uses this variable to write commands to the controller.

Depending on the action to be executed, these commands are executed on one of the two variables.

Exemple

`Switching command on`
Sends the command/the new value to the command variable. The success of the triggered action can be checked by means of the response variable.

`Status input off`
Resets status bits of the response variable configured in the action. The command variable is not relevant to this action.
*The **action variable** is the same as the **name of the response**.*

Note: You can find a description of the command actions in the action types (à la page 102) section.

NAME REPLACEMENT

To simplify or to generalize the definition of the variables, the variable references (for command variables, response variables and condition variables) can be defined using a name replacement. In doing so, wildcards '*' can be used. Wildcards are only permitted as a prefix or suffix; e.g. `*xxx` or `xxx*`.

As a result of this flexible definition, generally-valid procedures can be defined, which are then applicable for several data points. The number of command groups that must be defined is thus reduced considerably.

Example

- Definition of the command variables - **action variable** property = `*_CO`
- Definition of the response variables - **name of the response** = `*_RV`

In Runtime, the Command Processing automatically adds the name of the response variable, which is shown/selected in the process screen, to the name of the command variable. The names of both variables differ only in their endings.

This is also applicable for condition variables: `X01: *_ClsEna`.

Other variables - that have been linked to dynamic elements in the command processing screen - can also be replaced in Runtime.

You can find further details on this in the Substitution of additional variables (à la page 87) chapter.

EXECUTION

In general, the single-step operations are executed by means of the **context menu** of an element in the topology (such as a switch). A further typical use is the opening of a `Command Processing` screen instead of a **write set value** dialog.

The two-stage operations are executed by means of a **context menu** or the `Command Processing` screen type.

Specific control elements are available for this screen type. They enable an individual optical and functional design of the command processing. This way, individual actions, for example, can be assigned to **action buttons** directly. After this, these actions can be selected by the user directly. This screen type also includes the necessary requirements in order to carry out functions such as unlocking, two-step execution, two-hand operation, locking etc.

Note: You can find detailed information about the process in the `Command Processing` screen in the Process in the Command Processing screen (à la page 160) chapter.

Such a screen is called up on the screen element of the response variable by means of its context menu or instead of the **write set value** dialog. The call can also be by means of the **Appel de synoptique** function that is linked to a button.

An action (switching operation) can be the following in Runtime:

- Permitted
  If there is no locking condition applicable.
- Not permitted
  If there is an unlockable condition applicable.
- Permitted after unlocking
  If an unlockable condition is applicable.

This results from the command groups and the current status of the topological model **Configuration CAL** - **Interlockings** tab.

Note: You can find additional information on the procedure of a command in the Execution of a command (à la page 150) chapter.

In the zenon network, there is synchronization for actions from the command that concerns a certain response variable, by activating the `NET_SEL` status bit. The simultaneous execution on the same object (same variables) by different users is thus precluded. Parallel execution on different response variables is supported.

4.1 Command Processing

Menu item	Action
New command group	Creates a new command group.
Exporter tout (XML)	Exports all entries as an XML file.
Import XML...	Imports measuring units from an XML file.
Editor profile	Opens the drop-down list for selecting a Editor profile.
Help	Opens online help.

Informations

Command groups can be exported, imported and copied and pasted using the clipboard. The same applies for actions and their interlocking conditions, even different command groups.

4.2 Command processing detail view toolbar and context menu

COMMAND PROCESSING AND COMMAND GROUP CONTEXT MENU

Menu item	Action
New command group	Creates a new command group.
Export XML all...	Exports all entries as an XML file.
Export selected XML...	Exports selected entries as an XML file.
Import XML...	Imports from an XML file.
Copy	Copies the selected command group to the clipboard.
Coller	Pastes command groups from the clipboard.
Delete	Deletes the selected command group after requesting confirmation.
Rename	Enables renaming of a command group.
Properties	Opens the properties window for the selected command group.
Help	Opens online help.

CONTEXT MENU GROUP ACTIONS

Menu item	Action
Command new	Creates a new command and opens the properties.
New auto/remote command	Creates a new auto/remote command and opens the properties.
New forced command	Creates a new mandatory command and opens the properties.
New set value input	Creates a new set value input and opens the properties.
New status input	Creates a new status input and opens the properties.
New replace	Creates a new replace action and opens the properties.
New revision	Creates a new revision and opens the properties.
New manual correction	Creates a new manual correction action and opens the properties.
New block	Creates a new block action and opens the properties.
New release	Creates a new manual correction and opens the properties.
Check response value	Creates a new Check response action and opens the properties.
New lock	Creates a new lock and opens the properties.
Coller	Pastes action from the clipboard.
Help	Opens online help.

CONTEXT MENU INDIVIDUAL ACTION

Menu item	Action
New interlocking condition	Creates a new interlocking condition. **Note:** Grayed out for `mandatory command` command processing actions.
Copy	Copies the selected action to the clipboard.
Coller	Pastes action from the clipboard.
Delete	Deletes the selected action after requesting confirmation.
Help	Opens online help.

CONTEXT MENU CONDITION

Menu item	Action
Remove interlocking condition	Deletes selected condition.
Copy	Copies the selected condition.
Coller	Pastes the condition from the clipboard.
Properties	Opens the property window for the selected element.
Help	Opens online help.

CONTEXT MENU GROUP VARIABLES

Menu item	Action
Add variable...	Opens the dialog for selecting a variable.
Coller	Pastes variable from the clipboard.
Help	Opens online help.

CONTEXT MENU INDIVIDUAL VARIABLE

Menu item	Action
Remove variable	Deletes the selected variable from the group after requesting confirmation.
Copy	Copies selected variables to the clipboard.

Coller	Pastes variable from the clipboard.
Properties	Opens the property window for the selected element.
Help	Opens online help.

4.3 Engineering in the Editor

The **Command Processing** module is a comprehensive module with many possibilities for expanding the behavior in Runtime and amending it individually.

Please also note, for your project configuration in the zenon Editor, the information in the introduction for this manual. (à la page 63)

EXAMPLE OF AN INITIAL, SIMPLE COMMAND CONFIGURATION

1. Create a `command input` screen.
   Add the **control elements** from this template for this screen.

   Alternative project configuration for operation in Runtime by means of a context menu:
   Create a context menu with the following parameters:

   - **Type action** Command Processing
   - **Texte** ALL
   - **ID du menu** ID_CDM_AUTO

2. Create two variables.
   It is recommended that these variables are created with an Energy driver (IEC870, for example).
   Attention: The **internal driver** is not suitable for response variables.

   - Command variable
     Variable to write a command (open/close) to the controller, for example **IEC870** Variable `T46`;
   - Response variable
     Variable that displays the status (position: open/close/invalid etc.) of the relevant object in the same controller, for example `T03`.

   Name these variables.
   You can name these variables so that both names have a joint description at the start, for example `switch_Q0_CO` and `switch_Q0_RV`. This allows a command group to be used for several variable pairs.

3. Create a command group.
   Configure the following properties of the command group:

- **Nom de variable de réponse**:
  With name substitution: `*_RV`
  Without name substitution: Name of the response variable from step 2
- **Synoptique**
  Linking to a `command input` screen from step 1

4. Create actions in the command screen.

   a) Configure a **Type action** `switching command` action with the following parameters:

   - **Variable de l'action**
     With name substitution: `*_CO`
     Without name substitution: Name of the command variable from step 2.
   - **État de la réponse/direction de commutation**: `ON`
   - **Commande**: `1`
   - **Bouton d'action** `Action 1/Button: On`
     For execution without a context menu and without **A deux étapes**

   a) Configure a second action with the following different parameters:

   - **État de la réponse/direction de commutation :** `OFF`
   - **Commande**: 0
   - **Bouton d'action**: Action 2/Button: Off

   Hint: Copy the first configured action and amend the parameters.

5. Link the variables to the command group.
   Note: This is always required, regardless of whether name substitution has been configured or not.

   a) Select the two variables created in step 2.

   b) In the **Écrire valeur prescrite** properties group, configure the **Groupe de commandes** property. To do this, select the three created command groups created in step 3 from the drop-down list.

6. Configure a trigger for the created processing:

   a) Create a new zenon screen. The type of the screen can be any you want except `command input`.

   b) Place a dynamic screen element on this screen.

   c) Link this screen element to the response variable.

   d) In the drop-down list of the **Écrire valeur prescrite par** property, select the `Command` entry. You can find this property in the **Écrire valeur prescrite** properties group of the screen element.

   Alternative project configuration for operation in Runtime by means of context menu:
   In the drop-down list of the **Menu contextuel** property, select the name of the context menu

created in step 1.
You can find this property in the **Runtime** properties group of the screen element.

The user can now click on the configured screen element in Runtime (right-click for context menu) to trigger the actions of the command.

> **Informations**
>
> *For tests too, use a driver that supports the evaluation of the COT (Cause of Transmission - **Cause of transmission**) in full, for example the IEC 60870-5-101_104 driver. COT evaluation is an enhanced functionality to monitor communication during a command using the **Tempo watchdog** settings. The status bits `COTx` of the command variables can also be evaluated in the reaction matrices **multi-numerical** and **multi-binary**.*

4.3.1 Creating a screen of the type Command Processing

A **command processing** screen allows control in Runtime and an overview of the command processing. The command processing can be controlled in Runtime using buttons.

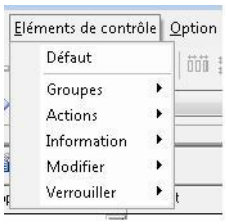
The command processing screen is created in the Editor configuring a new **command processing** screen. (You will find more information on the pre-defined screen types in the manual Screens/Pre-defined screen types'.)

The screen `Command Processing` is used for user interaction via command during the runtime (one and two-step command). It allows the user to perform all activities that are necessary for command execution. This can be, for example, the unlocking of an active action or the confirmation of the execution of a two-step command.

> **Informations**
>
> *When using one-step command processing, a context menu can also be used. The screen type command processing is then not required in the project.*

You can use specific control elements (à la page 159) for this screen type, which allow all user actions necessary for command processing and which visualize current information about the status of the action to be executed (e.g. display of the switching direction).

It is opened as an empty one after a new screen has been created. You add the default control elements via menu **Control elements**/**Add template**.



DÉVELOPPEMENT

Étapes de création du synoptique :

1. Créez un nouveau synoptique :

   Dans la barre d'outils ou le menu contextuel du nœud **Synoptiques**, sélectionnez la commande **Nouveau synoptique**.
   Un synoptique vide `Standard` est créé.

2. Modifiez les propriétés du synoptique :

   a) Nommez le synoptique dans la propriété **Nom**.

   b) Select `Command Processing` in the **Type de synoptique** property.

   c) Sélectionnez le cadre souhaité dans la propriété **Gabarit**.

3. Configurez le contenu du synoptique :

   a) Sélectionnez l'option de menu **Éléments de contrôle** dans la barre de menus

   b) Sélectionnez `Insérer un modèle` dans la liste déroulante.
   La boîte de dialogue de sélection de mises en forme prédéfinies s'affiche à l'écran. Certains éléments de contrôle sont insérés dans le synoptique à des positions prédéfinies.

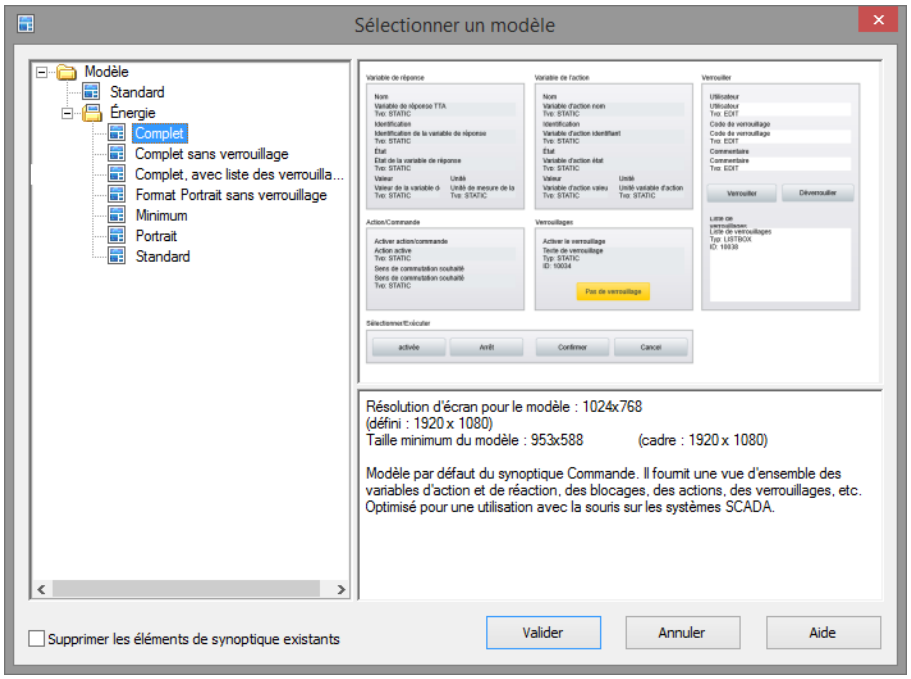
   c) Supprimez les éléments superflus du synoptique.

   d) Si nécessaire, sélectionnez des éléments supplémentaires dans la liste déroulante **Éléments**. Placez-les aux emplacements souhaités sur le synoptique.

4. Créez une fonction d'appel de synoptique.

Template

Several pre-defined templates are available for **Command Processing** screens.



Template	Description
List field templates (left)	Displays all pre-defined and user-defined template.
Preview and description (right)	Shows preview and description of the selected template.
Standard	Compact display of the command processing with visualization of: ▶ Actions ▶ Interlockings ▶ Buttons for selection and execution

ENERGY

Template	Description
Portrait Format	Display of command processing in portrait format, optimized for placing next to an overview screen: ▶ Actions ▶ Interlockings ▶ Buttons for selection and execution ▶ Lock
Portrait format without interlock	Simplified display of command processing in portrait format: ▶ Actions ▶ Interlockings ▶ Buttons for selection and execution
Complete	Enhanced display of the command processing with visualization of: ▶ Response variables ▶ Action variables ▶ Lock ▶ Actions ▶ Interlockings ▶ Buttons for selection and execution
Complete with interlocking list	Enhanced display of command processing including all interlockings: ▶ Response variables ▶ Action variables

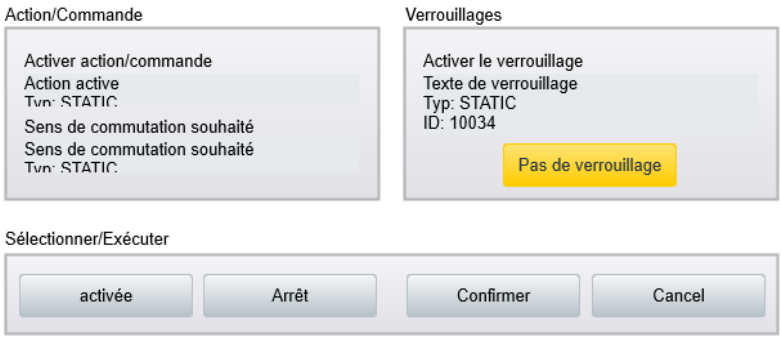
	▶ Interlocking (list) ▶ Actions ▶ Buttons for selection and execution
Complete without interlock	Enhanced display of command processing. With interlockings, only the currently-pending interlocking is visualized: ▶ Response variables ▶ Action variables ▶ Actions ▶ Interlockings (Text) ▶ Buttons for selection and execution
Minimum	Only contains visualization of buttons for selection and execution (à la page 168).

CLOSE DIALOG

Parameter	Description
Delete existing screen elements	Behavior when applying the template for configuration in the Editor. `Active`: Already existing elements in the screen are deleted when taking over the template. Default: `inactive`.
Apply	Inserts the elements of the selected template in the screen and closes the dialog.
Cancel	Closes the dialog without inserting elements.
Help	Opens online help.

Screen template - standard

The **Standard** template only contains the most important control elements for `command` actions. It is suitable for actions that are executed using a **context menu** or from a **command sequence**.



ACTION/COMMAND

Control element	Description
Active action/command	Displays the pending action of the command group.
Switching direction	The switching direction configured for the active action. The texts are documented with the "switching direction" setting. Depending on the active action, the following text is shown: ▸ Command, revision, correction, replace: Text from limit value, depending on switching direction. ▸ Status: On or Off ▸ Other: empty

INTERLOCKINGS

Control element	Description
Active interlocking	Interlocking text of the active interlocking.
Unlock	If an unlockable interlocking is upcoming, it can be unlocked with this button. Note: This control is shown only when the screen is in the step 'Unlock'. The Control is locked when the upcoming interlocking is not unlockable.

SELECT/EXECUTE

Control element	Description
On	Command button for switching command, to close a switch for example.
Off	Command button for switching command, to open a switch for example.
Confirm	Confirms for the pending two-step action. A two-step switching command, for example, is only executed after clicking on this button.
Cancel	▸ Closes the command processing screen. The pending action is not executed. ▸ Cancel for pending two-stage action (Cancel instead of Execute). ▸ Cancellation of the execution of an action (depending on the configuration, for example: cancel Operate if it has still

	not been terminated). The button is grayed out if the screen is in 'Step 1'.

Control elements - complete overview

The following elements are available in the **Control elements** menu bar in zenon for the `command input` screen:

Name	Control type		Default
Action buttons	Text	Buttons, which can have an action assigned to them. By clicking in the screen, the assigned action is activated and the screen changes to the step "Release" The button is not shown when: ▸ No action is assigned to the button in the current command group. ▸ The variable, with which the screen was loaded, is the command variable, and the action assigned to the button does not use the command variable as action variable. However, if the action 'Lock' was assigned to the button, it is visible. The button is shown as locked when: ▸ The screen is not in 'Step 1'. ▸ The response variable has set one of the status bits I_KENNUNG(18), OFF(20) or NICHT_AKTUELL(29) and writes the assigned action to the command variable. ▸ The response variable has the status REVISION(9) active and the assigned action writes to the command variable. ▸ The response variable has the status REVISION(9) active and the assigned action is 'Correct'. ▸ The assigned action is 'Release' and the response variable does not have the status Alternativevalue(27) active. ▸ The assigned action is 'Correct' and the value of the response variable matches the switching direction. ▸ The assigned action is 'Replace' and the value of the response variable matches the switching direction. ▸ The response variable has the status REVISION(9) active and the assigned action is 'Replace'. ▸ The assigned action is 'Revision' and the value of the response variable matches the switching direction.	Action1 Action2

RV TTA	Text	Name of the response variable	X
RV identification	Text	Name of the response variable	X
Action variable unit	Text	Unit of the current action variable.	X
Action variable set status	List	Defines the status to be set for the action 'Status default' for the switching direction 'None'. The statuses are set to the current status and updated when changes occur. Is locked when the active action is not 'Set status'.	
Switching direction	Text	The switching direction configured for the active action. The texts are documented with the setting 'Switching direction'. Depending on the active action, the following text is shown: Command, revision, correction, replace: Text from limit value, depending on switching direction. Status: On or Off Other: empty	X
Execute Step 2	Button	Delivers the actions to execution. This control element is visible only when the screen is in 'Step 2'. The control element is locked when: ▸ Two handed operation was configured and the key 'Ctrl' is not pressed. ▸ The status REVISION(9) of the response variable is set and the assigned action is 'Command', 'Set value', 'Replace' or 'Correct'. ▸ The button was already clicked.	X
Action variable minimum	Numeric	Minimum value of the action variable. Not visible if the action variable is of data type 'String'.	
Action variable maximum	Numeric	Minimum value of the action variable. Not visible if the action variable is of data type 'String'.	
Scrollbars	Numeric	Setpoint input with scroll bar Sets the value in the control element 'Set value' or is set by	

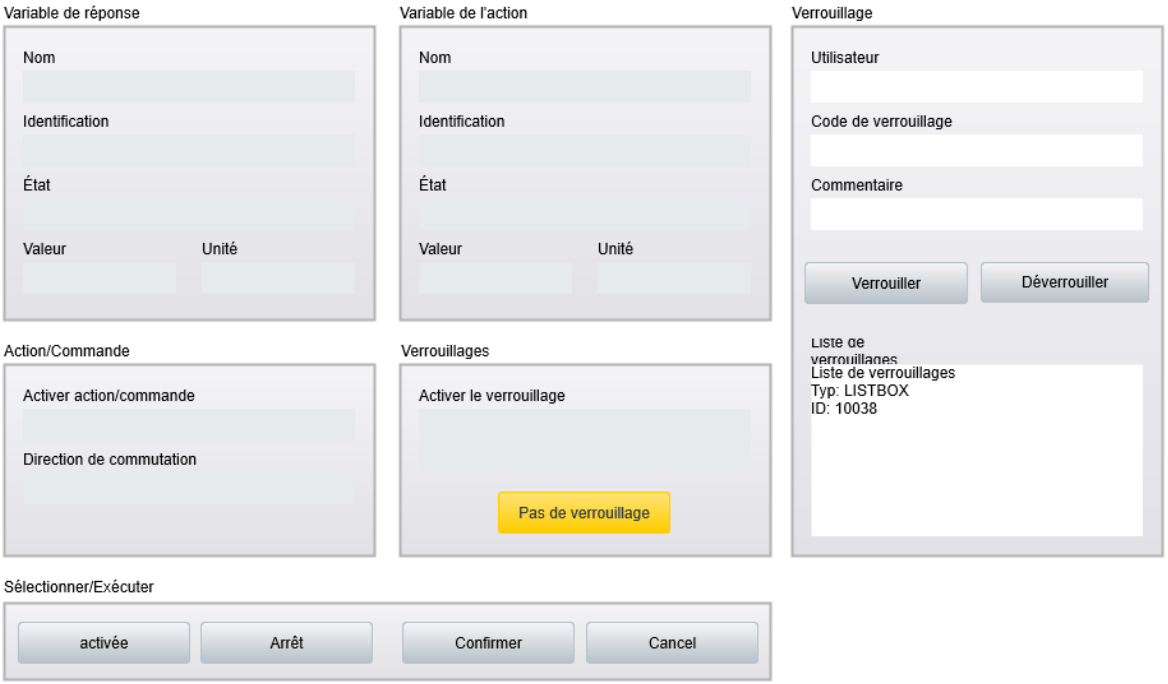
		this value. Not visible if the action variable is of data type 'String'. The control element is locked when: - No action is active. - The screen is not in 'Step 1'.	
Set value	Numerical, Text	Allows the input of the set value. By clicking the control element, it is switched to edit mode and the setpoint input is possible. The edit mode can be left again with "Enter". The new value is set only after clicking the control element 'Execute'. The desired value for the action 'Set value' is provided with this control element. The control element is locked when: - The status REVISION(9) of the response variable is set. - No action is active. - The screen is not in 'Step 1'.	
RV value	Text	Value of the response variable	X
RV status	Text	Contains the status of the response variable in the short form.	X
RV unit	Text	Unit of the response variable	X
Interlocking text	Text	Text of the upcoming interlocking. Text is online language switchable	X
No interlocking		If an unlockable interlocking is upcoming, it can be unlocked with this button. Note: This control element is shown only when the screen is in the step 'Unlock'. The control element is locked when the upcoming interlocking is not unlockable.	X
Exit	Button	Closes the screen without action execution. The button is only visible in a modal screen. This button is important for modal screens, because it is required to leave the screen in case of an error!	

Cancel	Button	Aborts the execution of the Command Processing and returns to 'Step 1'. The button is locked when the screen is in 'Step 1'.	X
Lock list	List	Contains the locks that were activated at the response variable. Is locked when no action 'Lock' was configured for the command group. Text is online language switchable	
User identification	Input field	For entering the user identification for the lock. Is locked when no action 'Lock' was configured for the command group.	
Lock code	Input field	For entering the user-specific lock code. Is locked when no action 'Lock' was configured for the command group.	
Execute lock	Button	Activates a lock for the user entered in the control element 'User identification'. Is locked when no action 'Lock' was configured for the command group. This user action is logged in the CEL, if not suppressed by the engineering.	
Unlock	Button	Removes the lock by the user entered in the user identification. Is locked when no action 'Lock' was configured for the command group. This user action is logged in the CEL, if not suppressed by the engineering.	
Execute	Button	Takes over the value of the control element 'Set value' or 'Set status' This control element is visible only when the screen is in 'Step 1'. The control element is locked additionally to the general lock, when: ▸ The active action is not 'Set status', 'Set value' or 'Correct set value'. ▸ The value in the control element 'Set value' for the action variable is invalid.	
Comment	Input field	Comment about the lock.	

Action variable Status	`Text`	Status of the active action variable in short form.	`X`
Action variable Name	`Text`	Name of the active action variable.	`X`
Action variable Identification	`Text`	Identification of the active action variable.	`X`
Action variable value	`Text`	Value of the active action variable.	`X`
Active action	`Text`	Name of the active action.	`X`

Screen Template - complete

The **complete** template contains all control elements for `command` and `lock` actions.



Parameter	Description
Response variable	
Name	Name of the response variable
Identification	Name of the response variable
Status	Contains the short description of the status bits for the response variable.
Value	Current value of the response variable
Measuring unit	Measuring unit of the response variable
Action variable	
Name	Name of the action variable
Identification	Identification of the action variable
Status	Contains the short description of the status bits for the action variable. Example: ▸ Bits for COT ▸ Status SE_870 during Select ▸ Status PN bit in the event of a negative response from the PLC Note: The status bits contain the response variable for the "status input" action.
Value	Current value of the action variable or input field for **setpoint input** command processing action. Note: This value changes during the course of the action from an existing to a current value. The display of the value is only refreshed with COT=7 (COT_actcon) or WR-SUC . ▸ The following is applicable for a configured setpoint input: The value to be set for the 'Set value' action is stipulated by this control element. By clicking the control element, it is switched to edit mode and the setpoint input is possible. It is possible to leave the editing mode again by pressing the Enter key. However the new value is only set when the **"Execute"** control element is clicked on. The control element is blocked if: - The response variable has set the status REVISION(9). - No action is active.

	- The screen is not in the "Step 1" stage.

Measuring unit	Measuring unit of the action variable
Lock	Control elements from the **Lock** group are locked if no "Lock" action is configured in the command group.
User	For entering the user identification for the lock.
Lock code	For entering the user-specific lock code.
Comment	Optional text that can be entered by the user for the lock.
Lock	Activates a lock by the user entered in the `User` control element. **Note:** This user action is logged in the CEL, if not suppressed by the engineering.
Unlock	Removes a lock that has already been activated. In doing so, only locks that the user themselves have activated can be deactivated. As a result, it is ensured that only people's own locks are removed. The user is visualized in the "`User`" control element. If there is no `Lock` action configured in the command group, this button is grayed out. **Note:** This user action is logged in the CEL, if not suppressed by the engineering.
`Lock list`	List of active locks: ▶ User Name of the user who has activated the lock. ▶ Locking time Time stamp of the interlocking ▶ Note Text for the interlocking.
Action/command	
Active action/command	Type of active command processing action such as dual command, for example.
Switching direction	The switching direction configured for the active action. The texts are documented with the setting 'Switching direction'. `Depending on the active action, the following text is shown:` Command, revision, correction, replace: Text from limit value, depending on switching direction. Status: On or Off Other: empty

Interlockings	
Active interlocking	The active interlocking (à la page 129) according to the configuration or texts from ALC - topological interlocking (à la page 35).
Unlock	This button unlocks an active, unlockable interlocking. **Note:** This control element is shown only when the screen is in the step 'Unlock'. The Control element is locked when the upcoming interlocking is not unlockable.
Select / execute	
On	First-step command button, to close a switch for example. **Note:** Only visible in Step 1.
Off	First-step command button, to open a switch for example. **Note:** Only visible in Step 1.
Confirm	Second-step command button. **Note:** Only visible in Step 2.
Cancel	Second-stage command button. Aborts the execution of the command processing and returns to 'Step 1'. The button is grayed out if the screen is in 'Step 1'.
Close	Closes the Command Processing screen.

Substitution of additional variables in the screen

For command input, in addition to variable substitution of zenon and the use of placeholders in response and command variables, further substitution rules can be configured for each command group and command action. You can thus place further dynamic elements in the command screen, which are linked to additional variables, whose names are then also automatically substituted in Runtime. Substitution is carried out in accordance with the response and command variables.

When configuring a project in the zenon Editor, you can find the **Remplacer dans le synoptique** property for each command group or command action. This properties are in the **Synoptique de gestion des commandes** property group.

REQUIREMENTS FOR SUBSTITUTION

Requirements for use are:

- The response variables and command variables were configured in the command processing with the * (star) placeholder.
  Example: `*_RV, *_CO`
  Response variable: **Nom de variable de réponse**
  Command variable: **Variable de l'action**
- A command processing screen is assigned in the **Synoptique** in the respective command group or command action.

Variables are substituted according to the following rule in the command processing screen:

- The text from the property is substituted in the variable name that is shown in the command field.
- It is substituted with a text which command found in the name of a response variable or action variable in place of the placeholder *.

Several texts to be substituted are configured separately with a semicolon (;). These phrases are substituted from left to right when calling up a screen in Runtime. The following phrases are ignored as soon as a text for replacement is applied.

If, when calling up the command processing screen in Runtime, there is no variable name with the configured text, there is also nothing substituted.

SCREEN SWITCHING AND COMMAND PROCESSING SUBSTITUTION RULES

Substitution via the screen switching function can be combined with the substitution of command processing. The following rules apply for substitution:

- If the screen is called up with a **Appel de synoptique** function, the substitution configured in the function is used in Runtime.
- If the screen is called up using the **Command Sequencer** module or the menu, Runtime gets the screen and the substitution from the project configuration in the respective command action. If there is no substitution configured in the command action, Runtime gets the screen and the substitution from the command group. If there is also no substitution configured in the command group, there is no replacement.
- When clicking on a dynamic element that has `new set value input` configured, Runtime gets the screen and the substitution from the **setpoint input** command action.

> **Informations**
>
> *You can get further information on substitution via the screen switching function in the command processing chapter in the functions and scripts manual.*

EXAMPLE

The following are configured:

- Nom de la variable de réponse : `abc_RV`.
- Variable de réponse configurée pour le groupe de commandes : `*_RV`
- Variables supplémentaires sur le synoptique : `xyz_lock, xy_Switch`

Result - scenario 1

- **Remplacement dans le synoptique** configuré : `xyz;xy`
- Variables existantes dans le projet : `abc_lock et abc_Switch`.
- **Résultat :** Affichage des variables sur le synoptique : `abc_lock et abc_Switch`.

Result - scenario 2

- Configured **replacement in the screen**: `xy;xyz`
- Existing variables in the project: `abcz_lock` and `abc_Switch`.
- **Result:** Display of the variables in the screen for `abcz_lock` and `abc_Switch`.

Result - scenario 3

- Configured **replacement in the screen**: `xy;xyz`
- Existing variables in the project: `abc_lock` and `abc_Switch`.
- **Result:** Display of the variables in the screen: `abc_lock` and `abc_Switch`.
  Because `abcz_lock` is not present.

4.3.2 Variables of the command group

Command groups use firstly the variables of the switching actions (the response variable an command variable) ans secondly, optionally, the variables of the **command conditions** and the variables of **breaker tripping detection**.

In order for the Command Processing module to be used, the respective response and command variables must be assigned to a command group. This assignment is made in the **variables** node => for the variable => in the **Écrire valeur prescrite** properties group => in the drop-down list of the **Groupe de commandes** property.

Ensure that this assignment is configured for both response variables and command variables.

Note: Errors in project configuration are listed in the output window of the zenon Editor when compiling the project. In Runtime, invalid or incompletely-configured commands for the variables concerned are not called up.

For the response variables and command variables, the set value can only be set using the command processing; it can no longer be set directly using dynamic screen elements. For screen elements that the user triggers with command processing, the `Command` value must be selected for the **Écrire valeur prescrite par** property or a **context menu** must be linked. To do this, it is preferable to use the screen elements that are linked to the response variable (not command variable). This guarantees the availability of all actions of command processing (provided the user is authorized).

Note: The screen element can also be used if the response variable is "read-only", from an IEC 60870 controller for example. A combined element with a circuit breaker symbol can trigger the command, although the response variable itself cannot be changed. The position of the switch (open/closed) corresponds to the value of the response variable.

Despite the linked command input, the values of the command variables also cannot be written to directly:

- via the **RGM**
- via **API**
- In zenon Logic with the **Visible par d'autres programmes** property activated.

The command processing is ignored in the process.

Informations

*If a variable is linked to a command group, it is not possible to describe the variable with the zenon **Ecriture valeur prescrite** function.*

*Exception: If a `write set value` (à la page 107) command with **switching direction** `set value` has been created, the zenon function calls up this action in the background without the `command processing` screen being called up. This means that the **command conditions** (à la page 128) are checked. An active interlocking condition prevents the writing of a set value. During the execution of an action, the `NET_SEL` status bit is not set and the **Select Before Operate** variable property is ignored.*

*This is also applicable for the value entry of a variable that is linked to a dynamic element if `Element` was selected for the **Écrire valeur prescrite par** property.*

GENERAL EXAMPLE

The command group "`DPI one stage`" was configured with the name of the response variable `*_RV` and the switching actions in this group with the name of the action variable `*_CO`.

In the SCADA project, variables with the name `ied9_100_RV` (position of the switch) and `ied9_100_CO` (command for switch) are configured. And the `ied9_100_RV` variable was linked to a screen element with **Écrire valeur prescrite par** = `command`.

Link the two variables `ied9_100_RV` and `ied9_100_CO` in the **Groupe de commandes** command group property to the "`DPI one stage` command group". The respective wild card * is replaced with "`ied9_100` in the Runtime.

Other variables, such as `ied9_101_RV` and `ied9_101_CO` (etc.) can thus be linked to this command group. In Runtime, the command groups are then instanced several times and can be operated independently.

Furthermore, you can also define the optional variables of the **command conditions** and the **breaker tripping detection** with the placeholder `*`, for example **X01:** `*_EnableClose`.

> **Informations**
>
> *As a result of the different use of limit values/reaction matrices for the command variable/return variable, individual switching directions can be displayed for the actions. Always depends on which of the variables the desired action is to be executed.*

Limit values and reaction matrices for switching direction texts

The command uses the limit value text of the command variables for the display of the **switching direction** in the command screen and in the **context menu**.

Example: during execution of a command in two-stage command processing, a corresponding text is shown for **command** actions in the **switching direction** control element. These texts can be defined via the limit values or via the states of the reaction matrices.

In the **context menu** in particular, these texts give the user a better understanding or a better overview of the actions that are available in Runtime (e.g. '`Command: Open disconnector`')

You therefore have the possibility to issue different texts for each variable that uses the same command group. Several variable pairs (each response variables and action variables) can thus only use one command group and can nevertheless be displayed in an individualized manner.

If no limit value has been created for a variable and no reaction matrices are linked, the action uses a standard text:

Switching direction of the action	Standard text
None	@NONE
OFF	@OFF
ON	@ON
DIFF	@INTER
DIST	@FAULT
DIR	@DIR

Informations

As the switching direction texts are read out from the limit value settings, they are completely language switchable.

Project overlapping variables

Attention

The variables used in the command groups must be in the same project in order for the command processing to work properly.

If you do use a variable from another project (e.g. subordinate project in multi-project administration), the command processing group, the response variable, the action variable and the action-specific screen ('Command Processing' screen) is expected to also exist in the other project.

Informations

You can also use project-overlapping variables for the interlockings by the process. The above limitations apply only to the variables of the command group.

4.3.3 Configure command input

Select the **Command Processing** entry in the project tree. Select **New command group** in the context menu.

After creating a new command group, it is added to the detail view of the project manager with standard name "`Command group + index`". The index is replaced by a consecutive number.

Note: This name serves for the unique identification in the system.

> **Informations**
>
> *You can assign any name you like to the command groups. However, it must be ensured that the names are unique within the project: applies for **general interlockings** and **command groups**.*

The following parameters are available for command groups:

Parameter	Description
Nom	Name of the command group. Must be unique for all interlockings in the project. This name is used later with the variable that uses this command group. The command group can be renamed at any time.
Nom de variable de réponse	This is the variable name or the mask for the replacement of the response variable. A wild card * (star) that appears in a name serves as a placeholder for the substitute text. **Example:** ▸ `*_RV` ▸ `*/stVal[ST]` Only one placeholder can be used in a name. **Attention:** If the name remains empty or the variable that is used here (replaced or absolute) does not exist at the time of compiling, this command group is not available in the Runtime. A corresponding message in the output window points this error out during compiling.
Définir état PROGRESS	If activated, status bit **In progress** (PROGRESS) is written for actions `command` and `Manual correction`. The value that the status bit is set to depends on the switching direction of the action. The status bit is set to 1 if: ▸ the **État de la réponse/direction de commutation** of the action is `ON` or `OFF`. ▸ The response variable does not already have the value of the set switching direction. The status bit is set when checking the interlockings and remains until the execution of the action has been completed. This also implies that, in the case of **Select Before Operate**, the status PROGRESS is only set after a successful 'Select' `(SE+COT_actcon)` and then remains set during watchdog timer or edge delay. If the execution of the action is triggered by a **context menu** or if it is a one-step action, the status bit is also set accordingly.
Tempo watchdog	There is the following setting for this drop-down list: ▸ `none`: The watchdog timer (à la page 150) is deactivated. However with **Select Before Operate**, there is a wait for confirmation of the `'Select' (SE+COT_actcon)` and it is then ensured that 'Select' has ended (PLC has reacted to `'Execute' COT_act` in the envisaged time). If `'Select'` has not yet been ended, the `'Select'` is deactivated a `'Cancel' (SE+COT_deact)` is sent to do this.

	▸ `Response variable only`: The value of the response variable (RV) is used to check whether the process was successful. ▸ `Cause of transmission only`: The status bits for Cause of Transmission (COT) of the command variable are used to check whether the process was successful. ▸ `COT and RV`: Both conditions defined above.

Synoptique modal	If activated, the screen is displayed modally, regardless of the configuration for the **Dialogue modal** property for the screen.
Titre de synoptique depuis la variable de réponse	The **Identification** of the response variable is shown in the screen title. This only happens when there a title was configured for the screen at the frame. Language switching is supported.
Synoptique	Name of the calling screen if the command is called up using a screen element of the response variable (or action variable). Note: Actions called up via the context menu open, for the confirmation of the second stage or interlocking text, a screen that has been defined for the action. Only if no screen has been linked for the action a defined screen is also used here in the context menu.
Détection déclenchement disjoncteur	Only available if property **Définir état PROGRESS** is activated. `Active`: The response variable is monitored for a change from `<> 0` to `0`. The identification only sets the status bit `CD_TRIP (50)` to 1 if: ▸ status bit `CB_TR_I` (51) is not `1`, otherwise the identification is suppressed. ▸ status bit `PROGRESS`(10) is not `1`, otherwise the value change of the response variable is considered a result of its own command. Attention: Value changes that are a delayed consequence of its own command can be recognized as breaker tripping. This happens if the PROGRESS bit has already been deleted or if the action does not support **Tempo watchdog**.
Supprimer détection	Entering the formula with which the detection of a breaker tripping can be suppressed. A click on button **...** opens the formula editor. All variables from the **Variables** node in the command group can be used for the formula. Variables from projects loaded in Runtime can be used. Name replacements with '*' - as with the definition of the interlocking conditions of an action - are possible. The suppression sets the status bit `CB_TR_I` (51) to 1. With active recognition, all variables whose status or value are used in the formula for breaker tripping detection are activated for reading when the program is started after loading all projects, and remain this way as long as Runtime is running. Note: Variables that are used in the formula cannot be deleted from the list of the variables linked to the command group.

Informations

The response and action variables do not need to be in the list of the variables linked to the command group. Their names need only be configured for the command group and in the action.

Command Processing in Distributed Engineering

Informations

*Because the **command conditions** and the **general interlockings** (standard functionality - without Energy Edition) are saved in the zenon Editor with the same structure, the check-out symbol (allow changes) is set to exactly the same for both nodes in the project tree. All actions on the command conditions also apply to the general interlockings and vice versa.*

Variables marked for deletion are considered as not existent for the compilation of the command conditions. During compiling, the respective error messages are displayed in the **output window** in the zenon Editor.

Create action

Actions define the switching commands that are possible for command groups. By selecting the element **Action** in the detail view of the command group, you can define a new action with a right mouse click. Details of the defined actions are also shown in the detail view after creation (e.g. "`switching command: *_BE [ON,1]`").

All further settings for the actions are made in the properties window. Some of the properties are inactive, depending on the action type.

The following properties are available for a command action

Parameter	Description
Paramètres d'action	
Variable de l'action	Variables on which is written. For some actions, this is the response variable. In this case, the field is locked. The placeholder for the replacement text is the character sequence '*' within a name. Only one placeholder can be used in a name. If the variable that is used here (replaced or absolute) does not exist during compiling, the action is not available in the Runtime. An according message announces this error during compiling. Click on the **...** button to open the dialog for selecting a variable. Default: `No Allocation`
Type action	Shows the type of command. For editing, only approved for **command** action type; possible settings are `switching command` or `pulse command`. Default: `Switching command`
État de la réponse/direction de commutation	Defines the expected value and the status of the response variable after action execution. Locked for the actions `block`, `lock` and `release`. Default: `Off`
Commande	Defines the value that is written to the command variable with the **Command** action. Note: only available for the command actions `switching command`- and `pulse command`, `auto/remote command` and `forced command`. Default: `0`
Durée d'impulsion	Time in milliseconds by which the resetting of the value is delayed for a `pulse` command. Note: Only available for the `pulse command` action. There is no wait until until **runtime monitoring** has ended. Default: `1000 ms`
Spécifier valeur prescrite	Defines the value that is written to the controller. Note: Only available if **État de la réponse/direction de commutation** has been set to `DIR`.
États modifiables	List of the states which can be modified with the `Set status` action. Note: Only available for the action '`Status input`'. Default: `None modifiable`
Synoptique de gestion des commandes	

Synoptique	`Command Processing` screen that is used when the action has been carried out using the context menu of the element. If no screen is entered, the screen, which is entered in property **Synoptique** for the command group, is used. An engineered screen which is not available, creates an error message when creating the Runtime files. In this case the action is not taken over. Default: `none`. Note: If the command processing is called up by a dynamic screen element, this property is ignored and the screen that is entered in the **Synoptique** property in the command processing group is always used. Not available for the `auto/remote command` action type.
Remplacer dans le synoptique	Substitution rule for command screens: The target of the substitution is configured in this property. The text that is to be replaced is configured. Several substitution rules are separated by a semi colon (;). In doing so, the configured substitutions are processed from left to right. If there is no * in the response variable or action variable, there is no substitution. There is no distinction between upper-case letters and lower-case letters in the project configuration.
Bouton d'action	Assignment of an action to an action button. Action buttons are configured in a command screen. If the command group is used for another screen (e.g. via function), the allocation to the action button remains nevertheless. In other words: the action is always placed on the button with the allocated action ID. If such a button is missing, the action is not available in the screen. Only the action buttons that were not allocated yet are provided in the selection list of this property. This setting is locked if no screen was allocated to the command group and for the `Lock` action type. Default: `No Allocation`
Comparaison valeur nominale/actuelle	If this is active, there will be a check whether the value of the response variable already matches the **État de la réponse/direction de commutation**. If this is true, an unlockable interlocking variable is shown. Note: Only active for `command` action type Default: `Inactive`
Can only be executed if target is <>	Deactivates an action button in the command screen if the value of the response variable already matches the value of the set value. If the value of a response variable is changed, the corresponding action button is active again. The same also applies for context menu entries. The corresponding command action is not displayed in the menu here. Note: Only available if the **Comparaison valeur nominale/actuelle** property is activated and only available for the `command` (switching or

	pulse command) command action. Default: `Inactive`
A deux étapes	If this is active, only after operating the **Execute 2** control. **Step** that executes action. If not active, the action is executed after releasing the last interlocking or, if there is no upcoming interlocking, immediately. Note: Locked for the `lock` and `auto/remote command` actions. Default: `Active`
Opération à deux mains	▸ `Active`: The **Execute 2** control element. **Step** is only unlocked if the `Ctrl` key is held down. In Multi-Touch applications, both pressure points must each be on their own screen with their own frame. Not available if no two-step execution has been configured. Note: For the **Execute 2nd step** control element, the **sélectionnable avec le lasso** property must not be active with two-hand operation. Default: `Inactive`
Fermer automatiquement	If this is active then the screen is closed automatically after action execution. Note: Not available for the `auto/remote command` action type. Default: `Inactive`
ID du menu	The menu ID is used for the creation of Context menus in the Runtime. Note: If two actions are fitted with the same ID, they are tagged with a special symbol in the action tree. They can then not be called up by the context menu.
Nom de l'action	Freely-definable name of the command action. This can, for the **Command Processing** module, be displayed in Runtime using a screen of type Command Processing. In the **Command Sequencer** module, this name must be used for the step.
Séquenceur de commandes	Project configurations for the **Command Sequencer** module.
Options	
Supprimer l'entrée de la liste d'événements	If this is active, no entry in the CEL will be made when executing an action. Default: `Inactive`
Timeout	Timeout for the runtime monitoring in seconds for `switching command` and `pulse command` actions. This setting is also applicable as a timeout for Select.

	Unit is seconds Only available for the actions `Command`, `Auto/Remote command`, `Check response`, `Setpoint input` and `Forced command`. Default: `30`
Le délai d'attente peut être annulé	Allows the cancellation of the timeout in runtime monitoring. Only available for the `command` and `Setpoint input` actions. If the command has already been executed - after **COT_actcon** has been received - the **Cancel** button cancels runtime monitoring. Buttons are therefore active and operable again. **Note:** Not all drivers support deactivation during execution. If not, no Cancel is sent to the controller; the action is canceled only.
Utiliser le qualificatif de commande	Allows commands to provide additional information (Qualifier of Command). The requirement for this is that the driver also supports this option. Possible drivers are, for example, IEC850, IEC870 and DNP3. Is only available for the actions `command`, `auto/remote command` and `forced command`. Default: `Inactive`
Qualificatif de commande	Entry of a numerical value that is sent to the driver as a command parameter. This input possibility is only available the **Utiliser le qualificatif de commande** property has been activated. ▸ Input range: `0 – 127` ▸ Default: `0`

Attention

*The identification of the action types in the **ID du menu** must be clear, so that they are clearly identifiable in the context menu (à la page 118). If two actions have the same ID, they are tagged with the special symbol **M** in the action tree.*

Conseil

Note:

- When selecting individual properties, you receive additional information about functionality in the embedded help.
- Defined actions and commands can be exported into XML and imported from XML. They can thus be easily archived or reused in other applications.
- The status can be set using the command **status input**.

Types d'actions

Les types d'actions sont les procédures de commande disponibles. Selon la commande, différentes actions sont exécutées.

Le système propose un certain nombre d'actions. Les types d'actions suivants peuvent être définis pour les groupes de commandes :

Type action	Annotation
Commande (à la page 104)	Commande de commutation ou commande d'impulsion. Utilise la valeur de la variable de commande pour écrire l'état configuré de la fonction de gestion commande sur le contrôleur. **Remarque :** La commande de commutation est adaptée aux commandes impulsion et doubles avec le driver Energy (IEC60870, IEC61850, DNP3).
Nouvelle commande auto/à distance (à la page 105)	La commande à distance est transmise depuis Process Gateway ou l'API de zenon au module de traitement de commandes, puis traitée comme commande de commutation. L'action est indisponible dans un synoptique de traitement de commandes et dans le menu contextuel.
Nouvelle commande forcée (à la page 106)	Le type d'action Commande forcée permet de définir une commande, même si la variable de réponse est empty, OFF, NT ou INVALID. **Remarque :** L'action est destinée aux arrêts d'urgence, et doit être utilisée avec précaution.
Envoi valeur (à la page 107)	Écrit la valeur numérique souhaitée dans la variable de commande.
Envoi état (à la page 108)	Modifie les bits d'état de la variable de réponse. Applicable uniquement aux bits d'état de la liste **État modifiable**.
Valeur de remplacement (à la page 109)	Modifie l'état de la variable de réponse en la valeur de remplacement (ALT_VAL) et écrit la valeur dans la variable de réponse. **Remarque :** L'écriture de variables en valeurs de substitution permet la visualisation du processus avec des données collectées manuellement lors d'un échec de communication, par exemple via la **Coloration Automatique des Lignes**.
État révision (à la page 110)	Définit le bit d'état REVISION de la variable de réponse. **Remarque :** La gestion des alarmes est supprimée durant la révision.
Nouvelle correction manuelle (à la page 110)	Définit la valeur de la variable de réponse sélectionnée en fonction de la direction de commutation. **Remarque :** les protocoles de communication dans Energy (IEC60870, IEC61850, DNP3) excluent l'écriture directe dans la variable de réponse.
Nouveau blocage (à la page 111)	Désactive la variable de réponse (bit d'état OFF). **Remarque :** les variables désactivées ne sont plus lues par le matériel connecté.
Nouveau déblocage (à la page 111)	Définit la valeur de la variable de remplacement (ALT_VAL) sur 0. **Remarque :** en conséquence, la variable de réponse reçoit une nouvelle fois la valeur du contrôleur.
Vérifier la valeur de réponse (à la page 112)	Contient l'état de la variable de réponse sans exécution d'une activité. **Remarque :** l'action est destinée à une utilisation dans le module **Séquences de commandes**.

`Nouveau verrouillage` (à la page 112)	La variable de réponse verrouille toute action ultérieure lors de la saisie d'un **code de verrouillage** valide.

Remarque : Les types d'action sont présentés dans la liste détaillée ci-dessus dans l'ordre d'affichage des types d'action dans le menu contextuel de zenon. Toutefois, l'ordre d'affichage dans la fenêtre principale est alphabétique.

Dans la vue détaillée du traitement de commandes, les actions dans l'arborescence sont présentées avec la direction de commutation sélectionnée et la valeur d'action configurée.

Attention

*L'identification des types d'actions dans l'élément **ID du menu** doit être claire, afin qu'ils soient facilement identifiables dans le menu contextuel (à la page 118). Si deux actions comportent le même ID, elles sont identifiées à l'aide d'un symbole spécial (**M**) dans l'arborescence des actions.*

Action type command

This **Type action** is used as a `switching command` or `pulse command` depending on the configuration.

When the command is executed, a value (0 or 1) is written to the command variable. The value to be written is configured with the **Commande** property.

This action type supports **Select Before Operate** and **Tempo watchdog**. The Select Before Operate depends on the corresponding property of the command variable and on the driver.

The **Tempo watchdog**, depending on the configured type (`via response variable` for example) can also check whether the response variable changes its value according to the command. The value which that is then expected for the response variable as a result of the command is to be defined under the **État de la réponse/direction de commutation** (`on/off/none`) property.

Switching direction	Value of the response variable after a command has been executed
None	No specific value change is envisaged. The action is ended if the configured **Timeout** has expired. **Note:** the **Tempo watchdog** can nevertheless be activated in order to take the `cause of transmission` (COT) into account.
Off	The action expects value 0. If the response variable already has the value 0 before the command is executed, an internal interlocking condition is reported if the **Comparaison valeur nominale/actuelle** property is activated. **Note:** If runtime monitoring is configured with the values `none` or `via cause of transmission`, there is no wait for the response variable.
On	The action expects value 1 and (in accordance with **Comparaison valeur nominale/actuelle**) is compared to value 1.

With `pulse commands` a value is written to the PLC twice. The second time, after the configured **Durée d'impulsion**, there is an automatic reset to `0` or `1` (depending on the configuration of the **État de la réponse/direction de commutation** property). However this does not happen if the **Select Before Operate** property has been activated for the command variable. The Energy protocols do not provide any possibilities to use a Select with pulse. If the **Select Before Operate** property has been activated for the action variable, a `pulse command` acts in the same way as a `switching command`.

Please note: The pulse command is not recommended for Energy drivers. The pulse command should only be used with a PLC that expects a pulse instead of an edge.

Note

*If, during the execution of the action, the current value of the response variable is different to the one defined in the switching direction and the switching direction was defined to be `on` or `off`, the **in progress** (PROGRESS) status bit is set. To do this, activate the **Définir état PROGRESS** property in the command group.*

Auto/remote command action type

The **Remote command** (via Process Gateway, VBA, etc.) is forwarded to the zenon command processing, which processes the sequence (checking of interlocking, forwarding to driver, response, etc.) like a **Switching command (à la page 104)** .

The command processing is not accessible via the command screen or the context menu.

The command is only supported by a previous Select . The action variable must have the **Select Before Operate** property activated.

When Runtime is ended, or reloaded, any Select that has been set is discarded. This means: The master connected to the Process Gateway is not informed and must get this itself using the interruption to the connection.

The VBA interface can use the **IVariable::SetValueWithStatusEx** method and the status bits to be transferred decide whether writing should be either direct or via the command processing. If the status bit `NET_SEL` (bit 8) has already been set (the command processing screen is open for example), the command is not executed. If the status bit is not set, it is set and writing is executed by Command Processing or commands are forwarded to the Command Processing. The response value of the method provides information on whether command processing has been activated or whether the command has been executed.

Transfer of the status bits of the action variable to the method:

- `SE_870` + `COT_act(6)` - Select activation
  Determines the command action to be executed and activates the command processing. The response variable of the method provides information on whether this is possible.
- `SE_870` + `COT_deact(8)` - Deactivation (Cancel)
  Ongoing command processing is canceled.
- `COT_act(6)` - Activation (Operate/Execute)
  Execute for command Command Processing is executed.

In order for this method to be able to execute command processing, a **remote command** action must exist whose switching direction corresponds to the transferred set value. The actual value written to the driver, Select etc, results form the properties of the action.

Note: When an interlocking takes effect, a (language-switchable) CEL entry with the configured text is created.

Informations

You can find further information in the Select before Operate chapter in the Process Gateway manual, chapter IEC870 Slave.

Mandatory command action type

The **forced command** action type allows the setting of a command, even if the response variable is `empty`, `OFF`, `Not topical` or invalid (`INVALID`). It is not intended for emergency shutdowns.

Interlocking conditions cannot be created for the forced command, because it cannot be guaranteed that the condition variables have a valid value in Runtime.

Note: The forced command corresponds to a **switching command** without conditions.

> **Attention**
>
> *The early or erroneous configuration of a forced command in Runtime can have dramatic consequences for the equipment. Always set this command with care and protect it with user authorizations.*

Action type set point input

The **setpoint input** action type offers the possibility to set any desired numerical value to the command variable. The **command processing** screen offers its own control elements for this, which also allow manual definition of the set value. With the help of property **État de la réponse/direction de commutation** you can define how the set value should be written:

Switching direction	Value of the response variable
`DIR`	Set value is written directly. You define the value which should be written with the help of function **Spécifier valeur prescrite** of the action. The text which should be displayed can be engineered using a limit value/rema for the state/value 5. If this is not the case, a standard text (à la page 91) is used. **Comparaison valeur nominale/actuelle** is not yet supported! The action can be carried out several times in a row.
`Set value`	Value of the command processing screen of the **Set value** control element is written to the action variable. In one-step execution, the value is written when clicking on the **Execute** button or when clicking on the **action button** (if configured). In two-step execution, the value is written when clicking on the **Execute 2 step** button.

This action type, for `DIR`, supports **Select Before Operate.**

Informations

If a variable is linked to a command group, it is not possible to describe the variable with the zenon ***Ecriture valeur prescrite*** *function.*

Exception: *If a* `write set value` *(à la page 107) command with* ***switching direction*** `set value` *has been created, the zenon function calls up this action in the background without the* `command processing` *screen being called up. This means that the* ***command conditions*** *(à la page 128) are checked. An active interlocking condition prevents the writing of a set value. During the execution of an action, the* `NET_SEL` *status bit is not set and the* ***Select Before Operate*** *variable property is ignored.*

This is also applicable for the value entry of a variable that is linked to a dynamic element if `Element` *was selected for the* ***Écrire valeur prescrite par*** *property.*

For further information, read the information in the Apply actions (à la page 114) chapter.

Attention

When writing the set value with the switching direction `DIR`*, neither the limits of the linked variable are checked, nor is a check carried out to see whether write set value is permitted for this variable.*

Action type status input

Changes the status bits of the response variable. The following is executed, depending on the definition of the **switching direction**:

Switching direction	Action
`Off`	The states configured in the **Modifiable states** list are all reset to 0.
`On`	The states configured in the list **Modifiable states** are all set to 1 (active).
`None`	The states configured in the **Modifiable states** list must be defined in Runtime in the Command Processing screen with the help of the `Set status` control element. Each status bit is defined individually using a checkbox in the control element. ▸ In single-stage execution, the status bits are set by clicking on the **Execute** button or when pressing the **action button** (if configured). ▸ In two-stage execution, the status bits are set when the **Execute 2nd stage** button is clicked on.

If you change a status bit in Runtime, the change is logged in the Chronological Event List (status including value). These language of these messages can be switched in Runtime.

Informations

For all status defaults, there is always a write to the response variable.

Info

If a switch is locked using the `Lock` action, the status bit `M1` of the response variable is set.

The status bits `OFF` and `REVISION` are also handled in other actions.

In addition, the status bits `CB_TRIP` and `CB_TR_I` reflect the results from the property of the ***breaker trip detection command group****.*

Action type replace

The process value of a remote-controlled switch is temporarily replaced with a replacement value (due to revision, maintenance work, or an ongoing connection outage, for example).

The response variable is set to the status alternative value **Alternate value** (ALT_VAL). In addition, the value defined by the **switching direction** is placed on the response variable.

Switching direction	Alternate value
`Off`	0
`On`	1
`Diff`	2
`Fault`	3
`None`	4

The substitute value is not sent to the connected hardware. It is for the substitution of values using manually-collected information.

Conseil

By switching the response variables to substitute values, it is possible to portray the current topological status of the network in **ALC** whilst the SCADA system was disconnected from the process.

Action type revision

Sets the value for the Revision status bit of the response variable. The value is defined in the **État de la réponse/direction de commutation** property.

Switching direction	Status
`Off`	Set to 0
`On`	Set to 1

Action type manual correction

The **correct** action sets the value of the response variable according to the setting of the **switching direction**:

Note: the communication protocols in Energy (IEC60870, IEC61850, DNP3) preclude direct writing to the response variable. The action will be unsuccessful in these drivers! To execute a command, the setting of the value to a command variable is expected.

Switching direction	Action
`Off`	0
`On`	1
`Diff`	2
`Fault`	3
`DIR`	The set value is written directly. You define the value which should be written with the help of function **Spécifier valeur prescrite**. The text to be displayed can be configured using a limit value or a reaction matrix for the state/value 5. If this is not the case, a standard text (à la page 91) is used. **Nominal/actual value comparison** is not supported. The action can be carried out several times in a row.
`Set value`	Value of the **Set value** control element is written to the response variable in the Command Processing screen.

Attention

When writing the set value directly neither the limits of the linked variable are checked nor is it checked if the write set value is allowed for this variable.

Informations

*The **In progress** (PROGRESS) status bit is set if:*

- When the action is carried out, the current value of the response variable is different to the value set for the **switching direction**
  and
- the **switching direction** was defined as `on` or `off`.

MANUAL CORRECTION

Manual correction is the manual correction of a non-remote switch in zenon. A variable is usually corrected without a connection to the process. There should never really be an invalid i-bit pending for such variables. It is indeed possible, but it makes no sense to correct a variable with a reference to the process! The PLC will overwrite this value again.

Behavior:

***Correction** is completely normal value setting from the perspective of the driver.*

Opposite of this - Action: Replace (à la page 109)

The process value of a remote-controlled switch is primarily replaced with a replacement value (due to maintenance work, for example).

Action type block

the response variable is switched off as a result of executing this action.

The status bit of the response variable is set to OFF. The switched-off variables are no longer read by the connected hardware.

If the response variable already has the OFF status bit set, the action of the status bit is no longer set when the action is executed. Runtime receives the current value of the response variable from the driver.

Note: Can only be configured once per command group.

Action type release

The **Release** actions resets the **replacement value** (ALT_VAL) status bit to `0` (inactive). If the **Switched off** (OFF) status bit is also active, it is also set to `0` (inactive). runtime receives the current value from the driver for the response variable once the **Release** action has been carried out.

The action can only be executed in Runtime, if the **replacement value** (ALT_VAL) (value: 1). is active for the selected response variable. `1`).

Note: Can only be configured once per command group.

Check response value action type

The **Check response value** action type is to check variables for the status `ON` or `OFF`.

Whilst the **Check response value** action is executed, the standard key **Cancel** is unlocked in the **Command Processing** screen.

In doing so - depending on the setting of the **runtime monitoring** (à la page 155) - there is a wait until the value of the response variable corresponds to the value of the checking direction - **switching direction** action property. If the checking value is `EIN`, this is the value 1; it is the value 0 for `OFF`.

If no runtime monitoring has been configured (**runtime monitoring**= "`none`"), the set waiting time (~24 hours) is the maximum time that is waited. Otherwise the action is ended and the `TIMEOUT` status bit is set for the response variable.

If, after execution of the action in the Command Processing screen, the other actions are not available, this is for the following reasons:

- The **timeout** for **runtime monitoring** has not yet expired.
- The response variable does not yet have the expected value (the value change has not yet been received).
- The action has not yet been canceled with the **Cancel** button.

> **Informations**
>
> *The **Check response value** action only serves to read the value of the response variable without executing an activity.*
>
> *The action is intended for use in the **Command Sequencer** module.*

If the response variable already has the value of the **switching direction**, the execution of the action is recognized as completed. The other buttons in the Command Processing screen are thus immediately available.

Note: If the response variable is set to `OFF` or `Revision` , the response value can nevertheless be checked.

Action type lock

Enables the lock of a response variable for the actions of the command processing.

Note:

- Can only be configured once per command group.

- **Interlocking conditions** are not supported for the action. Locks can always be executed.

> **Informations**
>
> *If a switch is locked using the `Lock` action, status bit M1 is set.*

A prerequisite for this is that users have a **Code de verrouillage** configured in the **user administration module**. Locking or unlocking a response variable can only be done with the correct input of a **Code de verrouillage**.

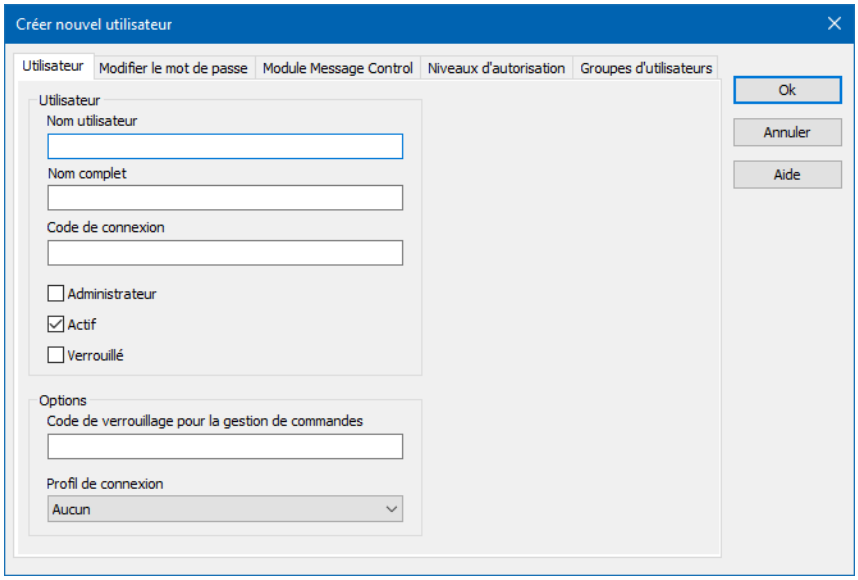
The same variable can be locked by multiple users in parallel. Actions for the response variables are possible only after alls locks have been unlocked by entering the **Code de verrouillage**.

There can be no actions executed if

- Actions of the command variables use the locked variable as response variable (e.g. `switching command`)
- Actions of the command variables use the locked variable as an action variable (e.g. `replace`)

A list of the currently-active locks can be shown in the command processing screen using a special `lock list` control element.

The **Code de verrouillage** can be defined individually for every user. These parameters are set directly for a pre-existing user with the **Code de verrouillage** property:



You can also set the **Code de verrouillage** for an existing user in the Runtime.

In the Runtime you cannot delete users who still have an active command lock.

Attention

Users can also be deleted in the development environment. This causes the loss of the defined locks after restarting or reloading.

Users locked (activated) in the **user administration** cannot activate or deactivate command locks.

Informations

Information about active locks is also synchronized in the redundant network and is therefore available after a redundancy switch.

Apply actions

Command Processing in the Energy Edition can be used in different situations. The user can choose the variant they prefer. A simultaneous use (related to an element) of the different types of use is possible at any time:

- Calling up a **screen switching** function on a **Command Processing** screen (à la page 115).
- Calling up a **numeric value**, **combined element**, **dynamic text**, **bar graph**, **clock**, **universal slider**, **pointer instrument** or **status element** screen.
  For activation, the **Écrire valeur prescrite par** property of the element must be configured with `Command Processing`.
- Call via a context menu if `Command Processing` was set for the **Type action** property. The command processing screen is opened for any possible interaction with the user (e.g. pending interlocking).
- It is called up using the **Command Sequencer module**.

  Conseil

  Always link all screen elements or functions that call up the command input to a response variable.

  Only by linking to a response variable is it ensured that all actions in the `command` screen are available for operation in in Runtime.

  Linking to a command variable is expressly not recommended!

As soon as the variable is linked to a command group, direct input of set values is only possible using a zenon `command processing screen` or a `command` context menu.
Exception: If the command group contains a **setpoint input** action with **État de la réponse/direction de commutation** '`set value`', this action is used for the command variables (not response variables).

This happens:

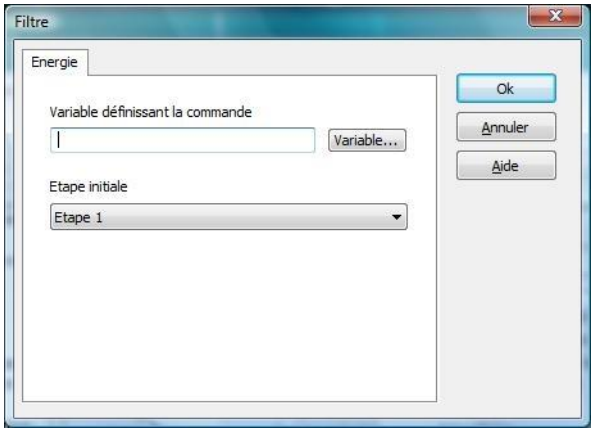
- When the **Ecriture valeur prescrite** is called up
- When calling up the screen element; also if the **Écrire valeur prescrite par** property of the element has the value `dialog box` or `element`.
- When calling up a `set value` context menu.

In doing so, the `NET_SEL` status bit of the response variable is not taken into account and no Select is executed.

> **Attention**
>
> With this type of execution, a pending interlocking condition in the **write set value** action prevents writing of a set value. In doing so, there is no interaction with the user.

Screen switch to screen of type Command Processing

If a `Command Processing` screen is selected with the **Appel de synoptique** function, the configuration dialog for the screen switching function has the following parameters:



Parameter	Description
Variable defining the Command Processing	The variable configured here defines the command group to be used. The screen determines the appropriate response variable and the associated action variable via the name of the variable.
Initial step	Defines the step (status) in which the command processing screen is loaded. ▸ `Step 1` The screen is loaded and waits for action definition and action execution. Action executions must be performed manually by the user. ▸ `Block` The screen is opened in the command step for the action block. **Note:** Note that not all configured control elements are visible with this initial step. You can find an overview of all visible control elements in the blocked or locked elements (à la page 163) chapter.

Attention

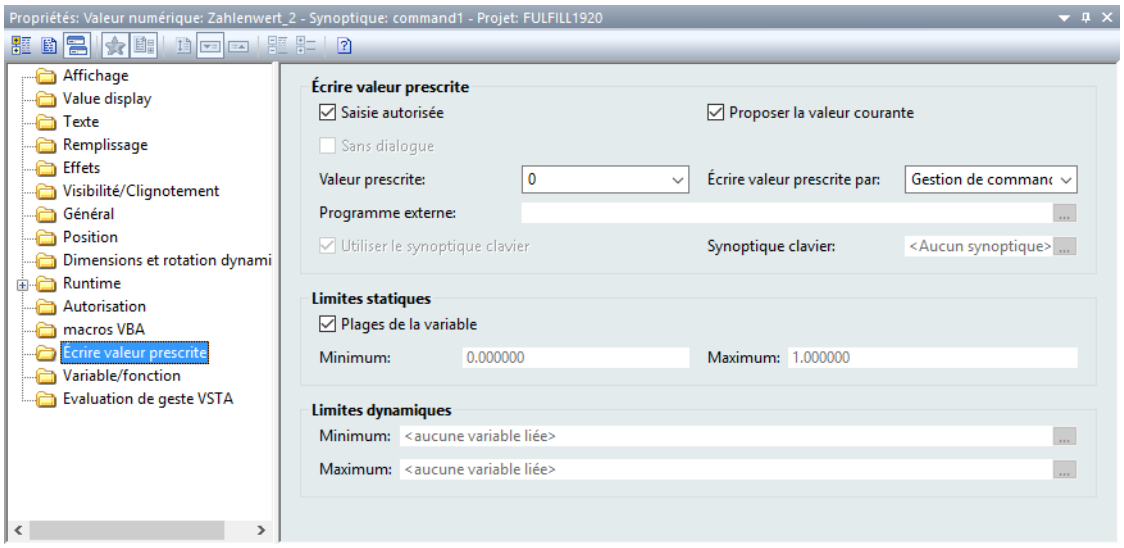
If there is no operating authorization for the command variable, screen switching to a `command` *screen is not possible.*

Command processing via dynamic screen elements

Command input can be instigated by clicking (left mouse button) on a dynamic screen element. In general, it is a combined element with a symbol of a switch in the topology or a numeric value element that displays the value of the position (`0` - off, `1` - on, `3` - invalid etc.) of the switch. The dynamic element should be:

- Configured with `command input` in the **Write set value via** property
- And linked to a response variable (an action variable for example) . That means to a variable that has been configured with a valid command group - using the **command group** property, as well as a suitable action variable (a response variable for example).

The screen to be opened - a `command input` screen - is defined at the **command group** of the variable linked to the element. The corresponding action variable (or response variable) is automatically determined from the response variable.



Informations

With the `Command Processing` setting selected, the command processing screen is called up instead of the standard `Write set value` dialog.

The following dynamic elements support the `Command` setting:

- Bar display
- Combined element
- Text element
- Clock
- Universal slider
- Numeric value
- Pointer instrument
- Status element

If no **command group** has been defined for the variable linked to the element, or the configuration of the command is invalid, an error entry for the Diagnosis Viewer is generated. The screen is not opened then.

Informations

If a variable is linked to a command group, it is not possible to describe the variable with the zenon ***Ecriture valeur prescrite*** *function.*

Exception: *If a* `write set value` *(à la page 107) command with* ***switching direction*** `set value` *has been created, the zenon function calls up this action in the background without the* `command processing` *screen being called up. This means that the* ***command conditions*** *(à la page 128) are checked. An active interlocking condition prevents the writing of a set value. During the execution of an action, the* `NET_SEL` *status bit is not set and the* ***Select Before Operate*** *variable property is ignored.*

This is also applicable for the value entry of a variable that is linked to a dynamic element if `Element` *was selected for the* ***Écrire valeur prescrite par*** *property.*

Command processing via context menu

The command processing can also be instigated at the element directly via the context menu - property **Runtime - context menu**. This is the most frequently used method. In this regard, the context menu is already the first step of the two-step action. For the second stage (**Execute** or **Cancel**), or an interlocking, a screen - which was linked for the action - may possibly be opened.

The menu must have an entry of the **command processing** action type. The display of the single action is defined automatically by the menu. The display of the actions can be influenced selectively, depending on the '**names**' of the menu entry.

Lors de la création d'une nouvelle action dans la fonction Gestion de commande (à la page 63), un identifiant de menu correspondant au type d'action et à la direction de commutation pour la propriété **Type action** est créé et proposé dans la liste déroulante. Si le contenu correspond à un identifiant défini en tant que texte standard pour le type d'action et la direction de commutation, le contenu est adapté si le type d'action ou la direction de commutation.

Pour créer un menu contextuel pour la fonction de gestion de commande :

1. Créez les actions souhaitées dans la fenêtre de gestion de commande (à la page 63).
2. Dans les propriétés de l'option de menu contextuel, sélectionnez la **Type action** **Gestion de commande**.
3. Sélectionnez l'action souhaitée et la direction de commutation via le menu déroulant avec la propriété **ID du menu**.
4. Spécifiez un intitulé clair dans la propriété **Texte**.
   Remarque : Si aucune entrée n'est définie pour **Texte**, le champ est automatiquement renseigné avec l'intitulé **Gestion de commande**.

Attention

La configuration de la propriété ***Texte*** *doit être unique. Si deux textes identiques sont fournis, les autres éléments de menu portant le même nom ne seront pas affichés.*

Les éléments de menu créés automatiquement avec la même action produisent le même texte ; aussi, des macros (à la page 123) sont disponibles pour ces derniers.

La séquence de caractères ID_CMD_AUTO est réservée pour les éléments de menu créés automatiquement. Ces derniers doivent toujours être utilisés conjointement à des macros, car dans le cas contraire, seul l'élément de menu est inséré.

Les règles suivantes s'appliquent lors de la recherche de doubles entrées :

- Les points de menu manuels sont prioritaires sur les points automatiques.
- Si deux entrées sont du même type, la priorité de la dernière entrée est deux fois plus élevée.
- Si une double entrée est détectée, un avertissement est inséré dans le fichier journal. Celui-ci inclut l'ID de menu et la description. Les ID des entrées pouvant être développées automatiquement reçoivent l'indication **<auto>**.

ACTIONS POUR LA TYPE ACTION GESTION DE COMMANDE

Action	Direction de commutation	ID de menu
ID_CMD_AUTO		Cette entrée de menu montre automatiquement toutes les actions possibles pour un élément, sauf les actions pour lesquelles une entrée de menu a été spécifiée explicitement.
Commande impulsion	On (1)	ID_CMD_EBEF_ON
Commande impulsion	OFF (1)	ID_CMD_EBEF_OFF
Commande impulsion	NONE	ID_CMD_EBEF_NONE
Commande de commutation	On (1)	ID_CMD_DBEF_ON
Commande de commutation	OFF (2)	ID_CMD_DBEF_OFF
Commande de commutation	NONE	ID_CMD_DBEF_NONE
Valeur prescrite	NONE	ID_CMD_SVALUE
Valeur prescrite	DIRECT	ID_CMD_SVALUE_DIR
Envoi de l'état	NONE	ID_CMD_STATE
Envoi de l'état	On (1)	ID_CMD_STATE_ON
Envoi de l'état	OFF (0)	ID_CMD_STATE_OFF
Remplacer	NONE	ID_CMD_REPL_NONE
Remplacer	On (1)	ID_CMD_REPL_ON
Remplacer	OFF (0)	ID_CMD_REPL_OFF
Remplacer	Défaut	ID_CMD_REPL_DEF
Remplacer	Diff	ID_CMD_REPL_DIFF
Correction manuelle	NONE	ID_CMD_UPD_NONE
Correction manuelle	On (1)	ID_CMD_UPD_ON
Correction manuelle	OFF (0)	ID_CMD_UPD_OFF
Correction manuelle	Diff	ID_CMD_UPD_DIFF
Correction manuelle	Défaut	ID_CMD_UPD_DEF
Correction manuelle	DIRECT	ID_CMD_UPD_DIR
Bloquer	NONE	ID_CMD_BLOCK
Valider	NONE	ID_CMD_UNLOCK
Verrouiller	NONE	ID_CMD_LOCK

Révision	OFF (0)	ID_CMD_REV_OFF
Révision	On (1)	**ID_CMD_REV_ON**
Commande forcée	On (1)	ID_CMD_FORCE_ON
Commande forcée	Off (0)	ID_CMD_FORCE_OFF
Commande forcée	NONE	ID_CMD_FORCE_NONE

NAME OF THE MENU ITEMS OF THE CONTEXT MENU

AUTOMATIC CREATION

Entries that were created using `ID_CMD_AUTO` automatically get a name according to the following pattern: 'Action name' plus 'Limit value text of the switching direction'.

MANUAL CREATION FROM TABLE

If the menu entries are created from the table, for every action under 'Display - Text' a text must be defined for the entry in the context menu.

Names for the menu entries:

- Command
- Set value
- Status
- Replace
- Release
- Manual correction
- Block
- Lock
- Revision

ACTION TEXTS

Action	Text
Pulse command Switching command	Text from the limit value text, according to the switching direction.
Manual correction Replace	If a switching direction (other than 'None') is defined, the text from the limit value text according to the switching direction is displayed.
Status	'OFF' or 'ON', depending on the set switching direction
Revision	Text from the limit value text, according to the switching direction.
Others	No special action text is displayed.

Exemple

Displayed text for a switching command with defined limit value:

'Command: switching direction ON'

Informations

- All displayed texts are language switchable with the standard mechanisms. See also: Which texts are language switchable?
- All displayed menu entries are automatically sorted alphabetically.

The currently used command group is determined via the variable which is linked with the screen element. If no command group is assigned to the variable or if there is no response variable, the context menu is not displayed in the Runtime (an according error message is transferred to the Diagnosis Server).

Informations

The menu entries of the command processing are displayed depending on the command group. The menu entry is showed only when the connected action exists. Consequently, if the variable of the element is the command variable, only ***the*** *actions for the command variable plus the action* `Lock` *can be displayed. Actions for the response variable are hidden automatically.*

AVAILABILITY CONDITIONS

The menu entries are only released when the corresponding actions are executable. The following conditions are requirements:

- All menu entries are locked if the `NET_SEL` status bit of the response variable is active.
- All menu entries are locked, when the response variable could not be determined.
- All menu entries are locked, when the response variable has no value and could not get a value within 30 seconds.
- All menu entries are locked on the Web Client without write access.
- Menu entries are locked when there is no connection to the Primary Server.
- The menu entry connected to the `Release` action is locked when the `ALT_VAL` status bit of the action variable is not active.
- The menu entry connected to the `Replace` or `Revision` action, whose switching direction matches the value of the action variable, is locked.
- All menu entries, except the one which is connected with the action `Lock`, are locked, when a change lock is active for the response variable.
- When the `REVISION` status bit of the response variable is active, the actions `Set value`, `Replace`, `Correct`, and `Command` are locked.
- As long as a watchdog timer, an edge generation or an SBO is active for the command group, all menu entries are locked. This results from the fact that the `NET_SEL` status bit also stays active.

Macros du menu contextuel

Une macro est une séquence de caractères définie, qui est remplacée par un texte lors de la création d'éléments de menu dans le Runtime. Presque toutes les macros peuvent apparaître plusieurs fois par élément de menu. Par conséquent, elles contiennent également d'autres macros. Dans ce cas, la séquence d'expansion doit être prise en compte. Les macros ne sont pas sensibles à la casse lors de la configuration des menus. Si les macros contiennent une macro en tant que résultat, la macro doit être insérée en majuscules dans le résultat. L'entrée est identifiée par un caractère `$` en préfixe et suffixe.

La séquence d'expansion est de gauche à droite, avec la priorité suivante.

1. **ALL**
2. **$NOTE$**
3. **TAG**
4. **$REMA<État>$**
5. **$RDIR$**
6. **DIR**

7. **ACT**
8. **$NAME$**

Macro	Description
ALL	Entraîne la **dénomination de l'action : Direction de commutation**. Correspond à la combinaison de la macro $**ACT$** : **DIR**. **Remarque :** Si un menu contextuel est créé pour la gestion de commande, le texte par défaut est `$ALL$`, même si du texte a déjà été configuré pour le menu, mais le type d'action change en Gestion de commande.
$NOTE$	L'ensemble du texte, macro incluse, est interprété comme une remarque. Si le texte résultant est vide, la macro **ALL** est utilisée. Pour la dernière macro, la macro de la note est vérifiée à nouveau et le texte inséré à droite de celle-ci, macro incluse, est supprimé. Si le texte résultant est vide ou ne comprend que des espaces, l'élément de menu n'est pas inséré.
TAG	Remplacé par l'identification de la variable d'action. L'identification peut être traduite par la fonction de traduction en ligne. Si aucun caractère de traduction (@) n'est inclus, l'identification entière est surlignée aux fins de la traduction.
$REMA<Status>$	**<Status>** est un état ou une valeur limite de matrice de réaction, dont le texte est utilisé en tant que remplacement. Si l'état n'est pas présent, l'élément de menu n'est pas affiché. Le texte de valeur limite est traduit, du point de vue linguistique, en fonction du placement du caractère @ . L'état peut être un nombre compris entre -2^{31} et 2^{31-1}. L'utilisation de caractères précédant le texte et d'un préfixe est autorisée. Si des caractères ne pouvant être convertis en nombre sont inclus ou le nombre est situé hors de la zone spécifiée, l'élément de menu n'est pas affiché.
$RDIR$	Texte de la direction de commutation, dérivé de la matrice de réaction ou valeur limite, comme dans la macro **DIR**, à l'exception de : ▸ Action `Write set value direct` (Envoyer valeur prescrite directement) Le texte est dérivé de la matrice de réaction ou valeur limite de l'état, qui correspond à la valeur de la valeur prescrite devant être définie. ▸ Action `Status on` (État activé) et `Status off` (État désactivé) Le texte est dérivé de la matrice de réaction/valeur limite des états `activé` ou `désactivé`. ▸ Action `Correct direct (Corriger directement)` Le texte est dérivé de la matrice de réaction ou valeur limite de l'état, qui correspond à la valeur de la valeur prescrite devant être définie.
DIR	Direction de commutation de l'action.
ACT	Dénomination de l'action.
$NAME$	La macro `$NAME$` peut être utilisée pour créer des menus et fournit le

	contenu configuré de la propriété **Nom de l'action**, dont la langue peut également être basculée dans le Runtime, avec le caractère @.

ÉLÉMENTS DE MENU CRÉÉS AUTOMATIQUEMENT

Les éléments de menu créés automatiquement sont créés sous forme d'ID de menu avec ID_CMD_AUTO. Dans ce cas, les macros doivent toujours être utilisées ; dans le cas contraire, seul un élément de menu sera inséré.

COMPATIBILITÉ

Avant la version 6.51, le texte associé aux éléments de menu automatiques était ignoré. Lors de la conversion de projets créés avec des versions antérieures à la version 6.51, des macros $ALL$$NOTE$ sont automatiquement insérées avant le texte configuré. Par conséquent, les éléments de menu se comportent comme auparavant.

CHANGEMENT DE LANGUE EN LIGNE

L'intitulé de l'élément de menu dans la propriété **Texte** est traduit, au niveau linguistique, avant l'expansion de la macro, à partir du caractère `@`.

Remarque : Si, pour la balise **$TAGS$**, aucun indicateur de traduction (`@`) n'est présent, le texte entier est traduit.

Error messages when the context menu is called up

When menus are loaded in the Runtime environment, their content is checked for consistency. If an error occurs, corresponding error messages are issued for the **Diagnosis Viewer**. The following messages can appear:

Parameters	Description
Menu entry for command processing suppressed, because name is several times in the menu!	The menu already contains a menu entry with the name used in the command processing. Do not use that name for any other menu entries for the command processing.
Menu entry for command processing suppressed, because description is several times in the menu!	There is already a menu entry with the same description in the menu. Automatically created menu entries are not added, when a menu entry with the same description is already there.
Text for menu entry cannot be detected!	The description of an automatically created menu entry could not be determined. This most probably indicates a missing limit value text.
No command group linked to variable of the screen element!	The variable associated with the screen element has no command group or a no longer valid command group. According error messages are given during compiling.
Response variable does not exist!	The response variable used in the command group does not exist.
Select cannot be activated!	Status bit `NET_SEL` (8) could not be activated within the timeout.

Execution of actions via the context menu

After activation of a menu item for command processing, the assigned action is carried out. Execution via a menu activates the setting of the `NET_SEL` status bit in the first step. Only if this was successful is the execution of the actual action (a `switching command`, for example) started.

A command processing screen is then opened if one of the following criteria has been met:

- If the action to be executed is **Write set value**, `Status input` with input or `Correction`, the screen assigned to the action in the "Stage 1" step is opened. The status or the set value to be written can then be defined in the screen.
- If the action to be carried out is `lock`, the action-specific screen is called up with the `lock` step.
- If an active locking condition prevents execution, the screen configured in the **Unlocking** step for the action is called up. Execution is also prevented if **Select Before Operate** could not be activated without errors.
- If two-stage execution is configured for the action, the action-specific screen is called up in the "Stage 2" step.
- If no specific screen has been configured for the action, the screen that has been configured centrally for the command group is opened.

Informations

If none of the above-mentioned conditions are applicable, the action is executed immediately, without further operations.

Set value context menu

If the variable assigned to a screen element is linked to a command group, the writing of a set value is also handled by the command processing. The requirement for this is that a `Write set value` action is present with **État de la réponse/direction de commutation switching direction** in the Command Processing. If this is missing, the writing of the set value is not carried out.

Informations

An active interlocking condition prevents the writing of a set value.

Command conditions

Command groups contain both the definition of the switch actions and the definition of the command interlocking conditions. Command conditions are optimum parameters that can be defined application-specifically.

Each action within a command group can also be supplemented with **interlocking conditions**. These process-controlled interlockings prevent unwanted execution of actions, depending on the current process state.

The following three parts are significant in a command group for the command conditions:

- The action for which the conditions were defined and for which the internal interlockings are also applicable.
  The actions define which command is executed, on which variables these actions are applied and set the parameters for the internal interlockings.
- The `condition variables`, listed in the **Variables** node of the command group.
  These define which variables can be used in the command conditions.
- The `command conditions`, created per action.
  These conditions contain the formulas on which the condition variables are based. This syntax is the same as the definition of the formulas in the Formula Editor. The execution of commands can thus also be made dependent on the current process status.

Informations

Configured general interlockings have no influence on this check.

In addition to the command conditions, the following interlocking types are automatically checked before the action is executed:

- Internal interlocking conditions
- Topological interlocking conditions

INTERNAL INTERLOCKING CONDITIONS

These conditions are checked automatically before every action execution; the engineer cannot influence this. These Internal interlocking conditions (à la page 131) are predefined by the system and serve as plausibility checks.

Exemple

Internal interlocking is applicable if:

- The response variable is already selected in the zenon network (has set `NET_SEL` status from other network client).
- The response variable already has the desired value and the action was configured with **Nominal/actual comparison**.
- In the SBO, the Select was rejected by the PLC (status bits: `SE_870` + `COT_actcon(7)` + `N_CONF`).

TOPOLOGICAL INTERLOCKING CONDITIONS

These conditions result from the current topological status during Runtime. These conditions are defined in the 'Configuration of the topological interlockings (à la page 35)' settings of the project for **Automatic Line Coloring**.

Create command conditions

Any number of command conditions can be defined for every action. These conditions allow for an additional restriction of the ability to execute an action. These conditions are defined with formulas, in which you can use the variables from the active projects. The formula addresses the linked variables via the index in the condition.

Informations

The condition variable is automatically replaced if a '' is used in the definition.*

DEFINE CONDITION VARIABLE

In the first step, the variables or substitute names of the variables must be configured. These are used later for the formulas of the command conditions. If the defined conditions are fulfilled by the linked process variables, the user has the respective actions available during Runtime.

> **Informations**
>
> *Variables that are used in a formula are:*
>
> - In an interlocking condition
> - In the breaker tripping detection - **detection suppression**
>
> *Cannot be removed - from the **Variables** of the command group node.*

ENGINEERING

A command condition is defined using formulas. Conditions that are not met cause the action to not be executed in Runtime or to initially have to be unlocked by the user. The user must have the corresponding **Niveau d'autorisation pour déverrouiller** for this.

The following procedure is recommended for defining a command condition:

1. Go to the **Variables** node in the detail view of command input.
   a) Select the **Insert variable...** entry in the context menu
   b) The variable selection dialog is opened
2. Select a process variable.
   This variable serves as the basis for the formulas of the command conditions.

   Optional:
   Create a replaceable definition:
   a) To do this, close the variable selection dialog by clicking on the **Cancel** button.
      An empty definition is created.
   b) In the input field of the **Variable de verrouillage** property, enter the name with the ***** placeholder.
      Automatic substitution is configured as a result.
3. Go to an action that already exists.
4. Select the **New interlocking condition** entry in the context menu
5. Define the desired formula in the **Lien logique** property.

Internal interlocking conditions

With the help of the internal interlocking conditions the basic requirements for the action are checked (plausibility check). The results, or the addressing of an interlocking, are displayed in Runtime in the `command processing` screen in the **interlocking text** screen element.

Parameter	Description
Status already exists	The state which should be set equals the current value of the response variable. This check is only active if the action has been activated for the **Comparaison valeur nominale/actuelle** property. This interlocking can be unlocked, provided the user has authorization to do this - in accordance with the **Niveau d'autorisation pour déverrouiller** property of the action.
Internal error occurred	Command Processing cannot execute the check. This happens when the data type of the action variable is not allowed for this action. Example: `Pulse command on` action for string variables. This interlocking cannot be unlocked.
No interlocking object	**Command group** cannot be determined (configuration error). This interlocking cannot be unlocked.
Action not defined	Action to be executed could not be determined (egineering error). This interlocking is not unlockable.
Differences between local and global interlocking	Pulse command parameter not consistent (configuration error). This interlocking cannot be unlocked.
One or more values are not available	Value of condition variable: ▸ Not available (**interlocking code:** 14) ▸ violated - status bit INVALID (**interlocking code:** 15) This interlocking cannot be unlocked.
Locking administration not valid	The administration of the `lockings` could not be loaded or is invalid. This interlocking is not unlockable.
Variable locked for changes	Command Processing locked by response variable (status bit `M1`). This interlocking is not unlockable. Note: You can both lock or unlock in the command input screen with the `Lock` action type.
SBO rejected	The activation of the Select has been rejected by the PLC. This interlocking is not unlockable. Note: Only the Energy drivers signal the rejection of the Select - the action variable gets the status bits `COT_actcon(7)` + `N_CONF` + `SE_870`.
Timeout for SBO activation	Within the configured **Timeout**, no confirmation, either positive or negative has been received for Select activation. This interlocking is not unlockable. Note: Only the Energy drivers support Select activation - the action variable gets the status bits `COT_act(6)` + `SE_870`.
Timeout for SBO deactivation	Within the configured **Timeout**, no confirmation, either positive or

	negative, was received for the deactivation (Cancel to the Select). This interlocking is not unlockable.
Timeout for execution	There was no notice for finishing the action execution within the engineered **Timeout**. The `TIMEOUT` status bit is set for the response variable. This interlocking cannot be unlocked. Note: the **Tempo watchdog** of the **command group** determines what needs to be fulfilled before the action is completed.
SBO expired	The PLC has reported the expiration of the SBO activation. The second execution step will attempt to send a Select again. This interlocking cannot be unlocked. Note: The respective protocol determines whether a controller can report the Select timeout. If so, the Energy driver signalizes the process of the Select - the action variable gets the status bits `COT_actterm(10)` + `N_CONF` + `SE_870`.

Note: The numbers of the internal interlocking conditions are also shown in the **system driver** variable `[command] interlocking code`, if this variable has been created in the project.

Éditeur de formules

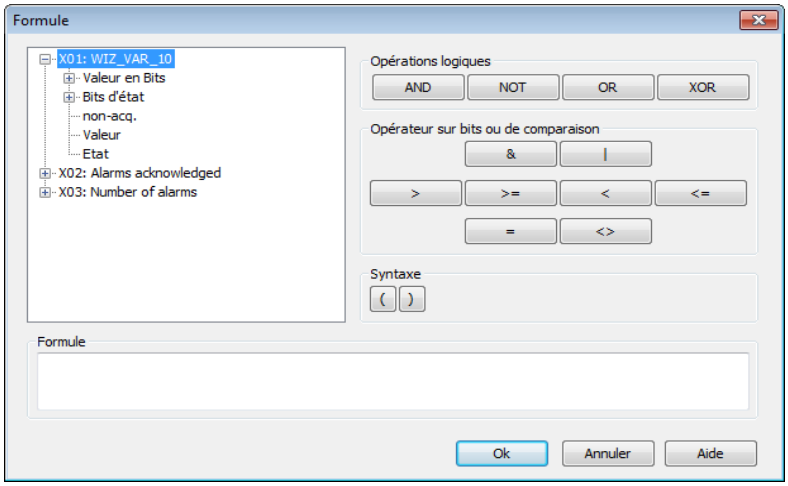
L'éditeur de formules fournit une assistance lors de la création de formules comportant des opérateurs logiques ou de comparaison avec un élément combiné, pour les verrouillages et les Gestion de commande. Si vous avez besoin de variables supplémentaires pour la formule, vous devez les créer dans la section **Variables de formule** de la fenêtre d'état en cliquant sur le bouton **Ajouter**. Les formules existantes sont affichées dans la liste d'état avec la lettre **F**.

Remarque concernant la saisie de points décimaux :

- Séparateur décimal : La virgule (,) est automatiquement converti en point (.) :
- les zéros situés après le point décimal sont automatiquement supprimés ; `23,000` devient automatiquement `23`.

CRÉER UNE FORMULE

Cliquez sur le bouton **Formule** dans la fenêtre d'état. L'éditeur de formule s'ouvre.



Sélectionnez les bits de votre formule dans la section gauche.

A droite, vous trouverez les opérateurs d'opérations logiques et de comparaison.

La formule créée est affichée dans la section **Formule**.

> **Informations**
>
> *Jusqu'à 99 variables peuvent être liées dans une formule. X01 à X99. La longueur de la formule ne doit pas dépasser 4096 caractères.*

SIGNIFICATION DES BITS :

Paramètre	Description
Valeur en bits	32 bits de valeurs (de 0 à 31) sont disponibles. Ils décrivent la variable, bit par bit. Pour les variables BINAIRE compte le bit `0` ; pour SINT et USINT, seuls comptent les bits compris entre `0 et 7`, etc. **Remarque :** la valeur fait référence à la valeur brute (plage de signal) des variables, et non à la plage de mesures convertie.
Bits d'état	Ici vous trouvez les bits d'états les plus couramment utilisés. Vous trouverez des informations concernant la définition et l'utilisation des bits d'état dans la Liste des bits d'état (à la page 137).
Non acquittée	**Non acquitté** est traité comme un bit d'état habituel. Mais il est montré séparément ici, car il n'appartient pas aux états classiques des variables.
Valeur et État	Dans les formules, toutes les valeurs (valeur des bits et bits d'état) sont traitées comme des valeurs binaires pouvant être liées de manière logique par des conditions AND et OR, etc. Les exceptions à cette règle sont la valeur totale et l'état global. Pour obtenir une expression booléenne, une condition logique ORed doit être appliquée à cette valeur totale, `de bit à bit` (à la page 140), avec une constante. Pour cela, on utilise l'opérateur `&`. For the result `0` (`FALSE`) of this logical ORing, we get the binary value `0` (`FALSE`), otherwise `1` (`TRUE`). Exemple : See the bitwise ORing example (à la page 140) chapter

Informations concernant

Les bits d'état NORM et N_NORM sont uniquement disponibles dans l'éditeur de formules et ne peuvent pas être configurés par l'intermédiaire de l'état.

Si d'autres paramètres externes à la formule sont définis pour l'état courant, ils sont "ajoutés" à la formule à l'aide d'un ET logique.

Reportez-vous à la section Exemples (à la page 142) pour obtenir des exemples.

Informations

Les formules comportant des valeurs X binaires et des liens au niveau des bits peuvent être utilisées avec 2 valeurs binaires maximum. Si d'autres valeurs sont requises, la liaison doit être établie sans valeurs X binaires.

Exemple :

X01.Value & X02.Value -> fonctionne

X01.Value & X02.Value & X03.Value -> ne fonctionne pas

Toutefois :

X01.00 AND X02.00 AND X03.00 AND X04.00 AND X05.00 -> fonctionne

Liste des bits d'état

Numéro de bit	Désignation abrégée	Nom long	Intitulé zenon Logic
0	M1	User status 1; for Command Processing: Type d'action "Bloquer" (à la page 112) ; Service Tracking (Main.chm::/IEC850.chm::/117281.htm) du driver IEC 850	_VSB_ST_M1
1	M2	État utilisateur 2	_VSB_ST_M2
2	M3	État utilisateur 3	_VSB_ST_M3
3	M4	État utilisateur 4	_VSB_ST_M4
4	M5	État utilisateur 5	_VSB_ST_M5
5	M6	État utilisateur 6	_VSB_ST_M6
6	M7	État utilisateur 7	_VSB_ST_M7
7	M8	État utilisateur 8	_VSB_ST_M8
8	NET_SEL	Sélectionné dans le réseau	_VSB_SELEC
9	REVISION	Révision	_VSB_REV
10	PROGRESS	En fonctionnement	_VSB_DIREC
11	TIMEOUT	Command "Timeout exceeded" (command runtime exceeded)	_VSB_RTE
12	MAN_VAL	Valeur manuelle	_VSB_MVALUE
13	M14	État utilisateur 14	_VSB_ST_14
14	M15	État utilisateur 15	_VSB_ST_15
15	M16	État utilisateur 16	_VSB_ST_16
16	GI	Requête générale	_VSB_GR
17	SPONT	Spontané	_VSB_SPONT
18	INVALID	Invalide	_VSB_I_BIT
19	T_STD_E	External standard time (standard time) **Caution:** up to version 7.50, this was the status bit T_CHG_A	_VSB_SUWI
20	OFF	Désactivé	_VSB_N_UPD
21	T_EXTERN	Temps réel - horodatage externe	_VSB_RT_E
22	T_INTERN	Informations d'horodatage internes	_VSB_RT_I
23	N_SORTAB	Non triable	_VSB_NSORT

24	FM_TR	MD_TR;Message de défaut du transfo	_VSB_DM_TR
25	RM_TR	Message de marche du transformateur	_VSB_RM_TR
26	INFO	Informations de la variable	_VSB_INFO
27	ALT_VAL	Valeur de remplacement	_VSB_AVALUE
28	RES28	Réservé à une utilisation interne (clignotement d'alarme)	_VSB_RES28
29	N_UPDATE	Not updated (zenon network)	_VSB_ACTUAL
30	T_STD	Heure d'hiver interne	_VSB_WINTER
31	RES31	Réservé à une utilisation interne (clignotement d'alarme)	_VSB_RES31
32	COT0	Cause de transmission bit 1	_VSB_TCB0
33	COT1	Cause de transmission bit 2	_VSB_TCB1
34	COT2	Cause de transmission bit 3	_VSB_TCB2
35	COT3	Cause de transmission bit 4	_VSB_TCB3
36	COT4	Cause de transmission bit 5	_VSB_TCB4
37	COT5	Cause de transmission bit 6	_VSB_TCB5
38	N_CONF	Confirmation négative de la commande par l'appareil (IEC 60870 [P/N])	_VSB_PN_BIT
39	TEST	Bit de test (IEC60870 [T])	_VSB_T_BIT
40	WR_ACK	Écriture reconnue	_VSB_WR_ACK
41	WR_SUC	Écriture réussie	_VSB_WR_SUC
42	NORM	NORM;État Normal	_VSB_NORM
43	N_NORM	État déviation normale	_VSB_ABNORM
44	BL_870	État IEC 60870 : `blocked`	_VSB_BL_BIT
45	SB_870	État IEC 60870 : `substituted`	_VSB_SP_BIT
46	NT_870	État IEC 60870 : `not topical`	_VSB_NT_BIT
47	OV_870	État IEC 60870 : `overflow`	_VSB_OV_BIT
48	SE_870	État IEC 60870 : `select`	_VSB_SE_BIT
49	T_INVAL	Horodatage externe invalide	non défini
50	CB_TRIP	Déclenchement de disjoncteur détecté	non défini
51	CB_TR_I	Détection de déclenchement de disjoncteur inactive	non défini

52	OR_DRV	Valeur non comprise dans la plage valide (IEC 61850)	non défini
53	T_UNSYNC	ClockNotSynchronized (IEC 61850)	non défini
54	PR_NR	Pas enregistré dans le Process Recorder	non défini
55	RES55	réservé	non défini
56	RES56	réservé	non défini
57	RES57	réservé	non défini
58	RES58	réservé	non défini
59	RES59	réservé	non défini
60	RES60	réservé	non défini
61	RES61	réservé	non défini
62	RES62	réservé	non défini
63	RES63	réservé	non défini

Informations

Dans les formules, tous les bits d'état sont disponibles. La disponibilité peut être réduite dans le cadre d'autres utilisations.

Pour plus de détails concernant la gestion des états, reportez-vous au chapitre Gestion d'états.

Opérateurs logiques

Liens logiques : seule la valeur logique '0' sera recherchée dans les variables ; si la valeur n'est pas égale à '0', elle sera considérée comme égale à '1'.

Contrairement aux formules bit, la portée technique peut être modifiée au moyen d'un facteur d'étirement -> (différent de '0' ou '1').

Opérateur	Signification
AND	ET logique
NOT	Négation
OR	OU logique
XOR	OU EXCLUSIF logique

Les opérateurs possèdent la priorité suivante dans le calcul de la formule :

Priorité	Opérateur
1	& (opérateur des formules bit (à la page 140))
2	NOT
3	AND
4	XOR/OR

Informations concernant

Jusqu'à 99 variables peuvent être liées dans une formule. X01 à X99.

Informations concernant

Les bits d'état NORM et N_NORM sont uniquement disponibles dans l'éditeur de formules et ne peuvent pas être configurés par l'intermédiaire de l'état.

Formules Bit

Les formules Bit comportent uniquement un état logique haut ou bas. Contrairement aux formules logiques, la valeur brute est déjà prédéfinie (**0**,1).

Opérateur	Description
&	AND
|	OR

Exemple : Comparaison de bits avec OR

Vous voulez savoir si l'un des bits d'état utilisateur 1 à 8 (M1 ... M8) de la variable X01 est défini :

FORMULE CLASSIQUE :

X01.M1 OR X01.M2 OR X01.M3 OR X01.M4 OR X01.M5 OR X01.M6 OR X01.M7 OR X01.M8.
Cette requête peut être considérablement simplifiée en utilisant la condition logique OR sur l'état général.

INTERROGATION OR LOGIQUE

X01.Status & 0xFF

La constante peut être saisie au format hexadécimal, comme ci-dessus.

`0xFF` en décimal vaut `255`, et correspond aux huit premiers bits d'état (en binaire, 11111111). Si l'un de ces bits est défini sur `1`, le résultat de l'application de la condition logique ORing au niveau des bits est `1` (True) ; dans le cas contraire, le résultat est `0` (False).

Si, par exemple, tous les bits d'états utilisateur sont à tester sauf le bit M7, la formule binaire serait : 10111111. Le bit 7 est sans intérêt, et est donc défini sur `0`. Ceci correspond à 0xBF en hexadécimal. L'expression de la formule est alors : **X01.Status & 0xBF**.

A la place d'une comparaison de bits avec OR à l'aide d'une constante, la valeur peut également être comparée directement à un nombre décimal. Si la comparaison est fausse, la valeur binaire sera égale à `0` (False) ; dans le cas contraire, elle prendra la valeur `1` (True).

Exemple :

Si vous voulez savoir si la valeur est égale à la constante `202` : La formule est alors :

X01.value = 202

Si la valeur est égale à la constante `202`, le résultat de la formule est `1` (`True`) ; dans le cas contraire, le résultat est `0` (`False`).

Remarque : avec le caractère OR (|), la condition logique OR au niveau des bits fonctionne de la même manière que dans cet exemple.

Opérateurs de comparaison

Les opérateurs de comparaison permettent de comparer directement deux valeurs numériques. Le résultat des comparaisons est une valeur binaire. "`0`" si la condition est fausse et "`1`" si la condition est vraie.

Opérateur	Description
<	Inférieur
>	supérieur
<=	Inférieur ou égal
>=	supérieur ou égal
=	Égal
<>	Différent de

À gauche et à droite de l'opérateur de comparaison doivent se trouver des valeurs complètes ou des états complets ; des bits simples ne peuvent pas être utilisés.

A droite d'un opérateur de comparaison, vous pouvez également utiliser une constante
Ces constantes sont saisies sous forme de valeurs hexadécimales ou de valeurs décimales dans l'élément combiné. Hexadecimal numbers are automatically converted to decimal numbers by clicking on **OK**. For example, `0x64` corresponds to the numerical value `100`.

Remarque : The combined element is not available in the **Batch Control** module.

Exemple

X01.valeur >= X02.valeur
Le résultat est 1, si la valeur de X01 est supérieure ou égale à la valeur de X02

X01.valeur = 0x64
Le résultat est 1, si la valeur de X01 est exactement égale à la valeur numérique 100 (= hex 0x64)

(X01.valeur = 0x64) OR (X01.valeur = 0x65)
Le résultat est 1, si la valeur de X01 est exactement égale à la valeur numérique 100 ou 101 (= hex 0x64 ou hex 0x65)

Exemples pour les formules

LIAISON LOGIQUE AND SIMPLE ENTRE DEUX VALEURS DE BITS

Exemple

Formule : X01.03 AND X02.03

Cette formule vaut TRUE si le **bit 3** de la variable `1` et le **bit 3** de la variable `2` sont tous deux égaux à `1`.

COMPARAISON D'UNE VALEUR ANALOGIQUE OU DE L'ÉTAT D'UNE VARIABLE

Exemple

(X01.Value > X02.Value)

COMPARAISON DE VALEURS ANALOGIQUES SUR UNE BASE LOGIQUE

Exemple

(X01.Value> X02.Value) AND (X01.Value = X02.Value)

COMPARAISON À L'AIDE DE BITS DE VALEUR ET DE BITS D'ÉTAT

Exemple

(X01.Value> X02.Value) AND (X01.Value = X02.Value) OR (X01.03 = X02.03)

COMPARAISON D'UNE VALEUR AVEC UNE VALEUR DÉCIMALE OU HEXADÉCIMALE

Exemple

Formule : (X01.Value = 111)

Formule : (X01.Value = 0x6F)

Si une valeur hexadécimale est utilisée, cette dernière est convertie en valeur décimale lorsque vous cliquez sur **OK**. Si une valeur décimale est saisie et confirmée, la valeur reste affichée au format décimal après la réouverture.

Informations concernant

Il n'est pas possible d'utiliser une virgule ou un point lors de la saisie d'une valeur.

4.3.4 Create menu

Command Processing can also be activated via a context menu. Context menus are created in the Editor using node **Menus** and are defined in the properties of the element they concern.

Généralement, il existe trois types d’entrées de menu :

Paramètre	Description
Type action	Définit le type d'action devant être exécuté via l'élément de menu correspondant dans le Runtime. Tous les types d'action ne sont pas disponibles dans le menu principal ; certains sont uniquement accessibles via le menu contextuel. ▸ Acquitter alarme (menu contextuel uniquement) ▸ Gestion de commande (menu contextuel uniquement) ▸ Acquitter clignotement (menu contextuel uniquement) ▸ Fonction ▸ Aide ▸ Aucune action ▸ Écrire valeur prescrite ▸ Macro VBA (menu contextuel uniquement)
Sous-menu	Ouvre un sous-menu dans le Runtime.
Séparateur	Une ligne horizontale divise les entrées de menu.

Souligner le texte : Saisissez un caractère & pour souligner les caractères suivants.

Planifier les entrées

Pour configurer un élément de menu dans le menu principal ou un menu contextuel :

1. Activez la cellule de menu correspondante
2. Dans propriétés, sélectionnez :
   - **Type action** : en fonction du type de menu.
     Voir également : Principaux types d'action des menus et Types d'action du menu contextuel.
   - **ID du menu**: Identifiant de l'entrée
     Remarque : Pour les entrées du module **Command Sequencer**, il existe des types prédéfinis, comportant des identifiants prédéfinis.
     D'autres informations à se sujet, ainsi qu'une liste de ces identifiants sont disponibles dans le manuel Energy Edition, au chapitre Gestion de commande via le menu contextuel (à la page 118).
   - **Texte** : intitulé clair des cellules du menu.

> **Attention**
>
> *La configuration de la propriété **Texte** doit être unique. Si deux textes identiques sont fournis, les autres éléments de menu portant le même nom ne seront pas affichés.*

You can find details on the definition on context menus for command processing in chapter menusCommand Processing.

4.3.5 Create Runtime files

When creating Runtime files for the command groups, a check for engineering errors is performed. In addition, there is validation for the correct substitution of the variable names.

For each variable that has a command group assigned, the command group for the operation in zenon Runtime is instanced. In each instance, only the actions that can be triggered by means of this variable are now included.

Exemple

The command group for the command variable now contains actions for the respective command variable.

Exception: the "`Lock`" action is also available with command input.

Informations

Changes to the configuration for variables require the project to be recompiled.

Substitution of variable names

In order to increase the reusabilty of the command group, there is the possibility to replace the variable references. Replacement is possible for the response, command and condition variable.

During the replacement, the placeholder (wild card *) is automatically replaced by the name of the variable that is assigned to the command processing.

EXAMPLE

The command group was configured in the **Nom de variable de réponse** property with the `*_RV` mask. In the project, the variables `xyz_RV`, `abcRV` and `bool_RV` are created and linked to the command group.

Variable	Part to be replaced	Result	Comment
xyz_RM	xyz	xyz_RM	Variable exists, assignment is possible
abcRM	<empty>	_RM	The mask is not correct, because the _ character is missing. The compiler will report an error. In Runtime, the operation of the abcRM does not trigger a command.
bool_RM	bool	bool_RM	Variable exists, assignment is possible

These rules are also applicable for the command variables.

Informations

When compiling the command group, the text that corresponds to the part to be replaced is searched as follows:

- The text is stipulated in comparison to the mask with the name of the response variable.
- Otherwise the text is defined by the action variable. In doing so, the first action appropriate for the variable is applied.
- If the text for substitution was determined correctly, the placeholder * is replaced by this text.

Please note the following points in relation to this when naming variables:

- The names of the variables and the mask should be selected in such a way that these can be clearly assigned.
- The names of the variables that are used for the response, command and condition variables should be able to be created from the same replacement text.
- If the response variable is replaced (was defined with a mask) but not the command variable, particular care should be taken to ensure that the command group that is created for the command variable also uses the expected response variable.
- Additional validation of the response variable for the command group ensures that it only contains actions whose action variable (for its compiled command groups) uses the same response variable. Incorrectly-configured actions create a warning and are removed during compilation.

Error while creating Runtime data

At the creation of the Runtime data for the command processing, an extensive validation is carried out concerning wrong engineering and not-available references.

Informations

After an `Error` the object the caused the error is not available during runtime.

If the command group has an `Error`, no command group is assigned to the variable. Consequently, during the Runtime, all user operations are locked.

A `Warning` is generated when the project would cause a problem but runs error-free.

In the error messages, the following placeholders are used:

<VERNAME>	Placeholder is replaced in the error message by the name of the command group.
<VERRM>	Placeholder is replaced in the error message by the name of the response variable.
<AUFVAR>	Placeholder is replaced in the error message by the name of the variable to which the command group is assigned.
<ACTVAR>	Placeholder is replaced in the error message by the name of the variable of the action.
<Actionname>	Placeholder is replaced in the error message by the description of the action.
<VARNAME>	Placeholder is replaced by the variable in the visualization.

The following error messages can occur during the creation of the Runtime files:

Message text	Description
<VERNAME>: Interlocking PV <VERRM> does not exist!	Condition variable for general interlocking not available.
Variable '<AUFVAR>' uses not existing command processing!	Variable uses a non-existing command processing.
(<AUFVAR>) command processing '<VERNAME>' contains no actions!	Command groups without action are not considered by the Runtime. This message can also be a follow-up error.
(<AUFVAR>) response variable '<VARRM>' does not use the command group '<VERNAME>'	A response variable using another command group is used. The response variable always has to be linked with the interlocking, which uses it as response variable.
(<AUFVAR>) response varibale '<VARRM>' for command processing '<VERNAME>' uses a driver without process linking!	The response variable must lie on a driver with process connection.
(<AUFVAR>Command processing '<VERNAME>' contains no actions after compiling!	A command group without actions does not make sense.
(<AUFVAR>) response variable '<VARRM>' of command processing '<VERNAME>' not available!	The used response variable is not present or marked as deleted.
(<AUFVAR>) command processing '<VERNAME>' uses screen '<Bild GUID>'(<BILDNAME>) which is not of the type Power!	This message is a warning. If a user action becomes necessary during execution, it cannot be performed.
(<AUFVAR>) command processing '<VERNAME>' uses not available screen '<Bild GUID>'!	The screen assigned to the command group does not exist.
(<AUFVAR>) Replaced action variable '<ACTVAR>' for action '<Action name>' of command processing '<VERNAME>' not available!	The action variable, after a replacement, is not present or marked as deleted.
(<AUFVAR>) action '<actionname>' of the command'<VERNAME>' uses the not existing variable '<ACTVAR>'	The action uses a varibale which is not present in the project or marked as deleted.
(<AUFVAR>) action variable '<ACTVAR>' for action '<Actionname>' of command processing '<VERNAME>' uses a driver without process connection!	The variable assigned to an action must not lie on an internal driver.
<VERNAME>(<AUFVAR>): Aktion '<Actionname>' already exists!	The following actions may only be configured once per action variable and command group: ▸ Switching command with the same command processing status. ▸ Correction with the same switching direction. ▸ Replacing with the same switching direction. ▸ Revision with the same switching direction.

	▸ Pulse command ▸ Setpoint input ▸ Release ▸ Block ▸ Lock
<VERNAME>(<AUFVAR>): Action '<Actionname>': Pulse and switching command not possible for the same command variable!	Pulse and switching command must not be used together.
(<AUFVAR>) Action '<Actionname>' of the command processing '<VERNAME>' uses screen '<Bild GUID>('<Bildname>') which is not of type Power!	This message is a warning. No user actions will be possible.
(<AUFVAR>) Action '<Actionname>' of the command processing'<VERNAME>' uses not existing screen '<Bild GUID>'!	The action is assigned to a non-exising screen.
<VERNAME>(<AUFVAR>): Interlocking PV '<VARNAME>' does not exist!	Replaced condition variable does not exist.
(<AUFVAR>) Variable '<VARNAME>' of action interlocking condition of the command processing '<VERNAME>' does not exist!	Variable of the interlocking condition does not exist.
(<AUFVAR>) action variable '<VARNAME>' for action '<Actionname>' of command group '<VERNAME>' uses another command group!	The action variables used for a command group may only be connected to no command group or to the command group in which they are used.
(<AUFVAR>) command variable <ACTVAR> does not have a validly compiled interlocking! Action <Actionname> removed.	The action variable used in the action has no compiled command group. This message can also be a follow-up error.
(<AUFVAR>) command variable <ACTVAR> uses response variable <VARNAME>! Action <Actionname> removed.	The compiled command group of the response variable contains actions with action variables which do not use the same response variable as <AUFVAR>. **Note**: There must not be any actions of response variables changing other response variables.

4.4 Operating during Runtime

A watchdog timer is automatically carried out in the background if a used enters commands in Runtime.

4.4.1 Execution of a command

This description for the procedure of a command with the command processing is applicable for the following action types of the Command Processing:

- Command (à la page 104)
- Auto/Remote command (à la page 105)
- Mandatory command (only in part) (à la page 106)
- Setpoint input (à la page 107)

> **Informations**
>
> You can find information such as the execution of a command that influences the display and availability of the control elements in a `command input` screen in the Process in the command processing screen (à la page 160) chapter.

The procedure of a command depends on the following parameters:

VARIABLE PROPERTIES

- **Select Before Operate** (SBO) inactive:
  In this case, a separate Select command is executed by an action.
  Note: A driver (for example: DNP3_NG or IEC850) can nevertheless execute a Select automatically. This has no effect on the command processing however. The Command Processing will, in this case, react to an unsuccessful Select in the same way as an unsuccessful Operate/Execute .
- **Select Before Operate** (SBO) active:
  In this case, a Select (`COT_act(6), SE`) is always forwarded to the driver by the command processing. In doing so, there is a wait - regardless of the type of **Tempo watchdog** configured - until the complete command process (including Operate) has been completed.

  Attention: The **Select Before Operate** variable property has corresponding effects on **Tempo watchdog**.
  If **Select Before Operate** is activated, the action buttons (and context menu entries) are deactivated in runtime monitoring for each configuration until `COT` contains the value `COT_accterm(10)`.
  This also applies if "`none`" or "`via response variable`" are configured for the watchdog timer.
  If no `COT_accterm` is received, only a `TIMEOUT` status bit is set.
- **Annuler 'Operate'** active:
  If activated, with command input, after an Operate, a Cancel can also be sent to the controller, not just after a Select. Both the execution of the command in the controller and ongoing runtime monitoring can thus be ended early. The property is automatically treated like activated **Le délai d'attente peut être annulé** in the action.

Info

If a variable is configured with an active **Select Before Operate** and the driver does not support a **COT** , then no reaction to a sent Select will be received from the PLC. Once the configured **Timeout** has expired, the command processing screen will inform you of the "**Timeout on SBO activation**" interlocking condition.

TYPE OF WATCHDOG TIMER:

The **Tempo watchdog** property in the command group determines which conditions need to be met in order to conclude the action as successful. In addition, the **Tempo watchdog** property determines how long the command input remains active in order to possibly send a deactivation (Cancel) to the PLC, to update the status bits of the response variable and to guarantee forwarding in the `auto/remote command`.

- `None`
  - without **Select Before Operate**:
    Direct command to the controller ("fire & forget") - sends the command to the controller and does not wait for a response.
  - with **Select Before Operate**:
    There is a wait in the background until the COT process has been completed in full. No `TIMEOUT` status bit is set, even if a **Timeout** has expired.
- `Via cause of transmission only (COT)`
  The action has been successfully completed after a complete COT process - if `COT_actterm(10)` has been received.
- `Only via response variable (RV)`
  The action has been successfully completed if the value of the response variable corresponds to the **response status/switching direction** in the action.
  - with **Select Before Operate**:
    There is a wait in the background until the complete COT process has been completed. Even if a **Timeout** has expired, no `TIMEOUT` status bit is set.
- `Via RV and COT`
  The action is successfully completed if both of the above-described conditions have been met.

Note: You can find further details in the runtime monitoring (à la page 155) chapter.

PROPERTIES IN THE COMMAND PROCESSING ACTION:

- `One-step`:
  After a successful Select , the command processing automatically transfers the Operate/Execute to the driver.

- `Two-step`:
  A successful Select activates the buttons `Execute 2nd Step` and `Cancel` in the command processing screen:
  - If `Execute 2nd Step` is clicked on, the Command Processing transfers an Operate/Execute (`COT_act(6)` without `SE`) to the driver.
  - If `Cancel` is clicked on, a Cancel(`COT_deact(8)`, `SE`) is forwarded to the driver via command processing.
- **Timeout**:
  The configured value states how long is waited for a response from the PLC. Respectively:
  - After writing an Select command.
  - After writing an Execute command.
  - After writing an Cancel command.
- **Le délai d'attente peut être annulé**:
  If this property has been activated, the user can also cancel a **Tempo watchdog** that is still running, for example during an attempt to synchronize the frequencies of the electrical grids in the controller. The **Cancel** button in the command screen remains operable whilst the Execute command is executed. Depending on the configuration of the **Annuler 'Operate'** variable property, it is also possible to inform the PLC of the cancellation.
  Note: not all Energy protocols offer a technical possibility to cancel an Operate. It is thus not possible for every Energy driver to forward a Cancel to the PLC.

Select before Operate

Select before Operate is a procedure in Energy protocols (such as DNP3, IEC61850 or IEC60870-101/104 for example). A "reservation/pre-assignment", the `Select` command, is first sent to the PLC. Only if this pre-assignment of the PLC is successful is the corresponding `Execute` command sent via the driver (using a command).

zenon Energy drivers update the following status bits in the process (by means of command variables):

- `SE_870, COTx`
- With the corresponding `N_CONF` configuration.

These status bits and NET-SEL cannot be configured by the person configuring the project. The status bits are set by the corresponding driver of zenon modules.

The **Tempo watchdog** property only has an influence on the Execute command. The Select command is not influenced by **Tempo watchdog**. After sending a Select, the command action waits for the time period configured in the **Timeout** property **Options**properties group in the command action).

In doing so, this can lead to the following dependencies:

- SPS does not react (à la page 153)

- SPS reacts negatively (à la page 154)
- SPS reacts positively (à la page 154)
  - User does not send an Execute or Cancel (à la page 155)

CANCELING A SELECT

For canceling (Cancel) a Select - regardless of the type of **Tempo watchdog** engineering - the following is applicable:

1. If the **Select Before Operate** property is activated for the action variable, a **Select** for **two-stage** commands (**A deux étapes** command action property activated) can be canceled by a user. After a successful Select (COT=7, SE), the Command Processing can send a Deactivation (COT=8, SE) if the **Cancel** button is clicked.
2. There is then a wait for a response from the PLC or a **Timeout**.
   The action ends:
   - if the configured **timeout time** has expired
   - if the PLC confirms the cancellation (COT=9, any SE or PN).
     The receipt of a COT=9 discards the PROGRESS status bit.
3. The measurement time starts when the user clicks the **Cancel** button. The time period that has expired - whilst the action waits for a Select - does not influence the time period in which the action waits for a confirmation of the cancellation.
4. The response variable does not receive a TIMEOUT status bit.
5. The same action always applies, regardless of the configuration of the **Tempo watchdog** property.

Note: the `PN` bit - Positive(0)/Negative(1) - is reflected on the status bit `N_CONF` of the action variable.

SBO - no reaction from the PLC

The command processing first sends a Select to the driver. All buttons in the `command processing` screen are grayed out while the action waits for a reaction from the PLC.

Information is shown in the command processing screen once the **Timeout** action has expired. This message is displayed in the field of the interlocking text. `Internal interlocking condition "Timeout for SBO activation".`

In this case, only the **Cancel** button is available. All other buttons of the Command Processing screen are still grayed out.

Entries in the context menu are not available.

Note: The response variable contains neither a `TIMEOUT` nor a `PROGRESS` status bit.
The same action applies for Select , regardless of which value is configured for **Tempo watchdog** . The process is the same for two-step actions and one-step actions.

SBO - negative reaction from the PLC

The command processing triggers the driver to send a Select (`COT_act(6) + SE`) . If a negative response is received by the PLC (`COT_actcon(7) + SE + PN`=1), waiting is no longer carried out.

- The `PROGRESS` bit is removed by clicking on the **Cancel** button. The response variable does not receive a `TIMEOUT` status bit in the process.
- If the action receives a negative response to Select, this information is displayed in the field of the interlocking text: `internal interlocking condition "SBO rejected"`.
  In this case, only the **Cancel** button is available. All other buttons of the Command Processing screen are grayed out.

Entries in the context menu are also not available.

The same action applies to Select , regardless of which value is configured for **Tempo watchdog**. The process is the same for two-step actions and one-step actions.

Note: the `PN` bit - Positive(0)/Negative(1) - is reflected on the status bit `N_CONF` of the action variable.

SBO - positive reaction from the PLC

If the PLC reacts positively to a Select (`COT_actcon(7) + SE`) (thus a confirmed Select), the action goes to the next step => execution

- The command processing ends the waiting for a Select.
- The response variable has its `PROGRESS` status bit set. However this is only if the current value of the response variable is different to the value of the command.

The following applies once the Select has been confirmed:

a) One-step commands:
   The command processing automatically sends an Operate/Execute (`COT_act(6)`, no SE) to the PLC.

b) Two-stage actions:
   The **Execute 2nd Stage** and **Cancel** become active in the command screen. If a user confirms the **Execute 2nd step** button, the command processing sends an Operate/Execute to the PLC.

The `PROGRESS` bit is reset if the PLC confirms the Execute (`COT_actcon(7)`) or after expiry of the following **Tempo watchdog**.

SBO - positive reaction from the PLC but the user does not send an Execute or Cancel

If the user, after a successful Select , triggers neither an Execute nor a Cancel , there is a wait for user interaction with no time limitation. It can thus happen - provided the PLC supports **Select Timeout** - that this time expires. If this happens, the PLC sends a Select-Termination (`COT=10, SE, PN=1`). The command processing reacts to the Select-Termination received so that - if the user does in fact trigger an Execute - the command automatically sends another Select first.
Once the **Cancel** button in a command processing screen has been clicked on, a Select or Cancel is not sent to the PLC again.

Note: Is only relevant for two-step actions.

Watchdog timer

The simplest runtime monitoring, not envisaged for Energy drivers, is carried out if:

- The **Tempo watchdog** property in the command group is `none` or `only via response variable (RV)`;
- Neither the **Select Before Operate** property (SBO) nor **Annuler 'Operate'** has been activated.

For this configuration, the runtime monitoring checks the interlockings and writes a command to the controller. After this, the runtime monitoring waits to see whether the response variable changes the value according to the command.

In conjunction with Energy drivers (for **Select Before Operate** or `COT` process), the cause of transmission (COT) is used to exchange information between zenon and the controller about whether a command is to be written or whether writing was successful. With Energy drivers, the action variable gets a `COT` according to the stage of the command. In the background, the command processing then checks to see if the response variable then changes its value and if the `COT` action variable changes according to the command.

Note: Value changes of the response variable will only be taken into account after `COT_act(6)`.

Informations

`COTx` Status bits result in a value. This value can be evaluated in Runtime - just like all other status bits - using multi numeric or multi binary reaction matrices.

Note: *`COT` is supported not only by IEC870, but also by some other Energy drivers - different versions thereof. Some drivers support COT although the protocol itself does not contain COT (e.g. IEC850, DNP3). You can find details in the corresponding driver documentation.*

RUNTIME MONITORING AND INTERLOCKING CONDITIONS OF THE ACTION:

Interlockings are checked during runtime monitoring:

1. For direct execution (= without **Select Before Operate**).

2. When the Select is activated (= before Select).
3. After the Select OK - before Execute.

- The current command action shows information about active interlockings in the `interlocking text` control element. The user must unlock these. To do this, the user must have the corresponding user permissions. In addition, the interlocking must also be configured as unlockable. Otherwise this action can only be canceled.
- If this action is canceled by a configured interlocking, no command is sent to the PLC.
- If a Select has already been sent, cancellation is automatic. In this case, a deactivation (Cancel) is sent.
- The response variable does not receive a `TIMEOUT` bit in the process. If a `PROGRESS` bit is already set, this is reset.

WITH ONE-STEP CONFIGURATION OF THE ACTION:

If there is an interlocking in the first stage, the `interlocking text` is displayed and there is a wait for a reaction from the user. If the user selected **On** or **Off**, the **Confirm** button will be active. The value is then sent to the PLC. If the user clicks on **Cancel** and a Select has already been carried out, the action sends a deactivation (Cancel).

WITH TWO-STEP CONFIGURATION OF THE ACTION:

The two-step action checks the interlocking and provides a message during the first step:

- Direct execution - no **Select Before Operate** :
  Applicable for the moment when the first button is clicked
- Active **Select Before Operate**:
  Applicable for the moment if the confirmation for Select is received or the **Timeout** for Select has expired.

*By clicking on the button to unlock the interlocking, the **Execute 2nd step** button becomes available. If the button is clicked on, an Execute is sent to the PLC. If **Cancel** has been clicked on and a Select has already been sent, the action sends a deactivation (`COT_deact(8)+SE`) to the PLC.*

*If there was no outstanding interlocking for the first stage before the conditions have changed and before the user has carried out a **Confirm** by clicking, the action will check the interlocking conditions again. If there are then still interlockings pending, a message is displayed and there is a wait for a cancellation from the user.*

Note: *In this case, the interlocking cannot be unlocked by a user.*

Informations

Single-step actions have the same action in all scenarios.

RUNTIME MONITORING AND NEGATIVE RESPONSES FROM THE PLC:

- `Execute` negative
  During watchdog timer, the PLC can react negatively to an Execute/Operate `(COT_actcon(7) + PN)`. If the negative ration to an Execute was received, runtime monitoring is ended.
- `Execute Termination` negative
  If Execution Termination `(COT_actterm(10) + PN)` is reported as negative by the PLC, runtime monitoring will no longer wait for a value change to the response variable and ends immediately.

The `PROGRESS` status bit is reset. The `TIMEOUT` status bit is not set for the response variable.

Note: the `PN` bit - Positive(0)/Negative(1) - is reflected on the status bit `N_CONF` of the action variable.

Cancellation of runtime monitoring

If the **Le délai d'attente peut être annulé** property has been activated for a command processing action, the user can cancel runtime monitoring - the second stage of execution (Execute) - with the "**Cancel**" button.

In doing so, the command processing sends the cancellation to the driver. The driver (the **IEC850 driver** for example) only forwards a Cancel Request to the PLC if the **Annuler 'Operate'** property has been activated during configuration. You can find this property in the **Écrire valeur prescrite** variable property group.

If the **Annuler 'Operate'** property is activated for the command variable, the user can also cancel the runtime monitoring with the **Cancel** button in the **command input** screen. It is automatically considered an activated **Le délai d'attente peut être annulé**.

Note: Not all drivers - that support the cancellation of a Select - also support the cancellation of an Execute. You can find further information in the respective driver documentation.

Runtime monitoring via response variable only (RV)

The **Tempo watchdog** via the response variable is the most-used type of runtime checking. This reacts to a change of the value of the response variable. The value which is expected from the response variable as a result of the command defined in the property **État de la réponse/direction de commutation** (`on/off`).

Negative responses from the PLC `(COT_actcon(7) + PN)` end runtime monitoring.

Note: Changes to the value of the response variable are only taken into account after `COT_act(6)` of the action variable. Early value changes of the response variable are ignored. This can occur, for example, if there are already value changes for the response variable after a Select without an Operate/Execute being sent.

Runtime monitoring via cause of transmission only (COT)

With **Tempo watchdog** `via COT only`, runtime monitoring does not react to the value of the response variable (RV) but only to the cause of transmission - the status bits `COTx` status bits of the action variable.

EXAMPLE WITHOUT SBO:

The process in detail:

1. The value and `COT_act(6)` are sent to the action variable.
2. The PROGRESS status bit is sent to the response variable.
3. If the controller receives the value `COT_act`, there is a wait for the subsequent values `COT_actcon(7)` and `COT_actterm(10)`.
4. End of the process:
   a) The process is ended if `COT_actterm` has been received.
   b) If, in the configured timeout time, no `COT_actterm` has been received, then:
   - the process is ended and
   - the TIMEOUT status bit of the response variable is activated.
   Note: You configure this time in the **Timeout** property for the command action.
5. The PROGRESS status bit is reset.

Runtime monitoring via COT (cause of transmission) and RV (response variable)

With **Tempo watchdog** `via COT and RV`, the runtime monitoring reacts to the value of the response variable (RV) and to the cause of transfer - the `COTx` status bits of the action variable.

EXAMPLE WITHOUT SBO:

The process in detail:

1. The value and `COT_act(6)` are sent to the action variable.
2. The PROGRESS status bit is sent to the response variable.
3. If the controller receives the value `COT_act`, there is a wait for the subsequent values `COT_actcon(7)` or `COT_actterm (10)`.
4. End of the process:
   a) The process is ended if both conditions have been met:

- `COT_actterm` was received

  and

- The value of the response variables corresponds to the **switching direction** (**État de la réponse/direction de commutation** property).
  It does not matter which of the two conditions is met first. As soon as both of them are fulfilled, the procedure will be terminated.

a) If only one or none of the above conditions from item 4a is met within the configured **Timeout**, then:
- the process is ended and will be terminated and
- the TIMEOUT status bit of the response variable is activated.

5. The PROGRESS status bit is reset.

4.4.2 Screen type Command Processing

A **command processing** screen allows control in Runtime and an overview of the command processing. The command processing can be controlled via buttons. Templates with different appearances are provided.

DÉVELOPPEMENT

Étapes de création du synoptique :

1. Créez un nouveau synoptique :

   Dans la barre d'outils ou le menu contextuel du nœud **Synoptiques**, sélectionnez la commande **Nouveau synoptique**.
   Un synoptique vide `Standard` est créé.

2. Modifiez les propriétés du synoptique :

   a) Nommez le synoptique dans la propriété **Nom**.

   b) Select `Command Processing` in the **Type de synoptique** property.

   c) Sélectionnez le cadre souhaité dans la propriété **Gabarit**.

3. Configurez le contenu du synoptique :

   a) Sélectionnez l'option de menu **Éléments de contrôle** dans la barre de menus

   b) Sélectionnez `Insérer un modèle` dans la liste déroulante.
   La boîte de dialogue de sélection de mises en forme prédéfinies s'affiche à l'écran. Certains éléments de contrôle sont insérés dans le synoptique à des positions prédéfinies.

c) Supprimez les éléments superflus du synoptique.

d) Si nécessaire, sélectionnez des éléments supplémentaires dans la liste déroulante **Éléments**. Placez-les aux emplacements souhaités sur le synoptique.

4. Créez une fonction d'appel de synoptique.

Process in the Command Processing screen

The `command input` screen has a multi-stage process, which consists of the following steps:

1. Initialization
2. Step 1
   Specifying the action
3. Unlocking
   Checking the interlockings and requesting a Select
4. Step 2
   Executing the command
5. Waiting until execution is complete

And an independent step - "lock"

Depending on the step in which the process is currently in, the control elements are updated, opened or hidden in the screen.

Note: These steps are shown in the `[command input] screen step` **system driver** variable.

INITIALIZATION

This step installs the internal process administration. There is then an immediate switch to "Step 1", without user interaction.

Initialization is executed regardless of the reason for switching when the screen is opened:

- The response variable and the command variable (if this is the switch variable) is requested by the driver. In doing so, there is an evaluation to see whether the data points exist. A LOG entry is generated in the event of an error.
- The `NET_SEL` status bit for the response variable is activated.
- Settings for the display are set:
  - The **Synoptique modal** properties and the title of the screen - from the **Titre de synoptique depuis la variable de réponse** property - are set accordingly.
  - The buttons without an assigned action are hidden.
    You can read more about this in the Blocked or locked elements (à la page 163).

STEP 1

The action to be executed can already be specified in this step. This can then be the case if the user has selected the action from a context menu. Even if the command screen is still open due to a previous (already-executed) action, it remains specified. If no action has been specified yet, the action to be executed is determined automatically. The result of this determination of the action to be executed is shown in the `Active action` control element.

The switch to the next step, "unlocking" is carried out:

- By clicking on an action button:
  The user thus specifies the action to execute manually.
  Example: **On/Off** (**Fonction spécifique au type de synoptique**: `Action N`).
- By clicking on the **Execute** button (**Fonction spécifique au type de synoptique**: `Execute`);
  The button relates to the action that is currently specified.

The action to be executed is determined by calling up the command input screen (by clicking on a screen element, for example). The action to be executed is then not yet specified, because the command input screen can include several actions. The result of the determination depends on the control elements configured in the screen and the actions that are linked to the switch variable.

The determination is carried out in accordance with the following criteria:

1. `Setpoint input`
   If a control element has been configured for the entry of `action variable value` or `action variable value (slider)` in the screen and the `setpoint input [set value]` action exists for the switch variable.
2. `Status input`
   If a `set status` control element has been configured in the screen and there is the `status input [none]` action for the switching variable.
3. `Lock`
   If the screen has been opened using the **Appel de synoptique** function with the **initial step** `lock` property and the command group contains the `lock` action.
4. If no action could be determined from any of the tests, the screen is opened without an active action. The control elements for entry and also the **Execute** button are then deactivated.

Note: As a result, you can call up certain `command input` screens you have configured yourself for certain actions and the focus is placed on the respective relevant screen.

UNLOCKING

The step is activated directly when operated via the **context menu**.

In this step, a check is carried out to see whether there are interlockings active. Only if no active interlocking has been detected is a switch to "Step 2" possible.

At the same time, there is a Select request to the controller if the **Select Before Operate** property is active for the command variable. There is a wait until the PLC confirms the Select or the **Timeout** defined for the action has expired. In the event of a timeout or rejection of the Select, a corresponding, non-unlockable interlocking message is shown.

If there are interlocking conditions, the step is only left once all interlockings have been unlocked. If logging in the CEL has not been deactivated for the action (**Supprimer l'entrée de la liste d'événements** property), unlocking is recorded in the CEL.

When the **Cancel** button is clicked on. (**Fonction spécifique au type de synoptique**: `Cancel`) a switch back to "Step 1" is made. This can be necessary if there is a non-unlockable interlocking active. A Select that is already active is then deactivated (Cancel).

STEP 2

In this step, the fundamental command (Operate) is sent to the controller.

If the **Confirm** (**Fonction spécifique au type de synoptique**: `Execute 2nd stage` button is clicked on and

- there is a valid Select ,
- no interlocking condition active,

execution of the action is started. If there is a signing configured with this button, execution is started after successful signing.

Status bits of the response variable:

- The `PROGRESS` bit is activated
- The `TIMEOUT` bit is deactivated if a previous action had activated this status bit.

STEP 2 FOR TWO-STAGE ACTIONS

With two-stage execution, there is always a wait for confirmation from a user. Only once the user has clicked in the **Confirm** button is another check for interlockings carried out. If interlocking conditions have become active in the meantime, execution is not started. In this case, there is a return to "Step 1" by clicking on the **Cancel** button and the command input is carried out again.

If the `NET_SEL` bit is no longer active (on the network client after redundancy switching for example), an error message is logged. The action is not executed.

If the Select in the controller has expired (command variable has received the status bits `SE_870` + `COT_actterm`(10) + `N_CONF` in this case) a Select is requested from the PLC again. The action is carried out after a positive confirmation. If, in the process, the repeated SBO activation fails, a message is shown accordingly.

If, instead of the **Confirm** button, the **Cancel** button is clicked on, there is a switch back to "Step 1". A Select that is already active is then deactivated (Cancel).

WAITING UNTIL EXECUTION IS COMPLETE

In this step, there is a wait until the action has been completed in full. The duration of the execution depends on the driver (or rather the controller) and the configuration. Negative responses from the controller cancel execution; a negative response to an Operate, for example.

Execution waits until all of the following points have been fulfilled:

- Configured **Select Before Operate** for the command variable:
  If Select has been confirmed positively by the PLC, there is a wait until Termination.
- Conclusion of the configured **Tempo watchdog** (if not all configured with `none`).
- Conclusion of edge generation for the `pulse command` action.
  Note: The **Durée d'impulsion** is not executed if **Select Before Operate** has been activated for the command variable.

If execution has been completed and the **Fermer automatiquement** setting has been configured for the action, the screen is closed. Otherwise there is a switch to the "initialization" stage.

If runtime monitoring has ended, the following applies for the status bits of the response variable:

- The `PROGRESS` bit is deactivated
- The `TIMEOUT` is activated if runtime monitoring has been ended with an error.

LOCK

This step is activated if the screen has been called up with the **Appel de synoptique** function and with the **initial step** `lock` in the process. What is special about this step is that some control elements are not displayed. Only the control elements from the response variable and lock groups are visible.

Note: the locking/unlocking functionality is also available in other steps if corresponding control elements have been placed in the screen.

If, for the `lock` action, the **Fermer automatiquement** has been configured, the screen is closed once the **Lock** or **Unlock** buttons are clicked on.

This step cannot be switched to another step.

Blocked or locked elements

Some requirements must be met in order for the entries in the **context menu** and the control elements in the screen to become active. Because these mostly concern several elements, a summary of these is documented here.

the entries in the context menu and the control elements in the opened screen are locked up to **Close** if:

- Another screen is already the current owner of the active `NET_SEL` status bit of the response variable (through a network client, for example)
- No command processing was configured for the add-on variable
- the response variable does not exist
- The response variable has not received a value yet
- The `INVALID` or `OFF` status is active for the selected variable
  Exception: `forced command`, `replace`
- The response variable was locked for command processing:
  - the status bit `M1`, i.e. the command lock, of the response variable was set

  Exception: the control elements for the `Lock` action
- An action is running for the action variable and runtime monitoring has not yet been completed
  Exception: The **Cancel** button can be active here - regardless of the configuration.
- There is a wait for the SBO confirmation from the Select (SBO)
- the data of the lock are being transmitted
- the data of the lock are invalid
- the currently-registered user does not have the necessary authorization levels.

UNLOCK

Unlocking is only possible if:

- The authorization level of the user who is logged in allows execution
- In accordance with the configuration of the interlocking:
  - For command condition:
    if property **Déverrouillable** was activated
  - for topological interlocking
    If, in the **Configuration CAL**, the `unlockable` status has been selected.

LOCK CONTEXT MENU ACTION

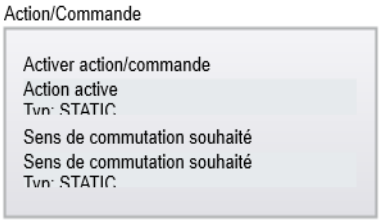
If the screen is called up with the `Lock` command action, only the following control elements are visible in the screen:

- Response variable (name)
- Response variable (identification)
- Response variable (value)
- Response variable (status)
- Response variable (measuring unit)

- Lock
- User
- Apply lock code

Other control elements in the screen are hidden.

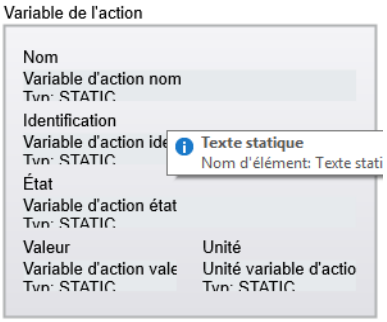
Control elements - group: Action/command



Control element	Description
Active action/command	Type of activated command action (**Type action**). Examples: `Switching command`, `Check response value`).
Switching direction	Value for the **État de la réponse/direction de commutation** property configured for the active action. Depending on the active action, the following text is shown: - Command - Revision - Manual correction - Replace Text from limit value, depending on switching direction. - Status: On or Off - Other: empty

Note: The information is also displayed in the `[command] identification of the action` and `[command] parameter of the action` **system driver** variables.

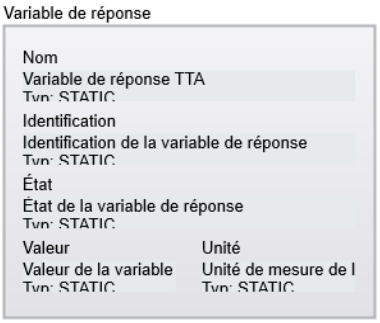
Éléments de contrôle - groupe : Variable de l’action



Élément de contrôle	Description
Nom	Nom variable d'action.
Identification	Identification variable d'action.
État	Contient la description abrégée des bits d'état pour la variable d'action. Exemple : ▶ Bits pour le paramètre COT ▶ État SE_870 durant la commande Select ▶ état N_CONF (Bit PN) en cas de réponse négative de l'automate **Remarque :** Inclut les bits d'état planifiés de la variable de réponse de l'action `Envoi état`.
Valeur	Valeur actuelle de la variable d'action ou champ de saisie de l'action de gestion de commande `Saisir point de consigne`. **Remarque :** Avec des pilotes qui permettent également la lecture des variables de commande du PLC, cette valeur change pendant le processus de l'action, d'une valeur existante à la valeur actuelle. L'affichage de la valeur est uniquement actualisé avec COT=7 (COT_actcon) ou WR-SUC. ▸ Les considérations suivantes s'appliquent à la saisie d'une valeur prescrite configurée : La valeur à définir pour l'action Saisie de `valeur prescrite` est stipulée par cet élément de contrôle. Lorsque vous cliquez sur l'élément de contrôle, il passe en mode édition et la saisie de la valeur prescrite devient possible. Pour quitter le mode édition, appuyez sur **Entrée**. Toutefois, la nouvelle valeur est uniquement définie lorsque l'utilisateur clique sur l'élément de contrôle **Exécuter**. L'élément de contrôle est verrouillé dans les cas suivants : ▸ L'état REVISION(9) de la variable de réponse est défini. ▸ Aucune action n'est active. ▸ Le synoptique ne se trouve pas à l'étape 1.
Unité de mesure	Unité de mesure de la variable d'action.

Remarque : Le nom est également affiché dans le `[commande] Nom variable d'action` **de driver système**.

Control elements - group: Response variable



Control element	Description
Nom	Name of the response variable
Identification	Name of the response variable
Status	Contains the short description of the status bits for the response variable.
Value	Current value of the response variable
Unité de mesure	Measuring unit of the response variable

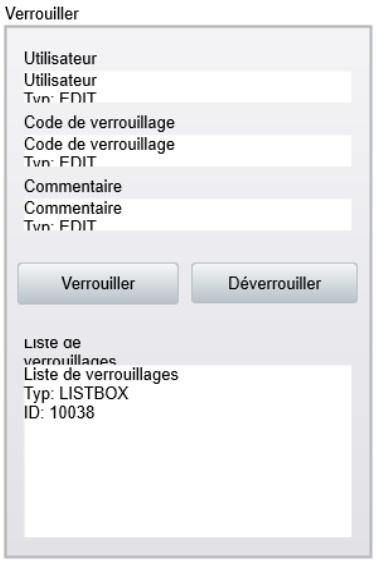
Note: The name is also in the `[command] name of the response variable` **system driver** variable.

Control elements - group: Select / execute

Control element	Description
On	First-step command button, to close a switch for example. If **SBO** is activated for the command variable - sends a Select to the controller. Note: Only visible in Step 1.
Off	First-step command button, to open a switch for example. If **SBO** is activated for the command variable - sends a Select to the controller. Note: Only visible in Step 1.
Confirm	Second-stage command button - Execute/Operate. Confirms the execution of the command after successful checking of the interlocking and positive confirmation of the Select from the PLC. Note: Only visible in Step 2.
Cancel	Second-stage command button Cancel. Aborts the execution of the command processing and returns to 'Step 1'. Note: Only visible in Step 2.
Close	Closes the command window in zenon Runtime

Note: The steps are also displayed in the `[command] screen step` **system driver** variable.

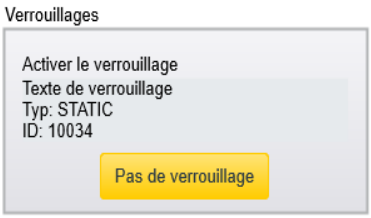
Control elements - group: Lock



Control elements from the **Lock** group are locked if no `Lock` action is configured in the command group.

Control element	Description
User	For entering the user identification for the lock.
Lock code	For entering the user-specific lock code.
Comment	Optional text that can be entered by the user for the lock.
Lock	Activates a lock by the user entered in the **User** control element. **Note:** This user action is logged in the CEL, if not suppressed by the engineering.
Unlock	Removes the lock that has been set up by the user entered in the **User** control element. This guarantees that only people's own locks can be removed. **Note:** This user action is logged in the CEL, if not suppressed by the engineering.
Lock list	List of active locks: ▶ `User` Name of the user who has activated the lock. ▶ `Locking time` Time stamp of the interlocking ▶ `Note` Text for the interlocking.

Control elements - group: Interlockings



Control element	Description
Interlocking text	The active interlocking (à la page 129) according to the configuration or texts from ALC - topological interlocking (à la page 35).
Unlock	This button unlocks an active, unlockable interlocking. Note: This control element is shown only when the screen is in the step 'Unlock' (Chapter: **Process in the Command Processing screen**). The Control element is locked when the upcoming interlocking is not unlockable.

Note: The interlocking texts are also displayed in the `[command] interlocking message` **system driver** variable.

4.4.3 Reload project online

The following is to be taken into account when reloading a command configuration:

- If Watchdog timer, edge generation or SBO is active, the reloading is delayed until this has ended.
- An opened `Command Processing` screen is closed and the process is started again after reloading depending on the current step:

Step before reloading	Action after reloading
Step 1	Screen is called up again for Step 1.
Interlocking or Step 2	Unlocking step is activated. The interlocking is then executed again.
Lock	Lock is activated again.

- Before it is called up again, the response variable, command variable, condition variables, command group and action are determined again. The control elements are locked if one of the objects is no longer present in the Runtime.
- If the command group is removed or replaced for the variable that was for calling up command input, the screen is called up with locked control elements. The screen must be called up again or the command processing must be executed again.
- If the command group was removed or replaced for the response variable, all locks triggered by the command processing are removed by the variables.
- If a user who has activated a command lock no longer exists, the lock is removed. The `M1` status bit and the `LOCK.BIN` file are updated accordingly.

4.4.4 Logging in the CEL

In the CEL, the following user actions are logged in addition to the switching actions.

Parameters	Description
Unlock	The unlocking of an active interlocking is noted in the CEL.
Unlock all	A corresponding CEL entry is created for each unlocked interlocking.
Execute action	If the "Suppress CEL entry" action setting is not active, the execution of the action is logged in the CEL.

4.4.5 Server change in redundant operation

If the standby server takes on the role of the Primary Server, the drivers take on the writing to the controller for the current Primary Server. An ongoing **Select Before Operate** is canceled as a result.

The handling of the **Select in the network** - `NET_SEL` status bit - of the response variable cannot be taken over from the previous Primary Server and the command must be executed again.

Please note the behavior or Runtime (à la page 173) if redundancy switching is not triggered by a failure of the Primary Server but by the user.

4.4.6 Exit Runtime

As long as there are still active actions in the system, the proper exiting of the runtime (e.g. over a function call) is delayed.

An active **Select Before Operate** process also delays the ending of zenon Runtime. If **Select Before Operate** is activated, a deactivation (Cancel) is carried out.

Informations

This situation occurs most of all with single-stage execution of the `pulse command` action with runtime monitoring or edge generation. Runtime is only ended once the action has been completed.

4.4.7 Lock return variable

A response variable is locked if the `M1` user status bit has been set.

In the `Lock` action, the lock can be activated or deactivated by entering the user name and a **Code de verrouillage** (configured during user definition). A user can activate a lock for each response variable. Several users can lock the same response variable.

The active locks are saved remanently in the `LOCK.BIN` file and are also taken into account after the system is restarted.

Informations

There is automatic synchronization of locking in the zenon network. Locking can thus also be used in redundant operation.