COPADATA
do it your way

Manuel de zenon

Process Gateway

v. 8.00

©2018 Ing. Punzenberger COPA-DATA GmbH

Tous droits réservés.

Toute distribution et/ou reproduction de ce document ou partie de ce document, sous quelque forme que ce soit, est uniquement autorisée avec la permission écrite de la société COPA-DATA. Les données techniques incluses sont uniquement fournies à titre d'information et ne comportent aucun caractère légal.

Contenu

1. Bienvenue dans l'aide de COPA-DATA 7

2. Process Gateway 7

3. Général 8

3.1 Attribution de variables avec des noms 9

4. Conditions requises 11

5. Début 11

5.1 Démarrez via Startup Tool. 13

5.2 Démarrer via la fonction zenon 14

5.3 Démarrage - Ligne de commande 16

5.4 Sélection du module 16

5.4.1 Premier démarrage 17

5.4.2 Modifier la sélection des modules existants 17

5.4.3 Démarrage avec un module configuré 18

5.5 Fichier INI 19

6. DEC 21

6.1 Configuration du module 21

6.2 Journalisation 22

7. DNP3_SG 22

7.1 Installation 23

7.2 Outstation 24

7.3 Datalink 27

7.3.1 Application Layer 31

7.3.2 Events/Unsolicited 34

7.3.3 File Transfer 38

7.4 Binary Input 40

7.5 Double Bit Binary Input 48

7.6 Binary Output 57

7.7 Running Counter 68

7.8 Frozen Counter 76
7.9 Analog Input 84
7.10 Analog Output 93
7.11 Octet String 102
7.12 Variable selection dialog 108
7.13 Warning dialog 109
7.14 Status mapping zenon - DNP3 112
7.15 Select before Operate 114
7.16 Command processing - Command Routing 114
7.17 LATCH_ON and LATCH_OFF 117

8. DNP3 Slave - AccessDNP3 obsolète 119

9. ICCP-TASE.2 120

9.1 Module configuration 122
9.1.1 General 124
9.1.2 Server variables 131
9.1.3 Client variables 135
9.2 Status variables of the connection 140
9.3 Show non-present variables in Runtime 141
9.4 Direct control (non-SBO) for ICCP control objects 142
9.5 Configuration file: INI entries for ICCP-Tase.2 144
9.6 LOG entries 147
9.7 Interoperability 151

10. IEC870 Slave 151

10.1 Module configuration 153
10.1.1 General Settings 156
10.1.2 Device List 159
10.1.3 Device - Configuration 161
10.1.4 Sector - Configuration 166
10.1.5 104 settings 188
10.1.6 DPI / DCS mapping 190
10.1.7 Configuration via XML file 191
10.2 Sequence of Events (SOE) 201
10.2.1 Warnings 203

10.2.2 Error message ... 204
10.3 Communication ... 205
10.3.1 Start ... 205
10.3.2 General interrogation ... 205
10.3.3 End of connection and resets ... 207
10.3.4 Transfer of the value changes ... 209
10.3.5 Reloading Runtime ... 209
10.3.6 Data priorities ... 210
10.3.7 Internal type T00 (status variable) ... 210
10.3.8 Mapping of status bits in message direction ... 215
10.3.9 Commands ... 217
10.3.10 Read - T102 - request ... 219
10.3.11 Time synchronization - T103 - request ... 220
10.3.12 Test - T104 and T107 - requests ... 220
10.3.13 File transfer ... 221
10.4 Broadcasts ... 224
10.5 Error analysis ... 224
10.5.1 Analysis tool ... 225
10.5.2 Error codes ... 231
10.5.3 Check list ... 232
10.6 Interoperability ... 232

11. Protocole MODBUS Slave ... 250

11.1 Configuration du module ... 251
11.2 Holding Registers ... 253
11.3 Coils ... 255
11.4 Paramètres d'adresse ... 257
11.5 Adresse matérielle du système Modbus ... 258
11.6 Fichier de configuration : entrées spécifiques pour AccessMODBUS ... 258

12. Microsoft Azure ... 260

12.1 Configuration du module ... 261
12.1.1 Paramètres de la connexion à Microsoft Azure ... 266
12.2 Format des messages ... 270
12.3 Message d'erreur ... 272
12.4 Fichier de configuration : entrées spécifiques pour Microsoft Azure ... 274

13. OPC UA Server 275

13.1 Comparaison du serveur OPC et du serveur OPC UA 276

13.2 Module configuration 278

13.2.1 Server 279

13.2.2 Variables 285

13.2.3 Archives 289

13.3 Start 291

13.4 Connection 292

13.5 Security 292

13.6 Alarms and Conditions 292

13.7 Service Sets 295

13.8 Communication 296

13.9 Configuration file: INI entries for OPC UA 299

13.10 Error message 302

13.11 Status mapping 305

14. Agent SNMP 306

14.1 Architecture SNMP 306

14.2 Management Information Base (MIB) 307

14.3 Structure MIB 307

14.4 Configuration du module 309

14.5 fichier de configuration : entrées spécifiques pour AccessSNMP 310

14.6 Configuration de l'agent SNMP Windows 310

15. SQL 313

15.1 Configuration du module 313

15.2 Sélection des variables 313

15.3 Création des tables 315

15.4 Cas d'un système redondant 315

15.5 Fichier de configuration : entrées spécifiques pour AccessSQL 316

15.6 Format des tables 317

15.6.1 Table pour l'image du procédé 317

15.6.2 Table des ID de variables 317

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.

AIDE GÉNÉRALE

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

ASSISTANCE PROJET

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

LICENCES ET MODULES

Si vous vous rendez compte que vous avez besoin de licences ou de modules supplémentaires, notre personnel commercial sera ravi de vous aider : sales@copadata.com.

2. Process Gateway

*Le module **Process Gateway** constitue une interface avec des systèmes de plus haut niveau, par exemple Control Center Cela signifie que des parties de l'image de processus zenon sont fournies pour d'autres applications et mises à jour par celles-ci.*

- Les données process de zenon Runtime peuvent être transmises à des systèmes supérieurs.
- Les systèmes supérieurs peuvent écrire des valeurs ou des commandes sur zenon Runtime.

> **Informations**
>
> *Le module **Process Gateway** est disponible en anglais uniquement.*
>
> *Tous les composants utilisés doivent faire l'objet d'une licence.*
>
> *Remarque : L'éditeur et/ou le runtime, qui fournissent les informations de licence, ainsi que les applications qui y accèdent, doivent fonctionner dans le même contexte utilisateur. They must be started by the same user. Démarrer une application en tant qu'administrateur et une autre en tant qu'utilisateur local ne fonctionne pas.*

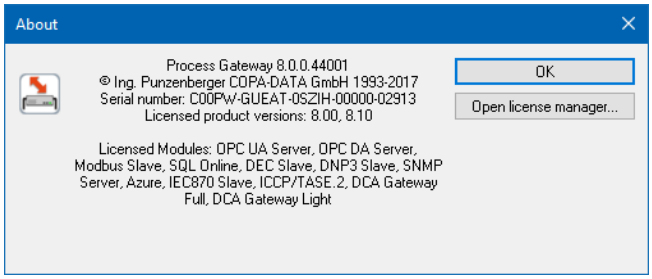
MONTRER LES INFORMATIONS DE LICENCE

Pour afficher les informations de licence dans le **Process Gateway** :

1. Lancez le **Process Gateway**.
2. Cliquez sur le bouton **About**.

   Le système affiche la boîte de dialogue **About**.

BOÎTE DE DIALOGUE À PROPOS



Option	Description
Informations de licence	Informations sur les modules sous licence avec leur numéro de série.
OK	Ferme la boîte de dialogue.
Open license manager	Lance le **Gestion des licences**. Cet outil sert à activer, retourner et gérer les licences. Vous avez besoin ici du numéro de série.

Si le **Process Gateway** est démarré sans configuration de module, l'utilisateur doit d'abord sélectionner un modèle. Ceci fait, le système affiche la boîte de dialogue de démarrage correspondante pour le **Process Gateway**. Le bouton **Settings** est inactif jusqu'à ce que le **Process Gateway** soit associé à une licence.

3. Général

Les considérations suivantes s'appliquent à la **Process Gateway** :

- Il a été conçu sous la forme d' outil additionnel du Runtime du système de contrôle, et ne fonctionne qu'avec lui.
  Ceci signifie : Si le Runtime de zenon n'a pas été démarré, **Process Gateway** ne démarre pas !
- Several **Process Gateways** can be started on one computer.
  Vous trouverez de plus amples informations dans le chapitre Démarrage - Ligne de commande (à la page 16).
- The **Process Gateway** can be started on both a standalone computer as well as in the zenon network (server/standby/client). All **Process Gateways** that are started in the network provide the same process image.
- Le module devant être démarré est sélectionné et configuré dans son propre fichier INI (à la page 11) de **Process Gateway**, et non dans le fichier zenon6.ini, comme c'est habituellement le cas dans zenon !
  The INI file for the **Process Gateway** is called `zenProcGateway.INI` and is in the `%CD_SYSTEM%` folder.With a standard installation, the storage folder is `C:\ProgramData\COPA-DATA\System` by default.

SÉLECTION DU PROJET

If no special project name is defined in the configuration file (`zenProcGateway.ini`), all configurations of the **Process Gateway** are applicable to the first project loaded by Runtime.

In a multi-project system, it is possible to access any desired project in the hierarchy. To do this, enter the project name in the **Project**= entry in the **[GENERAL]** section in the INI file. If this entry is empty, the integration project is applicable.

3.1 Attribution de variables avec des noms

Dans le module **Process Gateway**, l'attribution correspondante est effectuée à l'aide du nom de la variable.

Cette considération est importante si :

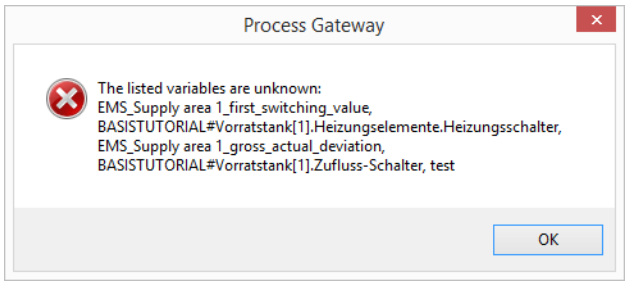
- Les noms de variables dans la configuration du projet sont modifiés dans zenon Editor.
- Les variables dans la configuration du Project dans Editor sont supprimées.
- zenon appelle un nouveau projet de démarrage.

Dans ce cas, des boîtes de dialogue d'erreur indépendantes des modules apparaissent au démarrage de **Process Gateway**.

Attention

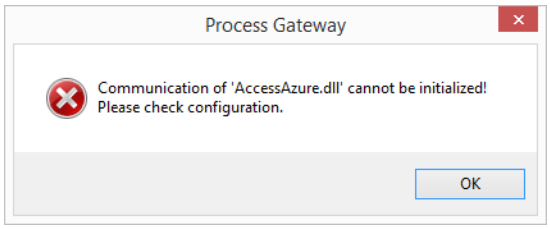
*Si une attribution de variable est erronée, le module **Process Gateway** ne démarre pas ! Le module **Process Gateway** démarre uniquement si l'attribution de variable erronée a été corrigée.*

La première boîte de dialogue d'erreur dresse la liste des variables inconnues.



- Dans ce cas, vous devez remodifier l'attribution de variable.
- Pour cela, supprimez toutes les variables incorrectement attribuées de la zone d'exportation concernée, puis rajoutez les variables correspondantes.

La boîte de dialogue suivant contient des messages d'erreur dépendants du module.



ENTRÉE DE FICHIER JOURNAL

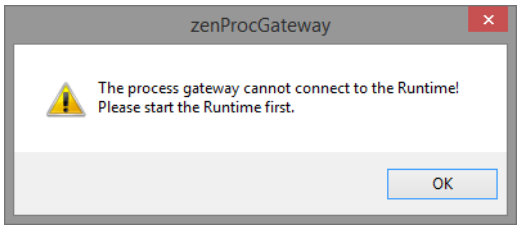
L'entrée de fichier journal suivante est créée en cas d'attribution incorrecte d'une variable :

Entrée de fichier journal	Description
`Failed to update value for variable %s.`	Aucune connexion COM n'a pu être établie à zenon ou la variable n'est pas (ou n'est plus) disponible dans zenon, ou elle a été renommée.

4. Conditions requises

Sur le PC sur lequel doit s'exécuter **Process Gateway**, un Runtime de la version correspondante doit s'exécuter et un projet correspondant à la configuration doit être chargé avant le démarrage de Process Gateway. En cas d'arrêt du Runtime, **Process Gateway** s'arrête également automatiquement.

Si zenon Runtime n'est pas chargé lorsque **Process Gateway** est lancé, une boîte de dialogue d'avertissement correspondante s'affiche :



Informations

*Le module **Process Gateway** utilise l'interface COM pour accéder aux données de zenon. L'interface COM est utilisée par VBA et est uniquement disponible si **zenon6.ini** est présent dans le fichier sous **[VBA]**, à l'entrée* `EVENT=1`.

5. Début

The **Process Gateway** consists of the **zenProcGateway.exe** file**.** This is copied to the zenon program folder during installation. However, the file can also be used if it is saved in another folder.
Attention : Le module (Access*.dll) à utiliser avec la passerelle **Process Gateway** doit se trouver dans le même dossier que le fichier **zenProcGateway.exe**.
Exception : **SNMP** (à la page 310).

The **zenProcGateway.ini** configuration file is in the `%CD_SYSTEM%` system folder.
In this INI file, the `DLL=` entry (in the **[GENERAL]** section) determines the module with which the **Process Gateway** starts.

Plusieurs **Process Gateways** peuvent être démarrés sur un seul ordinateur. La condition préalable est l'utilisation de différents fichiers INI. Vous trouverez de plus amples informations dans le chapitre Démarrage - Ligne de commande (à la page 16).

MODULE PROCESS GATEWAY SUR LE RÉSEAU ZENON

Process Gateway peut être démarré aussi bien sur un ordinateur autonome que sur le réseau zenon (serveur/serveur de secours/client). All **Process Gateways** that are started in the network provide the same process image.

Dans les réseaux redondants, il est recommandé de démarrer **Process Gateway** sur un client (dédié). Avec la commutation de redondance, un client réseau se connecte automatiquement au serveur principal respectif et continue à fournir les données aux systèmes de niveau supérieur.

PROCESS GATEWAY - DÉMARRAGE DU PROGRAMME

Process Gateway peut être démarré :

- Via l'outil Startup Tool (à la page 13):
- Via fonctions de zenon (à la page 14)

Dans les deux méthodes de démarrage, le **Process Gateway** peut être lancé avec des paramètres dans l' interface de ligne de commande (à la page 16). Différents fichiers INI du **Process Gateway** peuvent ainsi être déterminés.

> **Attention**
>
> **Process Gateway** ne peut être lancé que si zenon Runtime est exécuté. Démarrer sans Runtime n'est pas possible. Une boîte de dialogue d'avertissement correspondante s'affiche dans ce cas.

CONFIGURATION

Le transport à distance est constitué de deux parties :

- Sélection du module (à la page 16)
  Dans la première étape de la configuration, sélectionnez le module qui communique avec le **Process Gateway**.
  Vous trouverez de plus amples informations à ce sujet dans le chapitre Sélection du module.
- Module-specific configuration
  Depending on the selected module, connection parameters and variable assignments can be configured.
  You can find details on this from the module configuration chapter in the respective section in this handbook.

All configurations of the **Process Gateway** are saved in an INI file. Please note that not all entries of the corresponding INI file can be configured via a corresponding user interface.

The module is determined with the **DLL**= entry (in the **[GENERAL]** section).

Saisies possibles :

Module	Entrée
DEC	**DLL**=AccessDEC.dll
SQL	**DLL**=AccessSQL.dll
MODBUS	**DLL**=AccessMODBUS.dll
DNP3	**DLL**=AccessDNP3.dll
IEC870SI	**DLL**=AccessIEC870Sl.dll
SNMP	**DLL**=AccessSNMP.dll
OPCUA	**DLL**=AccessOPCUA.dll
ICCP/TASE.2	**DLL**=AccessICCP.dll
MS Azure	**DLL**=AccessAzure.dll
DNP3_SG	**DLL**=AccessDNP3_SG.dll

CONFIGURATION SÉCURITÉ

Le **Process Gateway** peut être exécuté de façon cachée de façon à le protéger des accès non autorisés. Si la passerelle **Process Gateway** est démarrée avec une ligne de commande avec le paramètre `/hide` (ou `-hide`), elle démarre en mode invisible et ne peut pas être configurée ou arrêtée. Le module **Process Gateway** se ferme automatiquement lorsque le zenon Runtime est fermé.

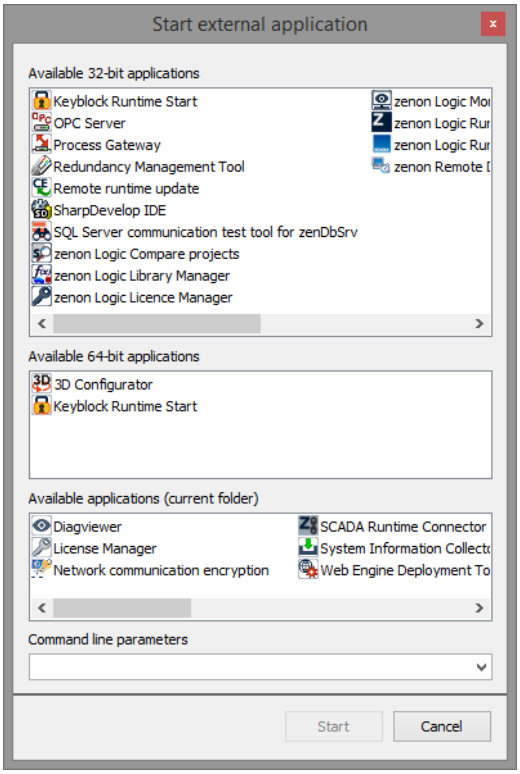
5.1 Démarrez via Startup Tool.

L'outil **Process Gateway** peut également être démarré depuis la zenon **Startup Tool**.

Procédure de configuration :

1. Ouvrez zenon **Startup Tool**.
2. Cliquez sur **Tools**.

La boîte de dialogue de configuration du zenon Tools s'affiche à l'écran.



3. Sélectionnez **Process Gateway**.
4. Click the Start button to start the **Process Gateway** with the configuration from the current `zenProcGateway.INI` file.

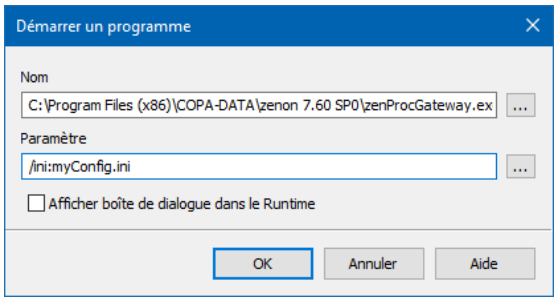
   Or:
   Enter the name of the EXE file and the name of the INI file as a start parameter into the **Command line parameters**:
   zenProcGateway.EXE /ini:"MyConfig.INI"

   Vous trouverez d'autres informations à ce sujet au chapitre Démarrage - Ligne de commande (à la page 16).
   Remarque : Les paramètres seuls ne suffisent pas. L'appel **zenProcGateway.EXE** doit être entré.
5. Cliquez sur **Start**.

5.2 Démarrer via la fonction zenon

Le **Process Gateway** peut - comme toute autre application Windows - être démarré avec la fonction **Exécuter un programme**.

DÉVELOPPEMENT



Pour créer une fonction :

1. Créez une nouvelle fonction :

   Dans la barre d'outils ou le menu contextuel du nœud Fonctions, sélectionnez **Nouvelle fonction**.
   La boîte de dialogue de sélection d'une fonction s'affiche à l'écran.

2. Accédez au nœud **Windows**.

3. Sélectionnez la fonction **Exécuter un programme** .
   La boîte de dialogue pour sélectionner le programme et entrer les paramètres de démarrage s'ouvre.

4. Sélectionnez le programme souhaités.

   a) Cliquez sur le bouton **....**
   La boîte de dialogue de sélection de fichiers s'ouvre.

   b) Go to the zenon Installation directory:
   `Program Files (x86)` - 64-bit operating system

   `Program Files` - Systèmes d'exploitation 32 bits

   c) Sélectionnez le fichier `zenProcGateway.exe`

   d) Confirmez la sélection en cliquant sur le bouton **Ouvrir**.

5. Optionnel : Configure the parameters hat are executed when the function is executed.
   You can find further information on this in the Start with cammand line (à la page 16) chapter.
   If this parameter is empty, the **Process Gateway** starts with the `zenProcGateway.INI` file.

6. Fermez la boîte de dialogue en cliquant sur le bouton **OK**.

7. Nommez le fonction dans la propriété **Nom**.

> **Conseil**
>
> *If this function is linked in the `AUTOSTART*` script, **Process Gateway** automatically starts with Runtime.*

5.3 Démarrage - Ligne de commande

Plusieurs **Process Gateways** peuvent être démarrés sur un seul ordinateur. To do this, a separate INI file must be present for each **Process Gateway** to be started. Le fichier INI doit se trouver dans le dossier système (%CD_SYSTEM%).

Exemple :

- Simultaneous start of several **Process Gateways** with different modules
- Start several instances of the **Process Gateway** with the same module (not applicable for all modules).
  Requirement: separate communication parameters for each instance, for example no conflicts in IP ports.

In this case, the **zenProcGateway.EXE** must be started with the command line parameter **/ini:"[file]"**.

Note: the name of the INI file can also be configured for starting with zenon function (à la page 14).

Ligne de commande - Syntaxe

- Dans l'outil Startup Tool:
  `zenProcGateway.EXE /ini:"[FileName].INI"`
- In the zenon **Exécuter un programme** function
  Parameter input field: `/ini:"[FileName].INI"`

`[FileName].ini` must be replaced by the correct name of the INI file.

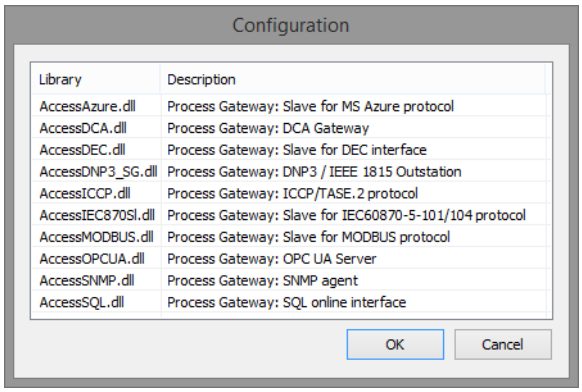
5.4 Sélection du module

The following possibilities are available for the selection of a module for the **Process Gateway**:

- First start
  When the **Process Gateway** is started for the first time, a dialog to select the module is shown. This selection dialog is also offered when **Process Gateway** is started if there is no INI file or if no module is configured in the selected INI file.
  The `zenProcGateway.INI` file is used by default.
- Module already configured
  If a module has already been configured for **Process Gateway**, the Process Gateway starts with the communication dialog of the configured module.
  When **Process Gateway** starts, the communication already starts if all necessary connection parameters for the module have been configured and are valid.

5.4.1 Premier démarrage

Le module doit être configuré lors du premier démarrage.



To start **Process Gateway** with a selectable module:

1. Start **Process Gateway** with an INI file that does not exist (there must be no other module configured in the INI file).
   The **Process Gateway** uses the file `zenProcGateway.INI` by default. You can define another file in the command line (`/ini:"[FileName].INI"`).
   You can find more information in the **Start with command line** (à la page 16).
2. La boîte de dialogue de sélection du module s'ouvre.
3. Sélectionnez les modules souhaités.
4. Confirmez votre sélection en cliquant sur le bouton **OK**.
   Le module **Process Gateway**généré un file INI.
5. Le module **Process Gateway** commence par la boîte de dialogue de configuration des paramètres de communication (à la page 18) du module sélectionné.

5.4.2 Modifier la sélection des modules existants

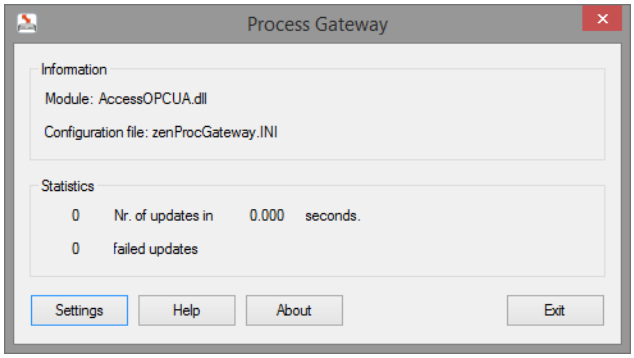
Démarre Process Gateway avec un autre module déjà configuré, sans afficher la boîte de dialogue de sélection de module, puis :

1. Fermez **Process Gateway**.
2. Modifiez le fichier zenProcGateway.INI existant.
   a) Pour ce faire, allez dans le répertoire %CD_SYSTEM%.
   b) Ouvrez le fichier dans n'importe quel éditeur de texte et ajoutez des commentaires pour le module dans le fichier ; ou renommez le fichier INI.

3. Redémarrez la **Process Gateway**:
   Si le fichier INI a été supprimé, la boîte de dialogue de sélection de module (à la page 17) s'affiche à nouveau au démarrage de **Process Gateway** .

5.4.3 Démarrage avec un module configuré

Lorsque vous démarrez **Process Gateway** avec un module préalablement configuré, une boîte de dialogue contenant des données statiques s'affiche.



INFO

Informations minimales sur le module configuré.

Paramètre	Description
Module:	Module configuré et démarré de **Process Gateway**.
Config File:	Nom du fichier de configuration `zenProcGateway.INI` de **Process Gateway**.

STATISTIQUE

Informations sur la communication avec la contrepartie :

Paramètre	Description
Nr. of updates in xx seconds	Nombre de tentatives de communication réussies, y compris la durée totale.
failed updates	Nombre d'échecs de communication.
Exit	Ferme l'outil **Process Gateway**.
Settings	Ouvre le dialogue de configuration du module.
Help	Ouvre l'aide en ligne.

Attention

*If the application is started as `invisible`, there is also no configuration dialog displayed. If the configuration is invalid or not present, the **Process Gateway** is automatically closed.*

5.5 Fichier INI

Process Gateway enregistre la sélection du module dans le fichier INI. Le fichier `zenProcGateway.INI` se trouve dans le chemin suivant : `%CD_SYSTEM%`. La configuration est sauvegardée pour chaque module dans un fichier XML. Le fichier de configuration est enregistré dans le dossier du système (%CD_SYSTEM%).

Les configurations spécifiques aux modules sont très différentes d'un module à l'autre. Veuillez tenir compte de la documentation détaillée dans les entrées dépendantes du module dans ce manuel.

En raison des restrictions du système, seuls les codages ANSI et Unicode sont pris en charge lors de la lecture de fichiers INI.

Attention

Le format UTF-8 n'est pas pris en charge !

Vous devez donc toujours enregistrer vos fichiers INI sous forme de fichier texte, avec un encodage ANSI ou Unicode.

CONTENU INDÉPENDANT DU MODULE

[GENERAL]

Entrée	Description
[GENERAL]	Paramètre général de Process Gateway, indépendamment des modules sélectionnés.
DLL=	Sélection du fichier DLL utilisé pour Process Gateway. Le fichier DLL sélectionné détermine la sélection du module. ▶ **AccessAzure.dll** ▶ **AccessDEC.dll** ▶ **AccessDNP3_SG.dll** ▶ **AccessICCP.dll** ▶ **AccessIEC870SI.dll** ▶ **AccessMODBUS.dll** ▶ **AccessOPCUA.dll** ▶ **AccessSNMP.dll** ▶ **AccessSQL.dll**

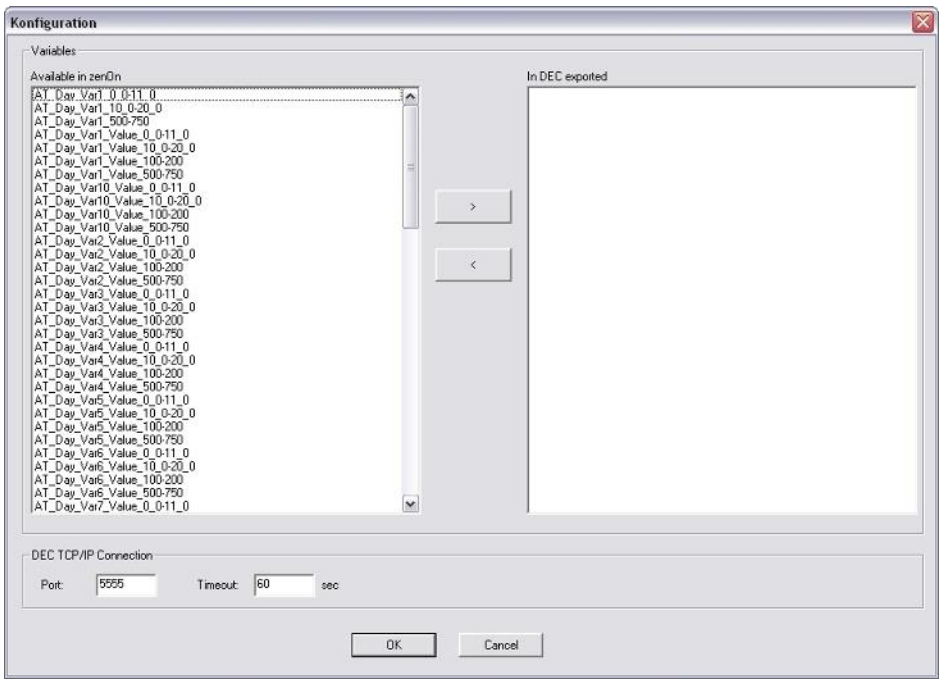
[DATABASE]

Entrée	Description
PROJECT=	Nom du projet zenon avec lequel le **Process Gateway** communique avec Runtime. In a multi-project system, it is possible to access any desired project in the hierarchy. If this entry is empty, the integration project is applicable. Par défaut : `Vide` (projet actuellement chargé dans Runtime)

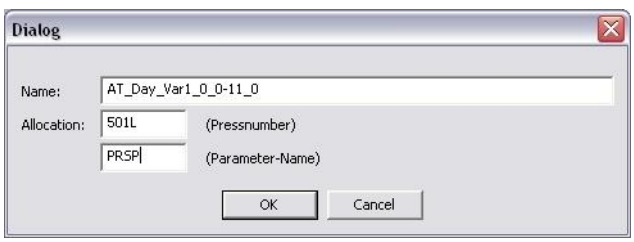
6. DEC

6.1 Configuration du module

La boîte de dialogue suivante apparaît lorsque vous cliquez sur le bouton **Settings** :



Vous pouvez sélectionner ici les variables qui doivent être connectées. Dans la liste affichée à gauche de l'écran, toutes les variables disponibles dans zenon sont affichées. Elles peuvent être sélectionnées et déplacées dans la liste d'exportation DEC en cliquant sur le bouton **>**. Avec le bouton '**<**' vous pouvez les enlever. Si une variable est déplacée vers la liste d'export, une boîte de dialogue s'ouvre pour définir la correspondance entre la variable de zenon et la variable DEC.



L'assignation est définie avec un numéro à quatre chiffres et un nom de paramètre de quatre lettres. Chaque caractère est automatiquement passé en majuscules. Les variables sélectionnées de cette manière sont ensuite continuellement synchronisées dans zenon ou dans le système DEC connecté.

Dans la partie basse de la boîte de dialogue, vous pouvez configurer la connexion TCP/IP au système DEC.

Le numéro de port indique le port où la passerelle de procédé attend les connexions entrantes.

Le timeout est le temps au-delà duquel, si le Process Gateway n'a pas de réponse, la connexion est fermée.

6.2 Journalisation

Tous les messages reçus par le système DEC et les réponses du Process Gateway peuvent être tracés.

The Diagnosis Viewer is used for logging; you can find further information in the Diagnosis Viewer chapter.

7. DNP3_SG

The **AccessDNP3_SG**module for the **Process Gateway** is based on a DNP3 stack developed by COPA-DATA.

- The **AccessDNP3_SG** module is compliant with subset level 1, 2 and 3 and supports file transfer.
- The configuration is not compatible with the existing **AccessDNP3 Process Gateway** of zenon versions before version 8.00.
- The previous **AccessDNP3** module in the **Process Gateway** can no longer be selected for new configurations.
- Existing configurations with the previous **AccessDNP3** module continue to be able to run.
- The **AccessDNP3** module will no longer be supplied in a future version. The old module will be completely replaced by the **AccessDNP3_SG** module.

An XML device profile is supplied with the new module. This document also contains a description of the abilities of the **AccessDNP3_SG** module.

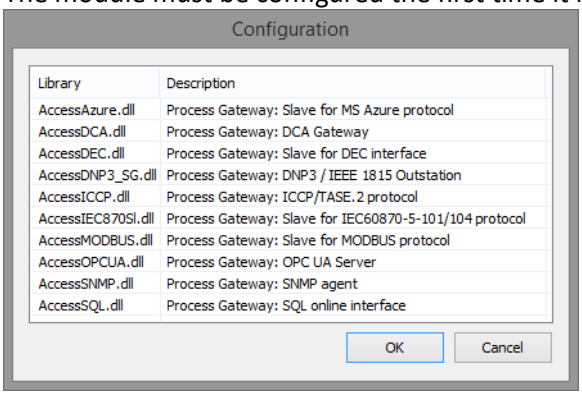
Installation path:

```
C:\ProgramData\COPA-DATA\zenon8.00\CommunicationProfiles\Dnp3\ProcessGateway
```

7.1 Installation

You have the following possibilities for installing the DNP3_SG Process Gateway:

1. The first time Process Gateway is started. No module has yet been configured:
   The module must be configured the first time it is started.



   a) Start **Process Gateway** with an empty file or an INI file that does not exist (there must be no other module configured in the INI file).
      The **Process Gateway** uses the file `zenProcGateway.INI` by default. You can define another file in the command line (`/ini:"[FileName].INI"`).
      You can find more information in the **Start with command line** (à la page 16).

   b) The **Configuration** dialog to select the module is opened.

   c) Select **AccessDNP3_SG.dll** as a module.

   d) Confirm your selection by clicking on the **OK** button.
      The **Process Gateway** then creates an INI file.

   e) **Process Gateway** starts with the Process Gateway communication dialog.

   f) Click on the **Settings** button to configure the **DNP3_SG** module.

2. Start with a **Process Gateway** module that has already been configured.

   If **Process Gateway** starts with a module that has already been configured for a different communication protocol, without offering the dialog to select a module, then carry out the following steps:

   a) Close the **Process Gateway**.

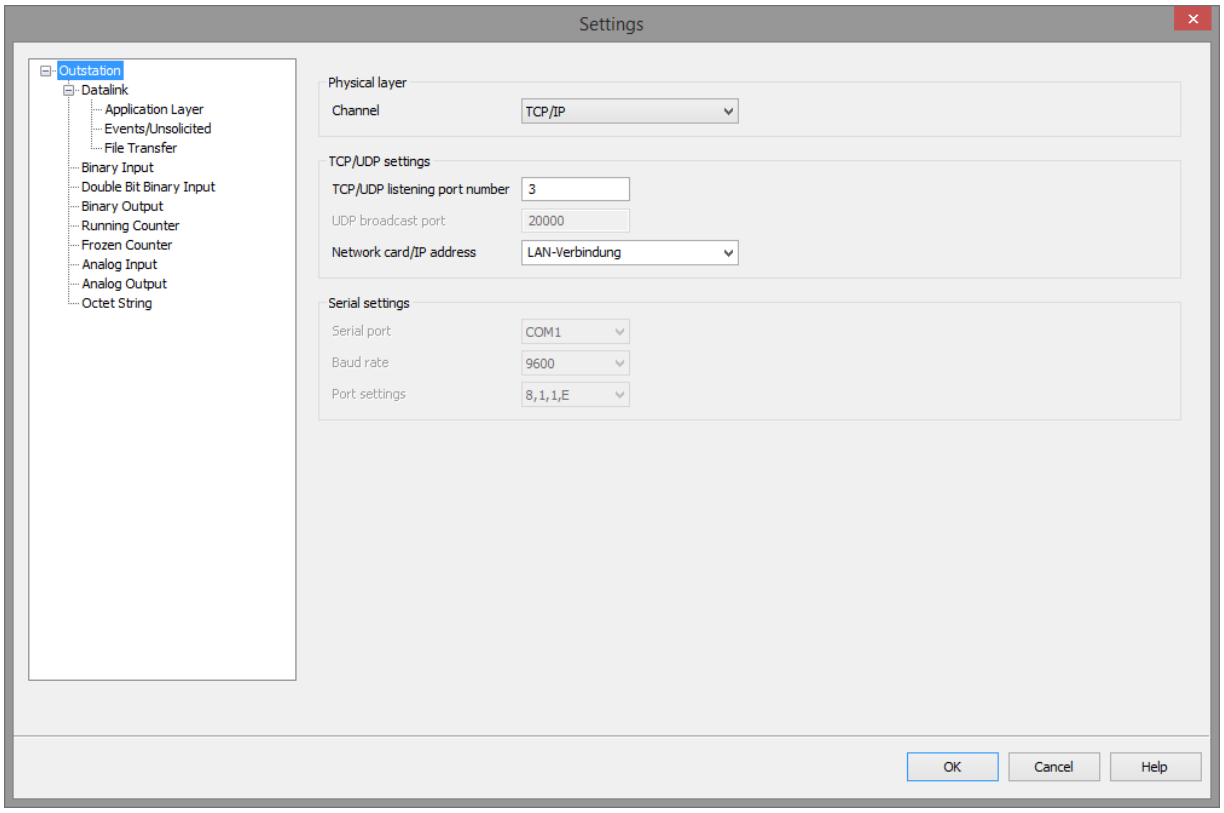
   b) Edit the existing zenProcGateway.INI file.

   c) To do this, go to the %CD_SYSTEM% directory.

   d) Open the file in any desired text editor and add comments for the module in the file; or rename the INI file.

   e) Restart the **Process Gateway**:
      If the INI file has been removed, the dialog to select a module (à la page 17) is offered again when **Process Gateway** starts.

7.2 Outstation



In this dialog, you configure the connection parameters to the outstation.

PHYSICAL LAYER

Parameter	Description
Channel	Type of connection to the outstation. Select from drop-down list: ▶ `Serial` Serial connection to the outstation. ▶ `TCP/IP` Connection to the outstation via TCP/IP protocol. ▶ `UDP` Connection to the outstation via UDP protocol. ▶ `TCP with UDP broadcast` Connection to the outstation via TCP/IP protocol with UDP broadcast. Default: `TCP/IP`

TCP/UDP SETTINGS

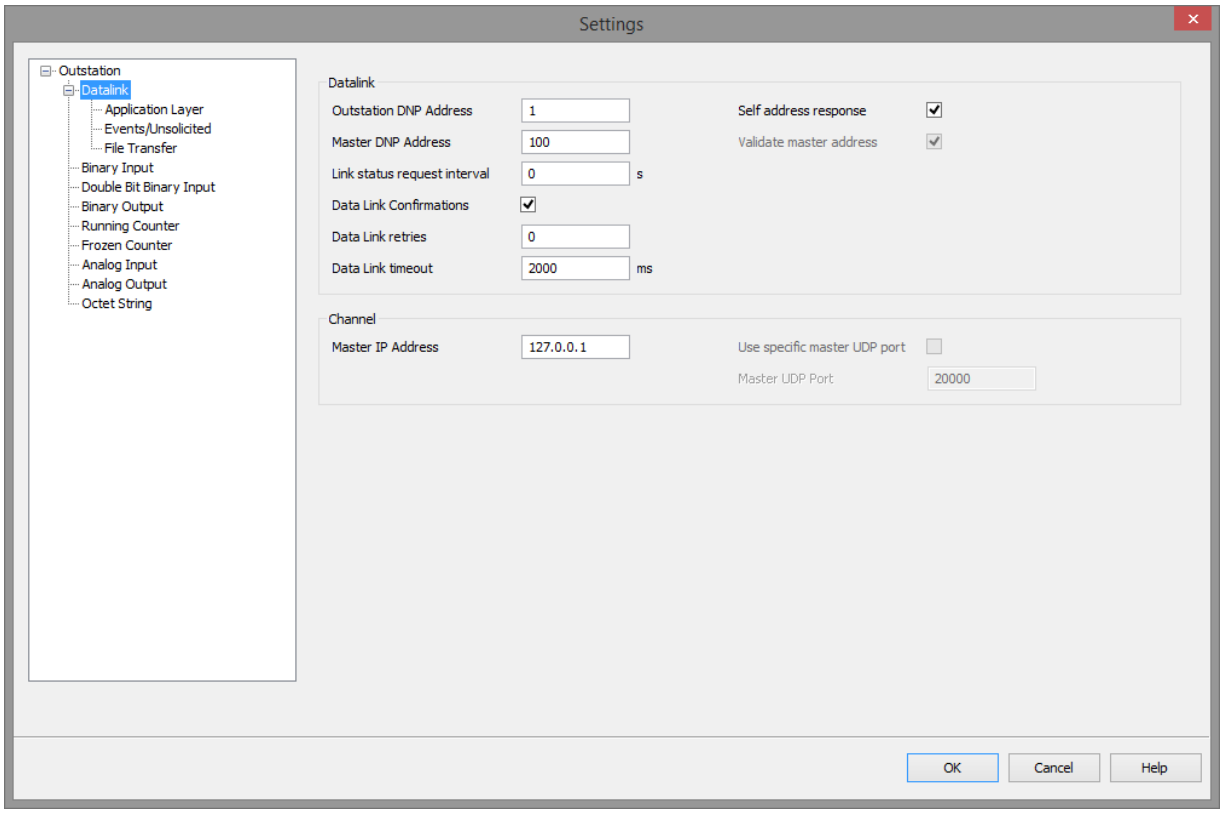
Parameter	Description
TCP/UDP listening port number	Port number for the TCP or UDP communication with the outstation. Default: `20000`
UDP Broadcast port	Port number for the receipt of UDP broadcasts. Default: `20000` Note: only active if, in the **Channel**option, the `TCP with UDP broadcast` value has been configured.
Network card/IP address	Network interface for communication with the outstation. Select from drop-down list. The outstation receives network telegrams at the configured interface. If no interface is selected, the outstation receives the telegrams at all interfaces. Default: `empty` Note: Not active if `Serial` is configured for the **Channel** option.

SERIAL SETTINGS

The options for this group are only active if the `Serial` connection type has been configured for the **Channel** option.

Parameter	Description
Serial port	Selection of the serial interface. Selection from a drop-down list. Default: `COM1`
Baud rate	Selection of the Baud rate for serial communication. Select from drop-down list. Default: `9600`
Port settings	Number of data bits, start bits and stop bits, as well as the parity for serial communication. Select from drop-down list: ▶ `8,1,1,N` ▶ `8,1,1,E` ▶ `8,1,2,E` ▶ `8,1,2,N` Default: `8,1,1,E`

7.3 Datalink


Settings
Outstation
Datalink
Application Layer
Events/Unsolicited
File Transfer
Binary Input
Double Bit Binary Input
Binary Output
Running Counter
Frozen Counter
Analog Input
Analog Output
Octet String
Datalink
Outstation DNP Address
1
Master DNP Address
100
Link status request interval
0
s
Data Link Confirmations
Data Link retries
0
Data Link timeout
2000
ms
Self address response
Validate master address
Channel
Master IP Address
127.0.0.1
Use specific master UDP port
Master UDP Port
20000
OK
Cancel
Help

Parameter	Description
Outstation DNP Address	DNP3 address of the outstation. Default: `1`
Self address response	Action of the outstations in response to a `Request`. ▶ `Active:` The outstation responds with its own DNP3 address if a `Master Request` is received with the "0xFFFC" reserved address. ▶ `Inactive:` The outstation only responds to `Requests` from the configured master address. Default: `Inactive` Note: It is recommended that this option is only activated temporarily to allow a master to detect the outstation.
Master DNP Address	The DNP3 address of the master that is to communicate to the outstation. Default: `100` Note: the **Process Gateway** currently only supports one master per instance.
Validate master address	Validation of the master address Note: This option is always active. The outstation only responds to `Master requests` that come from the configured **Master DNP Address**. An exception is the broadcast addresses, if `TCP with UDP Broadcast` has been configured in the **Channel** option in the **Outstation** node and the `Broadcasts allowed` option has been activated in the **Application Layer** node. Default: `active`
Link status request interval	Intervals in seconds in which a Datalink Layer `REQUEST_LINK_STATUS` is sent from the outstation to the master, if there is no further data traffic. This is, for configurations whereby no cyclical communication takes place with the master, to monitor whether the master is still available. Default: `0` With the value 0, the outstation does not send any link status requests. Note: There is a **TCP Keep Alive** for TCP connections. This

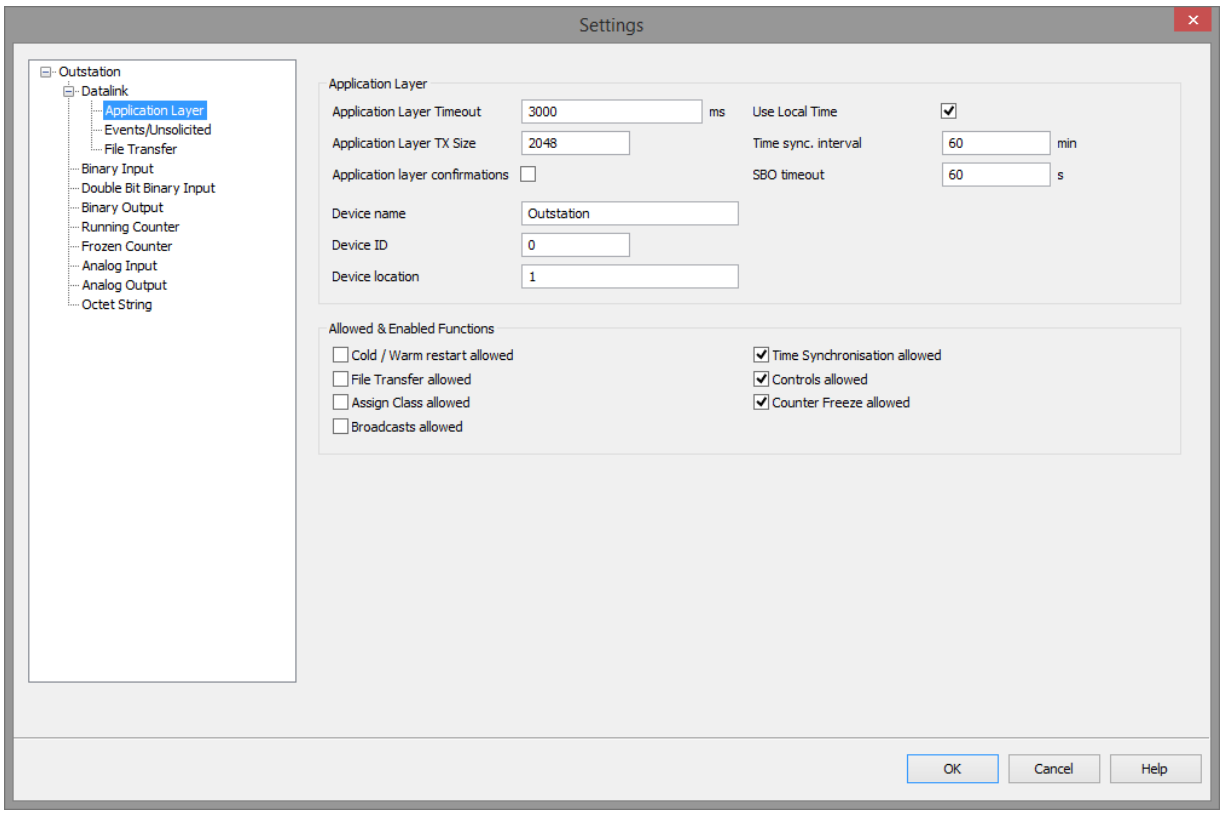
	must not be confused with the `Link status Requests`.
Data Link Confirmations	▶ `Active:` The outstation requests a `Datalink Layer` confirmation from the master for each Datalink Frame. ▶ `Inactive:` The outstation does not request confirmation for `Datalink Layer Frames`. Default: `Inactive` Note: This option should only be activated if it requires a low-quality connection. For TCP/IP connections, it is strongly recommended that this options is not activated.
Data Link retries	Number of recurrences for a `Datalink Layer Frame` if a confirmation has been requested from the master and this confirmation has not been received within the time configured in the **Datalink Layer Timeout** option. Default: `0`
Data Link Timeout	Timeout for the receipt of a `Datalink Layer` confirmation from the master to a `Datalink Layer Request` requested by an outstation for which a `Datalink Layer Confirmation` has been requested. Default: `2000`

CHANNEL

Parameter	Description
Master IP Address	IP address of the remote DNP3 master. Only network telegrams from the IP address configured here are accepted. Exception: `Requests` At the UDP broadcast address if the **Channel** option has been configured with `TCP/IP with UDP Broadcasts` in the **Outstation** node and the **Broadcasts allowed** option is active in the **Application Layer** node. Default: `127.0.0.1` Note: Not active if `Serial` is configured for the **Channel** option.
Use specific master UDP port	▶ `Active:` The outstation responds to the UDP port

	configured here. ▶ `Inactive:` The outstation responds to the UDP port from which the `Master Request` has been sent. Only available if the **Channel** has been configured with `UDP` in the **Outstation** node.
Master UDP Port	This port is used for the optional `unsolicited null response` message. Alternative UDP port number for the response to the master if the **Use specific master UDP port** option has been activated. Default: `20000` Note: Only available if the **Channel** has been configured with `UDP` in the **Outstation** node.

7.3.1 Application Layer



Parameter	Description
Application Layer Timeout	Time that the outstation waits for a response from the master if `Responses` demand a confirmation. Default: `3000 ms`
Use Local Time	Time format used. This option is applicable both for time synchronization of the outstation as well as for the `Event Objects` that contain a time stamp. ▶ `Active:` The time format uses the local time / time zone of the computer. This option can be set if the master also considers the time as local time. ▶ `Inactive:` The time format is UTC, corresponding to the IEEE Std 1815tm standard. Default: `active`
Application Layer TX Size	Maximum size of an `Application Layer response`. The value can be amended according to the ability of the master and the bandwidth of the connection Default: `2048`
Time sync. interval	Time period after which the outstation sets the Internal Indication Bit IIN1.4 (NEED_TIME) in the `Response` again. The value can be set to 0 if no time synchronization is required by the master. Example: The outstation master already synchronizes its time from another source. Default: `60 min`
Application layer confirmations	**Application Confirmation** for `Response` ▶ `Active:` The outstation requests an **Application Confirmation** from the master for each `Response`. ▶ `Inactive:` The outstation only requests an **Application Confirmation** from the master for the following `Responses`.

	- Response contains events - unsolicited Responses - multi-fragment responses (From the first to the second-to-last fragment. The last fragment does not demand a Confirmation) Default: `Inactive` Note: Only activate this option in exceptional cases, for example if the connection quality and bandwidth requires this (low bandwidth).
SBO timeout	Time that the master waits for an **Operate Request** after a **Select Request** has been received by the master. Default: `60 s` You can find further information about the **SBO** in the **Select before Operate** (à la page 114) chapter.
Device name	Description for the outstation. This option can be read and written by the master using the `User-assigned device name (g0v247)` **Device Attribute** Default: `Outstation`
Device ID	Identification for the outstation. This option can be read and written by the master using the `User-assigned ID code/number (g0v246)` **Device Attribute** Default: `0`
Device location	Site description for the outstation. This option can be read and written by the master using the `User-assigned location name (g0v245)` **Device Attribute** Default: `1`

ALLOWED & ENABLED FUNCTIONS

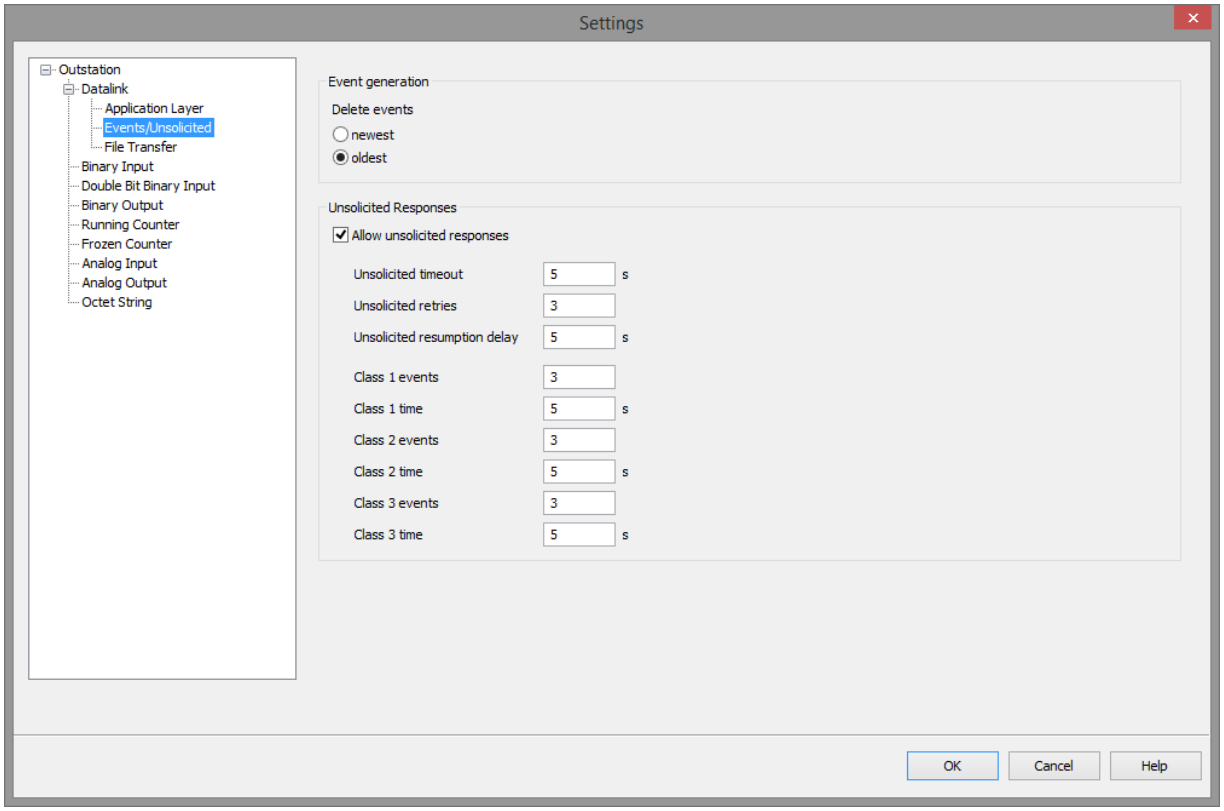
Functions that are offered by the outstation for the connection between the master and outstation.

If an option is not active, the outstation responds to the master with a **Null Response** with the `Internal Indication Bit IIN2.0` (NO_FUNC_CODE_SUPPORT).

Exception: there is generally no response to `Broadcasts requests`.

- **Cold/Warm restart allowed**
  Default: `Inactive`
- **File Transfer allowed**
  Default: `Inactive`
- **Assign Class allowed**
  Default: `Inactive`
- **Broadcasts allowed**
  Default: `Inactive`
- **Time Syncrhonisation allowed**
  Default: `active`
- **Controls allowed**
  Default: `active`
- **Counter Freeze allowed**
  Default: `active`

7.3.2 Events/Unsolicited



EVENT GENERATION

Parameter	Description
Delete events	The events are deleted in the event of an overflow of the Outstation event buffer: ▶ `newest:` Most recent (=youngest) events. ▶ `oldest:` Oldest events Default: `newest`

UNSOLICITED RESPONSES

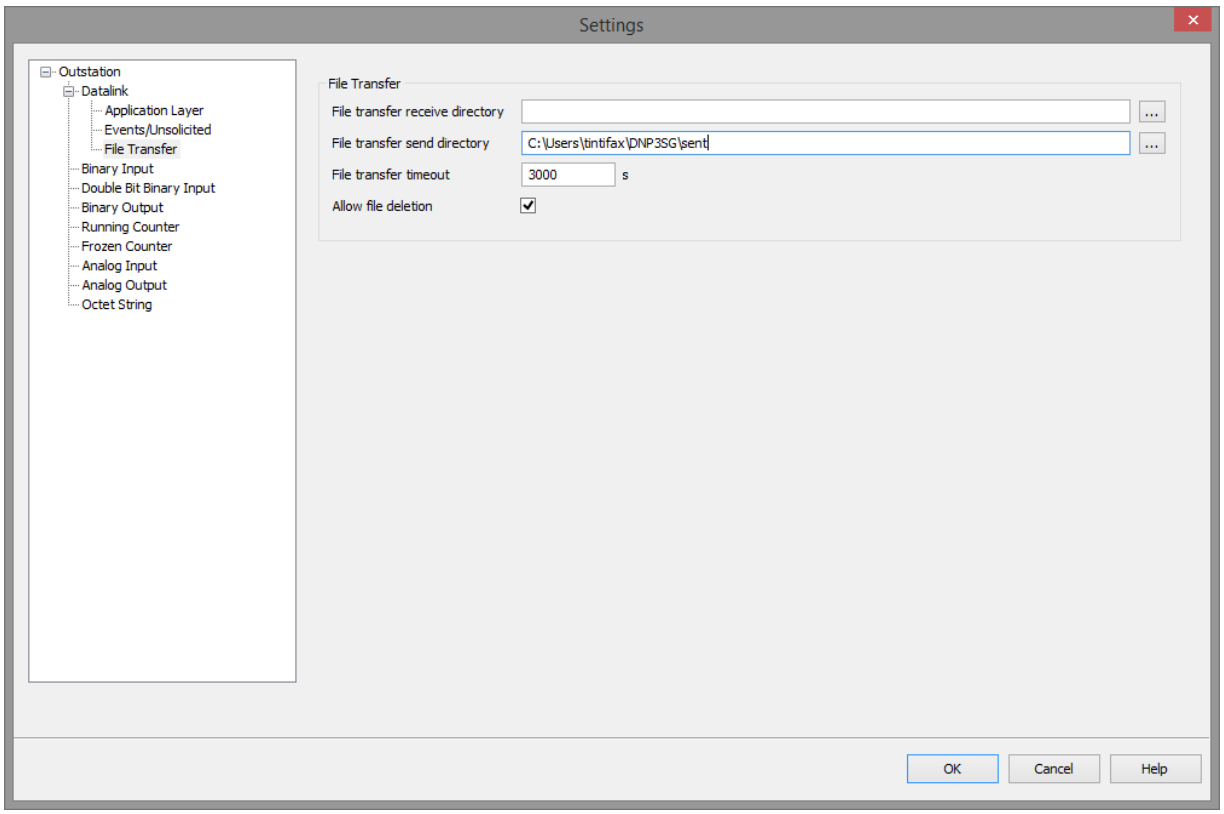
The options in this area are only available if the **Allow unsolicited responses** option has been activated.

Parameter	Description
Allow unsolicited responses	Check box to activate the use of **Unsollicited responses**. ▶ `Active:` **Unsolicited responses** are permitted and can be activated by the master. ▶ `Inactive:` **Unsolicited responses** are not permitted and cannot be activated by the master. Attention: No **unsolicited responses** are sent to the master. Even if these have been requested by the master. Default: `Inactive` Note: Match the configuration of this option to the configuration of the master.

Unsolicited timeout	Time that is waited for an `Application Confirmation` by the master for a previous `unsolicited Response` from the outstation. Input range: `1 to 60 s` Default: `5 s`
Unsolicited retries	Number of recurrences for an `Unsolicited Response` (sent by the outstation) after the **Unsolicited timeout** time has expired without an `Application Confirmation` being received by the master. Input range: `0 – 65535` (`0` = an infinite amount of attempts) Default: `3` Example: If an `Unsolicited Response` is sent to the master, the master should send an `Application Confirmation`. If this confirmation from the master remains outstanding, a repeat is sent as many times as configured in **Number of retries**. A new `Unsolicited Response` is then created.
Unsolicited resumption delay	Number of seconds that the outstation waits until it continues sending `Unsolicited Responses` again, if, after expiry of the **Unsolicited timeout** time and the sending of the repetitions, no `Application Confirmation` has been received by the master. Default: `15 s`
Class 1 events	Maximum number of events for the event class that can be buffered before an `Unsolicited Response` is sent to the master. Input range: `0 – 255` Default: `3`
Class 1 time	Maximum delay of the event class that is waited until an `Unsolicited Response` is sent to the master after an event has occurred. Input range: `0 – 65535 seconds` (`0` = no delay, i.e. for each event) Default: `5` Note: If new events are continually generated, it may be the case that no `Unsolicited Response` is sent to the master if the value is greater than 0.

Class 2 events	Maximum number of events for the event class that can be buffered before an `Unsolicited Response` is sent to the master. Input range: `0 - 255` Default: `3`
Class 2 time	Maximum delay of the event class that is waited until an `Unsolicited Response` is sent to the master after an event has occurred. Input range: `0 - 65535 seconds` (`0` = no delay, i.e. for each event) Default: `5` Note: If new events are continually generated, it may be the case that no `Unsolicited Response` is sent to the master if the value is greater than 0.
Class 3 events	Maximum number of events for the event class that can be buffered before an `Unsolicited Response` is sent to the master. Input range: `0 - 255` Default: `3`
Class 3 time	Maximum delay of the event class that is waited until an `Unsolicited Response` is sent to the master after an event has occurred. Input range: `0 - 65535 seconds` (`0` = no delay, i.e. for each event) Default: `5` Note: If new events are continually generated, it may be the case that no `Unsolicited Response` is sent to the master if the value is greater than 0.

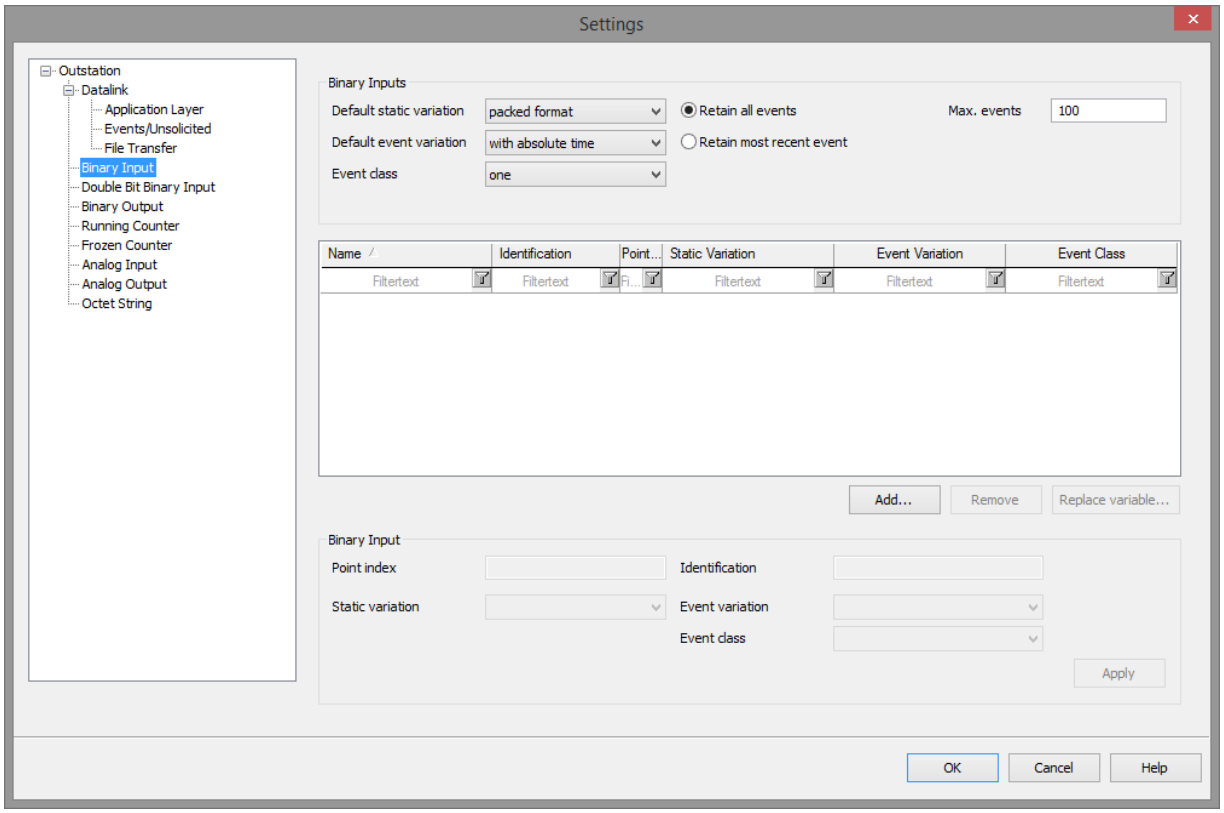
7.3.3 File Transfer



In this dialog, you configure the parameters for file transfer from and to the outstation.
A master can access subdirectories in the save location but not higher-level directories (\..\..\)

Parameter	Description
File transfer receive directory	Save location for the files sent to the outstation by the master. Click the **...** button and the dialog opens to select a save location. Note: If the save location is configured differently to the **File Transfer send directory** option, the master can neither read nor delete the files that it writes itself.
File transfer send directory	Save location for files that can be read by the master. Click the **...** button and the dialog opens to select a save location.
File transfer timeout	Time in seconds until the outstation closes the `File Handle` provided this `File Handle` has not already been closed by the master. Default: `3000 s` Note: The outstation only supports a single file handle.
Allow file deletion	Deletion by the master ▶ `Active:` Files can be deleted by the master. ▶ `Inactive:` The master cannot delete any files. Default: `active`

7.4 Binary Input



You configure the **DNP3 Binary Input points** in this node.

BINARY INPUTS

Parameter	Description
Default static variation	This option determines the default static variation for this Object Group for newly-added variables. A different static variation can be individually configured for each variable. Note: Select a static variation that is supported by the master. Not every master supports all static variations. Select from drop-down list: ▸ `packed format` Object Group 1 - Variation 1 ▸ `with flags` Object Group 1 - Variation 2 Default: `packed format`
Event retention	Behavior when forwarding value changes. Selection from option field. ▸ `Retain all events` Each time a value, status or time stamp of a variable is changed, a new event is created and saved in the event buffer. The master gets all changes. ▸ `Retain most recent event` The event buffer buffer only contains one event for the last change of value, status or time stamp. The master only gets the last change and, under certain circumstances, cannot distinguish whether or how often a value has changed in the meantime. Default: `Retain all events`

Max. events	Maximum number of events that are saved in the event buffer for this Object Group for all defined variables. If `Retain all events` is configured, this number should be at least as large as the number of configured variables. Default: `100` Note: Only active if `Retain all events` is active.
Default event variation	This option determines the default event variation for this Object Group for newly-added variables. A different event variation can be individually configured for each variable. Note: Select a event variation that is supported by the master. Not every master supports all event variations. Select from drop-down list: ▸ `without time` Object Group 2 - Variation 1 ▸ `with absolute time` Object Group 2 - Variation 2 ▸ `with relative time` Object Group 2 - Variation 3 Default: `with absolute time`
Default event class	This option determines the default event class for this Object Group for newly-added variables. A different event class can be individually configured for each variable. Select from drop-down list: ▸ `none` Variables are not assigned to an event class and do not generate events. No static objects are included for the variables in a response to a class 0 poll from the master. A master can only read the current value of the variables by means of a read request for the respective static object group. A master can assign the variables by means of an assign class request, not another event class. ▸ `zero` Variables are assigned to the event class 0, but do

	not generate any events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the variables by means of a read request for the respective static object group. A master can assign the variables by means of an assign class request, the event class 1, 2 or 3, as a result of which the variables also generate events from this point in time. ▸ `one` Variables are assigned to the event class 1, and generate events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the current value of the variables by means of a read request for the respective static object group. In a response to a class 1 poll from the master , any event objects that may be present are included for the events of the variables. A master can assign the variables by means of an assign class request, the event class 0, 2 or 3. If the variable is assigned by the master of event class 0, no more events are generated. ▸ `two` Variables are assigned to the event class 2, and generate events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the current value of the variables by means of a read request for the respective static object group. In a response to a class 2 poll from the master , any event objects that may be present are included for the events of the variables. A master can assign the variables by means of an assign class request, the event class 0, 1 or 3. If the variable is assigned by the master of event class 0, no more events are generated. ▸ `three` Variables are assigned to the event class 3, and generate events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the current value of the variables by means of a read request for the respective static object group. In a response to a class 3 poll from the master , any event objects that may be present are included for the events of the variables. A master can assign the variables by means of an assign class request, the event class 0, 1 or 2. If the variable is assigned by the master of event class 0, no more events are generated. Default: `one`

Variables taken from the zenon project are listed with their current configuration in this list.

- The list can be sorted
  - Click for the sorting on the column heading.
  - The sorting sequence is visualized with an arrow symbol next to the column heading:
    `Arrow upwards`: ascending sorting
    `Arrow downwards`: descending sorting
  - Another click on the column heading reverses the sorting order.
- The list can be filtered
  To filter the list:
  - Enter the desired filter term in the input field below the heading. The default description of an empty field is `filter text` (shown in gray font).

Parameter	Description
`Name`	Variable name. The project name is placed in front of the name: `ProjectName#VariableName`
`Identification`	Identification of the variable.
`Point Index`	Current configuration of the DNP3 point index of the variable in this Object Group.
`Static Variation`	Current configuration of the static variation for the point of the variable in this Object Group.
`Event Variation`	Current configuration of the event variation for the point of the variable in this Object Group. Note: This parameter is not available in the **Octet String** node.
`Event Class`	Current configuration of the event class for the point of the variable in this Object Group.
`Control Model`	Current configuration of the **Control Model**. Note: This parameter is only available in the **Binary Output** node.
`Command Routing`	Current behavior of the Binary Output point with select and operate requests from the master. Note: This parameter is only available in the **Binary Output** node.

Supported data types: `BOOL, USINT`

NAVIGATION BAR

Parameter	Description
Add...	Opens the variable selection dialog (à la page 108) to apply the zenon variables in the list. Note: Not active in the **Frozen Counter** node.
Remove	Removes selected variables from the list. A confirmation dialog is shown before a variable is deleted. Only active if one or more variables in the list are selected. Note: Not active in the **Frozen Counter** node.
Replace variable...	Opens the variable selection dialog (à la page 108) to replace a variable selected in the list with another zenon variable. Only active if precisely one variable in the list is selected.

BINARY INPUT

You configure the parameters for a selected variable in the list in this area. Editing by means of multiple selection is not possible.

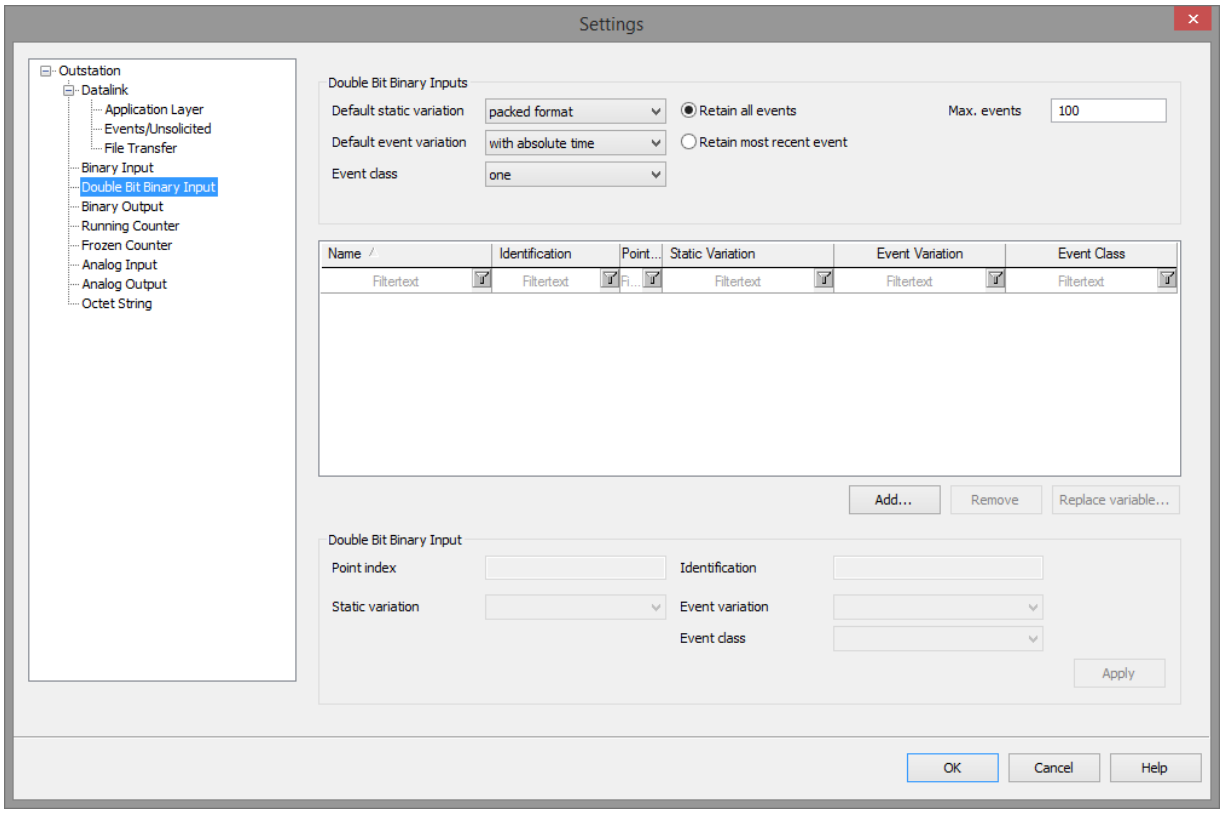
Parameter	Description
Point index	This option determines the DNP3 point index in the Object Group for the selected variable. If variables are added, the point index is automatically set for the newly-added variables and numbered consecutively starting with the highest point index already present. A point index can only be issued once within an Object Group. The entry is validated. If a point index has already been issued, this is shown with a warning dialog. Note: It is recommended that you issue the point index starting with 0 without gaps. The outstation supports the maximum point index. If a higher point index is used, it should be ensured that the master also supports qualifier codes with 2 octets or 4 octets accordingly.
Identification	The identification is primarily taken from the variable from zenon Runtime. The identification can also be amended locally with this option.
Static variation	This option determines the static variation for the selected variable for the point in this Object Group. With the `default` value, the general static variation defined for this Object Group is used. Note: Select a static variation that is supported by the master. Not every master supports all static variations. Select from drop-down list: ▸ `packed format` Object Group 1 - Variation 1 ▸ `with flags` Object Group 1 - Variation 2 Default: `default`

Event variation	This option determines the event variation for the selected variable for the point in this Object Group. With the `default` value, the general event variation defined for this Object Group is used. Note: Select a event variation that is supported by the master. Not every master supports all event variations. Select from drop-down list: ▸ `without time` Object Group 2 - Variation 1 ▸ `with absolute time` Object Group 2 - Variation 2 ▸ `with relative time` Object Group 2 - Variation 3 Default: `default`
Event class	This option determines the event class for the selected variable for the point in this Object Group. With the `default` value, the general event class defined for this Object Group is used. Selection: see description for **Default event class** option. Default: `default`
Apply	Applies the configuration for the selected variable(s).

FERMER BOÎTE DE DIALOGUE

Options	Description
OK	Applique les paramètres et ferme la boîte de dialogue.
Annuler	Annule toutes les modifications et ferme la boîte de dialogue.
Aide	Ouvre l'aide en ligne.

7.5 Double Bit Binary Input



You configure the **DNP3 Double bit binary Input points** in this node.

Parameter	Description
Default static variation	This option determines the default static variation for this Object Group for newly-added variables. A different static variation can be individually configured for each variable. Note: Select a static variation that is supported by the master. Not every master supports all static variations. Select from drop-down list: ▸ `packed format` Object Group 3 - Variation 1 ▸ `with flags` Object Group 3 - Variation 2 Default: `packed format`
Event retention	Behavior when forwarding value changes. Selection from option field. ▸ `Retain all events` Each time a value, status or time stamp of a variable is changed, a new event is created and saved in the event buffer. The master gets all changes. ▸ `Retain most recent event` The event buffer buffer only contains one event for the last change of value, status or time stamp. The master only gets the last change and, under certain circumstances, cannot distinguish whether or how often a value has changed in the meantime. Default: `Retain all events`
Max. events	Maximum number of events that are saved in the event buffer for this Object Group for all defined variables. If `Retain all events` is configured, this number should be at least as large as the number of configured variables. Default: `100` Note: Only active if `Retain all events` is active.
Default event variation	This option determines the default event variation for this Object Group for newly-added variables. A different event variation can be individually configured for each variable.

	Note: Select a event variation that is supported by the master. Not every master supports all event variations. Select from drop-down list: ▶ `without time` Object Group 4 - Variation 1 ▶ `with absolute time` Object Group 4 - Variation 2 ▶ `with relative time` Object Group 4 - Variation 3 Default: `with absolute time`

Event class	This option determines the default event class for this Object Group for newly-added variables. A different event class can be individually configured for each variable. Select from drop-down list: ▸ none Variables are not assigned to an event class and do not generate events. No static objects are included for the variables in a response to a class 0 poll from the master. A master can only read the current value of the variables by means of a read request for the respective static object group. A master can assign the variables by means of an assign class request, not another event class. ▸ zero Variables are assigned to the event class 0, but do not generate any events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the variables by means of a read request for the respective static object group. A master can assign the variables by means of an assign class request, the event class 1, 2 or 3, as a result of which the variables also generate events from this point in time. ▸ one Variables are assigned to the event class 1, and generate events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the current value of the variables by means of a read request for the respective static object group. In a response to a class 1 poll from the master , any event objects that may be present are included for the events of the variables. A master can assign the variables by means of an assign class request, the event class 0, 2 or 3. If the variable is assigned by the master of event class 0, no more events are generated. ▸ two Variables are assigned to the event class 2, and generate events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the current value of the variables by means of a read request for the respective static object group. In a response to a class 2 poll from the master , any event objects that may be present are included for the events of the variables. A master can assign the variables by

	means of an assign class request, the event class 0, 1 or 3. If the variable is assigned by the master of event class 0, no more events are generated. ▸ `three` Variables are assigned to the event class 3, and generate events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the current value of the variables by means of a read request for the respective static object group. In a response to a class 3 poll from the master , any event objects that may be present are included for the events of the variables. A master can assign the variables by means of an assign class request, the event class 0, 1 or 2. If the variable is assigned by the master of event class 0, no more events are generated. ▸ Default: `one`

Variables taken from the zenon project are listed with their current configuration in this list.

- The list can be sorted
  - Click for the sorting on the column heading.
  - The sorting sequence is visualized with an arrow symbol next to the column heading:
    `Arrow upwards`: ascending sorting
    `Arrow downwards`: descending sorting
  - Another click on the column heading reverses the sorting order.
- The list can be filtered
  To filter the list:
  - Enter the desired filter term in the input field below the heading. The default description of an empty field is `filter text` (shown in gray font).

Parameter	Description
`Name`	Variable name. The project name is placed in front of the name: `ProjectName#VariableName`
`Identification`	Identification of the variable.
`Point Index`	Current configuration of the DNP3 point index of the variable in this Object Group.
`Static Variation`	Current configuration of the static variation for the point of the variable in this Object Group.
`Event Variation`	Current configuration of the event variation for the point of the variable in this Object Group. Note: This parameter is not available in the **Octet String** node.
`Event Class`	Current configuration of the event class for the point of the variable in this Object Group.
`Control Model`	Current configuration of the **Control Model**. Note: This parameter is only available in the **Binary Output** node.
`Command Routing`	Current behavior of the Binary Output point with select and operate requests from the master. Note: This parameter is only available in the **Binary Output** node.

Supported data types: `BOOL, USINT`

NAVIGATION BAR

Parameter	Description
Add...	Opens the variable selection dialog (à la page 108) to apply the zenon variables in the list. Note: Not active in the **Frozen Counter** node.
Remove	Removes selected variables from the list. A confirmation dialog is shown before a variable is deleted. Only active if one or more variables in the list are selected. Note: Not active in the **Frozen Counter** node.
Replace variable...	Opens the variable selection dialog (à la page 108) to replace a variable selected in the list with another zenon variable. Only active if precisely one variable in the list is selected.

DOUBLE BIT BINARY INPUT

You configure the parameters for selected variable(s) in the list in this area.

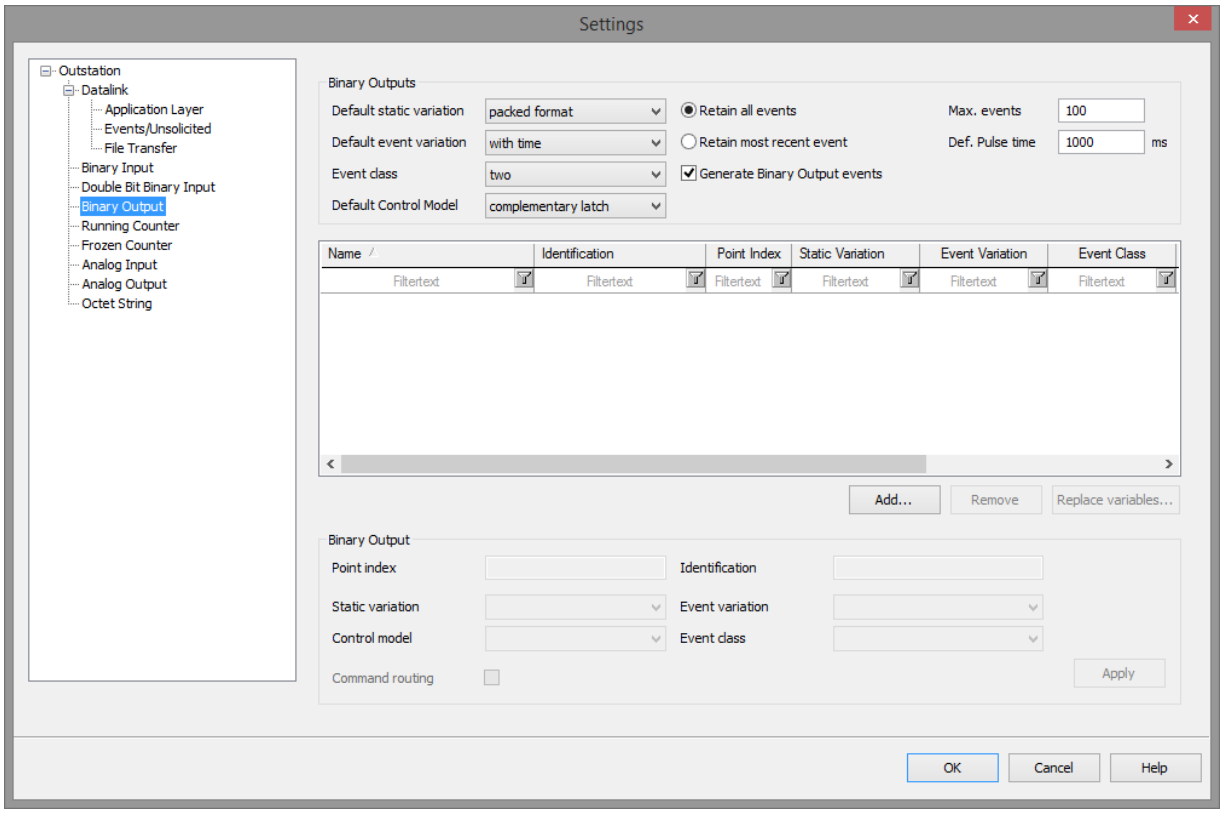
Parameter	Description
Point index	This option determines the DNP3 point index in the Object Group for the selected variable. If variables are added, the point index is automatically set for the newly-added variables and numbered consecutively starting with the highest point index already present. A point index can only be issued once within an Object Group. The entry is validated. If a point index has already been issued, this is shown with a warning dialog. Note: It is recommended that you issue the point index starting with 0 without gaps. The outstation supports the maximum point index. If a higher point index is used, it should be ensured that the master also supports qualifier codes with 2 octets or 4 octets accordingly.
Identification	The identification is primarily taken from the variable from zenon Runtime. The identification can also be amended locally with this option.
Static variation	This option determines the static variation for the selected variable for the point in this Object Group. With the `default` value, the general static variation defined for this Object Group is used. Note: Select a static variation that is supported by the master. Not every master supports all static variations. Select from drop-down list: ▸ `packed format` Object Group 3 - Variation 1 ▸ `with flags` Object Group 3 - Variation 2 Default: `default`

Event variation	This option determines the event variation for the selected variable for the point in this Object Group. With the `default` value, the general event variation defined for this Object Group is used. Note: Select a event variation that is supported by the master. Not every master supports all event variations. Select from drop-down list: ▸ `without time` Object Group 4 - Variation 1 ▸ `with absolute time` Object Group 4 - Variation 2 ▸ with relative time Object Group 4 - Variation 3 Default: `default`
Event class	This option determines the event class for the selected variable for the point in this Object Group. With the `default` value, the general event class defined for this Object Group is used. Selection: see description for **Default event class** option. Default: `default`
Apply	Applies the configuration for the selected variable(s).

FERMER BOÎTE DE DIALOGUE

Options	Description
OK	Applique les paramètres et ferme la boîte de dialogue.
Annuler	Annule toutes les modifications et ferme la boîte de dialogue.
Aide	Ouvre l'aide en ligne.

7.6 Binary Output



You configure the **DNP3 Binary Output points** in this node.

Parameter	Description
Default static variation	This option determines the default static variation for this Object Group for newly-added variables. A different static variation can be individually configured for each variable. Note: Select a static variation that is supported by the master. Not every master supports all static variations. Select from drop-down list: ▸ `packed format` Object Group 10 - Variation 1 ▸ `with flags` Object Group 10 - Variation 2 Default: `packed format`
Event retention	Behavior when forwarding value changes. Selection from option field. ▸ `Retain all events` Each time a value, status or time stamp of a variable is changed, a new event is created and saved in the event buffer. The master gets all changes. ▸ `Retain most recent event` The event buffer buffer only contains one event for the last change of value, status or time stamp. The master only gets the last change and, under certain circumstances, cannot distinguish whether or how often a value has changed in the meantime. Default: `Retain all events`
Max. events	Maximum number of events that are saved in the event buffer for this Object Group for all defined variables. If `Retain all events` is configured, this number should be at least as large as the number of configured variables. Default: `100` Note: Only active if `Retain all events` is active.

Default event variation	This option determines the default event variation for this Object Group for newly-added variables. A different event variation can be individually configured for each variable. Note: Select a event variation that is supported by the master. Not every master supports all event variations. Select from drop-down list: ▶ `without time` Object Group 11 - Variation 1 ▶ `with time` Object Group 11 - Variation 2 Default: `with time`
Event Class	This option determines the default event class for this Object Group for newly-added variables. A different event class can be individually configured for each variable. Select from drop-down list: ▸ `none` Variables are not assigned to an event class and do not generate events. No static objects are included for the variables in a response to a class 0 poll from the master. A master can only read the current value of the variables by means of a read request for the respective static object group. A master can assign the variables by means of an assign class request, not another event class. ▸ `zero` Variables are assigned to the event class 0, but do not generate any events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the variables by means of a read request for the respective static object group. A master can assign the variables by means of an assign class request, the event class 1, 2 or 3, as a result of which the variables also generate events from this point in time. ▸ `one` Variables are assigned to the event class 1, and generate events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the current value of the variables by means of a read request for the

	respective static object group. In a response to a class 1 poll from the master , any event objects that may be present are included for the events of the variables. A master can assign the variables by means of an assign class request, the event class 0, 2 or 3. If the variable is assigned by the master of event class 0, no more events are generated. ▸ `two` Variables are assigned to the event class 2, and generate events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the current value of the variables by means of a read request for the respective static object group. In a response to a class 2 poll from the master , any event objects that may be present are included for the events of the variables. A master can assign the variables by means of an assign class request, the event class 0, 1 or 3. If the variable is assigned by the master of event class 0, no more events are generated. ▸ `three` Variables are assigned to the event class 3, and generate events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the current value of the variables by means of a read request for the respective static object group. In a response to a class 3 poll from the master , any event objects that may be present are included for the events of the variables. A master can assign the variables by means of an assign class request, the event class 0, 1 or 2. If the variable is assigned by the master of event class 0, no more events are generated. Default: `two`

Default Pulse time	This option determines the **Default Pulse ON time**. The variable linked for the Binary Output point is set, for the duration of the time defined here, to the value "1", "3", "4", "65" or "129" (depending on the data type of the variable for the Binary Output) The variable is subsequently set to the value 0 if the Master sends a Pulse ON control command with the Pulse On value "0" (default). This is applicable for **Control Model** with the configuration "Activation" or "complementary two output". If the Master sends a value for Pulse On that is different to "0" (default) with the Pulse On control command, the time from the Request is used by the Master provided this time is not more or less than five times the value of the **Default Pulse time**. If the value is outside, the outstation uses the **Default Pulse time** configured here. Note: The **Default pulse time** is not used for Control Model complementary latch. Default: 1000 ms
Generate Binary Output events	This option determines whether value changes for variables that are configured for Binary Outputs , events are also to be generated. Not all masters support events for Binary Outputs. Default: active
Default Control Model	This option determines the default **Control Model** for newly-added variables. Note: Certain combinations of Master control requests and data type (BOOL or USINT) of the variable are not possible. Also depending on whether **Command Routing** is active or not active for a selected variable. Select from drop-down list: ▸ activation

	▸ `complementary latch` ▸ `complementary two output` Default: `complementary latch` Note: For more information, refer to the overview table in the "**LATCH ON und LATCH_OFF** (à la page 117)" chapter in this manual.

Variables taken from the zenon project are listed with their current configuration in this list.

- The list can be sorted
  - Click for the sorting on the column heading.
  - The sorting sequence is visualized with an arrow symbol next to the column heading:
    `Arrow upwards`: ascending sorting
    `Arrow downwards`: descending sorting
  - Another click on the column heading reverses the sorting order.
- The list can be filtered
  To filter the list:
  - Enter the desired filter term in the input field below the heading. The default description of an empty field is `filter text` (shown in gray font).

Parameter	Description
Name	Variable name. The project name is placed in front of the name: ProjectName#VariableName
Identification	Identification of the variable.
Point Index	Current configuration of the DNP3 point index of the variable in this Object Group.
Static Variation	Current configuration of the static variation for the point of the variable in this Object Group.
Event Variation	Current configuration of the event variation for the point of the variable in this Object Group. Note: This parameter is not available in the **Octet String** node.
Event Class	Current configuration of the event class for the point of the variable in this Object Group.
Control Model	Current configuration of the **Control Model**. Note: This parameter is only available in the **Binary Output** node.
Command Routing	Current behavior of the Binary Output point with select and operate requests from the master. Note: This parameter is only available in the **Binary Output** node.

Supported data types: UDINT, UINT

NAVIGATION BAR

Parameter	Description
Add...	Opens the variable selection dialog (à la page 108) to apply the zenon variables in the list. Note: Not active in the **Frozen Counter** node.
Remove	Removes selected variables from the list. A confirmation dialog is shown before a variable is deleted. Only active if one or more variables in the list are selected. Note: Not active in the **Frozen Counter** node.
Replace variable...	Opens the variable selection dialog (à la page 108) to replace a variable selected in the list with another zenon variable. Only active if precisely one variable in the list is selected.

BINARY OUTPUT

You configure the parameters for selected variable(s) in the list in this area.

Parameter	Description
Point index	This option determines the DNP3 point index in the Object Group for the selected variable. If variables are added, the point index is automatically set for the newly-added variables and numbered consecutively starting with the highest point index already present. A point index can only be issued once within an Object Group. The entry is validated. If a point index has already been issued, this is shown with a warning dialog. Note: It is recommended that you issue the point index starting with 0 without gaps. The outstation supports the maximum point index. If a higher point index is used, it should be ensured that the master also supports qualifier codes with 2 octets or 4 octets accordingly.
Identification	The identification is primarily taken from the variable from zenon Runtime. The identification can also be amended locally with this option.
Static variation	This option determines the static variation for the selected variable for the point in this Object Group. With the `default` value, the general static variation defined for this Object Group is used. Note: Select a static variation that is supported by the master. Not every master supports all static variations. Select from drop-down list: ▸ `packed format` Object Group 10 - Variation 1 ▸ `with flags` Object Group 10 - Variation 2 Default: `default`

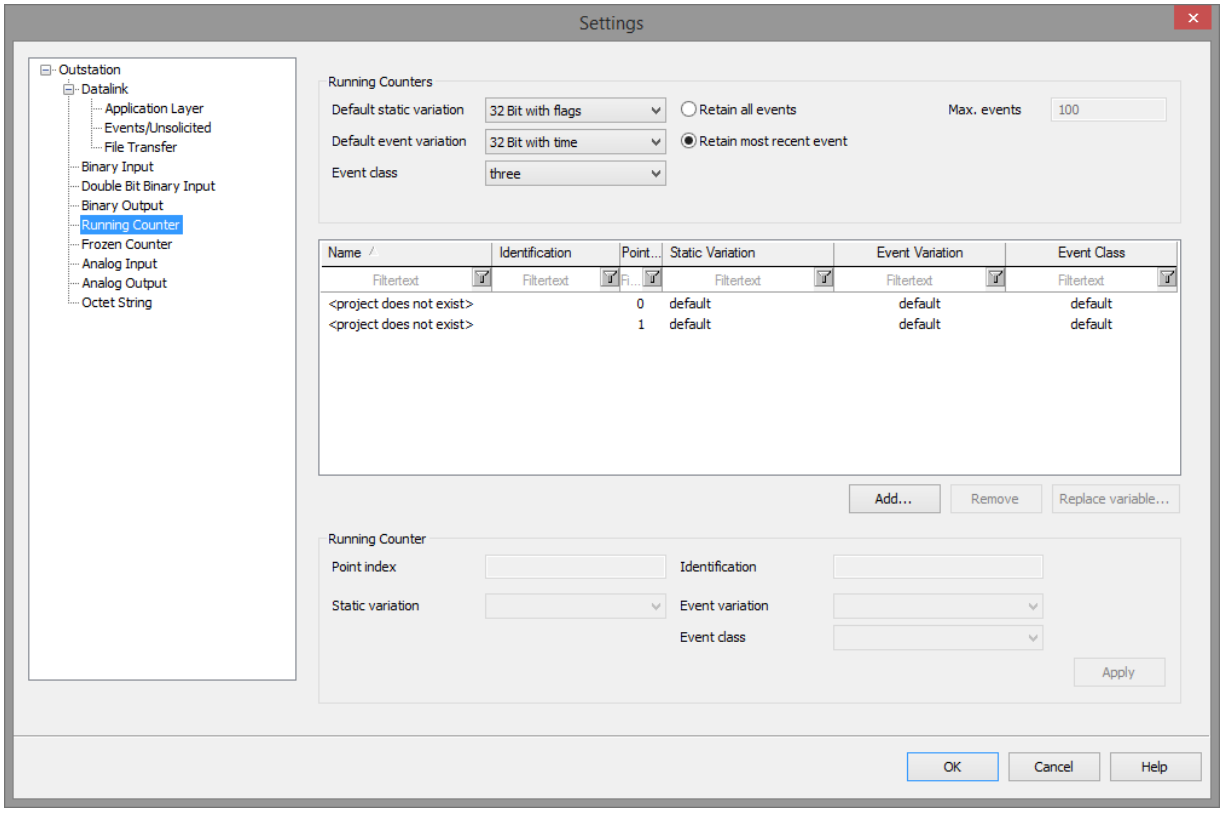
Event variation	This option determines the event variation for the selected variable for the point in this Object Group. With the `default` value, the general event variation defined for this Object Group is used. Note: Select a event variation that is supported by the master. Not every master supports all event variations. Select from drop-down list: ▸ `without time` Object Group 11 - Variation 1 ▸ `with time` Object Group 11 - Variation 2 Default: `default`
Control model	This option determines the Control Model for the selected variable. Note: Certain combinations of Master control requests and data type (BOOL or USINT) of the variable are not possible. Also depending on whether **Command Routing** is active or not active for a selected variable. Select from drop-down list: ▸ `activation` Object Group n - Variation n ▸ `complementary latch` Object Group n - Variation n ▸ `complementary two output` Object Group n - Variation n Default: `default` You can find detailed information on this in the **"Command processing - Command Routing"** (à la page 114) chapter.

Event class	This option determines the event class for the selected variable for the point in this Object Group. With the `default` value, the general event class defined for this Object Group is used. Selection: see description for **Default event class** option. Default: `default`
Command routing	This option determines the behavior of Binary Output point for the selected variable for select and operate requests from the master. ▶ `Active:` select requests from the master and operate requests from the master are routed further to the **command input** in zenon Runtime. Only once a response to the select or operate has been received by the PLC (to the **command input** in Runtime), a corresponding `positive`, `negative select response` or `operate response` is sent to the master by the outstation. Direct operate requests Are immediately responded to positively by the Outstation if the linked variable has been written in Runtime. ▶ `Inactive:` select requests Is always responded to immediately by the Outstation if a variable has been configured for the Binary Output point and the variable is present in Runtime. operate requests and direct operate requests are immediately responded to positively by the outstation if the linked variable has been written in Runtime. Default: `active` You can find further details on **Command routing** in the "**Select before Operate** (à la page 114)" chapter.
Apply	Applies the configuration for the selected variable(s).

FERMER BOÎTE DE DIALOGUE

Options	Description
OK	Applique les paramètres et ferme la boîte de dialogue.
Annuler	Annule toutes les modifications et ferme la boîte de dialogue.
Aide	Ouvre l'aide en ligne.

7.7 Running Counter



You configure the **DNP3 Running counter points** in this node.

RUNNING COUNTER

Parameter	Description
Default static variation	This option determines the event variation for the selected variable for the point in this Object Group. With the `default` value, the general event variation defined for this Object Group is used. Note: Select a event variation that is supported by the master. Not every master supports all event variations. Select from drop-down list: ▸ `32 Bit with flags` Object Group 20 - Variation 1 ▸ `16 Bit with flags` Object Group 20 - Variation 2 ▸ `32 Bit without flags` Object Group 20 - Variation 5 ▸ `16 Bit without flags` Object Group 20 - Variation 6 Default: `32 Bit with flags`
Event retention	Behavior when forwarding value changes. Selection from option field. ▸ `Retain all events` Each time a value, status or time stamp of a variable is changed, a new event is created and saved in the event buffer. The master gets all changes. ▸ `Retain most recent event` The event buffer buffer only contains one event for the last change of value, status or time stamp. The master only gets the last change and, under certain circumstances, cannot distinguish whether or how often a value has changed in the meantime. Default: `Retain most recent event`
Max. events	Maximum number of events that are saved in the event buffer for this Object Group for all defined variables. If `Retain all events` is configured, this number should be at least as large as the number of configured variables. Default: `100`

	Note: Only active if `Retain all events` is active.
Default event variation	This option determines the default event variation for this Object Group for newly-added variables. A different event variation can be individually configured for each variable. Note: Select a event variation that is supported by the master. Not every master supports all event variations. Select from drop-down list: ▸ `32 Bit with flags` Object Group 22 - Variation 1 ▸ `16 Bit with flags` Object Group 22 - Variation 2 ▸ `32 Bit with time` Object Group 22 - Variation 5 ▸ `16 Bit with time` Object Group 22 - Variation 6 Default: `32 Bit with time`
Event class	This option determines the default event class for this Object Group for newly-added variables. A different event class can be individually configured for each variable. Select from drop-down list: ▸ `none` Variables are not assigned to an event class and do not generate events. No static objects are included for the variables in a response to a class 0 poll from the master. A master can only read the current value of the variables by means of a read request for the respective static object group. A master can assign the variables by means of an assign class request, not another event class. ▸ `zero` Variables are assigned to the event class 0, but do not generate any events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the variables by means of a read request for the respective static object group. A master can assign the variables by means of an assign class request, the event class 1, 2 or 3, as a result of which the variables also generate events from this point in time.

	▸ one Variables are assigned to the event class 1, and generate events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the current value of the variables by means of a read request for the respective static object group. In a response to a class 1 poll from the master , any event objects that may be present are included for the events of the variables. A master can assign the variables by means of an assign class request, the event class 0, 2 or 3. If the variable is assigned by the master of event class 0, no more events are generated. ▸ two Variables are assigned to the event class 2, and generate events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the current value of the variables by means of a read request for the respective static object group. In a response to a class 2 poll from the master , any event objects that may be present are included for the events of the variables. A master can assign the variables by means of an assign class request, the event class 0, 1 or 3. If the variable is assigned by the master of event class 0, no more events are generated. ▸ three Variables are assigned to the event class 3, and generate events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the current value of the variables by means of a read request for the respective static object group. In a response to a class 3 poll from the master , any event objects that may be present are included for the events of the variables. A master can assign the variables by means of an assign class request, the event class 0, 1 or 2. If the variable is assigned by the master of event class 0, no more events are generated. Default: three

Variables taken from the zenon project are listed with their current configuration in this list.

- The list can be sorted
  - Click for the sorting on the column heading.
  - The sorting sequence is visualized with an arrow symbol next to the column heading:
    Arrow upwards: ascending sorting
    Arrow downwards: descending sorting

- Another click on the column heading reverses the sorting order.

▶ The list can be filtered
To filter the list:

- Enter the desired filter term in the input field below the heading. The default description of an empty field is `filter text` (shown in gray font).

Parameter	Description
`Name`	Variable name. The project name is placed in front of the name: `ProjectName#VariableName`
`Identification`	Identification of the variable.
`Point Index`	Current configuration of the DNP3 point index of the variable in this Object Group.
`Static Variation`	Current configuration of the static variation for the point of the variable in this Object Group.
`Event Variation`	Current configuration of the event variation for the point of the variable in this Object Group. Note: This parameter is not available in the **Octet String** node.
`Event Class`	Current configuration of the event class for the point of the variable in this Object Group.
`Control Model`	Current configuration of the **Control Model**. Note: This parameter is only available in the **Binary Output** node.
`Command Routing`	Current behavior of the Binary Output point with select and operate requests from the master. Note: This parameter is only available in the **Binary Output** node.

Supported data types: `UDINT, UINT`

NAVIGATION BAR

Parameter	Description
Add...	Opens the variable selection dialog (à la page 108) to apply the zenon variables in the list. Note: Not active in the **Frozen Counter** node.
Remove	Removes selected variables from the list. A confirmation dialog is shown before a variable is deleted. Only active if one or more variables in the list are selected. Note: Not active in the **Frozen Counter** node.
Replace variable...	Opens the variable selection dialog (à la page 108) to replace a variable selected in the list with another zenon variable. Only active if precisely one variable in the list is selected.

RUNNING COUNTER

You configure the parameters for selected variable(s) in the list in this area.

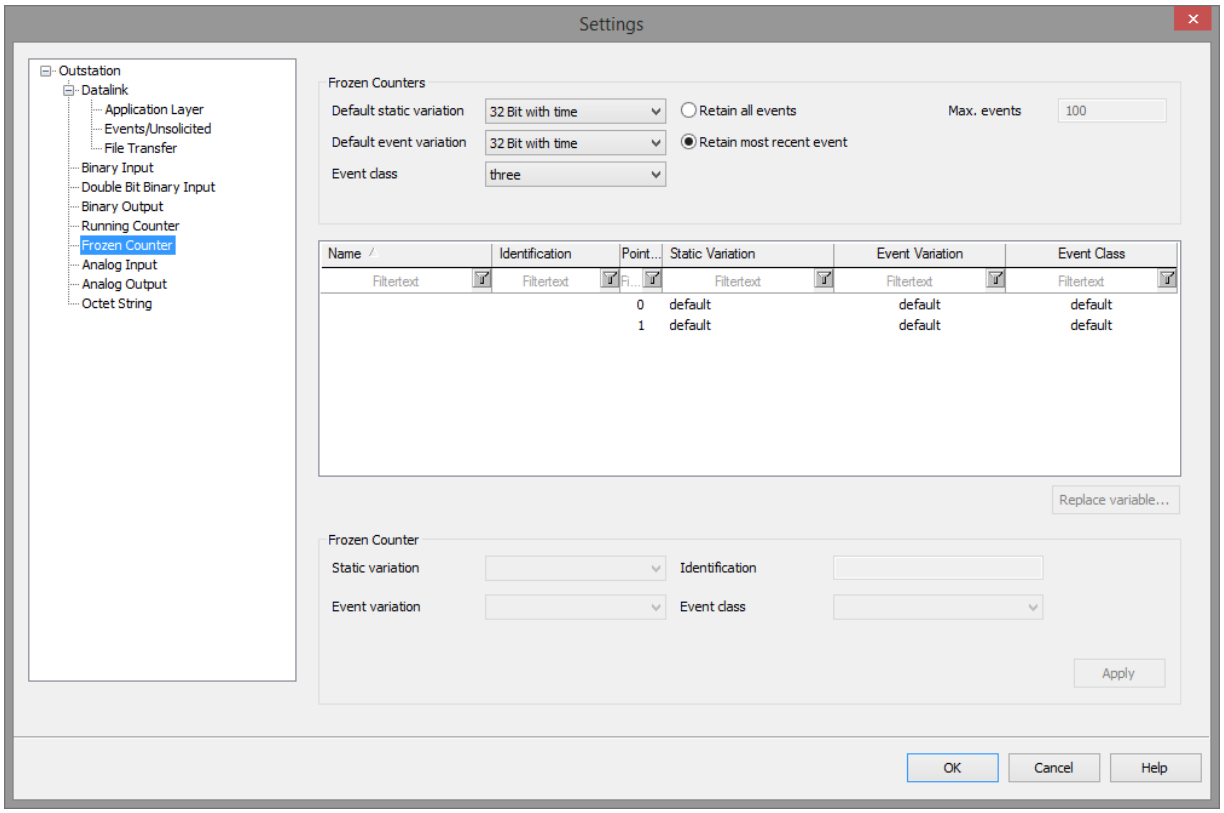
Parameter	Description
Point index	This option determines the DNP3 point index in the Object Group for the selected variable. If variables are added, the point index is automatically set for the newly-added variables and numbered consecutively starting with the highest point index already present. A point index can only be issued once within an Object Group. The entry is validated. If a point index has already been issued, this is shown with a warning dialog. Note: It is recommended that you issue the point index starting with 0 without gaps. The outstation supports the maximum point index. If a higher point index is used, it should be ensured that the master also supports qualifier codes with 2 octets or 4 octets accordingly.
Identification	The identification is primarily taken from the variable from zenon Runtime. The identification can also be amended locally with this option.
Static variation	This option determines the static variation for the selected variable for the point in this Object Group. With the `default` value, the general static variation defined for this Object Group is used. Note: Select a static variation that is supported by the master. Not every master supports all static variations. Select from drop-down list: ▸ `32 Bit with flags` Object Group 20 - Variation 1 ▸ `16 Bit with flags` Object Group 20 - Variation 2 ▸ `32 Bit without flags` Object Group 20 - Variation 5 ▸ `16 Bit without flags` Object Group 20 - Variation 6 Default: `default`
Event variation	This option determines the event variation for the selected variable for the point in this Object Group. With the `default` value, the general event variation defined for this Object Group is used.

	Note: Select a event variation that is supported by the master. Not every master supports all event variations. Select from drop-down list: ▸ `32 Bit with flags` Object Group 22 - Variation 1 ▸ `16 Bit with flags` Object Group 22 - Variation 2 ▸ `32 Bit with time` Object Group 22 - Variation 5 ▸ `16 Bit with time` Object Group 22 - Variation 6 Default: `default`
Event class	This option determines the event class for the selected variable for the point in this Object Group. With the `default` value, the general event class defined for this Object Group is used. Selection: see description for **Default event class** option. Default: `default`
Apply	Applies the configuration for the selected variable(s).

FERMER BOÎTE DE DIALOGUE

Options	Description
OK	Applique les paramètres et ferme la boîte de dialogue.
Annuler	Annule toutes les modifications et ferme la boîte de dialogue.
Aide	Ouvre l'aide en ligne.

7.8 Frozen Counter



You configure the **DNP3 Frozen counter points** in this node.

Parameter	Description
Default static variation	This option determines the default static variation for this Object Group for newly-added variables. A different static variation can be individually configured for each variable. Note: Select a static variation that is supported by the master. Not every master supports all static variations. Select from drop-down list: ▸ `32 Bit with flags` Object Group 21 - Variation 1 ▸ `16 Bit with flags` Object Group 21 - Variation 2 ▸ `32 Bit with time` Object Group 21 - Variation 5 ▸ `16 Bit with time` Object Group 21 - Variation 6 ▸ `32 Bit without flags` Object Group 21 - Variation 9 ▸ `16 Bit without flags` Object Group 21 - Variation 10 Default: `32 Bit with time`
Event retention	Behavior when forwarding value changes. Selection from option field. ▸ `Retain all events` Each time a value, status or time stamp of a variable is changed, a new event is created and saved in the event buffer. The master gets all changes. ▸ `Retain most recent event` The event buffer buffer only contains one event for the last change of value, status or time stamp. The master only gets the last change and, under certain circumstances, cannot distinguish whether or how often a value has changed in the meantime. Default: `Retain most recent event`
Max events	Maximum number of events that are saved in the event buffer for this Object Group for all defined variables. If `Retain all events` is configured, this number should

	be at least as large as the number of configured variables. Default: `100` Note: Only active if `Retain all events` is active.
Default event variations	This option determines the default event variation for this Object Group for newly-added variables. A different event variation can be individually configured for each variable. Note: Select a event variation that is supported by the master. Not every master supports all event variations. Select from drop-down list: ▸ `32 Bit with flags` Object Group 23 - Variation 1 ▸ `16 Bit with flags` Object Group 23 - Variation 2 ▸ `32 Bit with time` Object Group 23 - Variation 5 ▸ `16 Bit with time` Object Group 23 - Variation 6 Default: `32 Bit with time`
Event class	This option determines the default event class for this Object Group for newly-added variables. A different event class can be individually configured for each variable. Select from drop-down list: ▸ `none` Variables are not assigned to an event class and do not generate events. No static objects are included for the variables in a response to a class 0 poll from the master. A master can only read the current value of the variables by means of a read request for the respective static object group. A master can assign the variables by means of an assign class request, not another event class. ▸ `zero` Variables are assigned to the event class 0, but do not generate any events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the variables by means of a read request for the respective static object group. A master can assign the variables by means of an assign class request,

	the event class 1, 2 or 3, as a result of which the variables also generate events from this point in time. ▸ `one` Variables are assigned to the event class 1, and generate events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the current value of the variables by means of a read request for the respective static object group. In a response to a class 1 poll from the master , any event objects that may be present are included for the events of the variables. A master can assign the variables by means of an assign class request, the event class 0, 2 or 3. If the variable is assigned by the master of event class 0, no more events are generated. ▸ `two` Variables are assigned to the event class 2, and generate events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the current value of the variables by means of a read request for the respective static object group. In a response to a class 2 poll from the master , any event objects that may be present are included for the events of the variables. A master can assign the variables by means of an assign class request, the event class 0, 1 or 3. If the variable is assigned by the master of event class 0, no more events are generated. ▸ `three` Variables are assigned to the event class 3, and generate events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the current value of the variables by means of a read request for the respective static object group. In a response to a class 3 poll from the master , any event objects that may be present are included for the events of the variables. A master can assign the variables by means of an assign class request, the event class 0, 1 or 2. If the variable is assigned by the master of event class 0, no more events are generated. Default: `three`

Variables taken from the zenon project are listed with their current configuration in this list.

- The list can be sorted
  - Click for the sorting on the column heading.

- The sorting sequence is visualized with an arrow symbol next to the column heading:
  `Arrow upwards`: ascending sorting
  `Arrow downwards`: descending sorting
- Another click on the column heading reverses the sorting order.

▶ The list can be filtered
To filter the list:

- Enter the desired filter term in the input field below the heading. The default description of an empty field is `filter text` (shown in gray font).

Parameter	Description
`Name`	Variable name. The project name is placed in front of the name: `ProjectName#VariableName`
`Identification`	Identification of the variable.
`Point Index`	Current configuration of the DNP3 point index of the variable in this Object Group.
`Static Variation`	Current configuration of the static variation for the point of the variable in this Object Group.
`Event Variation`	Current configuration of the event variation for the point of the variable in this Object Group. Note: This parameter is not available in the **Octet String** node.
`Event Class`	Current configuration of the event class for the point of the variable in this Object Group.
`Control Model`	Current configuration of the **Control Model**. Note: This parameter is only available in the **Binary Output** node.
`Command Routing`	Current behavior of the Binary Output point with select and operate requests from the master. Note: This parameter is only available in the **Binary Output** node.

Supported data types: `UDINT, UINT`

NAVIGATION BAR

Parameter	Description
Add...	Opens the variable selection dialog (à la page 108) to apply the zenon variables in the list. Note: Not active in the **Frozen Counter** node.
Remove	Removes selected variables from the list. A confirmation dialog is shown before a variable is deleted. Only active if one or more variables in the list are selected. Note: Not active in the **Frozen Counter** node.
Replace variable...	Opens the variable selection dialog (à la page 108) to replace a variable selected in the list with another zenon variable. Only active if precisely one variable in the list is selected.

FROZEN COUNTER

You configure the parameters for selected variable(s) in the list in this area.

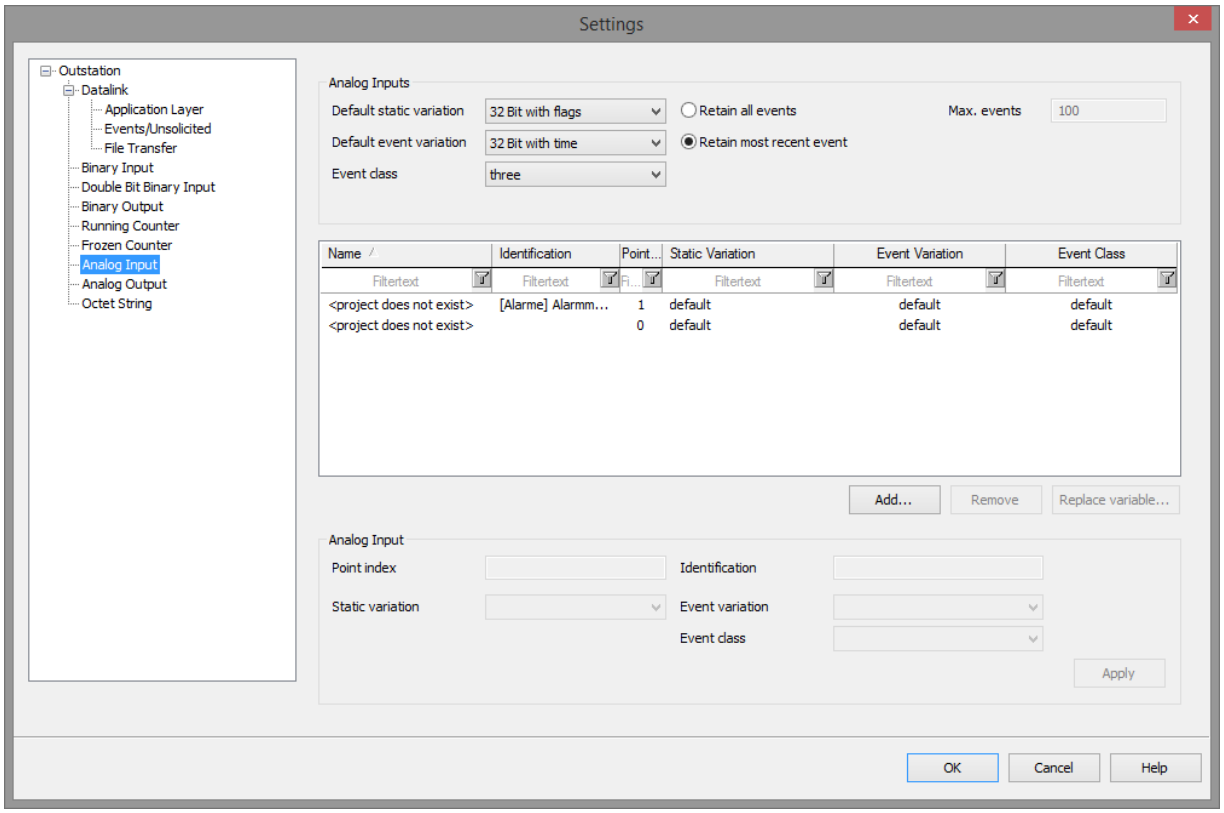
Parameter	Description
Static variation	This option determines the static variation for the selected variable for the point in this Object Group. With the `default` value, the general static variation defined for this Object Group is used. Note: Select a static variation that is supported by the master. Not every master supports all static variations. Select from drop-down list: ▸ `32 Bit with flags` Object Group 21 - Variation 1 ▸ `16 Bit with flags` Object Group 21 - Variation 2 ▸ `32 Bit with time` Object Group 21 - Variation 5 ▸ `16 Bit with time` Object Group 21 - Variation 6 ▸ `32 Bit without flags` Object Group 21 - Variation 9 ▸ `16 Bit without flags` Object Group 21 - Variation 10 Default: `default`
Identification	The identification is primarily taken from the variable from zenon Runtime. The identification can also be amended locally with this option.
Event variation	This option determines the event variation for the selected variable for the point in this Object Group. With the `default` value, the general event variation defined for this Object Group is used. Note: Select a event variation that is supported by the master. Not every master supports all event variations. Select from drop-down list: ▸ `32 Bit with flags` Object Group 23 - Variation 1 ▸ `16 Bit with flags` Object Group 23 - Variation 2 ▸ `w32 Bit with time` Object Group 23 - Variation 5

	▸ `16 Bit with time` Object Group 23 - Variation 6 Default: `default`
Event class	This option determines the event class for the selected variable for the point in this Object Group. With the `default` value, the general event class defined for this Object Group is used. Selection: see description for **Default event class** option. Default: `default`
Apply	Applies the configuration for the selected variable(s).

FERMER BOÎTE DE DIALOGUE

Options	Description
OK	Applique les paramètres et ferme la boîte de dialogue.
Annuler	Annule toutes les modifications et ferme la boîte de dialogue.
Aide	Ouvre l'aide en ligne.

7.9 Analog Input



You configure the **DNP3 Analog Input points** in this node.

Parameter	Description
Default static variation	This option determines the default static variation for this Object Group for newly-added variables. A different static variation can be individually configured for each variable. Note: Select a static variation that is supported by the master. Not every master supports all static variations. Select from drop-down list: ▸ `32 Bit with flags` Object Group 30 - Variation 1 ▸ `16 Bit with flags` Object Group 30 - Variation 2 ▸ `32 Bit without flags` Object Group 30 - Variation 3 ▸ `16 Bit without flags` Object Group 30 - Variation 4 ▸ `Real with flags (Single-precision with flag)` Object Group 30 - Variation 5 ▸ `LReal with flags (Double-precision with flag)` Object Group 30 - Variation 6 Default: `32 Bit with flags`
Event retention	Behavior when forwarding value changes. Selection from option field. ▸ `Retain all events` Each time a value, status or time stamp of a variable is changed, a new event is created and saved in the event buffer. The master gets all changes. ▸ `Retain most recent event` The event buffer buffer only contains one event for the last change of value, status or time stamp. The master only gets the last change and, under certain circumstances, cannot distinguish whether or how often a value has changed in the meantime. Default: `Retain most recent event`
Max. events	Maximum number of events that are saved in the event

	buffer for this Object Group for all defined variables. If `Retain all events` is configured, this number should be at least as large as the number of configured variables. Default: `100` Note: Only active if `Retain all events` is active.

Default event variation	This option determines the default event variation for this Object Group for newly-added variables. A different event variation can be individually configured for each variable.
	Note: Select a event variation that is supported by the master. Not every master supports all event variations.
	Select from drop-down list:
	`32 Bit without time`
	Object Group 32 - Variation 1
	`16 Bit without time`
	Object Group 32 - Variation 2
	`32 Bit with time`
	Object Group 32 - Variation 3
	`16 Bit with time`
	Object Group 32 - Variation 4
	`Real without time (Single-precision without time)`
	Object Group 32 - Variation 5
	`LReal without time (Double-precision without time)`
	Object Group 32 - Variation 6
	`Real with time (Single-precision with time)`
	Object Group 32 - Variation 7
	`LReal with time (Double-precision with time)`
	Object Group 32 - Variation 8
	Default: `32 Bit with time`

Event class	This option determines the default event class for this Object Group for newly-added variables. A different event class can be individually configured for each variable. Select from drop-down list: `none` Variables are not assigned to an event class and do not generate events. No static objects are included

	for the variables in a response to a class 0 poll from the master. A master can only read the current value of the variables by means of a read request for the respective static object group. A master can assign the variables by means of an assign class request, not another event class. ▸ zero Variables are assigned to the event class 0, but do not generate any events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the variables by means of a read request for the respective static object group. A master can assign the variables by means of an assign class request, the event class 1, 2 or 3, as a result of which the variables also generate events from this point in time. ▸ one Variables are assigned to the event class 1, and generate events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the current value of the variables by means of a read request for the respective static object group. In a response to a class 1 poll from the master , any event objects that may be present are included for the events of the variables. A master can assign the variables by means of an assign class request, the event class 0, 2 or 3. If the variable is assigned by the master of event class 0, no more events are generated. ▸ two Variables are assigned to the event class 2, and generate events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the current value of the variables by means of a read request for the respective static object group. In a response to a class 2 poll from the master , any event objects that may be present are included for the events of the variables. A master can assign the variables by means of an assign class request, the event class 0, 1 or 3. If the variable is assigned by the master of event class 0, no more events are generated. ▸ three Variables are assigned to the event class 3, and generate events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the current value of the variables by means of a read request for the

	respective static object group. In a response to a class 3 poll from the master , any event objects that may be present are included for the events of the variables. A master can assign the variables by means of an assign class request, the event class 0, 1 or 2. If the variable is assigned by the master of event class 0, no more events are generated. Default: `three`

Variables taken from the zenon project are listed with their current configuration in this list.

- The list can be sorted
  - Click for the sorting on the column heading.
  - The sorting sequence is visualized with an arrow symbol next to the column heading:
    `Arrow upwards`: ascending sorting
    `Arrow downwards`: descending sorting
  - Another click on the column heading reverses the sorting order.
- The list can be filtered
  To filter the list:
  - Enter the desired filter term in the input field below the heading. The default description of an empty field is `filter text` (shown in gray font).

Parameter	Description
Name	Variable name. The project name is placed in front of the name: ProjectName#VariableName
Identification	Identification of the variable.
Point Index	Current configuration of the DNP3 point index of the variable in this Object Group.
Static Variation	Current configuration of the static variation for the point of the variable in this Object Group.
Event Variation	Current configuration of the event variation for the point of the variable in this Object Group. Note: This parameter is not available in the **Octet String** node.
Event Class	Current configuration of the event class for the point of the variable in this Object Group.
Control Model	Current configuration of the **Control Model**. Note: This parameter is only available in the **Binary Output** node.
Command Routing	Current behavior of the Binary Output point with select and operate requests from the master. Note: This parameter is only available in the **Binary Output** node.

Supported data types: BOOL, Byte, DINT, DWORD, Date, Date_and_Time, INT, LINT, LREAL, LWORD, REAL, SINT, TIME, TOD, UDINT, UINT, ULINT, USINT, WORD

NAVIGATION BAR

Parameter	Description
Add...	Opens the variable selection dialog (à la page 108) to apply the zenon variables in the list. Note: Not active in the **Frozen Counter** node.
Remove	Removes selected variables from the list. A confirmation dialog is shown before a variable is deleted. Only active if one or more variables in the list are selected. Note: Not active in the **Frozen Counter** node.
Replace variable...	Opens the variable selection dialog (à la page 108) to replace a variable selected in the list with another zenon variable. Only active if precisely one variable in the list is selected.

ANALOG INPUT

You configure the parameters for selected variable(s) in the list in this area.

Parameter	Description
Point index	This option determines the DNP3 point index in the Object Group for the selected variable. If variables are added, the point index is automatically set for the newly-added variables and numbered consecutively starting with the highest point index already present. A point index can only be issued once within an Object Group. The entry is validated. If a point index has already been issued, this is shown with a warning dialog. Note: It is recommended that you issue the point index starting with 0 without gaps. The outstation supports the maximum point index. If a higher point index is used, it should be ensured that the master also supports qualifier codes with 2 octets or 4 octets accordingly.
Identification	The identification is primarily taken from the variable from zenon Runtime. The identification can also be amended locally with this option.
Static variation	This option determines the static variation for the selected variable for the point in this Object Group. With the `default` value, the general static variation defined for this Object Group is used. Note: Select a static variation that is supported by the master. Not every master supports all static variations. Select from drop-down list: ▸ `32 Bit with flags` Object Group 30 - Variation 1 ▸ `16 Bit with flags` Object Group 30 - Variation 2 ▸ `32 Bit without flags` Object Group 30 - Variation 3 ▸ `16 Bit without flags` Object Group 30 - Variation 4 ▸ `Real with flags (Single-precision with flag)` Object Group 30 - Variation 5 ▸ `LReal with flags (Double-precision with flag)` Object Group 30 - Variation 6

	Default: `default`

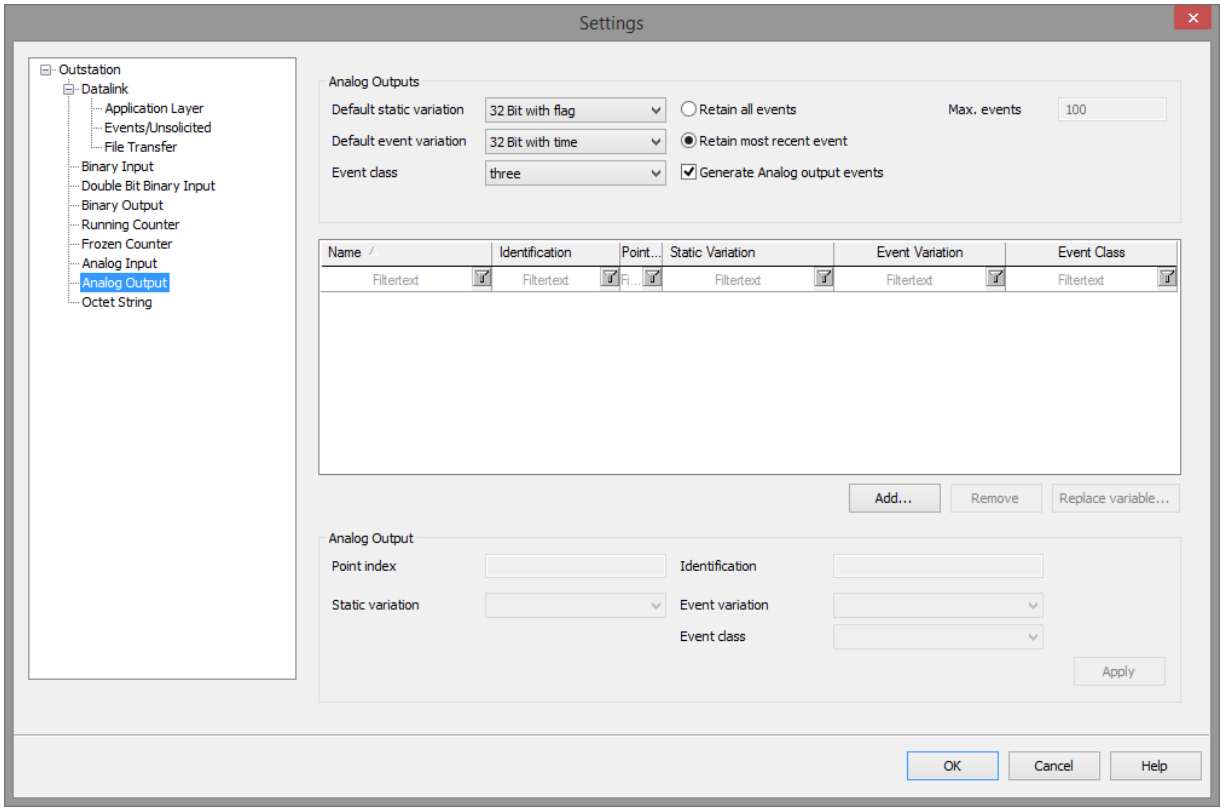
Event variation	This option determines the event variation for the selected variable for the point in this Object Group. With the `default` value, the general event variation defined for this Object Group is used.
	Note: Select a event variation that is supported by the master. Not every master supports all event variations.
	Select from drop-down list:
	`32 Bit without time`
	Object Group 32 - Variation 1
	`16 Bit without time`
	Object Group 32 - Variation 2
	`32 Bit with time`
	Object Group 32 - Variation 3
	`16 Bit with time`
	Object Group 32 - Variation 4
	`Real without time (Single-precision without time)`
	Object Group 32 - Variation 5
	`LReal without time (Double-precision without time)`
	Object Group 32 - Variation 6
	`Real with time (Single-precision with time)`
	Object Group 32 - Variation 7
	`LReal with time (Double-precision with time)`
	Object Group 32 - Variation 8
	Default: `default`

Event class	This option determines the event class for the selected variable for the point in this Object Group. With the `default` value, the general event class defined for this Object Group is used. Selection: see description for **Default event class** option. Default: `default`
Apply	Applies the configuration for the selected variable(s).

FERMER BOÎTE DE DIALOGUE

Options	Description
OK	Applique les paramètres et ferme la boîte de dialogue.
Annuler	Annule toutes les modifications et ferme la boîte de dialogue.
Aide	Ouvre l'aide en ligne.

7.10 Analog Output



You configure the **DNP3 Analog Output points** in this node.

Parameter	Description
Default static variation	This option determines the default static variation for this Object Group for newly-added variables. A different static variation can be individually configured for each variable. Note: Select a static variation that is supported by the master. Not every master supports all static variations. Select from drop-down list: ▸ `32 Bit with flag` Object Group 40 - Variation 1 ▸ `16 Bit with flag` Object Group 40 - Variation 2 ▸ `Real with flag (Single-precision with flag)` Object Group 40 - Variation 3 ▸ `LReal with flag (Double-precision with flag)` Object Group 40 - Variation 4 Default: `32 Bit with flag`
Event retention	Behavior when forwarding value changes. Selection from option field. ▸ `Retain all events` Each time a value, status or time stamp of a variable is changed, a new event is created and saved in the event buffer. The master gets all changes. ▸ `Retain most recent event` The event buffer buffer only contains one event for the last change of value, status or time stamp. The master only gets the last change and, under certain circumstances, cannot distinguish whether or how often a value has changed in the meantime. Default: `Retain most recent event`
Max events	Maximum number of events that are saved in the event buffer for this Object Group for all defined variables. If `Retain all events` is configured, this number should be at least as large as the number of configured variables. Default: `100`

	Note: Only active if `Retain all events` is active.
Default event variations	This option determines the default event variation for this Object Group for newly-added variables. A different event variation can be individually configured for each variable. Note: Select a event variation that is supported by the master. Not every master supports all event variations. Select from drop-down list: ▸ `32 Bit without time` Object Group 42 - Variation 1 ▸ `16 Bit without time` Object Group 42 - Variation 2 ▸ `32 Bit with time` Object Group 42 - Variation 3 ▸ `16 Bit with time` Object Group 42 - Variation 4 ▸ `Real without time (Single-precision without time)` Object Group 42 - Variation 5 ▸ `LReal without time (Double-precision without time)` Object Group 42 - Variation 6 ▸ `Real with time (Double-precision with time)` Object Group 42 - Variation 7 ▸ `LReal with time (Double-precision with time)` Object Group 42 - Variation 8 Default: `32 Bit with time`
Event class	This option determines the default event class for this Object Group for newly-added variables. A different event class can be individually configured for each variable. Select from drop-down list: ▸ `none` Variables are not assigned to an event class and do not generate events. No static objects are included for the variables in a response to a class 0 poll from the master. A master can only read the current value of the variables by means of a read request for the respective static object group. A master can assign

	the variables by means of an assign class request, not another event class. ▸ zero Variables are assigned to the event class 0, but do not generate any events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the variables by means of a read request for the respective static object group. A master can assign the variables by means of an assign class request, the event class 1, 2 or 3, as a result of which the variables also generate events from this point in time. ▸ one Variables are assigned to the event class 1, and generate events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the current value of the variables by means of a read request for the respective static object group. In a response to a class 1 poll from the master , any event objects that may be present are included for the events of the variables. A master can assign the variables by means of an assign class request, the event class 0, 2 or 3. If the variable is assigned by the master of event class 0, no more events are generated. ▸ two Variables are assigned to the event class 2, and generate events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the current value of the variables by means of a read request for the respective static object group. In a response to a class 2 poll from the master , any event objects that may be present are included for the events of the variables. A master can assign the variables by means of an assign class request, the event class 0, 1 or 3. If the variable is assigned by the master of event class 0, no more events are generated. ▸ three Variables are assigned to the event class 3, and generate events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the current value of the variables by means of a read request for the respective static object group. In a response to a class 3 poll from the master , any event objects that may be present are included for the events of the variables. A master can assign the variables by

	means of an assign class request, the event class 0, 1 or 2. If the variable is assigned by the master of event class 0, no more events are generated. Default: `three`
Generate Analog output events	This option determines whether value changes for variables that are configured for Binary Outputs , events are also to be generated. Not all masters support events for Binary Outputs. Default: `active`

Variables taken from the zenon project are listed with their current configuration in this list.

- The list can be sorted
  - Click for the sorting on the column heading.
  - The sorting sequence is visualized with an arrow symbol next to the column heading:
    `Arrow upwards`: ascending sorting
    `Arrow downwards`: descending sorting
  - Another click on the column heading reverses the sorting order.
- The list can be filtered
  To filter the list:
  - Enter the desired filter term in the input field below the heading. The default description of an empty field is `filter text` (shown in gray font).

Parameter	Description
Name	Variable name. The project name is placed in front of the name: ProjectName#VariableName
Identification	Identification of the variable.
Point Index	Current configuration of the DNP3 point index of the variable in this Object Group.
Static Variation	Current configuration of the static variation for the point of the variable in this Object Group.
Event Variation	Current configuration of the event variation for the point of the variable in this Object Group. Note: This parameter is not available in the **Octet String** node.
Event Class	Current configuration of the event class for the point of the variable in this Object Group.
Control Model	Current configuration of the **Control Model**. Note: This parameter is only available in the **Binary Output** node.
Command Routing	Current behavior of the Binary Output point with select and operate requests from the master. Note: This parameter is only available in the **Binary Output** node.

Supported data types: BOOL, Byte, DINT, DWORD, Date, Date_and_Time, INT, LINT, LREAL, LWORD, REAL, SINT, TIME, TOD, UDINT, UINT, ULINT, USINT, WORD

NAVIGATION BAR

Parameter	Description
Add...	Opens the variable selection dialog (à la page 108) to apply the zenon variables in the list. Note: Not active in the **Frozen Counter** node.
Remove	Removes selected variables from the list. A confirmation dialog is shown before a variable is deleted. Only active if one or more variables in the list are selected. Note: Not active in the **Frozen Counter** node.
Replace variable...	Opens the variable selection dialog (à la page 108) to replace a variable selected in the list with another zenon variable. Only active if precisely one variable in the list is selected.

ANALOG OUTPUT

You configure the parameters for selected variable(s) in the list in this area.

Parameter	Description
Pointer Index	This option determines the DNP3 point index in the Object Group for the selected variable. If variables are added, the point index is automatically set for the newly-added variables and numbered consecutively starting with the highest point index already present. A point index can only be issued once within an Object Group. The entry is validated. If a point index has already been issued, this is shown with a warning dialog. Note: It is recommended that you issue the point index starting with 0 without gaps. The outstation supports the maximum point index. If a higher point index is used, it should be ensured that the master also supports qualifier codes with 2 octets or 4 octets accordingly.
Identification	The identification is primarily taken from the variable from zenon Runtime. The identification can also be amended locally with this option.

Static variation	This option determines the static variation for the selected variable for the point in this Object Group. With the `default` value, the general static variation defined for this Object Group is used.
	
	Note: Select a static variation that is supported by the master. Not every master supports all static variations.
	
	Select from drop-down list:
	▸ `32 Bit with flag`
	Object Group 40 - Variation 1
	▸ `16 Bit with flag`
	Object Group 40 - Variation 2
	▸ `Real with flag`
	`(Single-precision with flag)`
	Object Group 40 - Variation 3
	▸ `LReal with flag`
	`(Double-precision with flag)`
	Object Group 40 - Variation 4
	
	Default: `default`

Event variation	This option determines the event variation for the selected variable for the point in this Object Group. With the `default` value, the general event

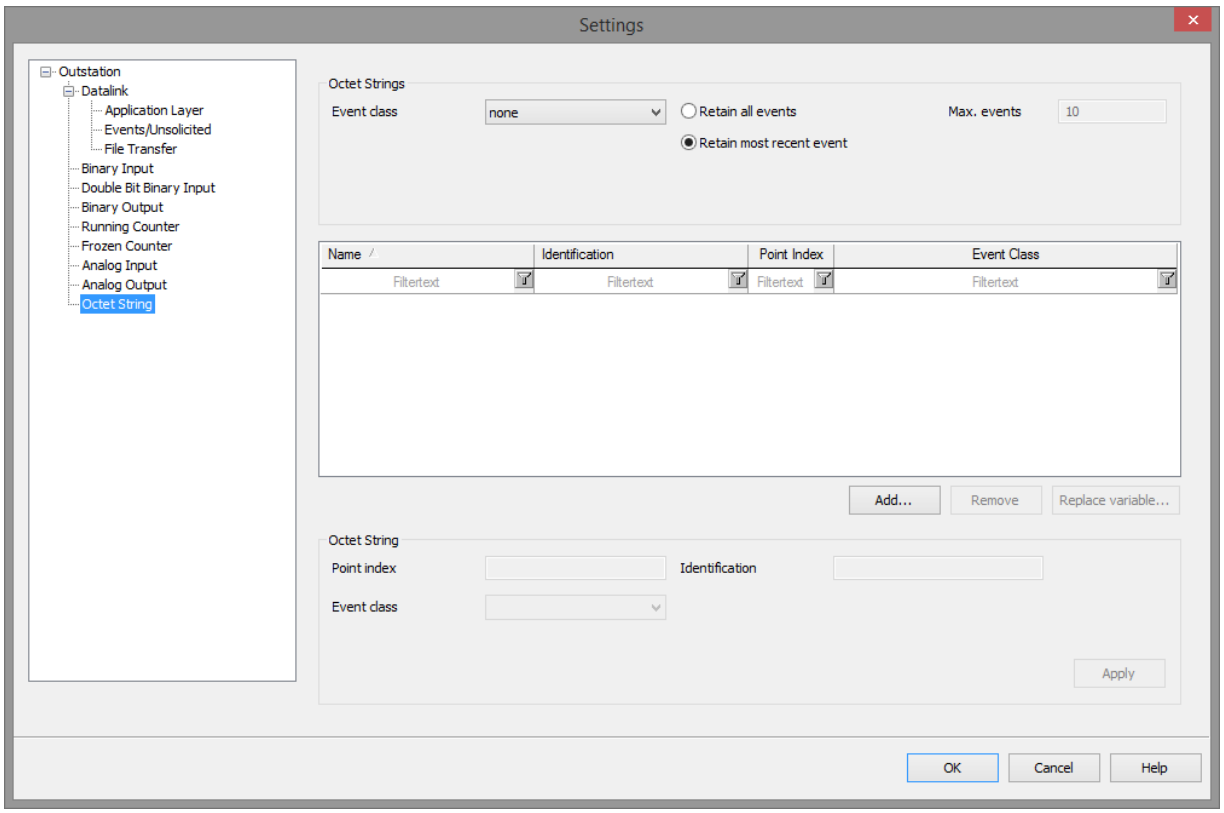
	variation defined for this Object Group is used. Note: Select a event variation that is supported by the master. Not every master supports all event variations. Select from drop-down list: ▸ `32 Bit without time` Object Group 42 - Variation 1 ▸ `16 Bit without time` Object Group 42 - Variation 2 ▸ `32 Bit with time` Object Group 42 - Variation 3 ▸ `16 Bit with time` Object Group 42 - Variation 4 ▸ `Real without time (Single-precision without time)` Object Group 42 - Variation 5 ▸ `LReal without time (Double-precision without time)` Object Group 42 - Variation 6 ▸ `Real with time (Double-precision with time)` Object Group 42 - Variation 7 ▸ `LReal with time (Double-precision with time)` Object Group 42 - Variation 8 Default: `default`
Event class	This option determines the event class for the selected variable for the point in this Object Group. With the `default` value, the general event class defined for this Object Group is used. Selection: see description for **Default event class** option. Default: `default`
Apply	Applies the configuration for the selected variable(s).

FERMER BOÎTE DE DIALOGUE

Options	Description
OK	Applique les paramètres et ferme la boîte de dialogue.
Annuler	Annule toutes les modifications et ferme la boîte de dialogue.

Aide	Ouvre l'aide en ligne.

7.11 Octet String



You configure the **DNP3 Octet string points** in this node.

Parameter	Description
Event Class	This option determines the default event class for this Object Group for newly-added variables. A different event class can be individually configured for each variable. Select from drop-down list: ▸ none Variables are not assigned to an event class and do not generate events. No static objects are included for the variables in a response to a class 0 poll from the master. A master can only read the current value of the variables by means of a read request for the respective static object group. A master can assign the variables by means of an assign class request, not another event class. ▸ zero Variables are assigned to the event class 0, but do not generate any events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the variables by means of a read request for the respective static object group. A master can assign the variables by means of an assign class request, the event class 1, 2 or 3, as a result of which the variables also generate events from this point in time. ▸ one Variables are assigned to the event class 1, and generate events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the current value of the variables by means of a read request for the respective static object group. In a response to a class 1 poll from the master , any event objects that may be present are included for the events of the variables. A master can assign the variables by means of an assign class request, the event class 0, 2 or 3. If the variable is assigned by the master of event class 0, no more events are generated. ▸ two Variables are assigned to the event class 2, and generate events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the current value of the variables by means of a read request for the respective static object group. In a response to a class 2 poll from the master , any event objects that

	may be present are included for the events of the variables. A master can assign the variables by means of an assign class request, the event class 0, 1 or 3. If the variable is assigned by the master of event class 0, no more events are generated. ▸ `three` Variables are assigned to the event class 3, and generate events. In a response to a class 0 poll from the master, the static objects are included for the variables. A master can also read the current value of the variables by means of a read request for the respective static object group. In a response to a class 3 poll from the master , any event objects that may be present are included for the events of the variables. A master can assign the variables by means of an assign class request, the event class 0, 1 or 2. If the variable is assigned by the master of event class 0, no more events are generated. Default: `none`
Event retention	Behavior when forwarding value changes. Selection from option field. ▸ `Retain all events` Each time a value, status or time stamp of a variable is changed, a new event is created and saved in the event buffer. The master gets all changes. ▸ `Retain most recent event` The event buffer buffer only contains one event for the last change of value, status or time stamp. The master only gets the last change and, under certain circumstances, cannot distinguish whether or how often a value has changed in the meantime. Default: `Retain most recent event`
Max. events	Maximum number of events that are saved in the event buffer for this Object Group for all defined variables. If `Retain all events` is configured, this number should be at least as large as the number of configured variables. Default: `100` Note: Only active if `Retain all events` is active.

Variables taken from the zenon project are listed with their current configuration in this list.

- The list can be sorted

- Click for the sorting on the column heading.
- The sorting sequence is visualized with an arrow symbol next to the column heading:
  `Arrow upwards`: ascending sorting
  `Arrow downwards`: descending sorting
- Another click on the column heading reverses the sorting order.

▶ The list can be filtered
To filter the list:

- Enter the desired filter term in the input field below the heading. The default description of an empty field is `filter text` (shown in gray font).

Parameter	Description
`Name`	Variable name. The project name is placed in front of the name: `ProjectName#VariableName`
`Identification`	Identification of the variable.
`Point Index`	Current configuration of the DNP3 point index of the variable in this Object Group.
`Static Variation`	Current configuration of the static variation for the point of the variable in this Object Group.
`Event Variation`	Current configuration of the event variation for the point of the variable in this Object Group. Note: This parameter is not available in the **Octet String** node.
`Event Class`	Current configuration of the event class for the point of the variable in this Object Group.
`Control Model`	Current configuration of the **Control Model**. Note: This parameter is only available in the **Binary Output** node.
`Command Routing`	Current behavior of the Binary Output point with select and operate requests from the master. Note: This parameter is only available in the **Binary Output** node.

Supported data types: `STRING, WSTRING`

NAVIGATION BAR

Parameter	Description
Add...	Opens the variable selection dialog (à la page 108) to apply the zenon variables in the list. Note: Not active in the **Frozen Counter** node.
Remove	Removes selected variables from the list. A confirmation dialog is shown before a variable is deleted. Only active if one or more variables in the list are selected. Note: Not active in the **Frozen Counter** node.
Replace variable...	Opens the variable selection dialog (à la page 108) to replace a variable selected in the list with another zenon variable. Only active if precisely one variable in the list is selected.

OCTET STRING

You configure the parameters for selected variable(s) in the list in this area.

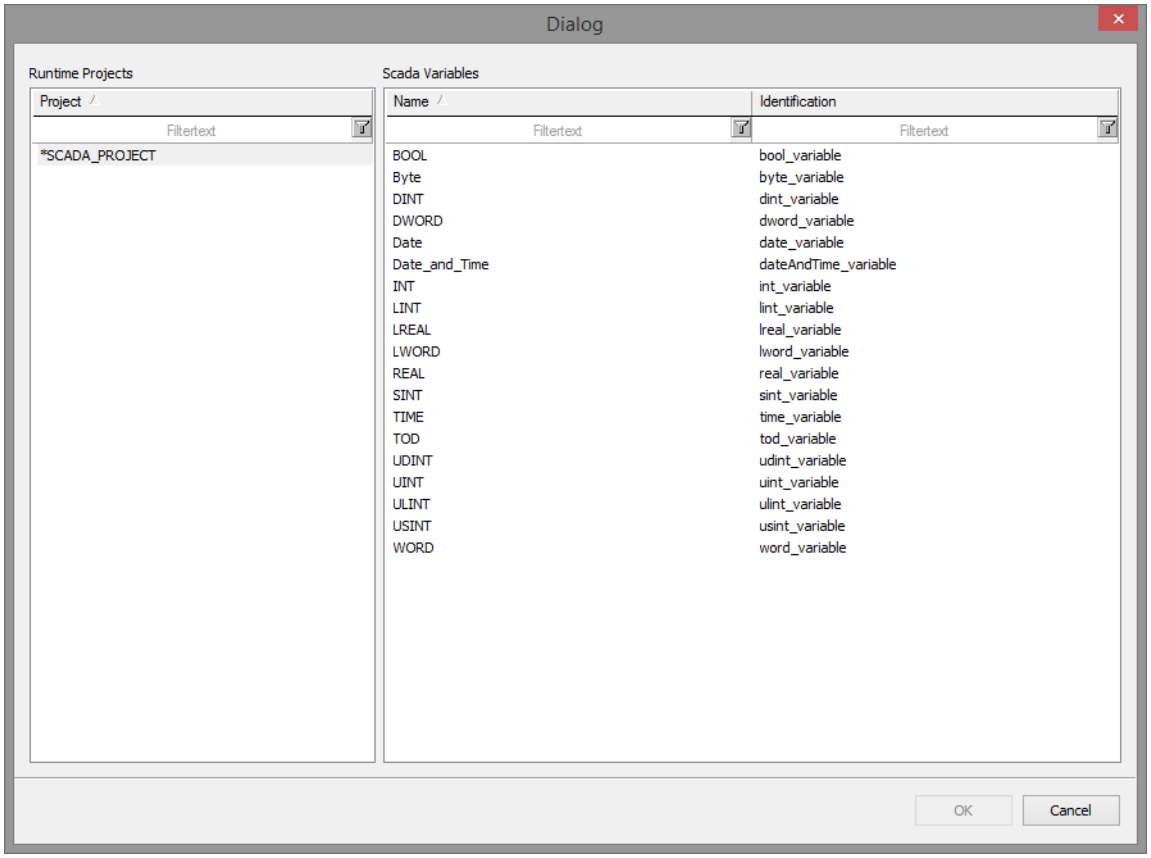
Parameter	Description
Point index	This option determines the DNP3 point index in the Object Group for the selected variable. If variables are added, the point index is automatically set for the newly-added variables and numbered consecutively starting with the highest point index already present. A point index can only be issued once within an Object Group. The entry is validated. If a point index has already been issued, this is shown with a warning dialog. Note: It is recommended that you issue the point index starting with 0 without gaps. The outstation supports the maximum point index. If a higher point index is used, it should be ensured that the master also supports qualifier codes with 2 octets or 4 octets accordingly.
Identification	The identification is primarily taken from the variable from zenon Runtime. The identification can also be amended locally with this option.
Event class	This option determines the event class for the selected variable for the point in this Object Group. With the `default` value, the general event class defined for this Object Group is used. Selection: see description for **Default event class** option. Default: `default`
Apply	Applies the configuration for the selected variable(s).

FERMER BOÎTE DE DIALOGUE

Options	Description
OK	Applique les paramètres et ferme la boîte de dialogue.
Annuler	Annule toutes les modifications et ferme la boîte de dialogue.
Aide	Ouvre l'aide en ligne.

7.12 Variable selection dialog

In this dialog, you apply variables from an existing zenon project configuration in the **DNP3_SG** configuration. Multiple selection is possible.



- The list can be sorted
  - Click for the sorting on the column heading.
  - The sorting sequence is visualized with an arrow symbol next to the column heading:
    `Arrow upwards`: ascending sorting
    `Arrow downwards`: descending sorting
  - Another click on the column heading reverses the sorting order.
- The list can be filtered
  To filter the list:
  - Enter the desired filter term in the input field below the heading. The default description of an empty field is `filter text` (shown in gray font).

RUNTIME PROJECTS

Parameter	Description
[Project name]	Name of the zenon projects loaded in Runtime. The active project is marked with a * (star character)

SCADA VARIABLES

Parameter	Description
[Variable list]	List of the configured variables in the currently-loaded zenon projects. The list offers, for selection, the data types that are supported for the respective nodes. ▸ Name: Name of the variable. This corresponds to the **Nom** variable property in the current zenon configuration. ▸ Identification: Identification of the variable. This corresponds to the **Identification** variable property in the current zenon configuration.

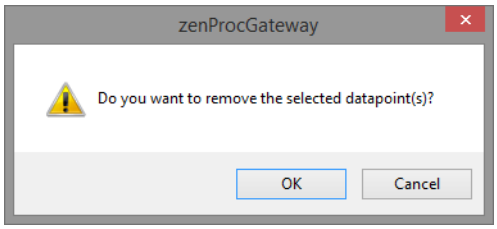
CLOSE DIALOG

OK	Applique les paramètres et ferme la boîte de dialogue. The selected variables are applied in the **DNP3_SG** configuration of the current node.
Cancel	Annule toutes les modifications et ferme la boîte de dialogue. No variables are applied in the DNP3_SG configuration.

7.13 Warning dialog

REMOVE VARIABLE (DATA POINT)

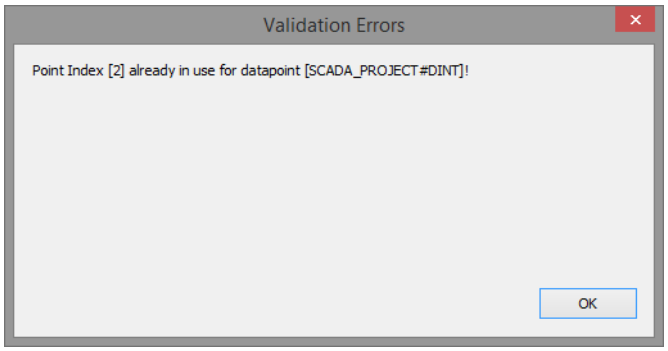
The following warning dialog is shown before a variable is deleted:



Parameter	Description
OK	Deletes the selected variable(s) from the list.
Cancel	Closes the dialog without deleting the selected variable(s).

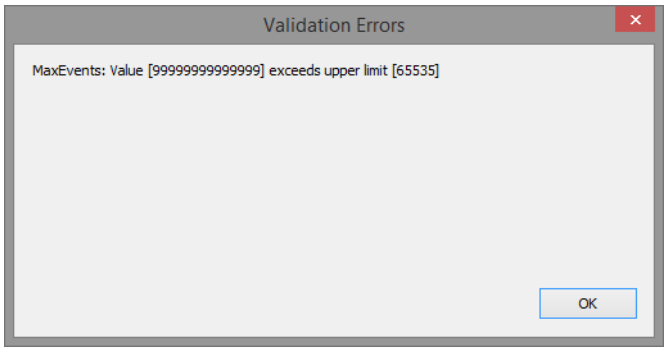
DATA POINT NOT UNIQUE

If a point index that is already present in the current configuration has been configured for a variable, this is shown with a warning dialog.



INVALID INPUT

If an invalid entry has been configured for an option, this is shown in a warning dialog.



7.14 Status mapping zenon - DNP3

The point mapping of the zenon status to the DNP status is carried out if an event is logged in the event buffer.

The mapping between zenon and DNP 3 is carried out as follows:

	Status DNP3							
Status zenon	**ONLINE** (Note 2)	**COMM_LOST** (Note 2)	**LOCAL_FORCED** (Note 2)	**OVER_RANGE** (Note 3)	**REFERENCEERR** (Note 3)	**DISCONTINUITY** (Note 3)	**ROLLOVER** (Note 3)	**CHATTERFILTER** (Note 3)
SPONT (Note 1)	1	0	0					
GA (Note 1)	1	0	0					
INVALID (Note 1)	0	1	0					
ALTVAL (Note 1)	0	0	1					
OFF (Note 1)	0	0	0					

N_UPDATE (Note 1)	0	0	0					
NT_870 (Note 1)	0	0	0					
SB_870 (Note 1)	1	0	1					
OV_870				1				
OR_870				1				

NOTES

- Note 1:

  At least one status must be set in zenon. A corresponding LOG entry is created if this is not the case. No additional flags are used.

- Note 2:

  Forced base flags, these flags are always set as a bit field.

- Note 3:

  Flags are only set for ONLINE or LOCAL_FORCED.

7.15 Select before Operate

Analog Outputs and Binary Outputs have SBO (Select before Operate) functionality. For this, the following applies:

- The DNP3 master sends a Select command to the outstation.
- When the Binary Output object type is requested, the following requirements must be met in order for the request to be answered in the positive:
  - The variable exists and is signed in or can be signed in.
  - The data type of the variable is IEC_BOOL or IEC_USINT
  - The value of the command is valid:
    NUL, PulseOn, PulseOff, LatchOn, LatchOff, PulseOn + Close, PulseOn + Trip
- Requests of the Analog Output object type are always answered in the positive
- The configured Select timeout is activated automatically.
  If, within this timeout, there is no Operate command received for the respective variable, the processing of the variable is ended automatically.
  Note: If, instead of an Operate Request, a different Request is received for the respective variable, the sequence number is incorrect or the Operate Request does not correspond to the Select Request. Processing is ended automatically.
- An Operate Request without a previous Select Request is acknowledged with an error.

> **Conseil**
>
> *You can find further general information on **Select before Operate** in the Energy Edition manual, **Command Processing** in the Execution of a command chapter.*

7.16 Command processing - Command Routing

Basic requirements for the positive execution of the Select or Operate command are:

- The variable type is Binary Output
- The variable exists, is signed in or can be signed in, and is valid (no I-Bit).
- The data type of the variable is IEC_REAL, IEC_BOOL or IEC_USINT.
- The control code is valid and can be converted to a Boolean value-
  The following values are valid (the corresponding Boolean value is stated in brackets):
  - NUL (false)

- PulseOn (true)
- PulseOff (false)
- LatchOn (true)
- LatchOff (false)
- PulseOn + Close (true)
- PulseOn + Trip (false)

PROCEDURE

In general, the following applies: Command routing is only activated for exported Binary Output variables for which the **Command Routing** option has been activated.

- If a Select command is received for a Binary Output for which the **Command Routing** option has been deactivated, a positive Response is returned to the master, provided the basic requirements have been met.
- If, for Binary Output, the **Command Routing** option has been set, the COT_act command with the control code is sent to the command variable (configured with **Binary Output**), in the Runtime.
  Note: COT_act is coded in the status value with the help of the COT_act, S_SE_BIT and S_SELECT bits.
- There is then a wait for the execution of the Select command by the auto-remote command action, before the Response is sent to the master.
  Only once there is a positive response to the Select command sent to the master does the Select Timeout Timer start.
- The processing of the following Operate command is carried out in the same way as for the Select command.In addition, for an Operate command, there is a check to see whether a Select– command has been received before the command is forwarded to the command processing in Runtime.
- If, for the Operate command, the **Command Routing** option is not active or no Select command has been received beforehand, an Operate Request is answered directly.
- After the Operate command has been processed successfully by the command processing in Runtime, a positive Operate Response is sent to the master. In the event of an error, an Operate Response is sent to the master with an error.

STATUS OF COMMAND ROUTING

The following status codes are sent to the master from the outstation:

- **Status Code 0 - SUCCESS**
  The command processing was successful.

- **Status Code 1 - TIMEOUT**
  A timeout has occurred during command processing.
  The command processing could not be successfully completed within 2 minutes.
- **Status Code 2 - NO_SELECT**
  The `Select` command has been sent by the command processing with an invalid status.
  Invalid statuses are:
  - `S_PN_BIT` (negative confirmation) received
  - The `S_SE_BIT` is no longer set
  - None of the `COT` bits are set
- **Status Code 6 - HARDWARE_ERROR**
  This error occurs under the following circumstances:
  - The variable is not signed in and cannot be signed in.
  - The variable has the **INVALID** bit
  - The value cannot be written or the command cannot be sent.
- **Status Code 10 - AUTOMATION_INHIBIT**
  The `Operate` command has been sent by the command processing with an invalid status.
  Invalid statuses are:
  - `S_PN_BIT` (negative confirmation) received
  - The `S_SE_BIT` is no longer set
  - None of the `COT` bits are set
- **Status Code 12 - OUT_OF_RANGE**
  The control code contained in the `Request` is invalid.

7.17 LATCH_ON and LATCH_OFF

If **Command Routing** is inactive the Command Control Code from the controller is is written to USINT variables as an 8-bit value via the **Process Gateway**.

Control Code		Control Mode	Variable linked	command routing	Acti
0x01	**Pulse On**	Activation	**BOOL**	disabled	set
0x01	**Pulse On**	Activation	**USINT**	disabled	set
0x01	**Pulse On**	Complementary latch	**n.a.**	disabled	ret
0x01	**Pulse On**	Complementary two output	**n.a.**	disabled	ret
0x01	**Pulse On**	Activation	**BOOL**	enabled	set
0x01	**Pulse On**	Activation	**USINT**	enabled	set
0x01	**Pulse On**	Complementary latch	**n.a.**	enabled	ret
0x01	**Pulse On**	Complementary two output	**n.a.**	enabled	ret
0x03	**Latch On**	Activation	**BOOL**	disabled	set
0x03	**Latch On**	Activation	**USINT**	disabled	set
0x03	**Latch On**	Complementary latch	**BOOL**	disabled	set
0x03	**Latch On**	Complementary latch	**USINT**	disabled	set
0x03	**Latch On**	Complementary two output	**BOOL**	disabled	ret
0x03	**Latch On**	Complementary two output	**USINT**	disabled	set
0x03	**Latch On**	Activation	**BOOL**	enabled	set
0x03	**Latch On**	Activation	**USINT**	enabled	set
0x03	**Latch On**	Complementary latch	**BOOL**	enabled	set
0x03	**Latch On**	Complementary latch	**USINT**	enabled	set
0x03	**Latch On**	Complementary two output	**BOOL**	enabled	set
0x03	**Latch On**	Complementary two output	**USINT**	enabled	set
0x04	**Latch Off**	Activation	**BOOL**	disabled	set
0x04	**Latch Off**	Activation	**USINT**	disabled	set
0x04	**Latch Off**	Complementary latch	**BOOL**	disabled	set
0x04	**Latch Off**	Complementary latch	**USINT**	disabled	set
0x04	**Latch Off**	Complementary two output	**BOOL**	disabled	ret
0x04	**Latch Off**	Complementary two output	**USINT**	disabled	set
0x04	**Latch Off**	Activation	**BOOL**	enabled	set
0x04	**Latch Off**	Activation	**USINT**	enabled	set
0x04	**Latch Off**	Complementary latch	**BOOL**	enabled	set
0x04	**Latch Off**	Complementary latch	**USINT**	enabled	set
0x04	**Latch Off**	Complementary two output	**BOOL**	enabled	set
0x04	**Latch Off**	Complementary two output	**USINT**	enabled	set

0x41	**Close**	Activation	**BOOL**	disabled	set
0x41	**Close**	Activation	**USINT**	disabled	set
0x41	**Close**	Complementary latch	**BOOL**	disabled	set
0x41	**Close**	Complementary latch	**USINT**	disabled	set
0x41	**Close**	Complementary two output	**BOOL**	disabled	ret
0x41	**Close**	Complementary two output	**USINT**	disabled	set
0x41	**Close**	Activation	**BOOL**	enabled	set
0x41	**Close**	Activation	**USINT**	enabled	set
0x41	**Close**	Complementary latch	**BOOL**	enabled	set
0x41	**Close**	Complementary latch	**USINT**	enabled	set
0x41	**Close**	Complementary two output	**BOOL**	enabled	set
0x41	**Close**	Complementary two output	**USINT**	enabled	set
0x81	**Trip**	Activation	**BOOL**	disabled	set
0x81	**Trip**	Activation	**USINT**	disabled	set
0x81	**Trip**	Complementary latch	**BOOL**	disabled	set
0x81	**Trip**	Complementary latch	**USINT**	disabled	set
0x81	**Trip**	Complementary two output	**BOOL**	disabled	ret
0x81	**Trip**	Complementary two output	**USINT**	disabled	set
0x81	**Trip**	Activation	**BOOL**	enabled	set
0x81	**Trip**	Activation	**USINT**	enabled	set
0x81	**Trip**	Complementary latch	**BOOL**	enabled	set
0x81	**Trip**	Complementary latch	**USINT**	enabled	set
0x81	**Trip**	Complementary two output	**BOOL**	enabled	set
0x81	**Trip**	Complementary two output	**USINT**	enabled	set

8. DNP3 Slave - AccessDNP3 obsolète

From zenon version 8.00, the **DNP3 Slave** (**AccessDNP3**) was replaced with the new **Process Gateway** module **DNP3_SG**.

- The outdated **AccessDNP3**module can no longer be selected for new configurations.
- Existing configurations with the previous **AccessDNP3** module continue to be able to run.

The **AccessDNP3** module will no longer be supplied in a future version. The old module will be completely replaced by the **AccessDNP3_SG** module.

Informations

The documentation for the outdated module is no longer supplied. Please contact your support partner at the email address support@copadata.com to request this documentation.

9. ICCP-TASE.2

The **IEC 60870-6 TASE.2 ICCP** protocol can be used with the **AccessICCP.dll** module via the zenon **Process Gateway**. ICCP-TASE.2 supports Conformance Blocks 1 and 2 as client and server. **Process Gateway** acts like a server by default when communication is established.

Communication in the ICCP protocol is spontaneous. An ICCP server reports the value changes to its communication partner, the remote client. However, an ICCP server cannot actively read the value changes from the remote client. In order to guarantee two-way exchange of data, both communication partners should have a client and a server.

POSSIBLE SYSTEM CONFIGURATIONS

- Server only

  Communication with several remote ICCP communication partners possible.
  This is the case if no variables have been configured on the **Client Variables** tab.

- Server and passive client

  The client will not initialize the communication with the remote ICCP server itself.
  Communication with several remote ICCP communication partners is possible.
  This is the case if the **Initalise communication** property is not active.
  Note: **General** tab in the configuration dialog of the **Process Gateway**.

- Server and active client

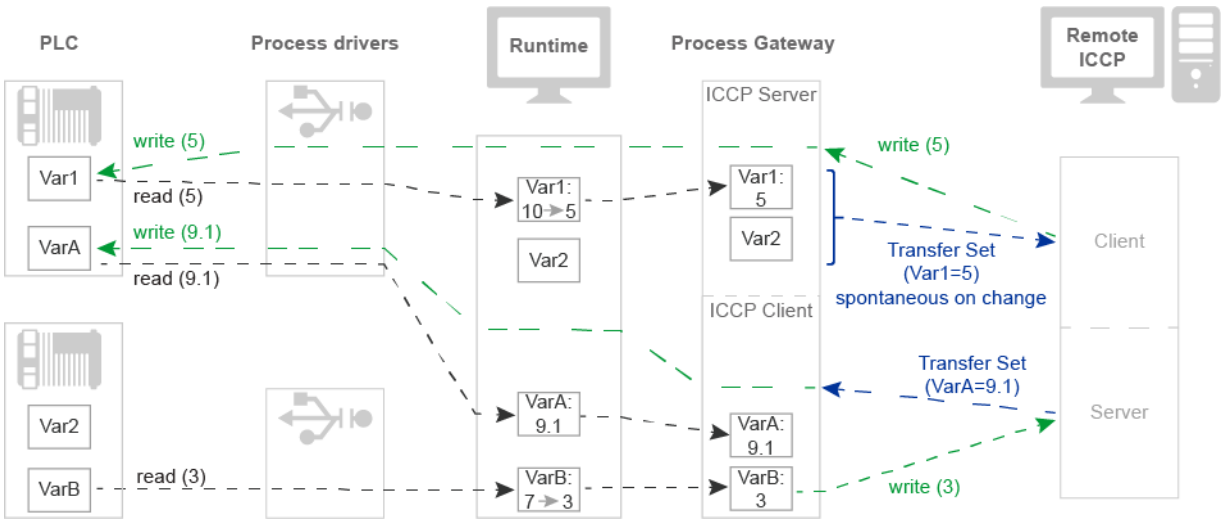
  The client can initialize communication with remote ICCP servers itself.
  Communication is only possible with a remote ICCP communication partner.

ESTABLISHING COMMUNICATION

After the initialization of the communication with the zenon ICCP server or initialization by its own client, the verification of the **bilateral table ID** is carried out. If the verification was successful, the zenon

ICCP client registers the variables to be communicated on the remote server as a DataSet and activates spontaneous communication of the DSTransferSet (`MMS.InformationReport`).

Note: The ICCP client also communicates with the server if, in the configuration of the **Client variables** (à la page 135) tab, no variables are contained in the **Available from ICCP** list. In this case, the client only checks the bilateral table.

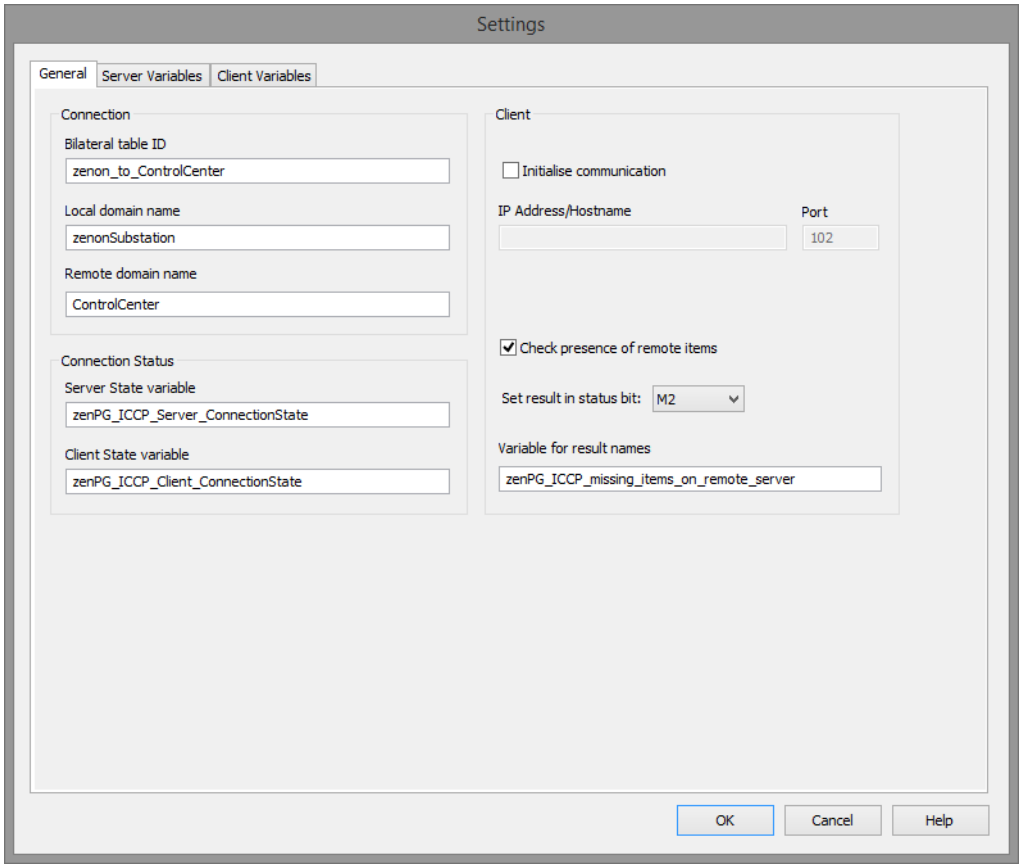


ICCP DATA EXCHANGE

- ICCP Server
  - The ICCP server sends value changes of variables via `DSTransferSets` (`MMS.InformationReport`) spontaneously. If the values of variables in zenon Runtime change, a `DSTransferSet` is sent to the remote client.
  - The ICCP server can also receive write requests from the remote client (`MMS.write request`). These are transferred to zenon Runtime and Runtime forwards this request to the corresponding zenon driver. If the writing is confirmed by the PLC, the zenon variable changes its value.
- ICCP Client
  - In the event that a variable in zenon Runtime sends the ICCP client a corresponding write request (`MMS.write request`) to the remote server.
  - If a `DSTransferSet` (`MMS.InformationReport`) is received by the ICCP client (from the remote server) the received values are sent to zenon Runtime. Runtime forwards these requests to the zenon driver. If the writing is confirmed by the PLC, the zenon variables do not change their value.

9.1 Module configuration

Configuration is carried out using three tabs. It can be confirmed and closed by clicking on OK if all three tabs have been configured.



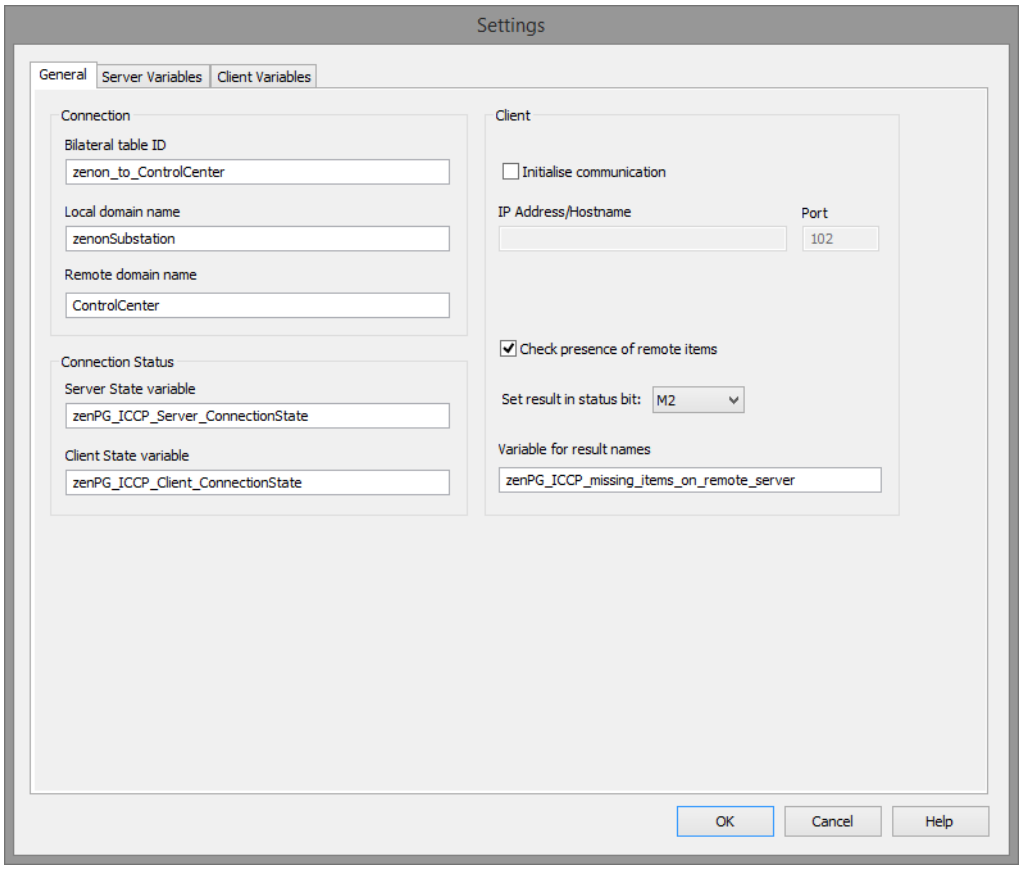
Parameter	Description
General (à la page 124)	General settings.
Server variables (à la page 131)	Configuration of the server variables. Selection of the variables that the ICCP server provides.
Client variables (à la page 135)	Configuration of the client variables. Selection of the variables that are received by the ICCP client.

FERMER BOÎTE DE DIALOGUE

Options	Description
OK	Applique les paramètres et ferme la boîte de dialogue.
Annuler	Annule toutes les modifications et ferme la boîte de dialogue.
Aide	Ouvre l'aide en ligne.

9.1.1 General

Configuration of the connection settings.



CONNECTION

The settings in the `Connection` section are any desired strings that are coordinated on all devices in the system.

The unique bilateral table ID must be the same on all devices that communicate with the **ICCP Process Gateway**. It is queried when a connection is established by both communication partners. The connection is disconnected if they do not correspond.

The domain names provide the references of the ICCP objects used at MMS protocol level. It must be set as the same mirror image for the communication partners.

These comparison rules are prescribed by the ICCP standard.

Parameter	Description
Bilateral Table ID	The bilateral table identification is synced when the communication partner establishes a connection. Access to the data is only permitted for authorized remote clients. The comparison is Domain-specific, the communication partner must, in the domain with the name **Remote Domain Name**, feature the table with the label **Bilateral Table ID**. Note: The ID must be configured as exactly the same on both sides of the communication. Note: Missing or incorrect entries are visualized as an error message in the dialog.
Local Domain Name	Name of the local ICCP domain that is used for objects of the **Process Gateway** . Note: the name must be set inversely for the communication partners - the remote client should use the name in its requests to **ICCP Process Gateway**. Note: Missing or incorrect entries are visualized as an error message in the dialog.
Remote Domain Name	Name of the domains on the remote ICCP server. The **Process Gateway** ICCP client uses Domain-specific requests with this name: ▸ In the comparison with the bilateral table; ▸ For the Data Set and TranferSet on the remote server. Note: a Data Set is created as Domain-specific , its elements - the ICCP data points - are defined as VMD-specific. ▸ In commands via **Device objects**. Note: Missing or incorrect entries are visualized as an error message in the dialog.
Error message	Display of missing configuration data.

CONNECTION STATUS

Connection status variables allow the display of the current connection status of **Process Gateway** to the remote ICCP communication partner in Runtime. The zenon variable names are given in the **Process Gateway**. The variables should be a numeric data type; recommendation: Internal variables.

You can find detailed information on this in the Status variables (à la page 140) chapter.

Parameter	Description
Server state variable	Input field for the entry of a variable name for the display of the current connection status of the server. If this field is empty, no status information can be displayed in zenon Runtime. Default: `empty`
Client state variable	Input field for the entry of a variable name for the display of the current connection status of the client. If this field is empty, no status information can be displayed in zenon Runtime. Default: `empty`

CLIENT

Parameter	Description
Initialise communication	Checkbox for the activation of the automatic establishment of a connection to the remote ICCP server when the Process Gateway is started. ▸ `active:` When Process Gateways is started, the client automatically attempts to establish a connection to the configured remote ICCP server. If the connection attempt is unsuccessful, another attempt to establish a connection is made after 20 seconds. During this waiting time, a connection attempt can be accepted from a remote ICCP communication partner if this corresponds to the configured IP address or host name. ▸ `inactive:` No automatic attempt to establish a connection to the ICCP server when the Process Gateway is started. Default: `Inactive` If this option is activated, the following limitation applies: ▸ The connection is only possible with a remote ICCP communication partner. ▸ Connection attempts from remote clients with an unknown address are not accepted.
IP Address/Hostname	Input field for IP address or host name for the automatic connection to the remote ICCP server. Note: Only active if **Initialise communication** is inactive.

	Note: Missing or incorrect entries are visualized as an error message in the dialog.
Port	Input field for port number that is used for communication to the remote ICCP server. Default: 102 Note: Only active if **Initialise communication** is inactive. Note: Missing or incorrect entries are visualized as an error message in the dialog.

Check presence of remote items	Checkbox for the activation of the checking of the variable names of the current client connection to the variable name of the remote ICCP server. The names are checked VMD-specifically . ▸ `active:` Activates the checking and allows the configuration of the **Set result in status bit** and **Variable for result names** properties. For the connection, the client on the remote server reads a DataSet that only contains existing variables. ▸ `inactive:` It is assumed that the configured ICCP variables are available on the remote server. If one or more configured ICCP variables are not on the remote server, no successful connection to the remote server can be established. Default: `Inactive`
Set result in status bit	Status bit for zenon variables of the client that are not on the remote ICCP server. Selection of the corresponding user status bit from drop-down list: ▸ `None` No user status bit is set for the variable if the variable is not present on the remote server. ICCP variables that are not present can be evaluated with **Variable for result names**. ▸ `M1 bis M8` The configured status bit is set for the variable if the variable is not present on the remote server. Default: `None` This user status bit is set for configured zenon variables if they are not found during checking on the remote server. Note: Only active if **Check presence of remote Items** is inactive. You can find further information on status processing in the Status processing manual in the User status chapter.
Variable for result names	Input field to configure a variable name for the display of the names of the missing ICCP variables. In the variable, variable names that are not present on the remote server are visualized. Is this field is empty, the evaluation is carried out with the configured user status bit. Default: `empty`

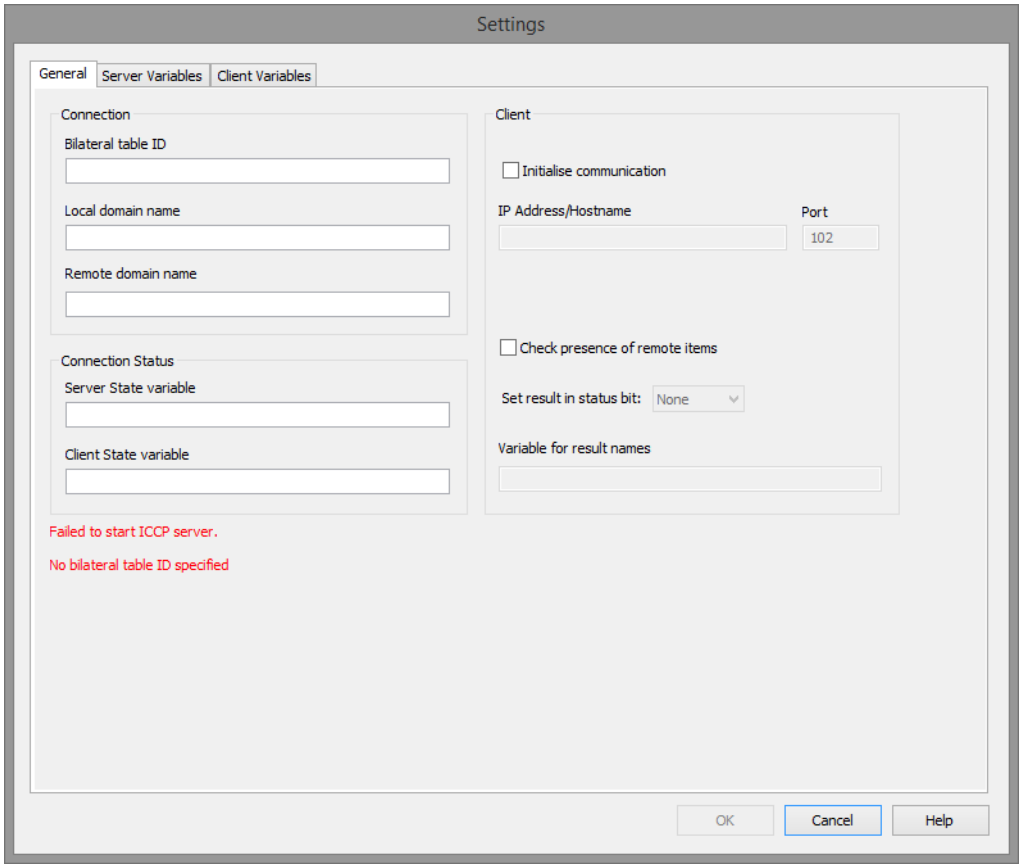
	Note: Only active if **Check presence of remote Items** is inactive. You can find further information on the display of variables that do not exist in the Show variables that are not present in Runtime (à la page 141) chapter.

CLOSE DIALOG

Parameter	Description
OK	Accepts the settings of all tabs, saves the configuration in the **zenProcGateway.ini** file, starts the ICCP server and waits for incoming connections.
Cancel	Annule toutes les modifications effectuées sur tous les onglets, puis ferme la boîte de dialogue.
Help	Ouvre l'aide en ligne.

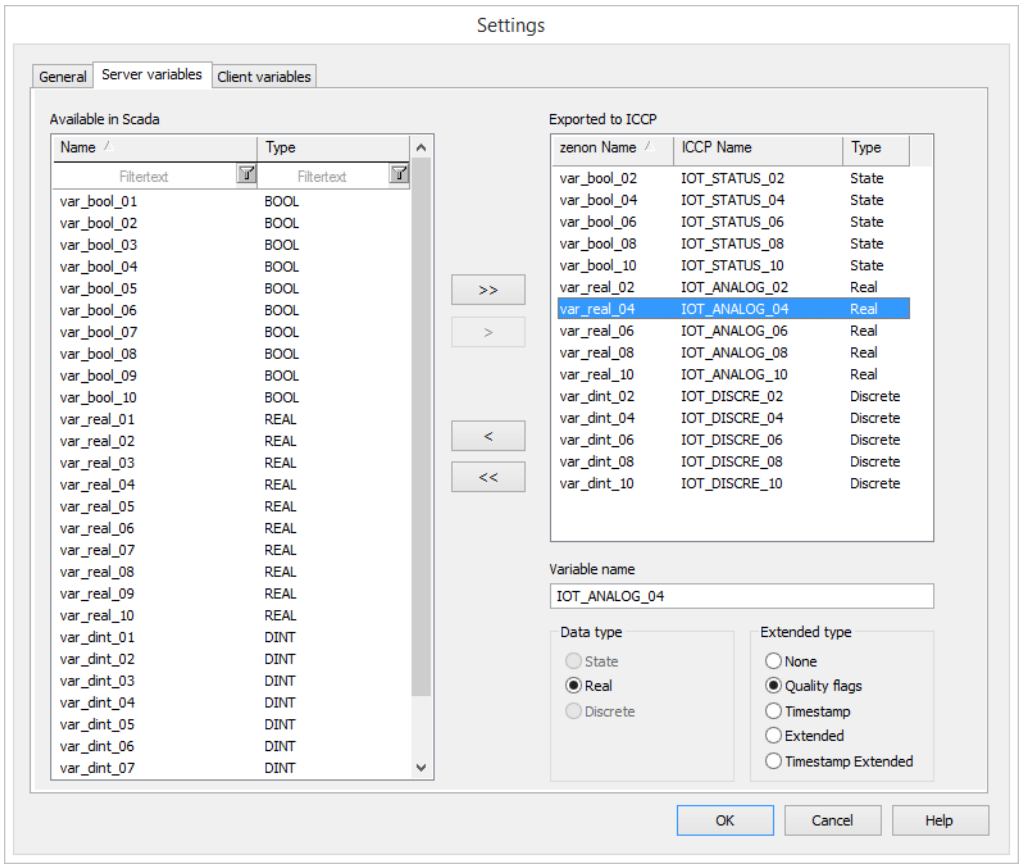
ERROR MESSAGES - EXAMPLE

Error messages are displayed in red font:



9.1.2 Server variables

Selection and configuration of the variables that the zenon Process Gateway provides as an ICCP server. The value changes of the variables are sent to the remote ICCP client spontaneously.



Parameter	Description
SCADA variables	List of all numeric variables available in the zenon project. The list can be sorted and filtered. The list shows: ▸ `Name`: Name in zenon ▸ `Type`: Data type of the zenon variable
Arrow keys	Selected variables are exchanged between the two lists by means of the `arrow key`. ▸ Key **>>**: Copies all variables from zenon to ICCP. ▸ Key **<<**: Removes all variables from the ICCP list. ▸ Key **>**: Copies selected variables from zenon to ICCP ▸ Key **<**: Removes selected variables from the ICCP list. Variables can also be added to the list for zenon by double clicking on the variable for the ICCP list.
Available for ICCP	Variables from the list are provided by the server for the remote ICCP client. The list shows: ▸ `SCADA Variable`: Name in zenon ▸ `ICCP Name`: Name in the ICCP protocol ▸ `Type`: ICCP Type The ICCP name and type can be modified using the `ICCP Name`, `Data type` and `Extended type` options.
ICCP Name	Name - VMD-specific - of the ICCP data point selected from the top list. Change to the **ICCP Name** by entering the new name. Empty if no entry is selected in the **Available for ICCP** list. Note: The ICCP name can usually be named as desired in accordance with the standard. However, it must be ensured that both communication partners use the same name.
Data type	ICCP data type of the variable selected from the top list. Change to the `Data type` by clicking on the option field:

	▸ **State** Not available if, in the **Object type** option, the `Device object` option field has been selected. ▸ **Command** This option field is only available if, in the **Object type** option field, the `Device object` option field has been activated. ▸ **Real** ▸ **Discrete** When transferring the list from the list of **SCADA Variables**, the `Type` is automatically proposed for the **Available from ICCP** list. Note: Standard display of zenon variables on ICCP: ▸ `USINT/BOOL` to **State**; value range 0..3, **DPI mapping**: • `0/false - off` (`tripped`), • `1/true - on` (`closed`), • `2 - intermediate` (`between`), • `3 - fault` (`invalid`). ▸ `REAL` to **Real**. ▸ `DINT/INT/SINT` to **Discrete**. Depending on the variables provided by zenon, data types can also be unavailable (grayed out).

Extended type	ICCP suffix of the variable selected from the top list. Change to the **Extended type** by clicking on the option field: ▸ `None`: No ICCP addition (only value of the variable); IEC 60870-6-802 IndicationPoint type: Data_Discrete Data_State, Data_Real. ▸ `Quality flags`: Value with quality; for example: Data_DiscreteQ. ▸ `Timestamp`: Value + quality + time stamp [s]; for example: Data_DiscreteQTimeTag. ▸ `Extended`: Value + quality + time stamp [s] + change counter; for example: Data_DiscreteExtended. ▸ `Timestamp Extended`: Value + quality + extended time stamp [ms]; for example: Data_DiscreteQTimeTagExtended. Depending on the variables provided by zenon, data types can also be unavailable (grayed out).

Attention

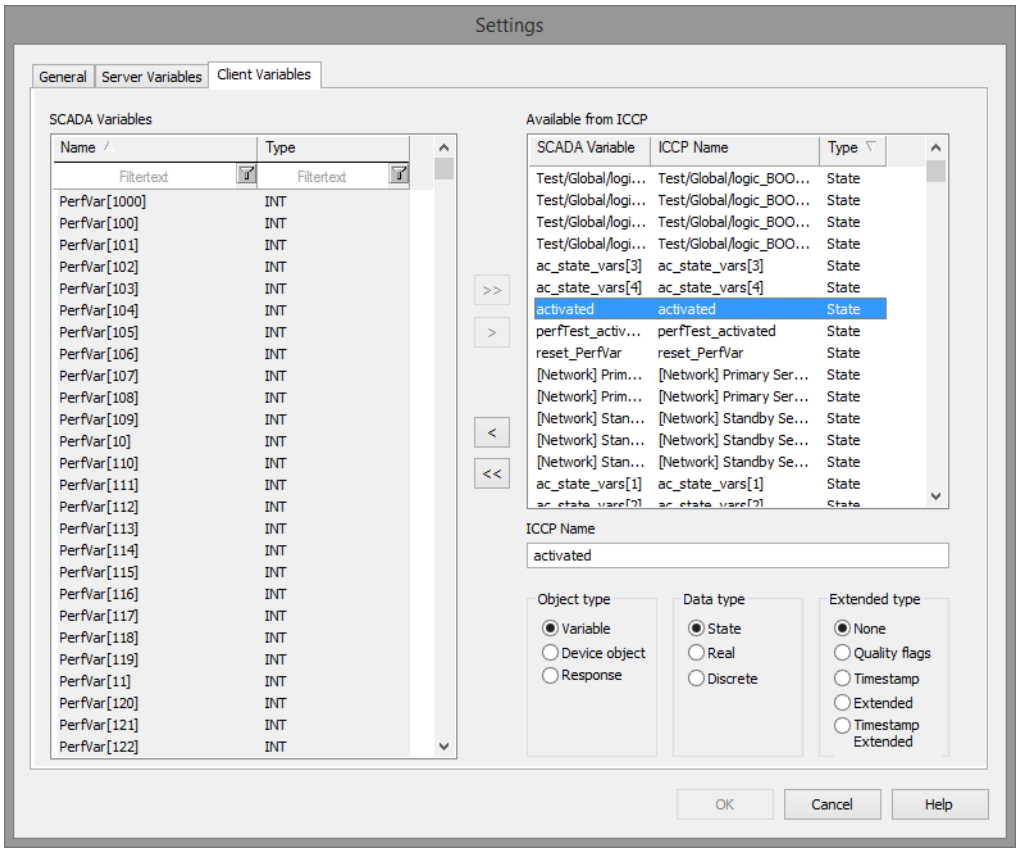
Data type and **Extended type** must be correctly assigned for both communication partners. Incorrect amendments lead to the received data sets not being interpreted 100% correctly. All other information in the data set is lost!

CLOSE DIALOG

Parameter	Description
OK	Accepts the settings of all tabs, saves the configuration in the **zenProcGateway.ini** file, starts the ICCP server and waits for incoming connections.
Cancel	Annule toutes les modifications effectuées sur tous les onglets, puis ferme la boîte de dialogue.
Help	Ouvre l'aide en ligne.

9.1.3 Client variables

Selection and configuration of the variables that the zenon Process Gateway uses as an ICCP client. The client transmits the value changes that it receives from the remote ICCP server to the zenon variables.



Parameter	Description
SCADA variables	List of all numeric variables available in the zenon project. The list can be sorted and filtered. The list shows: ▸ `Name`: Name in zenon ▸ `Type`: Data type of the zenon variable
Arrow keys	Selected variables are exchanged between the two lists by means of the `arrow key`. ▸ Key **>>**: Copies all variables from zenon to ICCP. ▸ Key **<<**: Removes all variables from the ICCP list. ▸ Key **>**: Copies selected variables from zenon to ICCP ▸ Key **<**: Removes selected variables from the ICCP list. Variables can also be added to the list for zenon by double clicking on the variable for the ICCP list.
Available from ICCP	Variables from the list are expected from the remote ICCP server. The list shows: ▸ `SCADA Variable`: Name in zenon ▸ `ICCP Name`: Name in the ICCP protocol ▸ `Type`: ICCP Type The ICCP name and type can be modified using the **ICCP Name, Data type** and **Extended type** settings.
ICCP Name	Name - VMD-specific - of the ICCP data point selected from the top list. Change to the **ICCP Name** by entering the new name. Empty if no entry is selected in the **Available for ICCP** list. Note: The ICCP name can usually be named as desired in accordance with the standard. However, it must be ensured that both communication partners use the same name.

Parameter	Description
Object type	Type of the SCADA variable for communication to the ICCP server. With this object, variables can be configured as command or response variables. Change the `Object type` by clicking on the option field: ▸ `Variable` Standard type of a variable. Activate this option if the zenon variable is not a command variable or a response variable. ▸ `Device object` The variables from the list of **SCADA Variables** are configured as a command variable if the option is activated. In doing so, the Process Gateway, on receipt of a spontaneous value, sends an `Operate Request` to the ICCP server. By activating this option field, in the **Data type** option, the `State` option field switches the selection possibility to `Command`. ▸ `Response` The variable from the **SCADA Variables** list is configured as a response variable if the option is activated. When receiving a spontaneous value, the response from the server (`Indication`) is written to the response variable via the `operate` command. No further project configuration is necessary for this process. Default: `Variable`

Parameter	Description
Data type	ICCP data type of the variable selected from the top list. Change to the `Data type` by clicking on the option field: ▸ **State** Not available if, in the **Object type** option, the `Device object` option field has been selected. ▸ **Command** This option field is only available if, in the **Object type** option field, the `Device object` option field has been activated. ▸ **Real** ▸ **Discrete** When transferring the list from the list of **SCADA Variables**, the `Type` is automatically proposed for the **Available from ICCP** list. Note: Standard display of zenon variables on ICCP: ▸ `USINT/BOOL` to **State**; value range 0..3, **DPI mapping**: • `0/false - off` (`tripped`), • `1/true - on` (`closed`), • `2 - intermediate` (`between`), • `3 - fault` (`invalid`). ▸ `REAL` to **Real**. ▸ `DINT/INT/SINT` to **Discrete**. Depending on the variables provided by zenon, data types can also be unavailable (grayed out). Note: Hidden if the `Response` action field is activated for the **Object Type** option.

Parameter	Description
Extended type	ICCP suffix of the variable selected from the top list. Change to the **Extended type** by clicking on the option field: ▸ None: No ICCP addition (only value of the variable); IEC 60870-6-802 IndicationPoint type: Data_Discrete Data_State, Data_Real. ▸ Quality flags: Value with quality; for example: Data_DiscreteQ. ▸ Timestamp: Value + quality + time stamp [s]; for example: Data_DiscreteQTimeTag. ▸ Extended: Value + quality + time stamp [s] + change counter; for example: Data_DiscreteExtended. ▸ Timestamp Extended: Value + quality + extended time stamp [ms]; for example: Data_DiscreteQTimeTagExtended. Depending on the variables provided by zenon, data types can also be unavailable (grayed out). Note: Hidden if either the Device object or Response option field is activated for the **Object Type** option.

Attention

Data type and **Extended type** must be correctly assigned for both communication partners. Incorrect amendments lead to the received data sets not being interpreted 100% correctly. All other information in the data set is lost!

CLOSE DIALOG

Parameter	Description
OK	Accepts the settings of all tabs, saves the configuration in the **zenProcGateway.ini** file, starts the ICCP server and waits for incoming connections.
Cancel	Annule toutes les modifications effectuées sur tous les onglets, puis ferme la boîte de dialogue.
Help	Ouvre l'aide en ligne.

9.2 Status variables of the connection

Status variables of communication are used for the display of the connection status between the Process Gateway and the remote ICCP communication partner. A separate variable each is configured for the ICCP server and ICCP client. The Process Gateway only communicates this status to zenon Runtime. There is no communication of values of these variables to remote ICCP communication partners.

ENGINEERING

To show the connection status in Runtime, carry out the following project configuration steps:

1. In your zenon Editor, create two numeric variables.
   It is recommended that these are configured for the internal driver.
   - The variable cannot be a STRING variable.
2. Configuration in the Process Gateway:
   - Status variable for server connection:
     Enter the name of the variable in the **Server state variable** input field in the **General** tab in the Process Gateway module configuration.
   - Status variable for client connection:
     Enter the name of the variable in the **Client state variable** input field in the **General** tab in the Process Gateway module configuration.

ASSIGNED STATES

The connection status variables have the following values:

Parameter	Description
0 IDLE	▸ Communication inactive: Directly after the start of Process Gateway or reloading the zenon project. ▸ Connection disconnected again: Once Process Gateway has been closed or after the settings in Process Gateway have been edited, and before the establishing of a new connection.
1 CONNECTING	Communication via TCP/IP and MMS is being established. This status is shown if an incoming CR (Connection Request) has been received or a CR (Connection Request) has been sent successfully. This depends on which page has established the TCP/IP connection. This status is the same for server and client.
2 CONNECTED	ICCP communication has been established successfully. This status is set to a logical connection if a dataset (DSTransferSet) has been agreed for the transfer of data and the first MMS.write request (logical server connection) or the first MMS.write response (logical client connection) has been received. If the ICCP client does not request any data from the remote server, the status of the logical connection remains CONNECTED. Note: The client does not request any data if: ▸ The list of variables is empty **Available from ICCP**. ▸ None of the variables from the **Available from ICCP** list are available and the **Check presence of remote items** check has been activated.
3 ERROR	This status is shown if the TCP/IP connection as been interrupted. This status is the same for both logical connections.

9.3 Show non-present variables in Runtime

With the help of this project configuration, you can evaluate in zenon Runtime which of the variables configured on the ICCP client are not present on the remote server.

This evaluation can be carried out with two project configurations:

1. Display of the missing variable names in an additional string variable.
2. Display of a user status bit when there are variables that do not exist.

REQUIREMENT

- The checking of the variable names of the current client connection must be activated: Checkbox **Check presence of remote items** in the **General** tab in the Process Gateway module configuration.

DISPLAY OF THE MISSING VARIABLES IN A STRING VARIABLE

To display the missing variables in Runtime in one variable:

1. In the zenon Editor, create a STRING variable.
   It is recommended that these are configured for the internal driver.
2. Configuration in Process Gateway:
   Enter the name of the variable in the input field**Variable for result names** in the **General** tab in the Process Gateway module configuration.

> **Informations**
>
> The value of the variable in Runtime contains the name of the non-present variables, separated by a comma (;).

DISPLAY OF A USER STATUS BIT WHEN THERE ARE VARIABLES THAT DO NOT EXIST.

To mark the zenon variables in Runtime with a user status bit if they are not present on a remote ICCP server:

- Select the desired user status bit from the drop-down list of the **Set result in status bit** property in the **General** tab in the Process Gateway module configuration.

Note: This user status bit can be evaluated in the Runtime with a **reaction matrix** or a **combined element**.

9.4 Direct control (non-SBO) for ICCP control objects

- A zenon variable can be linked to an `ICCP control object`. A `Direct Control` is executed on receipt of a spontaneous value for this command variable. In doing so, the process gateway sends an `Operate Request` to the ICCP server.
- A zenon variable can be linked to an `ICCP control object` as a response variable. The result of the `Direct Control` operation (= `indication`) is written to the configured zenon result variable (by writing a set value).

- Each received `indication` is written to the corresponding ICCP response variable.
  - In doing so, there is no check to see whether a command is pending.
  - If no `indication` is received, the value of the response variable remains unchanged.
- `direct control` - supported telegram types
  - `Operate (request)`
  - `Timeout (indication)`
  - `Local Reset (indication)`
  - `Success (indication)`
  - `Failure (indication)`
- The response variable is set to a stipulated value when sending a new command.
- The configuration is validated when the gateway is initialized:
  - If several response variables are configured for each `device object`, an error message is created when the **Process Gateway** is started and only the first one is used
  - Only one command can be configured per `device object` .
  - A command variable must be present for each response variable, otherwise there will be an error message.

PROJECT CONFIGURATION IN THE PROCESS GATEWAY

- Open the tab in the **Process Gateway**. **Client Variables**
- Configure command variable
- Configure response variable

  A response from an ICCP server is assigned to a **Device object** or to a variable by getting the same ICCP name.

RESULT OF THE WRITE REQUEST

If a **Response Object** is configured to a variable or a **Device Object**, the result of the `write Requests` is written to this object:

Parameter	Description
0	Write successful
1	Write pending – waiting for response
200	Write failed: OBJECT_INVALIDATED
201	Write failed: HW_FAULT
202	Write failed: TEMPORARILY_UNAVAILABLE
203	Write failed: OBJECT_ACCESS_DENIED
204	Write failed: OBJECT_UNDEFINED
205	Write failed: INVALID_ADDRESS
206	Write failed: TYPE_UNSUPPORTED
207	Write failed: TYPE_INCONSISTENT
208	Write failed: OBJECT_ATTRIBUTE_INCONSISTENT
209	Write failed: OBJECT_ACCESS_UNSUPPORTED
210	Write failed: OBJECT_NON_EXISTENT
211	Write failed: OBJECT_VALUE_INVALID

9.5 Configuration file: INI entries for ICCP-Tase.2

The configuration file `zenProcGateway.ini` contains this entries for ICCP-Tase.2.

[ICCP]

Entrée	Description
SERVER_PORT=	Numéro de port IP du serveur ; le port autorise la connexion des clients distant. Le port est ouvert au démarrage et reste ouvert. Par défaut : 102 Remarque : Cette entrée INI ne peut pas être configurée dans l'interface utilisateur graphique de la passerelle ICCP-TASE.2 Process.
MAX_CONNECTIONS=	Nombre maximum de connexions d'autres clients ICCP distant au serveur. Est ignoré si **CLIENT_AUTO_CONNECT**=1 a été configuré. Par défaut : 0 (= illimité) Remarque : Cette entrée INI ne peut pas être configurée dans l'interface utilisateur graphique de la passerelle ICCP-TASE.2 Process.
MAJOR_VERSION_NUMBER=	Numéro de version principal du protocole ICCP, par exemple : 1996, 2000. Par défaut : 2000 Remarque : Cette entrée INI ne peut pas être configurée dans l'interface utilisateur graphique de la passerelle ICCP-TASE.2 Process.
MINOR_VERSION_NUMBER=	Numéro de version secondaire du protocole ICCP. Par défaut : 8 Remarque : Cette entrée INI ne peut pas être configurée dans l'interface utilisateur graphique de la passerelle ICCP-TASE.2 Process.
BILATERAL_TABLE_ID=	Bidirectional Table ID Ceci est synchronisé lors de l'établissement d'une connexion entre tous les partenaires de communication de l'ICCP. Remarque : L'identifiant unique de tableau bilatéral doit être identique sur tous les équipements communiquant avec le module **ICCP-TASE.2 Process Gateway**.
LOCAL_DOMAIN_NAME=	Nom de domaine ICCP local ; configuration du serveur, doit être définie en miroir (c'est-à-dire inversée) par rapport au partenaire de communication.
REMOTE_DOMAIN_NAME=	Nom de domaine ICCP local ; configuration du client, doit être définie en miroir (c'est-à-dire inversée) par rapport au partenaire de communication.

CLIENT_AUTO_CONNECT=	Configuration du projet pour savoir si le client initie automatiquement l'établissement d'une connexion au serveur ICCP lors Process Gateway est démarré. 0 : ne pas initialiser - attendre simplement 1 : initialiser
CLIENT_IP_NAME=	Adresse IP ou nom d'hôte du serveur distant si ce client doit initialiser la communication.
CLIENT_PORT=	Numéro de port du serveur distant si ce client doit initialiser la communication.
CHECK_SERVER_Variables=	Vérifiez si toutes les variables configurées sont disponibles sur le serveur distant. 0 : pas de vérification des noms de variables configurés. 1: Vérification des noms de variables configurés.
CHECK_SERVER_VARIABLES_MERKER_BIT=	Bit d'état pour les variables client qui ne sont pas présentes sur le serveur si un contrôle des variables configurées est activé. 0 : pas de bit d'état pour les variables client manquantes 1 à 8 : Bit d'état utilisateur M1 à M8
SERVER_STATE_VARIABLE=	Nom de la variable pour l'affichage de l'état de la connexion entre le serveur ICCP et le client distant.
CLIENT_STATE_VARIABLE=	Nom de la variable pour l'affichage de l'état de la connexion entre le client ICCP et le serveur distant.
NOT_AVAILABLE_SERVER_VARIABLES_VARIABLE=	Nom de la variable pour l'affichage des noms de variables (dans zenon Runtime) qui ne sont pas disponibles pour la connexion client actuelle sur le serveur distant.
CALLING_OSI_SESSION_SELECTOR= **CALLING_OSI_PRESENTATION_SELECTOR=** **CALLING_OSI_AE_QUALIFIER=** **CALLING_OSI_AP_TITLE=** **CALLED_OSI_SESSION_SELECTOR=** **CALLED_OSI_PRESENTATION_SELECTOR=** **CALLED_OSI_AE_QUALIFIER=** **CALLED_OSI_AP_TITLE=**	Paramètres de communication OSI Remarque : Cette entrée INI ne peut pas être configurée dans l'interface utilisateur graphique de la passerelle ICCP-TASE.2 Process.

[VARIABLES]

Groupe de références configurées entre les variables de zenon et les variables ICCP. Les éléments suivants sont mentionnés dans la liste :

- Paramètres de chaque variable ICCP
- Nombre de variables
- Nom des variables

Entrée	Description
PARAM_0=	Paramètre de le nom ICCP. La numérotation commence par zéro `0`. Le compteur est incrémenté de 1 à chaque nouveau paramètre. L'identifiant correspond à l'identifiant du nom de variable. **PARAM_0** correspond à **NAME_0**. Exemple : **PARAM_0**=`1,0,0,ICCP_Name`
COUNT=	Nombre de références configurées
NAME_0=	Nom de la variable de zenon La numérotation commence par zéro `0`. Le compteur est incrémenté de 1 à chaque nouveau paramètre. Cet identifiant correspond à l'identifiant du paramètre. **PARAM_0** correspond à **NAME_0**. Exemple : **NAME_0**=`MyName`

9.6 LOG entries

GENERAL

Parameter	Level	Note
`Failed to get %s application object.`	**ERROR**	Start zenon Runtime before you start Process Gateway.
`Failed to start ICCP server.`	**ERROR**	Configure the settings of the Process Gateway. Configure a different port for each Process Gateway when starting several Process Gateways.
`Failed to interpret`	**ERROR**	Match the ICCP '**Data type**' and '**Extended type**'

report for data set %s.%s. Bad message structure. ▸ Failed to interpret report for %s. Bad message structure. ▸ Failed to interpret write request for %s.%s. Bad message structure. ▸ Failed to interpret received data for %s.		between Process Gateway and remote ICCP communication partners.
Failed to write '%s', error code: %u.	**ERROR**	The remote ICCP server responds to the write request in the negative, with MMS error code %u. Correct the settings on the remote server and/or move the variable from the client configuration to the server variables in Process Gateway.

AUTOMATIC ESTABLISHMENT OF A CONNECTION

Parameter	Level	Description
Could not resolve host name %s. Errorcode: %d	**ERROR**	The pre-set host name or the IP address could not be resolved. ▸ %1 Host name or IP address of the set counterparty. ▸ %2 Windows socket error code
Autoconnection is not possible because no hostname (or IP address) is defined	**ERROR**	The pre-set host name or the IP address are resolved to the address 0.0.0.0 and are thus invalid.
Maximum number of %1 connections reached. Declining new connection from %2.	**MSG**	Rejection of an incoming connection that has already achieved the maximum number of possible connections: ▸ %1 Maximum number of connections could be established. ▸ %2 IPv4 address of the counterparty.
Cannot verify selected host name %1. Declining new connection from %2.	**MSG**	The configured host name cannot be resolved to the IP address.

		▸ %1 Configured host name ▸ %2 IPv4 address of the counterparty.
Only connection from %1 allowed. Declining new connection from %2.	**MSG**	An incoming connection is rejected because the source address does not correspond to the IP address of the configured host for automatic establishment of a connection. ▸ %1 configured IP address of the set host. ▸ %2 IPv4 address of the counterparty.
Only one active connection allowed. Declining new connection from %2.	**MSG**	An incoming connection is rejected because a connection has been established automatically. ▸ %2 IPv4 address of the counterparty.
Creating a socket connection to the ICCP server %s [%2:%3].	**DEBUG**	A TCP socket connection to the configured counterparty is established. ▸ %1 Host name or IP address of the set counterparty. ▸ %2 IPv4 address of the counterparty. ▸ %3 Port number of the counterparty.
Could not create the socket connection to the ICCP server %1. Errorcode: %2.	**DEBUG**	A TCP socket connection to the set counterparty could not be established within the socket timeout. ▸ %1 Host name or IP address of the set counterparty. ▸ %2 Windows socket error code.

STATUS VARIABLES OF THE CONNECTION

Parameter	Level	Description
Server state variable with name '%s' does not exist or has an invalid type.	**ERROR**	The configured status variable to display the status of the server connection does not exist in the project or has an invalid data type (required data type: numeric). ▸ %s Configured name of the status variable.
Client state variable with name '%s' does not	**ERROR**	The configured status variable to display the status of the client connection does not exist in the

exist or has an invalid type.		project or has an invalid data type (required data type: numeric). ▸ %s Configured name of the status variable.
set server state: %1(%2)	**MSG**	New status of a pre-existing server connection. ▸ %1 Numeric value of the current status of the server connection ▸ %2 Current status of the server connection as a text (IDLE, CONNECTING, CONNECTED, ERROR)
set client state: %1(%2)	**MSG**	New status of a pre-existing client connection. ▸ %1 Numeric value of the current status of the client connection ▸ %2 Current status of the client connection as a text (IDLE, CONNECTING, CONNECTED, ERROR)

DISPLAY OF VARIABLES NOT PRESENT

Parameter	Level	Description
Unavailable server items variable with name '%1' does not exist or has an invalid type.	**ERROR**	The configured status variable to display the name of the variable that does not exist does not exist in the project or has an invalid data type (required data type: STRING). ▸ %1 Configured name of the status variable.
Error configured client variable '%1' is not available on the remote device.	**ERROR**	A variable configured in Process Gateway for the client is not available on the remote ICCP server. ▸ %1 ICCP name of the variable that is not present on the remote server

Parameter	Level	Description
Response variable [iccp name] already configured.	**ERROR**	Error message for error projecting in the Process Gateway. With this error message, an ICCP client was assigned more than one SCADA variable. The first Response Object is applied, all other Response Objects are discarded.

No ICCP object found for configured response variable [iccp name]	**ERROR**	Error message for error projecting in the Process Gateway. With this error message an ICCP client was assigned a SCADA variable for a Response Object however the assignment of a matching Device object is missing.

9.7 Interoperability

Process Gateway for ICCP-TASE.2 based on the **IEC 60870-6 TASE.2 ICCP** protocol.

Supported protocol versions: 2000.8 and 1996.8.

ICCP-TASE.2 supports Conformance Blocks 1 and 2 as client and server. As client Direct Control of Block 5 is supported.

The Process Gateway acts like a server by default when establishing communication.

10. IEC870 Slave

LANGUAGE

The IEC870Sl Process Gateway and its dialogs for configuration are only available in English.

STRUCTURE

You can configure several virtual controllers (**devices**) in the IEC870 slave gateway. According to the used protocol, this virual devices are associated directly to a Master (870-104) or polled by a Master (870-101). The **devices** can contain several sectors. These sectors contain IOs (Information Objects) representing the actual variables. An IO refers to a variable of the process control system.

OPERATION OF SEVERAL IEC870 SLAVES

IEC870Sl Process Gateway can be started on both a standalone computer as well as in the zenon **network** (server/standby/client). All Process Gateways that are started in the network provide the same process image, but to the extent that is given in the configuration file.

Network servers are usually computers with a higher load. In large and complex systems, it may be advisable to operate the gateways on (dedicated) network clients. This has another benefit: with redundancy switching, a network client automatically connects to the respective primary server and continues to provide the data to higher-level systems.

Several instances of the IEC870 slaves can use the same or very similar configurations that have been transferred to several files, such as with export/import (à la page 191) for example.

It is also possible to start several instances of the IEC870 slave on one computer.
requirement: separate communication partner for each instance - no conflicts with IP or COM ports.

REDUNDANCY OF THE IEC870 SLAVE

If two Process Gateways with the same configuration have been started then they are like a redundant slave for an 870 master, i.e. a redundant connection to the zenon system.

Two Process Gateways with the same configuration can be started on two computers in the zenon **network**, or on one computer with 2 Ethernet cards, or with different IP ports.

In the messaging direction, both IEC870 slaves provide the same Events and measured values, because zenon Runtime provides exactly the same process image. In command direction, one of the two slaves can be set to `silent mode` via an IO with the internal type T00 and IOA 2 (à la page 210). The commands of the 870 master are thus only forwarded via a connection to Runtime.

The redundancy at Data Layer - according to the procedures of Edition 2 of the IEC 60870 standard, is not supported.

IDENTIFICATION AND ADDRESSING

For each respective started instance of the IEC870Sl Process Gateway:

- 870-101: The virtual controller (**device**) is identified by the link address of the Slave.
- 870-104: The virtual controller is identified by the IP address of the master.
- Sector: Is addressed via **COA** (Common Object Address/Common Address of ASDU).
- IO: Is addressed by **Type Identification** and **IOA**.

In the following chapters, you will read how to configure the IEC870SI Gateway and how communication takes place.

Informations

*For variables that are configured in the IEC870Sl Process Gateway as command variables, each command is logged as **Send value** via the API interface during Runtime in the CEL if the **Valeurs modifiées par VBA** property is active (**Enregistrer dans la liste d'événements** group).*

10.1 Module configuration

You have many settings options available in the IEC870 slave process gateway. The configuration of the IEC870SI is stored in a XML file (à la page 191).

The lists in the configuration dialogs can generally be sorted. To change the sorting sequence, simply click on the column title.

870-101 PROTOCOL VS 104 PROTOCOL

The gateway supports both protocol IEC 60870-5-101 (abbreviated to: 870-101, serial communication) as well as IEC 60870-5-104 (abbreviated to: 870-104, TCP/IP communication). The settings for protocols -101 and -104 differ.

In the main window **Settings**, you can choose the protocol you want to use and define general settings:

- (Protocol) 870-101 (à la page 156)
  Unlocks the settings for protocol 101 and locks other, irrelevant input fields.
- (Protocol) 870-104
  Unlocks the settings for protocol 104 and locks other, irrelevant input fields.

DEVICES

Virtual devices (connections) are created and deleted in the **Devices** list in the main window. A **device** defines how the Process Gateway, as an IEC870 slave, should work for a certain 870 master. These **devices** are configured in the tabs. Several devices can be set up - for several masters.

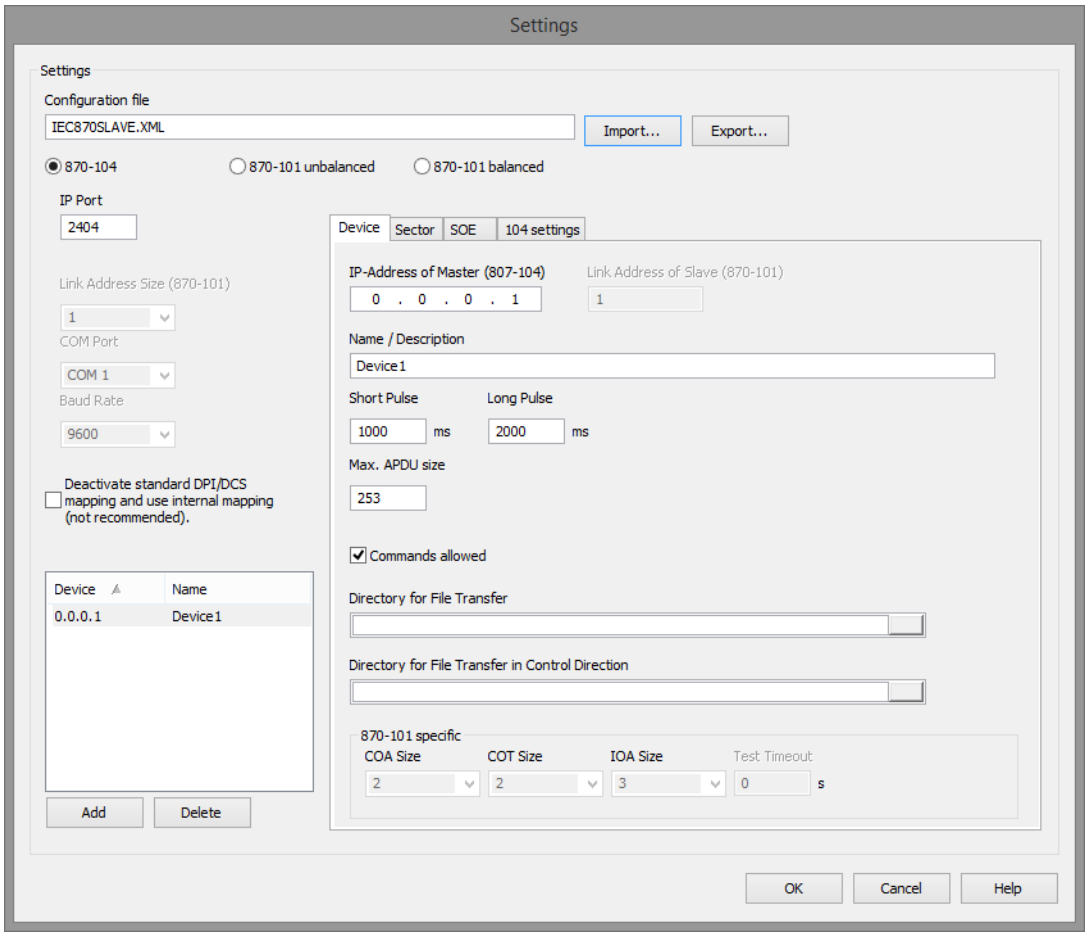
The following tabs are available in the process gateway:

- **Device** (à la page 161) - general settings and master IP address (-104) or link address of the slave (-101).
- **Sector** (à la page 166) - settings valid for a Common Address of ASDU (**COA**). A device should contain at least one sector; and one sector should contain at least one Information Object (**IO**):
  - **Information Objects** - the list of the Information Object Address (**IOA**), their Type Identifications (**TID**) and the assignment to zenon variables.
- **Sequences of Events (SOE (à la page 201))**
  SOE according to procedures defined in IEC 60870-5-101: 7.4.11.3 "Transmission of sequences of events" (in a section of a data file).
- **104 settings** (à la page 188)
  Data Layer Timeouts and APDU counters

Informations

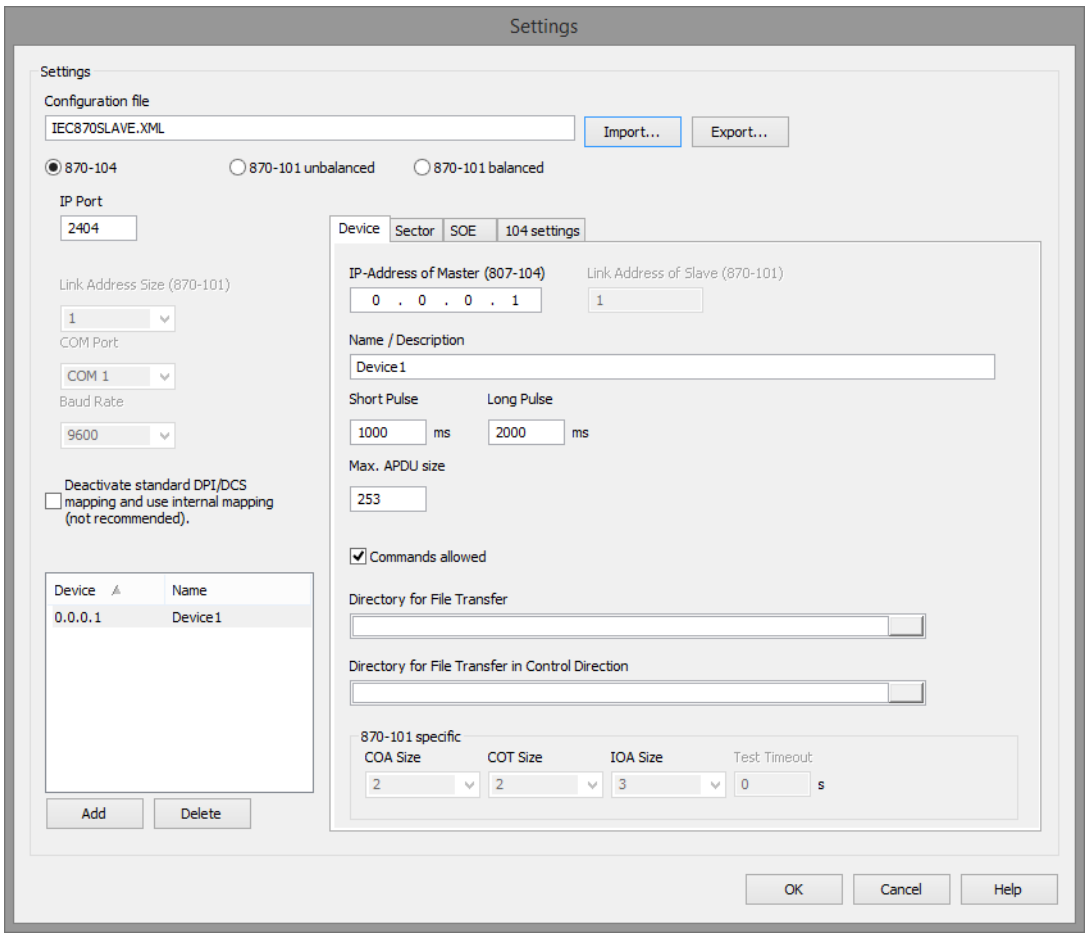
Only the general properties are documented in this section. You can find the protocol-dependent configurations in the section for the respective protocol (101) or in the description of the respective device settings (101 and 104).

TABS FOR THE CONFIGURATION OF THE DEVICES



Parameter	Description
Device	Tab for the configuration of a device (à la page 161). A virtual device corresponds to an interface with an 870 master.
Sector	Tab for the configuration of the sectors (à la page 166) of the device. A sector corresponds to a Common Object Address/Common Address of ASDU and is addressed via **COA** . Note: A sector can contain several **Information Objects** that are linked to zenon variables and addressed with **Type Identification** (**Type ID**) and **Information Object Address** (**IOA**).
SOE	Tab for the configuration of the recording and transfer of the SOE files (à la page 201) (Sequences of Events)
104 settings	Tab for the configuration of the Protocol-870-104-specific settings for timeouts (à la page 188) and APDU counter.
OK	Accepts all changed settings. They will immediately be stored in the current configuration file (as defined in the **Configuration file**). Attention: If you have not made any changes, close the dialog with the **Cancel**, not **OK**. After clicking on **OK**, all existing connections with the 870 master will be closed and restarted with the new settings.
Cancel	Discards all changes made and closes the dialog window. The dialog is closed without the pre-existing communication to 870 masters being interrupted.
Help	Ouvre l'aide en ligne.

10.1.1 General Settings



Informations

Only the general properties of the IEC870 Slave are documented in this section. You can find further information in the description of the virtual **device**.

SETTINGS

General settings:

Parameter	Description
Configuration file	Name of the XML configuration file with the configuration of the IEC870 slave in the Process Gateway. This is where all active settings are stored and also loaded from. The name you enter must be a valid file name. Otherwise, no data can be stored. In this case, you will get an error message. Only enter the file name without path. You do not have to add the file extension XML, as it will be added automatically during saving. The current INI directory will be used as path. Save location: %CD_SYSTEM% You can find the INI file in the system folder of the process control system. It is called '<SCADA>ProcGateway.ini' - <SCADA>is the placeholder for the name of the process control system. The file name will automatically be entered in the INI file of the process gateway as configuration. You can find the description of the structure of the XML file in the XML file (à la page 191) chapter.
Import...	Loads a project configuration (à la page 197) from an XML or CSV file. The current settings are replaced by the ones stored in the file. The current setting is retained for **Configuration file**. Clicking on the **...** button opens the file selection dialog to select a configuration file. Note: You can thus transfer the configurations of other projects or import the configurations with large amounts to variables that you have prepared and exported with other applications. The file content is checked during import. A corresponding warning dialog is shown in the event of an error.
Export...	Exports current project configuration (à la page 197) as an XML or CSV file. Clicking on the **...** button opens the file selection dialog to select a save location for the target file. Attention: A **Device** that still has the IP or `link address` issued on creation is not exported. To export it, you must first configure the connection to the 870 master in the **Device** (à la page 161) configuration window. Note: ▸ No file is exported if there are no IOs in the current Process Gateway configuration. ▸ The configured `IP address` is exported for 870-104 connections. ▸ The configured `Link address` is exported for 870-101.

870-104	activates protocol 104, unlocks all input fields for 104 and locks settings for 101. In this mode, several slaves can communicate with several masters via the TCP/IP interfaces. Each TCP socket defines a 1:1 connection between slave and master; it is thus possible to transfer the value changes to the slave spontaneously. The loss of the connection can be detected both by the slave and master thanks to the test procedures running in the background (replacement of small APDUs on the Link Layer).
870-101 unbalanced	Activates protocol 101 in `unbalanced Mode`, unlocks all specific settings for 101 and locks settings for 104. In this mode, several slaves can communicate with several masters via the serial interface. A slave should only transfer the data to a master on one (cyclical) Link Layer request, otherwise conflicts could occur in the Link Layer. There is no test procedure for slaves to check whether the connection to the master continues to exist.
870-101 balanced	Activates protocol 101 in `balanced Mode`, unlocks all specific settings for 101 and locks settings for 104. In this mode, only a slave can communicate with a master using the serial interface. It allows the slave to transfer the value changes spontaneously. The loss of the connection can be detected both by the slave and master thanks to the test procedures running in the background (replacement of small APDUs on the Link Layer).
IP Port	IP Port for communication via IEC 60870-5-104 protocol. Slave port to which the masters are to connect. The protocol defines this as fixed at 2404. Default: `2404` Allowed port numbers: `1200` to `65535` Note: Applicable for all connections for protocol 104. Inactive if `870-101 unbalanced` or `870-101 balanced` has been selected as a protocol. It is possible to have several instances of the Process Gateway running at the same time. That is why you can change this port.
Link Address Size (870-101)	Size of the link layer address, in bytes, for communication via the IEC 60870-5-101 protocol. Select from drop-down list. Default `1` Note: The same size is applicable to all connections to 870-101 masters. Inactive if 870-104 is selected as a protocol.
COM Port	Serial interface of the computer with zenon Runtime. Select from drop-down list. Default `1` Note: Applicable to all connections to 870-101 masters (`balanced` and `unbalanced`). Inactive if 870-104 is selected as a protocol.

Baud Rate	Data transfer rate. Select from drop-down list. Default `9600` Note: Applicable to all connections to 870-101 masters (`balanced`and `unbalanced`). Inactive if 870-104 is selected as a protocol.
Deaktivate standard DPI/DCS mapping and use internal mapping	Deactivates the standard communication via DPI/DCS and uses internal mapping in communication. The non-compliant behavior is disabled when this option is activated. Default: `Inactive` Attention: Activation of this option is not recommended.

10.1.2 Device List

The virtual devices in the IEC870 slave - **Devices** - are created and deleted in the left area of the configuration dialog.
The details are configured in the right area of the dialog with the corresponding tabs.

To change the configuration of an existing **device**, select the desired **Device** in the list. The attendant configurations can then be amended in the right area.

Parameter	Description
Device	Identification of the device: IP address of the master (104) or link address of the slave (101). Default: ???? (Not defined) Attention: Before you can use or export the device, you must configure it.
Name	Name of the device/connection - desired text for easier identification.
Add	Adds a new device with standard settings. By default, it gets the invalid ID "????" and invalid IP/link addresses. Attention: Before you can use or export the device, you must configure it.
Delete	Deletes the selected device. This button is not active if no device has been selected. Attention: The selected entry is deleted immediately without a request for confirmation beforehand.

A device is created or deleted on the left-hand side of the dialog.

Clicking on the **Add** button adds a device, which is configured in the **Device**, **Sector**, **SOE** and **104 settings** tabs. Several devices can be created. A simple click on the respective device in the list opens the attendant configuration.

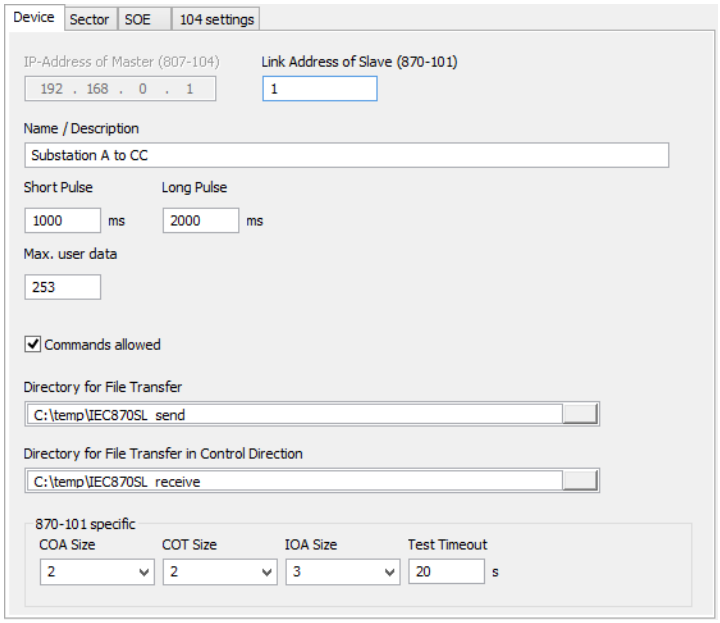
VALIDATION

Configurations are validated by clicking on the **OK** button. A corresponding warning dialog is shown in the event of an error. The check takes the selected connection into account (870-104 or 870-101). If the configuration is transferred from a file, the file content is also taken into account during validation.

When the IEC870 Process Gateway starts, there is also an initial check of the existing configuration.

10.1.3 Device - Configuration

You define the settings of the virtual device in the **Device** tab in the main window. Depending on the protocol selected - 870-104 or 101 - certain areas are grayed out. A precise definition of which input fields are available for which protocol is in the list below.



Parameter	Protocol	Description
IP-Address of Master (807-104)	104	IP address of the master, which is permitted to establish a connection to the slave. The **device** that the master communicates with is detected using the IP address. All IP addresses except 255.255.255.255 are allowed. The configured IP address is shown in the list of `Devices` in the `Device` column. Default: `255.255.255.255` Note: The current value is lost if the protocol is changed.
Link Adress of Slave (870-101)	101	The Link Layer address of the slave. The master uses the link address contained in the TF 1.2 frame to specify the **device** for which the telegram is applicable. With link addresses, a Master can communicate with multiple devices. The allowed range of the address depends on the **Link Address Size** setting in the main settings. ▸ **Link address size** = 0 Only applicable for balanced Mode. No range permitted. The connection is unique (1 Slave : 1 Master) ▸ **Link address size** = 1 Permitted range: 0 bis 254 ▸ **Link address size** = 2 Permitted range: 0 bis 65534 Default: `-1 (invalid)` Note: Configuration of this option is not necessary if `870-101 balanced` has been selected. The current value is lost if the protocol is changed.
Name/Description	101 104	Name of the device for easy identification, a desired text. The configured name is displayed in the list of **Devices** in the `Name` column.
Short pulse	101 104	Defines the length of the pulse for the `Qualifier of Command` (à la page 219) in the 'short pulse duration' (QOC = 1) version in milliseconds. Default: `1000 ms` Valid values: `0 bis 4294967295`
Long pulse	101 104	Defines the length of the pulse for the `Qualifier of Command` (à la page 219) in the 'long pulse duration' (QOC = 2) version in milliseconds. Default: `2000 ms` Valid values: `0 to 4294967295`.

Max APDU size	104	Maximum length of APDU data as defined in the standard IEC 60870-5-104. Default: `253` Valid values: `between 25 and 253.`
Max. user data	101	Maximum length of the User Data as defined for TF1.2 in IEC60870-5-2 Section 3.2. Default: `253` Valid values: `between (19 + LAF size + COA size + COT size + IOA size) and (255).`
Commands allowed	101 104	This option allows commands from the master to be prevented on a lasting basis. The commands are not accepted if the setting is deactivated. Corresponding entries are created in the LOG file. Default: `active` Note: The commands can be dynamically activated or deactivated using a variable in zenon (T00, IOA 2) (à la page 210).
Directory for File Transfer	101 104	Defines the directory for transfer of files and directories. Clicking on the **...** button opens the file selection dialog to select a source directory. Note: The files are sent to the master.
Direcotry for File Transfer in Control Direction	101 104	Defines the directory in which the files for transfer of files and directories are stored. Clicking on the **...** button opens the file selection dialog to select a target directory. Note: These files are received by the master. All files with the NOF 255 (the temporary NOF for reverse file transfer) are ignored for the directory query.

870-101 SPECIFIC

Parameter	Protocol	Description
COA size	101	Defines the length of the COA (Common Object Address/Common Address of ASDU). Selection of address size from drop-down list: ▸ 1 octet ▸ 2 octets ▸ Note: If the **60870-5-104 (TCP/IP)** connection type is selected for link layer, the value `2 octets` is expected in accordance with the standard.

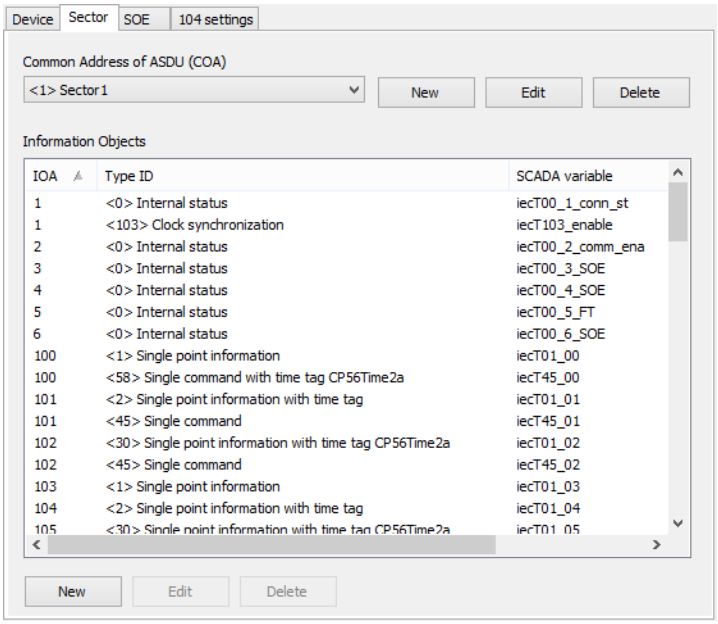
		Default: 2

COT size	101	Defines the length of the COT (Cause of Transmission). Selection of address size from drop-down list. Valid: ▸ 1 octet ▸ 2 octets ▸ Note: If the **60870-5-104 (TCP/IP)** connection type is selected for link layer, the value `2 octets` is expected in accordance with the standard. Default: `2`
IOA size	101	Defines the length of the IOA (Information Object Address). Selection of address size from drop-down list. Valid: ▸ 1 octet ▸ 2 octets ▸ 3 octets ▸ Note: If the **60870-5-104 (TCP/IP)** connection type is selected for link layer, the value `3 octets` is expected in accordance with the standard. Default: `3`
Test timeout	101	Timeout time for 870-101 connections. Entry in seconds. ▸ In `101 unbalanced`: States the time (in seconds) in which a master must execute a query to the slave. If, within this time, there is no query from the master, the slave ends the connection. ▸ In `101 balanced`: States the time (in seconds) in which the slave will execute the test procedures on the link layer. The test procedure is mandatory as prescribed by the IEC60870-5-101 standard for `balanced`. The slave can thus detect a loss of connection whilst there is no spontaneous transfer of data. Default: `20` Note: in `unbalanced Mode` , monitoring is deactivated in the event of a configured value of `0`. A value greater than 0 must be configured in `balanced Mode`. In the event of inactivity (no data to send), a test frame is sent after the time configured here. This must be confirmed by the master. If an answer is missing, the last frame (test frame or spontaneous data) is repeated. If no response to it is received, the Process Gateway restarts communication with a `Request Status of`

		Link.

10.1.4 Sector - Configuration

The **Sector** tab lists all sectors configured in the **device**. If you select a sector from the drop-down list, its Information Objects (à la page 172) (IOs) are displayed.



COMMON ADDRESS OF ASDU (COA)

You configure the sectors per **device** in the upper area. The **COA** and the name of the sector are displayed.

Configured sectors are displayed in a drop-down list. The first sector of the drop-down list is always shown by default. Configured Information Objects are shown for this sector in the area below.

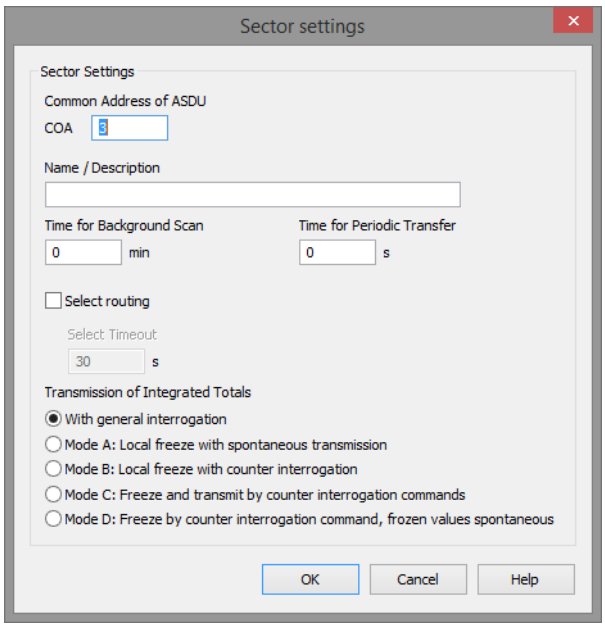
Parameter	Description
New	Opens the dialog to create a new sector (à la page 168).
Edit	Activates the dialog for editing the selected sector. Sectors that have already been configured are selected from the drop-down list. You can also launch the editor by double-clicking on the sector in the list. This button is inactive if no sector was selected.
Delete	Deletes the selected sector. This button is inactive if no sector was selected. Attention: the selected sector is deleted without a request for confirmation.

INFORMATION OBJECTS

`Information Objects (IOs)` for sectors are administered in the lower area. The `IOs` with **COA** of the selected sector are shown.

Parameter	Description
`List of the configured information objects per COA`	List of the configured IOAs for the selected COA. ▸ `IOA` Configured address of the `Information Objects`. ▸ `Type ID` Configured **Type Identification** of the `Information Object`. ▸ `SCADA variable` Linked SCADA variable of the `Information Object`. The list can be sorted by clicking on the column header. The column widths can be amended individually.
New	Activates the dialog for creating new IOs (à la page 172). This subsequent dialog is closed and a new IO is created by clicking on 'OK'. This button is inactive if no sector was selected.
Edit	Activates the dialog for editing the selected IO. You can also launch the editor by double-clicking on the IO in the list. This button is inactive if no IO was selected.
Delete	Deletes the selected information object. This button is inactive if no IO was selected.

Sector Settings



Parameter	Description
COA	The Common Address of ASDU/Common Object Address (IEC 60870-5-101 7.2.4) by which the sector is addressed. This number must be unique for each **device** (1..254).
Name/Description	Any name. For easy identification of a sector.
Time for Background scan [min]	The "Background scan" cycle in accordance with the requirements of the IEC 60870-5-101 standard, chapter 7.4.13 "Background scan". The values are sent with COT = 2 (background scan/COT_back). The time is stated in minutes. In the set time grid, there is a cyclical exchange of values for all IOs with the **Background scan** setting activated. Values are only transferred cyclically if no spontaneous value is transferred within the time set (no COT = 3). If the value is "0", cyclical transfer of values is deactivated for this sector. Default: 0
Time for periodic transfer [s]	Support of cyclical data transfer in accordance with IEC 60870-5-5 6.3. The values are sent with COT = 1 (periodic, cyclic / COT_per). The time is configured in seconds. The cyclical transfer of values is carried out for all IOs with the **Periodic data transfer** setting activated. These IOs are only transferred cyclically and not spontaneously. If the value is "0", cyclical transfer of values is deactivated for this sector. Default: 0 Note: the quickly-fluctuating measured values can unnecessarily impair spontaneous transfer. We recommend however - instead of periodic data transfer - configuring **Hysterese** for the respective IO.
Select routing	If active, a prior Select is necessary for an Execute when the command is executed. The switching commands from the master are forwarded using the **command processing** - via the auto/remote actions - to the respective Energy driver in zenon Runtime. Note: You can find further information in relation to this in the **Select routing** (à la page 217) chapter.
Select Timeout [s]	Timeout for Select before Execute in seconds. The Select process must be ended within the configured time and an Execute must be triggered. If there is no confirmation of the Select from zenon Runtime or the Select has been confirmed but the master does not send an Execute, the Process Gateway cancels the process. Note: Inactive if **Select routing**is not active. Default: 30 s

	Input range: 1 bis 65535 Note: You can find further information in relation to this in the **Select routing** (à la page 217) chapter.

TRANSMISSION OF INTEGRATED TOTALS

Selection of the transfer mode (see IEC 60870-5-101 7.4.8) for the integrated totals - the IOs of type ID:

- 15 (`M_IT_NA_1`)
- 16 (`M_IT_TA_1`)
- 37 (`M_IT_TB_1`)

No Reset is supported; an FRZ with Reset creates a warning in the LOG.

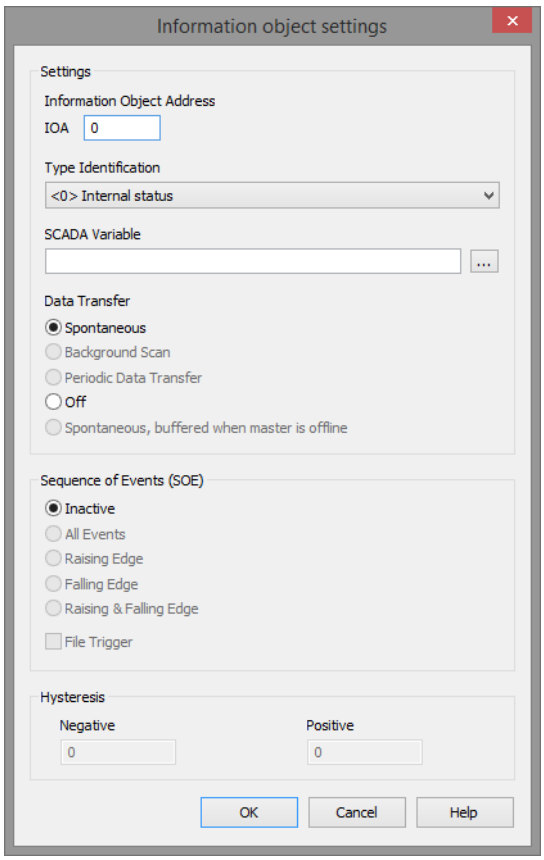
Selection from options list.
Default: `With general interrogation`

For `Mode A` to `Mode D`, IOs are requested immediately after connecting to the master of zenon. Value changes to the `Integrated Totals IOs` are administered in a separate buffer. This can buffer the greater value from (`number of the ITs IO`) * 4, or 2048 value changes.

Parameter	Description
	Integrated Totals are also transferred during a general interrogation (`C_IC_NA_1`).

Mode A: Local freeze with spontaneous transmission	Integrated totals are transferred when a value is changed.
Mode B: Local freeze with counter interrogation	Integrated totals are only transferred with a Counter Interrogation Command (`C_CI_NA_1`). To do this, a maximum of five seconds is waited after the interrogation is activated until all ITs have been supplied with values. ITs without value - the zenon variable without a value - are transferred with the qualifier `invalid`. The value changes are buffered until the interrogation is completed (freeze active). All ITs thus transfer the value of the point in time, when the last IT was supplied with a value.
Mode C: freeze and transmit by counter interrogation Transmition of integrated totals commands	Same behavior as for **Mode B**.
Mode D: freeze by counter interrogation command, frozen values spontaneous	Transfer as with mode C, mode A is activated after conclusion of the interrogation.
OK	A check is carried out to see if the COA is free for the device by pressing the **OK** button. If it is already in use you will receive an error message. If it is available, the setting is accepted and the dialog is closed.
Cancel	Discards all entries and closes the dialog.
Help	Ouvre l'aide en ligne.

Information Objects



SETTINGS

Parameter	Description
Information Object Address	Defines the address (**IOA**) under which the IO is addressed (IEC 60870-5-101 7.2.5). Default: `1` The IOA must be unique in each monitoring and controling direction per sector. Example: ▸ Valid addressing: Two IOs, both with the same IOA, a T01 (single-point - M_SP_NA_1) and a T45 (single command - C_SC_NA_1). The addressing is recommended if this single command checks precisely the single-point information. ▸ Invalid addressing: Two IOs, both with the same IOA, a T01 and a T15 (integrated totals - M_IT_NA_1). This addressing is not permitted because both type IDs belong to the monitoring direction.
Type identification	Drop-down list with the supported Type identifications (IEC 60870-5-101 7.2.1). Default: `<0> Internal status` This setting is also called type ID. It also determines the variables of the process control system that can be associated to the IO based on their data type. Note: You can also find more detailed information on type ID in the chapters Assignment of data types (à la page 179) and Interoperability (à la page 232).
SCADA variable	Defines the variable from the process control system associated with the IO. Syntax: <Project name>#<variable name> define the project connection. If there is no project reference for a variable, it is assumed that this variable is present in the start project. While entering, the system checks if the specified variable exists in the corresponding project and if it has a data type compatible with the type identification. The Button '**...**' opens the dialog for variable selection. Read more about this in the section 'Variable selection (à la page 178)'. (Scada stands for the name of the process control system.) Supported data types: `BOOL; BYTE; WORD; DWORD; LWORD; SINT; INT; DINT; LINT; USINT; UINT; UDINT; ULINT; REAL; LREAL; long double; Real number IEEE`

DATA TRANSFER

Determines the type of data transfer for the selected IO Selection from option list.
Default: `Spontaneous`

Parameter	Description
Spontaneous	Data transfer of the IOs is spontaneous - if the value of the linked SCADA variable changes.
Background Scan	IOs are transferred as a Background Scan in accordance with to the IEC 60870 standard. A data point that does not receive a new spontaneous value within the set **Time for Background Scan** is automatically sent by the gateway after the time has expired, with COT = 2. Values are only transferred if no spontaneous value (COT = 3) for this IO has yet been transferred within the time set. The setting is inactive if the background scan is locked in the sector (`Time=0`) or if the type ID used is not suitable for a background scan in accordance with the protocol. Permitted type IDs: ▸ `M_SP_NA_1` and `M_SP_TA_1`, `M_SP_TB_1` - for `T01`, ▸ `M_DP_NA_1` and `M_DP_TA_1`, `M_DP_TB_1` - for `T03`, ▸ `M_ST_NA_1` and `M_ST_TA_1`, `M_ST_TB_1` - for `T05`, ▸ `M_BO_NA_1` and `M_BO_TA_1`, `M_BO_TB_1` - for `T07`, ▸ `M_ME_NA_1` and `M_ME_TA_1`, `M_ME_TD_1` - for `T09`, ▸ `M_ME_NB_1` and `M_ME_TB_1`, `M_ME_TE_1` - for `T11`, ▸ `M_ME_NC_1` and `M_ME_TC_1`, `M_ME_TF_1` - for `T13` Note: The type IDs with time stamp (for example `M_ME_TA_1`) is mapped to the corresponding TID without time stamp (for example `M_ME_NA_1`) in the process of sending the background scan. The setting is not available for any other type ID.
Periodic Data Transfer	Periodic data transfer in accordance with the IEC 60870-5-5 Standard, Section 6.3. IOs are transferred cyclically and not spontaneously with COT = 1. The setting is inactive if periodic data transfer is locked for the sector (**Time for periodic transfer**=0) or if the type ID used does not support periodic data transfer. Only the type ID=9, 11 and 13 support a periodic transfer - `M_ME_NA_1`, `M_ME_NB_1`, `M_ME_NC_1` - "measured values" without time stamp. The setting is not available for any other type ID. Note: the quickly-fluctuating measured values can unnecessarily impair spontaneous transfer. We recommend however - instead of periodic data transfer - configuring **Hysterese** for the respective IO.
Off	Changes in value are not transferred. No value is sent for the IO during a

	general interrogation (GI).
Spontaneous, buffered when master is offline	When a connection fails, the data is buffered until the master connects again. All buffered data is sent with the status COT_spont (3). The last buffered value is sent with the status COT_inrogen (20) again. Buffering can be activated for the type IDs <30..37>: `M_SP_TB_1, M_DP_TB_1, M_ST_TB_1, M_BO_TB_1, M_ME_TD_1, M_ME_TE_1, M_ME_TF_1 und M_IT_TB_1` Note: Only type IDs that have a time stamp with a date are permitted as Events ("time tag CP56Time2a"). The setting is not available for any other type ID.

SEQUENCES OF EVENTS (SOE)

These settings define the IO-specific behavior for SOE. The saving of the SOE files and the transfer thereof (see IEC standard 60870-5-101 7.4.11.3.1) can only be activated for the type IDs <30..37>:

`M_SP_TB_1`, `M_DP_TB_1`, `M_ST_TB_1`, `M_BO_TB_1`, `M_ME_TD_1`, `M_ME_TE_1`, `M_ME_TF_1` und `M_IT_TB_1`.

Default: `Inactive`

Note: Only type IDs that have a time stamp with a date are permitted as Events ("time tag CP56Time2a").

Parameter	Description
Inactive	Data point is not used for SOE description. Is automatically used for all type IDs that do not support SOE.
All events	Each value or status change triggers a write in the open file. The setting is blocked if the type ID if not planned for the SOE function.
Raising Edge	If the value changes from 0 to <> 0, then it triggers the write mode in the open file. The setting is blocked if the type ID if not planned for the SOE function.
Falling Edge	If the value changes from <> 0 to 0, then it triggers the write mode in the open file. The setting is blocked if the type ID if not planned for the SOE function.
Raising + Falling Edge	A change of value from 0 to <> 0 or from <> 0 to 0 triggers a write in the open file. The setting is blocked if the type ID if not planned for the SOE function.
File trigger	If a value is to be written due to the settings previously described, a new file is created and opened for the activated file, provided this has not yet been opened. The trigger event itself is also written to the SOE file. The setting is blocked if the type ID if not planned for the SOE function. Default: `Inactive`

HYSTERESIS

Value changes that are within the the hysteresis are ignored - not transferred to the master.

Hysteresis is applicable for the permitted **TID** ("measured values") and only if these have been configured for spontaneous data transfer.

A transfer is triggered:

- If `Type Identification` allows hysteresis.
  The `Type Identifications <9> to <14>` and `<34> to <36>` are permitted.
- If **Data Transfer** = `spontaneous`
- With a value change that acts >= the hysteresis (according to the direction).

And notwithstanding the hysteresis:

- Change of the quality (status bits of the variable) always triggers the transfer;
- The current value is always sent to a GI (`C_IC_NA_1`) and a read (`C_RD_NA_1`) command.

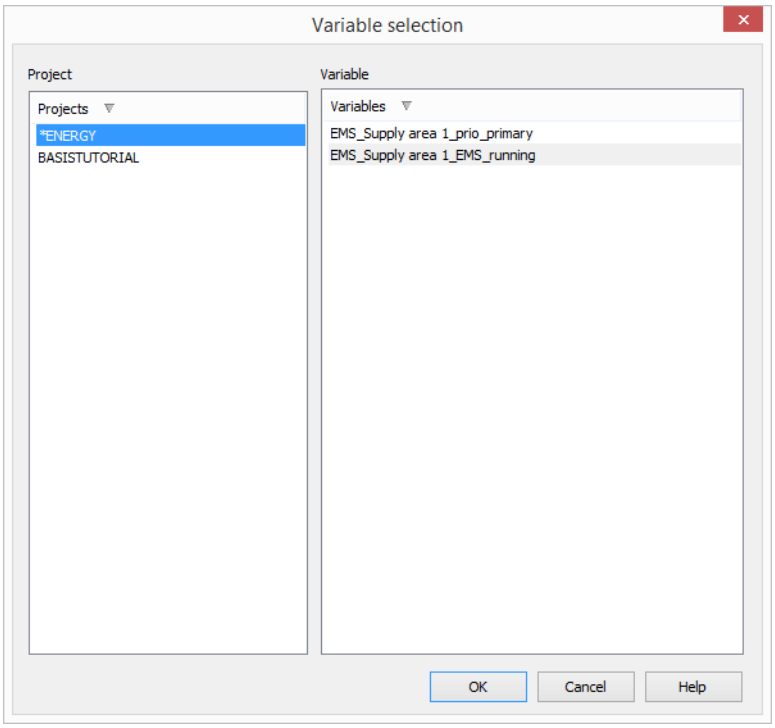
Parameter	Description
Negative	Negative hysteresis, starting from **measuring range** of the variable. Default: 0 Note: not active if type identification is not <9> to <14> or <34> to <36>. This field is also inactive if the value **Data Transfer** is not spontaneous. Erroneous and meaningless entries (such as text input or incorrect hysteresis values) are ignored and must be corrected by the engineer of the project.
Positive	Positive hysteresis, starting from **measuring range** of the variable. Default: 0 Note: not active if type identification is not <9> to <14> or <34> to <36>. This field is also inactive if the value **Data Transfer** is not spontaneous. Erroneous and meaningless entries (such as text input or incorrect hysteresis values) are ignored and must be corrected by the engineer of the project.
OK	Your settings are verified after pressing the button "OK". If they are valid, they are accepted and the dialog is closed.
Cancel	Discards all entries and closes the dialog.
Help	Ouvre l'aide en ligne.

Informations

*Communication for **Spontaneous**data transfer, **Background Scan** and **Periodic Data Transfer** only starts after the first general query received (C_IC_NA_1). For background scan and periodic data transfer, the start of the transfer cycle is calculated per IO from the time of the answer to the general query.*

Variable selection dialog

This dialog lists all available projects and their variables.



Parameter	Description
Projekte	List of the projects started in zenon Runtime. A star (*) marks the default project.
Variablen	The list of variables is filtered. Only variables that can be used with the selected **Type identification** for the IO are displayed. Selecting a project updates the list of variables.
OK	Assumes the selected variable. Alternatively, the variable can be assumed by double clicking. When loading, the currently configured project and the variable will be selected. The same variable can be used several times: for several IOs in the same sector or other sectors, for the same device or another one. If the dialog remains open and there is a reload in the Runtime, all lists will be emptied and refilled after the reload is finished.
Cancel	Discards all entries and closes the dialog.
Help	Ouvre l'aide en ligne.

Mapping of the data types

The zenon data types are compared to the IEC 60870 data types in the following table.

Process Gateway		zenon	
IEC 60870-5-101 ASDU	**Type identification**	**Data type**	**Comment**
M_SP_NA_1	1	BOOL	
M_SP_TA_1	2	BOOL	
M_SP_TB_1	30	BOOL	
M_DP_NA_1	3	USINT	0..3, with DPI/DCS Mapping
M_DP_TA_1	4	USINT	0..3, with DPI/DCS Mapping
M_DP_TB_1	31	USINT	0..3, with DPI/DCS Mapping
M_ST_NA_1	5	USINT	Corresponds to complete VTI (IEC60870-5-101 7.2.6.5); the highest bit in USINT is thus the Transient bit.
M_ST_TA_1	6	USINT	Corresponds to complete VTI (IEC60870-5-101 7.2.6.5); the highest bit in USINT is thus the Transient bit.
M_ST_TB_1	32	USINT	Corresponds to complete VTI (IEC60870-5-101 7.2.6.5); the highest bit in USINT is thus the Transient bit.
M_BO_NA_1	7	UDINT	
M_BO_TA_1	8	UDINT	
M_BO_TB_1	33	UDINT	
M_ME_NA_1	9	REAL	The value range in accordance with the IEC 60870 standard is <−1 .. 1−2^{-15}> with a precision of 4 decimal places. With zenon variables, this corresponds to a value range of −1.0 to 0.9999. Outside of the range for normalized values, in ASDU the value of the IOs is amended and the OV quality bit is set.
M_ME_TA_1	10	REAL	The value range in accordance with the IEC 60870 standard is <−1 .. 1−2^{-15}> with a precision of 4 decimal places. With zenon variables, this corresponds to a value range of −1.0 to 0.9999. Outside of the range for normalized values, in ASDU the value of the IOs is amended and the OV quality bit is set.
M_ME_TD_1	34	REAL	The value range in accordance with the IEC 60870 standard is <−1 .. 1−2^{-15}> with a precision of 4 decimal places. With zenon variables, this corresponds to a value range of −1.0 to 0.9999. Outside of the range for normalized values, in ASDU the value of the IOs is amended and the OV quality bit is set.

M_ME_NB_1	11	INT	
M_ME_TB_1	12	INT	
M_ME_TE_1	35	INT	
M_ME_NC_1	13	REAL	
M_ME_TC_1	14	REAL	
M_ME_TF_1	36	REAL	
M_IT_NA_1	15	DINT	
M_IT_TA_1	16	DINT	
M_IT_TB_1	37	DINT	
M_EI_NA_1	70	USINT	A value change from 126 to 127 causes End of initialization (à la page 208) to be sent to the master
C_SC_NA_1	45	BOOL	The variable reflects the value of the command that has been received from the 870 master and the status bits of the variable reflect the process of the command. Example: COTx.
C_SC_TA_1	58	BOOL	The variable reflects the value of the command that has been received from the 870 master and the status bits of the variable reflect the process of the command. Example: COTx.
C_DC_NA_1	46	USINT	The variable reflects the value of the command that has been received from the 870 master and the status bits of the variable reflect the process of the command. Example: COTx. 0 or 1, with DPI/DCS Mapping
C_DC_TA_1	59	USINT	The variable reflects the value of the command that has been received from the 870 master and the status bits of the variable reflect the process of the command. Example: COTx. 0 or 1, with DPI/DCS Mapping
C_RC_NA_1	47	USINT	The variable reflects the value of the command that has been received from the 870 master and the status bits of the variable reflect the process of the command. Example: COTx. 0 or 1, with DPI/DCS Mapping
C_RC_TA_1	60	USINT	The variable reflects the value of the command that has been received from the 870 master and the status bits of the variable reflect the process of the command. Example: COTx.

			0 or 1, with DPI/DCS Mapping
C_SE_NA_1	48	REAL	The variable reflects the value of the command that has been received from the 870 master and the status bits of the variable reflect the process of the command. Example: COTx.
C_SE_TA_1	61	REAL	The variable reflects the value of the command that has been received from the 870 master and the status bits of the variable reflect the process of the command. Example: COTx.
C_SE_NB_1	49	INT	The variable reflects the value of the command that has been received from the 870 master and the status bits of the variable reflect the process of the command. Example: COTx.
C_SE_TB_1	62	INT	The variable reflects the value of the command that has been received from the 870 master and the status bits of the variable reflect the process of the command. Example: COTx.
C_SE_NC_1	50	REAL	The variable reflects the value of the command that has been received from the 870 master and the status bits of the variable reflect the process of the command. Example: COTx.
C_SE_TC_1	63	REAL	The variable reflects the value of the command that has been received from the 870 master and the status bits of the variable reflect the process of the command. Example: COTx.
C_BO_NA_1	51	UDINT	The variable reflects the value of the command that has been received from the 870 master and the status bits of the variable reflect the process of the command. Example: COTx.
C_BO_TA_1	64	UDINT	The variable reflects the value of the command that has been received from the 870 master and the status bits of the variable reflect the process of the command. Example: COTx.
C_IC_NA_1	100	-	The command is executed automatically in the background. All ASDUs envisaged by the IEC 60870 standard are sent to the 870 master
C_CI_NA_1	101	-	The command is executed automatically in the background. All ASDUs envisaged by the IEC 60870 standard are sent to the 870 master
C_RD_NA_1	102	-	The command is executed automatically in the background. All ASDUs envisaged by the IEC 60870 standard are sent to the 870 master

C_CS_NA_1	103	BOOL	Enable variable of the time synchronization
C_TS_NA_1	104	-	The command is executed automatically in the background. All ASDUs envisaged by the IEC 60870 standard are sent to the 870 master
C_TS_TA_1	107	-	The command is executed automatically in the background. All ASDUs envisaged by the IEC 60870 standard are sent to the 870 master
C_RP_NA_1	105	USINT	The variable reflects the value QRP of the command that has been received from the 870 master.
F_DR_TA_1	126	USINT	The variable of the triggering of the spontaneous directory transfer (à la page 221)

AMENDMENT OF IO VALUES

It must be possible to link an IO (with a certain type ID) to a variable whose value exceeds the value range of the type ID of the IO.

Example: An IO with Type ID = 1 - single-point information was linked to a UDINT variable */stVal[ST] from the IEC850 driver.

In this case, the Process Gateway will amend the values for IO and also set the quality bits in ASDU, as envisaged in the IEC 60870 standard.

ASDU type ID	zenon variable value	Value and quality in ASDU
M_SP single-point	< 0 > 1	0 + IV 1 + IV
M_DP double-point	< 0 > 3	11b + IV 11b + IV
M_ST step position	< -64 > 255	x40 + OV xFF + OV
M_BO bitstring 32-bits	< 0 > xFFFFFFFF	0 + OV xFFFF FFFF + OV
M_ME_xA normalized	< -1.0 >= 1.0	x8000 + OV x7FFF + OV
M_ME_xB scaled	< -2^{15} > $2^{15}-1$	x8000 + OV x7FFF + OV
M_ME_xC short floating	< -MAX Float > MAX Float	-MAX Float + OV MAX Float + OV
M_IT integrated totals	< -2^{31} > $2^{32}-1$	x8000 0000 + IV x7FFF FFFF + IV

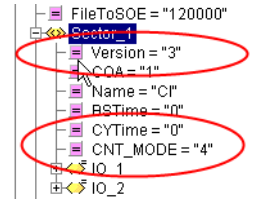
Informations

If the zenon variable currently has the status bit INVALID or the status bit OV_870 set, the value of the IO is also sent with Quality IV or Quality OV .

You can find further information in relation to this in the "Mapping of status bits in message direction (à la page 215)" chapter.

Sector

XML FIELDS



CNT_MODE	0
ToSbe	45000
SBE	1

(This tag has no subtags.)

Parameter	Description
Version	The CNT_MODE field is expected from version 3 onwards.
CNT_MODE	Configured mode of integrated totals IOs.
Sector	The new items ToSBe and ob SBE are added for the sector: ▸ ToSBe: Timeout [ms] for Select before execute: 1000 - 65535000 ▸ SBE: 0 => not required 1 => necessary for execute

CONFIGURED MODE

Parameter	Description
0	With general interrogation. (Default)
1	Mode A
2	Mode B
3	Mode C
4	Mode D

ERROR CODES

Code	Description
13	Advise for integrated totals fail.
14	Queue overrun. Integrated totals values lost.
16	Advise failed for command variable
17	List of value changes for command variables with SBE too long. Value change has been lost. 2048 value changes are buffered as a minimum, or four times the number of active Select Before Execute (SBE) - depending on which value is greater.
572	Counter interrogation. Unsupported QCC.
573	Counter interrogation. Unexpected IOA.
574	Counter interrogation. Unsupported COT.
575	Buffer for integrated totals interrogation data too small.

WARNINGS

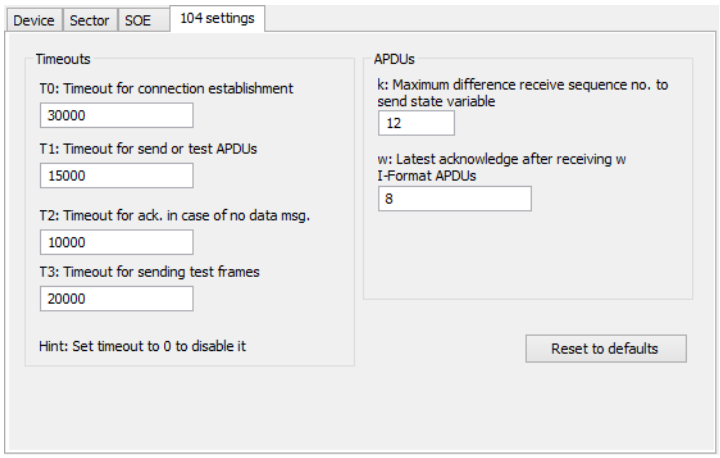
Code	Description
1011	QCC FRZ is only partly supported.
1012	Counter interrogation already active.
1014	Activation confirmation already confirmed.

ERROR MESSAGE

Message	Description
Unsupported qualifier of counter interrogation. <command>	Unknown QCC. RQT is only supported with 5. For `C_CI_NA_1` (`Counter interrogation`) it is the case that `RQT != 5` is treated the same as `RQT = 5`.
Counter interrogation processed without reset. < command>	Warning. Reset is not supported.
Can't advise IT data point. Device:(<device>)<device> Sector:(<COA>)<Sektorname> adv:0x<Advise-ID>	Data points for integrated totals could not be requested.
IT Values lost, queue full! Device:(<device>)<device> Sector:(<COA>)<Sektorname> <Wertänderung>	The buffer for the value changes of IT IOs is full. The value change is lost.

10.1.5 104 settings

In the **104 settings** tab, you set the timeout times of the Link Layers and the number of APDUs . The settings should be the same for slave and master. If a master overwrites a timeout or the number of frames to be confirmed, the slave closes the connection.



Note: The input fields of this tab are only available if 870-104 has been selected as a protocol.

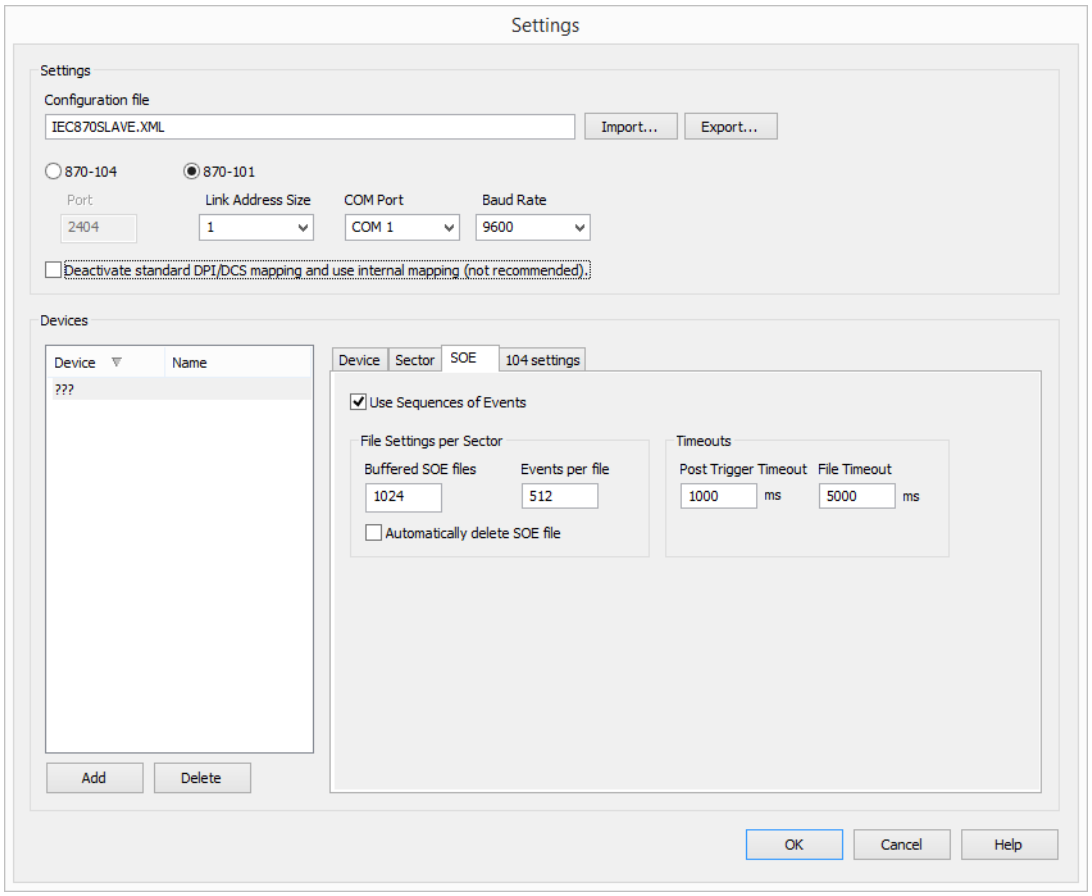
Parameter	Description
Timeouts	Timeout settings of the connection
T0: time-out of connection establishment	Timeout for establishing connection. Value range 0 - 4294967295 Default: 30000 There should also be at least one u-frame from the master in that time.
T1: Time-out send or test APDUs	Timeout for frame confirmation by the master. Value range: 0 - 4294967295 Default: 15000
T2: Time-out for ack. in case of no data msg	Timeout, within which the master should confirm if no data is exchanged. Value range: 0 - 4294967295 Default: 10000
T3: Time-out for sending test frames	Time after which a U-frame is sent to the master if no data is to be transferred. Value range: 0 - 4294967295 Default: 20000
APDUs	*Settings for the ISO 7816 standard: Application Protocol Data Unit (APDU).*
k: Maximum difference receive sequence nr to send state	Number of maximum I-frames not yet confirmed by the master. Value range: 0 - 4294967295 Default: 12
w: Latest acknowledge after receiving w I-Format APDUs	Number of I-frames received after a confirmation is sent. Value range: 0 - 4294967295 Default: 8

Informations

*The **Reset to default** button resets all input to the values prescribed by the IEC 60870-5-104 standard.*

10.1.6 DPI / DCS mapping

It is possible to configure whether DPI/DCS (`double-point information`/`double command state`) should be transferred in accordance with the IEC 60870 standard or in the zenon internal display. The default is transfer between Process Gateway and an 870 master in accordance with the standard.



Setting	Functionality
Deactivate standard	A setting for compatibility when converting projects from very early versions of zenon or use with zenon drivers for which the **Mapping** of **Double Point Values** has also been deactivated. Default: `Inactive`. `Inactive`: Values for DPI/DCS (and RCS) are mapped as follows, which also guarantees the correct behavior for the **Command Processing** module as well as for communication with an 870 master: ▸ zenon value `0` <-> DPI/DCS/RCS value `1` (= `OFF/LOWER`) ▸ zenon value `1` <-> DPI/DCS/RCS value `2` (= `ON/HIGHER`) ▸ zenon value `2` -> DPI value `0` (`intermediate`) ▸ zenon value `3` -> DPI value `3` (`failure/error`) DPI is "double-point information" IOs; DCS is "double command" IOs; RCS is "regulating step command"; Mapping from zenon version 8.00 `Active`: Mapping is deactivated. Values for DPI/DCS/RCS correspond directly to the values of zenon. Not recommended. Note: For compatibility reasons, this setting is also activated if an XML file is read with the value `1` in the `Version` field in the configuration of the Process Gateway.

10.1.7 Configuration via XML file

The configuration of the IEC870SI is stored in an XML file. The active XML file is read when the gateway is started. The active configuration is stated in the INI file (à la page 19), for example:
[IEC 870]
SETTINGS=`IEC870SLAVE.XML`

You can choose any name. The path is the same as in the INI file.

Note: You can import and export different configurations with the buttons **'Export'** and **'Import'** in the main window.

STRUCTURE OF THE XML FILE

The fields in the XML file correspond to settings that have been approved in the respective configuration dialogs.

The values and structure of the XML file are verified when the file is read. Erroneous entries trigger an error message that is also logged in the LOG - **Diagnosis Viewer** .

Attention

XML entries are case sensitive. Pay attention to use correct upper and lower cases.

Name	Property
process_gateway_IEC870Slave	The root node must have this name. Otherwise, the reading process will stop with an error.
Version	Contains the current version of attributes and nodes.
Is101active	Data Layer Selection: ▸ 0: 870-104 ▸ 1: 870-101 unbalanced ▸ 2: 870-101 balanced
DeActDPIStd	DPI /DCS compliant (IEC60870 compliant) data transfer or internal control system format

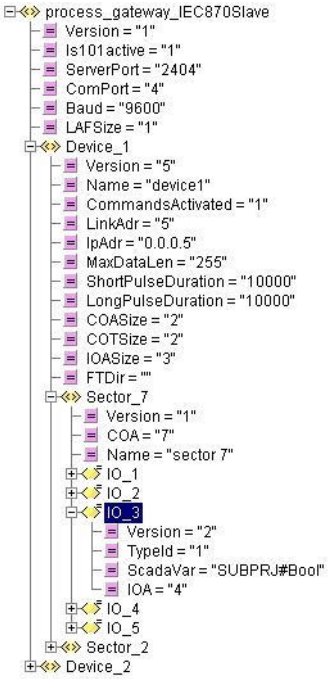
Device_x	Nodes for the device configuration. When saving, X is replaced by a consecutive number, starting with 1. Note: No particular node naming is required during import. A **Device** is created for each node.
Version	From 6 describes an XML file with the fields for SOE support
LinkAdr	In protocol 101, the `Link Address` is set here. It must be unique for all devices. For protocol 104, this entry has no meaning.
IpAdr	In protocol 104, the IP address of the Master is entered here. It must be unique for all devices. For protocol 101, this entry has no meaning.
EnableSOE	▸ 0 = inactive ▸ 1 = SOE active
MaxBufferedFilesSOE	The maximum number of buffered SOE files.
AutodeleteFilesSOE	▸ 0 = no Autodelete ▸ 1 = autodelete if the maximum number of the SOE files has been reached
MaxEventsPerFileSOE	Maximum number of ASDU per SOE file
PostTrgToSOE	**Post Trigger Timeout** in [ms]
FileToSOE	**File Timeout** in [ms]

Sector_x	Nodes for the configuration of a sector. X is replaced by the **COA** when saving. Note: No particular node naming is required during import. A sector is created for every node.
Version	From 4 marks an XML file with the fields for **Select Routing** support.
COA	`Common Address of ASDU` (`Common Object Address`) Must be unique for every device per sector.
BSTime	Periods for `Background Scan`. Unit of measurement: ms. 0 deactivates the background scan. Default: 0
CYTime	Periods for cyclical data transfer Unit of measurement: ms. 0 deactivates the cyclical transfer. Default: 0 Available in version 2 and above

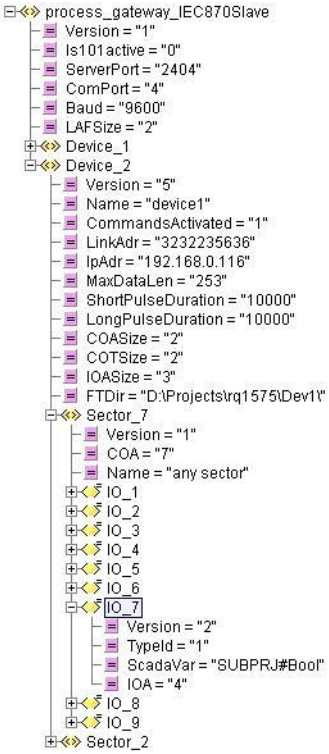
IO_x	Nodes for the configuration of an `Information Object`. When saving, X is replaced by a consecutive number, starting with 1. Note: No particular node naming is required during import. A `IO`is created for each node.
Version	▸ From 4 marks an XML file with the fields for SOE support; ▸ From 5 marks an XML file with the fields for support of hysteresis;
TypeId	`Type Identification` When loading, a check is carried out to see whether the **Type ID** is possible with the data type of the given variables.
ScadaVar	Name of the zenon variable, syntax: [ScadaProjectName#]ScadaVariableName. Variables without a project reference are searched for in the default project (INI file - [GENERAL] section: PROJECT=). When the XML file is loaded, the existence of the specified variable in the project is verified.
IOA	`Information Object Address` In combination with the **TypeId**, this must be unique per sector and the direction (controlling/monitoring) must be unique.
KindOfTransfer	States the type of data transfer for `IO` in monitoring direction . ▸ 0 spontaneous (`COT`=3) ▸ 1 `Background scan` (`COT`=2) ▸ 2 periodic/cyclical (`COT`=1) ▸ 3 off; no transfer to the 870 master ▸ 4 spontaneous transfer with buffering in the event of a communication failure Default: 0 Available in version 3 and above
TrgSOE	▸ 0 = inactive ▸ 1 = the `IO` acts as a trigger for SOE
EvDetectionSOE	Determines when a change to the value or status bit of the `IO` can trigger a write to the SOE file. ▸ 0 SOE active ▸ 1 Saves all changes. ▸ 2 Raising edges ▸ 3 Falling edges ▸ 4 Raising and falling edges

HysteresisNegative	Value of the negative hysteresis.
HysteresisPositive	Value of the positive hysteresis.

Example configuration for the protocol 870-101:



Example configuration for the protocol 870-104:



.CSV export and import

Configurations of the Process Gateways can be exported into a text file or imported into Process Gateway as a text file.

The first line in the CSV file must be a header with the names of the properties. The individual properties must be separated with a TAB.

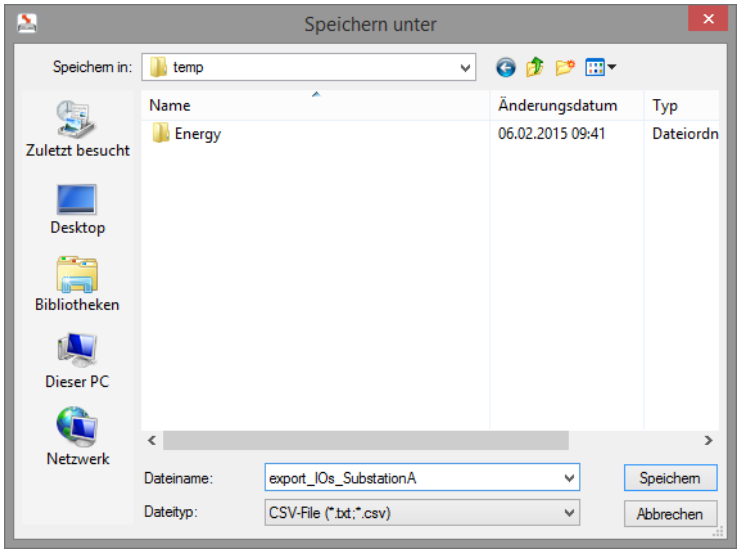
The following information of the IO must be saved per line of the CSV file:

- LinkAdr or IpAdr of the device
- COA of the sector
- All settings of the IO

New IOs are typically created in the Process Gateway during import.
In addition, the `KindOfTransfer` property can be used during import to update or delete pre-existing IOs of the Process Gateway configuration.

EXPORT

Carry out the following steps in order to export a Process Gateway configuration:



1. Click on the **Export...** button in the configuration dialog.
   The file selection dialog is opened.
2. Select the `CSV-File` entry in the **File Type** drop-down list.
3. Name the export file in the **Filename** input field
4. Click on **Save**.
   The file is exported and saved at the selected save location.

IMPORT

The configuration of the Process Gateway can be created with a .CSV file or supplemented with a .CSV file. The .CSV file can come from another process gateway or be created with an external program (such as MS Excel).

Carry out the following steps to import a .CSV file into the configuration of the Process Gateway:

1. Click on the **Import...** button in the configuration dialog.
   The file selection dialog is opened.
2. Select the `CSV-File` entry in the **File Type** drop-down list.
3. Select the file.
4. Click on Open.
   The file is loaded and the configuration of the Process Gateway is updated.

RULES FOR IMPORT

The following rules are applicable for import and amendment:

1. `LinkAdr` or `IpAdr`:

- The import of a `LinkAdr` is only permitted if the Process Gateway has already been configured for 870-101.
- The import of a `IpAdr` is only permitted if the Process Gateway has already been configured for 870-104.

  The following is applicable in the event of a conflict between protocol and addressing:

  - If no device has yet been created in the Process Gateway, the import process amends the protocol setting accordingly and creates a device in the Process Gateway. This device is configured with default settings.
  - If there is already a device present in the Process Gateway (for a different protocol), the import is canceled with an error message.

1. Sector

- If no sector has yet been configured for the COA of the import file to be imported, the import process creates a new sector.
- The IO is then imported into this sector.

1. IO
   The `KindOfTransfer` column determines the editing layer of the IO:

- 0...4 - normal value range for `KindOfTransfer`
  Creates new IO
- 100...104
  Deletes the IO from the sector that has precisely `IOA` and `TypeID`
- 300...304
  Deletes the IO from the sector that is linked to `ScadaVar`.
- 200...204
  Updates the IO that has precisely IOA and its `TypeID` is in the appropriate area.
  Areas are applicable:

  - 1...37
    Monitoring Direction
  - 45...126
    Controlling Direction
  - 0
    for internal TypeIDs
- 400...404
  Updates IO that is linked to `ScadaVar`.

After this editing step, `KindOfTransfer` is then converted to the normal value range.

Exemple

In the Process Gateway, an **IO** with `COA=1`, `IOA=1` and `TypeID=1` was configured for spontaneous transfer, and linked to the variable with the name `Var1`.

The import file contains, for variable `Var1`, a line with `COA=1`, `IOA=100`, `TypeID=30` and `KindOfTransfer=400`. As a result, for variable `Var1`, the `TypeID` of the **IO** is changed from `1` to `30` and `IOA` is changed from `1` to `100`.

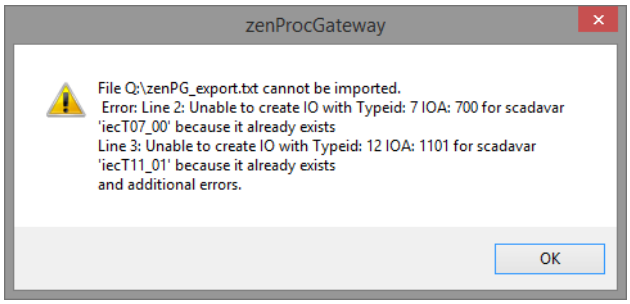
The **Data Transfer** remains spontaneous.

ERROR TREATMENT

If an error is detected during the import process, an entry is created in the LOG file. In addition, a corresponding warning dialog is shown in the Process Gateway at the end of the import.

Possible error causes:

- An IO is not yet present in a sector that has already been configured, but envisaged for updating in the file with `KindOfTransfer` .
- The import does not allow more than one IO with a certain `IOA` to be created in the respective `TypeID` area.
- The creation of IOs for variables that are not present in the project or that have a non-numerical data type is considered an error.



The header is applicable as a line with number 1.

10.2 Sequence of Events (SOE)

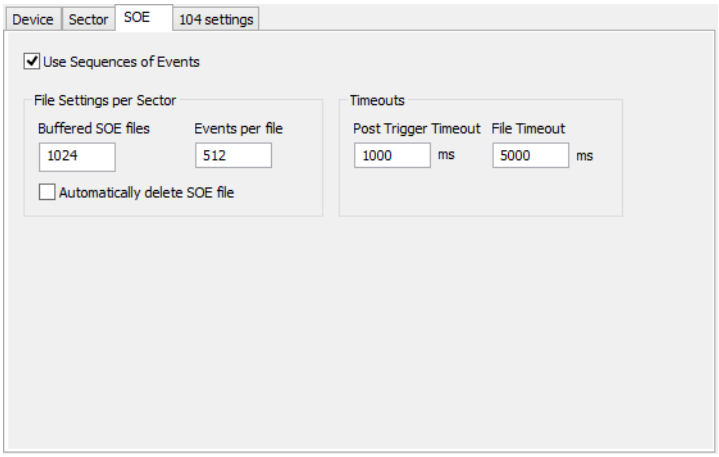
The **Process Gateway IEC870 Slave** supports the automatic recording and transfer of SOE files - in accordance with procedures defined in IEC 60870-5-101, Section 7.4.11.3 "Transmission of sequences of events" (in a section of a data file).

In the sector, at least one IO must have the **File Trigger** option activated, otherwise no SOE files are created.

Note:You can find the **File Trigger** option in the **Sector** tab - **Information Objects** (à la page 172), in the **Sequence of Events (SOE)** area.

The SOE files are saved in the folder for file transfer. A folder for file transfer must therefore be configured in **Device** (**Directory for File Transfer** option).

SETTINGS IN THE SOE TAB



Parameter	Description
Use sequences of events	If active, the SOE for the selected device is activated. If `inactive`, all SOE settings to the IOs are ignored. Default: `inactive` Note: The IEC870 slave only saves, in SOE files, events of the IOs for which the SOE has been activated in the **Sector** - **Information Objects** (à la page 172) tab.

FILE SETTINGS PER SECTOR

Parameter	Description
Buffered SOE Files	If the number set for SOE files in the folder is reached for the sector, a spontaneous directory transfer (à la page 221) (type ID `F_DT_TA_1`) is triggered. If the number of files in the folder is doubled, the oldest file will

	start to be deleted (ring buffer functionality). Value range 1 to 1024. Default: 1024 Attention:A directory - but not file transfer - is triggered automatically. The SOE files must be requested by the 870 master. Note: The current status of the SOE can be evaluated using variables that are linked to the internal type T00 (à la page 210).
Events per File	Number of value changes that are written in a file before it is closed. Value range: 1 to 4096. Default: 512
Automatically delete SOE file	If active, the reaction is that this is then automatically closed for type ID "F_AF_NA_1" (confirmation for file) for an SOE file.

TIMEOUTS

Parameter	Description
Post trigger timeout	If no ASDU is written to the file within this period of time, this is then closed. The time period restarts with each ASDU written. Value range: 100ms to 120000ms Default: 1000ms
File timeout	Maximum time for which the file remains open (regardless of the number of events). Value range: 100ms to 120000ms Default: 5000ms

FIRST VALUE

The exchange of data between IEC870Sl and zenon Runtime for SOE starts immediately when the gateway has finished installing. The first value that is initialized does not trigger an Event and is thus also not entered into the file. The same applies if Runtime is ready again to exchange data after reloading.

The exchange of data for data in Monitoring direction only starts when a master has connected and triggered a general interrogation (GI) (à la page 205).

FILE TRANSFER

As part of the SOE functionality for transparent (1), event (3) as NOF is also supported as NOF.

There is no automatic File transfer if the maximum number of SOE files to be buffered has been exceeded.

In the current version, the gateway supports spontaneous directory transfer, but not file transfer activated by the slave. The SOE files must be requested by the 870 master.

Note: the current status of the SOE can be evaluated using variables that are linked to internal type T00 (à la page 210).

DIRECTORY TRANSFER

In accordance with the IEC 60870 standard, an automatic spontaneous directory transfer (à la page 221) is activated for the following reasons with existing SOE files:

- 24 hours after the last automatic directory transfer.
- if the master connects.
- if the maximum number of files to be buffered is exceeded.

10.2.1 Warnings

Warning message	Meaning
`Sequences of events for device <Devicename> sector <Sectorname> are activated but no trigger defined`	The device has SOE activated, on the server there is an IO with SOE active but the trigger was not activated for any IO. No file is created without a trigger. At least one IO must have the trigger activated for the sector.
`SOE value change without trigger and file is closed. Value lost! Device:(<Device ID>)<Device-Name> Sector:(<COA>)<Sektorname> <Wertinformation>`	A value change was not written because the SOE file is not open. Code:1010

10.2.2 Error message

Error message	Meaning
Sequence of events active but no directory for the file transfer defined!	SOE files are archived in the file transfer folder. Therefore a folder for file transfer must be present.
SOE Values lost, queue full! Device:(<Device ID>)<Device-Name> Sector:(<COA>)<Sektorname> <Wertinformation>	for IOs that have a SOE activated, data is exchanged to the stack via a queue. This can assume a value of 2048 or 4 * the number of SOE IO's in sector entries. If this number is exceeded, this error message is created and the value change is lost. Error code: 6
Can't create subdirectory for file transfer. Device:(<Device ID>)<Devicename> Sector:(<COA>)<Sektorname> system error:<Systemfehlercode>	The subfolder for the sector is automatically created for SOE in the folder for file transfers. This error shows that this folder could not be created. SOE then indicates that this sector is NOT available. Error code: 10
Systemfehlercode	Contains the error code returned from the operating system.
Can't advise SOE data point. Device:(<Device ID>)<Devicename> Sector:(<COA>)<Sektorname> adv:0x<Advise-ID>	Data point request for SOE IOs failed. SOE then indicates that this sector is NOT available. Error code: 11
Write asdu to file <Dateiname> fail! Device:(<Device ID>)<Devicename> Sector:(<COA>)<Sektorname> <Value> system error: <Systemfehlercode>	ASDU write to SOE file failed. Error code: 8
Illegal file name <Dateiname> detected! Device:(<Device ID>)<Devicename> Sector:(<COA>)<Sektorname>	Filenames for the file transfer have a defined format <IOA>.<NOF>. The file does not adhere to this convention. Error code: 12
Auto delete for file failed! System error: <Systemfehlercode> <Dateitransferinformation>	Error code: 9
Dateitransferinformation	Contains the information that identifies the file transfer.

10.3 Communication

This chapter explains details on communication in the **Process Gateway**:

Informations

If an **ASDU** with unknown `Type-ID` is received by the **Process Gateway**, a negative response of `COT 44` is sent to the 870 master.

10.3.1 Start

It loads the INI file when the Process Gateway is started. The **Process Gateway** then attempts to read its active configuration from the XML file - entered in the INI file. If this is not possible, an error message is displayed and the configuration dialog is opened.

If the configuration dialog is not left by clicking on the **OK** button, the **Process Gateway** is closed.

Attention

*No configuration dialog is called up if the **Process Gateway** is started invisibly. The Process Gateway will automatically quit if its configuration is invalid or missing.*

After the IEC870 slave has successfully read its configuration from an XML file, it expects the establishment of a connection from a master. The IEC 60870 standard defines that a session between a master and slave must always start with a general query (GI - `C_IC_NA_1`). This is why the receipt of the GI from the master is a requirement for spontaneous communication. First the general query triggers the establishment of the connection to zenon Runtime in the IEC870 slave and the request for the current values of the variables for IOs.

Note: if a master opens the connection to the link layer but does not send a general query, you can attempt to provoke the GI by sending an ASDU T70 - End of Initialization (à la page 208).

10.3.2 General interrogation

The general query (GI) triggers an establishment of the connection to zenon Runtime in the IEC870 slave. The receipt of the GI - `C_IC_NA_1` - from a master is the requirement for spontaneous communication, `Background scan` or `Periodic data transfer`. As long as the IEC870 slave has not received the GI, it does not send any values to the master.

Attention

In accordance with the IEC60870 standard, a master must start communication with a general query. For this reason, the slave closes the connection if the master has not sent a GI in the envisaged time (`(T0)`).

Note: if a master opens the connection to the link layer but does not send a general query, you can attempt to provoke the GI by sending an ASDU T70 - End of Initialization (à la page 208).

After receipt of the GI, the connection to zenon Runtime is set up in sectors (**COA**). If the connection cannot be established in Runtime, there is a negative confirmation of the general query (`COT_actcon + PN`). The query that is already active for the sector is ended and a warning is logged.

A GI for a sector that does not exist in the slave (unknown **COA**) is confirmed in the negative (**COT=**`46 + PN`) and the connection to the master is disconnected. This is an indicator for a configuration error in the system.

Other system configuration errors are handled with more tolerance. The receipt of invalid values is logged in the LOG file. The following is applicable in this case:

- `COT != 6` Is responded to with `COT 45` and negatively. The connection remains intact.
- `QOI != 20` Is confirmed positively and nevertheless triggers `Station interrogation`. This corresponds to the behavior the same as for `QOI = 20`.
- `IOA !=0` Is handled the same as `IOA=0`.

For general queries, the slave supports the global address **COA=**`0xFF(FF)`. You can find detailed information on this in the Broadcast (à la page 224) chapter. A master can thus query several sectors with just one GI. This is an alternative envisaged by the standard, which the master can use. Otherwise a master must query the sectors consecutively (one after the other).

If the master sends a further GI whilst the current one has not yet been concluded, the ongoing GI is canceled and the new one is started. Overlapping GIs are logged as a warning. This is an indication of non-standard-compliant behavior with a master.

Value changes for IOs whose value has already been reported to the master as part of the current GI are buffered as long as the GI - due to other IOs is still running in the same sector (**COA**). If a sector has many IOs whose values change often and the connection to the master is comparatively slow, the buffer can grow more quickly than the master takes the values. This can happen if, for example, the **Baud Rate** is small, the **Max.user data** has been limited or the master reacts slowly. In such cases, the distribution of the IOs over several sectors can be advisable.

Attention

If the internal buffer is overfilled with value changes before the GI could be completed, the current connection to the master is disconnected.

You can find further information on this in the Transfer of the value changes (à la page 209) chapter.

The connection to Runtime cannot be established whilst a project is reloaded (à la page 209) in Runtime. The general request will fail in this case.

The GI and the buffering of the value changes are independent for different **Devices** (for different masters). A master can deactivate the ongoing general query by ASDU<100> with `COT_deact(8)`.

If the connection to Runtime has been established, but Runtime does not provide values for some IOs, the slave can confirm the GI but not close it (no`COT_actterm`). This is applicable for the IOs that are to be reported to the master (= GI) on the basis of the current query. Often, the error that no values can be provided is due to a configuration error.
Example: a write-only zenon variable (an output or command) is in the slave with an IO linked to the **Type ID** of a message (`T01-T37`).

The slave logs the information using **IOA** and **COA** addresses of the affected IOs until it detects the required values. This logging is carried out cyclically. This information is logged with level `DEEPDEBUG` in the LOG file and can be read with the `Diagnosis Viewer`.

Text: "Information object <IOA> missing value. Device: <IP> Sector: <COA>".

10.3.3 End of connection and resets

The reset in the slave is triggered if a master connects:

- 870-101: All commands will be rejected until the function code for reset (0) has been received. The code causes a reset in the link layer and the application layer.
- 870-104: As a result of the connection being disconnected, a master can detect the interruption immediately and starts the initialization process again.

ENDING COMMUNICATION

In general, communication is ended if:

- The Process Gateway application is ended
- Changes to the configuration of the slave are activated

- Runtime reloads
- the master has established the connection but no general query was sent
- The list of the value changes that have not been sent for the sector has overflowed
- A non-supported command has been received (command is answered in the negative and the connection is ended for security reasons).
- A command with a non-supported address is received (command is answered in the negative and the connection is ended for security reasons).

'Reset process command' - T105 - request

The slave can inform zenon Runtime of the receipt of a C_RP_NA_1 command. To do this:

- create an IO of type T105with IOA **0**.

If a master sends the command C_RP_NA_1, the value of the variable is first set to 0 and then to the value of QRP.

The slave confirms receipt of the command to the master and changes the value of the corresponding T105 variable for Runtime but otherwise the command has no effect in the slave. The optional sending of the 'End of initialization' (T70) is not executed automatically.

'End of Initialization' - T70 - send

From the process control system, it is possible to instigate an ASDU<70> - M_EI_NA_1 ('End of initialization) being sent to the master. To do this:

- Create a type T70 IO with **IOA** 0.
- and link to an internal variable with UINT data type

The transfer is triggered if the current value of the variable is 126 and this is amended to 127. As COI (Cause of initialization), 2 (remote reset) is used.

Note: some masters only send the general interrogation (GI) if they have received an End of initialization.

CONFIGURATION TIP

In systems where the 870 master expects an End of initialization in order to send the general interrogation, you can change the value of the variable that has linked the IO to T70 and **IOA** 0 between 126 and 127 if the other variable with T00 and **IOA** 1 (Master connected) (à la page 210) changes the value. To do this, create two **Ecriture valeur prescrite** functions for the T70 variable and link these functions in the **limit values** of the T00 IOA 1 variable:

- When the T00 IOA 1 variable has a value of 0 - set 126 to T70 variable;

- And when the value is 1 - set 127.

10.3.4 Transfer of the value changes

The IEC870 slave receives, from zenon Runtime, all value changes of the requested variables. Because, as a result of this, the list of process information that has not yet been transferred to the master can become very large, the list of outstanding value changes is limited.

Limit: at least 2048 values or 4 x number of IOs per sector.

If a master receives the values more slowly than they are generated in the process control system, the list can exceed this maximum. The slave can then no longer buffer the value changes that have not yet been transferred. The slave therefore terminates the connection to the master. The master is thus requested to obtain the next connection with a general query, so that it gets current, only recent variable values.

Attention

A reset of the link layer will be forced every time there is a possibility of data loss (lost value changes) .

The Master must be able to recognize a communication interrupt, so he can react with a new general request. This is why no process commands will be executed as long as the master has has not instigated a reset of the link.

10.3.5 Reloading Runtime

If one of the projects is reloaded in zenon Runtime, the connection to all projects is established and the connections to the master will be disconnected. For each `Device` in the IEC870 slave, a reset of the link layer is triggered. Accordingly, the rules for the start of communication are applicable, for example a general interrogation is expected.

Attention: While the reload is active, you cannot establish a connection to the Runtime. It is thus also not possible to close the general interrogation positively.

Note: After Runtime is reloaded, you can also - with the variable linked to `T00 IOA 8` - trigger reloading of the XML configuration of the Process Gateway.

10.3.6 Data priorities

The driver supports class 1 and class 2 files.

Spontaneous communication, **background scan** and **periodic data transfer** belong to class 1.

The whole file transfer is treated as class 2. This makes sure that value changes will be transferred to the Master even during a file transfer.

10.3.7 Internal type T00 (status variable)

The internal type `T00` is for the control and monitoring of the behavior of the IEC870 slave Process Gateway. The T00 variables are only communicated between the slave and the process control system, not to the 870 master.

The **IOs** with `T00` should be linked to numerical (`UINT`, `DWORD` etc.) zenon variables, from the **internal driver** for example.

IOA 1 - MASTER CONNECTED

The variable provides the status information as to whether there is already a Link Layer connection to the master. In **Device**, in the desired sector (**COA**), create a `T00` **IO** with the `IOA 1`. If a connection is established, the value will change from 0 to 1.

Note: If a master opens the connection to the Link Layer without a general interrogation (à la page 205) being sent, the **Process Gateway** will close after the expiry of the **T0** timeout. In such cases, you should attempt to provoke the GI from the master, for example due to the sending of an ASDU T70 - End of Initialization (à la page 208).

IOA 2 - REJECT OR IGNORE COMMANDS

The variable determines how the IEC870 slave should handle commands that the 870 master sends. It is possible to decide for each sector (**COA**) whether commands are executed in control direction. The commands are always enabled if no **IO** is configured.

Note: The master's commands can also be rejected in general using the **commands active** (à la page 161) property.

The zenon variable is assigned to **IO** with `T00` and `IOA 2` . You can link the same variable in all sectors (**COA**). The variable must not have the status `INVALID`.

The meaning of the variable:

Value	Description
0	Disabled The ASDUs with commands are rejected. ▸ For rejected commands, a master receives `COT_actcon (7) + PN` (negative) as confirmation. ▸ Rejected commands generate a warning entry in the LOG file. ▸ zenon Runtime is not informed of commands. Example: Depending on the local/remote mode of a Substation with Process Gateway to the remote 870 master. The remote 870 master can be informed of the current mode via an **IO** in monitoring direction (with `T30` for example) by means of the actual mode, and whilst the Substation is in local mode, the remote commands are not permitted.
1	Enabled Commands are approved.
2	Silent mode The ASDUs with commands are ignored. ▸ No reaction to the ASDUs with commands; without warning in the log. ▸ zenon Runtime is not informed of commands. Example: Redundant connection to an 870 master of two Process Gateways that have been started with the same configuration: on two computers in the zenon **network** or on a computer with 2 Ethernet cards or with different IP ports. A command from this master should only be forwarded to zenon Runtime via one of the Process Gateways.
3.	reserved for future versions

Informations

Attendant warning message:

`Commands for sector deactivated, execution prohibited.` *<Command text>*

The meaning: A command in control direction was received, but rejected due to a missing ***enable****.*

IOA 3 - SOE: FILES OVERFLOW

Status variable is set to 1 if the number of SOE files in the transfer directory exceeds the maximum number of buffered files.

IOA 4 - SOE: CURRENT FILE STATUS

Variable provides status information on the current or most recently processed SOE file:

0 - initialization. As long as no SOE file was written.

1 - SOE file is opened

2 - file was closed

3 - too many SOE files in the directory

4 - error when writing the SOE file

5 - errors when creating the SOE file

IOA 5 - STATUS OF FILE TRANSFER

Variable provides status information by means of file transfer in `Monitoring Direction` ("`Standard Direction`" - from the master to the slave).

0 - no transfer active

1 - section is selected

2 - section is requested

3 - waiting on section confirmation

4 - waiting on confirmation of file transfer

255 - error when transferring

IOA 6 - SOE: NUMBER OF FILES

Variable provides the number of SOE files present in the transfer directory.

IOA 7 - STATUS OF FILE TRANSFER IN REVERSE DIRECTION

Variable provides status information by means of file transfer in `Control Direction` ("`Reverse Direction`" - from the slave to the master).

The information is integrated in a DWORD as follows: [IOA of the transferred files] * 256 + [status information].

Possible status information:

1 = transfer is ongoing

2 - Transfer is ongoing - the transfer that had not been completed beforehand is terminated by the transfer that has currently been started.
Status 3 is also written for the transfer that has been terminated. However because the new status is immediately written afterwards, it cannot be guaranteed in the case of zenon that this is visible in Runtime.

3 - Transfer was terminated due to an error.

4 - transfer was completed and the file is available on the drive.

IOA 8 - RELOADING OF THE XML CONFIGURATION

In the IEC870 slave process gateway, it is possible to link a zenon variable to the IOA 8 with type ID <0> Internal Status.

If this linked variable has the value 1 in zenon Runtime, the configuration for the Process Gateway is read in again. After successful reading-in of the new configuration, communication is restarted in the IEC870 slave and the value of the variable is reset to 0. If this process is unsuccessful, the Process Gateway continues to work with the configuration loaded when the program started.

Restart and reloading of the XML configuration

From zenon 8.00, the restart for the Process Gateway - IEC870 slave can be triggered by a linked variable. With this restart, the XML file (à la page 191) is read in again with the configuration.

GENERAL

In the IEC870 slave process gateway, it is possible to link a zenon variable to the IOA 8 with type ID <0> Internal Status.

If this linked variable has the value 1 in zenon Runtime, the configuration for the Process Gateway is read in again. After successful reading-in of the new configuration, communication is restarted in the IEC870 slave and the value of the variable is reset to 0. If this process is unsuccessful, the Process Gateway continues to work with the configuration loaded when the program started.

ENGINEERING

Link an **Information Object** to an internal zenon variable for type identification 8 – <0> Internal Status.

To do this, carry out the following steps in the Process Gateway:

1. Select the **Sector** tab in the **Settings** configuration dialog.
2. Select a configured `Device` from the **Devices** list.
3. Select a configured `Sector` from the **Sectors** list.
4. Create a new `Information Object`.
   a) To do this, select a pre-configured object with `IOA 8` and click on the **Edit** button or click on the **New** button to create a new `Information Object`.
   The **Information object setting** (à la page 172) dialog is opened.
   b) Configure the `IOA` with the value `8` in the **Information Object Address** property.
   c) In the **Type Identification** drop-down list, select the `<0> Internal status` entry.
   Note: This entry is already selected by default.
   d) Click on the ... button to link a zenon variable.
   The **Variable selection** dialog to select a variable is opened.
   e) Select a variable from the list in the Variable area.
   f) Confirm the configuration by clicking on the **OK** button.
   g) Click on the OK button to close the IO settings dialog.

> **Informations**
>
> *This configuration is possible in every sector. This configuration is applicable for the complete Process Gateway, not for individual sectors.*
>
> *Configuration tip:*
>
> *configure the link in one sector only or always link the same zenon variable in different sectors.*

BEHAVIOR IN ZENON RUNTIME

When Runtime is started:

- The current values of the configured variables are read.
- The value of the linked variable for `IOA 8 - <0> Internal Status` is always reset to `0`.
  Note: The XML file with the configuration is only read once, even if the variable had the value `1` when communication was started.

When Runtime is running:

- When the value of the linked variable changes to `1`:
  a) All current Process Gateway connections are closed.
  b) The Process Gateway rereads the XML configuration file.

c) The value of the `IOA 8` variable is set to `0`.
Note: if there is no longer an **Information Object** with `IOA 8` in the new configuration, the value for this variable remains `1`.

- If the reading of the new configuration fails, communication continues with the old (= previous) configuration data. The variable for `IOA 8 – <0> Internal Status` is not reset.

CONFIGURATION TIP

With a variable for `IOA 8 – <0> Internal Status` , you have the following possibilities for selection in zenon Runtime:

- Monitoring of the correct start of the Process Gateway:
  If the internal IOA 8 variable has an initial value greater than `1` (`99` for example), this variable can be used to check that the Process Gateway has started correctly. If this initial value remains unchanged, the Process Gateway has not been started correctly.
  Possible cause of error: the configured variables do not exist or the IP port is assigned.
- Reloading the configuration for the Process Gateway via Runtime
  After reloading the project in zenon Runtime, the Process Gateway XML configuration file is not automatically reread.
  If variables that have also been configured in the Process Gateway are amended or recreated in the zenon Editor, you can also trigger the reading of a new XML configuration file in the Process Gateway after reloading the project in zenon Runtime. To do this, set the value 1 of the IOA 8 variable to 1 in Runtime.

10.3.8 Mapping of status bits in message direction

The IEC870 slave supports the sending of quality bits for the information objects. The quality bits of the IO are set - in ASDUs in 'montior direction' - using certain status bits of the linked zenon variable.

The quality bits are assigned to the status bits of the variables as follows:

Status bit in zenon	IEC60870 quality descriptor bits	Notes
INVALID	IV (invalid) NT (not topical)	The slave forwards the INVALID status bit in message direction as two quality bits. Note: The IV quality bit can also be set due to an IO value amendment.
ALT_VAL	SB (substituted)	Up to zenon version 7.10, it is mapped the same as the OFF bit.
OFF	IV (invalid)	
N_UPDATE (not updated in zenon network)	NT (not topical)	Example: The zenon **Process Gateway** is running on a network client. During a redundancy switching of the network server, variables are primarily marked with N_UPDATE. The values of the IOs are sent to the 870 master with the NT quality bit. After redundancy switching has been completed, the network client automatically connects to the current network server and the values are spontaneous again. The values of the IOs are sent to the 870 master without the NT quality bit.
BL_870	BL (blocked)	
SB_870	SB (substituted)	Like ALT_VAL
NT_870	NT (not topical)	
OV_870	OV (overflow)	OV_870 status bits that have already been set for the variables in zenon are sent to the master. The following is also applicable for normalized values: The value range in accordance with the IEC 60870 standard is <$-1 .. 1-2^{-15}$> with a precision of 4 decimal places. With zenon variables, this corresponds to a value range of -1.0 to 0.9999. Outside of the range for normalized values, in ASDU the value of the IOs is amended and the OV quality bit is set. Note: The OV quality bit can also be set to another type ID due to an IO value

		amendment.
T_INVAL (Zeit ungültig)	Only in "Binary Time 2a" (CP24Time2a/CP56Time2a) BS[24] - IV (invalid)	The slave forwards the T_INVAL status bit in the time stamp of the IOs with "time tag" only.

Informations

Process Gateway can mark the values with IV or OV Quality if the value of the zenon variable exceeds the value range of the ASDU type.

You can find further information in relation to this in the Assignment of the data types (à la page 179) chapter in the "Amendment of IO values" chapter.

10.3.9 Commands

A command received - T45..T64 - is set as the value for the zenon variables. In sectors without Select routing, a confirmation, which is always positive, is always sent to the master immediately - COT_actcon (7) and COT_actterm (10).

In such sectors, a Select is also immediately also confirmed automatically - COT_actcon + SE and the zenon variable change neither the value nor the status bit.

Note: For variables that are configured in the IEC870Sl Process Gateway as command variables, if the **Write set value via VBA** property is active (**Logging in** group), each command is logged as a **write set value** via the API interface in the **CEL**.

Select routing

The Select routing can only be used in conjunction with a project configuration of the zenon **command input** moduleand is only available for the **Auto/remote command** action.

The Process Gateway only reacts to a command variable by carrying out a Select Routing if the variable is configured in zenon with active **Select Before Operate** configured.

If, for the zenon variables, the **Select Before Operate** property is inactive, the **Process Gateway** behaves in the same way as for sectors without **Select routing** and always responds to a Select positively.

OVERVIEW OF ZENON VARIABLE CONFIGURATION - PROCESS GATEWAY

zenon variable Select Before Operate **property:**	**Process Gateway -** **Sector option:** Select Routing	**Behavior in the Process Gateway**
Active	Active	A Select is required. An Execute without Select is rejected. A Select is forwarded to the **command input**.
Active	Inactive	A Select is automatically responded to positively. The **command input** is not taken into account when executing the command. An Execute is always forwarded to the process driver directly.
Inactive	Active	

PROCEDURE

If, for the sector addressed, the Select routing option has been activated and **Select Before Operate** is activated for the zenon command variable, then there is a wait until a Select is received until the status SE_870 is returned with COT_actcon (7) and possibly with N_CONF (corresponds to PN) via the command variable of zenon.

Only once this has happened is a confirmation sent to the master (for example a Control Center) triggered. All others for this IO are rejected as long as a selection via Select is taking place.

TIMEOUT, CANCEL AND MISSING SELECT

The following rules apply for the different possibilities for responding to a Select routing :

- If Select routing is configured for a sector, one Select Timeout per IO is started for each Select.
- The slave checks whether the Timeout has expired and then sets the variable to SE, COT_actterm (10) and PN.
- If a Cancel - COT_deact (8) is received by the master for an ongoing Select , this is confirmed with COT_deactcon (9) and the Select is deleted.
- If an Execute or Cancel is received without a Select beforehand, it is responded to with PN (i.e. negative).

Pulse generation - 'Qualifier of Command'

The IEC870 slave supports pulse generation for commands received. The Qualifier of Command (QoC) is supported for all commands envisaged in the IEC60870 standard (T45, T46, T47 and T58, T59, T60).

For example, a pulse command received - an ASDU of type T45 - with QoC has the following effect on a zenon variable:

- **QoC 0**: Undefined in accordance with IEC standard. Behaves in slave as in QoC 3 persistent output.
- **QoC 1**: Short pulse. The zenon variable assumes the value 'ON' and changes, after the time defined in the Process Gateway settings (à la page 161) to 'OFF'.
- **QoC 2**: Long pulse. The zenon variable assumes the value 'ON' and changes, after the time defined in the Process Gateway settings (à la page 161) to 'OFF'.
- **QoC 3**: Persistent output. The zenon variable assumes the value 'ON' (or 'OFF') and retains the value. This must be reset to receive further commands.

As soon as communication is ended, the pulse for all other active pulse variables will be stopped (i.e. an OFF will be written). If a pulse is triggered for a running pulse generation, the active pulse generation will be stopped. The variable contains the value 'OFF'.

Note: For SCS (single command state) the value for ‘OFF’ is always 0. The following is applicable for DCS (double command state): An 'OFF' has the value 00b at protocol level; then either the value 2 or 0 in zenon, depending on the settings for **DPI/DCS mapping** (à la page 190).

Note: In sectors with Select routing , pulse generation cannot be carried out because 'OFF' after Select is in conflict with Execute .

10.3.10 Read - T102 - request

The IEC870 slave automatically responds to the Read command - C_RD_NA_1 - requests from the master. No additional project configuration steps are necessary for this.

A maximum of 2048 read requests can be active at the same time. Once this limit has been reached, further read requests are rejected.

The response to the read requests does not need to be configured and automatically runs in the background. For this, the following applies:

- For each IO with the requested IOA , a read is executed in zenon and a response is sent to the master with the value provided by Runtime.
- Several requests for one IO can be active at the same time.
- Read requests will also be processed even if the general request is not finished yet.

10.3.11 Time synchronization - T103 - request

In order for time synchronization to be carried out - on receipt of `C_CS_NA_1` , there must be in `Device`, at any desired `Sektor` a type `T103` IO connected to a process variable. The first IO that uses type `T103` is used as an enable.

This variable must have a value of <> 0 and must not have any of the following status bits set:

- INVALID (Bit 18)
- OFF (Bit 20)
- ALT_VAL (Bit 27)
- OV_870 (Bit 47)
- BL_870 (Bit 44)

If the `Enable` variable has not been configured or the value for this variable is missing, the connection is disconnected if a `C_CS_NA_1` is received, because it indicates that the system has not been fully configured.

Non-execution of the command due to a missing enable (value 0 or status) or rights in the operating system does not lead to the connection being terminated, the command is only confirmed negatively - `COT_actcon + PN`.

Note: to convert the computer's clock, the Process Gateway application needs administrator rights in Windows.

10.3.12 Test - T104 and T107 - requests

The IEC870 slave responds to the master automatically with the two test commands `C_TS_NA_1` (`ASDU<104>`) and `C_TS_TA_1` (`ASDU<107>`). No additional project configuration steps are necessary for this.

Informations

You can find further information on supported commands in the Interoperability (à la page 232) chapter.

10.3.13 File transfer

The **Process Gateway IEC870 Slave** supports the transfer of files in monitoring and controlling direction. To do this, configure, in **Device**, the folder property **Directory for File Transfer** and **Directory for File Transfer in Control Direction**.

The procedures of file transfer with the 870 master are automatically processed in the background. The ASDUs used are sent as class 2. This ensures that value changes are also transferred to the master during an ongoing file transfer.

The following is applicable for the ASDUs of the file transfer:

- The values 1-4 are supported in the AFQ UI4
- The values 0-6 are supported in the SCQ UI4;
  0 and 1 are both 'select file'

If a new transfer is started while another one is still active, the currently active transfer will be interrupted and the new transfer will be started.

Encrypted communication (à la page 222) of the **IEC870 Slave Process Gateway** is configured in the `zenProcGateway.INI` file.

Note: The current status of the file transfer can be evaluated using variables that are linked to the internal type T00 (à la page 210).

Storage and naming

Use the following scheme for storage and naming:

<Configured folder>\<COA>\<IOA>.<NOF>

Only 1 = 'Name of file' is supported as NOF (transparent file).
The maximum size of files for transfer is 16.711.680 Bytes.

EXAMPLE

Folder in the module configuration (à la page 161): `C:\TEMP\IEC870`

File `1100.1`: For IO of sector **COA=151** and **IOA 1100**

Save location: `C:\TEMP\IEC870\151\1100.1`

Spontaneous directory transfer

Spontaneous transfer of files in the folder for file transfer can be triggered by the process control system. To do this:

- create an IO of type F_DR_NA_1 (T126) with IOA '0'.

Value	Meaning
Change from 1 to 2	Starting transfer
3	Transfer active
4	Transfer finished
5	Transfer failure

Nested activation is prevented.

TLS encrypted communication

Encrypted communication for the **IEC870 Slave Process Gateway** is configured in the zenProcGateway.INI file.

Paramètre	Description
TLS_ACTIVE=	Type de communications sécurisées ▶ `0: deactivated` No encrypted communication ▶ `1: activated` Communication is in encrypted form on the basis of the following parameters Par défaut : `0`
TLS_CERTIFICATE_STORE_PATH=	Absolute save location of the certificates. This basic directory must contain the two subdirectories **CA** and **PRIVATE** . ▶ **CA** for trusted certificates et ▶ **PRIVATE** for your own certificates
TLS_CERTIFICATE_FILE=	Name of the TLS certificate. The default save location is configured with the entry for the `[save location of the certificates]` (**TLS_CERTIFICATE_STORE_PATH=)** . The TLS certificate must be present in the **PRIVATE** folder.
TLS_CIPHER_LIST=	List of the supported encryption processes. The list contains an abbreviation in openSSL format.
TLS_PEER_CERTIFICATE_SUBJECT=	Client certificate identification used. The client certificate must contain the **SUBJECT** configured here. La connexion est interrompue s'ils ne correspondent pas.
TLS_RENEGOTIATION_TIMEOUT=	Maximum duration of an encrypted connection before it is renewed. The encryption is reinitialized after the configured time has expired. In doing so, certificates and the key are replaced for new ones. Time indication in seconds. Par défaut : `86400`
TLS_RENEGOTIATION_MAX_BYTES=	Amount of data for which an encrypted connection is used before it is renewed. The encryption is reinitialized after the configured amount of data has been reinitialized. In doing so, certificates and the key are replaced for new ones. Indication in bytes. Par défaut : `1048576`
TLS_RESUMPTION_TIMEOUT=	Maximum duration of a key used before it is replaced.

	Only the key is renewed. The certificate is not updated in the process. The certificate is renewed after expiry of the configured time. Time indication in seconds. Par défaut : 43200
TLS_RESUMPTION_MAX_BYTES=	Amount of data for which a key is used before it is renewed. The encryption is renewed after the configured amount of data has been transferred. The certificate is not updated in the process. Indication in bytes. Par défaut : 10485760
TLS_CRL_CHECK_INTERVAL=21600	Time interval for the check to see whether the certificate currently being used is included in the **REVOCATION LIST**. The encrypted connection is no longer secure if the certificate is included in the **REVOCATION LIST**. The connection is terminated. Time indication in seconds. Par défaut : 21600

10.4 Broadcasts

Global address 0xFF(FF) in the Common Address of ASDU (COA) is supported for the following Typ IDs:

- C_IC_NA_1 - General query - ASDU<100>
- C_CS_NA_1 - Time synchronization - ASDU<103>

No additional project configuration steps are necessary for this.

Attention

The Broadcast Address 0xFF(FF) - "all outstations" - in Data Link Layer in 60870-4-101 communication is not supported.

10.5 Error analysis

Should there be communication problems, this chapter will assist you in finding out the error.

10.5.1 Analysis tool

Process Gateway behaves like a zenon driver and writes messages to a common log file. To display them correctly and clearly, use the Diagnosis Viewer (main.chm::/12464.htm) program that was also installed with zenon. You can find it under Start/All programs/zenon/Tools 8.00 -> Diagviewer. In general:

Le driver de zenon consigne toutes les erreurs dans les fichiers journaux.Les fichiers journaux sont des fichiers texte dotés d'une structure spéciale. Le dossier par défaut des fichiers journaux est le sous-dossier **LOG**, dans le dossier **ProgramData**. Par exemple :

%ProgramData%\COPA-DATA\LOG.

Attention : avec les paramètres par défaut, un driver consigne uniquement les informations d'erreur. Avec l'outil Diagnosis Viewer, vous pouvez améliorer le niveau de diagnostic de la plupart des drivers, grâce aux options "Debug" (Débogage) et "Deep Debug" (Débogage approfondi). Dans ce cas, le driver consigne également les autres tâches et événements importants.

L'outil Diagnosis Viewer vous permet également :

- Suivre les nouvelles entrées en temps réel
- De personnaliser les paramètres de journalisation
- De modifier le dossier dans lequel sont enregistrés les fichiers journaux

Remarque :

1. L'outil Diagnosis Viewer affiche toutes les entrées à l'heure UTC (temps universel coordonné), et pas à l'heure locale.
2. Par défaut, l'outil Diagnosis Viewer n'affiche pas toutes les colonnes d'un fichier journal. Pour afficher d'autres colonnes, activez la propriété **Add all columns with entry** (Ajouter toutes les colonnes avec une entrée) dans le menu contextuel de l'en-tête des colonnes.
3. Si vous utilisez uniquement **Error-Logging**, la description du problème se trouve dans la colonne **Error text**. Pour d'autres niveaux de diagnostic, la description est fournie dans la colonne **General text**.
4. En cas de problèmes de communication, de nombreux drivers consignent également les numéros d'erreur de journal qui leur sont attribués par l'automate. Ils sont affichés dans **Error text**, dans **Error code** ou dans **Driver error parameter** (**1** et **2)**. Des conseils concernant la signification des codes d'erreur sont disponibles dans la documentation du driver et le fichier journal/la description de l'automate.
5. À la fin de l'essai, réinitialisez le niveau de diagnostic **Debug** (Débogage) ou **Deep Debug** (Débogage approfondi). Pour les options **Debug** (Débogage) et **Deep Debug** (Débogage approfondi), de grands volumes de données sont enregistrés sur le disque dur, ce qui peut influencer les performances du système. L'enregistrement de ces données se poursuit même après la fermeture de l'outil Diagnosis Viewer.

> **Attention**
>
> Sous Windows CE, les erreurs ne sont pas consignées par défaut, pour préserver les performances du système.

Note: Process Gateway is not available under Windows CE.

> **Informations**
>
> *You can find further information on the Diagnosis Viewer in the Diagnosis Viewer (main.chm::/12464.htm) chapter.*

Logging

The driver supports a comprehensive logging function for:

- Client name
- zenProcGateway_IEC870Slave
- Modules

CLIENT NAME

The way the name of the client is displayed in the Diagnosis Viewer depends on the configuration. That is why the name changes if you change the TCP port, the COM port or the protocol. In this case, the logging settings may also change.

If the Process Gateway is started several times with the same settings, the same client name will be used. The error messages of the different instances can be distinguished by the process ID.

Exemple

zenProcGateway_IEC870Slave

- 104 protocol with standard port 2404
- 101 if COM1 selected.
- If the configuration has not been loaded yet, error messages during startup will always be visible under this client.

zenProcGateway_IEC870Slave:COM4

- 101 with COM4.
  The name contains the used COM port.

zenProcGateway_IEC870Slave:Port:12345

- 104 with port 12345.
  The name contains the port for the Master connection.

MODULES

Error messages are shown for all modules:

- IEC870 Slave
- OSI2
- OSI7
- SCADA Runtime (ScadaRT)

IEC870 SLAVE

Concerns the connection to the Process Gateway and its configuration.

Information about:

- Errors during engineering

OSI2

Concerns the protocol level. This where framing with its according functions is implemented.

Information about:

- Received data as byte dump
- Sent data as byte dump
- Function codes for 101
- Errors on this level, e.g. in the framing
- APCI for 104 (870-5-104 5)

Levels

- Msg: Frame interpreting
- Debug:
  Received and sent data
  Frame analysis.
  New Master connection.
- Deep Debug: Every request about existence of characters. Produces a large number of logging entries!

OSI7

Concerns the application layer.
This is where commands are interpreted and executed.

Information about:

- Logging 'process and system information'
- Errors in the ASDUs

Levels:

- Warning: Recursive interrogation
- Msg: Process information, System information
- Debug: Confirmed I-frames
- Deep Debug: Interrogation runs. Produces a large number of logging entries!

SCADA RUNTIME (SCADART)

Messages for connection to the Runtime:

Information about:

- advised and unadvised variables
- updated variables
- Deactivating the project

Levels

- Debug:
  changed values
  project inactive

LOG entries

The IEC870 slave uses the following parameters in entries in the LOG file:

Parameter	Description
deviceIp:	IP address from the device
devicename:	Name of device
COA:	Care of Address -> according to standard
SecName:	Sector description
TypeyId:	Command number
TypeIdAsTxt:	Command as text
IOA:	According to standard
AdviseID:	ID via which value changes are requested and assigned
Value:	Value transferred
TimeStamp:	Time of value change. From the driver or command
QUALDESC:	According to standard
SBEValue:	COT, select and pn mapped to unique enums: 0: Not select relevant 1: actcon+pn+select 2: actcon+select 3: deactcon+pn+select, 4: deactcon+select 5: actterm+pn+select, 6: actterm+select,
Confirmed:	1 if select already confirmed
COT:	Cause of transmission in accordance with standard
CotTxt:	COT as text
Qualifier:	According to standard and command
Originator:	According to standard

QU:	Qualifier of command QU 7.2.6.26
DCS:	7.2.6.16 Double command (IEV 371-03-03)
ControlCommand:	Depending on control command

EXAMPLE FOR 45:

```
<TypeId> <TypeIdAsTxt> Device:(<deviceip>)<devicename> Sector:(<IOA>)< SecName >
ioa:<IOA> cot:(<COT>)<CotTxt> ori:<Originator> sel:<QOC S/E> qu:<QU> dcs:<DCS>
ValueChange cmd(<TypeyId>)<<TypeId>> <TypeIdAsTxt> ioa:<IOA> id:<AdviseID> value:<Wert>
time:<TimeStamp> state:< QUALDESC > sir:<SBEValue>
```

LOG entry	Module/Level	Description
SBE IED processing. Device:(<deviceIp>)<devicename> Sector:(<COA>)<SecName> <ValueChange> SBE command: advise:<AdviseID> conf:<confirmed> <ControlCommand>	OSI7/Debug	Each value change that is processed for SBE creates this log message
Wrn:1014 SBE activation already confirmed! Device:(<DeviceIp>)<devicename> Sector:(<COA>)<SecName> <ValueChange>	OSI7/Warnung	Renewed conformation for Select received and ignored
Error:16 Advise for SBE data point fail. SBE command: advise:<AdviseId> conf:<confirmed> <ControlCommand>	Scada/error	Variable could not be requested.
Error:17 SOE value lost, queue full! Device:(<DeviceIp>)<devicename> Sector:(<COA>)<SecName> <ValueChange>	Scada/Error	Value change for SOE lost because the source is full.

10.5.2 Error codes

The following error codes can be displayed in the Diagnosis Viewer:

Error code	Meaning
570	Value change for unknown cyclical IO.
571	Transfer buffer too small to transfer a cyclical IO.
576	Loss of connection. (Master does not respond.)

ERROR MESSAGES FOR FILE TRANSFER IN REVERSE DIRECTION

Error number	Error level	Description
20	Error message	Creation of the temporary file for the file transfer in reverse direction was unsuccessful.
1017	Warning	File transfer in reverse direction is already active, the ongoing one will be ended and a new one started.
21	Error message	Segment data write error occurred
22	Error message	Length of the data exceeds the expected file length
23	Error message	Checksum of the transferred segments incorrect for the sector.
24	Error message	Checksum for file incorrect.
25	Error message	Segment data transferred for sector exceeds the data length defined at the start
26	Error message	Sector data transfer is reported as finished but the amount of segment data transferred does not correspond to that stated at the start.
27	Error message	File data is reported as completed but the given file length does not correspond.
581	Error message	Unexpected IOA with SG
582	Error message	Unexpected NOF with SG
583	Error message	Unexpected NOS with SG
584	Error message	Unexpected NOS with LS
585	Error message	Unexpected NOS with LS
586	Error message	Unexpected NOS with LS
587	Error message	FR with non-supported NOF. We only support 1 == transparent

10.5.3 Check list

870-104

870-101

1. Is the COM port in use by another application or are the settings incorrect?

General

1.
2. Is the device (PLC) that you are trying to communicate with connected to the power supply?
3. Is the cable between PLC and PC/IPC connected correctly?
4. Have you analyzed the error file (which errors did occur)?
5. For further error analysis, please send a project backup, the INI and XML files of the IEC870 slave and the LOG file of the **Diagnosis Viewer** to the support team responsible for you.

10.6 Interoperability

This companion standard presents sets of parameters and alternatives from which subsets must be selected to implement particular telecontrol systems. Certain parameter values, such as the choice of 'structured' or'unstructured' fields of the informationobject address of ASDUs represent mutually exclusive alternatives. This means that only one value of the defined parameters is admitted per system. Other parameters, such as the listed set of different process information incommand and in monitor direction allow the specification of the complete set orsubsets, as appropriate for given applications. This clause summarizes theparameters of the previous clauses to facilitate a suitable selection for aspecific application. If a system is composed of equipment stemming from different manufacturers, it is necessary that all partners agree on the selected parameters.

The interoperability list is defined as in IEC 60870-5-101 and extended with parameters used in this standard. The text descriptions of parameters which are not applicable to this companion standard are strike-through (corresponding check box is marked black).

NOTE In addition, the full specification of a system may require individual selection of certain parameters for certain parts of the system, such as the individual selection of scaling factors for individually addressable measured values.

The selected parameters should bemarked in the white boxes as follows:

[ ] Function or ASDU is not used
[X] Function or ASDU is used as standardized (default)
[R] Function or ASDU is used in reverse mode
[B] Function or ASDU is used in standard and reverse mode

The possible selection (blank, X , R , or B) is specified for each specific clause or parameter.

A black check box indicates that the option cannot be selected in this companion standard.

1. SYSTEM OR DEVICE

(system-specific parameter, indicate definition of a system or a device by marking one of the following with ' X ')

[ ] System definition
[ ] Controlling station definition (Master)
[X] Controlled station definition (Slave)

2. NETWORK CONFIGURATION: 101 ONLY

(network-specific parameter, all configurations that are used are to be marked ' X ')

[X] Point-to-point	[X] Multipoint
[X] Multiple point to point	[ ] Multipoint-star

3. PHYSICAL LAYER: 101 ONLY

(network-specific parameter, all interfaces and data rates that are used are to be marked ' X ')

TRANSMISSION SPEED (CONTROL DIRECTION)

Unbalanced interchange Circuit V.24/V.28 Standard	**Unbalanced interchange Circuit V.24/V.28 Recommended if >1 200 bit/s**	**Balanced interchange Circuit X.24/X.27**
[ ] 100 bit/s	[X] 2400 bit/s	[X] 2400 bit/s
[ ] 200 bit/s	[X] 4800 bit/s	[X] 4800 bit/s
[X] 300 bit/s	[X] 9600 bit/s	[X] 9600 bit/s
[X] 600 bit/s	[X] 19200 bit/s	[X] 19200 bit/s
[X] 1200 bit/s	[X] 38400 bit/s	[X] 38400 bit/s
	[X] 56000 bit/s	[X] 56000 bit/s
	[X] 57600 bit/s	[X] 57600 bit/s
	[X] 115200 bit/s	[X] 115200 bit/s
	[X] 128000 bit/s	[X] 128000 bit/s

	[X] 256000 bit/s	[X] 256000 bit/s

TRANSMISSION SPEED (MONITOR DIRECTION)

Unbalanced interchange Circuit V.24/V.28 Standard	Unbalanced interchange Circuit V.24/V.28 Recommended if >1 200 bit/s	Balanced interchange Circuit X.24/X.27
[ ] 100 bit/s	[X] 2400 bit/s	[X] 2400 bit/s
[ ] 200 bit/s	[X] 4800 bit/s	[X] 4800 bit/s
[X] 300 bit/s	[X] 9600 bit/s	[X] 9600 bit/s
[X] 600 bit/s	[X] 19200 bit/s	[X] 19200 bit/s
[X] 1200 bit/s	[X] 38400 bit/s	[X] 38400 bit/s
	[X] 56000 bit/s	[X] 56000 bit/s
	[X] 57600 bit/s	[X] 57600 bit/s
	[X] 115200 bit/s	[X] 115200 bit/s
	[X] 128000 bit/s	[X] 128000 bit/s
	[X] 256000 bit/s	[X] 256000 bit/s

4. LINK LAYER: 101 ONLY

(network-specific parameter, all options that are used are to be marked ' X '. Specify the maximum frame length. If a non-standard assignment of class 2 messages is implemented for unbalanced transmission, indicate the Type ID and COT of all messages assigned to class 2.)

Frame format FT 1.2, single character 1 and the fixed time out interval are used exclusively in this companion standard.

Link transmission	Frame length [octets]	Address field of the link
[X] Balanced transmission	[255] Maximum length L (both directions)*	[O] not present (balanced transmission only) optional
[X] Unbalanced transmission		[X] One octet
		[X] Two octets
		[X] Structured
		[X] Unstructured

*may be reduced by the system

Link Address 0xFF(FF) - broadcast to all outstations - is not supported.

The structure of Link Address is not interpreted.

When using an unbalanced link layer, the following ASDU types are returned in class 2 messages (low priority) with the indicated causes of transmission:

[ ] The standard assignment of ASDUs to class 2 messages is used as follows:

Type identification	Cause of transmission
9, 11, 13, 21	<1>

[X] A special assignment of ASDUs to class 2 messages is used as follows:

Type identification	Cause of transmission
120 - 126	All (as specified in the standard)

Note: (In response to a class 2 poll, a controlled station may respond with class 1 data when there is no class 2 data available).

5. APPLICATION LAYER

TRANSMISSION MODE FOR APPLICATION DATA

Mode 1 (Least significant octet first), as defined in 4.10 of IEC 60870-5-4, is used exclusively in this companion standard.

COMMON ADDRESS OF ASDU

(system-specific parameter, all configurations that are used are to be marked ' X ')

ASDU adress	
101 only [X] One octet	[X] Two octets

The ASDU address 0xFF(FF) - global address - is supported only for C_IC_NA_1 and C_CS_NA_1

INFORMATION OBJECT ADDRESS

(system-specific parameter, all configurations that are used are to be marked ' X ')

Object adress	
101 only	[X] Structured

[X] One octet	
101 only [X] Two octets	[X] Unstructured
[X] Three octets	

CAUSE OF TRANSMISSION

(system-specific parameter, all configurations that are used are to be marked ' X ')

Cause of transmission	
101 only [X] One octet	[X] Two octets (with originator address) Originator address is set to zero if not used.

LENGTH OF APDU: 104 ONLY

(system-specific parameter, specify the maximum length of the APDU per system)
The maximum length of APDU for both directions is 253. The maximum length may be reduced by the system.

[253] Maximum length of APDU per system

SELECTION OF STANDARD ASDUS

PROCESS INFORMATION IN MONITOR DIRECTION

(station-specific parameter, mark each Type ID ' X ' if it is only used in the standard direction, ' R ' if only used in the reverse direction, and ' B ' if used in both directions).

Mark	Parameter	Type
[B]	<1>: = Single-point information	M_SP_NA_1
[B]	<2>: = Single-point information with time TAG	M_SP_TA_1
[B]	<3>: = Double-point information	M_DP_NA_1
[B]	<4>: = Double-point information with time TAG	M_DP_TA_1
[B]	<5>: = Step position information	M_ST_NA_1
[B]	<6>: = Step position information with time TAG	M_ST_TA_1
[B]	<7>: = Bitstring of 32 bit	M_BO_NA_1
[B]	<8>: = Bitstring of 32 bit with time TAG	M_BO_TA_1
[B]	<9>: = Measured value, normalized value	M_ME_NA_1
[B]	<10>: = Measured value, normalized value with time TAG	M_ME_TA_1
[B]	<11>: = Measured value, scaled value	M_ME_NB_1
[B]	<12>: = Measured value, scaled value with time TAG	M_ME_TB_1
[B]	<13>: = Measured value, short floating point value	M_ME_NC_1
[B]	<14>: = Measured value, short floating point value with time TAG	M_ME_TC_1
[B]	<15>: = Integrated totals	M_IT_NA_1
[B]	<16>: = Integrated totals with time TAG	M_IT_TA_1
[ ]	<17>: = Event of protection equipment with time TAG	M_EP_TA_1
[ ]	<18>: = Packed start events of protection equipment with time TAG	M_EP_TB_1
[ ]	<19>: = Packed output circuit information of protection equipment with time TAG	M_EP_TC_1
[ ]	<20>: = Packed single-point information with status change detection	M_SP_NA_1
[ ]	<21>: = Measured value, normalized value without quality descriptor	M_ME_ND_1
[B]	<30>: = Single-point information with time TAG CP56Time2a	M_SP_TB_1
[B]	<31>: = Double-point information with time TAG CP56Time2a	M_DP_TB_1
[B]	<32>: = Step position information with time TAG CP56Time2a	M_ST_TB_1
[B]	<33>: = Bitstring of 32 bit with time TAG CP56Time2a	M_BO_TB_1
[B]	<34>: = Measured value, normalized value with time TAG CP56Time2a	M_ME_TD_1
[B]	<35>: = Measured value, scaled value with time TAG CP56Time2a	M_ME_TE_1
[B]	<36>: = Measured value, short floating point value with time TAG CP56Time2a	M_ME_TF_1
[B]	<37>: = Integrated totals with time TAG CP56Time2a	M_IT_TB_1
[ ]	<38>: = Event of protection equipment with time TAG CP56Time2a	M_EP_TD_1

[ ]	<39>: = Packed start events of protection equipment with time TAG CP56Time2a	M_EP_TE_1
[ ]	<40>:= Packed output circuit information of protection equipment with time TAG CP56Time2a	M_EP_TF_1

Either the ASDUs of the set <2>, <4>, <6>, <8>, <10>, <12>, <14>, <16>, <17>, <18>, <19> or of the set <30> – <40> are used.

PROCESS INFORMATION IN CONTROL DIRECTION

(station-specific parameter, mark each Type ID ' X ' if it is only used in the standard direction, ' R ' if only used in the reverse direction, and ' B ' if used in both directions).

Mark	Parameter	Type
[X]	<45>: = Single command	C_SC_NA_1
[X]	<46>: = Double command	C_DC_NA_1
[X]	<47>: = Regulating step command	C_RC_NA_1
[X]	<48>: = Set point command, normalized value	C_SE_NA_1
[X]	<49>: = Set point command, scaled value	C_SE_NB_1
[X]	<50>: = Set point command, short floating point value	C_SE_NC_1
[X]	<51>: = Bitstring of 32 bit	C_BO_NA_1
[X]	<58>: = Single command with time TAG CP56Time2a	C_SC_TA_1
[X]	<59>: = Double command with time TAG CP56Time2a	C_DC_TA_1
[X]	<60>: = Regulating step command with time TAG CP56Time2a	C_RC_TA_1
[X]	<61>: = Set point command, normalized value with time TAG CP56Time2a	C_SE_TA_1
[X]	<62>: = Set point command, scaled value with time TAG CP56Time2a	C_SE_TB_1
[X]	<63>: = Set point command, short floating point value with time TAG CP56Time2a	C_SE_TC_1
[X]	<64>:= Bitstring of 32 bit with time TAG CP56Time2a	C_BO_TA_1

Either the ASDUs of the set <45> – <51> or of the set <58> – <64> are used.

SYSTEM INFORMATION IN MONITOR DIRECTION

(station-specific parameter, mark ' X ' if used)

Mark	Parameter	Type
[X]	<70> : = End of initialization	M_EI_NA_1

SYSTEM INFORMATION IN CONTROL DIRECTION

(station-specific parameter, mark each Type ID ' X ' if it is only used in the standard direction, ' R ' if only used in the reverse direction, and ' B ' if used in both directions).

Mark	Parameter	Type
[X]	<100>: = Interrogation command*	C_IC_NA_1
[X]	<101>: = Counter interrogation command	C_CI_NA_1
[X]	<102>: = Read command	C_RD_NA_1
[X]	<103>: = Clock synchronization command (option see 7.6)*	C_CS_NA_1
[X]	<104>: = Test command	C_TS_NA_1
[X]	<105>: = Reset process command	C_RP_NA_1
[ ]	<106>: = Delay acquisition command	C_CD_NA_1
[X]	<107>: = Test command with time TAG CP56Time2a	C_TS_TA_1

*also with global address - with Common Address of ASDU = 0xFF(FF)

PARAMETER IN CONTROL DIRECTION

(station-specific parameter, mark each Type ID ' X ' if it is only used in the standard direction, ' R ' if only used in the reverse direction, and ' B ' if used in both directions).

Mark	Parameter	Type
[ ]	<110>: = Parameter of measured value, normalized value	P_ME_NA_1
[ ]	<111>: = Parameter of measured value, scaled value	P_ME_NB_1
[ ]	<112>: = Parameter of measured value, short floating point value	P_ME_NC_1
[ ]	<113>: = Parameter activation	P_AC_NA_1

FILE TRANSFER

(station-specific parameter, mark each Type ID ' X ' if it is only used in the standard direction, ' R ' if only used in the reverse direction, and ' B ' if used in both directions).

Mark	Parameter	Type
[B]	<120>: = File ready	F_FR_NA_1
[B]	<121>: = Section ready	F_SR_NA_1
[B]	<122>: = Call directory, select file, call file, call section	F_SC_NA_1
[B]	<123>: = Last section, last segment	F_LS_NA_1
[B]	<124>: = Ack file, ack section	F_AF_NA_1
[B]	<125>: = Segment	F_SG_NA_1
[X]	<126>: = Directory {blank or X, only available in monitor (standard) direction}	F_DR_TA_1

TYPE IDENTIFIER AND CAUSE OF TRANSMISSION ASSIGNMENTS

(station-specific parameters)

Shaded boxes: option not required.
Blank: functions or ASDU not used.

Mark Type Identification/Cause of transmission combinations:
'X ' if only used in the standard direction;
'R ' if only used in the reverse direction;
'B ' if used in both directions.

Type identification		Cause of transmission																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<1>	M_SP_NA_1		B	B		X									X					
<2>	M_SP_TA_1			B		X														
<3>	M_DP_NA_1		B	B		X									X					
<4>	M_DP_TA_1			B		X														
<5>	M_ST_NA_1		B	B		X									X					
<6>	M_ST_TA_1			B		X														
<7>	M_BO_NA_1		B	B		X									X					
<8>	M_BO_TA_1			B		X														
<9>	M_ME_NA_1	B	B	B		X									X					
<10>	M_ME_TA_1			B		X														
<11>	M_ME_NB_1	B	B	B		X									X					
<12>	M_ME_TB_1			B		X														
<13>	M_ME_NC_1	B	B	B		X									X					
<14>	M_ME_TC_1			B		X														
<15>	M_IT_NA_1			B		X									X[1]	X				
<16>	M_IT_TA_1			B		X										X				
<17>	M_EP_TA_1																			
<18>	M_EP_TB_1																			
<19>	M_EP_TC_1																			
<20>	M_PS_NA_1																			
<21>	M_ME_ND_1																			
<30>	M_SP_TB_1			B		X														
<31>	M_DP_TB_1			B		X														
<32>	M_ST_TB_1			B		X														
<33>	M_BO_TB_1			B		X														
<34>	M_ME_TD_1			B		X														
<35>	M_ME_TE_1			B		X														
<36>	M_ME_TF_1			B		X														
<37>	M_IT_TB_1			B		X										X				

Type identification		Cause of transmission																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<38>	M_EP_TD_1																			
<39>	M_EP_TE_1																			
<40>	M_EP_TF_1																			
<45>	C_SC_NA_1						X	X	X[2]	X[2]	X								X	X
<46>	C_DC_NA_1						X	X	X[2]	X[2]	X								X	X
<47>	C_RC_NA_1						X	X	X[2]	X[2]	X								X	X
<48>	C_SE_NA_1						X	X	X[2]	X[2]	X								X	X
<49>	C_SE_NB_1						X	X	X[2]	X[2]	X								X	X
<50>	C_SE_NC_1						X	X	X[2]	X[2]	X								X	X
<51>	C_BO_NA_1						X	X	X[2]	X[2]	X								X	X
<58>	C_SC_TA_1						X	X	X[2]	X[2]	X								X	X
<59>	C_DC_TA_1						X	X	X[2]	X[2]	X								X	X
<60>	C_RC_TA_1						X	X	X[2]	X[2]	X								X	X
<61>	C_SE_TA_1						X	X	X[2]	X[2]	X								X	X
<62>	C_SE_TB_1						X	X	X[2]	X[2]	X								X	X
<63>	C_SE_TC_1						X	X	X[2]	X[2]	X								X	X
<64>	C_BO_TA_1						X	X	X[2]	X[2]	X								X	X
<70>	M_EI_NA_1*				X															
<100>	C_IC_NA_1						X	X	X	X	X							X	X	
<101>	C_CI_NA_1						X	X			X							X	X	
<102>	C_RD_NA_1					X												X	X	X
<103>	C_CS_NA_1						X	X										X	X	
<104>	C_TS_NA_1						X	X												
<105>	C_RP_NA_1						X	X										X	X	
<106>	C_CD_NA_1																			
<107>	C_TS_TA_1						X	X												
<110>	P_ME_NA_1																			
<111>	P_ME_NB_1																			
<112>	P_ME_NC_1																			

Type identification		Cause of transmission																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<113>	P_AC_NA_1																			
<120>	F_FR_NA_1													X					X	
<121>	F_SR_NA_1													X					X	
<122>	F_SC_NA_1					X								X				X	X	
<123>	F_LS_NA_1													X					X	
<124>	F_AF_NA_1													X				X	X	
<125>	F_SG_NA_1													X					X	
<126>	F_DR_TA_1*			X		X														

* Blank or X only

[1] Optional.

[2] Slave confirms deactivations of write command (TI <45> ... <64>) but by deactivated '**Select routing**' the corresponding activation is already transferred to execution; COT_actcon and COT_actterm were already sent.

COT	Cause of Transmission	
<0>	not used	
<1>	periodic, cyclic	per/cyc
<2>	background scan	back
<3>	spontaneous	spont
<4>	initialized	init
<5>	request or requested	req
<6>	activation	act
<7>	activation confirmation	actcon
<8>	deactivation	deact
<9>	deactivation confirmation	deactcon
<10>	activation termination	actterm
<11>	return information caused by a remote command	retrem
<12>	return information caused by a local command	retloc
<13>	file transfer	file
<14...19>	reserved	
<20>	interrogated by station interrogation	inrogen
<21...36>	interrogated by interrogation of the group 1..16	inro1..16
<37>	requested by general counter request	reqcogen
<38...41>	requested by counter interrogation of the group 1 ... 4	reqco1..4
<42, 43>	reserved	
<44>	unknown type identification	
<45>	unknown cause of transmission	
<46>	unknown common address of ASDU	
<47>	unknown information object address	
<48, 63>	for special use (private range)	

6. BASIC APPLICATION FUNCTIONS

STATION INITIALIZATION

(station-specific parameter, mark 'X' if function is used)

[ ] Remote initialization

CYCLIC DATA TRANSMISSION

(station-specific parameter, mark ' X ' if function is only used in the standard direction, ' R ' if only used in the reverse direction, and ' B ' if used in both directions)

[X] Cyclic data transmission

READ PROCEDURE

(station-specific parameter, mark ' X ' if function is only used in the standard direction, ' R ' if only used in the reverse direction, and ' B ' if used in both directions)

[X] Read procedure

SPONTANEOUS TRANSMISSION

(station-specific parameter, mark ' X ' if function is only used in the standard direction, ' R ' if only used in the reverse direction, and ' B ' if used in both directions)

[B] Spontaneous transmission

DOUBLE TRANSMISSION OF INFORMATION OBJECTS WITH CAUSE OF TRANSMISSION SPONTANEOUS

(station-specific parameter, mark each information type ' X ' where both a Type ID without time and corresponding Type ID with time are issued in response to a single spontaneous change of a monitored object)

The following type identifications may be transmitted in succession caused by a single status change of an information object. The particular information object addresses for which double transmission is enabled are defined in a project-specific list.

[ ] Single-point information M_SP_NA_1, M_SP_TA_1, M_SP_TB_1 and M_PS_NA_1
[ ] Double-point information M_DP_NA_1, M_DP_TA_1 and M_DP_TB_1
[ ] Step position information M_ST_NA_1, M_ST_TA_1 and M_ST_TB_1
[ ] Bitstring of 32 bit M_BO_NA_1, M_BO_TA_1 and M_BO_TB_1 (if defined for a specific project)
[ ] Measured value, normalized value M_ME_NA_1, M_ME_TA_1, M_ME_ND_1 and M_ME_TD_1
[ ] Measured value, scaled value M_ME_NB_1, M_ME_TB_1 and M_ME_TE_1
[ ] Measured value, short floating point number M_ME_NC_1, M_ME_TC_1 and M_ME_TF_1

STATION INTERROGATION

(station-specific parameter, mark ' X ' if function is only used in the standard direction, ' R ' if only used in the reverse direction, and ' B ' if used in both directions).

[X] global		
[ ] group 1	[ ] group 7	[ ] group 13
[ ] group 2	[ ] group 8	[ ] group 14
[ ] group 3	[ ] group 9	[ ] group 15
[ ] group 4	[ ] group 10	[ ] group 16
[ ] group 5	[ ] group 11	Information object addresses assigned to each group must be shown in a separate table.
[ ] group 6	[ ] group 12	

CLOCK SYNCHRONIZATION

(station-specific parameter, mark ' X ' if function is only used in the standard direction, ' R ' if only used in the reverse direction, and ' B ' if used in both directions).

[X] Clock synchronization

[ ] Day of week used
[ ] RES1, GEN (time tag substituted/ not substituted) used
[B] SU-bit (summertime) used

optional, see 7.6

COMMAND TRANSMISSION

(station-specific parameter, mark ' X ' if function is only used in the standard direction, ' R ' if only used in the reverse direction, and ' B ' if used in both directions).

[X] Direct command transmission
[X] Direct set point command transmission
[X] Select and execute command
[X] Select and execute set point command
[X] C_SE ACTTERM used

[X] No additional definition
[X] Short-pulse duration (duration determined by a system parameter in the outstation)
[X] Long-pulse duration (duration determined by a system parameter in the outstation)
[X] Persistent output

[setting] Supervision of maximum delay in command direction of commands and set point commands
[no limit] Maximum allowable delay of commands and set point commands

TRANSMISSION OF INTEGRATED TOTALS

(station-specific parameter, mark ' X ' if function is only used in the standard direction, ' R ' if only used in the reverse direction, and ' B ' if used in both directions).

[X] Mode A: Local freeze with spontaneous transmission
[X] Mode B: Local freeze with counter interrogation
[X] Mode C: Freeze and transmit by counter-interrogation commands
[X] Mode D: Freeze by counter-interrogation command, frozen values reported spontaneously

[X] Counter read
[X] Counter freeze without reset
[ ] Counter freeze with reset
[ ] Counter reset

[X] General request counter
[ ] Request counter group 1
[ ] Request counter group 2
[ ] Request counter group 3
[ ] Request counter group 4

PARAMETER LOADING

(station-specific parameter, mark ' X ' if function is only used in the standard direction, ' R ' if only used in the reverse direction, and ' B ' if used in both directions).

[ ] Threshold value
[ ] Smoothing factor
[ ] Low limit for transmission of measured values
[ ] High limit for transmission of measured values

PARAMETER ACTIVATION

(station-specific parameter, mark ' X ' if function is only used in the standard direction, ' R ' if only used in the reverse direction, and ' B ' if used in both directions).

[ ] Act/deact of persistent cyclic or periodic transmission of the addressed object

TEST PROCEDURE

(station-specific parameter, mark ' X ' if function is only used in the standard direction, ' R ' if only used in the reverse direction, and ' B ' if used in both directions).

[X] Test procedure

FILE TRANSFER

(station-specific parameter, mark 'X' if function is used).
File transfer in monitor direction

[X*] Transparent file
[ ] Transmission of disturbance data of protection equipment
[X] Transmission of sequences of events
[ ] Transmission of sequences of recorded analogue values

* a data can be transparently transported by the system but not generated or evaluated. Maximum file size is 16711680 bytes.

File transfer in control direction

[X] Transparent file

BACKGROUND SCAN

(station-specific parameter, mark 'X ' if function is only used in the standard direction, 'R ' if only used in the reverse direction, and 'B ' if used in both directions).

[X] Background scan

ACQUISITION OF TRANSMISSION DELAY

(station-specific parameter, mark 'X ' if function is only used in the standard direction, 'R ' if only used in the reverse direction, and 'B ' if used in both directions).

[ ] Acquisition of transmission delay

DEFINITION OF TIME OUTS: 104 ONLY

Parameter	Default value	Remarks	Selected value
t0	30 s	Time-out of connection establishment	setting
t1	15 s	Time-out of send or test APDUs	setting
t2	10 s	Time-out for acknowledges in case of no data messages; t2 < t1	setting
t3	20 s	Time-out for sending test frames in case of a long idle state; t3 > t1	setting

Recommended range for timeouts t0 - t2 : 1s to 255s, accuracy 1s

Recommended range for timeout t3 : 0s to 48hrs, accuracy 1s
Long timeouts for t3 may be needed in special cases where satellite links or dialup connections are used (e.g. to establish connection and collect values only once per day or week). For dialup connections it may be necessary to give up the connection supervision completely. This is achievable by setting the timeout t3 to zero.

MAXIMUM NUMBER OF OUTSTANDING I FORMAT APDUS K AND LATEST ACKNOWLEDGE APDUS (W): 104 ONLY

Parameter	Default value	Remarks	Selected value
k	12 APDUs	Maximum difference receive sequence number to send state variable	setting
w	8 APDUs	Latest acknowledge after receiving w I format APDUs	setting

Recommended range of values k: 1 to 32767 APDUs, accuracy 1 APDU

Recommended range of values w: 1 to 32767 APDUs, accuracy 1 APDU (Recommendation: w should not exceed two-thirds of k)

PORTNUMBER: 104 ONLY

Parameter	Default value	Remarks
Portnumber	2404	setting

REDUNDANT CONNECTIONS

[ ] Number N of redundancy group connections used

RFC 2200 SUITE

RFC 2200 is an official Internet Standard which describes the state of standardization of protocols used in the Internet as determined by the Internet Architecture Board (IAB). It offers a broad spectrum of actual standards used in the Internet. The suitable selection of documents from RFC 2200 defined in this standard for given projects has to be chosen by the user of this standard.

[ ] Ethernet 802.3
[ ] Serial X.21 interface
[ ] Other selection from RFC 2200:

List of valid documents from RFC 2200
1. .....
2. .....
3. .....
4. .....
5. .....
6. .....
7. etc.

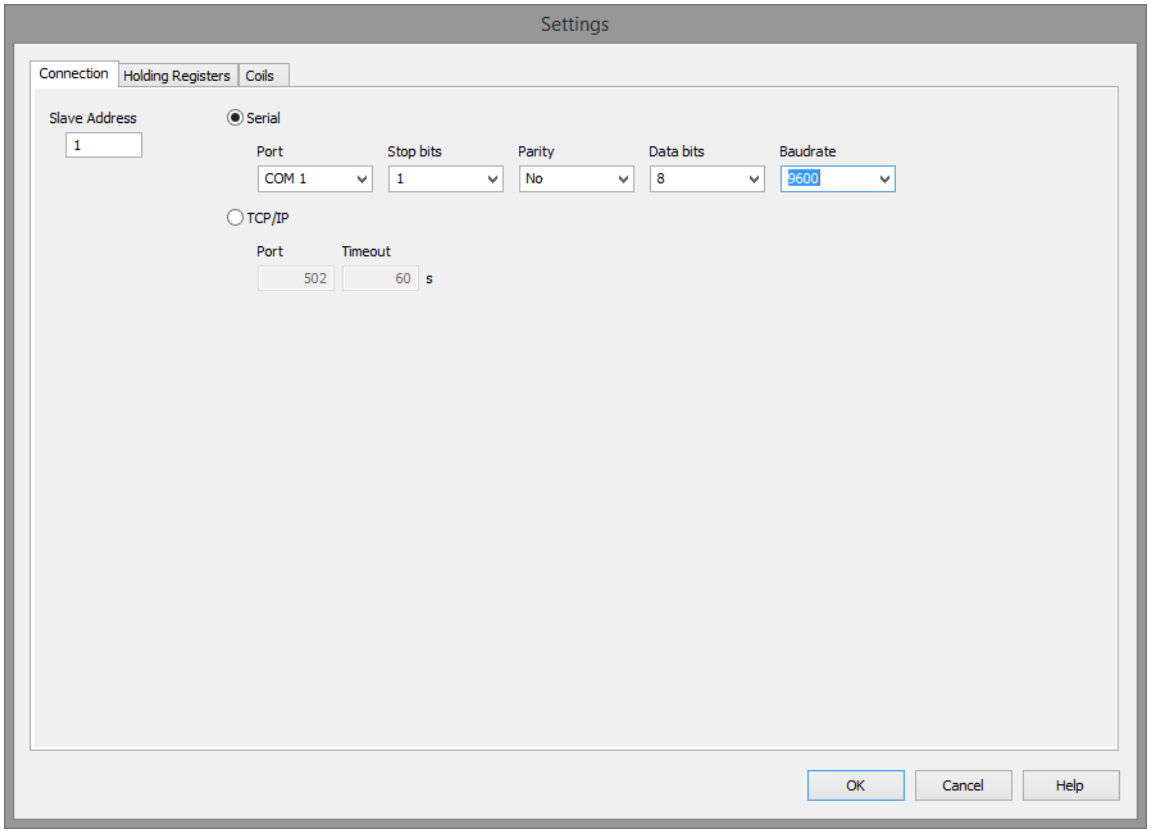
11. Protocole MODBUS Slave

The **Process Gateway** for **MODBUS Slave** supports:

- Function Code 1:
  Read Coils
- Function Code 3:
  Read Multiple Registers
- Function Code 5:
  Write Single Coil
- Function Code 6:
  Write single Register
- Function Code 15:
  Write Multiple Coils
- Function Code 16:
  Write Multiple Registers

11.1 Configuration du module

Cliquez sur le bouton Paramètres dans la boîte de dialogue de démarrage de Process Gateway pour ouvrir la boîte de dialogue de sélection de variables. Les variables sélectionnées de cette manière sont ensuite continuellement synchronisées dans zenon et/ou dans le système MODBUS.



MODBUS CONNECTION

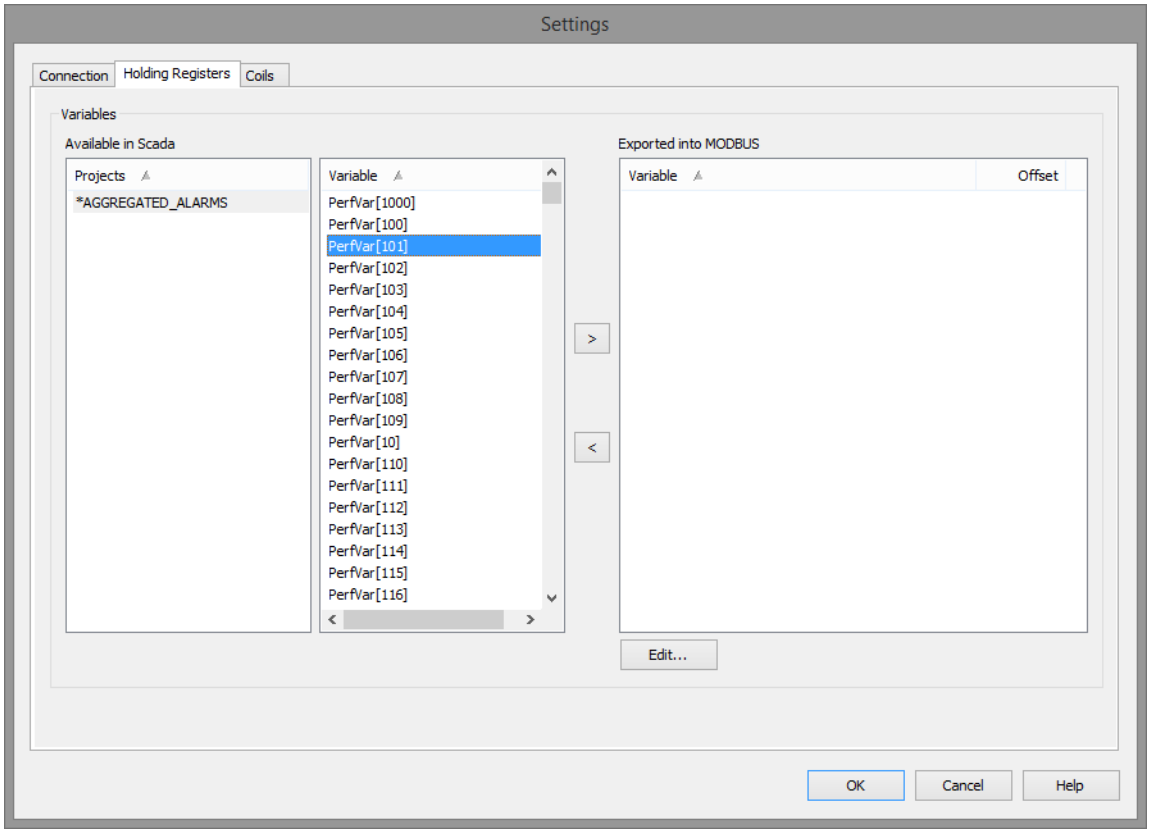
MODBUS Connection	Paramètres de connexion MODBUS.
Slave Adress **Unit Identifier**	MODBUS hardware address for the MODBUS communication. ▶ **Slave Adress** pour la communication série. ▶ **Unit Identifier** ou TCP/IP : Par défaut : `1`

Serial	`Active` : Une connexion série est utilisée.
	▸ **Port** : Interface série utilisée pour l'accès au système MODBUS. Sélectionnez dans la liste déroulante.
	▸ **Stop bits** : Nombre de bits de stop. Sélectionnez dans la liste déroulante :
	par défaut `1`
	▸ **Parity :** Réglage de la parité. Sélection dans la liste déroulante :
	`No` (par défaut)
	`Impair`
	`Pair`
	▸ **Data bit** : Nombre de bits de données. Sélectionnez dans la liste déroulante :
	par défaut `8`
	▸ **Débit de données** : Vitesse de l'interface série. Sélectionnez dans la liste déroulante :
	par défaut `9600`

TCP/IP	`Communication au maître via le protocole TCP/IP` ▶ `Active` : La connexion TCP/IP avec MODBUS TCP est utilisée. ▶ `Inactive` : connexion série avec MODBUS RTU ▸ Port : Adresse du port utilisé pour la connexion. ▸ Délai d'attente : Période d'attente.
OK	Applique les paramètres et ferme la boîte de dialogue.
Cancel	Annule toutes les modifications et ferme la boîte de dialogue.

11.2 Holding Registers

Cliquez sur le bouton Paramètres dans la boîte de dialogue de démarrage de Process Gateway pour ouvrir la boîte de dialogue de sélection de variables. Les variables sélectionnées de cette manière sont ensuite continuellement synchronisées dans zenon et/ou dans le système MODBUS.



VARIABLES

Sélection des variables exportées vers MODBUS et synchronisées avec zenon.

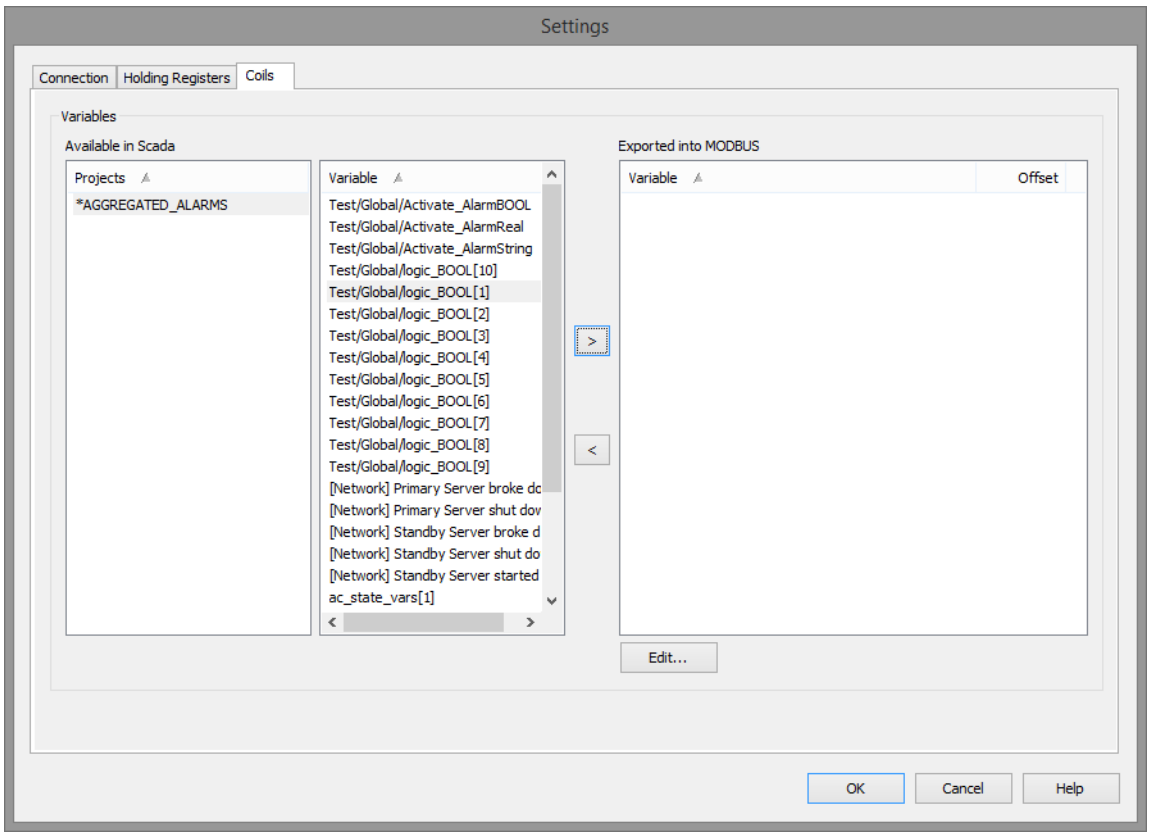
Paramètre	Description
Available in Scada	Affiche toutes les variables disponibles dans zenon. Les variables issues de projets actifs peuvent être sélectionnées dans le cadre de la gestion multi-projets. ▸ **Projects** : affiche tous les projets disponibles ; le projet standard est identifié par un ***** ▸ **Variables** : affiche toutes les variables à transférer issues du projet sélectionné
Exported into MODBUS	Dresse la liste de toutes les variables exportées vers MODBUS et leur offset. La liste peut être triée en cliquant sur les en-têtes. ▸ `Variables:` : Dresse la liste de toutes les variables exportées vers MODBUS. Les variables non issues du projet standard comportent le nom du projet en préfixe, séparé par un caractère **#** avant le nom de la variable. ▸ `Offset:` Offset de la variable (offset du Holding Register MODBUS)
Pfeil-Tasten	Attribution des variables à MODBUS : ▸ Cliquez sur le bouton **>** pour ajouter des variables de la liste **Available in Scada** à la liste **Exported into MODBUS**. Une boîte de dialogue d'attribution d'adresse s'affiche (voir la section **Boîte de dialogue d'attribution**). ▸ Cliquez sur le bouton **<** pour supprimer les variables de la liste `Exported into MODBUS` et les replacer dans la liste `Available in Scada`.
Edit...	▸ Opens the dialog for addressing (à la page 257) of the selected zenon variable to an offset MODBUS.

Remarque :

- Modification du nom d'un projet ou de variables dans Editor :
  En cas de modification d'un nom de projet ou d'une variable dans Editor, la modification doit également être effectuée dans le fichier de configuration (à la page 258), afin que le nom dans le fichier INI et le préfixe de la liste **Exported into MODBUS** soit modifié en conséquence.
- Modification en projet standard :
  Si le projet standard est modifié dans la gestion multi-projets, les variables doivent être modifiées.
  Les variables du projet standard ne doivent pas comporter de préfixe. Toutes les autres variables nécessitent un préfixe **Projectname#**.
- Write Holding Register:

- Maximum **block size** : 123

11.3 Coils



VARIABLES

Sélection des variables exportées vers MODBUS et synchronisées avec zenon.

Paramètre	Description
Available in Scada	Affiche toutes les variables disponibles dans zenon. Les variables issues de projets actifs peuvent être sélectionnées dans le cadre de la gestion multi-projets. ▸ **Projects**: affiche tous les projets disponibles ; le projet standard est identifié par un ***** ▸ **Variables**: affiche toutes les variables à transférer issues du projet sélectionné
Exported into MODBUS	Lists all variables exported to MODBUS of data type `BOOL` and their offset. La liste peut être triée en cliquant sur les en-têtes. ▸ `Variables::` Dresse la liste de toutes les variables exportées vers MODBUS. Les variables non issues du projet standard comportent le nom du projet en préfixe, séparé par un caractère **#** avant le nom de la variable. ▸ `Offset:` Offset de la variable (offset du Holding Register MODBUS)
Pfeil-Tasten	Attribution des variables à MODBUS : ▸ Cliquez sur le bouton **>** pour ajouter des variables de la liste **Available in Scada** à la liste **Exported into MODBUS**. Une boîte de dialogue d'attribution d'adresse s'affiche (voir la section **Boîte de dialogue d'attribution**). ▸ Cliquez sur le bouton **<** pour supprimer les variables de la liste `Exported into MODBUS` et les replacer dans la liste `Available in Scada`.
Edit	Opens the dialog for addressing (à la page 257) of the selected zenon variable to an offset MODBUS.

Remarque :

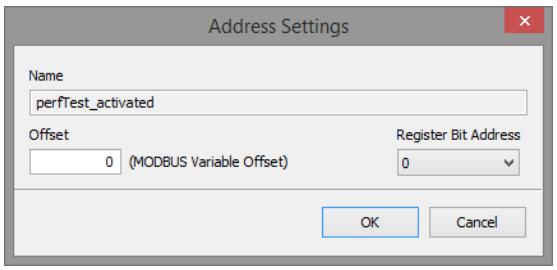
- Modification du nom d'un projet ou de variables dans Editor :
  En cas de modification d'un nom de projet ou d'une variable dans Editor, la modification doit également être effectuée dans le fichier de configuration (à la page 258), afin que le nom dans le fichier INI et le préfixe de la liste **Exported into MODBUS** soit modifié en conséquence.
- Modification en projet standard :
  Si le projet standard est modifié dans la gestion multi-projets, les variables doivent être modifiées.
  Les variables du projet standard ne doivent pas comporter de préfixe. Toutes les autres variables nécessitent un préfixe **Projectname#**.

- Write Holding Register:
  - Maximum **block size** : 123

11.4 Paramètres d'adresse

BOÎTE DE DIALOGUE D'ATTRIBUTION

Si une variable est déplacée vers la liste Exported into MODBUS en cliquant sur le bouton **>**, une boîte de dialogue d'attribution du nom de la variable de zenon à un offset MODBUS s'affiche.



Paramètre	Description
Name	Nom de la variable. Remarque: Le nom de la variable est grisé, car il ne peut plus être modifié.
Offset	Offset : Décalage du Holding Register MODBUS. Offset de la variable dans MODBUS Attention : 0 est toujours saisi en tant qu'offset par défaut.
Register Bit Address	▸ Sélectionnez dans la liste déroulante : Bool Bit-number in the Holding Register Default: 0 ▸ USINT: Low Byte oder High Byte the Holding Register. Default: High Byte Note: not available for the assignment of **Coils**.
OK	Applique les paramètres et ferme la boîte de dialogue.
Cancel	Annule toutes les modifications et ferme la boîte de dialogue.

11.5 Adresse matérielle du système Modbus

Vous pouvez définir l'adresse matérielle (adresse Modbus) pour le système MODBUS accessible avec le **Process Gateway**. Ce paramètre est défini dans la boîte de dialogue de configuration (à la page 251), par le biais de la propriété **Slave Address (Adresse de l'esclave)**. L'adresse `1` est fournie par défaut.

11.6 Fichier de configuration : entrées spécifiques pour AccessMODBUS

Le fichier de configuration `zenProcGateway.ini` contient des entrées spécifiques à MODBUS. Le fichier doit se trouver dans le dossier système.

[MODBUS]

Entrée	Description
COMPORT=	port de communication série (COM1=0, COM2=1, ...). Peut être défini dans la boîte de dialogue de configuration (à la page 251).
BAUD=	Vitesse de l'interface série. Peut être défini dans la boîte de dialogue de configuration (à la page 251).
BYTESIZE=	Nombre de bits de données de l'interface série
PARITY=	Paramètres de parité du port de communication série : 0=Non 1=Impair 2=Pair Peut être défini dans la boîte de dialogue de configuration (à la page 251).
STOPBITS=	Nombre de bits d'arrêt de l'interface série. 0=1 1=1,5 2=2 Peut être défini dans la boîte de dialogue de configuration (à la page 251).
TIMEOUT=	timeout pour l'interface série en millisecondes
HWADDRESS=	Adresse matérielle du système Modbus. Par défaut : 1 Peut être défini dans la boîte de dialogue de configuration (à la page 251).
REFRESHRATE=	Période de rafraichissement, en ms
SERIELL=	Communications série ou TCP/IP : 1 = Série 0 = TCP/IP Peut être défini dans la boîte de dialogue de configuration (à la page 251).
PORT=	port TCP/IP
TCPTIMEOUT=	timeout pour la communication TCP/IP en secondes

[MODBUS VARIABLES]

Variables à remplacer dans AccessMODBUS.

Remarque : Cette entrée remplace l'entrée **[VARIABLES]** originale à partir de zenon 7.11. This prevents unnecessary replacement using using its **Logic to SCADA connection** for variables from the standard project.

Entrée	Description
COUNT	Nombre de variables à exporter
OFFSET_n	Adresse Modbus attribuée au numéro correspondant (**n**). La numérotation commence par zéro (0).
NAME_n	Nom de la variable attribuée au numéro correspondant (**n**). La numérotation commence par zéro (0). Exemples : ▶ OFFSET_0=0 NAME_0=EMS_Supply area 1_forecast_final_consumption Variable du projet de démarrage ▶ OFFSET_1=2 NAME_1=BASISTUTORIAL#Temperaturfühler Variable issue du projet BASISTUTORIAL

▶

12. Microsoft Azure

The **Process Gateway** uses the **AccessAzure.dll** to establish a cyclical connection to MS Azure. Services dans MS Azure supportés par Process Gateway :

- Servicebus Queue
- Event Hub
- IoT Hub

En outre, le module Process Gateway peut recevoir des données du hub IoT via `Cloud2Device-Message`.

Les données issues de la **Queue** peuvent ensuite être obtenues du driver **AzureDrv**de MS Azure, puis être intégrées aux processus de zenon. **IoT Hub** ne peut être lu par le pilote que s'il est transféré dans **Servicebus Queue**. Ceci peut être implémenté avec les services standard de l'application MS Azure Cloud.

Les données issues d'un **Event Hub** sont destinées aux applications tierces et ne peuvent plus être lues par zenon.

DRIVER COMMUNICATION

Pour lire valeurs issues de la Servicebus Queue, le driver **AzureDrv** établit une connexion à la Queue du bus de service avec le nom configuré et lit tous les messages reçus de celle-ci. Ces messages sont décompressés et les valeurs en ligne qu'ils contiennent sont attribuées aux variables. La clé de cette opération est l'**Adresse symbolique**.
Tous les messages déjà présents dans la file d'attente lors du premier établissement réussi de la connexion sont chargés, puis effacés. Dans tous les cas, seules les valeurs actuelles sont affichées.

Ceci signifie :
Chaque instance de driver sur chaque ordinateur possède sa propre file d'attente (**Queue**) en tant que signal d'entrée.

Exemple

La Servicebus Queue pour l'ordinateur **MYSERVER1** et le préfixe **onlinedata** configuré sont appelés comme ceci dans Microsoft Azure : **onlinedata_myserver1**.

Les valeurs numériques `(DOUBLE)` et alphanumériques `(STRING)` sont prises en charge. Les informations d'horodatage et les bits d'état du système sont transférés vers la variable cible.

Vous trouverez également des informations générales concernant MS Azure dans le manuel **MS Azure**.

12.1 Configuration du module

Boîte de dialogue de configuration de la connexion Microsoft Azure via Process Gateway.

VARIABLES

Dans le groupe **Variables**, vous configurez les variables dont les valeurs sont enregistrées par zenon dans un bus de service MS Azure (MS Azure Service Bus.

Paramètre	Description
Available in Scada	Affiche toutes les variables disponibles dans zenon. Les variables issues de projets actifs peuvent être sélectionnées dans le cadre de la gestion multi-projets.
`Projects`	Liste de tous les projets disponibles. Le projet standard est identifié par un `*`.
`Variables`	Liste de toutes les variables du projet sélectionné. La liste peut être triée, et une sélection multiple est possible. Conseil : Double-cliquez sur une variable pour la déplacer.
Bouton >	Les variables sélectionnées dans la liste de variables sont déplacées vers la liste **Exported into MS Azure** .
Bouton <	Les variables sélectionnées sont retirées de la liste **Exported into MS Azure**.
Exported into MS Azure	Liste des variables écrites par Process Gateway dans la file d'attente du bus de service de Microsoft Azure (MS Azure Service Bus). These are displayed with name (**Points**) and communication **Direction**. Double click on an entry to reverse the direction of communication. La clé des valeurs dans Microsoft Azure est toujours `PROJECTNAME#VARIABLENAME.` **Points:** The name consists of: ▶ Nom du projet ▶ **#** (en guise de séparateur) ▶ Nom de la variable. **Direction:** Shows the direction of communication: ▶ `read only` ▶ `write only` ▶ `read/write` Par défaut : `read only` Changez de direction en double-cliquant sur la variable. Remarque : Only linked variables can be described.

CONNEXIONS MICROSOFT AZURE

Toutes les connexions cible dans lesquelles les valeurs actuelles de la variable sélectionnée doivent être insérées dans Microsoft Azure doivent être saisies dans le groupe **MS Azure Connections**.

Toutes les valeurs de variables actuelles sont alors simultanément ajoutées à la connexion créée dans Tous.

Paramètre	Description
Liste de connexions Microsoft Azure	Affiche la liste de toutes les connexions configurées à Microsoft Azure. Chaque connexion comprend le nom de la connexion (MS Azure Connection) et le Service Bus Queue Name. ► Pour créer une nouvelle connexion, cliquez sur le bouton **New...** ► Une connexion sélectionnée peut être modifiée en cliquant sur le bouton **Edit...**.
MS Azure Connection	**MS Azure** Adresse de connexion
Service Bus Queue Name	Nom de la file d'attente dans le bus de service de Microsoft Azure.
New...	Ouvre la boîte de dialogue de configuration de la connexion **MS Azure**.
Edit...	Ouvre les connexions existantes pour vous permettre de configurer la connexion à Microsoft Azure (**MS Azure**).
Delete	Supprime la connexion **MS Azure** sélectionnée de la liste.

PARAMÈTRES DE MICROSOFT AZURE

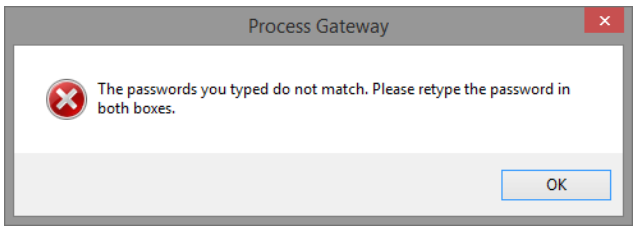
Paramètre	Description
Integrity period	Intervalle de temps durant lequel les valeurs actuelles des variables sélectionnées sont écrites sous forme d'image dans la file d'attente du bus de service de Microsoft Azure (**MS Azure Service Bus**). Si la valeur d'une variable change pendant cette période, le changement de valeur est immédiatement transféré vers la file d'attente de **MS Azure**. Par défaut : 5 s
HTTP Proxy Domain	Adresse du serveur proxy sur le réseau.
HTTP Proxy User	Champ de saisie du nom d'utilisateur de connexion au serveur proxy.
HTTP Proxy Password	Champ de saisie du mot de passe de connexion au serveur proxy. Remarque : La saisie est affichée sous forme de points (même pendant la saisie).
Confirm HTTP Proxy Password	Champ de saisie de confirmation du mot de passe de connexion au serveur proxy. Remarque : La saisie est affichée sous forme de points (même pendant la saisie).

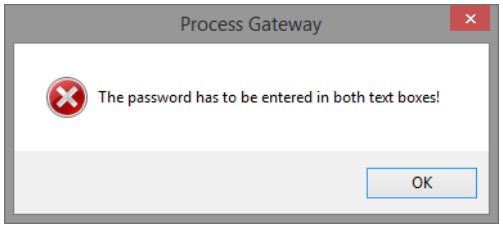
NAVIGATION

Paramètre	Description
OK	Applique les paramètres et ferme la boîte de dialogue.
Cancel	Annule toutes les modifications et ferme la boîte de dialogue.

BOÎTE DE DIALOGUE EN CAS DE SAISIE DE VALEURS ERRONÉES

Les configurations du serveur proxy sont validées. Une boîte de dialogue d'avertissement est affichée en cas de configuration incorrecte.



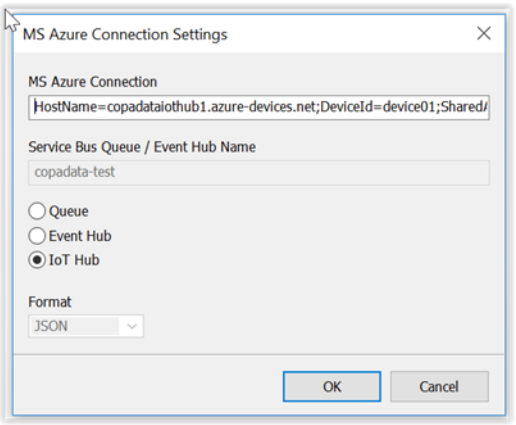


Attention

.NET Framework 4.5 doit être installé sur l'ordinateur pour que la configuration du serveur proxy fonctionne.

12.1.1 Paramètres de la connexion à Microsoft Azure

Boîte de dialogue de configuration de la connexion à Microsoft Azure (**MS Azure**) :



Paramètre	Description
MS Azure Connection	**MS Azure** Adresse de connexion Remarque : Vous pouvez lire et copier cette adresse dans le portail d'administration d'Azure, dans la section **Manage Connection Strings** de l'**Servicebus Namespace** concerné.
Service Bus Queue / Event Hub Name	Nom de la file d'attente ou du **Event Hub** du **MS Azure Service Bus**. La sélection de l'attribution s'effectue au moyen des cases d'option : Remarque : Seuls les caractères qui apparaissent dans le nom d'une MS Azure Service Bus Queue ou dans un Event Hub sont autorisés. Utilisez des noms simples, courts et évocateurs. Évitez les caractères spéciaux, les lettres spécifiques à une langue et les espaces. **Service Bus Queue** Le nom de la **Service Bus Queue** comprend : ▸ Le préfixe librement configurable ▸ Un tiret bas (_) ▸ Le nom NETBIOS de l'ordinateur (sans suffixe de nom de domaine) en caractères minuscules Si le nom n'existe pas encore dans l'Namespace, une Queue possédant ce nom est créée. Éléments requis : Les droits correspondants sont présents. **Event Hub** ▸ **Format des messages** : Dans la liste déroulante, sélectionnez l'option **Format**. ▸ **Event Hub Name**: Noms des Event Hub dans le MS Azure Service Bus Le Event Hub doit déjà avoir été créé avec ce nom dans l'Namespace. Remarque : Grayed out if the **IoT Hub** option is activated as the type of communication to MS Azure.
Queue	communique avec MS Azure via `Service Bus Queue`. ▸ `Active` : La connexion est établie avec le nom du **Service Bus Queue**.
Event Hub	▸ communique avec MS Azure via `Event Hub`. ▸ `Active` : La connexion est établie avec le nom du **Event Hub**. Sélection du format de message avec l'option **Format**.
IoT Hub	▸ communique avec MS Azure via `IoT Hub`. ▸ `Active` : Envoi des valeurs réelles et réception des messages via `IoT Hub`. Saisie de la `Device-Connection String` à l'appareil dans le champ de saisie de **MS Azure Connection**field.

Paramètre	Description
Format	Format de message d'établissement de la connexion via le **Event Hub**. Sélectionnez dans la liste déroulante : ▸ XML ▸ JSON ▸ BOND (compact binary)
OK	Applique les paramètres et ferme la boîte de dialogue.
Cancel	Annule toutes les modifications et ferme la boîte de dialogue.

Informations

Le Service Bus Queue Name peut être librement configuré.
Cette file d'attente est créée automatiquement dans Microsoft Azure pendant la première communication avec Microsoft Azure.

Pour chaque ordinateur appelant des données depuis MS Azure, utilisez le driver **AzureDrv** pour créer une MS Azure Connection distincte.

Communication via IoT hub

Ce qui suit est applicable pour la communication via IoT hub :

- All content is transferred in the correct format using the **Process Gateway**.
- Les valeurs réelles sont transmises au **IoT Hub** sous forme de message.
- Les messages reçus par le **IoT Hub** sont décodés et écrits dans la valeur contenue sous forme de valeur prescrite.

MESSAGE DU IOT HUB

Un message via IoT Hub contient le contenu suivant :

Paramètre	Type de données	Lire (recevoir)	Ecrire (envoyer)	Description
project	STRING	X	X	Nom du projet zenon
variable	STRING	X	X	Nom de variable zenon
isstring	NUMBER	X	X	Type de la variable de zenon ▸ 0 = La valeur de variable est numérique ▸ 1 = La valeur de variable est une chaîne
numvalue	NUMBER	X	X	Valeur numérique de la variable zenon . Remarque : uniquement pour les valeurs numériques si `isstring=0`. Pour les variables chaîne, la valeur est pour `numvalue 0`.
strvalue	STRING	X	X	Valeur de la chaîne Remarque : uniquement pour les variables de chaîne de caractères si `isstring=1`. La valeur pour `strvalue` est vide pour les valeurs numériques.
time	ISO timestamp	X	X	Horodatage de la variable de zenon.
status	NUMBER	X	X	État de la variable de zenon
isintegrity	NUMBER	--	X	Déclencheur pour le transfert de données. ▸ 0 = Message envoyé à MS Azure suite à un changement de valeur. ▸ 1 = message a été envoyé de façon cyclique. Remarque : L'envoi cyclique correspond à l'option **Integrity period** configurée dans la configuration du module (à la page 261).
sequence	NUMBER	--	X	Numéro de série pour les messages envoyés. Celle-ci est automatiquement exécutée par le module Process Gateway et est augmentée pour chaque message.
msgtime	ISO timestamp	--	X	Horodatage du message.

Touche :

X: pris en charge

--: indisponible

Les messages dans le sens `receive` sont identiques à ceux dans le sens envoi. The `isintegrity`, `sequence` and `msgtime` parameters are not needed to `receive`. All content is transferred in the correct format using the **Process Gateway**. All content is transferred in the correct format using the **Process Gateway**.

Exemple

Format possible d'un message envoyé par Process Gateway :

```
{"isintegrity":1,"isstring":1,"msgtime":"2017-05-03T08:51:23.260Z","numvalue":0,"project":"AZURE_TEST","sequence":1,"status":1078067200,"strvalue":"sunshine","time":"2017-04-11T05:28:33.542Z","variable":"weather"}
```

12.2 Format des messages

FILE D'ATTENTE DU BUS DE SERVICE

Les messages dans la file d'attente (Queue) doivent posséder un format sérialisé `OnlineValueMessage` pour .NET.

FORMAT :

```
public class OnlineValueMessage
{
public string strVarProject { get; set; }
public string strVarName { get; set; }
public long nVarID { get; set; }
public double fValue { get; set; }
public string strValue { get; set; }
public bool bValueIsString { get; set; }
public long nTime_s { get; set; }
public short nTime_milli { get; set; }
public long nStatus { get; set; }
}
```

HUB D'ÉVÉNEMENTS

Si vous utilisez le Event Hub en tant que connexion, vous pouvez choisir parmi les formats de message XML, JSON et BOND (compact binary). Le nom de la variable de zenon est Partition-Key.

FORMAT BOND

```
namespace AzureServiceBusShared

struct ArchiveMessage
{
0: string strArvProject;
1: string strArvName;
2: string strVarProject;
3: string strVarName;
4: int64 nVarID;
5: int16 nCalc;
6: double fValue;
7: string strValue;
8: bool bValueIsString;
9: int64 nTime_s;
10: int16 nTime_milli;
11: int64 nStatus;
}

struct OnlineValueMessage
{
0: string strVarProject;
1: string strVarName;
2: int64 nVarID;
3: double fValue;
4: string strValue;
5: bool bValueIsString;
6: int64 nTime_s;
7: int16 nTime_milli;
8: int64 nStatus;
}
```

12.3 Message d'erreur

Messages d'erreur dans le fichier journal :

Message d'erreur	Niveau de débogage	Description
`Project %s inactive fail!`	**Error**	L'événement de projet inactif n'a pas pu être traité par Process Gateway dans le délai autorisé.
`Project %s active fail!`	**Error**	L'événement de projet actif n'a pas pu être traité par Process Gateway dans le délai autorisé.
`Value change queue full: Value change for id:%u of project:%s lost!`	**Error**	La file d'attente des modifications de valeurs issues du Runtime est saturée. Les valeurs supplémentaires sont refusées. La connexion à Azure est probablement trop lente pour retransmettre toutes les modifications de valeurs.
`Adding of %d Online Rows failed for '%s' Queue '%s`	**Error**	L'insertion des modifications de valeurs dans la file d'attente du bus de service a échoué.
`Adding of %d Online Rows succeeded for '%s' Queue '%s'`	**Deep Debug**	L'insertion des modifications de valeurs dans la file d'attente du bus de service a réussi.
`Starting to add Online Rows failed for '%s' Queue '%s'`	**Error**	L'établissement de la connexion à la file d'attente du bus de service a échoué.
`Creating Azure Wrapper failed`	**Error**	Impossible de charger le fichier ManagedAzureWrapper.dll.

IOT

Paramètre	Description	Description
`Sending value to '%s' failed`	**Error**	L'écriture de la valeur prescrite a échoué
`IoTHubMessage_GetByteArray() failed`	**Error**	L'appel à l'API SDK a échoué.
`Received message from IoT Hub`	**Deep Debug**	Message from IoT hub received.
`Exception when deserializing payload: %s`	**Error**	Received message is in the wrong format
`IoTHubClient_CreateFromConnectionString() failed`	**Error**	L'appel à l'API SDK a échoué.
`IoTHubClient_SetMessageCallback() failed`	**Error**	L'appel à l'API SDK a échoué.
`IoTHubMessage_CreateFromByteArray() failed`	**Error**	L'appel à l'API SDK a échoué.
`Sent message to IoT Hub`	**Deep Debug**	Le message a été envoyé.

12.4 Fichier de configuration : entrées spécifiques pour Microsoft Azure

Le fichier de configuration `zenProcGateway.ini` contient des entrées à Microsoft Azure.

[AZURE]

Entrée	Description
INTEGRITYPERIOD=	Intervale du cycle d'écritures, en secondes.
PROXY_DOMAIN=	Adresse du serveur proxy.
PROXY_USER=	Nom d'utilisateur de connexion au serveur proxy.
PROXY_PWD=	Mot de passe de connexion au serveur proxy. Remarque : Le mot de passe est également affiché sous forme cryptée dans le fichier.INI.

Remarque : .NET Framework 4.5 doit être installé sur l'ordinateur pour que la configuration du serveur proxy fonctionne.

[VARIABLES]

Entrée	Description
Name_n=	Nom de la variable pour Process Gateway. Format : `SOURCEPROJECT#SOURCEVARIABLE`. Numérotation (n) : `n` Commence par le numéro de série de la configuration de la connexion. La numérotation commence par zéro (0). Exemple : ▶ 1 variable génère `n = 0` ▶ 3 variable donnent les résultats suivants pour la troisième entrée, `n = 2`
Count=	Nombre de variables. Remarque : La numérotation commence ici par `1`. Huit connexions génèrent `Count=8`.

[CONNECTIONS]

Entrée	Description
CONN_n=	Nom de la connexion MS Azure. Numérotation (n) : `n` Commence par le numéro de série de la configuration de la connexion. La numérotation commence par zéro (0). Exemple : ▶ 1 connexion génère `n = 0` ▶ 3 connexions donnent les résultats suivants pour la troisième entrée, `n = 2`
QUEUE_n=	Nom de la file d'attente MS Azure. Pour la numérotation (_n), les mêmes règles s'appliquent que pour **CONN**
COUNT=	Numéro de la connexion MS Azure configurée. Remarque : La numérotation commence ici par `1`. Deux connexions génèrent `Count=2`.

13. OPC UA Server

The **OPC UA Server** provides the variables used in the project to all standard-compliant **OPC UA Client** tools.

Informations

*The **OPC UA Server** can be used for all zenon versions from 5.50.*

The predecessor to the **OPC UA Server**is the OPC DA server. This is not part of **Process Gateway**.

The **OPC UA driver** and all dialogs are only available in English.

MULTI-PROJECT CAPABILITY

The **OPC UA Server** is multi-project compatible from zenon 7.11 onwards. Variable from the Runtime project and all its subprojects can be selected. In doing so, the object name from the variable name and the project name are combined. Configurations for **OPC UA Clients** that were created before zenon 7.11 are thus not compatible. These must be amended when using zenon 7.11 or higher.

From version 7.60, the NodeIDs change if variables are not explicitly defined in the configuration. The data type of the NodeID is string. The NodeID always consists of the project GUID and the variable ID. If variables have been explicitly configured, this change has no effect.

Exemple

NodeID: "6d5ea886-1b04-4d9c-aa35-2b5664c500cd-38d"; the following is applicable in the process: "6d5ea886-1b04-4d9c-aa35-2b5664c500cd" is the **Identifiant du projet** and "38d" corresponds to the variable ID 909.

From version 7.60, the NameSpace Index is "2" by default, instead of "80", for new configurations. The NameSpace Index can no longer be configured in the user interface. This change has no effect for existing configurations.

The earlier action can be activated again by means of the "**configuration_version=1**" entry in the INI file. This can, for example, make sense for a NameSpace Index with 80 or for a configurable NameSpace Index.

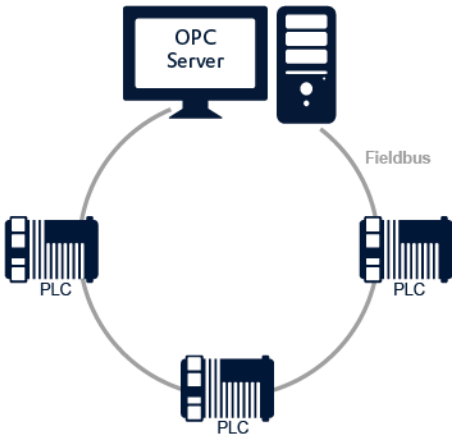
13.1 Comparaison du serveur OPC et du serveur OPC UA

Le système de serveur OPC UA (**OPC UA Server**) a remplacé le serveur OPC. Nous vous recommandons d'utiliser un serveur OPC UA (**OPC UA Server**) pour la plupart de vos applications.

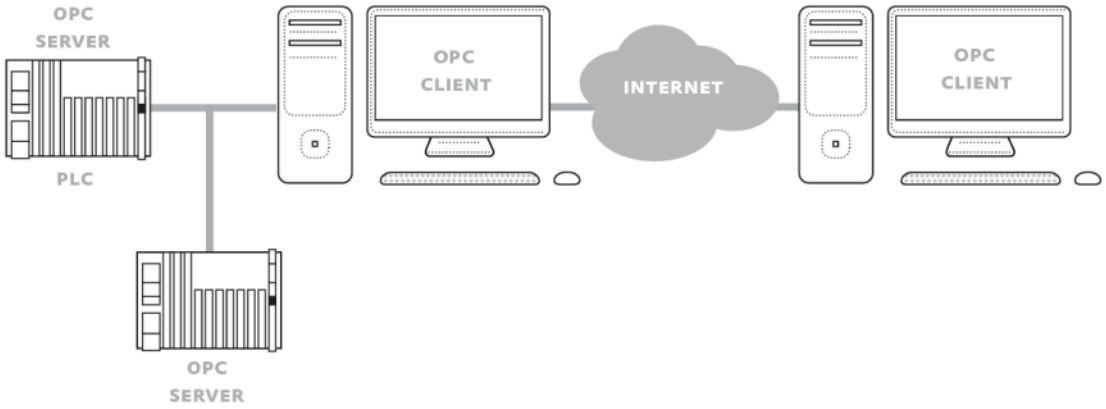
SERVEUR OPC

- OPC Task Force existe depuis 1995
- Interface uniforme pour les systèmes d'automation
- Repose sur la technologie COM/DCOM de Microsoft
- La fondation OPC existe depuis 1996

- Plusieurs spécifications pour différentes applications



OPC UA SERVER



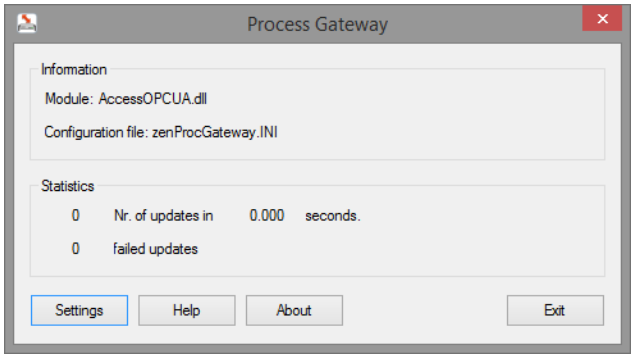
COMPARAISON DU SERVEUR OPC ET DU SERVEUR OPC UA

Paramètre	Serveur OPC	OPC UA Server
Modèle de données	Limité	Pour toutes les applications
Mise en œuvre	Dépendante des fabricants	Indépendante des fabricants
Interopérabilité	--	**X**
Configuration	Fastidieuse	Simple
Plusieurs instances	--	**X**
Utilisation en réseau	--	**X**
Technologie de réseau	--	`OPC UA TCP binary Protocol`
Performances	faibles	élevées
Plates-formes	Windows uniquement	Indépendant
Ressources requises	élevées	moindres
Architecture orientée services	--	**X**
Sécurité	Obsolète	Conforme aux normes actuelles
Sécurité de la connexion	--	**X**
Windows CE	--	**X**

Légende :

- **X**: pris en charge
- **--**: indisponible

13.2 Module configuration



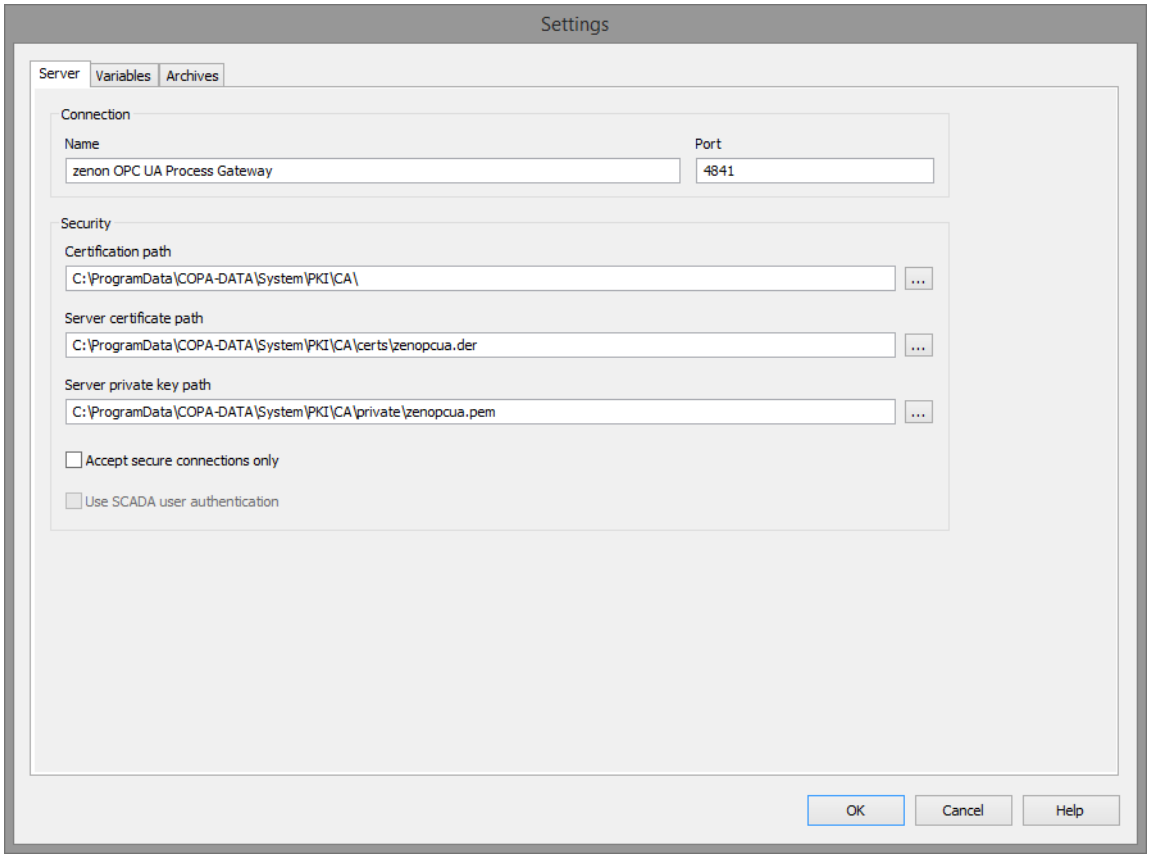
To configure the server and variables for OPC UA:

1. In the **Process Gateway** statistics dialog, click the **Configuration** button.
   The dialog for the configuration of the **OPC UA module** is opened.

2. In the module configuration dialog, you configure:
   - Server (à la page 279)
     Connection parameters to the **OPC UA Server**
   - Variables (à la page 285)
     Variable assignment

13.2.1 Server

You configure connection parameters to an OPC UA server in the **Server** tab.



CONNECTION

Communication parameters to the OPC/OPC UA server.

Parameter	Description
Name	Name of the server as it is shown on the client. Default: `zenon OPC UA Process Gateway`
Port	TCP port via which the client establishes the connection to the OPC UA server. Default: `4841`

SECURITY

Settings and assignment of certificates for secure and encrypted communication.

Parameter	Description
Certificates path	Path for the root folder with the certificate files and the respective subfolders. Click on the **...** button to open the file selection dialog Default: `%ProgramData%\PKI\CA\` Note: If this root folder is changed, the paths for **Server certificate path** and **Server private key path** must also be amended.
Server certificate path	File with the public key of the server. This public key is used by the client for the encryption of messages. Click on the **...** button to open the file selection dialog Files with the following file type are shown for selection: ▸ Format: `DER` ▸ Suffix: `.der` Default: `%ProgramData%\PKI\CA\certs\zenopcua.der` Note: This path must be a subfolder of the configured **Certificates path**. In the event of an error, a warning dialog is visualized and the **OK** button is grayed out.
Server private key path	File with the private key of the server. This private key is used by the server to decrypt the messages from the client. Click on the **...** button to open the file selection dialog Files with the following file type are shown for selection: ▸ Format: `PEM` ▸ No password Default: `%ProgramData%\PKI\CA\private\zenopcua.pem` Note: This path must be a subfolder of the configured **Certificates path**. In the event of an error, a warning dialog is visualized and the **OK** button is grayed out.
Accept secure connections only	Checkbox to be ticked if only one secure connection can be used to establish the connection. ▸ `Active`: Establishment of a connection is only accepted with a secure connection. Securing is carried out each time with a signature or with a signature and encryption. ▸ `Inactive`: Establishment of a connection is only accepted with a secure

	connection. The following are considered a secure connection: ▸ `Basic128Rsa15:` RSA15 as key wrap algorithm and 128-bit basic as algorithm for the message encryption ▸ `Basic256:` 256-bit basic as algorithm for message encryption Default: `Inactive` Note: for connections with a signature, the packages are secured against tampering, but the contents of the package are not encrypted.
Trust all client certificates	Checkbox to select whether all client certificates are accepted for the establishment of a connection or only client certificates known on the server. ▸ `Active`: All client certificates are accepted and the connection is made. ▸ `Inactive`: All clients that want to connect with the server must send a valid certificate. This certificate is also checked to see if it is in the list of known client certificates **Certificate trust list path**. Default: `active` Note: An **OPC UA Client** without a secure connection does not send a certificate. For an **OPC UA Client** with a secure connection, the certificate must always be stored in the **Trust List** folder.
Use SCADA user authentication	Checkbox to configure the connection. If there is no user administration configured in the zenon project, this checkbox is grayed out. ▸ `Active`: Clients can only log on to the server with a valid user ID and password. Note: To do this, the user administration must be activated and configured in zenon Editor. ▸ `Inactive`: Anonymous login is possible. Attention: The user name and password are not transferred in encrypted form in the **OPC UA** telegram. If necessary, use your own users in the project without assigned authorization levels.

CLOSE DIALOG

OK	Applique toutes les modifications effectuées sur tous les onglets, puis ferme la boîte de dialogue. Attention: All active connections are ended and the server is restarted with the new settings. Connection errors may occur in the process.
Cancel	Annule toutes les modifications effectuées sur tous les onglets, puis ferme la boîte de dialogue.
Help	Ouvre l'aide en ligne.

ERROR MESSAGE

Error messages are displayed in red font. In doing so, not all errors are shown together. The first error of the configuration is shown each time. It is therefore possible that a new error message is shown after a correction has been made.

Certificates

*All certificates must correspond to the format **X509**.*

CERTIFICATE STORE

*The **OPC UA Server** in the **Process Gateway** uses an `OpenSSL Directory certificate store`.*

SERVER CERTIFICATE

The server certificate created on initial startup is a self-signed certificate. The certificate contains a public key. In addition, a suitable private RSA key (1024 bit) is generated.

*Although the **OPC UA Gateway** can in principle run with the self-created certificate, it is recommended that a certificate that corresponds to the requirements is created for productive operation. To do this, the self-created certificate can serve as a template for applications that allows creation of the certificates according to user criteria.*

Attention

The file name `zenopcua` is used by both the **OPC UA Server** as well as by the **OPCUA32 Client driver**. If encrypted communication between the **OPCUA32 client driver** and **OPC UA Server** in the **Process Gateway** is configured for tests on the same computer, it is best that the certificate for the **OPCUA32 client driver** is renamed before the certificate is copied to the **Trust List** of the **OPC UA Server**.

*As a URI in the **SubjectAltName** field, the certificate gets the `Full Qualitfied Domainname` (FQN) of the computer, the default port number `4841` from the **OPC UA Server** and a DNS entry with the computer name. The URI of the **OPCUA Server** is compiled dynamically and changes if, for example, the port number is changed. The server certificate is not changed in this case. A client that carries out a strict check of the server certificate can then reject the server certificate. It can also be necessary to create your own certificate that contains an entry for the IP address in addition to the DNS entry for the `SubjectAltname`.*

CERTIFICATE TRUST LIST/CERTIFICATE TRUST CHAIN

*All certificates with the suffix `.der` are checked by the **OPC UA Server** and classed as a trustworthy certificate. As an alternative to the self-designated Application Instance **certificate** of an **OPC UA Client**, the public root certificate of the Certificate Authority that was used to sign an Application Instance **certificate** of an **OPC UA Client** can also be stored. The Application Instance **certificate** of the client does not need to be present on the server in this case.*

CERTIFICATE REVOCATION LIST

If an **OPC UA client** uses an Application Instance **certificate** that is signed by a Certificate Authority, the Certificate Revocation List is also checked by the server in addition to the certificate. The **OPC UA Server** checks all files with the suffix **.crl** in the **Crl** subfolder.
Attention: The **.crl** file must be stored in `PEM format`. Only one single **.crl** file can be present per Certificate Authority.

The connection is denied if:

- The client certificate is included in the **Revocationlist** BadCertificateRevoked
- The CRL is no longer current (BadCertificateTimeInvalid)

If, for a certificate that was signed by a Certificate Authority, there is no **Certificate Revocation List** present, it is assumed that there is no such Revocation List. In this case, the certificate is treated as trustworthy.

Generate new certificate

If necessary, new certificates can be generated when the OPC UA server is started.
To do this:

1. Remove the existing certificates.
   There are two possibilities for doing so:
   a) Delete the following folder: `%programdata%\COPA-DATA\system\PKI`.
   In doing so, all **Trusted certificates** and **Certificate revocation lists** are removed.
   Or:
   b) Delete the following files in order to get all other certificates and lists:
   zenopcua.der in the folder `%programdata%\COPA-DATA\system\PKI\CA\certs`

zenopcua.pem in the folder
`%programdata%\COPA-DATA\system\PKI\CA\private`

zenopcua.crl in the folder `%programdata%\COPA-DATA\system\PKI\CA\crl`

2. Open the **zenprocgateway.ini** configuration file.
3. Go to the section **[OPCUA]**.
4. Remove the following entries:
   - **CERTIFICATES_PATH=**
   - **SERVER_CERTIFICATE_LOCATION=**
   - **SERVER_PRIVATE_KEY_LOCATION=**
5. Start the OPC UA **Process Gateway**.
6. Confirm the message box by clicking on the **Yes** button to create new certificates.

> **Attention**
>
> *If server certificates need to be renewed, each OPC UA client that needs a valid server certificate must receive the renewed certificate.*

CREATE THE SERVER CERTIFICATE FROM SCRATCH

The following situations can make it necessary to create the server certificate from scratch:

- several instances of Process Gateway on the same system with different port numbers
- a change to the port number
- a change to the computer name
- the certificate has expired

13.2.2 Variables

Variables for the **OPC UA Process Gateway** can be selected automatically or individually.

Variables from the current project and its subprojects can be selected. The name on the server consists of the variable names and the project names.

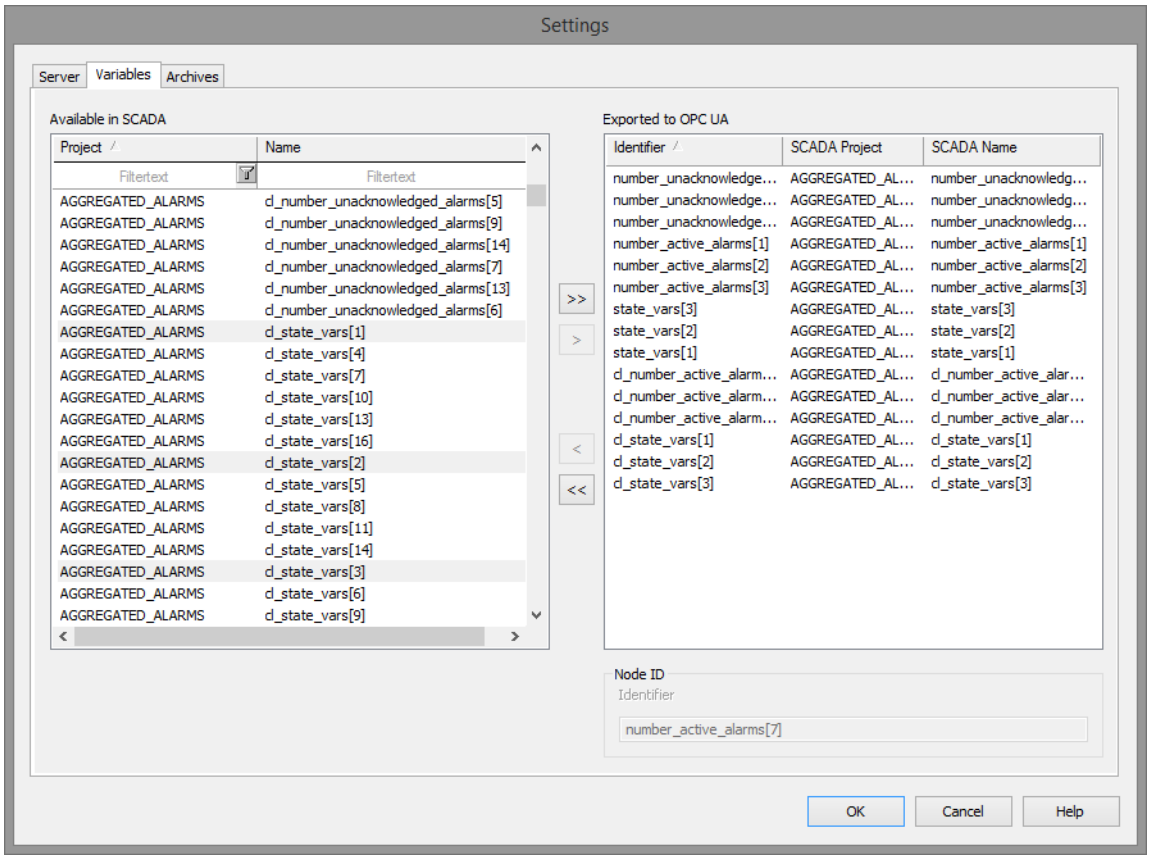
> **Attention**
>
> Variables that do not have the **Ecrire valeur prescrite** property activated in the zenon Editor cannot be written by an **OPC UA Client**.

DEFINING VARIABLES

If no variables are defined in the configuration for the **OPC UA Server**, all variables from the available zenon projects are added to the `OPC UA Address Space`. All variables (including system driver variables) are visible for the clients.

with manual configuration (at least one configured variable), the values defined by the user are used. The **Namespace Index 2** is used by default and a **String Identifier** with the variable names is entered. The **Namespace Index** cannot be configured.

You configure the variables that are to be provided as a Node in the OPC UA data model in the **Variables** tab.



Parameter	Description
Available in SCADA	List of the configured variables available in the zenon Runtime. ▶ `Project` zenon project name ▶ `Name` zenon variable name Corresponds to the **Nom** variable property in zenon Editor. The list can be sorted and filtered. ▶ Sort by clicking on the column heading. Another click changes the sorting order. ▶ Filter by entering the filter text in the input field. The characters * or ? are possible as placeholders.
Arrow keys	Assignment of the zenon variables to OPC UA: ▸ Clicking on the **>** button adds selected variables from the **Available in SCADA** list to the **Exported to OPC UA** list. ▸ Clicking on the **<** button removes variables from the **Exported to OPC UA** list and inserts them back into the **Available in SCADA** list. ▸ Clicking on the >> button adds selected variables from the **Available in SCADA** list to the **Exported to OPC UA** list. ▸ Clicking on the << button removes all variables from the **Exported to OPC UA** list. Multiple selection is possible by clicking the mouse and pressing the CTRL key.
Exported to OPC UA	List of exported variables. The variables can be sorted according to: ▸ Identifier ▸ SCADA Project ▸ SCADA name
Node Id	The input field for **Node ID** is only available if an entry in the **Exported to OPC UA** list is selected.

NODE ID

Parameter	Description
Identifier	Defines the `Node Identifier` for the selected OPC UA variable. The **Node Identifier** type is automatically defined through the entry: ▸ A numerical `Identifier` is automatically created with numerical input. ▸ If a GUID is entered, a `GUID Identifier` is created automatically, otherwise the `Identifier` is treated as a string. GUID form: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx) The field must not be empty. The **Node Identifier** must be unique in the complete address area.

VARIABLE EXPORT TO OPC UA

To export variables to OPC UA:

1. Select the desired variables and aggregation archives in **Available in SCADA** with a mouse click. Multiple selection is possible.
2. Click on the arrow **Exported to OPC UA**

   To export all variables, click on the double arrow

The zenon variable name is used as **Browsename** and **Displayname**.

Informations

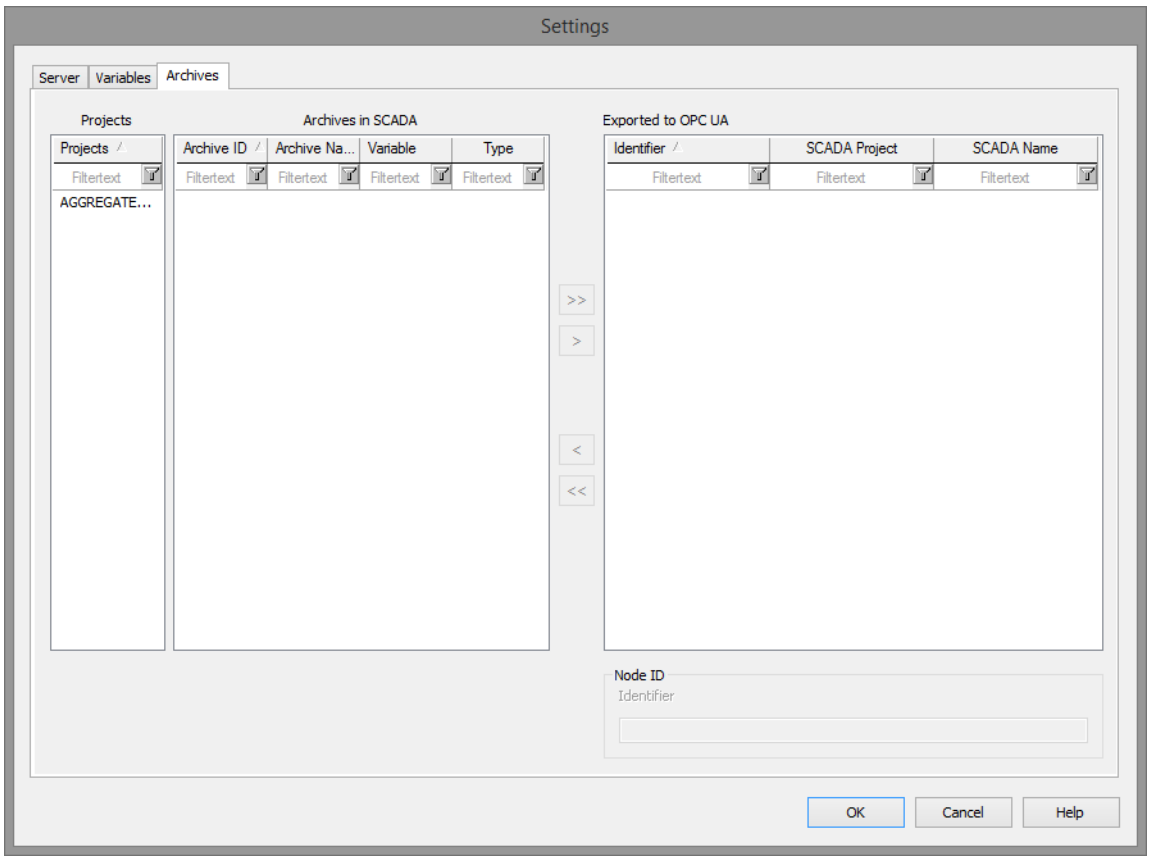
For zenon variables with a slash (/) in the names, the /is replaced by a dot (.). The parts of the name separated by a slash are saved in their own subfolders.

Example: The zenon variable "`Static/Scalar/Int32`" is divided into:

- An OPC UA variable "`Int32`"
- In a folder called "`Scalar`",
- which is a subfolder of the "`Static`" folder

This works with all data types, multi-dimensional arrays, umlauts in names and slashes at the end.

13.2.3 Archives



PROJECTS

List of the project activated in the zenon Editor.

Parameter	Description
[Project name]	Name of the project loaded in the zenon Editor. In **Archives in SCADA** the content of the selected project is displayed.

ARCHIVES IN SCADA

List of the archive variable of the selected project.

Parameter	Description
[Variable list]	List of the projected archive variables in the zenon Runtime. ▶ Archive ID Identification of the available zenon archive. ▶ Archive Name Projected name of the zenon archive. ▶ Variable Projected name of the archive variable. ▶ Type Archive type The list can be sorted and filtered. ▶ Sort by clicking on the column heading. Another click changes the sorting order. ▶ Filter by entering the filter text in the input field. The characters * or ? are possible as placeholders.
Arrow keys	Assignment of the archive variables to OPC UA: ▸ Clicking on the **>** button adds selected variables from the **Archives in SCADA** list to the **Exported to OPC UA** list. ▸ Clicking on the **<** button removes variables from the **Exported to OPC UA** list and inserts them back into the Archives in SCADA list. ▸ Clicking on the >> button adds selected variables from the **Archives in SCADA** list to the **Exported to OPC UA** list. ▸ Clicking on the << button removes all variables from the **Exported to OPC UA** list. Multiple selection is possible by clicking the mouse and pressing the CTRL key.
Exported to OPC UA	List of exported variables. The variables can be sorted according to: ▸ Identifier - put together from: Project name + archive ID + archive variable name + archive type ▸ SCADA Project Name of the zenon Logic project. ▸ SCADA name -put together from: archive ID + archive variable name +

	archive type
Node ID	The input field for **Node ID** is only available if an entry in the **Exported to OPC UA** list is selected.

FERMER

Option	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.

13.3 Start

When starting, **Process Gateway** loads the configured module.

When first starting or if no variables have been explicitly configured, the **OPC UA Server** automatically creates the data model for all variables from all projects in Runtime. With many variables, or with many variables configured with alarms, the start can take a corresponding length of time. It is, in principle, to be recommended that only the variables that are actually needed by the **OPC UA Client** are provided through the **OPC UA Server**. Several instances of the **OPC UA Process Gateway** can also be started, with a different TCP port and a different configuration in terms of variables.

The driver tries to read the active configuration from the given INI file. If the configuration contains error, or an error occurs on starting, the configuration dialog (à la page 278) is opened and an error message is shown with an error text. If the configuration is not confirmed with **OK**, the **Process Gateway** is closed.

Attention

*If the application is started as `invisible`, there is also no configuration dialog displayed. If the configuration is invalid or not present, the **Process Gateway** is automatically closed.*

13.4 Connection

The **OPC UA Process Gateway** only supports the **OPC UA TCP binary Protocol** and does not contain a **Discovery Server**. The client must also connect directly to the server. The connection address corresponds to the scheme: **opc.tcp://hostname.of.server:port**

The connection can be secured (à la page 292).

13.5 Security

The module uses the security procedures that correspond to **OPC UA** (signing and encrypting messages) as defined in the standard. Certificates can be checked or all accepted, depending on the setting (à la page 278).

SECURE CONNECTIONS

If the **Accept secure connections only** (à la page 279) option is activated, the server only accepts connections that use the **Security Mode** `Sign` or `Sign & Encrypt`. In addition, the **Security Policy** used by the client must be `Basic128Rsa15` or `Basic256`. If this option is deactivated, any desired combination of **Security Mode** and **Security Policy** is accepted.

USER ADMINISTRATION

If user administration is activated in the zenon project and at least one user is created, the option **Use SCADA user authentication** (à la page 279) is available. If this option is activated, only clients that send a valid combination of `user` and `password` have access to the **OPC UA Server**data. No check is carried out if this option is inactive. In this case, both anonymous connections and also all user/password combinations are accepted.

13.6 Alarms and Conditions

The **OPC UA Process Gateway** also supports **OPC UA** Alarms and Conditions. When starting, the **OPC UA Server** automatically checks the configured variables for existing limit values or a reaction matrix, with conditions that are defined as an alarm.

An **OPC UA Client** can create a `Subscription` for `Event Notifications` and add the `Server Object` . The **OPC UA Client** can thus be informed of limit value breaches of a variable with an alarm active. The client is also in a position to confirm the alarm, as in the zenon Runtime.

Attention: The comments for alarms for the **OPC UA** method call for acknowledgment is no longer submitted to zenon Runtime. If an alarm is configured in zenon with "Comment required", an **OPC UA Client** cannot confirm this alarm with a comment.

Attention: If an alarm is configured so that an alarm cause is required, the alarm cannot be acknowledged on the **OPC UA** side.

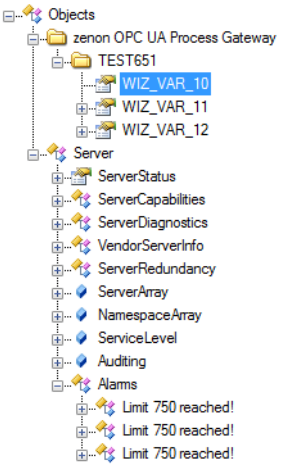
The standard-compliant OPC weighting is shown on the index of the alarm/event classes linked to the limit value. The index of an alarm/event class is automatically incremented by the zenon Editor with each new alarm/event class and can be checked in the properties. Up to 20 classes with the following allocation are supported:

OPC UA range	zenon alarm/event class	OPC weighting
HIGH	20	1000
	19	950
	18	900
	17	850
MEDIUM HIGH	16	800
	15	750
	14	700
	13	650
MEDIUM	12	600
	11	550
	10	500
	9	450
MEDIUM LOW	8	400
	7	350
	6	300
	5	250
LOW	4	200
	3	150
	2	100
	1	50

If a variable with no alarm/event class is linked, the lowest possible classification `1 (LOW)` is used.

ILLUSTRATION ZENON ALARM AS OBJECT

zenon variables and related alarms are illustrated in the **OPC UA Server** as objects.



The illustration shows the inclusion of the zenon project (`TEST651`) and the zenon variables (`WIZ_VAR_10, WIZ_VAR_11, WIZ_VAR_12`) as objects in the upper area. Below this are the alarms that have been generated when the limit values are reached (`Limit 750 reached!` etc.).

ALARM CONDITION NOTE IDENTIFIER

The **Node Identifier** type of the `Alarm Condition` is a **String Type Node ID**.

The following structure is used:

```
<project GUID>-<variable ID hexadecimal>@<variable name>:_<limit value
description or "Rema">
```

About the INI entry: USE_LIMIT_TEXT_FOR_ALARM_NODES=TRUE in the OPCUA section in the ini file for the **OPC UA Process Gateway**, an alternative structure can be activated where the limit value text is used. This alternative structure is only available for limit values and not for alarms via a reaction matrix. A dynamic limit value text is not supported in doing so. It must be ensured that the same limit value text is not used for several limit value texts for the same variable, because otherwise there is no uniqueness:

```
<project GUID>-<variable ID hexadecimal>@<static limit value text>
```

LOGGING

The driver supports a comprehensive LOG function. These LOG entries can be evaluated with the Diagnosis Viewer tool.

13.7 Service Sets

Service Set	Is supported
Discovery Service Set	
FindServers	yes
GetEndpoints	yes
RegisterServer	no
SecureChannel Service Set	
OpenSecureChannel	Yes
CloseSecureChannel	Yes
Session Service Set	
CreateSession	yes
ActivateSession	yes
CloseSession	yes
Cancel	yes
NodeManagement Service Set	
AddNodes	no
AddReferences	no
DeleteNodes	no
DeleteReferences	no
View Service Set	
Browse	yes
BrowseNext	yes
TranslateBrowsePathsToNodeIds	yes
RegisterNodes	yes
UnregisterNodes	yes
Query Service Set	
QueryFirst	no
QueryNext	no
Attribute Service Set	
Read	yes

HistoryRead	yes
Write	yes
HistoryUpdate	no
Method Service Set	
Call	yes
MonitoredItem Service Set	
CreateScreeenedItems	yes
ModifyScreenedItems	yes
SetMonitoringMode	yes
SetTriggering	yes
DeleteScreenedItems	yes
Subscription Service Set	
CreateSubscription	yes
ModifySubscription	yes
SetPublishingMode	yes
Publish	yes
Republish	yes
TransferSubscriptions	yes
DeleteSubscriptions	yes

13.8 Communication

SUBSCRIPTIONS

The **OPC UA Server** supports communication by means of Subscriptions. Several Subscriptions with different settings can be created per session by a client. The maximum number of Subscriptions per Session is limited to 10 by default. The "MAX_COUNT_OF_SUBSCRIPTIONS_PER_SESSION" ini entry in the "OPCUA" section can be used to configure the maximum number between "1" and "100".

PUBLISHREQUESTS

A maximum of 10 PublishRequests for the queue are permitted by the **OPC UA Server** per session. This depends on the number of Subscriptions.

- Sequence for publish requests (first in, first out):
  When sending, the incoming sequence of the PublishRequest is taken into account. If a Subscription is executed and a PublishResponse must be sent for a DataChangeNotification or a KeepAlive, the oldest PublishRequest from the queue is always taken.
- BadNoSubscription for all publish requests if the last Subscription has been deleted or the session has been closed:
  If the last Subscription has been logged off from the server, for PublishRequests that are still logged onto the server, PublishResponse with the value OpcUa_BadNoSubscription is created and sent to the client. The open PublishResponses are then deleted on the server.
- timeoutHint for PublishRequests:
  Before the PublishResponses for the Subscriptions are sent, it is ensured that only valid - i.e. not expired PublishRequests are present. The validity is checked using the time stamp of the PublishRequests as well as the timeoutHint stated in the Request. The resulting end time must be less than the current time of the server. If this is not the case, a Response with OpcUa_BadTimeout is sent back to the client. The Request is deleted in this case.
- SubscriptionAcknowledgement that is included in the PublishRequest processes the **OPC UA Server** if the PublishRequest is taken from the queue

REPUBLISHREQUESTS

The **OPC UA Server** also supports the RepublishRequests in the event that the TCP connection between **OPC UA Client** and **OPC UA Server** is interrupted.

SYNCHRONOUS TIME

For PublishRequests, the OPC UA server takes the "timeoutHint" field into account.

If the OPC UA server has to send a PublishResponse, all existing PublishRequests in the queue (max. 10) are checked. This check validates whether the time stamp is more recent that the current time. If this is the case, the oldest PublishRequest from the queue is used.

If the queue contains PublishRequests whereby the time stamp in the PublishRequest + timeoutHint in the PublishReuest is the same or older then the current time in the OPC UA server, the PublishRequests are removed from the queue and for each PublishRequest, a ServiceFault with "Bad_Timeout" is sent.

It is therefore important that the system time of the OPC UA server and the system time of the OPC UA client is synchronized accordingly. For this time synchronization, it is recommended that, for both the OPC UA server and the OPC UA client, a common synchronization source is used. A common source can,

for example, be the same NTP server for server and client. Otherwise it is possible that the OPC UA server immediately rejects all `PublishRequests` from the OPC UA client and no value changes can be transferred.

Informations

THE TIME STAMP NAME CONSISTS OF: `PUBLISHREQUEST` + `TIMEOUTHINT` FROM THE `PUBLISHREQUEST`.

MONITORED ITEMS

The OPC UA server supports the adding, removal and modification of `MonitoredItems`.

If a queue of greater than 1 is requested from a OPC UA client, this request is supported. As a result, several value changes for the same variable are buffered between two publish intervals by the OPC UA server. A `PublishResponse` is used to send the value changes to the OPC UA client.

DATACHANGEFILTER - DEADBAND

The OPC UA server supports the `absolute deadband`.

READ REQUESTS

The OPC UA server also supports, in addition to communication by means of `Subscriptions`, `Read Requests`. In principle, communication by means of `Subscriptions` is preferred to communication by means of read requests. If an OPC UA client does not support subscriptions and sends read requests cyclically in order to get the current value, it is strongly recommended that the client uses the RegisterNodes call once for such nodes that are read cyclically.

This allows the OPC UA server to optimize communication with Runtime for cyclical `Read Requests`.

WRITE REQUESTS

The OPC UA server supports write requests. The set value limits of the variable are taken into account.

Attention: If a variable in the Energy Edition is linked to a command group and this variable is written by an OPC UA client, the command interlocking is not taken into account but the command variable is written directly.

13.9 Configuration file: INI entries for OPC UA

The configuration file `zenProcGateway.ini` contains this entries for OPC UA:

[OPCUA]

Entrée	Description
SERVER_NAME=	Name of the **OPC UA Server**to which the Gateway establishes a connection.
SERVER_PORT=	Port pour la communication vers le **serveur OPC UA**. Par défaut : `4841`
CERTIFICATES_PATH=	Chemin d'accès pour le dossier racine avec les fichiers de certificat et les sous-dossiers respectifs. Par défaut : `%ProgramData%\PKI\CA\` Remarque : L'entrée doit être supprimée si un nouveau certificat doit être créé au démarrage de Process Gateway.
SERVER_CERTIFICATE_LOCATION=	Fichier contenant la clé publique du serveur. Le chemin est déclaré absolu. Cette clé publique est utilisée par le client pour le cryptage des messages. Par défaut : `%ProgramData%\PKI\CA\certs\zenopcua.der` Remarque : L'entrée doit être supprimée si un nouveau certificat doit être créé au démarrage de Process Gateway.
SERVER_PRIVATE_KEY_LOCATION=	Fichier contenant la clé privée du serveur. Le chemin est déclaré absolu. Clé privée est utilisée par le serveur pour déchiffrer les messages provenant du client. Par défaut : `%ProgramData%\PKI\CA\private\zenopcua.pem` Remarque : L'entrée doit être supprimée si un nouveau certificat doit être créé au démarrage de Process Gateway.
ONLY_SECURE_CONNECTIONS=	Sélection du type de connexion (sécurisée ou non sécurisée) approuvé pour l'établissement d'une connexion. ▸ `TRUE` Une connexion au serveur n'est possible qu'avec une connexion sécurisée. ▸ `FALSE` Une connexion au serveur est également possible avec une connexion non sécurisée. Par défaut : `TRUE`

TRUST_ALL_CLIENT_CERTIFICATES=	Sélection du type de certificats (certificats client ou serveur) approuvé pour l'établissement d'une connexion. ▸ TRUE tous les certificats de clients sont acceptés, et la connexion est établie. ▸ FALSE tous les clients souhaitant se connecter au serveur doivent transmettre un certificat valide. Ce certificat est également vérifié pour déterminer s'il se trouve dans la liste des certificats de clients connus (**Certificate trust list path**). Par défaut : TRUE
USE_USER_AUTHENTICATION=	Permet de choisir si l'administration des utilisateurs du projet zenon en cours doit être utilisée pour l'identification lors de la connexion au serveur. ▸ TRUE Les clients peuvent uniquement se connecter au serveur avec un identifiant d'utilisateur et un mot de passe valides. Remarque : Pour ce faire, l' administration des utilisateurs doit être activée et configurée dans l'éditeur zenon . ▸ FALSE Les connexions anonymes sont autorisées. Par défaut : FALSE
MAX_COUNT_OF_SUBSCRIPTIONS_PER_SESSION=	Nombre de Subscriptions par session. Peut être réglé entre un minimum de 1 et un maximum de 100 Subscriptions. Par défaut : 10

[VARIABLES]

Entrée	Description
PROJECT_x=	Nom du projet duquel sont issues les variables. **x** représente la numérotation du projet, qui commence par 0. Exemple : **PROJECT_0=**TEST
PARAM_x=	Paramètres du projet **x**. La numérotation commence à partir de 0. Exemple : **PARAM_0=**80:INI.Driverforinternalvariables_ModemHwAdrSet

COUNT=	Nombre de variables configurées.
NAME_x=	Nom de la variable. **x** représente la numérotation du projet, qui commence par `0`. Exemple : **NAME_0=**`Driverforinternalvariables_ModemHwAdrSet`

2 = constantes pour l'index de l'espace de nom

Ces valeurs sont définies par le système et ne doivent pas être modifiées !

13.10 Error message

Error messages in the log file:

Error message	Debug Level	Description
`Failed to validate client certificate`	**Error**	The client certificate could not be validated. It is either invalid (the date of validity may have been overwritten) or it was not found in the list of known client certificates. Additional information including an error number describes the error in more detail.
`Wrong user/password supplied while trying to activate session (User: %s)`	**Error**	User authentication failed. ▸ `%s`: Name of the user to be authenticated.
`Login of user '%s' succeeded.`	**Deep Debug**	User authentication successful. ▸ `%s`: Name of the user to be authenticated.
`Variable '%s' could not be added to OPC UA address space, because node '%s' with the same node identifier already exists`	**Error**	A variable defined by the user could not be created, because another variable with the same `Identifier` already exists.
`Failed to initialize server architecture: An internal error occurred as a result of a programming or configuration error. (0x80020000)`	**Error**	The file format for one or more of the following files is not present in a valid format: ▸ Certificate ▸ `Private Key` ▸ `Certificate Revocation List`

%1 PublishResponse with OpcUa_BadNoScubscriptions sent because no subscriptions are created in session %2	**Error**	There is no valid Subscription for a publish request or the session was deleted. The OPC UA server sends a messages to the OPC UA client for all active PublishRequests if the last subscription was deleted by the server or the client. ▸ %1 Number of PublishResponses that have been sent to a client with the result value OpcUa_BadNoSubscription. ▸ %2 Name of the session from which the PublishResponse has been sent.
%1 PublishResponse with OpcUa_BadTimeout sent in session %2	**Error**	The timestamp + timeoutHint in the PublishRequest that is provided by the client is older than the current time on the OPC UA server. Possible causes: the timeoutHint in the PublishRequest is too short, the OPC UA client provides more PublishRequests than are needed and the PublishRequests expire or the time of the computer with the OPC UA server and the time of the computer with the OPC UA client are not synchronous. ▸ %1 Number of PublishResponses that have been sent to a client with the result value OpcUa_BadTimeout. ▸ %2 Name of the session from which the PublishResponse has been sent. Note: The PublishRequest is deleted.
%1 advise/unadvise thread procedure	**DEBUG**	Start or stop of an Advise or Unadvise thread: ▸ %1 Entering Thread started ▸ %1 Leaving Thread stopped
%1 shutdown	**DEBUG**	Start and end of a shutdown sequence: ▸ %1 beginning Shutdown start ▸ %1 finisched Shutdown ended
%1 %2 request	**DEBUG**	Start or end of a Request when reloading Runtime: ▸ %1 Processing

		Start of a request ▶ %1 Finished End of a request ▶ %2 ACTIVE Request is active. ▶ % 2 INACTIVE Request is inactive.
Server shut down.	**DEBUG**	Notification that the server has been fully shut down.
The runtime does not provide archive variable [%1] of project [%2]	**ERROR**	The archive of a configured archive variable can no longer be reached in zenon Runtime. ▶ %1 Name of the archive variable ▶ %2 Project name
Reading configuration from %1	**MSG**	The configuration is read from the INI file. ▶ %1 Name of the INI file.
Linking configuration to the Runtime	**MSG**	The variables configured in **Process Gateway** are linked to zenon Runtime.
some configured variables are not linked to the Runtime	**ERROR**	When linking, it is established that one or more configured variables are not available in zenon Runtime. The projects or variables that are not available are listed.
Project [%1] is not linked to the Runtime	**MSG**	The project configured with the variables is not available in zenon Runtime. ▶ %1 Project name Possible causes: The project is not loaded or has been deleted.
Variable [%1] of Project [%2] is not linked to the Runtime	**MSG**	The variable is not available in zenon Runtime: ▶ 1% Variable name ▶ %2 Project name
Could not write value [%1] to key [%2] in section [%3]	**ERROR**	Error when writing a value. ▶ %1

		Value that is to be written. ▶ %2 Key in which the value is to be written. ▶ %3 Section of the key in which the value is to be written.
No separator for archive flag [,] found in value [%s] of key [%s]	**ERROR**	Missing separator character for a value. The variable concerned can therefore not be applied in the configuration: ▶ %1 Variable name ▶ %2 Key of the value

13.11 Status mapping

In certain situations, the OPC UA server communicates a different StatusCode for the value (Value Attribute). This concerns each node that represents the variable.

The table below lists the zenon status with the corresponding **OPC UA StatusCode**.

Status of the variable in the Runtime	StatusCode Value Attribute OPC UA Node
INVALID	0x80050000 BadCommunicationError
ALT_VAL	0x40910000 UncertainSubstituteValue
OFF	0x808C0000 BadSensorFailure
OV_870	0x803C0000 BadOutOfRange
N_UPDATE	0x80310000 BadNoCommunication

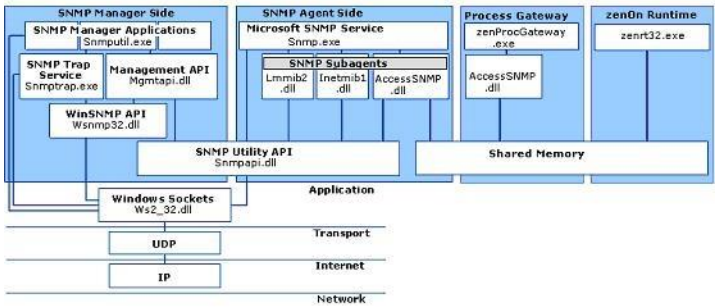
14. Agent SNMP

14.1 Architecture SNMP

L'agent SNMP Windows supporte le SNMP version 2 (SNMPv2c). Il supporte aussi les fonctions SNMP (GET, GETNEXT, GETBULK, SET, TRAP).

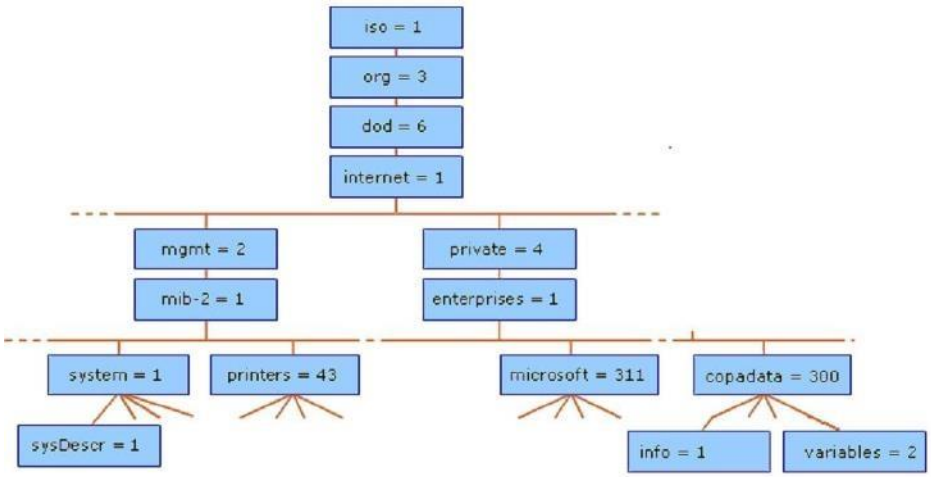
Vous trouverez plus de détails sur l'implémentation SNMP de Microsoft sur le site Web TechNet de Microsoft : " https://technet.microsoft.com/en-us/library/cc783142(v=ws.10).aspx (https://technet.microsoft.com/en-us/library/cc783142(v=ws.10).aspx)

LES COMPOSANTS DU SYSTÈME D'EXPLOITATION WINDOWS AVEC UNE CONNEXION AU PROCESS GATEWAY ET AU RUNTIME :



14.2 Management Information Base (MIB)

Management Information Base (MIB) est une structure d'arborescence dans laquelle l'agent SNMP fournit des variables (OID, Object Identifier). Ci-dessous se trouve la structure de la MIB, avec des variables supplémentaires visibles, qui ont été fournies par le Process Gateway.



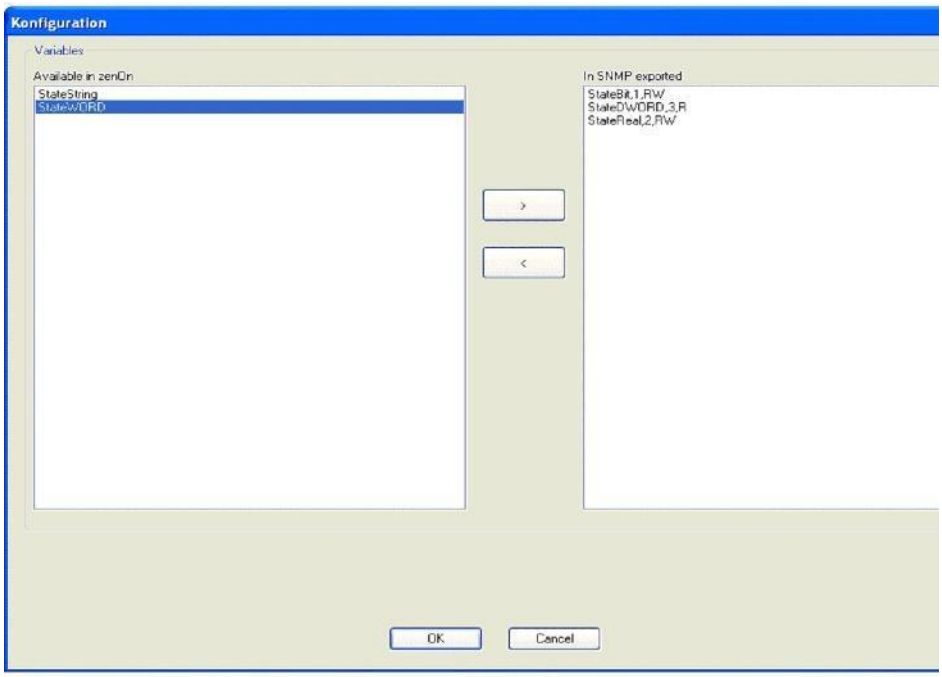
14.3 Structure MIB

la structure Info block est toujours présente				
OID	**Description**	**Type de données**	**Accès**	**Modification envoyée en tant que 'Trap'**
1.3.6.1.4.1.300.1.1.0	copadata.info.statusinfo, le Runtime est actif ou inactif (Online, Offline)	String	lecture	oui
1.3.6.1.4.1.300.1.2.0	copadata.info.statusval, le Runtime est 1=actif, 0=inactif	Integer	lecture	oui
1.3.6.1.4.1.300.1.3.0	copadata.info.watchdog	Integer	lecture	non
1.3.6.1.4.1.300.1.4.0	copadata.info.project, nom du projet où le Process Gateway lit les variables	String	lecture	non

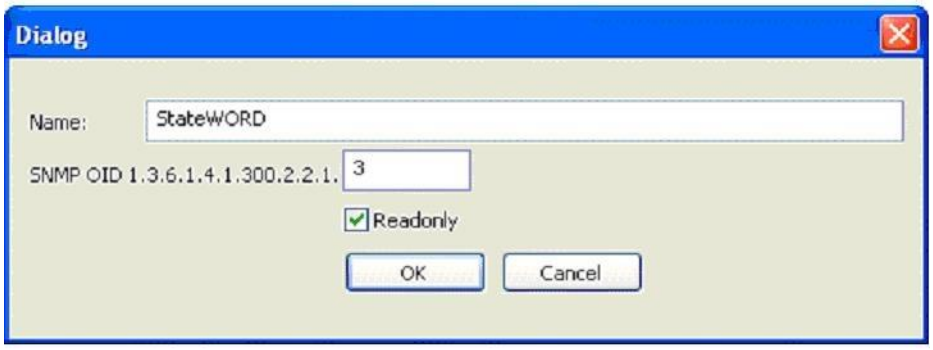
Structure des variables (table) ajustée dynamiquement, selon la configuration du Process Gateway				
OID	**Description**	**Type de données**	**Accès**	**Modification envoyée en tant que 'Trap'**
1.3.6.1.4.1.300.2.1.0	copadata.variables.ifNumber, nombre de variables OID configurées	Integer	lecture	oui
1.3.6.1.4.1.300.2.2.1.X	copadata.variables.ifTables.ifIndex, index OID configuré dans la configuration du Process Gateway	Integer	lecture	non
1.3.6.1.4.1.300.2.2.2.X	copadata.variables.ifTables.ifName, nom de la variable	String	lecture	non
1.3.6.1.4.1.300.2.2.3.X	copadata.variables.ifTables.ifValue, valeur entière de la variable. Tous les types numériques sont formatés comme le type SNMP Integer (entier). ATTENTION ! Les positions des décimales pour les REAL (réels) et DOUBLE sont perdus.	Integer	selon la configuration du Process Gateway	non
1.3.6.1.4.1.300.2.2.4.X	copadata.variables.ifTables.ifValueAsString, valeur chaîne (String) de la variable. Tous les types numériques sont formatés comme le type SNMP String (chaîne).	String	selon la configuration du Process Gateway	oui
1.3.6.1.4.1.300.2.2.5.X	copadata.variables.ifTables.ifStatus, valeur de l'état de la variable	Integer	lecture	oui
1.3.6.1.4.1.300.2.2.6.X	copadata.variables.ifTables.ifTimestamp, horodatage de la variable	Integer	lecture	oui

14.4 Configuration du module

La boîte de dialogue suivante apparaît lorsque vous cliquez sur le bouton **Settings** :



Vous pouvez sélectionner ici les variables qui doivent être connectées. Dans la partie gauche de l'écran, toutes les variables disponibles dans zenon sont affichées. Vous pouvez les sélectionner ici et les déplacer vers la liste d'exportation SNMP en cliquant sur le bouton **>**. Le bouton **<** permet de les supprimer de la liste. Si vous déplacez une variable dans la liste d'export, une boîte de dialogue s'ouvre permettant de définir le lien entre le nom de la variable zenon et une adresse OID dans la structure SNMP :



Les variables sélectionnées de cette façon sont alors disponibles sur le serveur SNMP et sont synchronisées de façon continue entre zenon et la structure SNMP OID. Pour chaque adresse OID, vous pouvez définir si elle est accessible en écriture via SNMP SET. La structure exacte pour chaque adresse OID est décrite dans la section "Structure MIB".

14.5 fichier de configuration : entrées spécifiques pour AccessSNMP

Le fichier de configuration doit se trouver dans le dossier C:\Documents and Settings\All Users\Application Data\COPA-DATA\System.

Paramètres :	Description
[SNMP]	
REFRESHRATE	Période de rafraichissement, en ms

Paramètres :	Description
[VARIABLES]	
COUNT	Nombre de variables à exporter
NAME_n	Nom de la variable comportant le nombre n (0<= n<COUNT)
PARAM_n	Le paramètre SNMP OID table offset et l'information R=read only (lecture seule) / RW=read and write (lecture et écriture) avec le nombre n (0<=n<COUNT)

EXPORTATION DE VARIABLES

Pour que l'exportation se déroule correctement, le nom du projet doit être saisi correctement dans le fichier `zenProcgateway.ini` du dossier %CD_SYSTEM%. Saisissez-le ici :

```
[DEFAULT]
PROJECT=
```

Attention :Si l'entrée est erronée ou manquante, la passerelle est fermée sans message d'erreur. La variable n'est donc pas disponible.

14.6 Configuration de l'agent SNMP Windows

Pour vous assurer que l'agent SNMP de Windows (**SNMP.exe**) fournit la structure partielle MIB de zenon, vous devez configurer le fichier **AccessSNMP.dll** de la passerelle de procédé (Process Gateway) dans la base de registres Windows comme suit :

1. Installation du service Windows SNMP par le Panneau de Configuration Windows :
   - Via le module : Programmes et fonctionnalités -> Activer ou désactiver des fonctionnalités Windows.

   Remarque : Sous Windows 8, l'ordinateur doit être redémarré après l'ajout du service SNMP.

2. Installation de Process Gateway avec le fichier `AccessSNMP.dll` dans le dossier `%Program Files (x86)%\COPA-DATA\zenon 8.00`.
3. Ajoutez les clés et chaînes de caractères suivantes à l'aide de l'éditeur de la base de registres. Les chemins et les contenus diffèrent parfois sur les systèmes 322 bits et 64 bits :

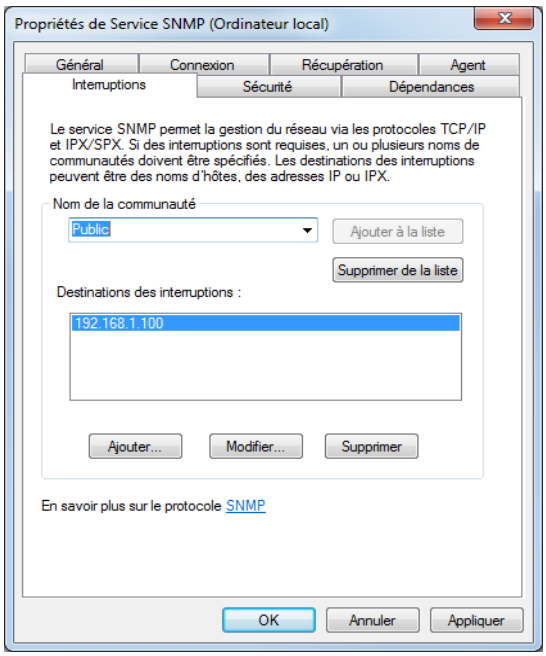
   32 bits :

   - **[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SNMP\Parameters\ExtensionAgents]**
     Name=`ProcessGateway`
     Wert=`SOFTWARE\COPA-DATA\SNMP\CurrentVersion`
   - **[HKEY_LOCAL_MACHINE\SOFTWARE\COPA-DATA\SNMP\CurrentVersion]**
     Name=`Pathname`
     Valeur=`C:\Program Files\COPA-DATA\zenon 8.00 SP0\AccessSNMP.dll`

   64 bits :

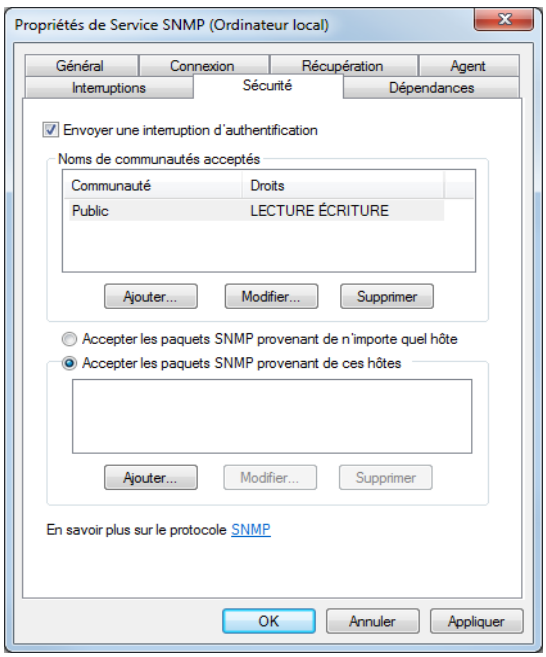
   - **[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SNMP\Parameters\ExtensionAgents]**
     Name=`ProcessGateway`
     Wert=`SOFTWARE\COPA-DATA\SNMP\CurrentVersion`
   - **[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\COPA-DATA\SNMP\CurrentVersion]**
     Name=`Pathname`
     Valeur=`C:\Program Files (x86)\COPA-DATA\zenon 8.00 SP0\AccessSNMP.dll`
4. Démarrer le service SNMP.
   Ceci permet de charger le fichier de la passerelle Process Gateway `AccessSNMP.dll` via la configuration de la base de registres.
   Autres paramètres de configuration du service Windows SNMP : Panneau de configuration -> Administration-> Services-> Service SNMP -> Propriétés

a) **Onglet Interruptions :** définit l'adresse IP du réseau à laquelle sont transmises les interruptions



- **Nom de la communauté** : doit être identique sur le gestionnaire et l'agent
- **Destinations des interruptions** : adresses des destinations sous forme de noms d'hôtes, d'adresses IP ou d'adresses IPX

a) **Sécurité** définit les droits avec lesquels un client SNMP peut accéder à l'agent SNMP



- **Droits** : doit être défini sur "`READ WRITE (LECTURE ÉCRITURE)`"

- **Nom de la communauté** : doit être identique sur le gestionnaire et l'agent

15. SQL

15.1 Configuration du module

La configuration de la connexion à la base de données est faite en deux parties :

1. Connexion OLE-DB : Cette connexion OLE-DB peut être définie dans la boîte de dialogue de configuration à l'aide du bouton de sélection de la chaîne de connexion. Ensuite la boîte de dialogue standard pour la connexion de données est ouverte.
2. En plus, le nom de la table peut être défini si vous ne voulez pas utiliser le nom standard ONLINE_VALUES.
   Ceci se définit par l'entrée **TABLE** dans la section **[DATABASE]** du fichier de configuration (`zenProcGateway.ini`). Les paramètres sont pris en compte par l'outil lors du démarrage de celui-ci.

> **Attention**
>
> *Si des noms de tables sont définis manuellement dans le fichier de configuration (zenProcGateway.ini), ces noms de tables doivent également être modifiés ou créés manuellement dans la base de données.*

BASES DE DONNÉES ORACLE

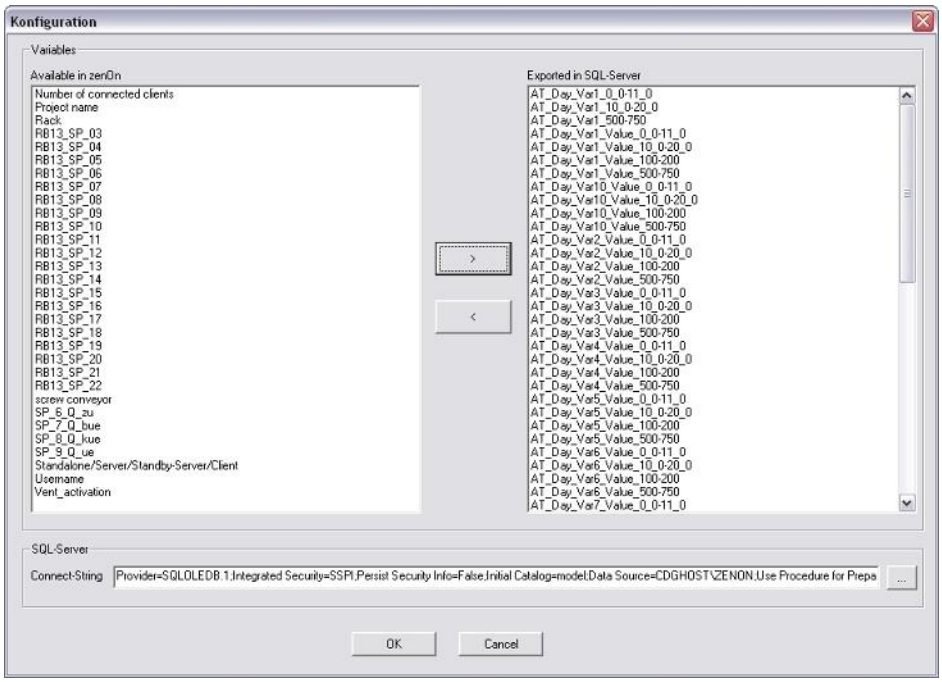
L'utilisation de la passerelle Process Gateway et du fichier `AccessSQL.dll` conjointement à des bases de données Oracle peut provoquer des erreurs : il est possible, avec certains drivers ODBC, qu'aucune valeur en ligne ne soit écrite dans la base de données. La fonction de journalisation avancée d'ODBC produit alors le message d'erreur **1843**.

Le problème peut être résolu en changeant de driver ODBC. Exemple : ODBC driver 9.02.00.05 for Oracle 9.2.

15.2 Sélection des variables

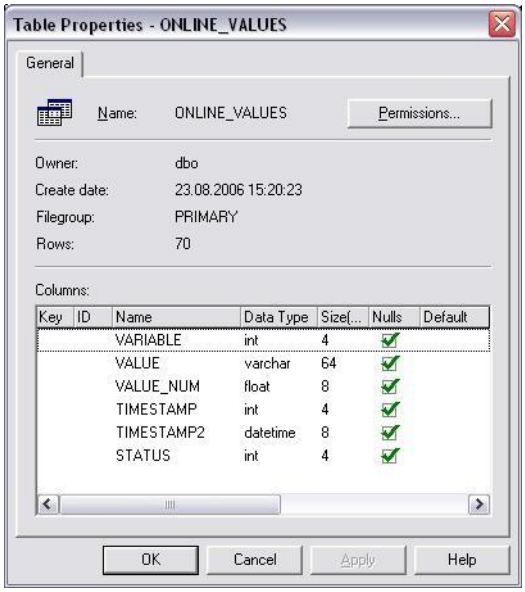
Dans la boîte de dialogue de configuration, les variables zenon devant être écrites dans l'image du procédé SQL peuvent être définies. À cette fin, les variables disponibles dans le projet zenon (dont la

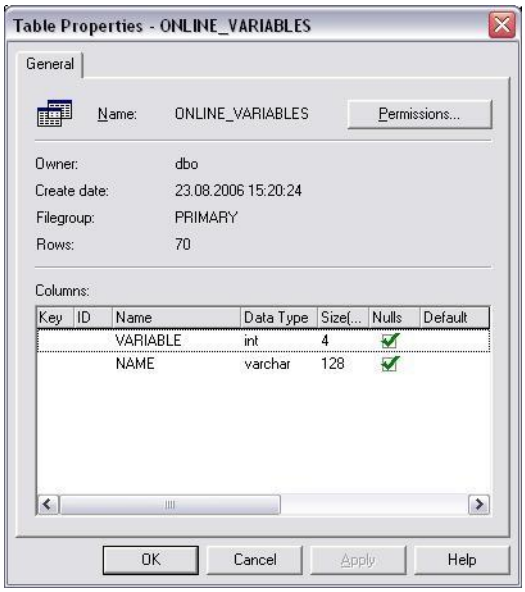
liste est affichée dans la partie gauche de l'écran) doivent être ajoutées à l'image (affichée dans la partie droite de l'écran).



15.3 Création des tables

Lorsque vous fermez la boîte de dialogue de configuration en cliquant sur **OK**, la table dans la base de données SQL est générée automatiquement. Dans le cas où elle doit être créée manuellement :





15.4 Cas d'un système redondant

Dans le cas de système redondant, vous avez deux solutions. Dans la première, la base de données SQL s'exécute localement sur les deux serveurs, ainsi que le module Process Gateway. Comme l'outil ne gère

que les données en ligne, il n'y a pas de synchronisation des bases à faire. Le gros inconvénient de cette solution est la charge importante du réseau qui résulte du fait de la lecture des données en ligne supplémentaires faites par le serveur redondant.

Dans la deuxième solution, le module Process Gateway s'exécute uniquement sur le serveur principal et écrit les données dans une base de données SQL locale ou distante. Avec cette solution, il n'y a pas de charge supplémentaire sur le réseau entre le serveur principal et le serveur redondant, et dans le cas d'une base SQL distante, les performances du serveur ne sont pas affectées par les requêtes externes à cette base.

15.5 Fichier de configuration : entrées spécifiques pour AccessSQL

Le fichier de configuration doit se trouver dans le dossier C:\Documents and Settings\All Users\Application Data\COPA-DATA\System.

Paramètre	Description
[DATABASE]	
INITSTRING	Chaîne de connexion OLE-DB vers la base de données SQL
TABLE_ONL	nom de la table qui reçoit l'image du procédé
TABLE_VAR	nom de la table qui reçoit les ID des variables
REFRESHRATE	temps maximum de rafraîchissement en millisecondes

Paramètre	Description
[VARIABLES]	
COUNT	Nombre de variables à exporter
NAME_n	Nom de la variable numéro n. Règle ▸ commence par 0 ▸ Doit être consécutif sans interruption

15.6 Format des tables

15.6.1 Table pour l'image du procédé

VARIABLE	int[4]	ID de la variable de zenon
VALUE	varchar[64]	valeur courante en temps que chaîne (string)
VALUE_NUM	float	valeur actuelle sous forme de valeur réelle (float)
TIMESTAMP	int[4]	horodatage de la valeur courante (en date/heure UNIX)
TIMESTAMP2	datetime	horodatage de la valeur courante (au format date/heure)
STATUS	int[4]	état de la valeur courante

15.6.2 Table des ID de variables

VARIABLE	int[4]	ID de la variable de zenon
NAME	varchar[128]	Nom de la variable de zenon