Unity Application Generator (UAG) 3.3 SP4

a SoCollaborative software

Manuel utilisateur de base

09/2014

33002831.13

www.schneider-electric.com

Schneider Electric

Le présent document comprend des descriptions générales et/ou des caractéristiques techniques des produits mentionnés. Il ne peut pas être utilisé pour définir ou déterminer l'adéquation ou la fiabilité de ces produits pour des applications utilisateur spécifiques. Il incombe à chaque utilisateur ou intégrateur de réaliser l'analyse de risques complète et appropriée, l'évaluation et le test des produits pour ce qui est de l'application à utiliser et de l'exécution de cette application. Ni la société Schneider Electric ni aucune de ses sociétés affiliées ou filiales ne peuvent être tenues pour responsables de la mauvaise utilisation des informations contenues dans le présent document. Si vous avez des suggestions, des améliorations ou des corrections à apporter à cette publication, veuillez nous en informer.

Aucune partie de ce document ne peut être reproduite sous quelque forme ou par quelque moyen que ce soit, électronique, mécanique ou photocopie, sans autorisation préalable de Schneider Electric.

Toutes les réglementations de sécurité pertinentes locales doivent être observées lors de l'installation et de l'utilisation de ce produit. Pour des raisons de sécurité et afin de garantir la conformité aux données système documentées, seul le fabricant est habilité à effectuer des réparations sur les composants.

Lorsque des équipements sont utilisés pour des applications présentant des exigences techniques de sécurité, suivez les instructions appropriées.

La non-utilisation du logiciel Schneider Electric ou d'un logiciel approuvé avec nos produits matériels peut entraîner des blessures, des dommages ou un fonctionnement incorrect.

Le non-respect de cette consigne peut entraîner des lésions corporelles ou des dommages matériels.

© 2014 Schneider Electric. Tous droits réservés.

Table des matières

	Consignes de sécurité	**11**
	A propos de ce manuel	**13**
Partie I	**Comprendre Unity Application Generator**	**15**
Chapitre 1	**Configuration et procédure de mise à niveau**	**17**
	Configuration matérielle minimale	18
	Configuration logicielle minimale	19
	Informations d'installation	22
	Mise à niveau des projets existants vers UAG et Unity Pro	25
Chapitre 2	**Introduction à Unity Application Generator**	**27**
	Informations générales concernant la sécurité	28
	Qu'est-ce que Unity Application Generator ?	30
	Les fondations de Unity Application Generator	33
	Création d'une application	37
Chapitre 3	**Sécurité des accès UAG**	**39**
	Gestion de la sécurité d'accès à UAG	40
	Éditeur de sécurité	42
	Informations utilisateur	43
	Fonctions utilisateur	45
	Profils utilisateur	49
	Règles	54
	Procédure de reprise après détection d'une erreur	56
	Droits d'accès	57
Chapitre 4	**Modèle physique**	**59**
4.1	Présentation du modèle physique	60
	Structure du modèle physique	60
4.2	Eléments physiques en général	64
	Éléments physiques, description et propriétés	64
4.3	Elément physique fonctionnel	66
	Qu'est-ce qu'un élément physique fonctionnel ?	67
	Caractéristiques de l'élément physique fonctionnel	68
	Propriétés de l'élément physique fonctionnel	70
	Types de modèles	72

4.4 L'élément Module de commande et les types de module de commande (modules de commande intelligents - SCoD) 73
Introduction 74
Caractéristiques d'un module de commande 75
Types de module de commande ou modules de commande intelligents (SCoD) 76
Module de commande libre 78
Propriétés d'un module de commande 79
Instruments 81
Forçages des modules de commande 82
Liaisons entre les SCoD 84
4.5 Variables 86
Introduction 87
Types de connexions et types de données de variables 88
Propriétés générales des variables 90
Propriétés des variables liées à l'API 99
Propriétés des variables relatives à l'IHM 102
Définition d'alarme dans UAG 108
Onglet Utilisations des propriétés de variables 114
Chapitre 5 Modèle topologique 117
5.1 Topologie d'un automatisme et modèle topologique 118
Topologie d'un automatisme 119
Structure du modèle topologique 124
Communication via Modbus, Modbus Plus et Ethernet 126
Autres informations relatives à Ethernet 129
Configuration de la redondance d'UC Quantum 132
Configuration de redondance d'UC Premium 135
5.2 Les groupes segments de réseau et chemins de routage 142
Description et propriétés des segments de réseau 143
Liste des abonnés réseau 144
Description et propriétés des chemins de routage 146
5.3 Le groupe API 148
Introduction 149
Propriétés de l'API 151
Canal API 156
Fonctions de l'API 159
Communication API <-> API via Modbus Plus 160
Communication API <-> API via Ethernet 162
Afficheur des communications 165

Châssis supplémentaires 167
Châssis et modules matériels 169
Informations générales sur les bus de terrain 177
Module Ethernet amélioré 179
Passerelle Modbus/Ethernet 182
Passerelle Profibus/Ethernet 188
Ethernet Quantum distant d'E/S (QERIO) 203
E/S M580 Ethernet 207
5.4 Groupe IHM 214
Introduction 215
Propriétés de l'IHM 216
Propriétés d'une zone de conduite 217
Suppression d'une IHM 218
Images et groupes d'images 219
5.5 Le groupe Serveur de données 221
Description générale 222
Propriétés du serveur de données 223
Onglet Général 225
Onglet Réseaux (serveur principal) 226
Onglet Réseaux (serveur secondaire) 228
Serveurs de données redondants 230
Onglet Générateur de serveur de données et d'IHM 234
Suppression d'un serveur de données 236
5.6 Le groupe Abonnés réseau 237
Description d'un partenaire réseau 238
Propriétés d'un partenaire réseau 240
Description des autres abonnés 241
Propriétés des autres abonnés 242
Partie II Travailler avec Unity Application Generator 243
Chapitre 6 Règles de travail avec Unity Application Generator . . 245
Règles générales pour la configuration et la génération de projet . . . 246
Ouverture des éditeurs SCoD et de personnalisation 247
Règles concernant l'API et Unity Pro 248
Règles concernant les IHM 250

Chapitre 7 Manipulation de l'outil et caractéristiques pour un travail efficace 251
7.1 Concepts généraux de l'interface utilisateur 252
Concepts de l'interface utilisateur 253
Utilisation des tables (listes). 255
Glisser-déposer des objets, modules et variables 257
Recherche d'objets à l'aide de critères de recherche 259
Utilisation de la liste d'instruments 261
Copier et coller dans les modèles physique et topologique 264
Etablissement d'une définition de forçage 266
Utilisation de l'afficheur topologique. 269
Comment visualiser l'état de la génération 272
Comment échanger les noms 273
Fenêtre - Raccourcis clavier 276
7.2 Editeur graphique de forçages 277
Description générale 278
Blocs fonctions (FFB). 283
Liaisons 287
Barre d'outils graphiques 289
Zone de dessin. 293
Analyse logique et syntaxique de la logique de forçage 297
Enregistrement et récupération de la logique de forçage 299
7.3 Afficheur de variables. 300
Description générale 301
Zone d'informations 303
Filtre 307
Grille des variables. 311
Chapitre 8 Schéma d'élaboration d'une application 317
Schéma global d'élaboration d'une application 318
Définition de la personnalisation 320
Définition du modèle physique 322
Définition du modèle topologique. 323
Définition du modèle physique 326
Génération d'applications API et IHM 327
Génération d'un fichier d'importation pour les partenaires réseau. . . . 329
Assistants de maintenance 330
Documentation de l'application ou d'objets individuels (Rapport) 347

Chapitre 9 Rubriques spécifiques aux bus de terrain 349
9.1 Informations générales sur les bus de terrain 350
Informations générales sur les bus de terrain 351
Modules de bus de terrain dans le modèle topologique 353
Description de l'interface utilisateur 354
Boîte de dialogue Esclave Profibus 357
9.2 Importation de la configuration de bus de terrain 359
Catalogue de modules Profibus 360
Importation de la configuration de bus de terrain 361
Génération d'une application API 367
9.3 Configuration de bus de terrain pour Quantum et Premium 368
Unity Pro Quantum 369
Unity Pro Premium 370
Chapitre 10 Intégration Advantys 371
Intégration des systèmes d'E/S distribuées Advantys STB 372
Importation de la configuration Advantys 373
Catalogue des modules Advantys 375
Affectation des entrées/sorties des modules Advantys 376
Génération du scrutateur d'E/S et de la diffusion des E/S pour le logiciel de programmation d'API 379
Génération de la logique FBD 380
Fonctionnalités améliorées 381
Partie III Gestion de projet 383
Chapitre 11 Gestion du développement de projets distribués – Fusion de projets 385
Présentation 386
Génération de code pour des API individuels 387
Schéma du développement de projet distribué 388
Conditions générales préalables à la fusion de projets 391
Fusion des modèles physiques 393
Fusion des modèles topologiques 394

Chapitre 12 Interfaces avec d'autres outils (fonctions Importer et Exporter) 397
12.1 Informations générales sur l'importation/exportation 398
Concept d'interfaces ouvertes 399
Exportation d'un projet UAG 401
Importer dans un projet UAG 403
Mode d'importation/exportation de fichiers CSV/XML 404
Importation incrémentielle 406
12.2 Exemples d'importation/Exportation CSV 407
Exemple : Importation d'instruments / de la hiérarchie du modèle physique 408
Exemple : importer des valeurs initiales de variables 411
12.3 Importer/exporter XML 413
Exportation depuis l'éditeur SCoD 413
Chapitre 13 Types de données dérivés 415
13.1 Notions fondamentales sur les types de données dérivés 416
Gestion des types de données dérivés (DDT) 416
13.2 DDT et l'éditeur SCoD 417
Importation de DDT dans l'éditeur SCoD 418
Utilisation des DDT dans l'éditeur SCoD 420
Définition de l'héritage avec des variables DDT 424
Connexion de variables DDT à des broches 426
Exemple d'importation de DDT pour l'éditeur SCoD 427
Règles d'affectation en mémoire 429
13.3 DDT et l'éditeur de personnalisation 430
Importation de DDT dans l'éditeur de personnalisation 431
Utilisation des DDT dans l'éditeur de personnalisation 432
13.4 DDT et l'éditeur UAG 433
Utilisation des variables DDT dans l'éditeur UAG 434
Diffusion des DDT entre les API 436
Affectation des variables DDT à des points d'E/S 437
13.5 Les DDT et les Interfaces Homme/Machine (IHM) 438
Transfert de variables DDT vers Vijeo Citect 439
Transfert de variables DDT vers Monitor Pro 440
Transfert de variables DDT vers une IHM générique 442
Annexes 443

Annexe A Format des fichiers CSV d'importation et d'exportation **445**
Format général **446**
Les éléments du modèle physique : format de fichier CSV **449**
Eléments du modèle topologique : format de fichier CSV **456**
Instruments : format de fichier CSV **469**
Valeurs initiales : format de fichier CSV **470**
Format d'exportation des bus de terrain **471**
Bibliothèque des types de modèles : format de fichier CSV **475**
Suppression et changement de nom : Format de fichier CSV **479**
Bibliothèque SCoD : format de fichier CSV **482**
Annexe B Format des fichiers XML d'importation et d'exportation **483**
B.1 Informations générales sur le format XML **484**
Informations générales sur le format XML **484**
B.2 Format XML pour un projet UAG **485**
Format XML des données d'en-tête **486**
Format XML des éléments physiques **487**
Format XML des éléments topologiques **492**
Format XML des listes d'instruments **500**
Règles de suppression et de changement de nom d'objets **501**
B.3 Format XML pour des bibliothèques de modèles **502**
Format XML des bibliothèques de modèles **502**
B.4 Format XML pour l'éditeur SCoD **503**
Format XML des données d'en-tête (SCoD) **504**
Format XML du type de module de commande (SCoD) **505**
Format XML des variables de type de module de commande (SCoD) **506**
Format XML des propriétés du module de commande, des FFBPins et des propriétés héritées (SCoD) **507**
Format XML des données générales (SCoD) **508**
Glossaire **509**
Index **519**

Consignes de sécurité

Informations importantes

AVIS

Lisez attentivement ces instructions et examinez le matériel pour vous familiariser avec l'appareil avant de tenter de l'installer, de le faire fonctionner ou d'assurer sa maintenance. Les messages spéciaux suivants que vous trouverez dans cette documentation ou sur l'appareil ont pour but de vous mettre en garde contre des risques potentiels ou d'attirer votre attention sur des informations qui clarifient ou simplifient une procédure.

La présence de ce symbole sur une étiquette "Danger" ou "Avertissement" signale un risque d'électrocution qui provoquera des blessures physiques en cas de non-respect des consignes de sécurité.

Ce symbole est le symbole d'alerte de sécurité. Il vous avertit d'un risque de blessures corporelles. Respectez scrupuleusement les consignes de sécurité associées à ce symbole pour éviter de vous blesser ou de mettre votre vie en danger.

⚠ DANGER

DANGER signale un risque qui, en cas de non-respect des consignes de sécurité, **provoque** la mort ou des blessures graves.

⚠ AVERTISSEMENT

AVERTISSEMENT signale un risque qui, en cas de non-respect des consignes de sécurité, **peut provoquer** la mort ou des blessures graves.

⚠ ATTENTION

ATTENTION signale un risque qui, en cas de non-respect des consignes de sécurité, **peut provoquer** des blessures légères ou moyennement graves.

AVIS

AVIS indique des pratiques n'entraînant pas de risques corporels.

REMARQUE IMPORTANTE

L'installation, l'utilisation, la réparation et la maintenance des équipements électriques doivent être assurées par du personnel qualifié uniquement. Schneider Electric décline toute responsabilité quant aux conséquences de l'utilisation de ce matériel.

Une personne qualifiée est une personne disposant de compétences et de connaissances dans le domaine de la construction, du fonctionnement et de l'installation des équipements électriques, et ayant suivi une formation en sécurité leur permettant d'identifier et d'éviter les risques encourus.

A propos de ce manuel

Présentation

Objectif du document

Ce manuel a pour objet :

- de vous permettre de comprendre ce qu'est l'application Unity Application Generator (UAG) et ce qu'elle peut faire pour vous ;
- de donner une vue d'ensemble des fonctionnalités de Unity Application Generator ;
- d'expliquer tous les éléments intervenant dans l'élaboration d'une application.

Ce manuel n'a, en revanche, **pas** pour objet :

- de vous donner une véritable "feuille de route" sur la manière de créer une application sous Unity Application Generator (UAG). Vous trouverez ces explications dans le manuel avancé de UAG.
- d'apporter une référence sur tous les éléments de UAG. Vous trouverez ces explications dans le manuel avancé de UAG.
- de détailler tous les menus et boîtes de dialogue de Unity Application Generator. Vous trouverez ces explications dans l'aide du logiciel.

Champ d'application

Ce manuel est valable pour Unity Application Generator (UAG) 3.3 SP4en association avec :

- Schneider Electric Unity Pro 4.1, 5.0, 6.0, 7.0, 8.0, and 8.1
- Schneider Electric Vijeo Citect 6.10, 7.0, 7.10, 7.20, 7.30, 7.40, and 7.40 SP1
- Wonderware ArchestrA 2012 R2 (non pris en charge en ce qui concerne les nouvelles fonctionnalités de la version 3.2 ou ultérieure de UAG)
- Schneider Electric Monitor Pro 7.6 (non pris en charge en ce qui concerne les nouvelles fonctionnalités de la version 3.2 ou ultérieure de UAG)
- Intellution iFIX 3.5 + 4.0 comprenant le pilote MBT V7.156k pour iFIX de chez Dimension Software Inc. (non pris en charge en ce qui concerne les nouvelles fonctionnalités de la version 3.2 ou ultérieure de UAG)
- Logiciel de configuration Advantys 4.7, 5.0, 5.5, 7.0, and 8.0 de Schneider Electric
- Prosoft Configuration Builder 4.0.0.0/4.1.0.0/4.1.1.1/4.1.2.3 (fourni sur le DVD d'installation d'UAG)
- Hilscher Sycon 2.10

Document(s) à consulter

Titre de documentation	Référence
Bibliothèque SCoD IATBASIC10 ; reportez-vous aux documents Word séparés fournis pour chaque SCoD dans le sous-répertoire DOC du répertoire Unity Application Generator	-
Manuel utilisateur avancé de Unity Application Generator 3.3 SP4	33003669
Manuel utilisateur de l'éditeur SCoD 3.3 SP4	33002608
Manuel utilisateur de Unity Application Generator (UAG) ArchestrA Generator	33004144
Guide de référence du logiciel Unity Pro	35056144
Manuel utilisateur du système de redondance d'UC Quantum avec Unity Pro	35010533
Manuel utilisateur du système de redondance d'UC Premium avec Unity Pro	35012068
D. W. Fleming, V. Pillai: S88 Guide d'application	McGraw Hill, ISBN 0-07-021697-5

Vous pouvez télécharger ces publications et autres informations techniques depuis notre site web à l'adresse : www.schneider-electric.com.

Information spécifique au produit

⚠ AVERTISSEMENT

FONCTIONNEMENT IMPREVU DE L'EQUIPEMENT

L'application de ce produit nécessite d'être compétent dans la conception et la programmation des automatismes. Seules les personnes dotées de ces compétences doivent être autorisées à programmer, installer, modifier et appliquer ce produit.

Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

Partie I
Comprendre Unity Application Generator

Introduction

UAG est un outil perfectionné servant à la conception d'un procédé en vue de son automatisation. Pour travailler efficacement, il est nécessaire de comprendre les principes de base de cet outil. Vous enseigner ces principes est l'objet de cette partie.

De plus, elle expliquera en détail tous les éléments dans UAG servant à la description du procédé.

Contenu de cette partie

Cette partie contient les chapitres suivants :

Chapitre	Titre du chapitre	Page
1	Configuration et procédure de mise à niveau	17
2	Introduction à Unity Application Generator	27
3	Sécurité des accès UAG	39
4	Modèle physique	59
5	Modèle topologique	117

Chapitre 1
Configuration et procédure de mise à niveau

Introduction

Unity Application Generator 3.3 SP4 remplace toutes les autres versions de Unity Application Generator.

Un certain nombre de nouvelles caractéristiques et fonctionnalités ont été mises en œuvre.

Ce chapitre présente :

- les configurations matérielle et logicielle minimales
- la mise à niveau de vos projets existants vers les nouvelles versions de Unity Application Generator

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Configuration matérielle minimale	18
Configuration logicielle minimale	19
Informations d'installation	22
Mise à niveau des projets existants vers UAG et Unity Pro	25

Configuration matérielle minimale

Configuration matérielle minimale

Configuration matérielle minimale

Configuration minimale :

Composant	Description
Système	Processeur Pentium 2 GHz, ou supérieur
Mémoire RAM	1024 Mo (conseillé : 2048 Mo)
Disque dur	500 Mo de mémoire pour l'installation
Système d'exploitation	Windows 2000 ou Windows XP Professionnel ou Windows Vista Business ou Windows 7 Professionnel (32/64 bits)
Lecteur	Lecteur de DVD-ROM

Configuration logicielle minimale

Système d'exploitation

Unity Application Generator fonctionne sous Microsoft Windows XP Professional, Windows Vista Business and Windows 7 Professional (32/64 bit)

Pour exécuter UAG sur un système Windows 7 Professionnel, vous devez posséder des droits d'administrateur. Si vous n'êtes pas l'administrateur de la machine, vous ne pourrez pas exécuter UAG.

Installation de base

Vous avez besoin d'au moins un outil de programmation d'API et d'un programme IHM pour travailler avec Unity Application Generator. Il vous faut donc installer les programmes suivants avant d'installer UAG :

- Un outil de programmation d'API (Unity Pro)
- Un programme d'IHM (Vijeo Citect, ArchestrA, Monitor Pro ou iFIX)
- D'autres programmes pour les communications Advantys ou les bus de terrain.

Logiciel de programmation

UAG génère une logique et une configuration de l'API pour les API de type Quantum/Premium/M340/M580 dans Unity Pro.

Par conséquent, les logiciels de programmation suivants doivent avoir été installés sur votre PC :

- Unity Pro 4.1/5.0/6.0/7.0/8.0/8.1

NOTE : Lors de l'installation, Unity Application Generator détecte les versions existantes de Unity Pro sur votre PC et installe les DLL et les blocs fonctions adéquats. Le paramétrage d'un PC avec d'autres versions de Unity en parallèle **n'est pas** pris en charge.

Vijeo Citect

Unity Application Generator génère des tags et des graphiques pour le système SCADA Vijeo Citect.

Ainsi les logiciels suivants doivent avoir été installés sur votre PC :

- Vijeo Citect 6.10/7.0/7.10/7.20/7.30/7.40/7.40 SP1

ArchestrA

Unity Application Generator génère des tags pour le système SCADA ArchestrA.

Par conséquent, les logiciels suivants doivent avoir été installés sur votre PC :

- ArchestrA

Monitor Pro

Unity Application Generator génère des tags et des graphiques pour le système SCADA Monitor Pro.

Ainsi les logiciels suivants doivent avoir été installés sur votre PC :

- Monitor Pro 7.6 ou FactoryLink 8

Non pris en charge en ce qui concerne les nouvelles fonctionnalités de la version 3.2 ou ultérieure de UAG.

iFIX

Unity Application Generator génère des tags et des graphiques pour le système SCADA Intellution iFIX.

Par conséquent, les logiciels suivants doivent avoir été installés sur votre PC :

- Intellution iFIX 3.5 ou 4.0
- Pilote MBT V7.156l pour iFIX de chez Dimension Software Inc.

Non pris en charge en ce qui concerne les nouvelles fonctionnalités de la version 3.2 ou ultérieure de UAG.

Logiciel de configuration Advantys

Unity Application Generator peut gérer les modules Advantys STB. Si vous souhaitez utiliser la fonction d'importation Advantys STB de UAG, vous devez installer le logiciel de configuration Advantys ou vous devez fournir des fichiers d'export Advantys.

Par conséquent, les logiciels suivants doivent avoir été installés sur votre PC :

- Logiciel de configuration Advantys 4.7/5.0/5.5/7.0/8.0

Hilscher Sycon

Unity Application Generator peut générer des données pour les communications Profibus. Vous devez pour cela paramétrer la configuration Profibus à l'aide d'un outil tiers comme Hilscher Sycon.

Par conséquent, les logiciels suivants doivent avoir été installés sur votre PC :

- Hilscher Sycon 2.10

Prosoft Configuration Builder

Unity Application Generator peut générer des données pour les communications Profibus. Vous devez pour cela paramétrer la configuration Profibus à l'aide d'un outil tiers comme Prosoft Configuration Builder.

Par conséquent, les logiciels suivants doivent avoir été installés sur votre PC :

- Prosoft Configuration Builder 4.0.0.0/4.1.0.0/4.1.1.1/4.1.2.3 (fourni sur le DVD d'installation d'UAG)

Acrobat Reader

Vous avez reçu avec Unity Application Generator le manuel utilisateur ainsi que la documentation sur les équipements de commande intelligents.

Par conséquent, les logiciels suivants doivent avoir été installés sur votre PC :

- Adobe Acrobat Reader

MS Word

Unity Application Generator permet de générer des rapports pour la documentation du projet. Celle-ci s'effectue sous Microsoft Word.

Par conséquent, les logiciels suivants doivent avoir été installés sur votre PC :

- Microsoft Word 97 ou Word 2000/2003/2007

Informations d'installation

Présentation de l'installation

Si vous êtes nouvel utilisateur de Unity Application Generator, il vous faut installer les logiciels suivants :

1. Logiciel de programmation d'API :
   - Schneider Electric Unity Pro 4.1/5.0/6.0/7.0/8.0/8.1
2. Logiciel de programmation d'IHM :
   - Schneider Electric Vijeo Citect 6.10/7.0/7.10/7.20/7.30/7.40/7.40 SP1 ou
   - Wonderware ArchestrA ou
   - Schneider Electric Monitor Pro 7.6 (non pris en charge en ce qui concerne les nouvelles fonctionnalités de la version 3.2 ou ultérieure de UAG) ou
   - Intellution iFIX 3.5 ou 4.0 comprenant le pilote MBT V7.156I pour iFIX de chez Dimension Software Inc. (non pris en charge en ce qui concerne les nouvelles fonctionnalités de la version 3.2 ou ultérieure de UAG)
3. Logiciel de configuration Advantys 4.7/5.0/5.5/7.0/8.0 de Schneider
4. Unity Application Generator 3.3 SP4 de Schneider Electric

NOTE :

- Le logiciel de configuration Advantys, les logiciels Hilscher Sycon et Prosoft Configuration Builder ne sont pas nécessaires à l'installation du logiciel UAG. Si vous devez utiliser les fonctions Advantys ou Profibus de UAG, vous devez installer l'outil prévu à cet effet.
- Si vous voulez générer de la documentation sur vos projets Unity Application Generator, Microsoft Word 97 ou Word 2000/2003/2007 doit être installé sur votre PC.

Installation de Unity Pro

Procédez comme suit pour installer Unity Pro :

Etape	Action
1	A partir du CD de Unity Pro, lancez *setup.exe*
2	Suivez les instructions du programme d'installation.

Installation de Vijeo Citect

Procédez comme suit pour installer Vijeo Citect :

Etape	Action
1	A partir du CD de Vijeo Citect, lancez *setup.exe*.
2	Suivez les instructions du programme d'installation.

Installation de ArchestrA

Procédez comme suit pour installer ArchestrA :

Etape	Action
1	A partir du CD de ArchestrA, lancez *setup.exe*
2	Suivez les instructions du programme d'installation.

Installation de Monitor Pro

Procédez comme suit pour installer Monitor Pro :

Etape	Action
1	A partir du CD de Monitor Pro, lancez *setup.exe*
2	Suivez les instructions du programme d'installation. **Remarque** : n'installez aucun pilote.

Installation de iFIX

Procédez comme suit pour installer iFIX :

Etape	Action
1	A partir du CD iFIX, lancez *setup.exe*
2	Suivez les instructions du programme d'installation. **Remarque** : n'installez aucun pilote.

Installation du pilote MBT

Procédez comme suit pour installer le pilote MBT V7.156l Modbus Plus/Ethernet :

Etape	Action
1	A partir du CD de MBT, lancez *setup.exe*
2	Suivez les instructions du programme d'installation. **Remarque** : utilisez les valeurs par défaut proposées par le programme d'installation.
3	Configurez le pilote MBT dans la configuration système iFIX. **Remarque** : ne définissez pas l'option **Création auto** dans **Power Tool** sur "non" (**Options** →**Configuration** →**Avancé**).

Installation de Unity Application Generator

Procédez comme suit pour installer Unity Application Generator :

Etape	Action
1	A partir du DVD de Unity Application Generator, lancez *setup.exe*
2	Suivez les instructions du programme d'installation.

Définition de répertoires pour les fichiers de données d'application

UAG utilise plusieurs fichiers pour lire ou écrire différents types de données d'application. Des sous-répertoires du répertoire d'installation de UAG sont censés contenir certains types de fichiers avec des noms figés. Ces sous-répertoires sont créés lors de l'installation.

La liste suivante répertorie les répertoires prédéfinis contenant les fichiers de données d'application de UAG.

- Bibliothèques SCoD (**.osl*) dans le sous-répertoire *\DB*
- Feuilles de style XML (**.xsl*) pour la génération d'IHM génériques dans le sous-répertoire *\DB*
- Fichiers d'aide de SCoD (définis par l'utilisateur, par exemple **.pdf*) dans le sous-répertoire *\Doc*
- Bibliothèques de modèles (**.ost*) dans le sous-répertoire *\DB*
- Fichiers d'aide de modèles (définis par l'utilisateur, par exemple **.pdf*) dans le sous-répertoire *\Doc*

UAG générera un répertoire pour les fichiers d'archive. Le nom du répertoire est *.\archive*.

Mise à niveau des projets existants vers UAG et Unity Pro

Mise à niveau automatique

La version actuelle de Unity Application Generator est à compatibilité descendante avec la version précédente de Unity Application Generator.

Les projets, personnalisations et bibliothèques SCoD existants sous Unity Application Generator seront automatiquement mis à jour.

Migration vers Unity Pro

Consultez le manuel Unity Pro pour migrer les projets d'anciennes versions vers une nouvelle version de Unity Pro.

Chapitre 2
Introduction à Unity Application Generator

Introduction

Ce chapitre présente les principes de UAG ainsi que les avantages qu'en tire l'utilisateur. Il explique le concept de Unity Application Generator ainsi que la procédure permettant de développer une application à l'aide de ce logiciel.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Informations générales concernant la sécurité	28
Qu'est-ce que Unity Application Generator ?	30
Les fondations de Unity Application Generator	33
Création d'une application	37

Informations générales concernant la sécurité

⚠ AVERTISSEMENT

FONCTIONNEMENT IMPREVU DE L'APPAREIL

Testez et vérifiez le fonctionnement des applications avant leur diffusion.

Procédez à une analyse de sécurité et à des tests adaptés à l'application. Cet ensemble logiciel est un moyen destiné à concevoir rapidement une application. La programmation système et les forçages sont obligatoires pour assurer la sécurité des applications réalisées.

Toutes les exigences réglementaires applicables doivent être respectées avant de diffuser une application destinée à une utilisation générale.

Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

⚠ AVERTISSEMENT

PERTE DE CONTROLE

Le concepteur d'un système de commande doit envisager les modes de défaillance détectés possibles des chemins de commande et, pour certaines fonctions de commande critiques, prévoir un moyen d'atteindre un état sécurisé lors de la détection d'une défaillance d'un chemin, et après. L'arrêt d'urgence et l'arrêt en cas de sur-course constituent des exemples de fonctions de commande essentielles.

Des chemins de commande distincts ou redondants doivent être prévus pour les fonctions de commande critiques.

Les chemins de commande système peuvent inclure les liaisons de communication. Il est nécessaire de tenir compte des conséquences des retards de transmission inattendus ou des défaillances d'une liaison détectées.

Chaque application doit être testée individuellement et de manière approfondie afin de vérifier son fonctionnement correct avant sa mise en service.

Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

⚠ ATTENTION

APPLICATION IMPREVISIBLE

N’utilisez pas d’outil comme Unity Pro pour modifier des segments de l’application générée par UAG. Pour assurer une maintenance correcte des liaisons de communication et des propriétés, toute modification apportée à ces segments doit être effectuée dans l’environnement UAG.

Le non-respect de ces instructions peut provoquer des blessures ou des dommages matériels.

Qu’est-ce que Unity Application Generator ?

Qu’est-ce que Unity Application Generator ?

Unity Application Generator est un nouvel outil de conception de procédé pour l’automatisation de procédé. Il comble le vide existant normalement entre l’ingénieur procédé et l’ingénieur automaticien. Auparavant, les deux groupes avaient travaillé avec des outils spécialisés non compatibles. Les modifications apportées à la conception du procédé défini par l’ingénieur procédé devaient être répétées par l’automaticien. Ce n’est plus le cas avec UAG.

Pour l’ingénieur procédé, UAG permet à l’utilisateur de :

- définir un schéma général du procédé sur la base d’objets définis dans le modèle physique de la norme ISA S88.01 tels que zone, cellule procédé, unité, équipement et module de commande ;
- de faire le lien entre des objets de UAG et ses outils de base tels que les plans d’électricité, Autocad, les PID, etc.

Pour l’ingénieur automaticien, UAG permet à l’utilisateur de :

- définir l’architecture de l’automatisme à l’aide d’API, d’IHM et de réseaux en tant que modèle topologique;
- d’affecter la logique de commande aux objets que l’ingénieur procédé aura définis ;
- de générer une grande partie de la logique de commande de l’API et de l’IHM à partir de la conception du procédé.

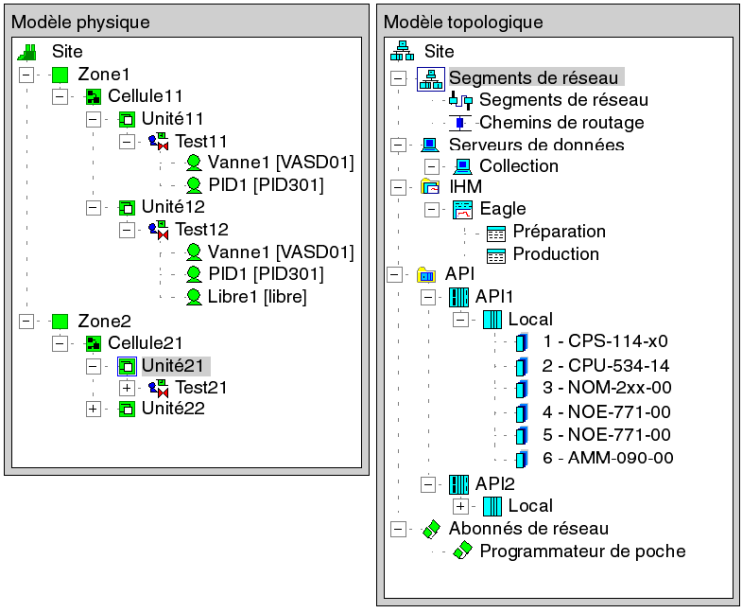
Avec cette approche, UAG permet à l’utilisateur de concevoir un automatisme distribué à plusieurs API et IHM en une seule phase.

Comment se présente-t-il ?

La fenêtre cadre de Unity Application Generator affiche le modèle physique et/ou le modèle topologique comme une arborescence d’objets, de façon similaire à l’Explorateur de Windows.

- Le modèle physique est généralement conçu par l’ingénieur procédé. Les éléments du modèle physique sont affectés aux ressources du modèle topologique.
- Le modèle topologique est généralement conçu par l’automaticien.

L'illustration suivante présente les fenêtres cadres affichant le modèle physique et le modèle topologique d'un exemple de procédé :



En outre, l'arborescence des bibliothèques peut s'afficher. Elle répertorie les bibliothèques SCoD et les bibliothèques de modèles d'un projet.

Pour des informations détaillées, veuillez consulter le chapitre *Bibliothèques de modèles* du Manuel utilisateur avancé UAG.

Que trouve-t-on en sortie de UAG ?

Unity Application Generator génère un code pour :

- API
  Programme utilisateur pour Unity Pro, logiciel de programmation CEI 61131 de Schneider Electric.
- IHM
  Tags IHM et graphiques relatifs aux objets créés avec UAG pour les différentes IHM.
  - Vijeo Citect
  - Monitor Pro
  - iFIX
  - ArchestrA
  - IHM générique

Tags IHM contenant toutes les informations significatives enregistrées dans un fichier d'échange de données à utiliser par toute IHM possédant une fonctionnalité d'importation. Reportez-vous également au Manuel utilisateur avancé de UAG, Utilisation de Unity Application Generator avec une IHM générique *(voir Unity Application Generator (UAG) 3.3 SP4, a SoCollaborative software, Manuel utilisateur avancé)*.

La production de UAG est chargée directement sur l'API ou l'IHM. Elle constitue la conception statique du procédé (environ 30 % de la conception totale).

Afin de constituer l'application finale, il faut compléter la sortie de UAG en ajoutant le comportement dynamique du procédé à l'API et au système IHM.

Autres caractéristiques et avantages

Unity Application Generator présente les autres caractéristiques suivantes :

- Une seule base de données pour les paramètres API et IHM. La communication est définie de manière cohérente entre l'API et l'IHM.
- Une conception orientée objet avec des objets technologiques réutilisables tels que les moteurs, les vannes, les pompes, etc.
- Un système de suivi des modifications permettant à l'utilisateur de garder une trace de tous les changements effectués par plusieurs utilisateurs pouvant travailler en parallèle sur le projet.
- Utilisation des normes de l'industrie et des quasi-normes telles qu'ActiveX.
- La documentation, telle que les plans CAO, les documents texte, etc. peut être liée aux objets respectifs.

L'utilisateur tire de Unity Application Generator les avantages suivants :

- Une réduction du temps passé en application et en test du système du fait des objets réutilisables
- Une plus grande rapidité de conception réduisant d'autant les coûts du cycle de vie
- Une plus grande rapidité de maintenance du système
- Une revalidation plus rapide de la conception du procédé à l'aide du système de suivi des modifications. Chaque modification du procédé est consignée par l'`historique des changements`.
- Un effort de validation rendu plus facile pour les projets devant être validés du fait des réglementations nationales
- Une qualité de logiciel applicatif sérieusement augmentée
- La réduction du temps nécessaire à l'installation et à la mise en service

Les fondations de Unity Application Generator

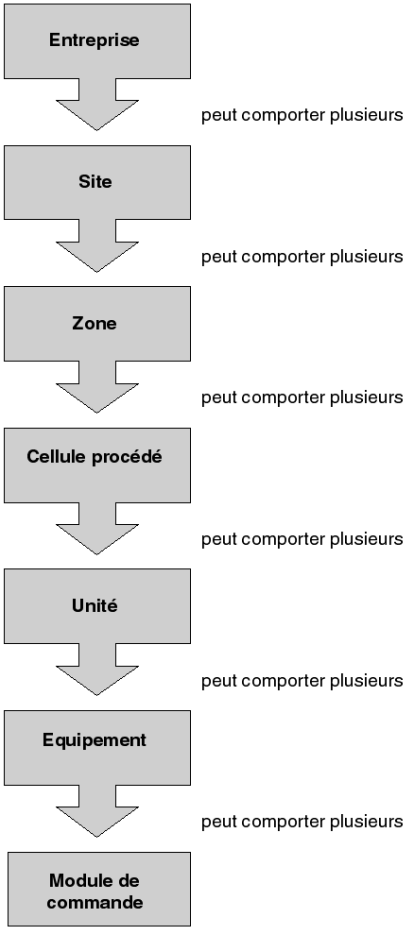
Conformité à la norme ISA S88.01

Unity Application Generator emploie la terminologie de la norme ISA S88.01 sur les traitements par lot, 1ère partie : "Modèles et terminologie". En adoptant la structure S88, l'utilisateur de Unity Application Generator est en mesure de décomposer les tâches de son procédé suivant la norme, et d'effectuer une mise en œuvre par le bas assistée par Unity Application Generator.

La norme S88 décrit une hiérarchie de structuration de l'ensemble des installations d'automatisation dans une entreprise en tant que modèle physique. Dans cette version de Unity Application Generator, la décomposition d'une entreprise commence au niveau du site.

Le schéma montre l'ensemble de la hiérarchie que le concepteur peut définir pour décrire l'automatisation du site. Les lignes symbolisent une relation un à plusieurs, par exemple une unité consiste en un ou plusieurs équipements. Les zones, cellules procédé et unités servent à structurer l'usine en parties qui effectuent différentes tâches.

Modèle physique selon l'ISA S88.01



Base de données de procédé unique

Développer une application d'automatisation à l'aide de Unity Application Generator, c'est créer en une seule étape le programme nécessaire pour

- l'API ;
- les communications ;
- l'IHM.

Cela permet à l'utilisateur de n'avoir qu'une seule base de données de procédé, laquelle contiendra toutes les données de procédé pour l'IHM mais aussi l'API. Ainsi ces deux applications, IHM et API, auparavant distinctes, fonctionnent maintenant sur le même modèle de données et établissent une unique solution d'automatisation de procédé.

Nouvel objet : modules de commande intelligents (SCoD)

Un module de commande est un terme générique pour définir des capteurs, actionneurs et accessoires de régulation qui, d'un point de vue de l'automatisme, fonctionnent comme une même entité. Un module de commande peut être un objet réel pouvant être mesuré et surveillé, par exemple un moteur, une vanne, un transmetteur de température etc., ou il peut être un objet logiciel logique, utilisé en régulation ou autres fonctions de commande (boucle PID, temporisateur, compteur, etc.).

Un module de commande intelligent (SCoD) est un module prédéfini d'un type spécial au sein de Unity Application Generator, et que l'on appelle type de module de commande.

Un SCoD peut être instancié autant de fois que le nécessite le procédé. Il a été entièrement testé. Cela réduit de manière significative le coût des tests de toute l'application.

Un SCoD peut contenir :

- la logique d'API nécessaire à la commande d'un module réel d'un type défini ;
- des symboles IHM animés ;
- la commande manuelle du module par l'IHM ;
- l'accès à toutes les variables nécessaires à l'API et l'IHM ;
- une connexion vers la gestion d'alarme IHM.

Fonctionnalités pour 21 CFR Partie 11

Afin de respecter les exigences de la 21 CFR Partie 11, UAG délivre les informations suivantes :

- Affichage de l'horodatage UAG dans l'`historique des changements` et dans la boîte de dialogue des propriétés de l'API
- Affichage de l'horodatage de personnalisation dans la boîte de dialogue des propriétés du projet
- Affichage de l'horodatage UAG dans le fichier journal de Unity Pro
- Affichage du fuseau horaire UTC si les heures sont enregistrées en journal

Adaptable aux normes du client

UAG peut être adapté aux besoins spécifiques d'un projet en fonction des machines, des produits, de la configuration d'automatisation, des outils de conception, etc.

L'administrateur système peut définir individuellement les conventions de nommage ou d'étiquetage pour tous les objets nommés utilisant les champs configurables :

- Eléments physiques (ex. : zone, cellule procédé, unité, équipement)
- API
- IHM
- ...

Il est de plus possible de définir des configurations valides d'automatisation, par exemple :

- les gammes d'API (Quantum, Premium, M340, M580, etc.)
- les composants matériels d'API (châssis, modules, processeurs, etc.)

Création d'une application

Les différentes personnes impliquées dans la création d'une application

Pour que la méthodologie appliquée à l'outil soit transparente, cette documentation expose les différents rôles utilisés ici.

La création d'une application à l'aide de Unity Application Generator nécessite différents types d'utilisateurs ou groupes d'utilisateurs. Chaque groupe est responsable d'un certain aspect du développement de l'application.

Les principaux rôles sont les suivants :

- l'administrateur
- l'ingénieur procédé
- l'ingénieur automaticien

NOTE :

- Dans Unity Application Generator, les différents rôles ne peuvent pas être gérés comme des comptes.
- Il est possible de définir ces rôles de manière différente dans chaque projet selon la structure de celui-ci.

Personnalisation : tâche de l'administrateur

L'administrateur configure UAG en fonction des normes de la société. Il est possible de définir de nombreuses options différentes avant que le projet ne démarre réellement (il est cependant possible de commencer un projet avec la configuration par défaut), par exemple :

- Conventions de nommage
- Configurations standard API
- Unités physiques particulières
- ...

Conception du procédé : tâche de l'ingénieur procédé

L'ingénieur procédé est la personne définissant la décomposition générale du site en unités logiques en fonction du procédé qu'il décrit.

Les éléments par défaut sont les suivants :

- Site
- Éléments physiques (ex. : zone, Cellule procédé, unité, Équipement)
- Module de commande

Ces éléments sont variables. Il est possible de définir jusqu'à 8 éléments.

L'ingénieur procédé définit chaque élément, ses constituants et forçages. Il peut également relier les informations aux différents éléments préexistants, par exemple des plans DAO. En revanche, ce n'est pas à lui qu'incombe l'élaboration de l'architecture de la commande.

Architecture du système : tâche de l'ingénieur automaticien

Les tâches de l'automaticien sont les suivantes :

- la définition du modèle topologique de l'automatisme ;
- l'attribution de toutes les entrées et sorties ;
- l'affectation des données émanant du modèle topologique au modèle physique ;
- la configuration de communication, par exemple API - API.

En effectuant ces tâches, l'ingénieur automaticien définit en une seule étape les différents IHM, API ainsi que le réseau du système. UAG générera une grande partie du code pour les API et IHM. Le reste de la logique doit ensuite être ajouté manuellement à l'aide des outils de programmation API et IHM respectifs.

Exemples :

- Comportement dynamique en diagramme fonctionnel de séquence (SFC) dans la logique de l'API
- Autres graphiques de tuyauterie, réservoirs, etc.
- Navigation entre écrans opérateur dans l'application IHM,
- etc.

Avantages de Unity Application Generator

En réunissant les tâches de l'ingénieur procédé et de l'automaticien en un seul logiciel, la cohérence entre des tâches auparavant distinctes est établie.

De plus, Unity Application Generator intègre également les tâches suivantes :

- Définition du réseau
- Configuration et programmation des API
- Configuration et programmation des IHM
- Définition des communications entre les API et les IHM

Chapitre 3
Sécurité des accès UAG

Introduction

Ce chapitre vous présente les outils utilisés pour gérer la sécurité des accès pour les projets Unity Application Generator.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Gestion de la sécurité d'accès à UAG	40
Éditeur de sécurité	42
Informations utilisateur	43
Fonctions utilisateur	45
Profils utilisateur	49
Règles	54
Procédure de reprise après détection d'une erreur	56
Droits d'accès	57

Gestion de la sécurité d'accès à UAG

Introduction

L'Editeur de sécurité vous permet de limiter et de contrôler l'accès aux diverses fonctionnalités du logiciel.

NOTE : Le contrôle d'accès au logiciel est facultatif.

La sécurité d'accès à Unity Application Generator concerne le terminal sur lequel le logiciel est installé et pas le projet, qui dispose de son propre système de protection.

Il est possible de tenir un fichier journal pour consigner, par ordre chronologique, les diverses opérations effectuées par les utilisateurs ayant accédé au logiciel.

Super-utilisateur

Le super-utilisateur est la seule personne ayant le droit de gérer la sécurité d'accès à l'application. Le super-utilisateur définit la liste des noms des utilisateurs qui sont autorisés à accéder à l'application et de leurs droits d'accès.

Après l'installation de l'Editeur de sécurité, seul le super-utilisateur peut accéder à la configuration de la sécurité sans limitation de droits (sans mot de passe).

NOTE : Le nom d'utilisateur réservé au super-utilisateur est `Supervisor`.

Le super-utilisateur dispose des droits suivants. Il peut :

- créer ou modifier la liste des utilisateurs,
- créer ou modifier les profils utilisateur,
- importer ou exporter des utilisateurs,
- modifier les règles de la sécurité d'accès au logiciel,
- modifier son mot de passe,
- réinitialiser les mots de passe des utilisateurs.

Utilisateurs

Les utilisateurs du logiciel sont définis dans la liste des utilisateurs par le super-utilisateur, si la sécurité d'accès à UAG est active. Si votre nom figure dans la liste des utilisateurs, vous pouvez accéder à une instance du logiciel en entrant votre nom (exactement tel qu'il apparaît sur la liste) et votre mot de passe.

L'utilisateur dispose des droits suivants. Il peut :

- accéder en mode lecture aux droits définis par son profil utilisateur,
- modifier son mot de passe.

Profils utilisateur

Le profil d'un utilisateur comporte tous ses droits d'accès. Il est défini par un nom (2 à 16 caractères), un commentaire facultatif (maximum de 256 caractères) et une liste des droits d'accès. UAG fournit 2 profils utilisateur préconfigurés qui ne peuvent pas être modifiés. Pour renseigner cette liste, le super-utilisateur peut créer tous les profils personnalisés dont il a besoin.

Profils utilisateur préconfigurés

UAG fournit les 2 profils utilisateur préconfigurés suivants :

Profil	Description
Lecture seule	L'utilisateur peut accéder au projet en mode lecture seulement.
Lecture/Écriture	L'utilisateur dispose d'un accès illimité au projet.

Profil utilisateur par défaut

Lorsque la sécurité d'accès au logiciel est active, le super-utilisateur peut octroyer à un utilisateur donné le droit d'ouvrir une instance du logiciel, sans avoir à entrer son nom et son mot de passe. Dans ce cas, l'utilisateur aura le profil par défaut.

Informations de consignation générées par l'Editeur de sécurité

Les informations de consignation de l'Editeur de sécurité sont gérées par l'**Afficheur d'événements** de Microsoft Windows. Si vous souhaitez afficher les informations consignées, vous devez lancer l'afficheur d'événements. Vous pouvez filtrer les informations selon tous les événements **séquentiels de serveur**.

Vous pouvez utiliser toutes les fonctions de l'afficheur d'événements, telles que **trier**, **filtrer** etc.

Confirmer

Lorsque cette option est sélectionnée par le super-utilisateur, vous devez entrer vos nom et mot de passe à chaque fois que l'action utilisateur envisagée fait l'objet d'un accès .

Multiple instanciation

Lorsque UAG exécute simultanément de multiples instances sur le même terminal, chacune de ces instances accordera l'accès aux fonctions qui ont été définies par les droits de l'utilisateur qui a créé l'instance en question. Cependant, la méthode d'accès à UAG utilisée (sécurité inactive ou type de sécurité active) est la même pour toutes les instances.

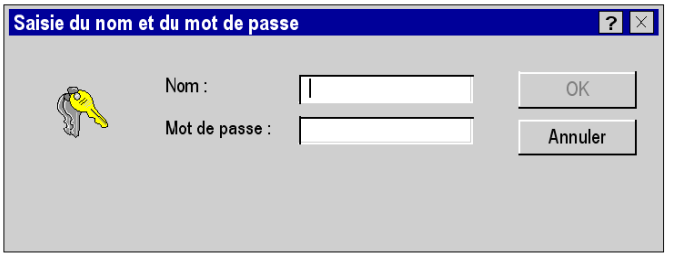
Éditeur de sécurité

Introduction

L'Éditeur de sécurité permet de définir les utilisateurs du logiciel et leurs droits d'accès. Il vous permet aussi de définir les actions qui seront protégées par la saisie d'un nom d'utilisateur et d'un mot de passe et qui seront consignées dans un fichier journal.

Lancement de l'Éditeur de sécurité

Exécutez les commandes suivantes pour démarrer l'Éditeur de sécurité :

Etape	Action
1	Ouvrez avec **Démarrer** →**Programmes** →**Schneider Electric** l'**Éditeur de sécurité**. La boîte de dialogue suivante apparaît. 
2	Entrez vos nom et mot de passe. **Remarque :lors de votre premier accès,** entrez `Supervisor` sans mot de passe.
3	Validez par **OK**.
4	Si vous êtes un utilisateur, vous pouvez accéder à l'écran d'informations utilisateur qui vous permet d'éditer votre profil utilisateur ou de modifier votre mot de passe.
5	Si vous êtes le super-utilisateur, l'Éditeur de sécurité démarre et active la gestion de la sécurité d'accès à UAG.

Informations utilisateur

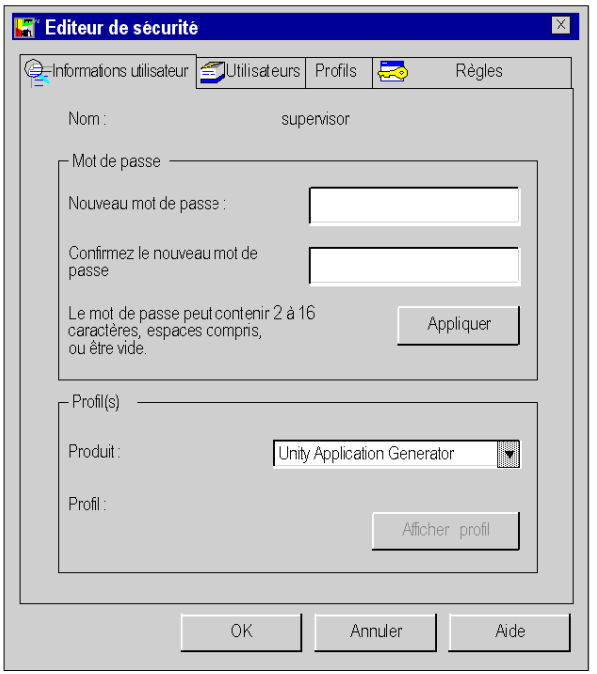
Introduction

Cet onglet de l'Editeur de sécurité est accessible à tous les utilisateurs. Il permet d'exécuter les fonctions suivantes :

- modifier son propre mot de passe. Les utilisateurs, y compris le super-utilisateur, peuvent uniquement modifier leur propre mot de passe,
- éditer son propre profil et les droits d'utilisateur associés.

Onglet Informations utilisateur

L'onglet **Informations utilisateur** a l'aspect suivant. Il indique le nom de l'utilisateur et contient les 2 sections **Mot de passe** et **Profil** :



Description des paramètres

Le tableau suivant décrit les paramètres **Mot de passe** :

Paramètre	Description
Nouveau mot de passe	Champ de saisie du nouveau mot de passe : de 2 à 16 caractères. Ce champ peut être vide (aucun caractère saisi).
Confirmez le nouveau mot de passe	Champ de confirmation du nouveau mot de passe. Le contenu de ce champ doit être identique à celui du champ **Nouveau mot de passe**.
Appliquer	Commande utilisée pour appliquer le nouveau mot de passe. **Remarque** : Le nouveau mot de passe ne sera définitivement enregistré qu'une fois validé à l'aide du bouton **OK**. Cliquez sur le bouton**Annuler** pour annuler l'enregistrement du nouveau mot de passe.

Le tableau suivant décrit les paramètres **Profil(s)** :

Paramètre	Description
Produit	Active la sélection du produit pour lequel vous souhaitez afficher votre profil utilisateur : Dans notre cas, il s'agit de Unity Application Generator.
Profil	Indique le nom de votre profil utilisateur pour le produit en question.
Afficher profil	Utilisez cette commande pour afficher tous les droits d'accès inclus dans votre profil utilisateur.

Fonctions utilisateur

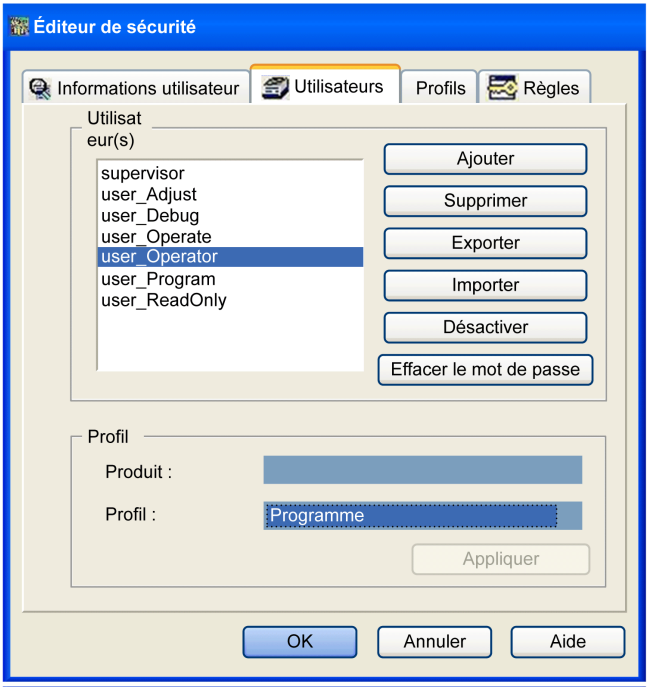
Introduction

Seul le super-utilisateur peut avoir accès à cet onglet. Il permet d'exécuter les fonctions suivantes :

- modifier la liste des utilisateurs,
- attribuer un profil à chaque utilisateur,
- importer/exporter des informations pour un ou plusieurs utilisateurs.

Onglet Utilisateurs

L'onglet **Utilisateurs** se présente comme suit. Il contient les 2 sections **Utilisateurs** et **Profil**. L'écran suivant montre l'onglet **Utilisateurs**.



Description des paramètres

Le tableau suivant décrit les paramètres **Utilisateurs** :

Paramètre	Description
Zone des utilisateurs	Liste des utilisateurs ayant accès à Unity Application Generator. Le nom du super-utilisateur est **supervisor**. Il est également possible de sélectionner le champ Utilisateurs préconfigurés : ● **user_Adjust**, ● **user_Debug**, ● **user_Operate**, ● **user_Operator**, ● **user_Program**, ● **user_ReadOnly**.
Ajouter	Commande utilisée pour ajouter un nouvel utilisateur à la liste.
Supprimer	Commande utilisée pour supprimer de la liste le ou les utilisateurs sélectionnés.
Exporter	Commande utilisée pour exporter vers un fichier les informations (nom et profils) du ou des utilisateurs sélectionnés. Les mots de passe ne sont pas exportés.
Importer	Commande utilisée pour importer les informations (nom et profils) du ou des utilisateurs sélectionnés depuis un fichier. Si l'utilisateur figure déjà sur la liste, un message d'information apparaît.
Désactiver	Commande utilisée pour désactiver le profil sélectionné.
Effacer le mot de passe	Commande utilisée pour effacer le mot de passe de l'utilisateur sélectionné.

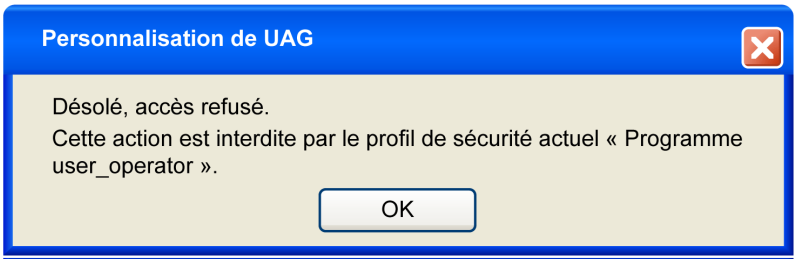
Le tableau suivant décrit les paramètres **Profils** :

Paramètre	Description
Produit	Utilisé pour choisir le produit pour lequel vous souhaitez définir le profil utilisateur.
Profil	Utilisé pour choisir le profil qui sera attribué à l'utilisateur sélectionné.
Appliquer	Commande utilisée pour appliquer le profil attribué. Le profil ne sera définitivement enregistré qu'une fois validé à l'aide du bouton **OK**. Cliquez sur le bouton **Annuler** pour annuler l'enregistrement du profil.

Un utilisateur peut créer un profil **user_Operator** dans l'**Éditeur de sécurité** de UnityPro pour UAG. Le **user_Operator** a un accès restreint défini comme suit :

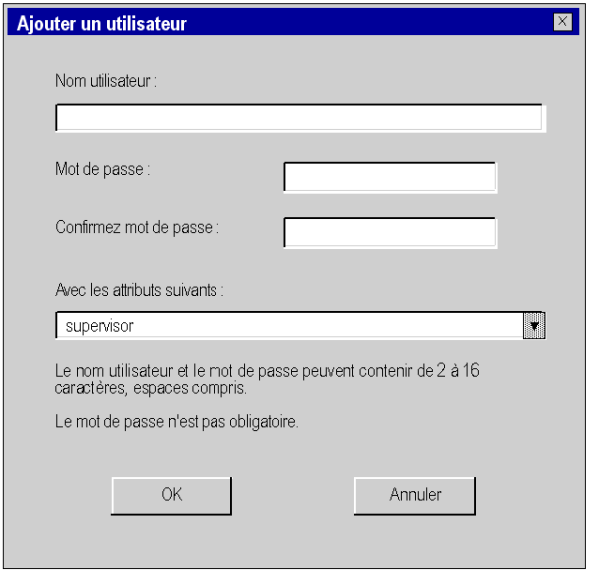
Accès autorisé	Accès refusé
Génération incrémentale →Cartographie mémoire	**Fichier →Fusionner projets**
Génération incrémentale →API	**Fichier →Réparer base de données du projet**
Génération incrémentale →IHM	**Fichier →Importer**
–	**Génération complète →Cartographie mémoire**
–	**Génération complète →API**
–	**Génération complète →IHM**

Si l'utilisateur se connecte en tant **user_Operator**, il ne peut pas modifier le **Générateur de serveur de données et d'IHM** dans la **personnalisation de UAG** et le message suivant s'affiche :



Ajouter un utilisateur

Un clic sur **Ajouter** permet d'afficher la boîte de dialogue suivante :



2 champs de saisie de données sont proposés :

Paramètre	Description
Nom d'utilisateur	Ce champ sert à entrer le nom du nouvel utilisateur (2 à 16 caractères). Si le nom entré n'est pas correct, ou s'il existe déjà, un message d'information s'affiche.
Mot de passe	Champ de saisie du mot de passe : de 2 à 16 caractères. Ce champ peut être vide (aucun caractère saisi).
Confirmer mot de passe	Champ de confirmation du mot de passe. Le contenu de ce champ doit être identique à celui du champ **Mot de passe**.
Avec les attributs suivants	Champ utilisé pour sélectionner l'utilisateur dont vous souhaitez récupérer les attributs.

Profils utilisateur

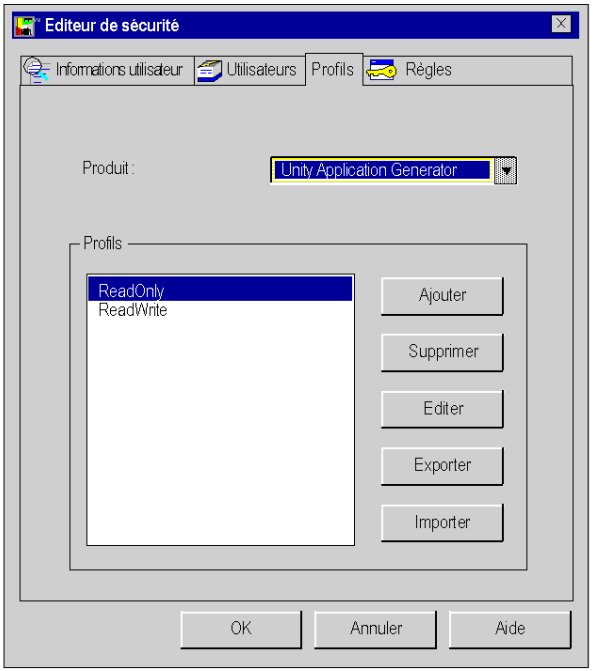
Introduction

Seul le super-utilisateur peut avoir accès à cet onglet. Il permet de réaliser les opérations suivantes :

- ajouter ou supprimer un profil de la liste,
- lire et modifier les droits d'accès associés à un profil,
- importer/exporter des informations pour un ou plusieurs profils utilisateur.

Onglet Profils

L'onglet **Profils** a l'aspect suivant. Il comporte les 2 zones **Produit** et **Profils** :



Description des paramètres

Le tableau suivant décrit les paramètres **Produit** :

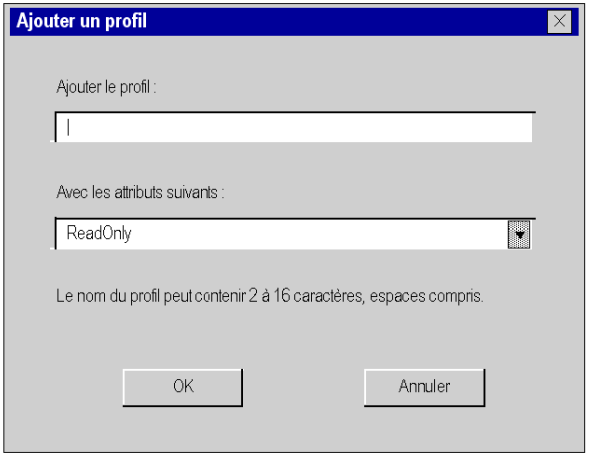
Paramètre	Description
Produit	Utilisé pour choisir le produit pour lequel vous souhaitez avoir accès au profil utilisateur.

Le tableau suivant décrit les paramètres **Profils** :

Paramètre	Description
Zone Profils	Liste des profils utilisateur associés à Unity Application Generator.
Ajouter	Commande utilisée pour ajouter un nouveau profil utilisateur à la liste.
Supprimer	Commande utilisée pour supprimer de la liste un ou plusieurs profils utilisateur sélectionnés. Si vous supprimez un profil de la liste, les utilisateurs ayant ce profil auront le profil par défaut. Les profils utilisateur préconfigurés ne peuvent pas être supprimés de la liste.
Editer	Commande utilisée pour modifier les droits d'accès associés au profil utilisateur sélectionné. Les profils utilisateur préconfigurés ne peuvent pas être modifiés à partir de la liste.
Exporter	Commande utilisée pour exporter vers un fichier les informations (nom et profils) sur le ou les profils utilisateur sélectionnés.
Importer	Commande utilisée pour importer depuis un fichier les informations (nom et profils) sur le ou les profils utilisateur sélectionnés. Si un profil d'utilisateur figure déjà sur la liste, un message d'information apparaît.

Ajouter un profil utilisateur

Utilisez la commande **Ajouter** pour afficher la boîte de dialogue suivante :

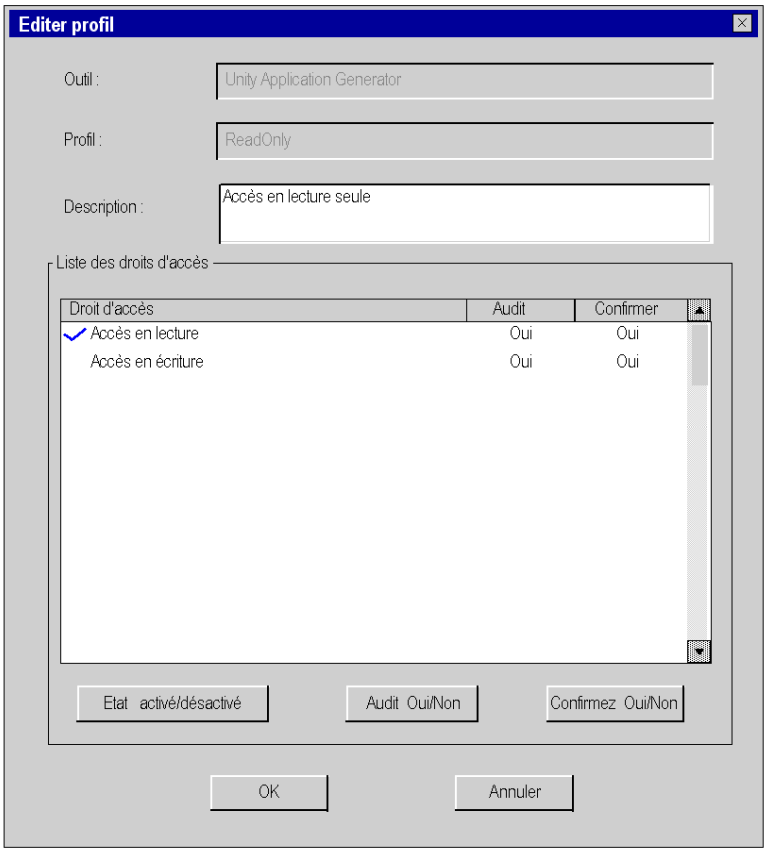


2 champs de saisie de données sont proposés :

Paramètre	Description
Ajouter le profil :	Ce champ sert à entrer le nom du nouveau profil utilisateur (2 à 16 caractères). Si le nom entré n'est pas correct, ou s'il existe déjà, un message d'information s'affiche.
Avec les attributs suivants :	Champ utilisé pour sélectionner le profil utilisateur dont vous souhaitez récupérer les attributs.

Modifier un profil utilisateur

Utiliser la commande **Éditer** pour afficher la boîte de dialogue suivante :



Cette boîte de dialogue comporte 4 zones pour l'affichage ou la saisie de données :

Paramètre	Description
Outil	Indique le nom du produit que vous avez sélectionné à l'écran précédent.
Profil	Indique le nom du profil utilisateur que vous avez sélectionné à l'écran précédent.
Description	Ce champ de saisie de données est utilisé pour associer un commentaire à la modification du profil utilisateur.

Paramètre	Description
Liste des droits d'accès	La liste affiche les droits d'accès associés au profil utilisateur sélectionné. ● **Droit d'accès** Liste des droits d'accès associés au profil utilisateur. ● **État activé/désactivé** Cette option permet d'activer/de désactiver un droit particulier du profil actuel : ● Pour activer le droit, sélectionnez le **Droit d'accès** dans la liste. Puis cliquez sur **État activé/désactivé** ; la coche de sélection apparaît. ● Pour désactiver le droit, sélectionnez le **Droit d'accès** dans la liste. Puis cliquez sur **État activé/désactivé** ; la coche de sélection disparaît. ● **Audit Oui/Non** Lorsque cette option est activée, elle permet de consigner les opérations dans le fichier journal. Pour activer/désactiver l'option pour le droit d'accès correspondant : ● sélectionnez le **Droit d'accès** dans la liste, ● puis cliquez sur **Audit Oui/Non**. ● **Confirmez Oui/Non** Pour les opérations qui doivent faire l'objet d'une confirmation, vous devez activer cette option. Pour activer/désactiver l'option pour le droit d'accès correspondant : ● sélectionnez le **Droit d'accès** dans la liste, ● puis cliquez sur **Confirmer Oui/Non**.

NOTE : Vous pouvez consulter les droits d'accès associés à un profil préconfiguré mais vous ne pouvez pas les modifier.

Règles

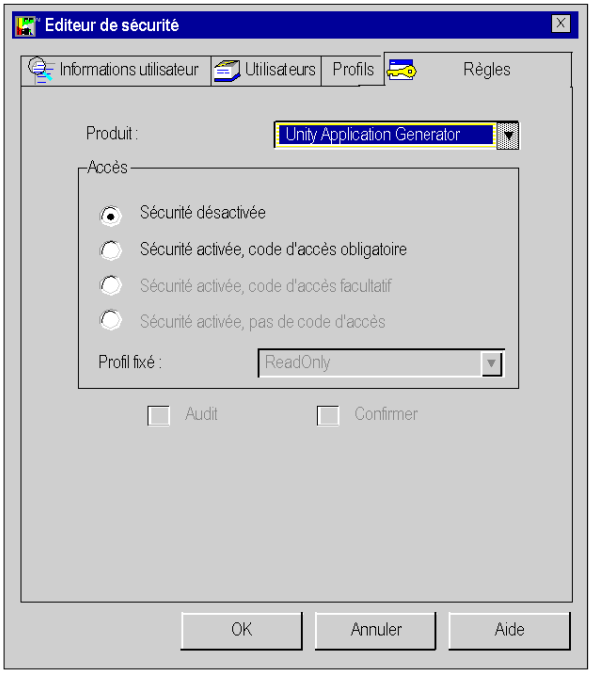
Introduction

Cet onglet de l'Editeur de sécurité est uniquement accessible au super-utilisateur. Il permet d'exécuter les fonctions suivantes :

- déterminer les directives associées à un produit,
- définir le profil par défaut,
- activer/désactiver l'option **Audit**.
- activer/désactiver l'option **Confirmer**.

Onglet Règles

L'onglet **Règles** de l'Editeur de sécurité a l'aspect suivant :



Description des paramètres

Le tableau suivant décrit les paramètres de la boîte de dialogue :

Paramètre	Description
Produit	Utilisé pour choisir le produit pour lequel vous souhaitez définir les règles de sécurité, dans notre cas, UAG.
Accès	Utilisé pour définir les règles générales d'accès à UAG : • **Sécurité désactivée** La sécurité est désactivée. Vous avez un accès direct à UAG. C'est le paramètre par défaut. • **Sécurité activée, code d'accès obligatoire** La sécurité est activée. Pour accéder à UAG, il est obligatoire d'entrer vos nom et mot de passe, qui détermineront votre profil. • **Sécurité activée, code d'accès facultatif** non disponible pour UAG • **Sécurité activée, pas de code d'accès** non disponible pour UAG Lorsque la sécurité est désactivée, les options **Audit** et **Confirmer** seront aussi désactivées.
Profil fixé	Utilisé pour définir le profil utilisateur par défaut.
Audit	Utilisé pour activer ou désactiver l'option **Audit** (cette option est uniquement disponible si la sécurité est activée). Si cette case à cocher est activée, un **fichier journal** sera créé pour mémoriser les événements des utilisateurs de UAG.
Confirmer	Utilisé pour activer ou désactiver l'option **Confirmer**. Si cette case est cochée, vous devez à chaque fois entrer vos nom et mot de passe. **Remarque :** Si vous utilisez l'option **Confirmer** de l'onglet **Règles** de l'Editeur de sécurité, choisissez avec précaution les droits d'accès lorsque vous créez un nouveau profil. Sinon, les modifications exigeront de multiples confirmations dans les sections de programmation.

NOTE : Les options **Audit** et **Confirmer** ne sont pas affectées à un utilisateur mais s'appliquent de manière globale à tous les produits UAG.

Procédure de reprise après détection d'une erreur

Introduction

L'Editeur de sécurité est utilisé pour contrôler l'accès à UAG. Si une difficulté survient (tentative d'accès illicite, mot de passe non valide ou autre), vous devez respecter la procédure de reprise correspondant à cette difficulté.

Accès par un utilisateur inconnu

Si un utilisateur n'est pas connu de l'Editeur de sécurité, et que la sécurité est activée, il existe 2 possibilités :

- L'accès est configuré avec un **code d'accès facultatif** (**Sécurité activée, mot de passe facultatif**).
  Dans ce cas, l'utilisateur inconnu peut accéder à UAG avec le profil par défaut.
- L'accès est configuré avec un **code d'accès obligatoire** (**Sécurité activée, mot de passe obligatoire**).
  Dans ce cas, l'utilisateur inconnu ne peut pas avoir accès à UAG.

Si la base de données de l'Editeur de sécurité est détériorée ou a été supprimée, l'accès à UAG est impossible, même si la sécurité n'est pas active (**Sécurité désactivée**).

Mot de passe non valide

Si vous avez oublié votre mot de passe, la procédure à suivre diffère selon que vous êtes un utilisateur ou le super-utilisateur :

- si vous êtes un utilisateur, prenez contact avec le super-utilisateur. Il peut réinitialiser votre mot de passe. Vous pouvez ensuite entrer un nouveau mot de passe,
- si vous êtes le super-utilisateur, réinstallez UAG en choisissant une installation personnalisée : Installez uniquement l'Editeur de sécurité.

Protection de la base de données

Pour aider à protéger la base de données de toute corruption possible, un fichier de sauvegarde est créé au moment de l'installation. Ce fichier caché est uniquement accessible en lecture seule. Il permet la restauration de la base de données en cas de détection d'une erreur.

NOTE : Le fichier de sauvegarde est actualisé et si la base de données est altérée, ce fichier est utilisé. Si une erreur est détectée au cours de la procédure automatique, vous devrez procéder à une nouvelle installation.

Droits d’accès

Introduction

Vous pouvez attribuer les droits d’accès suivants aux différents utilisateurs de UAG :

Description des droits d’accès

L’Éditeur de sécurité accorde 2 droits d’accès, Lecture seule et Lecture/Écriture, aux 3 éditeurs de UAG (éditeur de SCoD, éditeur de personnalisation, Unity Application Generator).

- Lecture seule
  Avec le droit en lecture seule, vous n’avez qu’un accès limité aux projets. Vous pouvez tout lire et voir. Mais vous ne pouvez pas utiliser les commandes suivantes :
  - **Fichier** →**Nouveau**
  - **Fichier** →**Ouvrir sans partage** (UAG seulement)
  - **Fichier** →**Fusionner projet** (UAG seulement)
  - **Fichier** →**Réparer base de données de projet** (UAG seulement)
  - **Fichier** →**Importer** (UAG seulement)
  - **Générer** →**Cartographie mémoire**
  - **Générer** →**API**
  - **Générer** →**IHM**
  - **Nouveau** (ex. pour les variables)
  - **Supprimer** (ex pour les variables)
  - **Coller** (ex pour les variables)
- Lecture/Écriture
  Pas de limitations.

Chapitre 4
Modèle physique

Introduction

Ce chapitre décrit la structure générale du modèle de base de l'ISA S88.01. Il fournit des informations détaillées sur les éléments et leurs propriétés.

L'ingénieur procédé doit connaître le modèle physique pour définir le modèle physique de son procédé.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
4.1	Présentation du modèle physique	60
4.2	Eléments physiques en général	64
4.3	Elément physique fonctionnel	66
4.4	L'élément Module de commande et les types de module de commande (modules de commande intelligents - SCoD)	73
4.5	Variables	86

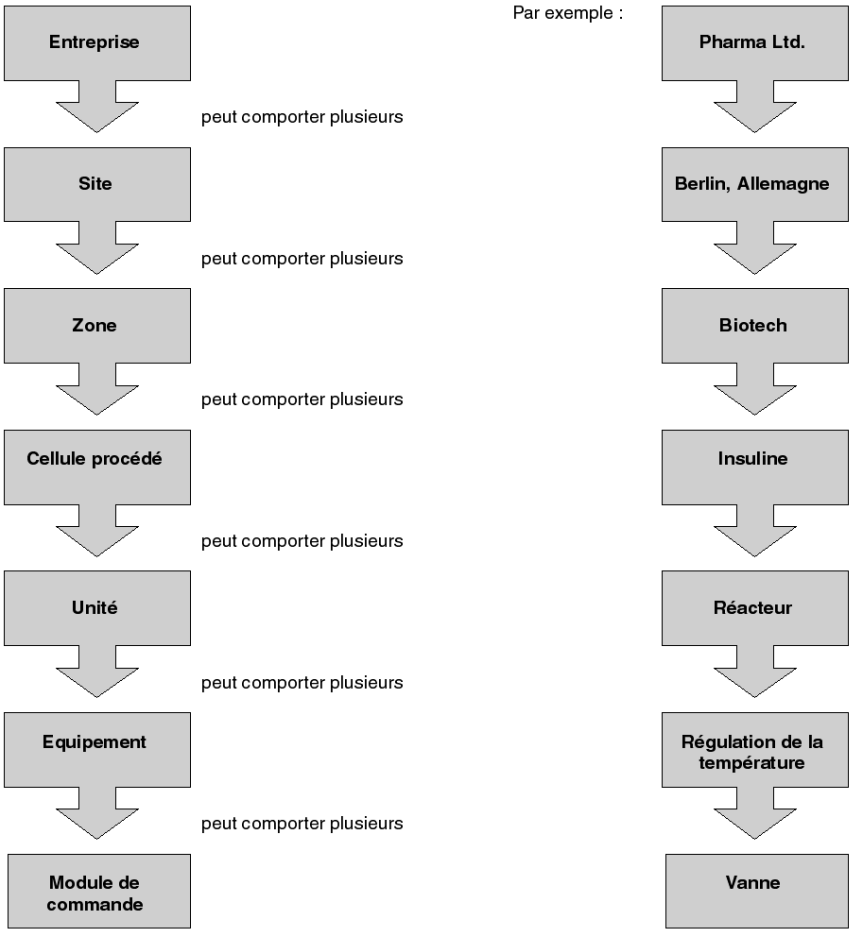
Sous-chapitre 4.1
Présentation du modèle physique

Structure du modèle physique

Structure

A l'aide du modèle physique, l'ingénieur procédé est en mesure de représenter la perspective du procédé au sein de Unity Application Generator.

Modèle physique selon l’ISA S88.01



Eléments du modèle physique

Le modèle physique de UAG débute par l'élément site. Dans UAG, le site constitue le sommet du modèle physique.

Les éléments suivants servent à structurer le procédé selon la norme ISA S88.01 :

- Site
- Zone
- Cellule procédé
- Unité
- Equipement
- Module de commande

La liste suivante décrit les éléments du modèle.

- Un site peut être constitué d'une ou de plusieurs zones.
- Une zone peut être constituée d'une ou de plusieurs unités.
- Une unité peut être constituée d'un ou de plusieurs équipements, modules de commande ou autres équipements de procédé. Au sein du groupe que constitue l'unité, il est possible d'exécuter une ou plusieurs fonctions de procédé.
- Un équipement est un groupe fonctionnel de modules de commande associant les modules de commande et de traitement physique nécessaires à la réalisation d'un certain nombre de tâches du procédé, par exemple un convoyeur. D'un point de vue fonctionnel, le domaine de l'équipement est défini par la tâche finie qu'il doit effectuer et pour laquelle il est conçu. Un équipement est généralement un ensemble de capteurs, d'actionneurs et de composants de procédé qui, du point de vue de l'automatisme, fonctionne comme une seule entité.
  - Il se représente comme une section dans l'API.
  - Il est représenté dans l'IHM comme un écran opérateur.
- Du point de vue contrôle-commande, les modules de commande représentent une seule et même entité constituée de capteurs, actionneurs et modules de régulation. Un module de commande peut représenter un objet réel pouvant être mesuré et surveillé, par exemple un moteur, une vanne, un transmetteur de température, etc., ou il représente un objet logiciel logique, utilisé dans le cadre de la régulation ou autres fonctions de commande (boucle PID, temporisateur, compteur, etc.). Tous les objets de plus haut rang sont constitués de modules de commande, destinés à former les objets de procédé plus complexes, tels que l'équipement. Dans UAG, les modules de commande sont des instances de types de module de commande (également appelés SCoD - modules de commande intelligents). Un SCoD possède tous les attributs qui lui sont associés. Pour plus d'informations, consultez *L'élément Module de commande et les types de module de commande (modules de commande intelligents - SCoD), page 73*.

NOTE : Certains des éléments décrits précédemment sont utilisés aux fins uniques de la structuration du procédé, d'autres sont responsables de la génération de code dans l'API ou l'IHM, etc. Tous ces éléments peuvent néanmoins se voir adjoindre différents types de fichiers permettant à l'utilisateur de documenter la conception du procédé dans le détail.

Définition du modèle physique

Unity Application Generator vous permet de définir par vous-même la structure de l'arborescence physique dans la personnalisation du projet.

Vous pouvez définir le nombre de niveaux physiques ainsi que les propriétés et le comportement de chacun de ces niveaux.

Pour des informations plus détaillées, veuillez vous reporter au *Manuel utilisateur avancé de Unity Application Generator (UAG)* section Définition du modèle physique *(voir Unity Application Generator (UAG) 3.3 SP4, a SoCollaborative software, Manuel utilisateur avancé)*.

Sous-chapitre 4.2
Eléments physiques en général

Éléments physiques, description et propriétés

Eléments physiques

Terme générique utilisé pour des objets, comme une zone, une cellule procédé, une unité et un équipement selon la définition utilisée dans le modèle physique de la norme ISA S88.01.

Les éléments physiques structurent les installations d'automatisation d'un site. Comme l'indique la norme ISA S88.01, cette structure est déterminée par des raisons physiques, géographiques ou logiques.

Les limites entre ces éléments sont généralement définies sur la base de critères organisationnels ou commerciaux, par opposition aux critères techniques. Il existe beaucoup d'autres facteurs autres que l'automatisme qui affectent ces limites.

2 types d'éléments physiques

Il existe 2 types d'éléments physiques dans Unity Application Generator :

- Eléments physiques structurels
  Ils structurent uniquement les installations d'automatisation d'un site et peuvent contenir d'autres éléments physiques.
- Eléments physiques fonctionnels
  Il s'agit d'entités fonctionnelles selon la définition des équipements dans le modèle physique de la norme ISA S88.01.
  Contrairement à la norme ISA S88.01, ils sont utilisables à chaque niveau hiérarchique du modèle physique dans Unity Application Generator.
  Les éléments physiques fonctionnels peuvent contenir d'autre éléments physiques, modules de commande et variables.

Eléments physiques structurels

Dans Unity Pro, les éléments physiques structurels sont représentés par des modules fonctionnels dans la vue fonctionnelle du navigateur de projet (sauf l'élément Site).

Pour ces éléments, il n'y a généralement aucun équivalent créé dans l'IHM.

Propriétés

Les éléments physiques ont comme propriétés **Nom** et **Description**.

En outre, il est possible d'attribuer un **Commentaire** et des références de **documents**.

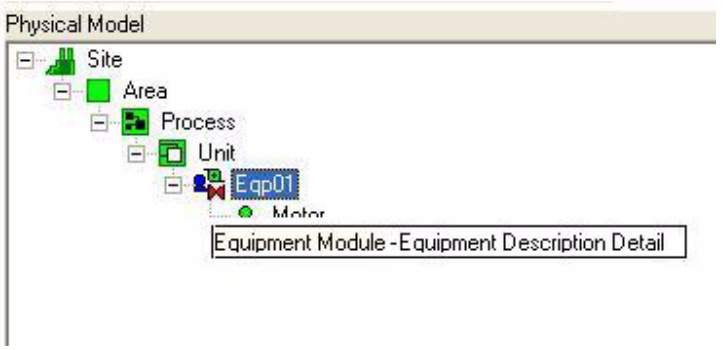
Propriétés des éléments physiques :

Propriété	Commentaire
Nom	Généralement un nom unique dans tout le projet. Comme beaucoup d'autres conventions, le nom utilisé pour les éléments physiques doit respecter les conventions de nommage définies dans la personnalisation. La personnalisation doit être adoptée avant de commencer le projet.
Description	Possibilité de donner une brève description de l'élément physique dans le champ Description.
Commentaire	Le champ commentaire est un champ texte libre. Il peut servir à documenter la conception du procédé.
Documents	Il est possible d'affecter un nombre quelconque de documents de type différent à chaque élément. Les types de fichiers disponibles sont définis lors de la personnalisation.

Pour plus d'informations sur la personnalisation, voir *Définition de la personnalisation, page 320*

Tous les éléments physiques du **Modèle physique** disposent d'une infobulle permettant d'obtenir des informations supplémentaires sur l'élément physique, ce qui comprend les détails fournis au champ **Description**.

La figure suivante illustre l'infobulle d'un élément physique :



Sous-chapitre 4.3
Elément physique fonctionnel

Introduction

Cette section décrit l'élément physique fonctionnel et ses propriétés.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Qu'est-ce qu'un élément physique fonctionnel ?	67
Caractéristiques de l'élément physique fonctionnel	68
Propriétés de l'élément physique fonctionnel	70
Types de modèles	72

Qu’est-ce qu’un élément physique fonctionnel ?

Qu’est-ce qu’un élément physique fonctionnel ?

Un élément physique fonctionnel est un groupe fonctionnel de modules de commande associant les modules de commande et de traitement physique nécessaires à la réalisation d’un nombre déterminé de tâches du procédé (comme les équipements dans le modèle physique de la norme ISA S88.01).

Citons comme exemples d’éléments physiques fonctionnels les convoyeurs ou les réacteurs. D’un point de vue fonctionnel, le domaine de l’élément physique fonctionnel est défini par la tâche finie qu’il doit effectuer et pour lequel il est conçu.

Un élément physique fonctionnel est à la base un ensemble de capteurs, d’actionneurs et de composants de procédé qui, du point de vue de l’automatisme, fonctionne comme une seule entité.

Caractéristiques de l'élément physique fonctionnel

Caractéristiques

La liste suivante décrit les caractéristiques d'un élément physique fonctionnel :

- Il effectue une tâche particulière du procédé pour la conversion, le transport ou le stockage de matériau ou d'énergie.
- Il est composé de modules de commande, dépendants les uns des autres d'un point de vue fonctionnel et qui échangent des informations pour effectuer la tâche de l'élément physique fonctionnel.
- Il ne communique pas directement avec les E/S du procédé. Les modules de commande sont connectés aux capteurs et actionneurs.
- Il peut communiquer directement avec l'IHM affichant les valeurs du procédé et fournit à l'opérateur les moyens d'agir de manière interactive avec le procédé.

Elément physique fonctionnel et API

- Généralement, un élément physique fonctionnel et tous ses modules de commande sont affectés à un API. Il est néanmoins possible d'affecter certains modules de commande du même élément physique fonctionnel à différents API.
- Dans Unity Pro, un élément physique fonctionnel est représenté sous forme de section FBD. Chaque module de commande est représenté par un bloc fonction (BF) au sein de cette section. Dans le navigateur de projet Unity Pro (vue fonctionnelle), cette section Elément physique fonctionnel se trouve dans la structure fonctionnelle du module (par exemple, selon la norme ISA S88.01) **Zone** →**Cellule procédé** →**Unité** →**Elément physique fonctionnel**.

Lignes de séparation

Dans Unity Pro, un élément physique fonctionnel est représenté sous forme de section FBD.

Si un élément physique fonctionnel génère un code API qui s'étend sur plusieurs sections, la césure entre les sections s'effectue en fonction de l'occupation des sections.

Vous pouvez regrouper les modules de commande d'un élément physique fonctionnel en sélectionnant **Insérer ligne de séparation** dans le menu contextuel d'un module de commande.

Les lignes de séparation des modules de commande seront utilisées pendant la génération du code.

Si le code programme qui est généré pour les modules de commande, regroupés au moyen des lignes de séparation, ne tient pas dans une section, une nouvelle section est créée.

Les lignes de séparation sont uniquement prises en compte pendant la génération complète d'API, pas lors de la génération incrémentale.

Le nombre de modules de commande par section API fourni dans la personnalisation a une priorité plus élevée que les lignes de séparation qui ont pour seule fonction de partager la section.

Exemple de lignes de séparation

On suppose que le **Nombre maxi. de modules de commande par section** est fixé à 5 dans l'éditeur de **Personnalisation de UAG**.

UAG possède un élément physique fonctionnel appelé **EQ1** avec 15 modules de commande enfants, affecté aux mêmes API et lignes de séparation après le 3ème et le 9ème module de commande :

```
EQ1 +--CM1 +--CM2 +--CM3 +======= +--CM4 +--CM5 +--CM6 +--CM7 +--CM8
+--CM9 +======= +--CM10 +--CM11 +--CM12 +--CM13 +--CM14 +--CM15
```

Les sections créées pendant la génération complète seront alors les suivantes :

EQ1 +--CM1 +--CM2 +--CM3 EQ1_01 +--CM4 +--CM5 +--CM6 +--CM7 +--CM8 EQ1_02 +--CM9 EQ1_03 +--CM10 +--CM11 +-- CM12 +--CM13 +--CM14 EQ1_04 +--CM15	Section EQ1 avec 3 modules de commande (CM1, CM2, CM3). La ligne de séparation partage les sections ici.
	Section EQ1_01 avec les 5 prochains modules de commande (CM4, CM5, CM6, CM7, CM8). La limite spécifiée dans la personnalisation est atteinte si bien que le 9ème module de commande sera dans la section suivante.
	Section EQ1_02 avec l'unique module de commande suivant (CM9). La ligne de séparation partage les sections ici.
	Section EQ1_03 avec les 5 prochains modules de commande (CM10, CM11, CM12, CM13, CM14). La limite spécifiée dans la personnalisation est atteinte si bien que le 15ème module de commande sera dans la section suivante.
	Section EQ1_04 avec le dernier module de commande (CM15).

Elément physique fonctionnel et IHM

Si configuré dans la boîte de dialogue Propriétés, Unity Application Generator génère une représentation graphique dans l'IHM pour chaque élément physique fonctionnel. La relation entre l'élément physique fonctionnel et l'IHM est définie par la zone de conduite et non directement par l'IHM.

Elément physique fonctionnel et variables

Après avoir généré la logique d'automate et les affectations IHM avec Unity Application Generator, il faut encore ajouter de la logique dans le programme Unity Pro et terminer la conception de l'IHM. Néanmoins, toutes les variables utilisées dans l'API et l'IHM sont définies dans Unity Application Generator. Ces variables doivent être affectées aux objets du modèle physique auquel elles appartiennent. C'est pourquoi le concepteur peut affecter d'autres variables à un élément physique fonctionnel.

Propriétés de l'élément physique fonctionnel

Présentation

Il faut définir 3 types de propriétés pour l'élément physique fonctionnel :

- Propriétés du procédé
- Propriétés relatives à l'API
- Propriétés relatives à l'IHM

Pour la première création d'un élément physique fonctionnel, l'ingénieur procédé ne doit définir que les propriétés du procédé.

Les caractéristiques relatives à l'API et à l'IHM peuvent être appliquées ultérieurement par l'ingénieur automaticien.

Propriétés générales

Propriétés générales de l'élément physique fonctionnel

Propriété	Commentaire
`Nom`	Chaque élément physique fonctionnel est identifié par un nom unique. Ce nom doit respecter les conventions de nommage définies par l'administrateur.
`Description`	Vous avez la possibilité de donner une brève description de l'élément physique fonctionnel dans le champ Description.
`Commentaire...`	Le champ commentaire est un champ texte libre. Il peut servir à documenter la conception du procédé.
`Documents...`	Il est possible d'affecter un nombre quelconque de documents de type différent à l'élément physique fonctionnel. Les types de fichiers disponibles sont définis lors de la personnalisation.

Propriétés relatives à l'API

Les propriétés relatives à l'API définissent la relation entre le modèle physique au sein duquel est créé l'élément physique fonctionnel et le modèle topologique de l'API.

Propriétés de l'élément physique fonctionnel relatives à l'API

Propriété	Commentaire
`Nom de l'API`	Définit l'API auquel appartient l'élément physique fonctionnel et où la logique est exécutée. L'API doit être défini dans le modèle topologique avant de pouvoir être affecté à l'élément physique fonctionnel.
`Nom de section API`	Définit le nom de la section au sein de Unity Pro. Le nom doit être unique et Unity Application Generator ne tolère pas l'utilisation de noms doubles.

Propriétés relatives à l'IHM

Les propriétés relatives à l'IHM définissent la relation entre le modèle physique au sein duquel est créé l'élément physique fonctionnel et le modèle topologique. La relation entre l'élément physique fonctionnel et l'IHM est définie par la zone de conduite et non directement par l'IHM.

Propriétés de l'élément physique fonctionnel relatives à l'IHM

Propriété	Commentaire
`Zone de conduite`	Définit à quelle zone de conduite l'élément physique fonctionnel appartient. La zone de conduite doit être définie dans le modèle topologique avant de pouvoir être affectée à l'élément physique fonctionnel. Pour plus d'informations, reportez-vous à la rubrique the HMI's node *(voir page 214)*.
`Image` (ne concerne pas iFIX)	Définit l'image de l'application IHM qui contient les symboles IHM appartenant aux modules de commande et variables de cet élément physique fonctionnel. L'image doit être définie dans le modèle topologique avant de pouvoir être affectée à l'élément physique fonctionnel. Pour plus d'informations, reportez-vous à la rubrique Pictures and Picture Groups *(voir page 219)*.

Types de modèles

Présentation

Dans UAG, vous pouvez créer et utiliser des types de modèles.

Un type de modèle est créé dans une bibliothèque de modèles.

Un type de modèle consiste en :

- un seul élément parent (similaire à un élément physique dans un projet)
- des modules de commande
- des variables

Une bibliothèque de modèles peut être utilisée dans différents projets UAG.

Lorsqu'un type de modèle est instancié dans un projet, il existe un élément physique avec des instances des modules de commande du type de modèle et des variables.

Pour des informations détaillées, veuillez consulter le *Manuel utilisateur avancé UAG*, chapitre *Bibliothèque de modèles*.

Sous-chapitre 4.4
L’élément Module de commande et les types de module de commande (modules de commande intelligents - SCoD)

Introduction

Ce chapitre décrit les propriétés des modules de commande et les fonctionnalités générales des types de module de commande.

Pour obtenir des informations détaillées sur des types de modules de commande particuliers, reportez-vous à la documentation spécifique relative à la bibliothèque des types de modules de commande en cliquant sur **Aide** sur **Type** dans la boîte de dialogue **Propriétés des modules de commande**.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Introduction	74
Caractéristiques d’un module de commande	75
Types de module de commande ou modules de commande intelligents (SCoD)	76
Module de commande libre	78
Propriétés d’un module de commande	79
Instruments	81
Forçages des modules de commande	82
Liaisons entre les SCoD	84

Introduction

Qu’est-ce qu’un module de commande ?

Du point de vue contrôle-commande, les modules de commande représentent une seule et même entité constituée de capteurs, d’actionneurs et d’équipements de régulation. Un module peut représenter un objet réel pouvant être mesuré et surveillé, comme un moteur, une vanne, un transmetteur de température, etc., ou il peut représenter un objet logiciel utilisé pour la régulation ou pour d’autres fonctions de commande, comme une boucle PID, un temporisateur ou un compteur.

Tous les objets de plus haut rang sont constitués de modules, destinés à former les objets de procédé plus complexes.

Le comportement particulier d’un module est défini dans le type de module, ce que l’on appelle le Module de commande intelligent (SCoD) duquel le module est dérivé, par exemple

- un variateur à deux vitesses,
- une vanne trois positions,
- un moteur inverseur,
- ...

Qu’est- ce qu’un type de module de commande ?

Un module de commande intelligent (SCoD) est un type de module prédéfini. Il désigne une partie spécifique du procédé et comprend tous les aspects fonctionnels de la tâche automatisée. Il tient compte de tous les aspects de l’objet technologique qu’il représente.

Il contient :

- la logique d’API nécessaire à la commande d’un module réel d’un type défini,
- la représentation de l’opérateur dans le système d’IHM,
- la commande manuelle du module par l’IHM,
- toutes les variables nécessaires à l’API et à l’IHM et
- une connexion vers la gestion d’alarme IHM.

Les types de module sont classés dans des bibliothèques standard fournies avec Unity Application Generator. Le client peut générer ses propres modules de commande ou Schneider Electric peut mettre en œuvre des types de module de commande spécifiques à un client et les intégrer dans des bibliothèques dédiées client, (par exemple, la bibliothèque *UAG Process Application Library pour Vijeo Citect*).

Si d’autres types de module spécifiques au projet sont nécessaires, veuillez contacter votre fournisseur Schneider Electric.

Caractéristiques d'un module de commande

Caractéristiques

La liste suivante décrit les caractéristiques d'un module de commande :

- Il représente un objet de procédé, comme par exemple un actionneur, un capteur ou une fonction du procédé qui manipule les données.
- Il fait partie d'un élément physique fonctionnel. Les modules de commande ne peuvent pas être partagés entre des éléments physiques fonctionnels.
- Il s'agit soit d'un type de module de commande spécifique (SCoD), soit d'un module de commande libre ; à un stade intermédiaire du développement d'un projet, il est aussi possible d'avoir des modules de commande sans type.
- Il peut communiquer avec les modules de commande d'autres éléments physiques fonctionnels.
- Il offre à l'IHM le (principal) point d'accès à la visualisation et à l'interaction avec le procédé.
- Il communique directement entre l'API et l'IHM.

Module de commande et IHM

- Chaque module de commande est représenté par un bloc fonction au sein de la section Unity Pro générée pour l'élément physique fonctionnel.
  Vous pouvez regrouper les modules de commande d'un élément physique fonctionnel en sélectionnant **Insérer ligne de séparation** dans le menu contextuel d'un module de commande. Voir *Lignes de séparation, page 68*.
- Un module de commande est affecté à un API. Il ne peut être fractionné entre plusieurs API.

Module de commande et IHM

- Un module est automatiquement affecté à l'IHM selon la définition dans les propriétés de l'élément physique fonctionnel.
- Chaque module de commande est représenté par un graphique de l'objet physique s'y rapportant, sur l'écran généré pour l'élément physique fonctionnel.

Modules de commande et variables

Après avoir généré la logique d'API et la représentation IHM avec Unity Application Generator, il faut encore ajouter de la logique dans le programme Unity Pro et achever le développement de l'IHM. Néanmoins, toutes les variables utilisées dans l'API et l'IHM doivent être définies par le modèle physique auquel elles appartiennent. C'est pourquoi le concepteur peut attribuer d'autres variables au module de commande, appelées variables libres.

Types de module de commande ou modules de commande intelligents (SCoD)

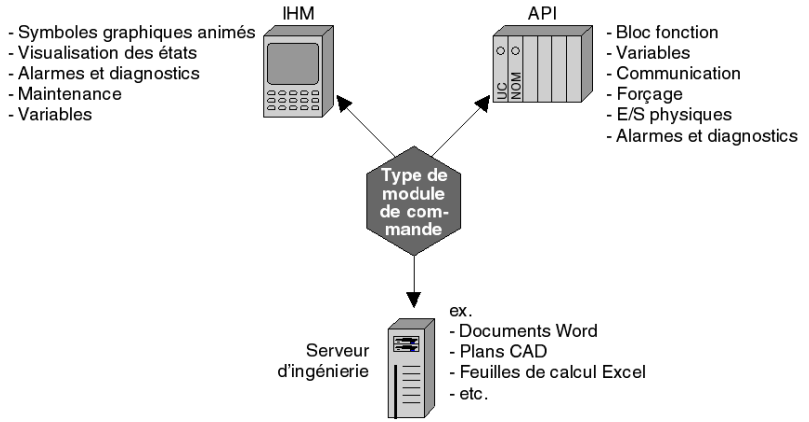
Introduction

Un type de module de commande ou SCoD inclut tous les aspects de la tâche automatisée.

Il s'agit d'un module de commande prédéfini qui tient compte de tous les aspects de l'objet technologique qu'il représente.

Que définit-on dans un type de module de commande ?

Illustration d'un type de module de commande



Pour l'API :

- La logique de pilotage du module de commande
- La logique de détection des dysfonctionnements internes et alarmes de la logique d'API
- La logique de détection des anomalies de procédé et de génération des alarmes
- La communication avec l'IHM
- Les attributs et variables de procédé affichés par l'IHM
- Les points d'E/S connectés au module de commande
- Les variables de la logique du module de commande
- Les commandes de pilotage du module de commande par la logique de l'API (p. ex. réinitialiser, démarrer, arrêter)
- La liaison avec les autres modules de commande
- La possibilité d'optimiser les valeurs par défaut
- La logique permettant la gestion du procédé par l'opérateur.

Pour l'IHM :

- La représentation graphique sur l'écran opérateur
- Les unités physiques devant être affichées à l'écran opérateur
- La communication avec l'API
- Les commandes opérateur de pilotage du module de commande (par exemple, bouton-poussoir marche/arrêt, invite à régler un point de consigne, etc.)
- La gestion des alarmes
- La consignation des actions opérateurs

Module de commande libre

Qu’est-ce qu’un module de commande libre ?

Un Module de commande libre est un module de commande qui n’est pas dérivé d’un type de module de commande. Il ne définit aucune logique de l’API ni aucune fonctionnalité prédéfinie pour l’IHM. Il est également possible de définir des variables libres pour un module de commande libre.

Rôle d’un module de commande libre

Certains types spéciaux de module de commande risquent de ne pas être disponibles pour tous les modules de procédé. Ces objets doivent néanmoins faire partie de la conception du procédé.

La logique de l’API et les graphiques de l’IHM seront définis manuellement mais toutes les variables nécessaires à l’accomplissement de la tâche en question doivent être définies au sein du module de commande libre.

NOTE : Toutes les variables de la totalité de la conception du procédé doivent être définies dans Unity Application Generator afin d’assurer la cohérence des données dans l’application d’automatisme.

Les modules de commande libres permettent à l’utilisateur de construire des objets placés de manière logique au bon emplacement dans le modèle physique.

Propriétés d’un module de commande

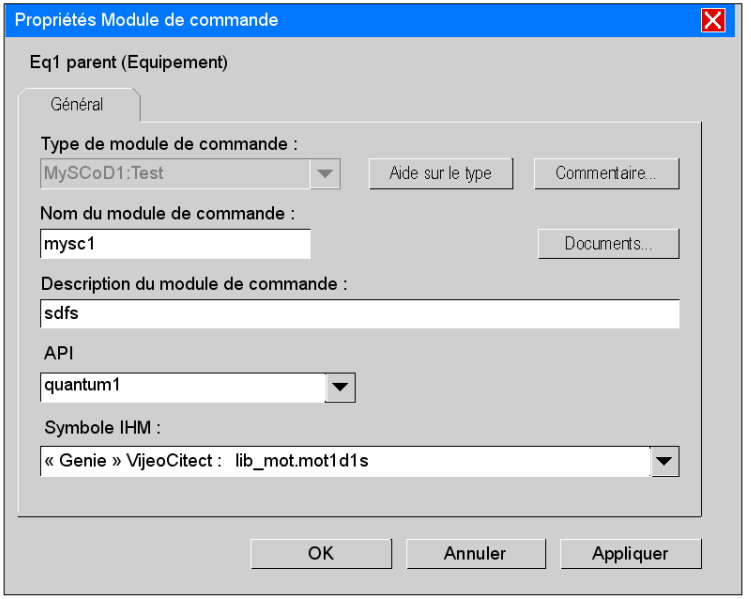
Présentation

Seules quelques propriétés sont nécessaires à la définition d’un module de commande. Le comportement du module de commande est tout d’abord défini par la personnalisation de ses propriétés, lesquelles diffèrent selon le type de module de commande.

Veuillez également consulter *Ajout de modules de commande à un type de modèle* dans le *Manuel utilisateur avancé UAG*.

Représentation

Propriétés Module de commande



Propriétés

Propriété	Commentaire
Type de module de commande	Définit le type de module de commande. Les options dépendent de la bibliothèque de types de modules de commande.
Nom du module de commande	Chaque module de commande est identifié par son nom unique au sein de l'élément physique fonctionnel auquel ce module appartient. Ce nom doit respecter les conventions de nommage définies lors de la personnalisation.
Description du module de commande	Vous avez la possibilité de donner une brève description du module de commande dans le champ Description.
Commentaire...	Le champ commentaire est un champ texte libre. Il peut servir à documenter la conception du procédé.
Documents...	Il est possible d'affecter un nombre quelconque de documents de type différent à chaque module de commande. Les types de fichiers disponibles sont définis lors de la personnalisation.
API	Chaque module de commande est affecté à un API. Par défaut, ce champ indique l'API défini dans l'élément physique fonctionnel parent correspondant. Il est possible de sélectionner un autre API si nécessaire. Le bloc fonction du module de commande sera traité sur l'API sélectionné.
Symbole IHM	Liste des symboles IHM pouvant être sélectionnés pour le module de commande. La liste des **symboles IHM** est définie dans l'éditeur SCoD. Elle n'est visible et modifiable que si elle est activée dans l'éditeur SCoD.

Instruments

Qu’est-ce qu’un instrument ?

Un instrument est un capteur, un actionneur ou un équipement de traitement associé quelconque, utilisé dans le procédé. Un instrument devient un module de commande dès qu’il est intégré au modèle physique. Lesinstruments sont répertoriés dans la liste d’instruments. Depuis celle-ci, ils peuvent être déplacés vers le modèle physique.

Reportez-vous également à : *Utilisation de la liste d’instruments, page 261*

A quoi correspond-il ?

En temps qu’étape préliminaire d’un module de commande, l’instrument a été défini afin de permettre à l’ingénieur procédé de saisir une liste brute des instruments **avant** de définir l’ensemble de la hiérarchie du modèle physique.

La création d’instruments dans la liste d’instruments est très efficace et permet de gagner du temps par rapport à la création de modules de commande dans l’arborescence du modèle physique. Il est possible de gagner encore plus de temps en important les instruments du schéma PID (schéma de tuyauterie et d’instrumentation).

Propriétés

Un instrument possède les propriétés suivantes :

- Nom
- Description
- Type de module de commande
- Commentaires…
- Documents...
- Propriétés spécifiques au type de module de commande
- Variables

NOTE : Comparé à un module de commande, un instrument ne peut se voir affecter **ni** API, **ni** forçages.

Il est possible de créer un instrument sans définir de type de module de commande (`type_module_commande = Non_affecté`). Un instrument sans type peut être déplacé vers le modèle physique sans restriction. Ce sera un module de commande sans type au sein du modèle physique.

Forçages des modules de commande

Représentation

Pour les modules de commande comportant des entrées de forçage (par exemple les moteurs, vannes, commandes d'axe, pompes), il est possible de définir des commandes de forçage dans le modèle physique. Un module de commande peut posséder une ou plusieurs entrées de forçage, ces dernières étant définies comme broches de forçage dans les SCoD.

Unity Application Generator génère le code API sous forme de réseau de blocs fonctions pour toutes les définitions de forçage API si leur syntaxe est correcte. Si la syntaxe n'est pas valide, Unity Application Generator génère la description littérale de la définition du forçage comme commentaire pour le bloc fonction correspondant sous Unity Pro. Dans ce cas, la logique d'API doit être complétée en conséquence par l'ingénieur automaticien.

L'éditeur de forçage vous permet de connecter directement une variable à la broche de forçage du module de commande. Pour cela, vous pouvez sélectionner la case à cocher **Connecter directement à la variable** et sélectionner ensuite la variable souhaitée à partir de la fenêtre de navigation ou par glisser-déposer de la variable.

Pour plus d'informations, reportez-vous à la section How to Build an Interlock Definition *(voir page 266)*.

Restriction

Les restrictions suivantes doivent être prises en compte :

- Il n'est pas possible de définir de forçage pour les modules de commande sans entrée de forçage. Les entrées de forçage (broches de forçage) se définissent dans les SCoD.
- Lors de l'importation d'une importante définition de forçage (comportant plus de 150 conditions de forçage), vous ne pourrez pas voir les conditions de forçage dans l'éditeur graphique de forçage. Les conditions de forçage seront générées dans le projet d'API et seront alors visibles dans ce projet.

Définition du forçage

Une définition du forçage est formée de un ou plusieurs forçages et est affectée à une entrée de forçage spécifique. Une définition de forçage peut être établie pour chaque entrée de type forçage d'un module de commande.

Un forçage se définit par

- des conditions de forçage ;
- des opérateurs logiques permettant d'associer les conditions de forçage (AND, OR, XOR, NOT) ;
- des parenthèses d'ouverture et de fermeture.

Exemple de définition de forçage :

```
((Boiler1_Motor1_FTR = 1) And (Boiler1_Valve1_FT >= 100)) Or
(Boiler1_AIn1_AHH = 1)
```

Copie des forçages

La définition du forçage fait partie de la définition du module de commande.

Un module de commande fait partie d'un élément physique fonctionnel.

Considérons la mise en œuvre de l'élément physique fonctionnel suivant : une section d'entrée d'un réservoir avec une vanne et une pompe.

La pompe possède, par exemple, un forçage. Ce forçage indique que la pompe ne fonctionnera pas si le forçage est défini. Ce comportement est commun à toutes les sections d'entrée des réservoirs de l'usine.

Lorsque vous copiez un élément physique fonctionnel, le forçage sera collé au niveau du nouvel élément physique fonctionnel. Les variables utilisées dans la condition de forçage seront remplacées par les nouvelles variables du module de commande.

Exemple :

Un élément physique fonctionnel `InletletTank1` est copié comme `InletTank2`.

La variable de module de commande associée `InletTank1_Valve1_OPND` est remplacée par la nouvelle variable du module de commande `InletTank2_Valve2_OPND`.

Liaisons entre les SCoD

Introduction

Les broches d'entrée, de sortie et d'entrée/sortie des ScoD peuvent être liées aux autres broches des ScoD ou aux variables libres.

Boîte de dialogue Ouvrir lien(s)

Le menu contextuel d'un module de commande possède une option **Ouvrir liaison**.

Si cette boîte de dialogue est ouverte, une table affiche les broches d'entrée, de sortie et d'entrée/de sortie du module de commande.

Le menu contextuel d'un élément superordonné quelconque du **modèle physique** (par ex. une zone, une cellule procédé, une unité, etc.) présente une option appelée **Ouvrir liaisons**.

Si cette boîte de dialogue est ouverte, une table affiche toutes les broches d'entrée, de sortie et d'entrée/de sortie de tous les modules de commande subordonnés.

Pour lier une variable à une broche d'un module de commande, effectuez un glisser-déposer de cette variable à partir de la table des variables vers la table de liaison.

En outre, une variable de liaison peut également être déplacée dans la table de liaison d'un module de commande par glisser-déposer.

Liaisons dans Unity Pro

Si une liaison a été créée dans UAG, elle est générée dans Unity Pro lors de la génération du programme API.

Si la variable de liaison est la broche d'un SCoD qui existe dans la même section que la broche d'entrée, une liaison graphique est créée. Sinon, la broche est connectée à la variable.

Si la case à cocher **Créer des liens uniquement sous forme de variables** est cochée dans le dialogue **Options** de UAG, la broche sera alors connectée dans tous les cas à la variable et aucun lien graphique n'est créé. Le paramétrage de cette option n'a pas d'incidence sur les liens déjà créés.

Plusieurs connexions

Une variable de liaison peut être utilisée plusieurs fois dans la table de liaison de modules de commande identiques ou différents. Plusieurs connexions sont donc possibles.

Restrictions

Les restrictions suivantes doivent être prises en compte :

- Le type de données de la broche d'entrée et celui de la variable de liaison doivent être identiques.
- Les variables de type de données IHM ne peuvent pas être liées (elles ne font pas partie de la section API).
- Les types de données dérivés (DDT) ou les éléments des DDT peuvent être liés à des broches d'entrée, de sortie et d'entrée/de sortie.
- Les variables peuvent être liées à des broches en sélectionnant **Non connecté** dans la liste **Utilisation des broches**.
  Même si une variable **E/S_API** est connectée à une broche dans l'éditeur SCoD, l'utilisateur peut créer une liaison à cette broche dans UAG si l'une des conditions suivantes est remplie :
  - Dans l'éditeur SCoD, sélection obligatoire de **Non** dans la liste **Variable utilisée**.
  - Dans le projet UAG, sélection obligatoire de **Non** dans la liste **Variable utilisée**.

  Dans la liste **Variable utilisée**, si vous passez de la valeur **Non** à la valeur **Oui** dans le projet UAG, la liaison existante à cette broche sera alors supprimée après confirmation.
  Dans la liste **Variable utilisée**, si vous sélectionnez **Non** pour une variable **E/S_API**, cette variable ne pourra être utilisée nulle part ailleurs dans UAG (liaisons, forçages, canaux, points d'E/S).

Sous-chapitre 4.5
Variables

Introduction

Cette partie décrit les propriétés des variables et leurs possibilités d'utilisation.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Introduction	87
Types de connexions et types de données de variables	88
Propriétés générales des variables	90
Propriétés des variables liées à l'API	99
Propriétés des variables relatives à l'IHM	102
Définition d'alarme dans UAG	108
Onglet Utilisations des propriétés de variables	114

Introduction

De l'importance des variables

Dans les outils habituels de programmation d'API ou d'IHM, les variables sont utilisées partout. L'inconvénient est que ces variables doivent être définies séparément dans chaque système. Par conséquent, si une variable est nécessaire dans les deux systèmes, elle doit être définie deux fois. La synchronisation manuelle des variables est une des causes principales d'erreur dans une application de contrôle-commande. Avec Unity Application Generator, toutes les variables ne sont définies qu'une seule fois et sont générées automatiquement dans tous les systèmes qui en ont besoin.

A quels objets les variables sont-elles affectées ?

Les variables font généralement référence à un type de module de commande. Elles sont générées automatiquement pour chaque instance d'un type de module de commande. Ces variables sont appelées variables de type de module de commande.

De plus, il est possible d'ajouter des variables aux éléments suivants :

- Eléments physiques fonctionnels et
- modules de commande

Ces variables sont appelées variables libres.

NOTE : Les modules de commande sans type n'ont pas de variables.

Conventions de nommage

Les variables doivent être nommées conformément aux conventions de nommage définies lors de la personnalisation. Le nom ne doit être unique qu'au sein de l'élément auquel il se réfère. Le nom dans l'API ou dans l'IHM est la concaténation du nom de variable et du nom d'objet auquel il est affecté.

Le tableau suivant illustre des exemples de noms de variable :

Nom de variable	Inexploitable
Variable relative à un module de commande	`Motor1`
Module de commande relatif à l'élément physique fonctionnel	`EQ411A`
Nom de variable résultant dans Unity Pro et l'IHM	`EQ411A_Moteur1_Inexploitable`

Types de connexions et types de données de variables

Types de connexions

Si une variable est définie dans Unity Pro ou dans une IHM, le système auquel elle appartient est évident. Si une variable est définie dans Unity Application Generator, il est nécessaire de définir à quelle partie de l'automatisme elle appartient. Le type de connexion nécessaire dépend du lieu où est utilisée la variable.

Le tableau suivant illustre des types simples de connexions de variables

Type de connexion	Commentaire
`API`	Utilisée uniquement en interne à l'API
`ES_API`	Variable utilisée dans la logique de l'API et connectée à un point d'E/S physique
`IHM`	Utilisée uniquement en interne à l'IHM

Bien souvent les variables ne sont pas seulement utilisées dans une partie de l'automatisme mais dans plusieurs parties, par exemple l'API et l'IHM. Dans l'approche classique d'élaboration d'un automatisme, cela implique de devoir définir les variables deux fois. Avec Unity Application Generator, cela n'est effectué qu'une seule fois grâce à l'association de deux types de connexions. Les variables de ces types de connexions sont transmises aux différentes parties de l'automatisme.

Le tableau suivant illustre des combinaisons de types de connexions de variables

Type de connexion	Commentaire
`API_IHM`	La variable est communiquée entre l'API et l'IHM
`API_RES`	La variable est communiquée entre l'API et un abonné réseau, par exemple un programmateur de poche

Types de données des variables

Il faut définir le type de données de chaque variable. Les types de données disponibles dépendent de la catégorie de la variable. En général, tous les types de données du système cible sont disponibles. Unity Application Generator génère également les types de données structurés de Unity Pro, à l'exception des types de données génériques (GDT).

Traitement des valeurs initiales dans Unity Pro

Il est possible d'affecter une valeur initiale à chaque variable (excepté aux variables `IODDT` et aux variables structurées). Il n'existe aucune restriction pour les valeurs initiales des variables localisées du type `BOOL`.

Propriétés des variables et des modules de commande librement définies par les utilisateurs

Certaines propriétés des variables et des modules de commande peuvent être configurées dans l'éditeur SCoD et utilisées dans des applications UAG.

Les types de propriétés de base sont, par exemple, `Commande`, `Valeur initiale`, `Alarme`, `Unité de mesure`, etc.

Dans certains cas, vous pouvez souhaiter décrire les variables et les modules de commande au moyen d'autres propriétés qui ne sont pas prédéfinies dans UAG. Elles sont visibles dans l'intégralité du système, sous la forme de propriétés normales, avec une possibilité d'héritage.

Une caractéristique intéressante des propriétés librement définies par l'utilisateur concerne la génération d'IHM génériques. Les propriétés seront incluses dans les fichiers XML et CSV. Vous pourrez ainsi les importer dans un système IHM spécifique.

Les propriétés librement définies sont spécifiées dans l'éditeur SCoD, comme les propriétés normales. Le nombre de propriétés librement définies est fixé à 20. Les propriétés prennent la forme de chaînes. Aucun contrôle du type de données n'est effectué dans UAG.

Propriétés générales des variables

Variables de type de module de commande

Les propriétés des variables issues d'un type de module de commande sont pour la plupart prédéfinies. Seul le concepteur peut modifier les champs nécessaires.

Variables libres

Les variables libres définies par le concepteur du procédé n'ont pas de propriétés prédéfinies. Les propriétés doivent être définies de manière détaillée. Les propriétés disponibles dépendent des catégories de la variable. Les variables relatives à l'IHM ont plusieurs autres propriétés qui définissent leur comportement au sein de l'IHM. Si vous utilisez Monitor Pro, Vijeo Citect ou iFIX, une variable libre peut être représentée par un symbole graphique (option dans les propriétés générales).

Propriétés de base

Le tableau suivant illustre les propriétés disponibles pour une variable

Propriété	Description	Options	Commentaire
`Nom`	Nom unique de variable	-	Le nom du module de commande ou de l'élément physique fonctionnel doit être unique. Ce nom doit respecter les conventions de nommage définies lors de la personnalisation.
`Description`	Description de la variable	-	Champ texte libre Ce texte fait partie par défaut du texte d'alarme.
`Type de connexion`	Définit le type de connexion des variables	• API • IHM • ES_API • API_IHM • API_RES	Reportez-vous à *Types de connexions et types de données de variables, page 88*.

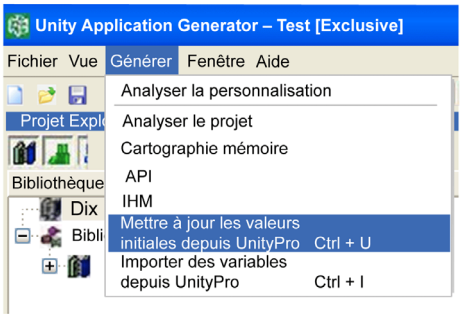
Propriété	Description	Options	Commentaire
Type de données	Type de données de la variable	• ANL_IN • ANL_OUT • BOOL • BYTE • DATE • DINT • DT • DWORD • INT • REAL • STRING • STRING[n] • TIME • TOD • UDINT • UINT • WORD • Types de données dérivés de Unity Pro	Les options disponibles dépendent du type de connexion. Pour STRING[n], la valeur maximale de n est 1000.
Entrée/Sortie	Définit l'utilisation de la variable	• Entrée • Sortie • Entrée/Sortie	La variable est connectée à une entrée et à une sortie si le bloc fonction change la valeur de la variable.
Variable utilisée	Permet de définir l'état utilisé ou non utilisé d'une variable E/S_API d'un SCoD	• Oui • Non	Une variable peut être définie comme étant non utilisée, par exemple parce qu'un objet réel "vanne" n'a pas de fin de course de retour mais que le type de module en offre un. Les variables non utilisées ne seront pas générées pour Unity Pro.
Inverser la variable	Inverse la variable (concerne uniquement les variables ES_API)	-	-
Valeur initiale	Valeur initiale de la variable	-	La valeur doit correspondre au Type de données des variables. Selon le Type de données de la variable, des vérifications supplémentaires liées aux valeurs brutes et aux échelles peuvent être effectuées.
Actualiser valeur initiale	Définit l'attribut de sauvegarde dans Unity Pro.	-	Pour Unity Pro seulement

Propriété	Description	Options	Commentaire
`Mode Eone morte`	Définit le mode Bande morte d'une variable	• Non_affecté(e) • Absolu(e) • Pourcentage	Non utilisé pour iFIX et Monitor Pro
`Valeur Zone morte`	La zone morte est une quantité, absolue ou en pourcentage, qu'une valeur de variable peut modifier sans que cette valeur ne soit actualisée dans le système. La `valeur zone morte` correspond à la valeur `Echelle min/max`. Il vous appartient d'introduire des données correctes.	• Absolu(e) Saisissez une valeur correspondant au type de données de la variable. • Pourcentage Saisissez une valeur de pourcentage positive.	Non utilisé(e) pour iFIX et Monitor Pro

Mettre à jour les valeurs initiales depuis UnityPro

Vous pouvez mettre à jour les valeurs initiales de toutes les variables générées pas UAG dans UAG depuis UnityPro à l'aide de l'option **Mettre à jour les valeurs initiales depuis UnityPro**.

Si vous cliquez sur **Générer** →**Mettre à jour les valeurs initiales depuis UnityPro** depuis la barre de menus, toutes les variables générées du projet UAG, peu importe leur API, sont mises à jour avec leurs valeurs actuelles depuis leurs projets UnityPro respectifs.



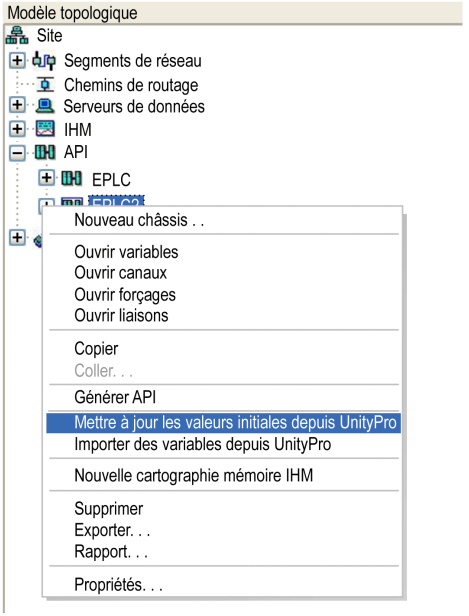
NOTE : L'option **Mettre à jour les valeurs initiales depuis UnityPro** ne peut être activée que si au moins un projet API est généré.

Les valeurs initiales seront mises à jour si leurs valeurs sont modifiées dans UnityPro.

Une fois les valeurs initiales des variables mises à jour depuis UnityPro, vous ne pouvez annuler la mise à jour.

Cette option s'affiche également dans le menu contextuel de tout API dans le **Modèle topologique**. Une fois cette option sélectionnée, seules les variables des API sélectionnés sont mises à jour avec leurs valeurs initiales depuis UnityPro.

NOTE : Les valeurs initiales des variables de DDT de l'appareil, générées par UAG pour l'API M580, ne sont pas mises à jour.



L'**Historique des changements** est mis à jour respectivement pour toutes les variables dont les valeurs ont été modifiées.

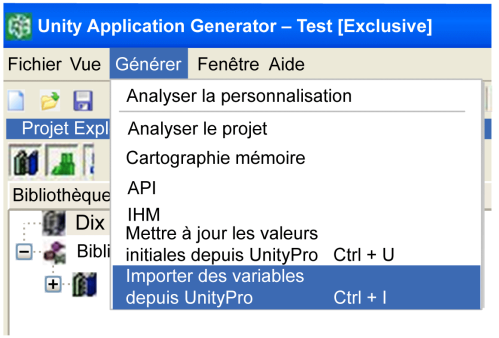
Importer des variables depuis UnityPro

Vous pouvez importer des variables depuis UnityPro à l'aide de l'option **Importer des variables depuis UnityPro**. Grâce à cette fonctionnalité, les variables créées dans UnityPro se synchronisent avec les variables de UAG.

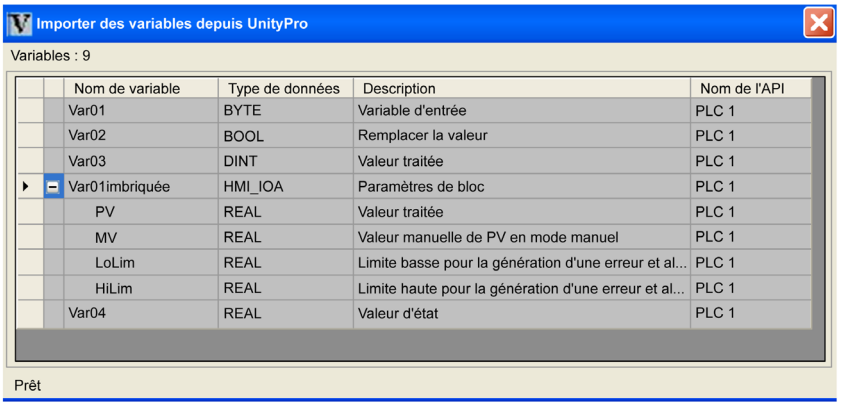
NOTE : Les variables de DDT de l'appareil, générées par UAG pour l'API M580, ne sont pas importées.

Cette fonctionnalité s'active uniquement lorsque :

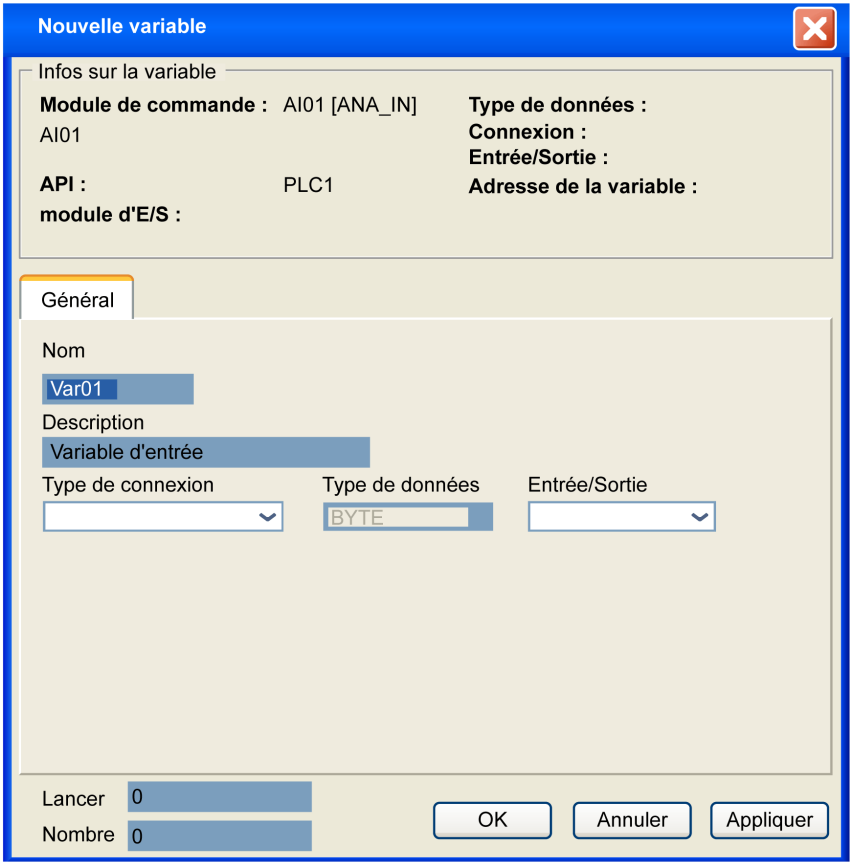
- un projet UAG est ouvert en mode exclusif.
- les projets Unity Pro sont générés depuis UAG.



Etape	Action
1	Cliquez sur **Générer →Importer des variables depuis UnityPro** depuis la barre de menus. **Résultat :** Les variables qui sont créées dans Unity Pro sont importées et répertoriées dans la boîte de dialogue **Importer des variables depuis UnityPro**. Les variables sont importées dans UAG si le type de données de la variable est pris en charge.



Nom de variable	Type de données	Description	Nom de l'API
Var01	BYTE	Variable d'entrée	PLC 1
Var02	BOOL	Remplacer la valeur	PLC 1
Var03	DINT	Valeur traitée	PLC 1
Var01imbriquée	HMI_IOA	Paramètres de bloc	PLC 1
PV	REAL	Valeur traitée	PLC 1
MV	REAL	Valeur manuelle de PV en mode manuel	PLC 1
LoLim	REAL	Limite basse pour la génération d'une erreur et al...	PLC 1
HiLim	REAL	Limite haute pour la génération d'une erreur et al...	PLC 1
Var04	REAL	Valeur d'état	PLC 1

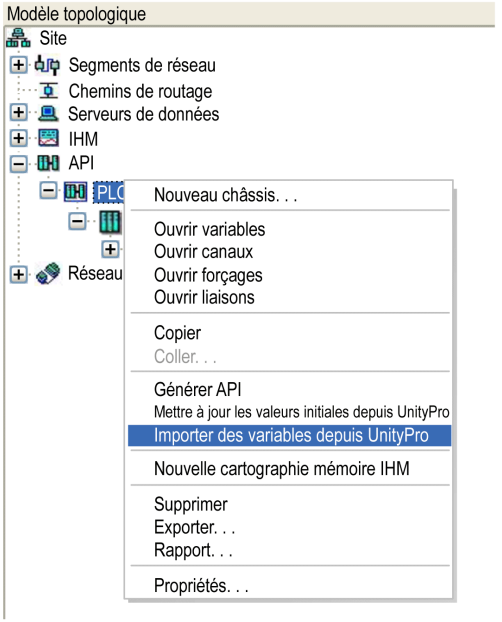
Etape	Action
2	Sélectionnez une variable depuis la boîte de dialogue **Importer des variables depuis UnityPro**.
3	Faites glisser la variable sélectionnée et déposez-la sur l'élément physique respectif ou sur le module de commande. **Résultat :** La boîte de dialogue de la **Nouvelle variable** apparaît et elle affiche la variable déplacée avec les valeurs de l'onglet **Général**. 
4	Sélectionnez un type de connexion depuis la liste **Type de connexion**.
5	Sélectionnez un type d'entrée/sortie depuis la liste **Entrée/sortie**.
6	Cliquez sur **Appliquer**. **Résultat :**Une variable est créée dans UAG comme variable libre. D'autres propriétés de la variable à configurer s'affichent. La variable créée est supprimée de la boîte de dialogue **Importer des variables depuis UnityPro**.

Les règles suivantes sont applicables pour les variables qui sont importées :

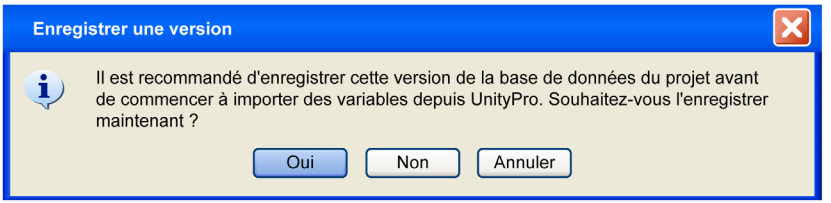
- Vous ne pouvez pas faire glisser plusieurs variables vers les éléments physiques/modules de commande en une seule fois.
- L'importation des variables depuis UnityPro en mode lecture seule de UAG n'est pas possible.
- Il affiche les variables qui ne sont pas générées à partir de UAG.
- Les informations de la variable comme le nom, la description et le type de données sont importées depuis UnityPro avec le nom de l'API.
- Vous ne pouvez pas glisser et déposer la variable dans l'élément physique/le module de commande dans lequel l'API n'est pas affecté.
- Vous pouvez glisser et déposer la variable du même API. Les variables sont affectées à l'élément de l'API à partir duquel elles ont été importées. Par exemple, si vous sélectionnez la variable dans l'API1 et si vous la faites glisser dans l'élément physique/le module de commande attribué à API2, cette procédure n'est pas autorisée.
- Si le type de données n'est pas disponible ou n'est pas validé dans l'éditeur SCoD, les variables ne sont pas importées.
- Si la même variable comportant des données de type différent ou des données de même type avec des valeurs de propriété différentes existe pendant l'importation des variables, une notification indiquant que la variable existe déjà s'affiche lors de la création de la variable.
- Si vous créez une variable et que le nom de celle-ci n'est pas conforme aux règles de création des noms spécifiées dans la personnalisation de projet, vous devez modifier le nom de la variable dans Unity Pro pour corriger la différence et l'importer à nouveau.
- Le nom de la variable créée dans UnityPro est toujours considéré comme le nom de la variable, alors que UAG génère un nom complet de la variable basé sur la règle de création du nom parent. Dans la génération d'API et IHM suivante, les variables sont créées conformément à la règle de création de nom spécifiée dans la personnalisation de projet.
- Une fois l'importation des variables dans UAG achevée, ces dernières remplacent les variables de UnityPro dans la génération d'API suivante. En cas de génération d'IHM, ces variables sont générées dans le cadre de la génération normale.
- Si la nouvelle variable est créée dans UAG et est supprimée avant la génération, une nouvelle variable est créée à nouveau avec le même nom, indépendamment de la variable importée de UnityPro. Cette variable Unity Pro ne change pas au cours de la génération incrémentale d'API suivante depuis UAG.

L'**historique des changements** est mis à jour respectivement pour toutes les variables importées avec leurs valeurs une fois l'importation des variables depuis Unity Pro réussie.

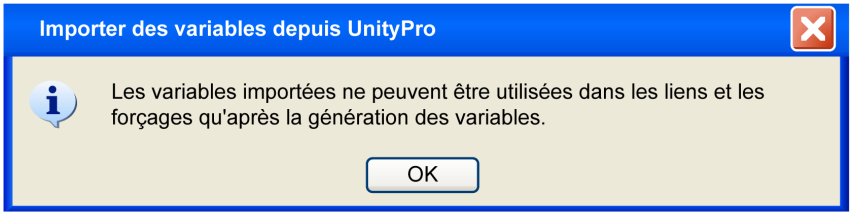
Cette option s'affiche également dans le menu contextuel de tout API dans le **modèle topologique**. Une fois cette option sélectionnée, l'importation des variables est limitée à l'API sélectionné.



Après avoir cliqué sur **Importer des variables depuis UnityPro** dans **Générer →Importer des variables depuis UnityPro** ou le menu contextuel, un message s'affiche pour enregistrer le projet avant le début effectif de l'importation. L'enregistrement d'un projet vous permet de récupérer la copie du projet enregistré avant l'importation des variables depuis UnityPro ultérieurement si nécessaire.



Lors de l'importation des variables, une notification apparaît et affiche le message **Les variables importées ne peuvent être utilisées dans les liens et les verrouillages qu'après la génération des variables**. Après avoir accusé réception du message, la boîte de dialogue**Importer des variables depuis UnityPro** apparaît.



Onglet Utilisations

L'onglet **Utilisations** de la boîte de dialogue **Propriétés de variable** donne des renseignements sur les utilisations de la variable.

Reportez-vous à *Onglet Utilisations des propriétés de variables, page 114*.

Propriétés des variables liées à l'API

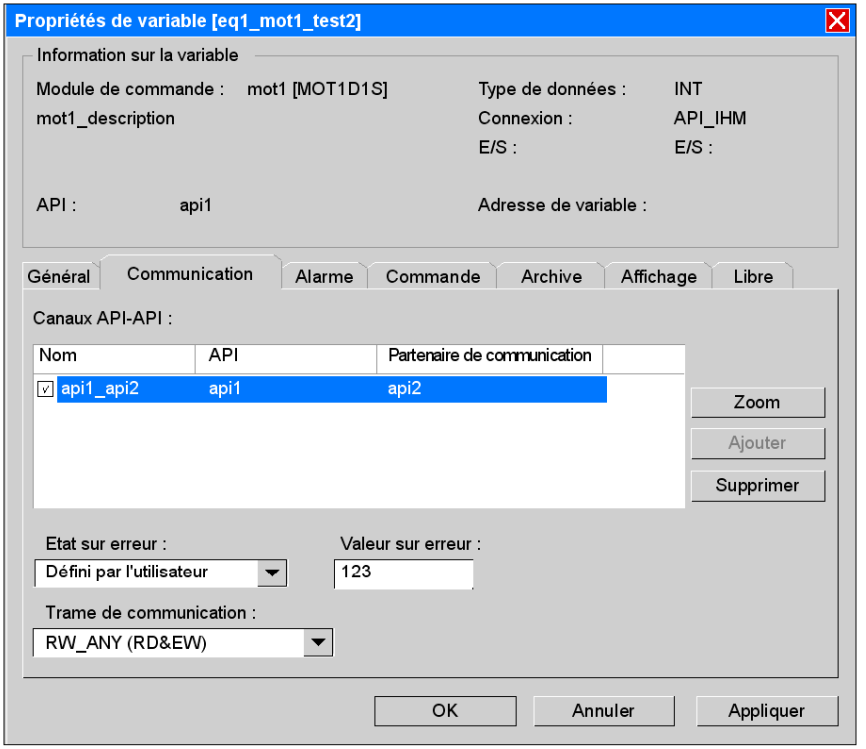
Présentation

L'onglet **Communication** de la boîte de dialogue **Propriétés de variable** donne des renseignements sur la trame de communication API-IHM.

Lorsque vous cliquez sur l'onglet **Communication**, si un canal est précédé d'une coche, cela signifie que la variable (ex. : `eq1_mot1_test2`) est incluse dans le canal en question (`API1_API2`).

Onglet Communication

Le diagramme ci-dessous illustre l'onglet **Communication** de la boîte de dialogue **Propriétés de variable**.



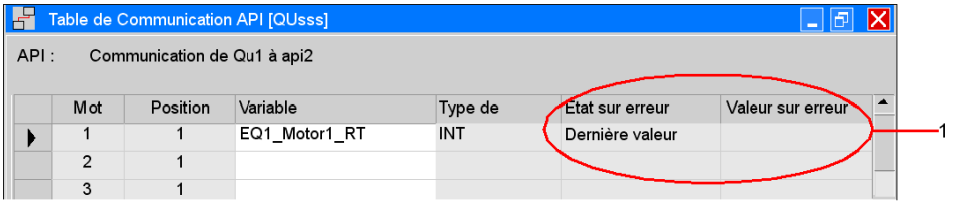
Propriétés de l'onglet **Communication**

Propriété	Description
Tableau **Canaux API-API**	Répertorie tous les canaux déjà configurés dans votre projet
Nom	Nom du canal API-API Lorsque vous cliquez sur l'onglet **Communication**, si un canal est précédé d'une coche, cela signifie que la variable (ex. : `eq1_mot1_test2`) est incluse dans le canal en question (`API1_API2`).
API	API source
Communication	Equipement cible (API)
Zoom	Ouvre la boîte de dialogue **Sélection des canaux à ajouter** afin de sélectionner un canal existant
Ajouter	Active la case à cocher dans la colonne nom du canal sélectionné La variable sera insérée dans les canaux sélectionnés (si ce n'est pas déjà le cas). Vous pouvez activer/désactiver la case à cocher en cliquant dessus.
Supprimer	Désactive la case à cocher dans la colonne nom du canal sélectionné
Etat d'erreur	Définit l'action sur timeout d'une variable Si une erreur est détectée, vous pouvez définir les répercussions sur le signal comme suit : ● Non_Affecté(e) (aucune affectation d'action n'a eu lieu) ● Non activé : le signal est mis à 0 ● Dernière valeur : le signal conserve la dernière valeur connue avant la détection de l'erreur de timeout. ● Défini par l'utilisateur : l'utilisateur peut définir l'état par défaut du signal en cas de détection d'une erreur de timeout.
Valeur d'erreur	Définit la valeur du signal si un événement de timeout survient. Cette propriété est uniquement disponible si **Etat d'erreur** est réglé sur **Défini par l'utilisateur**.
Trame de communication	Définit la trame de communication utilisée pour échanger les données de variables avec l'IHM, et n'est disponible que pour des variables de type de connexion API_IHM Les trames affichées dépendent de la liste des trames disponibles définies lors de la personnalisation et du type de données de la variable.

Etat d’erreur/Valeur d’erreur à lecture seule

Les attributs **Etat d’erreur** et **Valeur d’erreur** sont affectés à la variable dans la boîte de dialogue **Propriétés de variable**.

Dans la **table de communication API**, ces informations sont en lecture seule.



1 Lecture seule

Propriétés des variables relatives à l'IHM

Présentation

Les propriétés relatives à l'IHM sont disponibles uniquement pour les variables du type de connexion `IHM` ou `API_IHM`.

Ces propriétés sont constituées des catégories suivantes, selon l'élément auquel elles se réfèrent :

- Propriétés relatives à l'alarme
- Propriétés relatives à la commande
- Propriétés relatives à l'archive
- Propriétés relatives à l'affichage

Propriétés relatives aux alarmes

Ces propriétés sont disponibles pour des variables de type de données élémentaires (sauf le type STRING)

Le nombre d'alarmes qui peut être défini pour une variable dépend du type de données de la variable. Pour une variable booléenne, vous ne pouvez définir qu'une seule alarme.

Veuillez consulter la sectionDéfinition d'alarme dans UAG *(voir page 108)*.

Propriété	Description	Options	Commentaire
`Alarme`	Définit si cette variable est une alarme	• Oui • Non	-
`Groupe d'alarmes`	Définit le groupement des alarmes	Sélection dans la liste personnalisée	Les groupes d'alarmes sont définis au moment de la personnalisation.

Propriété	Description	Options	Commentaire
Opérateur d'alarme	Opérateur de comparaison pour comparer la valeur de la variable avec la limite de l'alarme correspondante.	Les opérateurs possibles sont =, <>, >, <, >=, <=	Non disponible pour les variables booléennes ("=")
	expression libre	Expression libre	Par exemple, avec Vijeo Citect, vous pouvez utiliser Expression libre pour définir une expression Cicode d'une alarme dans la propriété Limite de l'alarme.
	valeurs de bits de la variable en cas d'alarme liée aux bits	Bit X (Bit 0 ... Bit n-1)	Le nombre de valeurs de bits dépend du type de données de la variable (par ex. pour INT, Bit 0 jusqu'à Bit 15). Les types de données pris en charge sont BYTE, INT, DINT, UDINT, UINT, WORD, DWORD. Si Bit X est sélectionné, la Limite de l'alarme de cette alarme peut être soit 0, soit 1. Par défaut, la limite de l'alarme 1 est automatiquement ajoutée à la colonne Limite de l'alarme.
Limite de l'alarme	Définit la valeur active de l'alarme	Toute valeur du même type de données que la variable (0 ou 1 pour les variables booléennes ou toute chaîne si Expression libre est choisie comme Opérateur d'alarme).	Valeur à comparer avec la valeur de la variable. Avec Vijeo Citect et si vous choisissez Expression libre comme Opérateur d'alarme : l'expression peut contenir des chaînes de substitution pour les noms de variables (par exemple, %var1% pour une variable nommée var1), qui seront remplacés par les noms définitifs des tags au cours de la génération.

Propriété	Description	Options	Commentaire
Texte d'alarme	Message destiné à l'opérateur si une alarme survient	STRING	Champ texte libre Si aucun texte n'est défini, le texte est automatiquement établi. La valeur par défaut est générée par concaténation de la désignation de la variable, du module de commande et de l'équipement.
Priorité d'alarme	Définit la priorité de l'alarme	Sélection dans la liste personnalisée	Les priorités de l'alarme sont définies au moment de la personnalisation.
Mode temporisation d'alarme	définit le mode temporisation d'alarme :	• Pas de temporisation • Variable • Expression libre	uniquement pour les variables booléennes API_IHM ou IHM afin de créer des alarmes horodatées pour Vijeo Citect.
Temporisation d'alarme	variable fournissant le temps d'API (doit avoir le même SCoD que la variable d'alarme)	Variable	uniquement pour les variables booléennes API_IHM afin de créer des alarmes horodatées pour Vijeo Citect. Les types de données pris en charge sont TOD, DINT, UDINT ou DWORD.
	La chaîne peut renvoyer à des variables appartenant au même objet parent que la variable d'alarme en plaçant le nom de la variable entre les caractères %, par ex. %Var1%.	Expression libre	La chaîne ne sera pas validée dans l'éditeur SCoD ni dans UAG et la Temporisation d'alarme sera générée sous forme de Cicode dans Vijeo Citect.

Propriétés relatives à la commande

Ces propriétés sont disponibles pour des variables de type de données élémentaires (sauf le type STRING)

Propriété	Description	Options	Commentaire
Commande	Type de commande de la variable	A sélectionner dans la liste • Visualisation seulement • Opérateur • Logique • Constante • Paramètre • Non affecté(e)	Pour les variables de SCoD, le type de commande est prédéfini : • Les variables reliées à une entrée sont Opérateur ou Paramètre. • Les variables reliées à une sortie sont Visualisation seulement. • Les variables reliées à une entrée/sortie peuvent être quelconques à l'exception de Visualisation seulement.
Droit d'accès	Niveau d'accès de l'opérateur	A sélectionner dans la liste	Les droits d'accès sont définis au moment de la personnalisation.
Texte état 0	Texte à afficher sur l'IHM si la variable est 0	• STRING (chaîne) de 9 caractères max • Non affecté(e)	Seulement pour les variables booléennes API_IHM
Texte état 1	Texte à afficher sur l'IHM si la variable est 1	• STRING (chaîne) de 9 caractères max • Non affecté(e)	Seulement pour les variables booléennes API_IHM

Options de Commande

Option	Modification par l'opérateur ?	Commentaire
Visualisation seulement	Non	La valeur de la variable est affichée sur l'IHM, mais l'opérateur ne peut pas la modifier.
Logique	Non	La valeur de la variable est affichée sur l'IHM mais l'opérateur ne peut pas la modifier (le code généré sera le même que pour Visualisation seulement, mais choisir cet attribut indique à l'automaticien d'ajouter de la logique).
Opérateur	Oui	La valeur de la variable est affichée sur l'IHM et l'opérateur peut la modifier.
Paramètre	Oui	La valeur de la variable est affichée sur l'IHM et l'opérateur peut en modifier la valeur initiale. La transmission de données entre l'API et l'IHM est plus lente que pour l'option Opérateur.
Constante	Non	La méthode de paramétrage de la valeur est fonction du type de variable : ● S'il s'agit d'une variable SCoD, l'entrée du BF du SCoD est reliée à la valeur indiquée en valeur initiale. ● S'il s'agit d'une variable libre, la valeur initiale est paramétrée dans Unity Pro.

Propriétés relatives à l'affichage

Ces propriétés ne sont disponibles que pour les valeurs analogiques (comme INT, REAL, etc.).

Propriété	Description	Options	Commentaire
Format d'affichage	Format du système IHM pour afficher les valeurs analogiques	● 999.9 ● 99.99 ● 9.999…	Valeurs analogiques seulement
Groupe d'unités (Groupe d'unités de mesure)	Groupe d'unités physiques	● A sélectionner dans la liste ● Non affecté(e)	Variables analogiques seulement Les groupes sont définis au moment de la personnalisation. Pour connaître les manières de définir le groupe d'unités et l'unité, reportez-vous à la rubrique *Groupes d'unités de mesure dans le manuel de l'éditeur SCoD*.

Propriété	Description	Options	Commentaire
Unité (Unité de mesure)	Unité physique de la variable	• A sélectionner dans la liste • Non affecté(e)	Variables analogiques seulement Les unités sont définies au moment de la personnalisation.
Brut min. Brut max.	Limite inférieure/supérieure de la valeur non mise à l'échelle provenant du système API Les valeurs Brut min/max et Echelle min/max permettent de calculer la valeur réelle.	La valeur doit correspondre au Type de données des variables. La valeur initiale doit être comprise entre les valeurs brutes.	Non utilisé pour iFIX et Monitor Pro
Echelle min. Echelle max.	Plage de valeurs de la variable dans le système IHM	Il n'est pas nécessaire que la valeur corresponde au type de données de la variable. EchelleMax < EchelleMin est également possible pour une plus grande flexibilité. Si aucune valeur brute n'est définie, la valeur initiale doit alors être comprise entre les valeurs d'échelle.	Disponible à l'affichage pour l'IHM
Saisie min. Saisie max.	Les bornes de saisie de l'opérateur consistent en une valeur minimum et une maximum.	Il n'est pas nécessaire que la valeur corresponde au type de données de la variable. Les valeurs de limites de saisie doivent être comprises dans les valeurs d'échelle.	non utilisé pour Monitor Pro et Vijeo Citect

Définition d’alarme dans UAG

Présentation

Vous pouvez définir des alarmes pour une variable à partir de l’onglet **Alarme** de la boîte de dialogue **Propriétés de variable**.

- Pour des variables libres, vous pouvez ajouter, modifier ou supprimer des alarmes. Pour les alarmes des variables libres, vous pouvez également utiliser les boutons **Monter/Descendre**.
- Les alarmes des variables de type de module de commande peuvent être uniquement modifiées.

Type de connexion

Le type de connexion de la variable est `PLC_HMI` ou `HMI`.

Nombre maximal d’alarmes

Le nombre d’alarmes dépend du type de données de la variable.

Le nombre maximal d’alarmes (incluant le type de valeur et les alarmes concernant les bits) autorisé pour chaque variable est de 2^ (décompte des octets du type de données) *8.

Par exemple, le nombre maximal d’alarmes pour le type de données `INT` est :

2^ 2 *8 = 65536

Mises à jour de versions UAG antérieures

Dans les projets UAG antérieurs à la V3.1SP1, jusqu’à 8 alarmes pouvaient être définies pour chaque variable analogique et des espaces (des alarmes en rien définies) étaient autorisés entre les définitions d’alarme.

Par exemple, les 2ème, 5ème et 7ème alarmes pouvaient être définies, les autres alarmes restant non définies.

Dans UAG V3.1SP1 ou ultérieure, plus de 8 alarmes peuvent être définies pour chaque variable analogique mais aucun espace entre elles n’est possible.

Si un projet antérieur, contenant des alarmes non définies avec des numéros d’alarmes inférieurs au dernier chiffre d’alarme utilisé d’une variable, est mis à jour vers la V3.1SP1 ou ultérieure, ces espaces sont supprimés automatiquement. Ainsi, certains numéros d’alarmes sont parfois modifiés.

Par exemple, les alarmes antérieures portant les numéros 2, 5 et 7 sont converties en alarmes avec les numéros 1, 2 et 3.

Si votre application ne dépend pas des numéros d’alarmes antérieurs, aucune action n’est nécessaire.

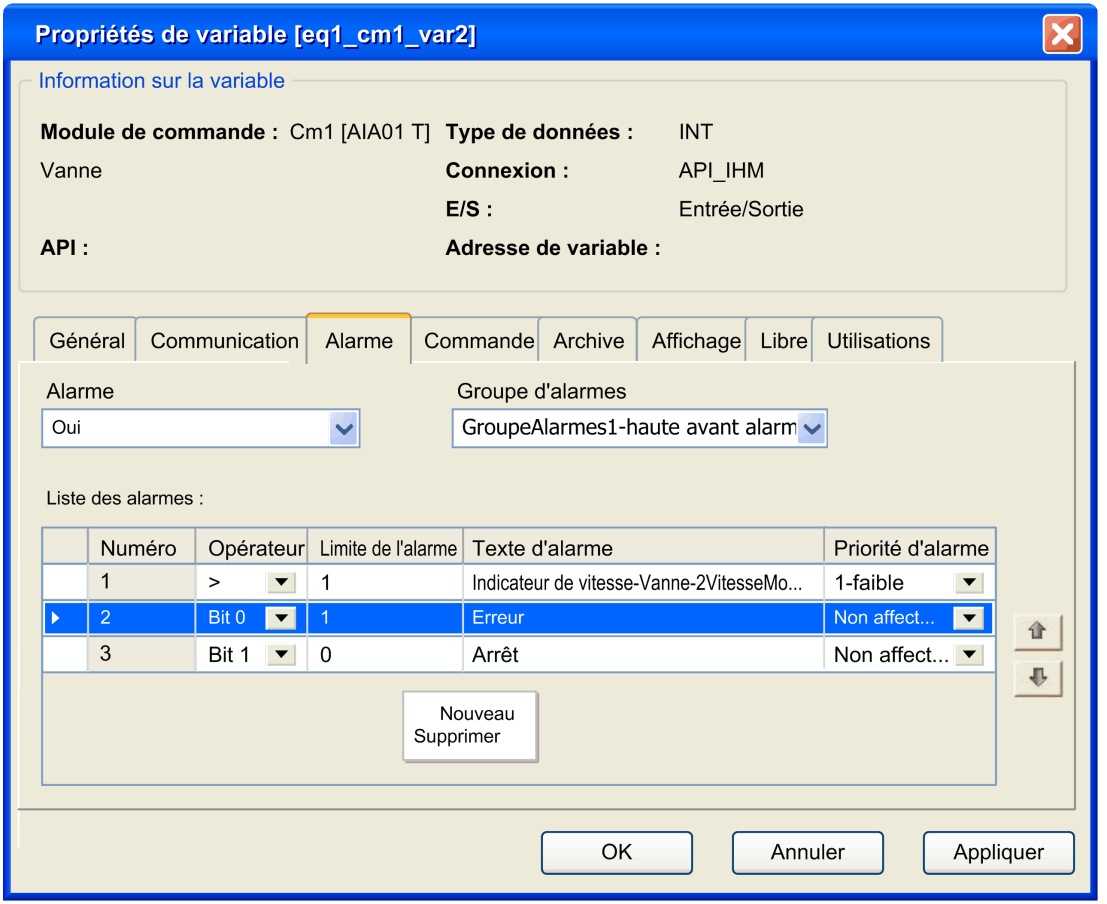
⚠ AVERTISSEMENT

FONCTIONNEMENT IMPREVU DE L'EQUIPEMENT

Si l'application s'appuie sur les anciens numéros d'alarmes, vous devez l'adapter pour qu'elle utilise les nouveaux numéros d'alarme. Si ce n'est pas le cas, de fausses alarmes peuvent apparaître lors de l'exécution de l'application.

Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

Onglet Alarme des Propriétés de variable (non-bool)



Propriété	Description
Numéro de l'alarme	numérotage consécutif des alarmes définies (numéro automatique)
Opérateur	Il existe 3 différents types de valeurs possibles : ● Opérateur de comparaison pour comparer la valeur de la variable à la **Limite de l'alarme** correspondante. Les opérateurs possibles sont **=**, **<>**, **>**, **<**, **>=**, **<=** et ● **Expression libre** Permet d'entrer des chaînes arbitraires que le générateur de l'IHM utilisera. Par exemple, utilisez **Expression libre** pour saisir une expression Cicode pour une alarme. ● **Bit X** Sélectionne le bit de la valeur de variable à utiliser pour des définitions d'alarmes au niveau du bit (Bit 0 ... Bit n-1). Le nombre de valeurs de bits dépend du type de données de la variable (par ex. pour `INT` **, Bit 0** jusqu'à **Bit 15**). Si vous sélectionnez **Bit X**, la **Limite de l'alarme** de cette alarme peut être soit **0**, soit **1**. Par défaut, la limite de l'alarme **1** est automatiquement ajoutée à la colonne **Limite de l'alarme**. Vous pouvez également associer des alarmes au niveau du bit en utilisant des variables du champ d'élément binaire. Veuillez consulter le *Manuel utilisateur avancé*.
Limite de l'alarme	Définit la valeur active de l'alarme.
Texte d'alarme	Affiche le texte par défaut de l'alarme défini pour le projet dans l'éditeur de personnalisation.
Priorité d'alarme	Définit la priorité de l'alarme.
Mode temporisation d'alarme	Définit le **mode temporisation d'alarme** : ● **Pas de temporisation** ● **Variable** ● **Expression libre** Cette propriété est utilisée uniquement pour créer des alarmes booléennes horodatées pour Vijeo Citect.

Propriété	Description
Temporisation d'alarme	Dépend du **mode temporisation d'alarme**. ● Si **Variable** est sélectionné : Sélectionnez le type de connexion `API_IHM` pour la variable fournissant le temps de l'API. La variable peut avoir le type de données suivant : TOD, DINT, UDINT ou DWORD. La variable peut être saisie directement ou elle peut être sélectionnée à partir d'une fenêtre d'affichage des variables dont le domaine est réglé sur le parent de la variable de temporisation, si cette dernière est déjà affectée ; sinon, sur le parent de la variable d'alarme. Le domaine de la variable de temporisation n'a pas de restriction dans le projet, mais il est limité au type de modèle identique figurant la bibliothèque des types de modèle. La variable peut être utilisée pour plusieurs alarmes sous le même SCoD. ● Si **Expression libre** est sélectionné : Vous pouvez saisir une chaîne qui peut renvoyer à des variables appartenant au même objet parent que la variable d'alarme en plaçant le nom de la variable entre les caractères `%`, par ex. `%Var1%`. Si la chaîne saisie comprend des manipulations de bits, UAG la génère en tant que cicode dans Vijeo Citect. La chaîne saisie ne sera pas validée dans UAG et la **Temporisation d'alarme** sera générée sous forme de Cicode dans Vijeo Citect. Cette propriété est utilisée uniquement pour créer des alarmes booléennes horodatées pour Vijeo Citect.

NOTE : Mode temporisation d'alarme passera de **Variable** à **Pas de temporisation** et la valeur **Temporisation d'alarme** sera effacée si la variable de temporisation est supprimée.

Si une variable possédant une alarme horodatée est copiée d'un SCoD à un autre, les deux propriétés **Mode temporisation d'alarme** et **Temporisation d'alarme** seront copiées vers la variable cible.

Si l'expression libre de la variable à alarme horodatée est attribuée à %sw%, UAG génère une variable sans validation pour Vijeo Citect et la traite comme Cicode.

L'éditeur SCoD interdit la modification du type de données et du type de connexion d'une variable **Temporisation d'alarme** tant que celle-ci est utilisée à titre de variable de temporisation.

Par défaut, aucune alarme ne s'affiche dans la **Liste des alarmes**. Pour ajouter ou supprimer des alarmes, cliquez avec le bouton droit de la souris sur la **Liste des alarmes** et sélectionnez l'élément correspondant dans le menu contextuel.

Texte d'alarme par défaut/libre

Pour modifier le texte d'une alarme, placez le curseur sur la cellule de texte de l'alarme en question puis cliquez sur le bouton de navigation (**...**).

Dans la boîte de dialogue **Texte d'alarme** qui s'affiche, le mot-clé **$** représente le texte d'alarme par défaut défini pour le projet dans l'éditeur de personnalisation.

Vous pouvez ajouter un texte libre en plus du texte d'alarme par défaut.

Le texte libre peut être apposé et/ou préfixé au texte d'alarme par défaut.

Pour n'utiliser que du texte libre, supprimez le mot-clé **$**.

Pour restaurer le texte d'alarme par défaut, cliquez sur le bouton **Par défaut**.

La fenêtre **Aperçu** affiche le texte d'alarme concaténé.

Nouvelle alarme

Cliquez avec le bouton droit de la souris sur la **Liste des alarmes** puis sélectionnez **Nouveau** dans le menu contextuel.

La boîte de dialogue qui vous permet de configurer les détails de la nouvelle alarme s'affiche.

Vous ne pouvez pas créer de nouvelle alarme si l'une des alarmes déjà créées est vide (c.-à-d. qu'une des alarmes n'est pas configurée avec au moins une propriété).

Les nouvelles alarmes sont toujours ajoutées à la fin de la liste des alarmes.

Nouveau sélectionne par défaut l'opérateur suivant libre dans la colonne **Opérateur**.

Par exemple, si vous configurez l'alarme **Bit 1** puis sélectionnez **Nouveau** dans le menu contextuel, **Bit 2** s'affiche comme la valeur suivante dans la colonne **Opérateur**.

Suppression d'une alarme

Cliquez avec le bouton droit de la souris sur une alarme de la **Liste des alarmes** puis sélectionnez **Supprimer** dans le menu contextuel pour supprimer l'alarme de la liste.

Vous pouvez également sélectionner et supprimer plusieurs alarmes simultanément.

Il vous est demandé de confirmer l'opération de suppression.

Au cas où vous supprimeriez une alarme située au milieu de la liste, les numéros des alarmes sont réattribués aux alarmes restantes dans la liste.

Les numéros des alarmes sont conservés, allant de 1 jusqu'au nombre d'alarmes total sans aucune interruption.

Bouton Appliquer

Utilisez le bouton **Appliquer** pour enregistrer les modifications apportées dans la boîte de dialogue **Propriétés de variable**.

Les restrictions suivantes existent :

- Nouvelle alarme
  Les alarmes qui n'ont pas été modifiées dans les propriétés d'alarme ne sont pas stockées.
- Mise à jour d'une alarme
  Si les Propriétés sont réinitialisées, cette alarme est alors supprimée après confirmation par l'utilisateur.
- Suppression d'une alarme
  Si une alarme est supprimée mais que cette alarme fait partie d'un héritage ou d'une propriété d'un périphérique, ces liens sont également supprimés.
- Il vous est demandé de confirmer la suppression des alarmes.

Boutons Descendre/Monter

Ces 2 boutons sont uniquement activés si une alarme est sélectionnée dans la **Liste des alarmes**.

Le fait de déplacer une alarme ne fait qu'échanger les numéros des alarmes (mais pas le nombre des alarmes).

Par exemple, si vous faites descendre une ligne sélectionnée, les valeurs des 4 propriétés de l'alarme sélectionnée (**Opérateur**, **Limite de l'alarme**, **Texte d'alarme** et **Priorité d'alarme**) deviennent les valeurs de l'alarme du dessous et vice versa.

Le mécanisme de défilement automatique est actif lors des opération de descente/montée si vous avez défini plus d'alarmes que la fenêtre n'est capable d'en afficher.

Vous ne pouvez pas utiliser le glisser-déposer dans la **Liste des alarmes**.

Lorsque vous descendez/montez une alarme, les héritages ou les propriétés SCoD associées à cette alarme ne bougent pas.

La boîte de dialogue de confirmation suivante s'affiche si l'alarme fait état d'héritages ou de propriétés SCoD :

Veuillez vraiment déplacer l'alarme ?

Les propriétés SCoD et les attributs hérités associés à cette alarme ne sont pas mis à jour et doivent être modifiés manuellement.

Tri

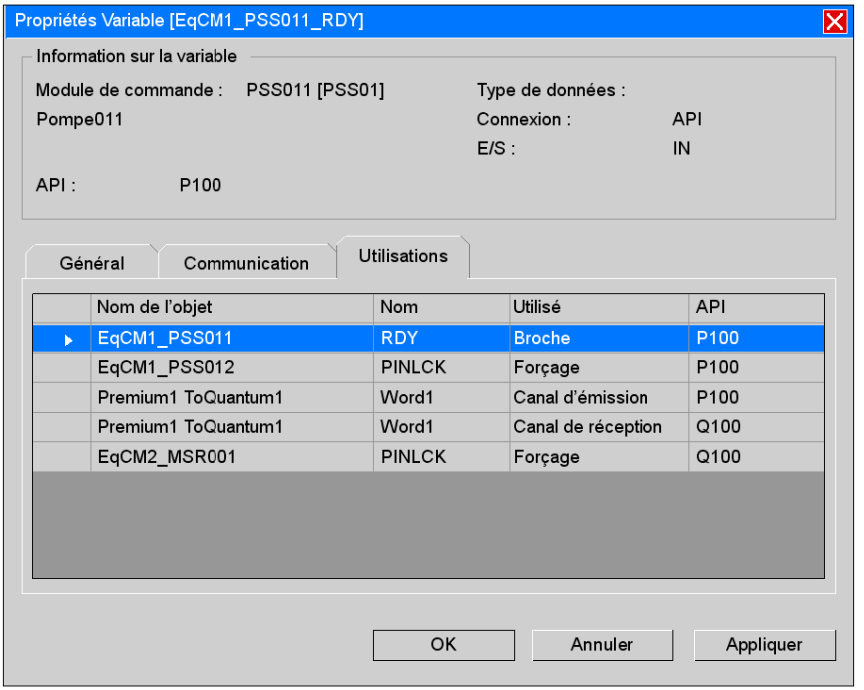
Vous pouvez trier les alarmes dans les colonnes de la **Liste des alarmes**.

Onglet Utilisations des propriétés de variables

Présentation

L'onglet **Utilisations** de la boîte de dialogue **Propriétés de variable** donne des renseignements sur toutes les utilisations d'une variable.

Représentation



L'onglet **Utilisations** répertorie chaque occurrence de la variable sur une ligne distincte, en donnant les informations suivantes :

Elément	Description
Nom de l'objet	Objet dans lequel la variable est utilisée
Nom	Sous-élément d'objet dans lequel la variable est utilisée
Utilisé	Type d'utilisation de la variable
API	API dans lequel la variable est utilisée

Exemple sur la ligne 3 :

La variable `EqCM1_PSS011_RDY` est utilisée pour le canal `Premium1ToQuantum1`. Le numéro de mot pour cette variable dans le tableau de communication API est `1`.

Informations d'utilisation

Combinaisons possibles des informations d'utilisation

Nom de l'objet	Nom	Utilisé	API	Remarque
Module de commande	Broche connectée	Broche	Nom de l'API s'il est affecté	Variable reliée à une broche d'un bloc fonction SCoD affecté au module de commande
Module de commande	Broche connectée	Liaison	Nom de l'API s'il est affecté	Variable connectée à une liaison
Module de commande	Broche connectée	Forçage	Nom de l'API s'il est affecté	Variable connectée à un forçage
Nom de canal	Numéro de mot	Canal d'émission	Nom de l'API	Variable déposée sur le canal d'émission
Nom de canal	Numéro de mot	Canal de réception	Nom de l'API	Variable apparaissant sur le canal de réception
Nom de canal	Nom de la trame de communication	Canal de serveur de données	Nom de l'API	Variable apparaissant sur le canal API vers serveur de données
Nom de canal	Adresse	Canal de partenaire réseau	Nom de l'API	Variable apparaissant sur le canal API vers partenaire réseau
Nom de châssis	Numéro de châssis	Point d'E/S	Nom de l'API	Variable déposée sur le module d'E/S

Emplacement de l'utilisation

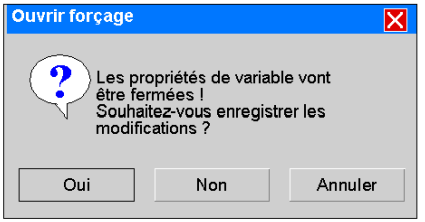
Pour accéder à l'emplacement exact d'utilisation des variables, double-cliquez sur une ligne de la grille, ou cliquez sur une ligne et sélectionnez l'option **Ouvrir...** dans le menu contextuel.

Le texte de l'option du menu contextuel dépend de l'utilisation de la variable.

Utilisé	Entrée	Remarque
Broche	Ouvrir bloc fonction	Lance l'éditeur SCoD, ouvre la vue des blocs fonctions (FFB) pour le module de commande indiqué dans la colonne **Nom de l'objet** et sélectionne la broche à laquelle la variable est connectée.
Liaison	Ouvrir liaison	Ouvre la boîte de dialogue de l'éditeur de liaisons.
Forçage	Ouvrir forçage	Ouvre l'éditeur graphique de forçages.

Utilisé	Entrée	Remarque
Canal d'émission	Ouvrir canal	Ouvre le tableau de communication d'API et sélectionne la ligne qui contient la variable.
Canal de réception	Ouvrir canal	Ouvre le tableau de communication d'API et sélectionne la ligne qui contient la variable.
Canal de serveur de données	Ouvrir canal	Ouvre le tableau de communication d'IHM et sélectionne la trame de communication à laquelle la variable est affectée.
Canal de partenaire réseau	Ouvrir canal	Ouvre le tableau de communication de réseau et sélectionne la ligne qui contient cette variable.
Point d'E/S	Ouvrir points d'E/S	Ouvre la liste des points d'E/S et sélectionne la ligne qui contient la variable.

Si vous tenter d'accéder à l'emplacement d'utilisation des variables, la boîte de dialogue suivante s'affiche :



Bouton	Description
Oui	Enregistre toutes les modifications de la boîte de dialogue **Propriétés** de la variable, ferme la boîte de dialogue et ouvre l'emplacement d'utilisation des variables
Non	Ferme la boîte de dialogue **Propriétés** de la variable sans enregistrer les modifications et ouvre l'emplacement d'utilisation des variables
Annuler	N'ouvre pas l'emplacement d'utilisation des variables, mais la boîte de dialogue **Propriétés** de la variable reste affichée

Chapitre 5
Modèle topologique

Introduction

Le modèle topologique décrit la topologie de l'automatisme. Il sert à l'ingénieur automaticien pour définir l'architecture du système automatisé. Ce chapitre traite du modèle topologique général géré par Unity Application Generator. De plus, il détaille les éléments utilisés dans Unity Application Generator pour définir ce modèle.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
5.1	Topologie d'un automatisme et modèle topologique	118
5.2	Les groupes segments de réseau et chemins de routage	142
5.3	Le groupe API	148
5.4	Groupe IHM	214
5.5	Le groupe Serveur de données	221
5.6	Le groupe Abonnés réseau	237

Sous-chapitre 5.1
Topologie d’un automatisme et modèle topologique

Introduction

Vous apprendrez dans cette partie comment la topologie d’un automatisme type est représentée dans le modèle topologique.

Vous découvrirez en outre les éléments qui sont gérés par Unity Application Generator.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Topologie d’un automatisme	119
Structure du modèle topologique	124
Communication via Modbus, Modbus Plus et Ethernet	126
Autres informations relatives à Ethernet	129
Configuration de la redondance d’UC Quantum	132
Configuration de redondance d’UC Premium	135

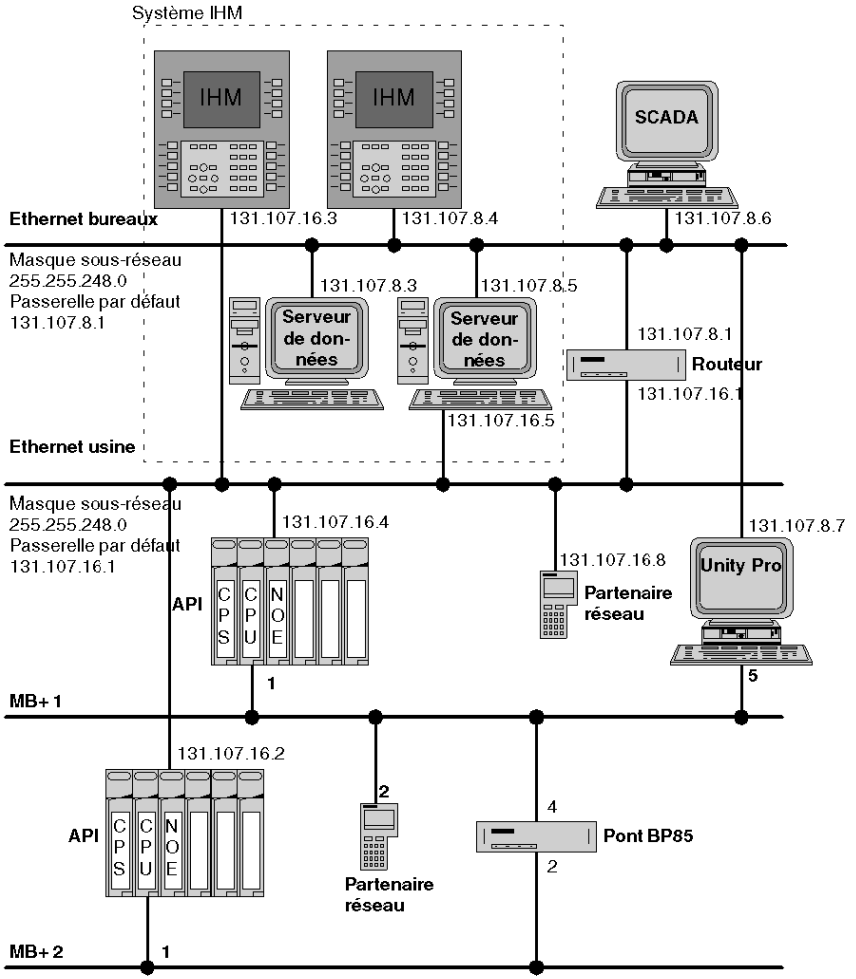
Topologie d’un automatisme

Introduction

Unity Application Generator gère avec souplesse des topologies complexes d’automatisme. Ces topologies sont constituées d’éléments typiques de tous les automatismes. Le schéma suivant donne un aperçu des éléments de la topologie d’un automatisme gérés par UAG. Vous trouverez la description de ces éléments et de leurs relations dans le modèle topologique de Unity Application Generator.

Exemple d'automatisme type

Le schéma suivant illustre la topologie d'un automatisme type



Eléments gérés

Les éléments suivants sont définis au sein du modèle topologique UAG.

- API Unity/Quantum comportant :
  - E/S déportées et distribuées
  - Systèmes de redondance d'UC
  - E/S Momentum Modbus Plus
  - E/S Momentum Ethernet

- E/S Advantys Modbus Plus
- E/S Advantys Ethernet
- E/S Profibus
- Passerelle Modbus/Ethernet *(voir page 182)*
- Passerelle Profibus/Ethernet *(voir page 188)*
- Modules matériels définis par l'utilisateur (Manuel utilisateur avancé de UAG, Modules définis par l'utilisateur – Présentation)

NOTE : Pour les modules **140-CPU-651-60S** (autonomes), **140-CPU-671-60S** (HSBY), **140-SAI-940-00S**, **140-SDI-953-00S** et **140-SDO-953-00S**, UAG prend uniquement en charge la configuration ; la logique ne sera pas générée.

- API Unity/Premium comportant :
  - Systèmes de redondance d'UC
  - E/S Momentum Modbus Plus
  - E/S Momentum Ethernet
  - E/S Advantys Modbus Plus
  - E/S Advantys Ethernet
  - E/S Profibus
  - E/S XBus
  - Passerelle Modbus/Ethernet *(voir page 182)*
  - Passerelle Profibus/Ethernet *(voir page 188)*
  - Modules matériels définis par l'utilisateur (Manuel utilisateur avancé de UAG, Modules définis par l'utilisateur – Présentation)
- API Unity/M340 comportant :
  - E/S Momentum Ethernet
  - E/S Advantys Ethernet
  - Passerelle Modbus/Ethernet *(voir page 182)*
  - Passerelle Profibus/Ethernet *(voir page 188)*
  - Modules matériels définis par l'utilisateur (Manuel utilisateur avancé de UAG, Modules définis par l'utilisateur – Présentation)
- API Unity/M580 comportant :
  - Modules matériels définis par l'utilisateur (Manuel utilisateur avancé de UAG, Modules définis par l'utilisateur – Présentation)
- Serveurs de données
- IHM (Vijeo Citect, ArchestrA, Monitor Pro, IHM génériques et spécifiques d'un client)
- Abonnés réseau (Partenaires réseau et Autres abonnés)
- Routeurs
- Ponts
- Segments de réseau de type Modbus Plus ou Ethernet

Réseau

UAG prend en charge les réseaux Modbus, Modbus Plus et Ethernet.

Les fonctions de communication des types de réseaux sont différentes.

- Pour les IHM, le serveur de données peut être connecté via Modbus Plus ou Ethernet.
- Les abonnés réseau peuvent se connecter via Modbus Plus ou Ethernet.
- Les API peuvent se connecter via Modbus Plus ou Ethernet.
- Possibilité de configurer des ponts Modbus Plus.
  Les routeurs Ethernet ne sont pas configurés explicitement dans UAG, mais une passerelle existante par défaut dans la définition du segment de réseau est censée représenter un routeur.
- Il est possible de configurer les passerelles Modbus/Ethernet *(voir page 182)*
- Il est possible de configurer les passerelles Profibus/Ethernet *(voir page 188)*

Quantum

Tous ces modules peuvent servir à la communication entre divers API et vers des serveurs de données ou vers des partenaires réseau. Mais seuls les modules UC/NOE répertoriés peuvent servir dans l'API émetteur à une communication API <-> API via Ethernet.

- 140-NOE-771-00
- 140-NOE-771-01
- 140-NOE-771-11

Les autres NOE peuvent uniquement servir dans l'API récepteur.

Momentum

Tous les modules processeur Ethernet peuvent servir à la communication entre divers API et vers des serveurs de données ou vers des partenaires réseau sans restriction.

Premium

Tous ces modules peuvent servir à la communication entre divers API et vers des serveurs de données ou vers des partenaires réseau. Mais seuls les modules Ethernet répertoriés peuvent servir dans l'API émetteur à une communication API <-> API par Ethernet.

- TSX ETY4103
- TSX ETY5103
- TSX P57 5634M

Les autres modules prenant en charge Ethernet ne peuvent servir que dans l'API récepteur.

M340

Tous ces modules peuvent servir à la communication entre divers API et vers des serveurs de données ou vers des partenaires réseau. Mais seuls les modules Ethernet BMX NOE 0100 et BMX NOE 0110 peuvent servir dans l'API émetteur à une communication API <-> API via Ethernet.

Les autres modules prenant en charge Ethernet ne peuvent servir que dans l'API récepteur.

M580

Les UC et le module NOC (BMX NOC 0402) peuvent être utilisés pour une communication sur Serveur de données.

La communication API <-> API n'est pas prise en charge pour un API M580 sur UAG.

Redondance avec Unity Application Generator

Unity Application Generator gère la configuration redondante de plusieurs parties de la topologie de commande :

- Possibilité de définir plusieurs segments de réseau.
- Pour la communication entre divers API Quantum et Momentum, il est possible de définir des chemins d'accès redondants sur le réseau.
- Les API Unity Pro/Quantum peuvent être définis comme redondants à l'aide du module 140 CPU 671 60, voir Hot Standby Configuration *(voir page 132)*.
- Les API Unity Pro / Premium peuvent être définis comme redondants à l'aide des modules TSX H57 24M et TSX H57 44M.
- Serveurs de données à carte réseau redondante
- Les serveurs de données peuvent être configurés en redondance pour Vijeo Citect. Concernant d'autres systèmes SCADA, cette configuration, qui n'est pas définie dans UAG, doit l'être dans le système SCADA.
- Pour la communication entre divers API Quantum, Premium, M340, M580 et Momentum et des serveurs de données, il est possible de définir des chemins d'accès redondants sur le réseau.

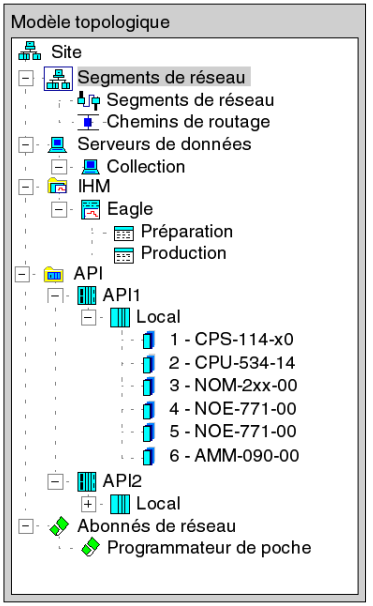
Structure du modèle topologique

Introduction

Au sein du modèle topologique, l'ingénieur automaticien définit l'architecture du système d'automatisme. Ce faisant, il définit les ressources utilisées par les éléments du modèle physique.

A quoi cela ressemble-t-il ?

Le modèle topologique avec quelques éléments définis par l'utilisateur.



Vue d'ensemble des éléments

Le modèle topologique contient les groupes suivants :

- Le groupe Segments de réseau
- Le groupe Serveur de données
- Le groupe IHM
- Le groupe API
- Le groupe Abonnés de réseau

Groupe API

Plusieurs automates sont définis dans le groupe API.

- L'intégralité de la configuration en châssis est établie pour chaque API.
- Chaque châssis est lui-même entièrement configuré avec ses modules.
- Les adresses réseau sont définies pour les modules de communication.
- Les canaux de communication sont définis pour chaque API. Ces canaux définissent les données qui sont échangées entre les objets respectifs de l'automatisme.
- Chemins de routage,
  qui définissent les connexions entre différents segments de réseau par des ponts ou des routeurs.

Groupe IHM

On définit dans le groupe IHM plusieurs applications IHM.

- Chaque IHM peut être définie avec plusieurs zones de conduite, groupes d'images et images.

Groupe Serveur de données

Le groupe Serveur de données définit les serveurs de données, ainsi que leurs connexions au réseau (type de réseau et adresses).

Groupe Abonnés de réseau

On définit dans les abonnés réseau les partenaires réseau et/ou les autres abonnés, leur connexion au réseau (type de réseau et adresse), ainsi que leur icône dans l'afficheur topologique.

Divers

Le groupe Réseau contient l'intégralité de la structure de communication.

Les éléments constituant le groupe Réseau sont les suivants :

- les segments de réseau,
  qui définissent les noms et types de segments de réseau.
- Chaque module E/S définit les points d'E/S.

Communication via Modbus, Modbus Plus et Ethernet

Que sont les segments de réseau ?

Un segment de réseau est une simple ligne de réseau d'un type défini (Modbus Plus, par exemple) disposant de toute la plage d'adresses. Les abonnés réseau tels que les API, IHM, serveurs de données, partenaires réseau et autres abonnés, sont connectés à un ou plusieurs segments de réseau.

Types de segments gérés

Unity Application Generator gère les réseaux de type

- Modbus (via passerelle Modbus/Ethernet, voir ci-après)
- Modbus Plus
- Ethernet
- Profibus (via passerelle Profibus/Ethernet, voir ci-après)

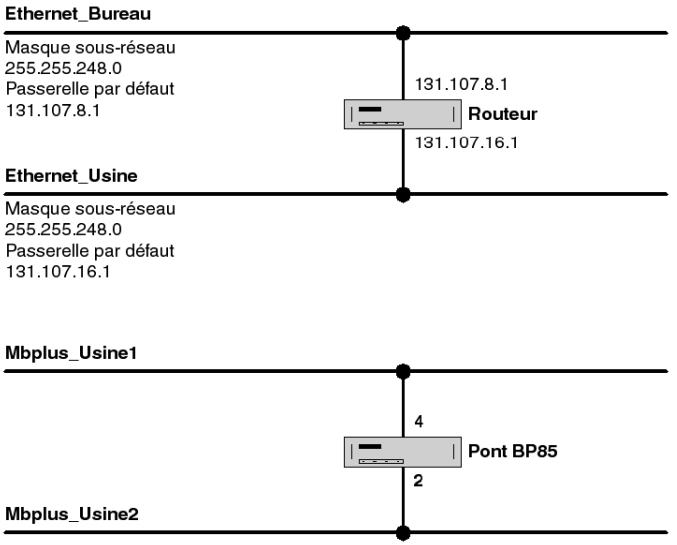
Routeurs et ponts

En général, les routeurs et les ponts connectent différents segments de réseau et différents types de réseaux tels que Modbus Plus et Ethernet.

Dans Unity Application Generator, il est possible de définir des ponts entre les segments Modbus Plus.

Unity Application Generator ne fait pas la distinction entre un routeur ou un pont. Du point de vue de Unity Application Generator, les deux effectuent la même tâche ; ils connectent différents segments de réseau.

Le schéma ci-dessous illustre un routeur Ethernet et un pont Modbus Plus.



Passerelle Modbus/Ethernet

Une passerelle Modbus-Ethernet permet l'accès à un réseau série Modbus au moyen du protocole Modbus/TCP. Dans UAG, une passerelle apparaît sous la forme d'un châssis. Les équipements Modbus sont connectés à ce châssis virtuel.

Pour de plus amples informations, consultez la section Passerelle Modbus/Ethernet *(voir page 182)*.

Passerelle Profibus/Ethernet

Une passerelle Profibus-Ethernet permet l'accès à un réseau Profibus au moyen du protocole Modbus/TCP. Dans UAG, une passerelle apparaît sous la forme d'un châssis. Les équipements Profibus sont connectés à ce châssis.

Pour de plus amples informations, consultez la section Passerelle Profibus/Ethernet *(voir page 188)*.

Définition d'un routeur entre segments Ethernet

Dans un environnement Windows, l'adresse d'un routeur se définit comme l'adresse de passerelle par défaut.

Il en est de même dans UAG. Pour chaque segment Ethernet, il est possible de définir une adresse de passerelle par défaut ainsi que le masque de sous-réseau dans les propriétés du segment.

Dans UAG, les routeurs ne sont pas illustrés.

Définition d'un pont entre segments Modbus

Les ponts Modbus Plus se définissent séparément des segments de réseau dans les `chemins de routage`. Le chemin de routage définit simplement l'adresse du pont du point de vue des segments. Pour plus d'informations, consultez la section *Description et propriétés des chemins de routage, page 146*

Qu'est-ce qu'un canal ?

Un canal décrit :

- les objets de l'automatisme avec lesquels l'API communique ;
- sur quel chemin de communication aura lieu la communication ;
- un chemin alternatif de communication ;
- les variables devant être échangées.

La communication s'effectue par un API. Cela signifie que l'un des partenaires de communication d'un canal est un API.

Comment définir la communication

La définition du réseau entre les différentes parties de l'automatisme est établie à l'aide de plusieurs objets de Unity Application Generator.

Un élément déterminant dans la conception du réseau est de définir quelles sont les données échangées

- avec quels objets de l'automatisme (les partenaires de communication),
- par quels chemins de communication.

Cela s'effectue généralement en définissant les canaux de l'API.

La définition des canaux s'appuie sur les informations concernant la conception du réseau, définis à partir des éléments réseau du modèle topologique. Ces éléments doivent être définis préalablement, avant de pouvoir définir le canal.

Ces éléments sont les suivants :

- les adresses réseau des modules de communication de l'API, voir *Propriétés du module matériel, page 175*,
- le segment de réseau, voir *Description et propriétés des segments de réseau, page 143*,
- les chemins de routage, voir *Description et propriétés des chemins de routage, page 146*,
- les adresses réseau des serveurs de données, voir *Propriétés du serveur de données, page 223*, et les partenaires réseau, voir *Propriétés d'un partenaire réseau, page 240*.

Autres informations relatives à Ethernet

Exigences relatives à un segment Ethernet

Un segment de réseau Ethernet se définit par :

- Masque sous-réseau
  qui définit la partie de l'adresse IP relative au réseau et celle relative aux abonnés ;
- Passerelle par défaut
  qui définit l'adresse de routage vers d'autres segments de réseau.

Tous les abonnés d'un segment Ethernet possèdent le même masque de sous-réseau et la même passerelle.

Règles d'adressage vérifiées par UAG

Les conditions vérifiées sont les suivantes :

- Le nom du segment de réseau doit être unique.
- Le masque de sous-réseau et la passerelle ont le format d'une adresse IP (4 octets).
- Des segments Ethernet différents peuvent posséder les mêmes masques de sous-réseau, mais pas la même adresse de passerelle par défaut.

NOTE : Le masque de sous-réseau ne garantit pas l'exactitude de l'adresse de l'abonné.

Segments de réseau Ethernet local

En plus des segments faisant partie du réseau usine ou même de l'Internet, il est possible de définir des segments de réseau local. Ces réseaux ne sont pas connectés au réseau usine (via des routeurs). Ils sont uniquement connectés localement à un module Ethernet (NOE, M1 Ethernet, ETY, etc.).

Les réseaux locaux sont définis avec une adresse de passerelle par défaut vide (pas de connexion aux autres réseaux). Dans ce cas, les adresses IP ne doivent pas nécessairement être uniques pour tous les segments, mais seulement sur ce segment local.

La norme IP recommande d'utiliser les plages d'adresses suivantes pour les réseaux privés (locaux) :

- 10.0.0.0 à 10.255.255.254 : classe A
- 172.16.0.0 à 172.31.255.254 : classe B
- 192.168.0.0 à 192.168.255.254 : classe C

Néanmoins, Unity Application Generator n'oblige pas à utiliser ces adresses pour les segments de réseau local.

Abonnés Ethernet

Pour un abonné connecté à Ethernet, seule l'adresse IP doit être définie.

Les conditions vérifiées sont les suivantes :

- Chaque abonné doit être affecté à un segment de réseau.
- Chaque abonné possède une adresse TCP/IP (4 octets N1.N2.N3.N4).
  Plages d'adresses :
  - N1 entre 1 et 126 ou 128 et 223
  - N2 entre 0 et 254
  - N3 entre 0 et 254
  - N4 entre 1 et 254
- L'adresse 1.1.1.1 est interdite.
- L'adresse TCP/IP des abonnés doit être unique sur tous les segments, à l'exception des réseaux locaux, voir ci-dessus.

Modules Ethernet Quantum gérés

Les modules de communication Ethernet Quantum gérés sont les suivants :

- 140-NOE-771-00
- 140-NOE-771-01
- 140-NOE-771-10
- 140-NOE-771-11

NOTE :

Tous ces modules peuvent servir à la communication entre API et vers des serveurs de données ou vers des abonnés réseau. Mais seuls les modules UC/NOE répertoriés peuvent servir dans l'API émetteur à une communication API <-> API via Ethernet.

- 140-NOE-771-00
- 140-NOE-771-01
- 140-NOE-771-11
  Les autres NOE peuvent uniquement servir dans l'API récepteur.

Cartes processeur Ethernet Momentum gérées

Les cartes processeur Ethernet M1 gérées sont les suivantes :

- 171-CCC-960-30-IEC
- 171-CCC-960-91
- 171-CCC-980-30-IEC
- 171-CCC-980-91
  Tous ces modules peuvent servir à la communication entre API et vers des serveurs de données ou vers des abonnés réseau.

Modules Ethernet Premium gérés

Les modules de communication Ethernet Premium gérés sont les suivants :

- TSX ETY4103
- TSX ETY5103
- TSX P57 5634M
- TSX WMY 100

NOTE :

Tous ces modules peuvent servir à la communication entre API et vers des serveurs de données ou vers des abonnés réseau. Mais seuls les modules Ethernet répertoriés peuvent servir dans l'API émetteur à une communication API <-> API par Ethernet.

- TSX ETY4103
- TSX ETY5103
- TSX P57 5634M
- TSX WMY 100

  Les autres modules Ethernet peuvent uniquement servir dans l'API récepteur.

Modules Ethernet M340 gérés

Les modules de communication Ethernet M340 gérés sont les suivants :

- BMX P34 2020
- BMX P34 2030
- BMX NOE 0100

NOTE : Tous ces modules peuvent servir à la communication entre API et vers des serveurs de données ou vers des abonnés réseau. Mais seuls les modules CPU/NOE listés peuvent servir dans l'API émetteur à une communication API <-> API via Ethernet.

Les autres modules Ethernet peuvent uniquement servir dans l'API récepteur.

Modules Ethernet M580 pris en charge

Les modules de communication Ethernet M580 pris en charge sont les suivants :

- BME P58 1020
- BME P58 2020
- BME P58 2040
- BME P58 3020
- BME P58 3040
- BME P58 4020
- BME P58 4040
- BMX NOC 0402

Configuration de la redondance d'UC Quantum

Définition d'un système de redondance d'UC

La redondance d'UC Quantum est conçue pour servir dans les cas où les arrêts ne peuvent pas être tolérés. Le système offre une grande disponibilité grâce à la redondance.

Deux fonds de panier sont configurés avec les mêmes matériels et logiciels.

L'un des API agit en tant qu'automate principal ou primaire. Il exécute l'application en scrutant le programme utilisateur et en agissant sur les E/S déportées.

L'autre API agit en automate redondant. L'automate principal met à jour l'automate redondant après chaque cycle. La redondance est prête à prendre le relais de la commande au bout d'un cycle programme si l'automate principal n'est pas en mesure de répondre.

Il est possible de commuter l'UC principale et l'UC redondante. Tout automate peut être désigné automate principal, mais à la condition que l'autre soit en mode redondance. Le réseau des E/S déportées est piloté par l'automate principal.

NOTE : Une redondance d'UC Quantum ne prend en charge que des E/S déportées. Elle ne gère pas d'E/S locales ou distribuées (DIO) sauf dans la redondance d'UC Unity Pro. Dans Unity Pro, il est permis d'utiliser des E/S locales. Pour assurer l'intégrité des données, il est conseillé de ne pas utiliser d'E/S locales dans un automate en redondance.

Pour de plus amples détails concernant la redondance d'UC Quantum, reportez-vous au guide de planification et d'installation de la redondance d'UC Quantum.

Principe de fonctionnement

La redondance d'UC Quantum est conçue pour servir dans les cas où les arrêts ne peuvent pas être tolérés. Le système offre une grande disponibilité grâce à la redondance. Deux fonds de panier sont configurés avec les mêmes matériels et logiciels. Utilisez l'automate 140 CPU 671 60 pour la redondance d'UC avec Unity Pro. Le module surveille son propre automate et communique avec l'autre module de redondance d'UC. Le système se surveille en permanence.

Automate principal et redondant

L'un des API agit en tant qu'automate principal ou primaire. Il exécute l'application en scrutant la logique et en agissant sur les E/S déportées.

L'autre API agit en automate redondant. L'automate principal met à jour l'automate redondant après chaque cycle. La redondance est prête à prendre le relais de la commande au bout d'un cycle programme si l'automate principal n'est pas en mesure de répondre.

Redondance d'UC et Unity Application Generator

Au sein de Unity Application Generator, une redondance d'UC se définit au niveau du groupe API. Dès qu'un module 140 CPU 671 60 est placé dans un châssis local d'un API Unity Pro, cet API est automatiquement considéré comme une redondance d'UC. Pour plus d'informations sur le placement des modules, consultez *Le groupe API, page 148*.

Restrictions

Restrictions générales d'une redondance d'UC :

- une redondance d'UC Quantum ne prend en charge que des E/S déportées. Elle ne gère pas d'E/S locales ou distribuées (DIO). Sauf dans la redondance d'UC Unity Pro. Dans Unity Pro, il est permis d'utiliser des E/S locales. Pour assurer l'intégrité des données, il est conseillé de ne pas utiliser d'E/S locales dans un automate en redondance.
- Unity Pro
  Seul l'automate 140 CPU 671 60 est pris en charge pour la redondance d'UC.

NOTE : Dès qu'un module 140 CPU 671 60 (Unity Pro) est placé dans un API, Unity Application Generator applique ces restrictions.

Configuration minimale d'un API redondant

La configuration minimale pour Unity Pro est constituée des modules suivants :

- Fond de panier
- Module d'alimentation
- Module de redondance d'UC (140 CPU 671 60)
- Au moins une station RIO

Redondance d'UC et communication

Même si un système de redondance d'UC est constitué de deux API, du point de vue communication, il fonctionne comme un seul API. Néanmoins les deux API sont connectés au réseau. Pour cela, une adresse spéciale sur le réseau est automatiquement affectée à l'API redondant. Ces adresses ne peuvent pas être utilisées pour d'autres participants au réseau. L'adresse en redondance n'est pas réservée pour le module de réseau Advantys connecté à l'API HSBY.

Adresse automatiquement affectée pour Modbus Plus

- Adresse de l'API principal = n
- Adresse de l'API en redondance = n+32
  Si n+32>64, alors l'adresse de l'API en redondance d'UC = n -32

Adresse automatiquement affectée pour Ethernet

- Adresse de l'API principal = x.y.z.n
- Adresse de l'API redondant = x.y.z.n+1
  Si n+1 = 255, alors l'adresse de l'API redondant = x.y.z.1

Affectation de registres

Dans Unity Pro, le mot système %SW60 est utilisé comme registre de commande, les mots %SW62 et %SW63 étant utilisés comme registres de transfert. Cela est effectué automatiquement par Unity Application Generator. UAG n'utilisera pas ces registres à d'autres fins et n'en modifiera pas le contenu.

Unity Application Generator n'affecte pas d'autres registres pour la zone de non-transfert.

Registre de commande de la redondance d'UC

Ce registre permet à l'utilisateur de manipuler les différents modes de fonctionnement de la redondance d'UC en cours d'exécution. Vous pouvez effectuer cette action soit grâce à une logique programmée, soit en accédant simplement au tableau d'animation dans Unity Pro.

⚠ AVERTISSEMENT

FONCTIONNEMENT IMPREVU DE L'EQUIPEMENT

N'utilisez pas le registre de commande à une autre fin dans toute l'application programmée. Toute utilisation incorrecte du registre de commande peut provoquer un comportement inattendu.

Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

Registres de transfert inverse

Les registres de transfert inverse peuvent servir à transmettre des données de diagnostic de l'automate en redondance d'UC vers l'automate principal. Elles sont copiées du redondant sur le principal.

Registre d'état

Ce registre offre des informations sur la logique utilisateur concernant la situation actuelle du système de redondance d'UC. Utilisez-le pour surveiller l'état actuel de la machine.

Configuration de redondance d'UC Premium

Définition d'un système de redondance d'UC

La redondance d'UC Premium est conçue pour servir dans les cas où les arrêts ne peuvent pas être tolérés. Le système offre une grande disponibilité grâce à la redondance.

Deux API à redondance d'UC sont configurés avec les mêmes matériels et logiciels.

L'automate principal exécute le programme d'application, gère les E/S Ethernet ainsi que les E/S "in-rack", et met à jour l'automate redondant au début de chaque scrutation (cycle de programme).

Si l'automate principal devient inopérant, l'automate redondant prend la relève au bout d'une durée maximale d'une scrutation.

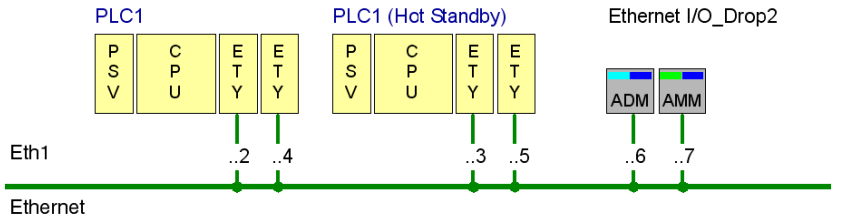
Il est possible de commuter l'UC principale et l'UC redondante. N'importe lequel des 2 automates peut jouer le rôle d'automate principal et l'autre d'automate redondant.

Par conséquent, si un des deux automates joue le rôle d'automate principal, l'autre doit être en mode redondant. Sinon, le second automate est en mode par défaut, c.à.d. hors ligne.

Les E/S Ethernet et les E/S "In-rack" redondantes sont gérées par l'automate principal.

Pour de plus amples détails concernant la redondance d'UC Premium, reportez-vous au *manuel utilisateur de la redondance d'UC Premium*.

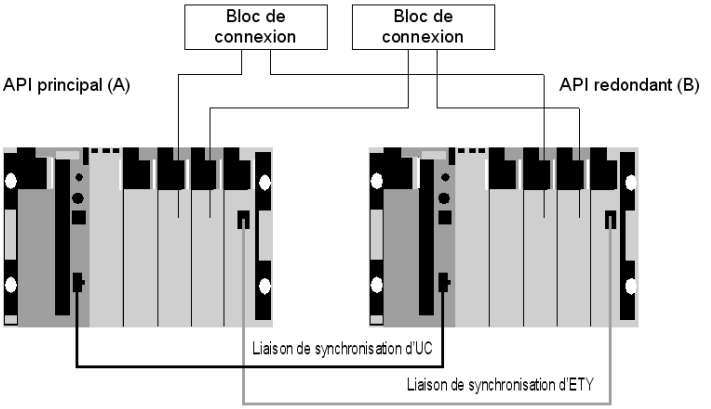
Exemple d'un système de redondance d'UC



NOTE : Chaque module ETY de l'API redondant reçoit l'adresse IP définie pour le module principal miroir + 1.

Principe de fonctionnement

Un système de redondance d'UC consiste en deux API identiques (le principal et le redondant) dont la configuration est identique en termes de matériel, de micrologiciels et de logiciels.



En mode normal, les deux API fonctionnent en parallèle mais seul l'API principal exécute l'application d'API.

L'API redondant exécute uniquement la première section de l'application d'API (section 0). Après chaque scrutation, la mémoire de l'API principal est copiée dans l'API redondant via la liaison de synchronisation d'UC.

La seconde liaison de synchronisation (ETY) est utilisée pour gérer le processus de transition. Seuls des modules ETY autorisés peuvent être utilisés pour cette fonctionnalité (voir les restrictions ci-dessous).

Le module ETY peut être utilisé en parallèle pour les scrutations d'E/S ou d'autres usages Ethernet. Des modules ETY supplémentaires (ne figurant pas dans les schémas de principe) peuvent également être déployés dans la configuration.

Restrictions

Limites de la configuration particulière de la redondance d'UC Premium :

- Modules autorisés
  - ETY4103/ETY5103 comportant des fonctionnalités plus évoluées pour la redondance d'UC à intégration de PV
  - toutes les alimentations
  - tous les châssis
  - toutes les cartes mémoires PCMCIA
  - tous les modules d'E/S discrètes et analogiques et le boîtier d'extension REY200 BusX
  - le module SCY pour les communications maîtres Modbus sur RS485
  - le module SCY et les modules SCP114 intégrés pour les communications maîtres et esclaves Modbus

  Tous les modules non répertoriés ci-dessus sont des modules non autorisés et ne peuvent pas être utilisés dans les configurations à redondance d'UC.

NOTE : UAG ne gère pas les versions (PV) de micrologiciel des modules matériels.

Configuration minimale d'un API redondant

La configuration minimale pour Unity Pro est constituée des modules suivants :

- fond de panier
- alimentation
- UC redondant Premium (TSX H57 24M ou TSX H57 44M)
- au moins un module de communication TCP/IP Ethernet (TSX ETY 4103 ou TSX ETY 5103)

Redondance d'UC et communication

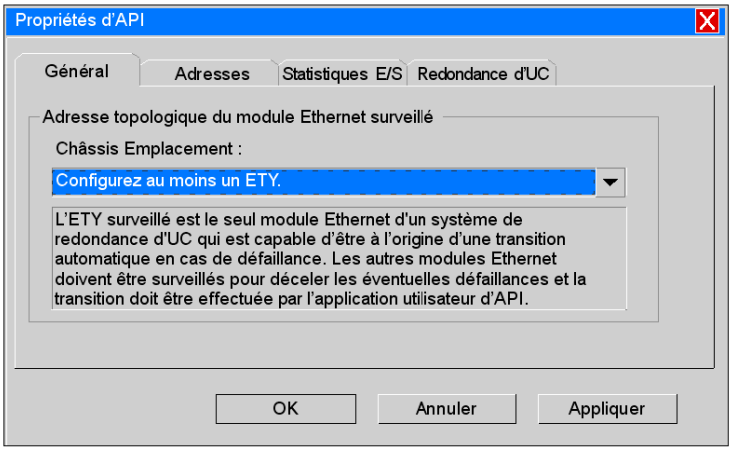
Même si un système de redondance d'UC est constitué de deux API, du point de vue des communications, il fonctionne comme un seul API. Néanmoins les deux API sont connectés au réseau. Pour cela, une adresse spéciale sur le réseau est automatiquement affectée à l'API redondant. Ces adresses ne peuvent pas être utilisées pour d'autres participants au réseau.

Adresse automatiquement affectée pour Ethernet :

- Adresse de l'API principal = x.y.z.n
- Adresse de l'API redondant = x.y.z.n+1
  Si n+1 = 255, alors l'adresse de l'API redondant = x.y.z.1

Onglet Redondance d'UC

Si vous choisissez comme nouvel API un automate Premium pour la redondance d'UC (TSX H57 24M ou TSX H57 44M), un onglet supplémentaire **Redondance d'UC** s'affiche dans la boîte de dialogue **Propriétés d'API**.



La boîte de dialogue vous invite à configurer au moins un module Ethernet.

Après avoir ajouté un ou plusieurs modules de communication Ethernet (TSX ETY 4103 ou TSX ETY 5103) au châssis d'API, vous pouvez faire de l'un de ces modules le "module Ethernet surveillé" en le sélectionnant dans la liste **Châssis, Emplacement** de l'onglet **Redondance d'UC**.

Zones de mémoire spécifiques à la redondance d'UC

Zone de mémoire/Mot système	Description
%MW0 à %MW100	zone non transférable Le bloc qui n'est pas transféré de l'API principal à l'API redondant. La taille et le décalage de ce bloc ne peuvent pas être modifiés.
%SW60 à %SW65	registres d'état/de commande/de transfert inverse

Pour plus d'informations, reportez-vous à la documentation Unity Pro.

Emploi de la zone de mémoire non transférable

Pour le système de redondance d'UC, la zone non transférable (%IW) est utilisée en association avec le diagnostic de redondance par l'intermédiaire d'une seule section (la première) exécutée dans l'automate redondant.

Cette logique doit fonctionner sur ses propres données (celles de l'automate redondant)

Pour Premium, l'emplacement non transférable commence à %MW0 et sa taille est de 100 mots (pour Quantum, l'emplacement non transférable et sa taille sont définis par l'utilisateur).

Pour contourner cette situation, l'idée, décrite ci-dessous, est d'utiliser l'affectation des adresses de UAG pour les communications API_IHM, situées dans une zone d'adresses définie par l'utilisateur.

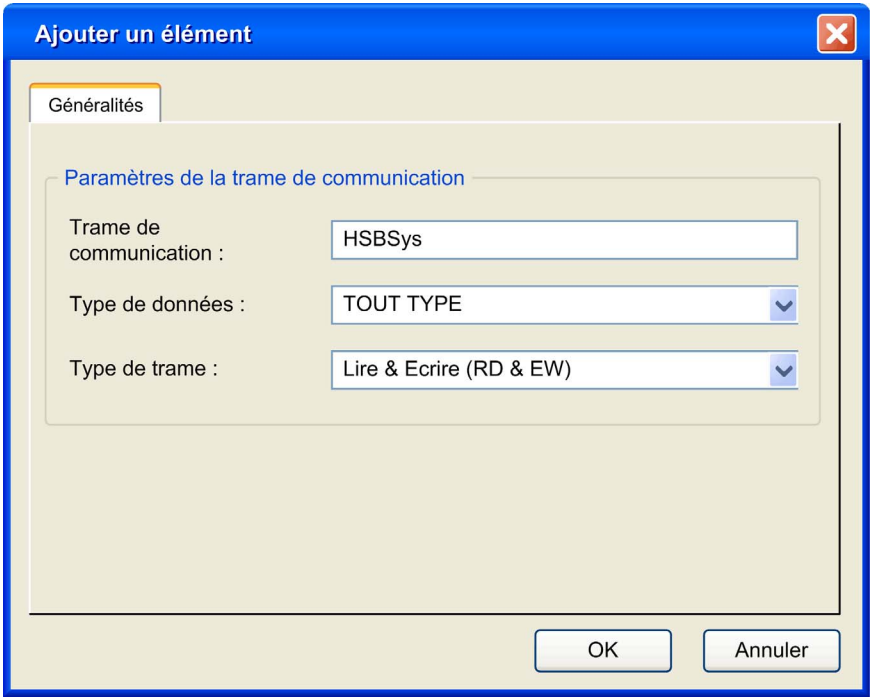
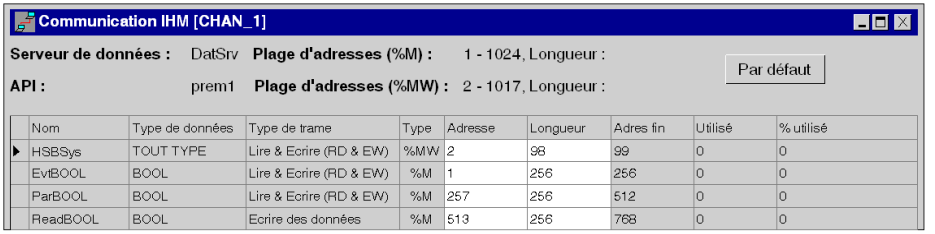
Au lieu d'avoir une interrogation des informations des variables en direction de l'IHM (ou parallèlement à cette interrogation), les variables seront affectées dans la zone non transférable.

Solution contournant l'utilisation de la zone non transférable

Voici 3 étapes qui décrivent la solution de contournement dans le cas de Premium (zone non transférable de %MW0 à %MW99).

Etape	Action
1	Définissez pour l'API le schéma d'adressage de telle manière que la zone %MW de la zone non transférable fasse partie du segment d'adresse réservé aux communications IHM. **Remarque :** l'adresse de début minimum pour la zone de mémoire IHM de Premium est généralement 1.
2	Définissez une trame de communication spécifique (par ex. étiquetée HsbyArea) dont l'emplacement correspond à la zone non transférable (adresse de début : 2; longueur 98) **Remarque :** l'adresse de début de la trame de communication doit être une adresse paire.
3	Créez un lien manuel ou automatique (par le biais de la définition des variables dans le SCoD) pour lier les variables à la trame de communication spécifique.

Il vous également possible d'utiliser la fonction prédéfinie UAG.

Etape	Action
1	Définissez la trame de communication prédéfinie appelée **HSBSys** (**Personnalisation** →**Données** → **Trame de communication**) avec un emplacement correspondant à la zone de non transfert (adresse de départ : 2; longueur 98). 
2	En utilisant le bouton **Par défaut** à l'intérieur du canal de l'IHM, le canal **HSBSys** est automatiquement configuré selon la longueur appropriée. 

Nom	Type de données	Type de trame	Type	Adresse	Longueur	Adres fin	Utilisé	% utilisé
HSBSys	TOUT TYPE	Lire & Ecrire (RD & EW)	%MW	2	98	99	0	0
EvtBOOL	BOOL	Lire & Ecrire (RD & EW)	%M	1	256	256	0	0
ParBOOL	BOOL	Lire & Ecrire (RD & EW)	%M	257	256	512	0	0
ReadBOOL	BOOL	Ecrire des données	%M	513	256	768	0	0

Solution contournant la définition de la logique spécifique de la redondance d'UC

UAG ne gère pas la section spécifique à la redondance d'UC (première section exécutée sur l'API principal et l'API redondant).

Vous devez créer l'équipement spécifique (par exemple, étiqueté HsbyLogic), notamment les modules de commande définis par l'utilisateur qui vont assurer la gestion du système de redondance d'UC.

Après la génération du projet UAG, vous devez modifier l'application Unity Pro afin que la section spécifique à la redondance d'UC soit exécutée en premier.

Sous-chapitre 5.2
Les groupes segments de réseau et chemins de routage

Introduction

Le groupe réseau renferme la configuration générale du réseau.

On définit dans le groupe réseau les éléments suivants :

- Les segments de réseau
- Les chemins de routage, et implicitement les routeurs

Vous trouverez une vue d'ensemble d'un automatisme typique et de son réseau au chapitre *Exemple d'automatisme type, page 120*.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Description et propriétés des segments de réseau	143
Liste des abonnés réseau	144
Description et propriétés des chemins de routage	146

Description et propriétés des segments de réseau

Que sont les segments de réseau ?

Un segment de réseau est une ligne simple de réseau d'un type défini (Modbus Plus ou Ethernet) disposant de toute la plage d'adresses. Les abonnés réseau tels que les API, IHM, serveurs de données, partenaires réseau et autres abonnés sont affectés à un ou plusieurs segments de réseau.

Comment les abonnés réseau sont-ils associés à un segment ?

Le rattachement de composants matériels à un segment n'est pas défini au sein du nœud de segment de réseau. Ce type de définition est établi comme propriété du groupe d'objets respectif.

Comment plusieurs segments sont-ils connectés entre eux ?

Les segments de réseau sont connectés par des ponts. La définition des ponts dépend du type de réseau.

- Pour Ethernet, les ponts sont définis par affectation de l'adresse de passerelle par défaut (dans UAG, les routeurs ne sont pas illustrés).
- Pour Modbus Plus, les ponts sont définis dans le groupe Chemins de routage.

Pour effectuer une liaison Modbus vers Ethernet, veuillez consulter la section Passerelle Modbus/Ethernet *(voir page 182)*.

Pour effectuer une liaison Profibus vers Ethernet, veuillez consulter la section Passerelle Profibus/Ethernet *(voir page 188)*.

Propriétés

Propriétés du segment :

Propriété	Commentaire
`Nom`	Nom unique du segment Ce nom doit respecter les conventions de nommage définies lors de la personnalisation.
`Type`	Définit le type de réseau A choisir dans une liste

Autres propriétés des segments Ethernet :

Propriété	Commentaire
`Masque sous-réseau`	Le masque de sous-réseau définit la partie de l'adresse IP relative au réseau et celle relative aux abonnés.
`Passerelle par défaut`	La passerelle par défaut définit l'adresse de routage vers d'autres segments de réseau.

Liste des abonnés réseau

Présentation

Pour une meilleure vue d'ensemble des partenaires réseau, un tableau affiche tous les abonnés connectés au segment réseau sélectionné. Ce tableau permet en outre de modifier aisément plusieurs adresses réseau.

Il est possible d'afficher deux types de réseaux : Ethernet et Modbus Plus.

Comment accéder aux abonnés réseau

Le tableau suivant explique comment obtenir les informations relatives aux `abonnés réseau`.

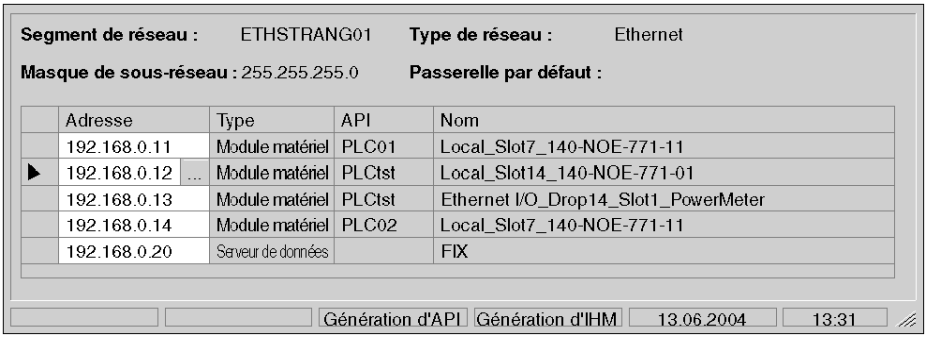
Etape	Action
1	Ouvrez l'explorateur **Vue** →**Explorateur**.
2	Sélectionnez les segments de réseau dans la vue topologique et développez l'arborescence `Segments de réseau`.
3	Sélectionnez un `segment de réseau`.
4	Ouvrez le menu contextuel en cliquant avec le bouton droit de la souris sur le segment sélectionné, puis choisissez `Ouvrir les abonnés réseau`.

Tableau des abonnés réseau

Le tableau des abonnés réseau comporte les colonnes suivantes :

- `Adresse`
  Indique l'adresse réseau des abonnés réseau connectés.
  Il est possible de modifier cet attribut.
- `Type`
  Indique le type d'abonné réseau, par exemple, `Module matériel`, `Serveur de donnés` ou autres.
  Il est impossible de modifier cet attribut.
- `API`
  Si le type est `Module matériel`, c'est le nom de l'API qui s'affiche.
  Il est impossible de modifier cet attribut.
- `Nom`
  Indique le nom de l'abonné réseau. Il peut s'agir, par exemple, du nom d'un serveur de données ou de l'identifiant d'un module matériel avec le type de châssis, le numéro d'emplacement et le type de module.
  Il est impossible de modifier cet attribut.

Le schéma suivant présente la liste des points de raccordement réseau pour un segment Ethernet.



L’en-tête du tableau contient le nom du `segment de réseau` ainsi que le `type de réseau` (Modbus Plus ou Ethernet). En outre, le `masque de sous-réseau` et la `passerelle par défaut` s’affichent pour les segments Ethernet.

Description et propriétés des chemins de routage

Qu'appelle-t-on chemins de routage ?

Les chemins de routage définissent les ponts qui relient différents segments de réseau Modbus Plus ainsi que leurs adresses.

Les chemins de routage doivent être définis très tôt dans le projet. Les définitions effectuées ici apparaîtront comme options pour les autres éléments.

Propriétés

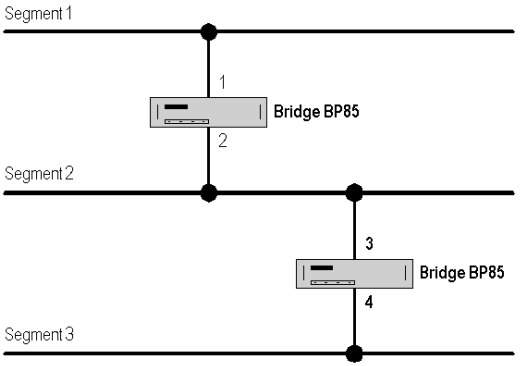
Propriétés des chemins de routage :

Propriété	Description	Options	Commentaire
`Emetteur`	Nom du segment d'émission	A choisir dans une liste	Les segments doivent être définis avant de pouvoir définir les ponts
`Récepteur`	Nom du segment de réception	A choisir dans une liste	Les segments doivent être définis avant de pouvoir définir les ponts
`Chemin`	Adresse du ou des ponts sur le segment émetteur		En fonction des options d'adressage réseau respectives

Exemple de définition de chemin

Les chemins entre segments, connectés au travers de plusieurs ponts, doivent être définis de façon explicite.

Exemple de définition de chemin :



Chemins pour différentes combinaisons émission / réception

Emetteur	Récepteur	Chemin
Segment 1	Segment 2	1
Segment 2	Segment 1	2
Segment 2	Segment 3	3
Segment 3	Segment 2	4
Segment 1	Segment 3	1.3
Segment 3	Segment 1	4.2

Sous-chapitre 5.3
Le groupe API

Introduction

Cette partie décrit le groupe API et ses propriétés.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Introduction	149
Propriétés de l'API	151
Canal API	156
Fonctions de l'API	159
Communication API <-> API via Modbus Plus	160
Communication API <-> API via Ethernet	162
Afficheur des communications	165
Châssis supplémentaires	167
Châssis et modules matériels	169
Informations générales sur les bus de terrain	177
Module Ethernet amélioré	179
Passerelle Modbus/Ethernet	182
Passerelle Profibus/Ethernet	188
Ethernet Quantum distant d'E/S (QERIO)	203
E/S M580 Ethernet	207

Introduction

Introduction

Introduction

Dans le groupe API, les automates ont les propriétés suivantes :

- Un API est constitué de plusieurs châssis.
- Un API possède des propriétés matérielles qui définissent le type d'API, d'UC, de châssis local, etc.
- Un API dispose de canaux de communication affectés. Ils définissent les autres nœuds de réseau avec lesquels une communication a lieu.

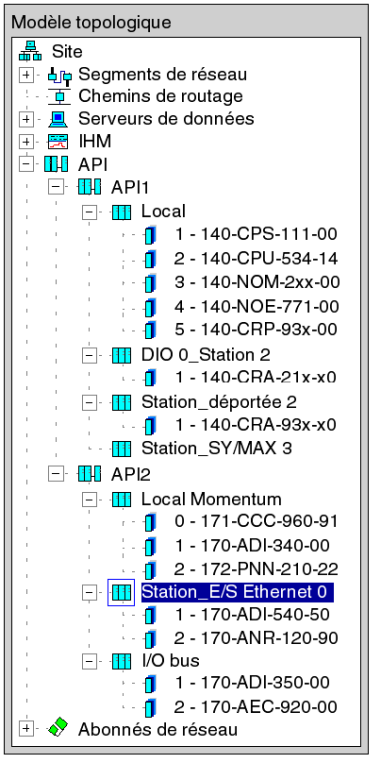
La configuration complète des châssis et des modules dans les châssis y est définie pour chaque API. La communication des modules avec le procédé y est également définie, par exemple comment les variables sont connectées aux modules d'E/S.

Structure du groupe API

Les groupes API contiennent :

- des API
  qui contiennent
- des châssis,
  qui contiennent
- des modules.

Exemple d’un groupe API



Propriétés de l'API

Présentation

Un API possède différents types de propriétés. Certaines doivent être définies de manière explicite par le concepteur, d'autres sont dérivées des définitions données dans d'autres objets.

Structure d'un API

Un API représente un seul projet Unity Pro. S'il existe plusieurs API, chacun d'eux dispose de son propre projet Unity Pro.

Vous pouvez panacher les API Unity Pro au sein du même projet UAG.

Pour chaque projet Unity Pro, l'abonné API est constitué des éléments suivants :

- le type d'API,
- la configuration d'API,
- l'affectation des E/S et
- des canaux de communication.

Propriétés à définir

Les propriétés suivantes doivent être définies pour chaque API :

Général	Commentaire
Nom	Nom unique de l'API
Description	Brève description de l'API
Nom de projet API	Nom unique pour le projet Unity Pro. Le projet est généré dans le **chemin d'accès du projet API** tel que défini dans la boîte de dialogue **Options** de Unity Application Generator. Chaque projet est enregistré dans son propre répertoire.
Gamme d'API	Définit le logiciel de programmation (Unity Pro), ainsi que la gamme d'automates dont l'API est issu. Les gammes d'automates possibles sont affichées dans une liste.
Châssis local	Définit le type du châssis local. **Pour Unity Pro :** Pour les API Quantum, vous pouvez définir des types de châssis entre 2 et 16. Sur les API Quantum, Premium, M340 et M580, il est possible de sélectionner le **châssis local** au cours de la création du nouvel automate. Il est possible de changer le châssis en accord avec ses propriétés.
UC	Type d'UC Les types d'UC possibles sont affichés dans une liste.

Adresses	Commentaire
Adresses	Les valeurs par défaut des plages d'adresses sont prédéfinies. Pour une configuration optimale, les plages d'adresses peuvent être affectées par l'utilisateur en remplissant les colonnes **Début** et **Longueur**. Au sein de ces plages, les adresses sont automatiquement affectées lors du processus de génération de cartographie mémoire. ● Quantum/Momentum : les E/S (E/S locales, E/S Ethernet, E/S Momentum, etc.) sont affectées à la plage d'adresses des **modules matériels**. Pour les API Unity Quantum, sont en outre prévues des plages d'adresses réservées pour le **bus de terrain** (Profibus) (%IW, %MW, voir ci-après). ● Premium : pour les E/S situées sur le XBus, aucune plage d'adresses n'est affectée, car les adresses utilisées sont topologiques. Les E/S situées sur un bus (E/S Ethernet, E/S Modbus Plus, E/S Momentum, etc.) sont affectées à la plage d'adresses de **Communication API**. ● M340 : pour les E/S situées sur le châssis local, aucune plage d'adresses n'est affectée car les adresses utilisées sont topologiques. Les E/S situées sur un bus (E/S Ethernet, E/S Momentum, etc.) sont affectées à la plage d'adresses de **Communication API**. ● M580 : Pour les E/S situées sur une station XBus, locale ou d'E/S Ethernet M580, l'adressage n'est pas pris en charge. **Remarque** : une modification des plages d'adresses provoque une réaffectation des adresses. Cette opération peut nécessiter l'arrêt de l'API.
Réserve	Pour les différents types de registres (%M, %I, %IW et %MW de Unity/Quantum et %M et %MW de Unity/Premium), il est possible de définir des plages de registres réservées. Ces plages ne seront pas utilisées par Unity Application Generator. Elles peuvent l'être dans Unity Pro pour des fonctionnalités non définies dans Unity Application Generator, par exemple pour des modules E/S sur bus de terrain.

Bus de terrain %IW, %MW

Pour les API Unity Quantum, des plages d'adresses de bus terrain (Profibus) sont prévues pour faciliter la vérification de la non-utilisation des plages de mémoire du maître Profibus par une autre partie du système API.

Dans l'onglet **Adresses**, deux lignes supplémentaires sont proposées :

- Bus de terrain %IW
  L'**Adresse de début** par défaut d'un nouvel API est l'adresse de fin des modules matériels + 1 et la **Longueur** par défaut est 1024.
- Bus de terrain %MW
  L'**Adresse de début** par défaut d'un nouvel API est l'adresse de fin des modules matériels + 1 et la **Longueur** par défaut est 1024.

Modification des plages d'adresses du bus de terrain

Vous pouvez modifier l'**Adresse de début** et la **Longueur** du bus de terrain.

L'adresse de début doit être une valeur entière positive comprise entre 1 et 99999.

La longueur doit être une valeur entière positive comprise entre 0 et 99999.

La **Longueur réelle** et l'**Adresse de fin** sont en lecture seule.

La longueur réelle est calculée en se basant sur la configuration Profibus actuelle du projet. La longueur de la plage est constituée de la somme de l'ensemble des mots de mémoire (dans cette plage), occupée par tous les modules maîtres Profibus que contrôle cet API.

L'adresse de fin est constituée de la somme de l'**Adresse de début** et de la **Longueur**, moins 1.

Si vous cliquez sur **OK** ou sur **Appliquer**, UAG teste la cohérence des valeurs d'**Adresse de début** et d'**Adresse de fin** pour la configuration Profibus du projet UAG, et vérifie l'existence de conflits éventuels avec d'autres plages de mémoire.

Vérifications de la cohérence des adresses de bus terrain

Les vérifications suivantes sont effectuées :

- L'adresse de début ne peut pas être supérieure à l'adresse de valeur la plus faible de tous les modules maîtres Profibus du projet UAG. Si l'adresse de début ne répond pas à ce critère, le message suivant apparaît :
  "L'adresse de début de la plage de bus terrain ne peut pas être supérieure à la valeur définie pour le module. Souhaitez-vous définir l'adresse de début en correspondance avec votre configuration de bus de terrain ?"
  Si vous acceptez le message en choisissant **Oui**, l'**Adresse de début** est fixée à la valeur proposée. Indépendamment de votre réponse, le champ d'adresse de début devient actif pour vous permettre de corriger la valeur.
- L'adresse de fin ne peut pas être inférieure à l'adresse de valeur maximale de tous les modules maître Profibus du projet UAG. Si l'adresse de fin ne répond pas à ce critère, le message suivant apparaît :
  "L'adresse de fin de la plage de bus terrain ne peut pas être inférieure à la valeur définie pour le module. Souhaitez-vous définir la longueur requise minimale en correspondance avec votre configuration de bus de terrain ?"
  Si vous acceptez le message en choisissant **Oui**, la **Longueur** est fixée à la longueur minimale requise. Indépendamment de votre réponse, le champ de longueur devient actif pour vous permettre de corriger la valeur.
- La plage de mémoire de bus de terrain ne peut pas recouvrir une autre plage de mémoire. Si la plage de mémoire du bus de terrain ne répond pas à ce critère, le message suivant apparaît :
  "Le bus de terrain (%xW) occupe la plage de mémoire xxxxx (%xW)".

Projets UAG de versions antérieures

Les plages d'adresses réservées pour un bus de terrain (Profibus) sont prises en charge depuis UAG 3.1.

Si vous ouvrez un projet UAG existant (antérieur à UAG 3.1), la plage d'adresses de bus de terrain est ajoutée aux API Unity Quantum et les valeurs d'adresses de début et de longueur sont fixées.

Deux cas sont possibles :

- Le projet comporte une configuration Profibus.
  L'adresse de début et la longueur doivent être définies en se basant sur la configuration Profibus présente.
- Le projet ne comporte pas de configuration Profibus.
  Adresses de début d'entrée bus de terrain %IW : 1025, Longueur : 0
  Adresses de début de sortie bus de terrain %IW : 1025, Longueur : 0

Modification du nom d'un API

Pour renommer un API, vous devez modifier le nom dans la boîte de dialogue Propriétés de l'API puis générer l'API. Avec la génération suivante, UAG fait en sorte que les anciens noms de l'API soient modifiés et deviennent le nouveau nom.

Modification du nom d'un API et d'un nom de canal approprié

Pour modifier le nom d'un API ainsi que celui d'un canal approprié, procédez comme suit :

Etape	Action
1	Pour le changement du nom d'un API, consultez également la section *Modification du nom d'un API, page 154*. **Résultat :** Le nom de l'API est modifié dans l'intégralité du projet UAG.
2	Sélectionnez le nom de l'API dans la vue topologique et cliquez le bouton droit de la souris. **Résultat :** Le menu contextuel s'affiche.
3	Sélectionnez **Ouvrir canal** dans le menu contextuel. **Résultat :** Les canaux s'affichent dans la vue Liste.
4	Sélectionnez le nom du canal dans la vue Liste et appuyez sur le bouton droit de la souris. **Résultat :** Le menu contextuel s'affiche.
5	Sélectionnez **Propriétés...** dans le menu contextuel. **Résultat :** La boîte de dialogue **Propriétés** s'affiche.

Etape	Action
6	Modifiez le **Nom** et cliquez sur **OK**. **Résultat :** La boîte de dialogue **Propriétés** se ferme.
7	Sélectionnez **Génération** →**API** pour générer le projet UAG avec le nouveau nom de canal. **Résultat :** La progression de la génération s'affiche dans la vue Liste.

Propriétés affichées

Les propriétés suivantes peuvent être affichées pour chaque API :

Propriété	Commentaire
Statistiques d'E/S	L'onglet **Statistiques d'E/S** donne des informations sur le nombre de modules E/S requis par les variables d'E/S des éléments physiques fonctionnels et modules de commande existants. Il indique le nombre de toutes les variables d'E/S, les variables d'E/S affectées et les points d'E/S libres séparément pour les entrées/sorties, numériques/analogiques.

Canal API

Qu’est-ce qu’un canal ?

Un canal décrit :

- les objets de l’automatisme avec lesquels l’API communique ;
- sur quel chemin de communication aura lieu la communication ;
- un chemin alternatif de communication ;
- les variables devant être échangées.

La communication s’effectue par un API. Cela signifie que l’un des partenaires de communication d’un canal est un API.

Définition de la communication

La définition de la communication et la nature des données à communiquer par un canal dépendent du partenaire de communication.

Définition de la communication :

Communication Partenaire de la communication	**Méthode de définition**
Serveur de données	Pour chaque API, un canal doit être affecté à un seul serveur de données. Les variables devant être communiquées au serveur de données se définissent implicitement par la relation entre un élément physique fonctionnel et un API, la zone de conduite (et donc l’IHM) à laquelle l’élément physique fonctionnel est affecté et le type de communication de la variable. Toutes les variables du type de communication API_IHM sont communiquées.
API	Chaque variable à communiquer est placée dans la table de communication API par une opération glisser-déposer.
Partenaire réseau	Chaque variable à communiquer doit être mise dans la table de communication réseau par une opération glisser-déposer (seulement les variables de type API_RES).

Méthodes de communication

La méthode de communication dépend du partenaire de communication.

Méthodes de communication

Partenaire de communication	Méthode	Commentaire
API	Pour Modbus : Diffusion des E/S	Les données de diffusion des E/S ne sont transmises qu'entre abonnés sur le réseau local. Le transfert par des ponts vers d'autres abonnés n'est pas possible. Les données de diffusion des E/S sont limitées à 32 mots. Un mot est utilisé pour vérifier l'état de santé de la communication. **Remarque** : La communication entre différents réseaux avec MSTR n'est pas gérée.
	Pour Ethernet : Scrutateur d'E/S	Les données du scrutateur d'E/S sont limitées à 100 mots. Un mot est utilisé pour vérifier l'état de santé de la communication. Pour les modules matériels des châssis Ethernet et Advantys (Ethernet NIM seulement) et des passerelles Modbus/Ethernet, il est possible de définir le **taux de répétition de scrutateur d'E/S**. **Remarque** : La communication entre différents réseaux avec MSTR n'est pas gérée.
IHM	Par variables localisées	Les entrées opérateur sont immédiatement transmises à l'API. **Pour iFIX :** La fréquence d'actualisation en lecture des valeurs API dépend de la catégorie d'actualisation. Il existe trois catégories de fréquence d'actualisation (périodicité d'établissement de la communication). La catégorie est définie par type de commande et alarme de la variable : ● Événement Les événements ont la plus haute priorité parce que les événements comme les alarmes doivent être transmis dès que possible. ● Synoptique Les variables de synoptique représentent en général les valeurs temps réel des écrans animés. Elles sont transmises lorsque l'écran est affiché. ● Paramètre Les paramètres sont des valeurs qui sont transmises moins souvent. **Pour Monitor Pro et VijeoCitect :** L'utilisateur doit définir les trames de communication. Les variables seront affectées aux trames.

Partenaire de communication	Méthode	Commentaire
Partenaires réseau	Par variables localisées	Dans l'outil de programmation du partenaire réseau, l'ingénieur automaticien doit veiller à scruter les données nécessaires. Les partenaires réseau communiquent par des adresses API, les variables à communiquer doivent donc être situées dans la plage de mémoire appropriée. Il faut, pour chaque canal vers un partenaire réseau, configurer les plages des bits (0x/%M) et des mots (4x/%MW). Il est possible de créer un fichier de valeurs séparées par virgules (CSV) contenant ces variables et adresses. Ce fichier peut être alors importé dans le logiciel de programmation du partenaire réseau.

Fonctions de l’API

Copier et coller des API

L’utilisateur de Unity Application Generator peut effectuer des opérations de copier-coller sur tous les API au sein du modèle topologique. La copie de l’API inclut chaque châssis local ou distant appartenant à cet automate.

NOTE : Après avoir copié et collé un API, toutes les adresses réseau des segments Modbus Plus et/ou Ethernet doivent être mises à jour.

Supprimer des API

NOTE : Si vous travaillez avec iFIX, observez les règles suivantes :

si vous supprimez un API, vous devez générer votre application iFIX tout de suite après, avant d’apporter de quelconques modifications à votre projet UAG.

Communication API <-> API via Modbus Plus

Méthode de communication

Pour les communications API <-> API via Modbus Plus, Unity Application Generator établit la communication sous la forme d'une diffusion des E/S. La communication par bloc fonction MSTR n'est pas gérée.

Unity Application Generator génère automatiquement la configuration de diffusion des E/S ainsi que la logique nécessaire à la communication. Afin de définir la communication de diffusion des E/S avec Unity Application Generator, il faut comprendre certains aspects de base de la diffusion des E/S.

Fonctions de communication prises en charge

La communication de diffusion des E/S est limitée à 32 mots par canal. Unity Application Generator utilise un mot pour contrôler l'état de santé. Les 31 autres mots sont libres.

16 bits (BOOL) sont communiqués dans un mot (WORD). Unity Application Generator génère la logique pour transférer les BOOL en WORD.

Les variables des autres types sont également transformées en WORD, par ex. un REAL est divisé en deux WORD.

Chaque connexion Modbus Plus au réseau Modbus Plus peut servir pour :

- Quantum/Momentum :
  Un canal global, ce qui signifie que l'information transitée est envoyée à tous les API du même segment et reçue par tous ces API.
- Quantum/Momentum/Premium :
  Un canal spécifique, ce qui signifie que l'information est explicitement envoyée à un autre API ou reçue par un autre API.

Un API Quantum peut disposer d'un maximum de 3 connexions au réseau Modbus Plus (par l'UC et 2 NOM).

Un API Premium peut avoir 1 connexion au réseau Modbus Plus (par la carte TSX MBP 100 PCMCIA).

Un API Momentum peut avoir 1 connexion au réseau Modbus Plus (par la carte de communication 172-PNN-210-22 ou 172-PNN-260-22).

Vérification de l'état de santé

Pour Quantum et Momentum :

Unity Application Generator crée sa propre logique de communication. Un mot sert de compteur pour vérifier si la communication est valide ou a été altérée. Un mot de chaque canal est affecté à cet usage.

Pour Premium :

Pour vérifier l'état de santé, utilisez les IODDT (`T_COM_ET*`) dans Unity Pro.

Pour M340 :

Pour vérifier l'état de santé, utilisez le IODDT (`T_COM_ET_BMX`) dans Unity Pro.

Redondance de canal

Pour Quantum et Momentum :

Afin de configurer un canal de communication redondant, il est possible de définir deux autres chemins de routage.

Interruptions de communication

Si la première connexion Modbus Plus est interrompue, la logique générée bascule sur les canaux de communication redondants, ce sont donc les valeurs de la seconde connexion de la diffusion des E/S qui sont utilisées.

Si la seconde connexion Modbus Plus est également interrompue, les valeurs utilisées sont celles définies par défaut. Si aucune valeur par défaut n'est spécifiée, les sorties sont positionnées à 0 ou Faux.

Si aucune communication redondante n'est définie, en cas d'interruption de la communication, la logique générée passe directement aux valeurs par défaut.

Paramétrage d'un canal spécifique

Pour paramétrer un canal particulier, les règles suivantes s'appliquent :

- le canal doit être défini uniquement pour l'API émetteur.
- Un canal spécifique est défini par sélection d'un API spécifique comme "Partenaire de communication".
- Un canal pour l'API récepteur est automatiquement créé en définissant le canal de l'API émetteur.

Paramétrage d'un canal global

Pour paramétrer un canal global, les règles suivantes s'appliquent :

- Il faut définir un canal pour l'API émetteur et pour tous les API récepteurs.
- Pour définir un canal global pour l'API émetteur, sélectionnez "TOUS" comme "Partenaire de communication".
- Pour définir un canal global pour l'API récepteur, sélectionnez l'option "Sélection parmi les réceptions globales existantes" et sélectionnez ensuite un canal existant dans la liste.

Communication API <-> API via Ethernet

Méthode de communication

Pour les communications API <-> API via Ethernet, Unity Application Generator établit la communication par le scrutateur d'E/S Ethernet.

La communication par bloc fonction MSTR n'est pas prise en charge.

Unity Application Generator génère automatiquement la configuration du scrutateur d'E/S ainsi que la logique nécessaire à la communication. Afin de définir la communication de scrutateur d'E/S avec Unity Application Generator, il faut comprendre certains aspects de base du scrutateur d'E/S Ethernet.

Fonctions de communication prises en charge

Fonctions de communication

- Selon le module, il est possible de définir 64 ou 128 canaux.
- Jusqu'à 100 mots peuvent être échangés sur chaque canal. L'un de ces mots sert à vérifier l'état de santé du canal de communication.
- 16 bits (BOOL) sont communiqués dans un mot (WORD). Unity Application Generator génère la logique pour transférer les BOOL en WORD.
  Les variables des autres types sont également transformées en WORD ; par ex. un REAL est divisé en deux WORD.
- En général, chaque entrée du tableau de configuration du scrutateur d'E/S Ethernet permet de définir la lecture et l'écriture de registres donnés. Unity Application Generator n'utilise que la fonction écriture de chaque canal/entrée.

Vérification de l'état de santé

Pour Quantum et Momentum :

Unity Application Generator crée sa propre logique de communication. Un mot sert de compteur pour vérifier si la communication est valide ou a été altérée. Un mot de chaque canal est affecté à cet usage.

Le scrutateur d'E/S Ethernet Quantum nécessite, selon la norme, la définition d'un bloc d'état de santé de 8 mots. Unity Application Generator affecte donc 8 mots dans la plage des registres 3x.

Le scrutateur d'E/S Ethernet Quantum nécessite en standard la définition d'un bloc de diagnostic de 128 mots. Unity Application Generator affecte donc 128 mots dans la plage des registres 4x.

Le scrutateur d'E/S Ethernet Momentum nécessite en standard la définition d'un bloc d'état de santé de 4 mots. Unity Application Generator affecte donc 4 mots dans la plage des registres 3x.

Le scrutateur d'E/S Ethernet Momentum nécessite en standard la définition d'un bloc de diagnostic de 64 mots. Unity Application Generator affecte donc 64 mots dans la plage des registres 4x.

Ces plages ne sont pas utilisées par Unity Application Generator pour vérifier l'état de santé, voir ci-dessus.

Pour Premium :

Pour vérifier l'état de santé, utilisez les IODDT dans Unity Pro.

Redondance de canal

Pour Quantum et Momentum :

Afin de configurer un canal de communication redondant, il est possible de définir deux autres chemins de routage. Si la redondance est sélectionnée, un canal utilise 2 entrées du scrutateur d'E/S Ethernet.

Interruptions de communication

Si la première connexion Ethernet est interrompue, la logique générée bascule sur les canaux de communication redondants, ce sont donc les valeurs de la seconde entrée du scrutateur d'E/S qui sont utilisées.

Si la seconde connexion du scrutateur d'E/S Ethernet est également interrompue, les valeurs utilisées sont celles définies par défaut. Si aucune valeur par défaut n'est spécifiée, les sorties sont positionnées sur 0 ou Faux.

Si aucune communication redondante n'est définie, en cas d'interruption de la communication, la logique générée passe directement aux valeurs par défaut.

Configuration du scrutateur d'E/S Ethernet générée

Pour configurer un canal à l'aide du scrutateur d'E/S Ethernet, Unity Application Generator établit la configuration du scrutateur d'E/S Ethernet conformément aux définitions suivantes :

- Adresse IP esclave (adresse abonné cible)
- Timeout de l'état de santé : Unity Application Generator utilise sa propre logique pour vérifier l'état de santé.
- Le taux de répétition de scrutateur d'E/S est fixé par Unity Application Generator à 100 ms par défaut, mais peut être modifié dans les propriétés de canal.
- Réf. d'écriture maître (adresse de départ dans l'API émetteur)
- Réf. écriture esclave (adresse de départ dans l'API récepteur)
- Longueur d'écriture (nombre de mots à communiquer, 100 max.)

Ces définitions ne sont effectuées que dans l'API émetteur. Le partenaire de communication récepteur ne nécessite aucun paramétrage dans son scrutateur d'E/S Ethernet.

⚠ AVERTISSEMENT

FONCTIONNEMENT IMPREVU DE L'EQUIPEMENT

Dans Unity Pro, il est interdit de procéder à des changements manuels du scrutateur d'E/S Ethernet sauf pour le réglage du **taux de répétition du scrutateur d'E/S**. Le non-respect de cette règle peut engendrer des problèmes de communication.

Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

Paramétrage d'un canal Ethernet

Pour paramétrer un canal Ethernet, les règles suivantes s'appliquent :

- le canal doit être défini uniquement pour l'API émetteur.
- Un canal pour l'API récepteur est automatiquement créé en définissant le canal de l'API émetteur.

Afficheur des communications

Présentation

Les informations affichées indiquent l'état des variables sur les communications API-API.

Il est possible de faire glisser des variables depuis l'**afficheur des communications** et de les déposer dans des canaux API-API.

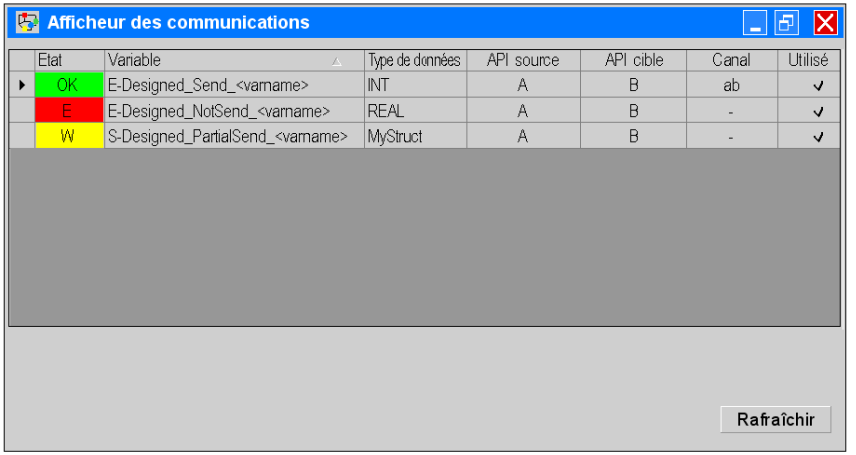
Double-cliquez sur une variable pour lancer la boîte de dialogue Propriété de variable associée (reportez-vous à la section *Propriétés des variables liées à l'API, page 99*).

Les informations présentées dans l'**afficheur des communications** se rapportent au moment où l'afficheur est ouvert. L'afficheur possède un bouton d'actualisation pour indiquer qu'il s'agit d'une vue à un instant donné. L'actualisation n'est pas dynamique.

Accès

Vous pouvez accéder à cet afficheur dans **UAG Application Generator** via **Affichage** → **Afficheur des communications** ou en cliquant sur l'icône correspondante dans la barre d'outils.

Afficheur des communications



Propriétés de l'**afficheur des communications**

Propriété	Description
Etat	Indique si une variable est correctement configurée en vue des communications "Correctement configurée en vue des communications" signifie que la variable (ou le composant d'une variable structurée ou d'une variable tableau) est utilisée dans des liaisons ou des forçages qui traversent une limite d'API, et à ce titre est correctement affectée à un canal de communication entre les API respectifs. ● **OK** (couleur verte) La variable est correctement configurée en vue des communications. La colonne **Utilisé** indique que la variable est utilisée dans des liaisons ou dans des forçages dans l'API déporté. Les variables inutilisées ne sont pas cochées dans cette colonne. ● **W** = Alerte (couleur jaune) La variable fait partie d'une structure utilisée dans l'API déporté mais non explicement utilisée et la variable n'est pas configurée en vue des communications. Les informations sont héritées de son objet parent. ● **E** = Erreur détectée (couleur rouge) La variable n'est pas configurée correctement en vue des communications. S'il existe au moins 1 élément d'une variable structurée qui n'est pas correctement configuré, la lettre **E** apparaît également pour tous les parents de cet élément de manière à ce que vous puissiez passer à l'élément pour lequel une erreur a été détectée, en développant l'entrée correspondante. Pour une entrée de variable marquée d'un **E**, double-cliquez sur l'entrée pour ouvrir la boîte de dialogue Propriété de variable.
Variable	Nom de la variable de communication
Type de données	Type de données de la variable de communication
API source	API source
API cible	API cible
Canal	Nom du canal
Utilisé	Indique si la variable est utilisée ou non dans des liaisons/forçages de l'API déporté
Bouton **Rafraîchir**	Activez ce bouton pour actualiser l'afficheur des communications

Châssis supplémentaires

Présentation

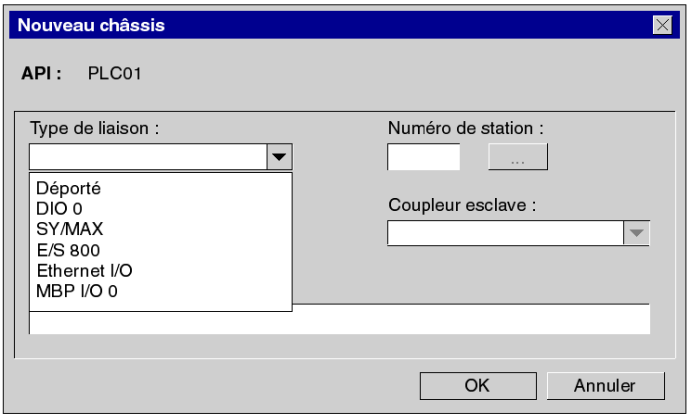
Il est impossible de connecter des modules Ethernet ou Modbus Plus directement dans des châssis locaux ou distants. L'utilisateur doit donc réorganiser les châssis "virtuels" dans UAG pour la communication par diffusion des E/S (Modbus Plus) et/ou par scrutation d'E/S (Ethernet).

Création d'un châssis supplémentaire

Après la création d'un nouvel API avec un châssis local, l'utilisateur peut être obligé de créer des châssis supplémentaires. Le châssis API local est automatiquement équipé d'un module d'UC. La communication (via Modbus Plus ou Ethernet) ne nécessite aucun réglage. Des réglages sont nécessaires pour pouvoir créer un châssis supplémentaire.

Ces châssis peuvent être nécessaires pour pouvoir utiliser des modules Ethernet, Modbus Plus ou autres améliorés.

Le schéma suivant présente un exemple de `types de liaisons` pour les nouveaux châssis.



Nouveau châssis contre nouveau châssis d'extension

L'utilisateur a la possibilité de sélectionner un `nouveau châssis` ou un `nouveau châssis d'extension`. Dans le cas d'un `nouveau châssis d'extension`, vous devez installer un module d'extension sur le châssis local. Pour un nouveau châssis d'E/S Modbus Plus ou d'E/S Ethernet installé en parallèle d'un châssis existant, l'utilisateur a besoin d'une communication Modbus Plus ou Ethernet. Ces châssis sont virtuels, ils sont uniquement destinés à la communication par diffusion et/ou scrutation d'E/S. Après la génération du code dans Unity Pro, vous n'obtenez pas de châssis physiques mais des réglages de registres pour la communication par diffusion ou scrutation d'E/S avec les modules connectés.

Paramètres de communication

La configuration de la communication est nécessaire pour pouvoir créer un châssis parallèle à un châssis existant. L'utilisateur doit définir les paramètres de communication sur le châssis local, par exemple, il doit ajouter un module Ethernet destiné à la communication avec un module Ethernet amélioré ou bien régler la communication Modbus Plus. L'utilisateur doit d'abord créer des `segments de réseau`.

Les E/S déportées requièrent uniquement un coupleur maître, mais elles ne sont pas connectées à un réseau.

Châssis possibles

L'utilisateur a la possibilité d'ajouter différents châssis en fonction des paramètres et des possibilités de communication de l'API. Consultez également *Châssis et modules matériels, page 169*

Châssis et modules matériels

Présentation

Tous les API avec leurs châssis et modules sont définis dans Unity Application Generator pour les applications Unity Pro. Les châssis et modules disponibles dépendent de la gamme d'automate définie dans les Propriétés API, ainsi que du paramétrage du fichier de personnalisation.

Tous les API avec leurs châssis par défaut sont définis dans Unity Application Generator pour les applications Unity.

Selon le type de module sélectionné, il existe d'autres boîtes de dialogue et fenêtres dans Unity Application Generator :

- A l'aide de la fenêtre de module d'E/S et de la fenêtre de variable, il est possible d'affecter les variables d'E/S aux points d'E/S (glisser une variable depuis l'afficheur de variables et la déposer sur un point d'E/S).
- Les modules de communication ou les UC possèdent une boîte de dialogue des propriétés pour affecter le segment de réseau et les adresses réseau.
- Les modules de sorties logiques possèdent une boîte de dialogue des propriétés pour affecter le comportement en cas de timeout.
- Pour les modules des châssis Ethernet et Advantys (Ethernet NIM seulement), sur des passerelles Modbus/Ethernet et Profibus/Ethernet, il est possible de définir le **taux de répétition du scrutateur d'E/S**.

La définition des châssis et modules crée la configuration API et l'affectation des E/S, ainsi que l'affectation des variables aux points d'E/S des modules.

Dans les cas suivants, aucune entrée n'est générée dans l'affectation des E/S :

- modules d'E/S Momentum Modbus Plus,
- modules d'E/S Momentum Ethernet et
- modules génériques

Device Control Block dans Unity Pro

Lorsque la fonction **Device Control Block** de Unity Pro est activée, la mémoire dédiée doit être prise dans la zone **réservée**. Cette définition s'effectue dans la boîte de dialogue Propriétés de l'API.

NOTE : Si vous n'activez pas la mémoire, vous risquez d'avoir de graves interruptions de la communication.

Propriétés du châssis (Unity Pro)

Les propriétés suivantes doivent être définies.

Propriété	Description	Options Quantum	Options Premium	Options M340	Options M580	Commentaire
`Type de liaison`	définit le mode de connexion du châssis à l'API	• Local • Local (Ex) • Déporté • Déporté (Ex) • DIO 0 • DIO 1 • DIO 2 • SY/MAX • E/S 800 • E/S MBP 0 • E/S MBP 1 • E/S MBP 2 • Périphérique MBP 0 • Périphérique MBP 1 • Périphérique MBP 2 • E/S Ethernet • Périphérique Ethernet • Generic • E/S Modbus • E/S Profibus • Quantum ERIO **NOTE :** Le type de liaison ERIO est pris en charge pour UnityPro 6.0 et les versions ultérieures.	• XBus • E/S Ethernet • Périphérique Ethernet • E/S MBP 0 • Périphérique MBP 0 • E/S Modbus • E/S Profibus	• E/S Ethernet • Périphérique Ethernet • E/S Modbus • E/S Profibus	• Local • XBus • E/S Ethernet	Vous trouverez dans les tableaux ci-dessous les commentaires sur les options.

Propriété	Description	Options Quantum	Options Premium	Options M340	Options M580	Commentaire
Numéro de station	définit le numéro de la station	• E/S déportées : 1...31 • Quantum ERIO : 1...31 • E/S distribuées : 1...64 • E/S MBP : 1 • E/S Ethernet : Le numéro de station correspond à l'emplacement du module Ethernet. • E/S Modbus : Le numéro de station correspond à l'emplacement du module Ethernet.	• XBus : 0...7 • E/S MBP : 1 • E/S Ethernet : Le numéro de station correspond à l'emplacement du processeur/ETY. • E/S Modbus : Le numéro de station correspond à l'emplacement du module Ethernet.	• E/S Ethernet : Le numéro de station correspond à l'emplacement du processeur/ETY.	• Locale et XBus : 0...7 **NOTE :** Le nombre de stations locales et XBus pouvant être configurées dépend du processeur sélectionné. • E/S Ethernet : 1...16 Le numéro de station correspond à l'emplacement du processeur/ETY.	Numéro de station (numéro d'équipement) définit l'adresse Modbus Plus de ce segment dans le cas d'E/S distribuées.

Propriété	Description	Options Quantum	Options Premium	Options M340	Options M580	Commentaire
Type de châssis	définit le type du châssis	• Local, RIO, DIO • Quantum ERIO • SY/MAX • E/S 800 • E/S MBP : E/S MBP (64 emplacements) • E/S Ethernet : E/S Ethernet (128) • E/S Modbus : Type de passerelle (passerelle Modbus vers Ethernet) • E/S Profibus : Type de passerelle (passerelle Profibus vers Ethernet) • Générique : Générique (255 emplacements)	• XBus • E/S MBP : E/S MBP (64 emplacements) • E/S Ethernet : E/S Ethernet (64) • E/S Modbus : Type de passerelle (passerelle Modbus vers Ethernet) • E/S Profibus : Type de passerelle (passerelle Profibus vers Ethernet)	• E/S Ethernet : E/S Ethernet (64) • E/S Modbus : Type de passerelle (passerelle Modbus vers Ethernet) • E/S Profibus : Type de passerelle (passerelle Profibus vers Ethernet)	• Local • XBus • E/S Ethernet	-

Commentaires sur les options du `type de liaison` pour Quantum :

Option du type de liaison	Commentaire
Local	Le châssis local est créé automatiquement en même temps que l'API.
Local (Ex)	Un châssis connecté au châssis local par une extension en fond de panier (XBE).
Déporté	Un châssis connecté comme E/S déportée. Il est nécessaire de configurer un module déporté (CRP-93x-00) dans le châssis local pour connecter l'E/S déportée.
Déporté (Ex)	Un châssis connecté au châssis d'E/S déporté par une extension en fond de panier (XBE).
Liaison DIO	Un châssis connecté comme E/S distribuée. Un châssis connecté au segment Modbus Plus connecté à l'UC est appelé DIO 0. Il est possible de connecter d'autres E/S distribuées avec un ou deux NOM-2xx-00 en DIO 1 et DIO 2.
E/S MBP	Un châssis virtuel connecté sous la forme d'E/S Modbus Plus. Un châssis relié à l'UC est appelé MBP 0. Il est possible de relier d'autres E/S Modbus Plus avec un ou deux NOM-2xx-00 en tant que liaisons MBP 1 et MBP 2.
E/S Ethernet	Un châssis virtuel qui est connecté par des E/S Momentum Ethernet. Pour chaque châssis d'E/S Ethernet, il est nécessaire de configurer un module de communication Ethernet dans le châssis local.
SY/MAX	Un châssis appartenant à la gamme de produits SY/MAX du matériel ES. Il faut configurer un module déporté (140-CRP-93x-00) dans le châssis local.
E/S 800	Les E/S 800 nécessitent la présence d'un module déporté (140-CRP-93x-00) dans le châssis local.
E/S Modbus	Une passerelle virtuelle, connectée sous la forme d'E/S Ethernet et qui communique avec des esclaves Modbus (modules d'E/S complexes). Pour chaque châssis de passerelle, il faut configurer un module de communication Ethernet dans le châssis local.
Quantum ERIO	Un châssis qui est connecté en tant que Quantum Ethernet distant d'E/S. Ce châssis nécessite que le module de coupleur maître Ethernet (140-CRP-312-00) soit configuré dans le châssis local de l'API.

Commentaires sur les options du `type de liaison` pour Premium :

Option du type de liaison	Commentaire
XBus	Le châssis XBus est créé automatiquement en même temps que l'API. **Remarque :** les modules de communication ne sont autorisés que dans le XBus_Châssis 0 et son extension.
E/S MBP 0	Un châssis connecté sous la forme d'E/S Modbus Plus. Il est nécessaire de configurer un adaptateur Modbus Plus dans le châssis local.
E/S Ethernet	Un châssis connecté comme E/S Ethernet.
Périphérique Ethernet	Un châssis connecté à un périphérique configurable (châssis Advantys) via Ethernet.
E/S Modbus	Une passerelle virtuelle, connectée sous la forme d'E/S Ethernet et qui communique avec des esclaves Modbus (modules d'E/S complexes). Pour chaque châssis de passerelle, il faut configurer un module de communication Ethernet dans le châssis local.
E/S Profibus	Une passerelle virtuelle, connectée sous la forme d'E/S Ethernet et qui communique avec des esclaves Profibus. Pour chaque châssis de passerelle, il faut configurer un module de communication Ethernet dans le châssis local.

Commentaires sur les options du `type de liaison` pour M340 :

Option du type de liaison	Commentaire
E/S Ethernet	Un châssis connecté comme E/S Ethernet.
Périphérique Ethernet	Un châssis connecté à un périphérique configurable (châssis Advantys) via Ethernet.
E/S Modbus	Une passerelle virtuelle, connectée sous la forme d'E/S Ethernet et qui communique avec des esclaves Modbus (modules d'E/S complexes). Pour chaque châssis de passerelle, il faut configurer un module de communication Ethernet dans le châssis local.
E/S Profibus	Une passerelle virtuelle, connectée sous la forme d'E/S Ethernet et qui communique avec des esclaves Profibus. Pour chaque châssis de passerelle, il faut configurer un module de communication Ethernet dans le châssis local.

Commentaires sur les options du `type de liaison` pour M580 :

Option du type de liaison	Commentaire
Local	Le châssis local est créé automatiquement en même temps que l'API.
XBus	Un châssis connecté à un châssis local pour configurer des modules Premium (uniquements des modules d'E/S discrètes, analogiques et de comptage).
E/S Ethernet	Un châssis connecté comme E/S Ethernet.

Propriétés du module matériel

Les propriétés dépendent de la catégorie de module sélectionnée.

Propriétés des modules/cartes de communication ou UC :

Propriété	Description	Options	Commentaire
`Type de liaison`	Le type de liaison est défini par le système. Types de liaisons pour Quantum : • L'UC est en liaison 0 • Deux autres NOM sont en liaison 1 ou en liaison 2 Types de liaisons pour Momentum : • La carte de communication sur l'UC est en liaison 0	-	Cette propriété n'est disponible pour Quantum qu'avec Modbus Plus.
`Segment de réseau`	définit les segments auxquels est rattaché le module	A sélectionner dans la liste.	La liste dépend des définitions données dans la configuration du segment de réseau.
`Adresse Modbus Plus` ou `Adresse IP`	définit l'adresse du module	Adresse sur le réseau	Les adresses réseau valides dépendent du type de réseau.
`Bus enfant du bus distant`	définit le nombre de bus enfant dans le bus d'E/S	Nombre de bus enfant	Cette propriété n'est disponible que pour les modules Momentum BNO.

Tableau de points d'E/S pour modules d'E/S :

Propriété	Description	Options	Commentaire
E/S	affiche la position d'un point d'E/S dans un module d'E/S	Aucun	affichage seul
Type d'E/S	indique le type d'E/S	● E/S TOR ● E/S analogique	affichage seul
Variable	définit la variable affectée au point d'E/S	Toute variable du type de données adéquate	L'affectation peut être effectuée en faisant glisser les variables voulues. **Remarque :** pour changer l'association d'une variable à un point d'E/S, il est possible de faire glisser la variable au sein d'un module ou même entre modules.
Adresse	Affiche l'adresse du point d'E/S si la mémoire a déjà été affectée.	-	affichage seul
Etat sur erreur	définit l'action d'une sortie logique en cas de timeout	● Non_affecté(e) ● Non activé ● Dernière valeur ● Défini par l'utilisateur	Si un timeout se produit dans un module de sortie d'API, il est possible de programmer le signal de sortie de la manière suivante : ● Non activé : le signal est mis à 0. ● Dernière valeur : le signal conserve la dernière valeur connue avant l'événement de timeout. ● Défini par l'utilisateur : l'utilisateur définit l'état par défaut dans le cas où un timeout surviendrait.
Valeur sur erreur	Définit la valeur de la variable si un événement de timeout survient.	-	Disponible uniquement si l'état sur erreur est réglé sur "défini par l'utilisateur".

NOTE : L'état sur erreur du module doit être le même que pour les variables affectées au module.

Informations générales sur les bus de terrain

Introduction

Unity Application Generator gère la configuration des bus de terrain. La technologie de bus de terrain suivante sera gérée :

- Profibus

Vous pouvez pré-configurer des configurations et gérer tous les paramètres dans UAG. Vous avez besoin d'un outil tiers de chez Prosoft ou Hilscher pour obtenir la configuration.

Les outils nécessaires sont les suivants :

- ProSoft Configuration Builder
- Hilscher SyCon

ProSoft Configuration Builder

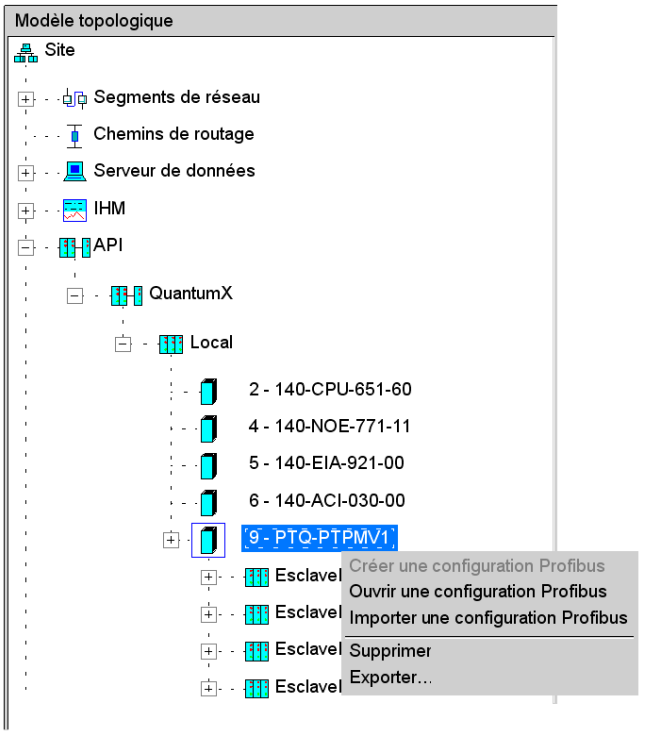
Vous devez utiliser le logiciel ProSoft Configuration Builder pour configurer le module Quantum Profibus.

Hilscher SyCon

Vous devez utiliser le logiciel Hilscher SyCon pour configurer le module Premium Profibus.

Exemple d'intégration Profibus

La figure suivante illustre un exemple de l'intégration Profibus dans le modèle topologique de UAG.



NOTE : Si une configuration Profibus existe déjà, l'option de menu **Créer une configuration Profibus** n'est pas active.

Les informations suivantes donnent uniquement les étapes générales permettant d'obtenir une configuration Profibus. Pour avoir des informations plus détaillées, voir aussi *Importation de la configuration de bus de terrain, page 361*.

Module Ethernet amélioré

Présentation

Il est possible de créer des modules Ethernet définis par l'utilisateur qui ne sont pas pris en charge par défaut dans UAG. Un châssis Ethernet fait partie d'un module de communication Ethernet (comme ETY 4103 ou NOE 771-xx). Après la création d'un châssis Ethernet, l'utilisateur peut affecter ces modules de communication Ethernet. Consultez également la section *Châssis supplémentaires, page 167*

Modules disponibles pour le châssis Ethernet amélioré

Les modules pouvant être intégrés dans le châssis Ethernet amélioré sont les suivants :

- toutes les bases Momentum disposant d'un module de communication Ethernet ;
- les modules définis par l'utilisateur, tels que l'Altivar 58 avec connexion Ethernet.

Modules spéciaux

Tous les modules Momentum communiquent (par lecture/écriture) sur l'adresse 4x00001. Cependant, d'autres équipements de communication comme, par exemple, l'automate SPANG Power Electronics, série 850 de type 853 0040 E1 00 utilise un décalage pour la communication.

Les registres et les décalages correspondants sont les suivants :

- Adresse de début de registre Ref. lecture 4:200, Longueur 15
- Adresse de début de registre Ref. d'écriture 4:200, Longueur 5

Il est toutefois possible de saisir les adresses de début de registre "Ref lecture esclave" et "Ref écriture esclave". UAG génère automatiquement ces adresses dans la table du scrutateur d'E/S.

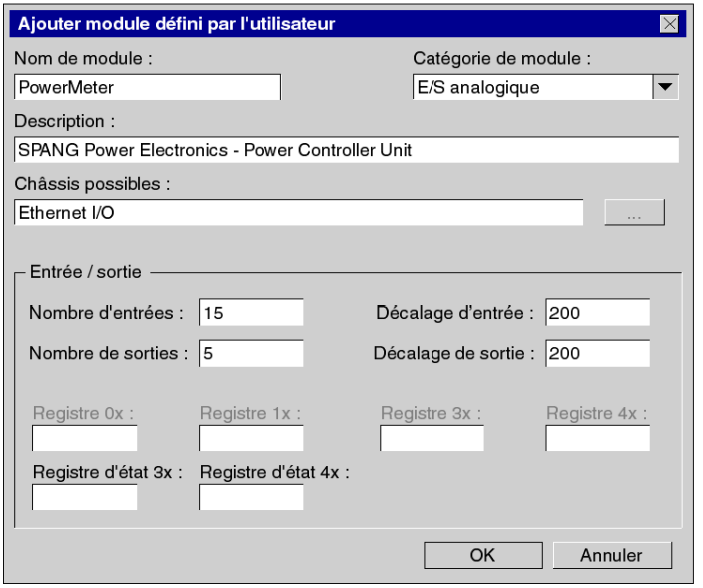
Décalage en entrée/sortie

Les décalages en entrée ("Ref. lecture esclave") et en sortie ("Ref. écriture esclave") peuvent être saisis dans l'éditeur de personnalisation. Ainsi, tous les modules sont générés avec les décalages configurés.

Personnalisation des modules Ethernet améliorés

Pour annoncer un module Ethernet amélioré dans Unity Application Generator, l'utilisateur doit le personnaliser dans la personnalisation de UAG en tant que module défini par l'utilisateur.

Le schéma suivant illustre un exemple de configuration d'un module Ethernet amélioré dans la personnalisation de UAG.

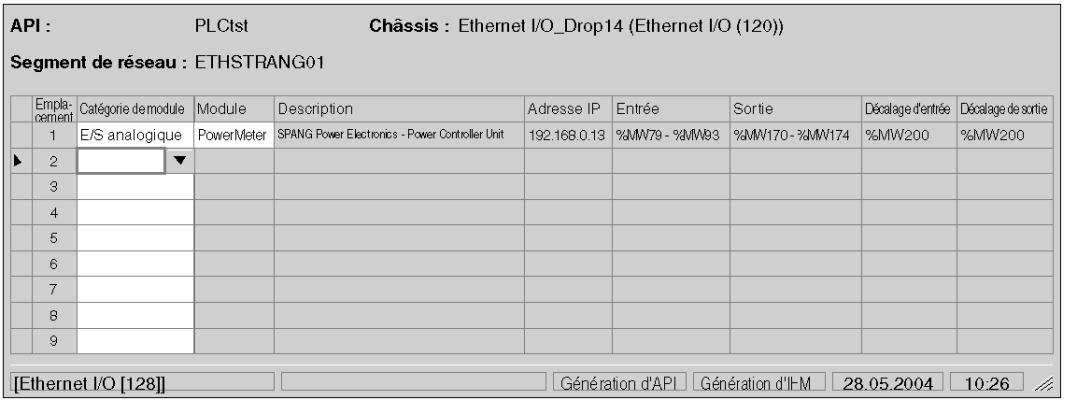


NOTE : Pour les modules Ethernet améliorés, il est possible de définir le **taux de répétition du scrutateur d'E/S** dans Unity Application Generator.

Modules Ethernet améliorés dans UAG

Une fois la personnalisation terminée, les modules peuvent être utilisés dans Unity Application Generator.

Le schéma suivant illustre un exemple de configuration d'un module Ethernet amélioré dans un autre châssis Ethernet.



Modules Ethernet améliorés dans Unity Pro

Le schéma suivant illustre un exemple de table de scrutateur d'E/S générée automatiquement (Unity Pro) :

	Adresse IP esclave	ID unité	Timeout d'état de santé (ms)	Taux répét. (ms)	Réf. lecture maître	Réf. lecture esclave	Longueur lecture	Dernière valeur (entrée)	Réf. écriture maître	Réf. écriture esclave	Lon-gueur écriture	Code diagn.	D
1	123.34.111.11	0	2000	0	100001	400200	15	Hold Last	000049	400200	5		Power
2	123.34.111.12	0	2000	0	100017	400001	1	Hold Last	000065	400001	1		TIO

Passerelle Modbus/Ethernet

Présentation

Dans UAG, les passerelles Modbus/Ethernet sont représentées sous forme de châssis.

Modules possibles pour la passerelle

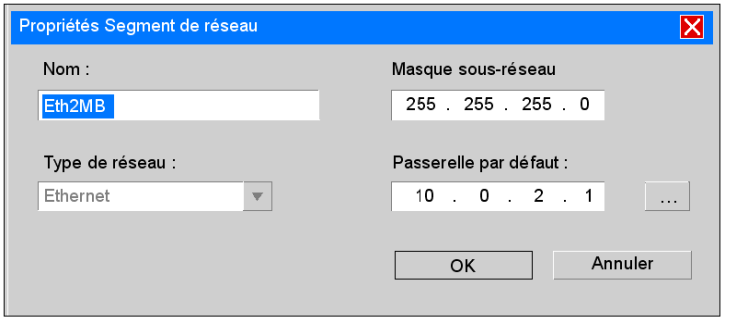
Les modules pouvant être intégrés dans les passerelles Modbus/Ethernet sont les suivants :

- équipements esclaves Schneider Electric Modbus (par exemple, ATV31) ;
  Ces modules sont proposés en tant que modules d'**E/S complexes** si vous avez activé l'un des châssis **Passerelle Modbus/Ethernet** dans la grille **Châssis** de la **personnalisation UAG**.
- modules d'**E/S complexes**, définis par l'utilisateur dans **Personnalisation UAG** →**API** → **Module défini par l'utilisateur**

NOTE : Pour tous les modules des passerelles Modbus/Ethernet, le **taux de répétition de scrutateur d'E/S** peut être défini dans UAG.

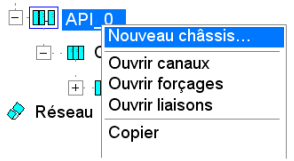
Conditions préalables

Pour utiliser la passerelle Modbus/Ethernet, vous devez configurer un réseau Ethernet.

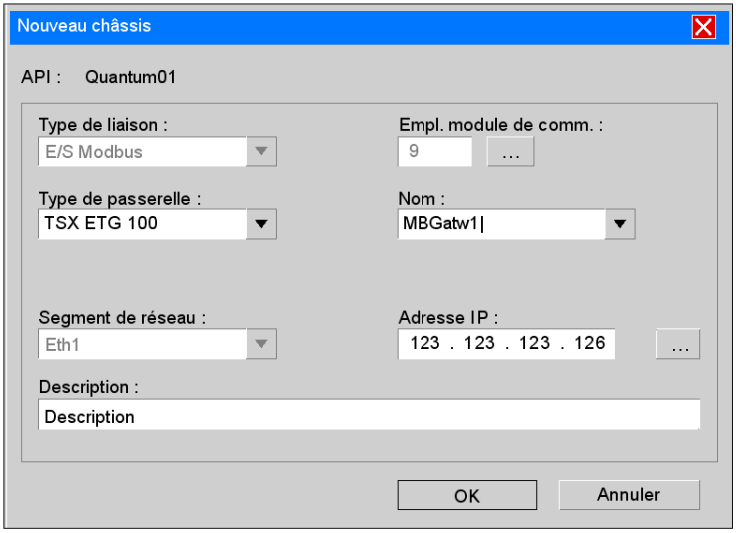


Boîte de dialogue Nouveau châssis

Dans le **modèle topologique**, sélectionnez **Nouveau châssis** dans le menu contextuel de l'API.



Dans la boîte de dialogue qui s'affiche, sélectionnez **E/S Modbus** pour le **Type de liaison**. La boîte de dialogue affichée a alors l'aspect suivant :

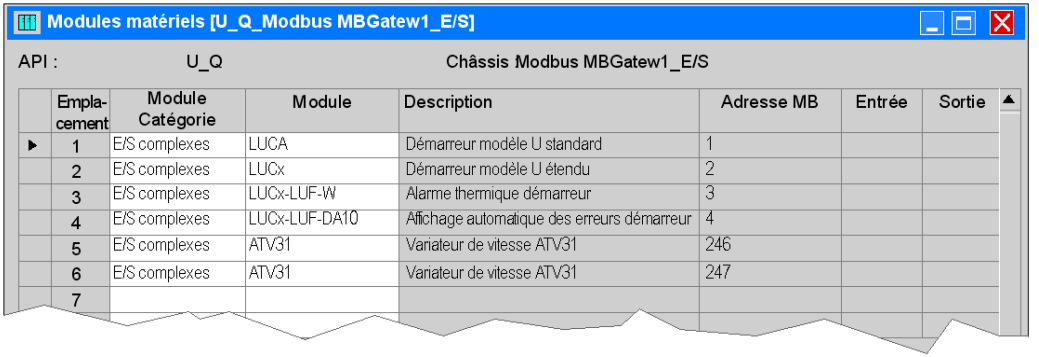


Elément	Description
Type de liaison	Sélectionnez E/S Modbus pour définir une passerelle Modbus/Ethernet.
Type de passerelle	Répertorie tous les types de passerelles présélectionnés dans **Personnalisation UAG**
Empl. module de comm.	Affiche l'emplacement du module de communication sélectionné
Nom	Saisissez un nom unique pour la passerelle Modbus/Ethernet. Le caractère unique de la passerelle est vérifié par UAG. Si des passerelles existent déjà pour le même segment de réseau, vous pouvez en sélectionner une dans la liste. Une passerelle choisie dans la liste sera partagée par plusieurs API.
Segment de réseau	Affiche le segment de réseau du module de communication réseau
Adresse IP	Adresse IP du segment de réseau
Description	Texte libre de description de la passerelle
OK	Confirmez vos saisies avec **OK**.
Annuler	Cliquez sur **Annuler** pour fermer la boîte de dialogue sans modification.

Pour modifier les propriétés d'une passerelle Modbus/Ethernet, sélectionnez la passerelle dans le **modèle topologique** et sélectionnez **Propriétés** dans son menu contextuel.

Boîte de dialogue Module matériel

Pour configurer le châssis de passerelle, sélectionnez-le dans le **modèle topologique** et sélectionnez **Ouvrir châssis** dans son menu contextuel.

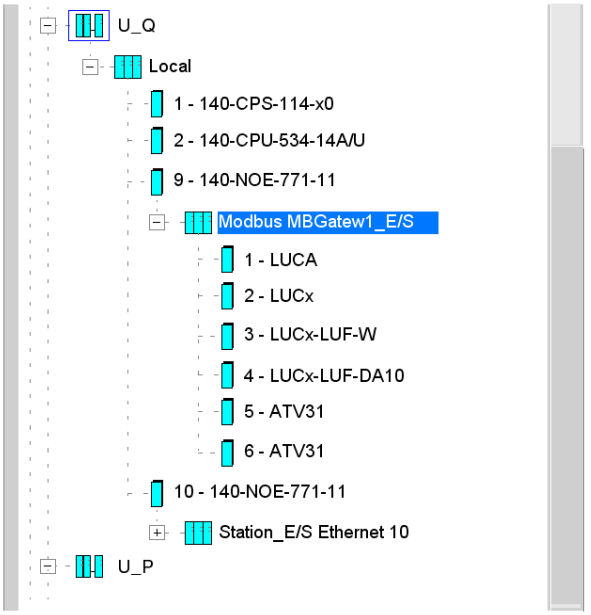


Elément	Description
Légende	Affiche le nom de l'API, le type de liaison et le nom de la passerelle (par exemple, U_Q_E/S Modbus_PassMB1)
Zone d'informations	Affiche ● Nom de l'API (par exemple, U_Q) ● Type de liaison (par exemple, E/S Modbus) ● Nom de la passerelle (par exemple, PassMB1)
Emplacement	Comporte 32 ou 64 emplacements pour ajouter des modules
Catégories de modules	Répertorie les catégories de modules prédéfinies dans **Personnalisation UAG**
Module	Répertorie les modules disponibles pour la catégorie de modules sélectionnée
Description	Description du module sélectionné
Adresse MB	Affiche l'adresse Modbus du module Pour affecter l'adresse MB à un module, cliquez avec le bouton droit de la souris dans la colonne **Adresse MB** du module et sélectionnez **Propriétés** dans le menu contextuel. La boîte de dialogue affichée permet de saisir l'**adresse MB** et le **taux de répétition de scrutateur d'E/S** (durée de scrutation des entrées/sorties).
Entrée **Sortie** **Décalage d'entrée** **Décalage de sortie**	Affichent les valeurs du module Double-cliquez ou cliquez avec le bouton droit pour ouvrir la boîte de dialogue **Points d'E/S** du module.

Pour les passerelles partagées entre plusieurs API, tous les modules sont affichés dans la grille **Modules matériels**. Les modules qui appartiennent à d'autres API sont affichés avec une couleur d'arrière-plan grise.

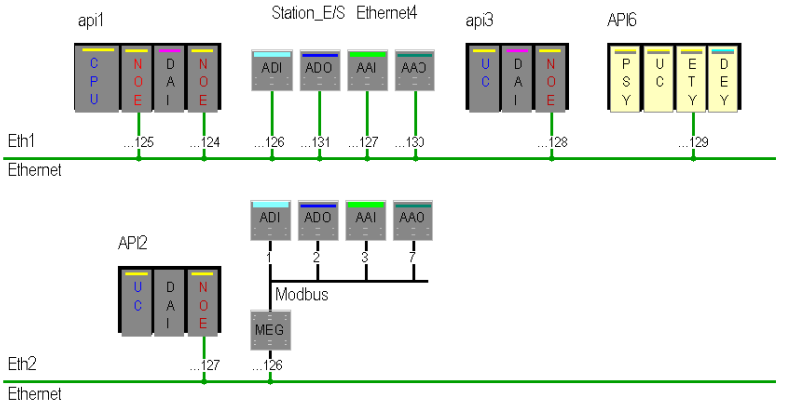
Passerelles dans le modèle topologique

Une passerelle Modbus/Ethernet apparaît dans le **modèle topologique** sous la forme d'un châssis comportant des modules ajoutés.



Passerelles dans l'afficheur topologique

Une représentation graphique d'une passerelle Modbus/Ethernet et des modules ajoutés connectés au réseau Modbus apparaît dans l'**afficheur topologique**.



Le réseau Modbus est affiché en noir.

Le même symbole (MEG) est utilisé pour toutes les passerelles.

Double-cliquez sur un symbole de passerelle pour ouvrir sa boîte de dialogue **Propriétés**.

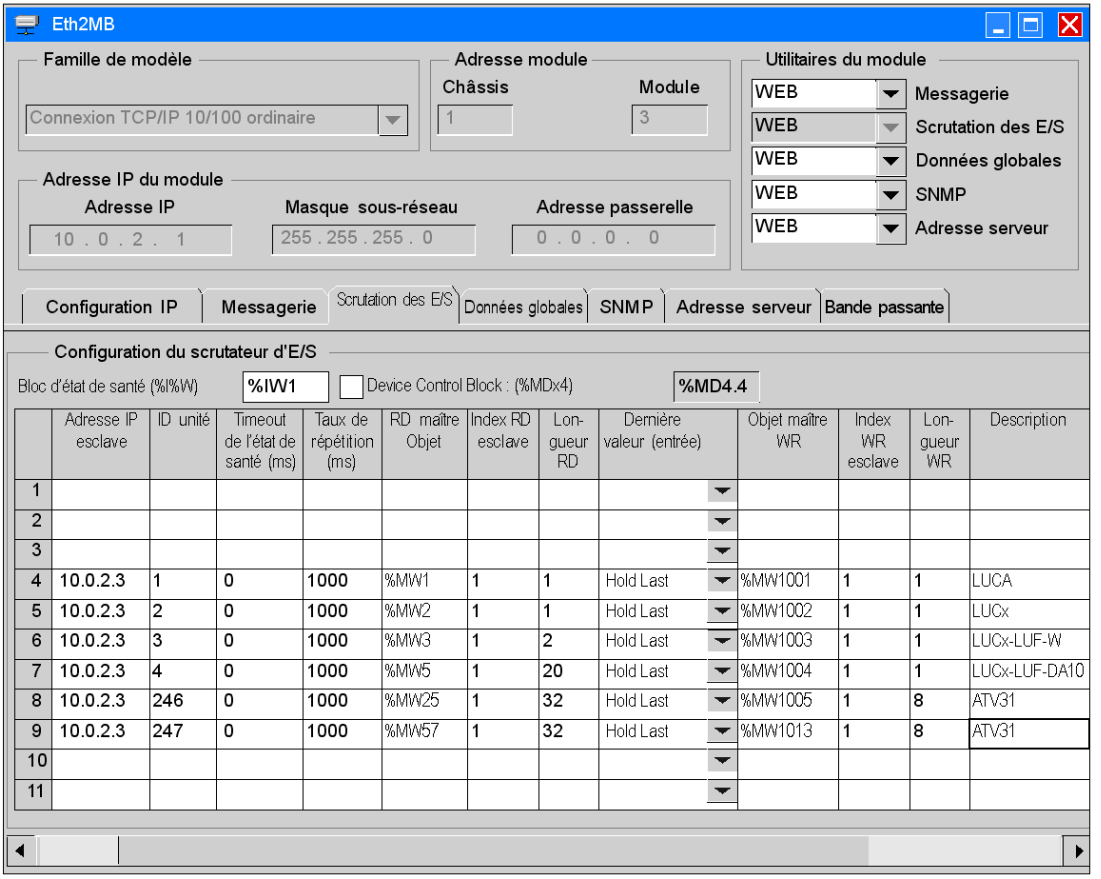
Une infobulle indique le nom, le type et l'adresse IP d'un module de passerelle.

Génération d'API pour passerelles Modbus/Ethernet

Le générateur de code UAG dérive des lignes du scrutateur d'E/S pour Unity Pro.

- La valeur indiquée dans la colonne **ID unité** correspond à l'**adresse Modbus** du module.
- Les valeurs **Timeout état de santé** et **Dernière valeur** sont fixées à leurs valeurs spécifiques Unity par défaut.
- Le **taux de répétition** est fixé à la valeur définie pour le module dans UAG.
- Les entrées de scrutateur d'E/S sont créées en fonctions des **zones d'E/S** définies pour les modules.

Lignes de scrutateur d'E/S générées dans UnityPro pour des équipements Modbus reliés à une passerelle Modbus/Ethernet



	Adresse IP esclave	ID unité	Timeout de l'état de santé (ms)	Taux de répétition (ms)	RD maître Objet	Index RD esclave	Longueur RD	Dernière valeur (entrée)	Objet maître WR	Index WR esclave	Longueur WR	Description
1												
2												
3												
4	10.0.2.3	1	0	1000	%MW1	1	1	Hold Last	%MW1001	1	1	LUCA
5	10.0.2.3	2	0	1000	%MW2	1	1	Hold Last	%MW1002	1	1	LUCx
6	10.0.2.3	3	0	1000	%MW3	1	2	Hold Last	%MW1003	1	1	LUCx-LUF-W
7	10.0.2.3	4	0	1000	%MW5	1	20	Hold Last	%MW1004	1	1	LUCx-LUF-DA10
8	10.0.2.3	246	0	1000	%MW25	1	32	Hold Last	%MW1005	1	8	ATV31
9	10.0.2.3	247	0	1000	%MW57	1	32	Hold Last	%MW1013	1	8	ATV31
10												
11												

NOTE : La configuration des passerelles Modbus doit être effectuée de manière spécifique pour ces équipements, avec leurs capacités de configuration particulières. L'exemple ci-dessus représente seulement les principes de communication. Les indices des esclaves, la longueur des données et les capacités Modbus ne sont que de simples exemples et ne correspondent pas au comportement réel du module.

Passerelle Profibus/Ethernet

Présentation

Dans UAG, les passerelles Profibus/Ethernet sont représentées sous forme de châssis (**Passerelle Profibus/Ethernet**).

UAG prend uniquement en charge un module de passerelle Profibus/Ethernet :

5240-MNET-PDPMV1

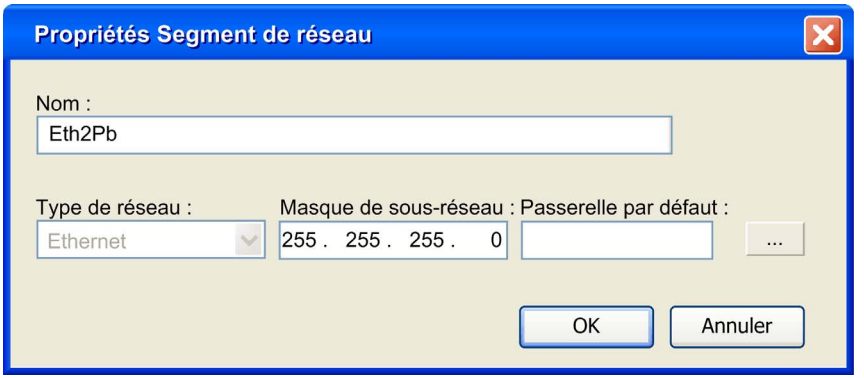
NOTE : Pour les modules des passerelles Profibus/Ethernet, le **taux de répétition du scrutateur d'E/S** peut être défini dans UAG.

Conditions préalables

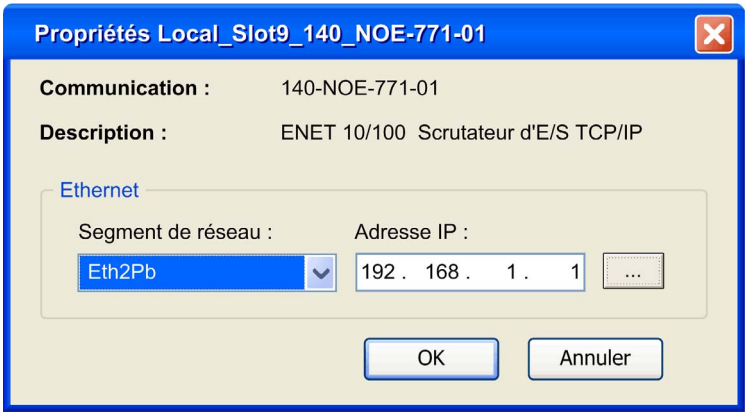
Pour utiliser la passerelle Profibus/Ethernet, vous devez au moins configurer :

- un réseau Ethernet
- un module de communication

Réseau Ethernet

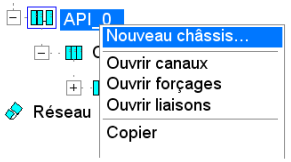


Module de communication

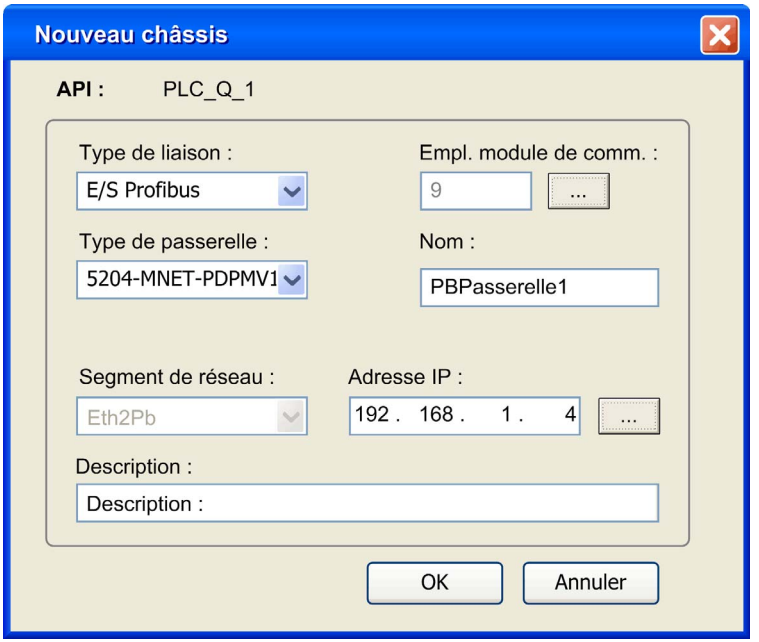


Boîte de dialogue Nouveau châssis

Dans le **modèle topologique**, sélectionnez **Nouveau châssis** dans le menu contextuel de l'API.



Dans la boîte de dialogue qui s'affiche, sélectionnez **E/S Profibus** pour le **Type de liaison**. La boîte de dialogue affichée a alors l'aspect suivant :



Elément	Description
Type de liaison	Sélectionnez **E/S Profibus** pour définir une passerelle Profibus/Ethernet.
Type de passerelle	**5240-MNET-PDPMV1**
Empl. module de comm.	Affiche l'emplacement du module de communication sélectionné
Nom	Saisissez un nom unique pour la passerelle Profibus/Ethernet. Le caractère unique de la passerelle est vérifié par UAG.
Segment de réseau	Affiche le segment de réseau du module de communication réseau
Adresse IP	Adresse IP de la passerelle Profibus/Ethernet
Description	Texte libre de description de la passerelle
OK	Confirmez vos saisies avec **OK**.
Annuler	Cliquez sur **Annuler** pour fermer la boîte de dialogue sans modification.

Après avoir cliqué sur **OK**, le message suivant apparaît :

Boîte de dialogue Propriétés

Pour modifier les propriétés d'une passerelle Profibus/Ethernet, sélectionnez la passerelle dans le **modèle topologique** et sélectionnez **Propriétés** dans son menu contextuel.

La légende de la boîte de dialogue **Propriétés** s'affiche

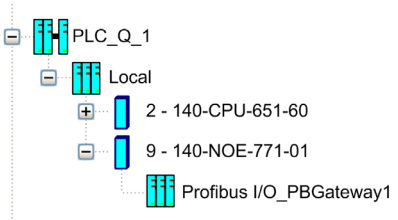
<PLC Name>_<Link Type>_<Gateway Name>.

Vous pouvez modifier :

- **Empl. module de comm.**
- **Type de passerelle**
- **Nom de passerelle**
- **Adresse IP**
- **Description**

Passerelles dans le modèle topologique

La passerelle Profibus/Ethernet s'affiche dans le **modèle topologique** en tant que châssis d'E/S Profibus sous le module de communication correspondant.

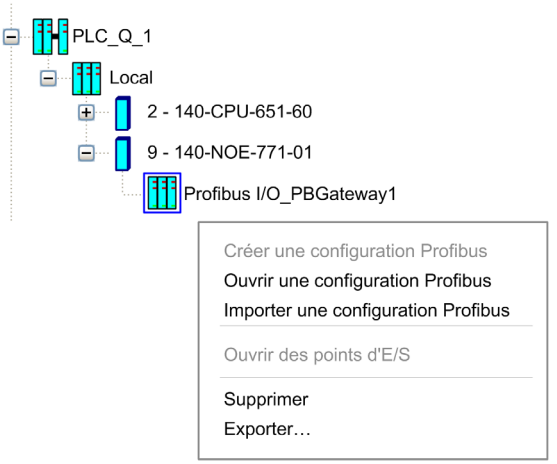


Créer une configuration Profibus

Cliquez avec le bouton droit droit de la souris sur le châssis d'E/S Profibus dans le **modèle topologique** puis sélectionnez **Créer une configuration Profibus** dans le menu contextuel.

Un fichier de configuration ProSoft (avec des paramètres par défaut) est créé sous *<UAGProjectPath>\ProfibusMasterGatewayConfiguration_<PLC Name>_< RackID>*.

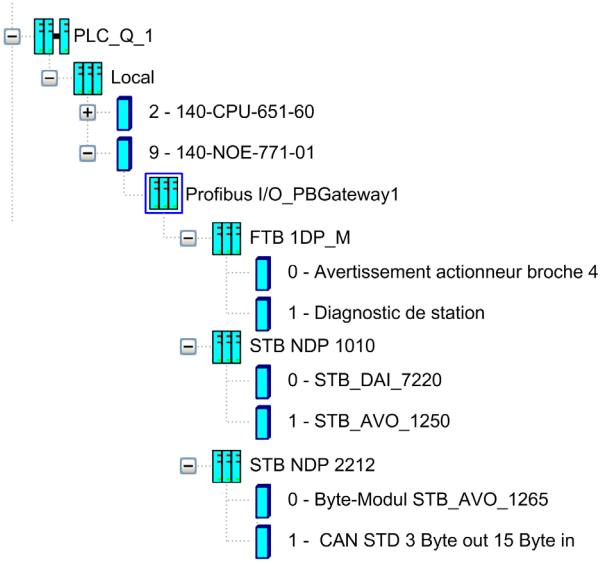
Dans le menu contextuel du châssis d'E/S Profibus, **Ouvrir une configuration Profibus** s'active.



Importer une configuration Profibus

Une fois les périphériques esclaves utilisant l'outil de configuration ProSoft configurés, vous pouvez importer le fichier dans UAG en sélectionnant **Importer une configuration Profibus** dans le menu contextuel.

Après l'importation, le **modèle topologique** aura l'aspect suivant :

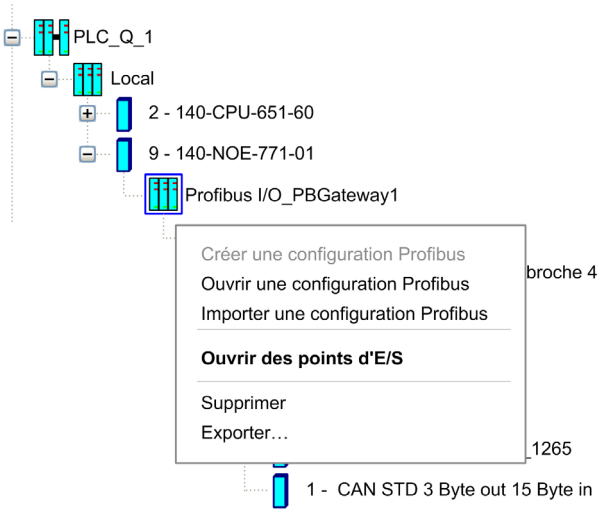


Importer sans créer de configuration Profibus

Si vous cliquez directement sur **Importer une configuration Profibus** sans préalablement **Créer une configuration Profibus**, la boîte de dialogue **Sélectionner un fichier de configuration** s'affiche. Elle vous permet de sélectionner un fichier de configuration Profibus existant à importer.

Ouvrir des points d’E/S

Après avoir correctement importé un fichier de configuration, **Ouvrir des points d’E/S** s’active dans le menu contextuel du châssis d’E/S Profibus.

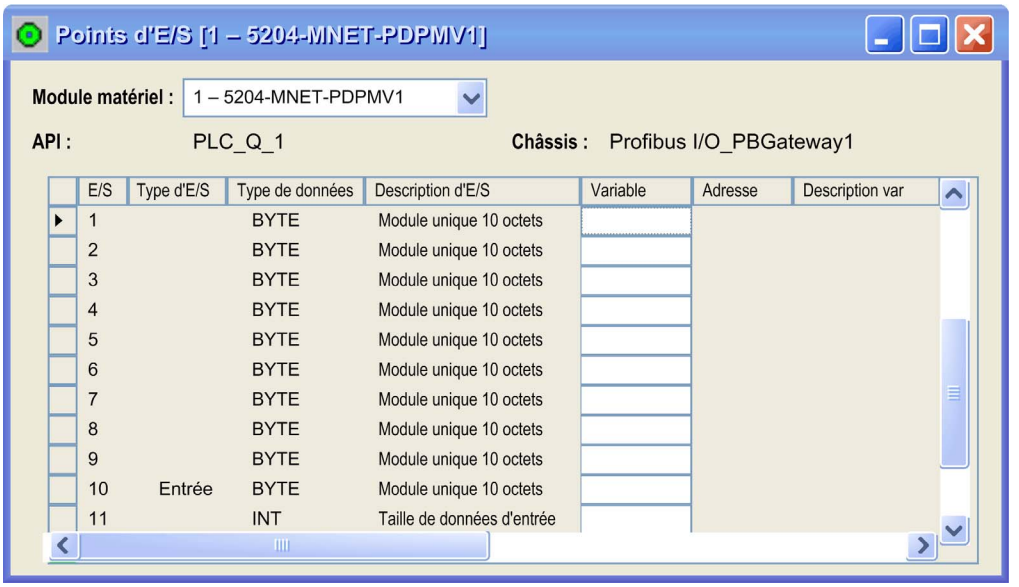


NOTE : Ouvrir des points d’E/S est uniquement activé si le **Registre des données d’état** du fichier de configuration Profibus prend une valeur (c.-à-d. sa valeur est différente de -1).

Boîte de dialogue Points d'E/S

Cliquez sur **Ouvrir des points d'E/S** dans le menu contextuel d'un module Profibus pour ouvrir la boîte de dialogue **Points d'E/S**.

Dans le module maître Profibus, vous pouvez atteindre les données d'état via les points d'E/S.



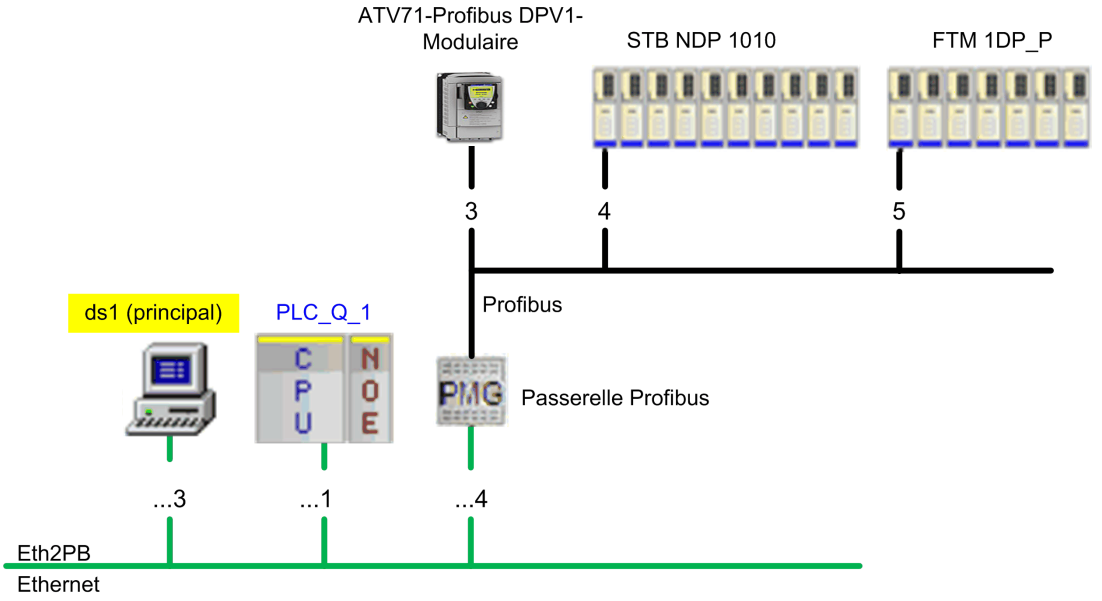
Copier/Coller

Dans le **modèle topologique**, vous pouvez copier et coller des API et des châssis.

Lors de la copie et du collage, seul le châssis d'E/S Profibus est créé (pas l'ensemble de la configuration).

Passerelles dans l'afficheur topologique

Une représentation graphique d'une passerelle Profibus/Ethernet et des modules ajoutés connectés au réseau Profibus apparaît dans l'**afficheur topologique**.

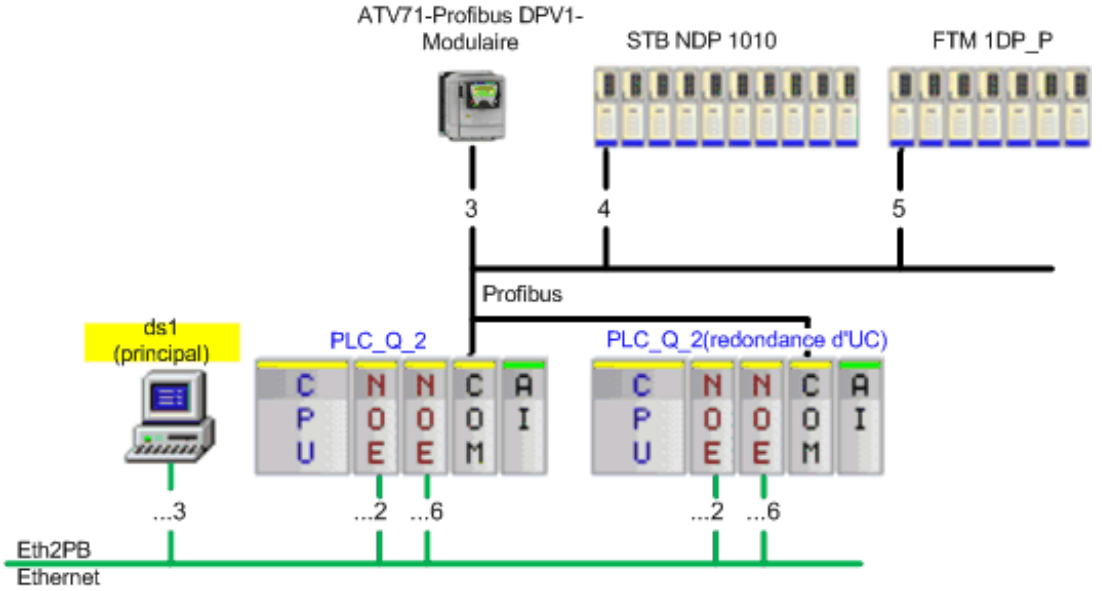


Le même symbole (PMG) est utilisé pour toutes les passerelles.

Double-cliquez sur un symbole de passerelle pour ouvrir sa boîte de dialogue **Propriétés**.

Une infobulle indique le nom, le type et l'adresse IP d'un module de passerelle.

Les périphériques Profibus liés à un module maître Profibus (PTQ-PDP-MV1 dans les API Quantum et TSX PBY 100 dans les API Premium) s'affichent également dans l'**afficheur topologique**.



NOTE : Les adresses des bus de terrain des périphériques Profibus n'étant pas disponibles dans des projets existants, ces adresses seront vides pour un projet existant, à moins que vous n'importiez de nouveau les périphériques Profibus.

Génération d'API pour passerelles Profibus/Ethernet

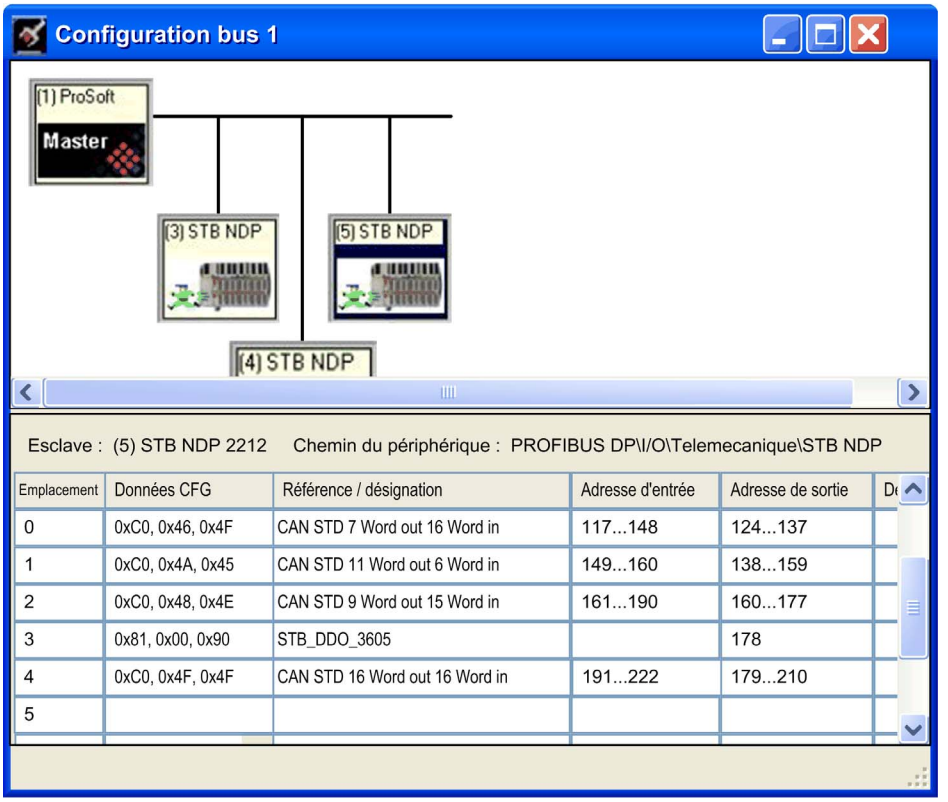
Le générateur de code UAG dérive des lignes du scrutateur d'E/S pour Unity Pro.

- L'**adresse IP esclave** est l'adresse IP de la passerelle.
- Les valeurs **ID unité**, **Timeout état de santé** et **Dernière valeur** sont fixées à leurs valeurs par défaut spécifiques à Unity.
- Le **taux de répétition** est réglé sur 100.
- Les valeurs **Index RD esclave** et **Index WR esclave** sont extraites de la configuration Profibus.

Une variable est lue ou écrite dans une seule entrée du scrutateur d'E/S.

Les données lues d'un module sont situées sous la ligne du scrutateur d'E/S et les données écrites de ce module se trouvent sous la même ligne du scrutateur d'E/S.

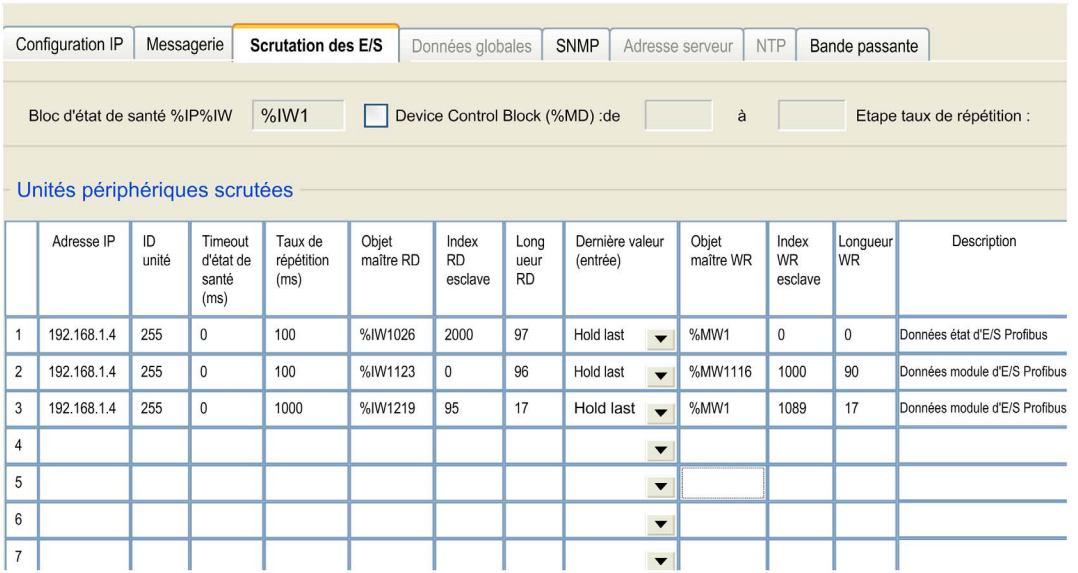
Exemple (adresses en octets) :



Le module **CAN STD16 Word out 16 Word in** de l'exemple ci-dessus ne peut pas être généré dans la même ligne du scrutateur d'E/S avec d'autres modules car leurs plages d'adresses (entrée comme sortie) dépassent 200 octets (étant donné que la longueur maximale de lecture ou d'écriture de chaque ligne du scrutateur d'E/S est de 100 mots, soit 200 octets).

C'est ainsi que dans le cas du module, deux lignes du scrutateur d'E/S sont générées.

Veuillez voir les deux dernières lignes du scrutateur d'E/S dans le graphique ci-dessous (la première ligne est destinée aux données d'état).



	Adresse IP	ID unité	Timeout d'état de santé (ms)	Taux de répétition (ms)	Objet maître RD	Index RD esclave	Long ueur RD	Dernière valeur (entrée)	Objet maître WR	Index WR esclave	Longueur WR	Description
1	192.168.1.4	255	0	100	%IW1026	2000	97	Hold last	%MW1	0	0	Données état d'E/S Profibus
2	192.168.1.4	255	0	100	%IW1123	0	96	Hold last	%MW1116	1000	90	Données module d'E/S Profibus
3	192.168.1.4	255	0	1000	%IW1219	95	17	Hold last	%MW1	1089	17	Données module d'E/S Profibus
4												
5												
6												
7												

NOTE : [Le générateur d'API ne génère pas les lignes du scrutateur d'E/S pour toutes les tailles de données d'entrée et de sortie. Il ne crée que les lignes du scrutateur d'E/S destinées aux données de module et d'état.]

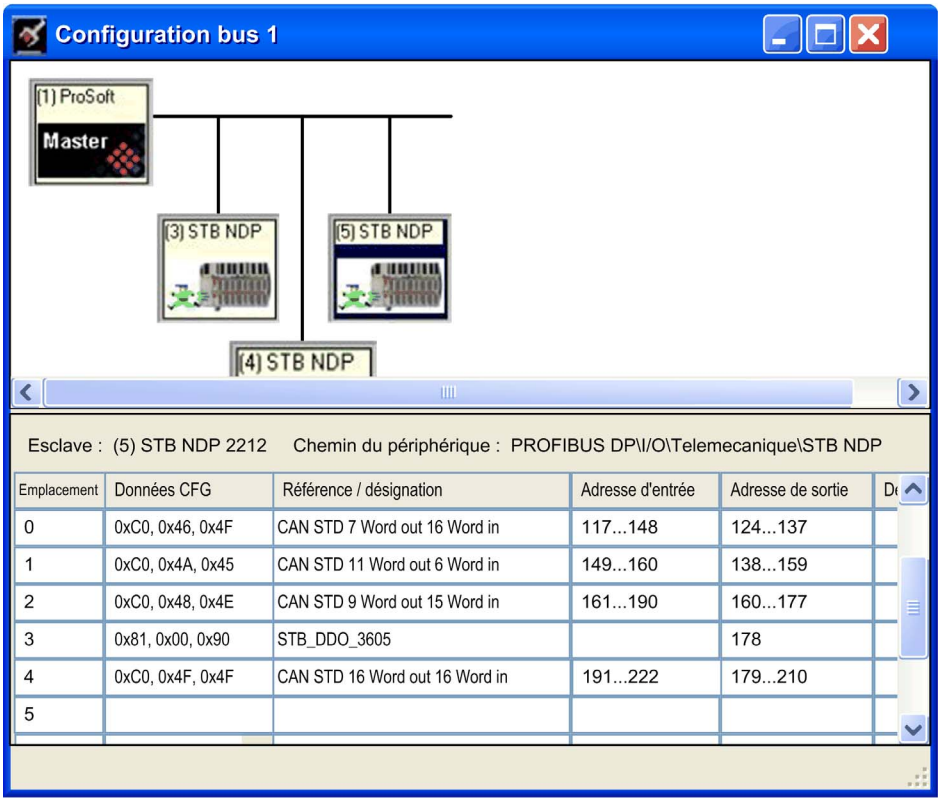
Importer un fichier de configuration Profibus (étude de cas)

Après avoir configuré les périphériques esclaves à l'aide de l'outil de configuration ProSoft, vous pouvez importer le fichier dans UAG.

Lors de la génération d'API, UAG génère des lignes du scrutateur d'E/S pour lire ou écrire des données à depuis/vers des registres Profibus.

Etant donné que le scrutateur d'E/S utilise une longueur de mot pour lire ou écrire des données, il peut arriver que des données de sortie Profibus deviennent incohérentes et UAG/Unity Pro n'est alors pas en mesure de les gérer.

Cas 1



Puisque l'adresse de fin de sortie du module **CAN STD 16 Word out 16 Word in** de l'exemple ci-dessus dépasse 200 octets, ses données de sortie doivent être écrites sur une seconde ligne du scrutateur d'E/S. Mais la première ligne du scrutateur d'E/S doit être écrite avec un nombre pair d'octets. Ainsi, le 179ème octet (**CAN STD 16 Word out 16 Word in**) est écrit lors de la première écriture du scrutateur d'E/S et le 178ème octet (**STB_DDO_3605**) est écrit lors de la seconde écriture du scrutateur d'E/S.

Cela entraîne une incohérence des données dans les registres Profibus.

Dans ce cas, lors de l'importation du fichier de configuration Profibus, UAG affiche un message d'information et l'importation n'a pas lieu.

Configuration d'importation

Type	Information
I	Début : 02/12/2009 16:39:15 +0530
E	L'adresse de début de la sortie [179] du module "CAN STD 16 Word out 16 Word in" peut entraîner une incohérence de données lors de l'écriture du scrutateur d'E/S. Veuillez modifier l'adresse dans le fichier de configuration [adresse suggérée = 180].
I	Terminé : avertissements : 0 erreurs : 1
I	Fin : 02/12/2009 16:39:15 +0530
I	Temps écoulé : 0 s

Cas 2

Pour l'exemple suivant, il n'est pas question d'incohérence d'écriture des données.

Esclave : (5) STB NDP 2212 Chemin du périphérique : PROFIBUS DP\I/O\Telemecanique\STB NDP

Emplacement	Données CFG	Référence / désignation	Adresse d'entrée	Adresse de sortie	D
0	0xC0, 0x46, 0x4F	CAN STD 7 Word out 16 Word in	117...148	124...137	
1	0xC0, 0x4A, 0x45	CAN STD 11 Word out 6 Word in	149...160	138...159	
2	0xC0, 0x48, 0x4E	CAN STD 9 Word out 15 Word in	161...190	160...177	
3	0x81, 0x00, 0x90	STB_DDO_3605		178	
4	0xC0, 0x4F, 0x4F	CAN STD 16 Word out 16 Word in	191...222	181...212	
5					

Etant donné qu'il existe un espace entre l'adresse de fin de sortie du module **STB_DDO_3605** et l'adresse de début de sortie du module **CAN STD 16 Word out 16 Word in** et qu'aucun autre module n'utilise les adresses de sortie du 179ème et du 180ème octet, la première ligne du scrutateur d'E/S écrit des données jusqu'au 179ème octet tout comme la seconde ligne du scrutateur d'E/S qui écrit des données à partir du 180ème octet.

Ainsi, lors de l'importation de ce fichier de configuration, UAG n'affiche pas de message d'information et l'importation a lieu.

Fichier XML d'importation et d'exportation

La configuration de la passerelle et les modules d'E/S configurés dans le châssis sont importés/exportés vers/depuis un projet UAG en utilisant le format XML.

```
- <rack name="PBGateway1" type="5204-MNET-PDPMV1" description="Descr" linkType="Profibus IO"
    dropNumber="9" configurationFilePath="ProfibusConfiguration_PLC_Q_1_Slot9_17\1_ID17.ppf">
  - <networkConnections>
      <networkConnection index="1" networkSegment="Eth2PB" address="192.168.1.4" />
    </networkConnections>
  </rack>
- <rack name="FTM 1DP_P" description="Descr" linkType="Fieldbus" fieldbusMasterGateway ="PBGateway1"
    fieldbusAddress="3">
  - <hwModules>
    - <hwModule type="042173 FTM 1DE16C12" moduleCategory="Profibus" slotNumber="1"
        addressPatterInput="%IW<MasterHwm.StartAddress3x + (hwm.ByteOffsetInput \ 2)>"
        addressPatterOutput="%MW<MasterHwm.StartAddress4x + (hwm.ByteOffsetOutput \ 2)>"
        addressingSchema="bytewise" bytesInput="2" bytesOffsetInput="465" bytesOutput="0"
        bytesOffsetOutput="0" fbInput="InputFBI" fbOutput="OutputFBI">
      - <ioPoints>
          <ioPoint type="IN" position="1" variableParentPath="ar01\pc01\un01\eq01\cm01"
            variableFullName="eq01_cm_01_varByte" description="" byteOrder="1" dataType="byte"
            bitNumber="-1" />
        </ioPoints>
      </hwModule>
    </hwModules>
  </rack>
```

Pour obtenir des informations détaillées, consultez la section Format des fichiers XML d'importation et d'exportation *(voir page 483)*.

NOTE : Les données du châssis esclave et des modules sont fournies à titre d'information uniquement. Lors de l'importation, les données réelles sont importées du fichier de configuration, seuls les points d'E/S liés aux variables sont importés.

Génération d'IHM

Seule une IHM générique génère la passerelle Profibus/Ethernet ainsi que les informations sur ses périphériques.

Cartographie mémoire

Pour les API Quantum, la cartographie mémoire utilise une plage de mémoire de **bus de terrain** pour cartographier des modules d'E/S esclaves.

Pour les API Premium ou Mirano, elle utilise une plage de mémoire de **Communication API**.

Ethernet Quantum distant d'E/S (QERIO)

Introduction

UAG prend en charge la caractéristique Ethernet Quantum distant d'E/S à partir de la version v3.3 SP2 et ultérieures.

Configuration du coupleur maître 140-CRP-312-00

Le coupleur maître **140-CRP-312-00** est disponible dans la **catégorie de module** des modules matériel sur un châssis local d'un API Quantum pour configuration.

Vous pouvez sélectionner le coupleur maître **140-CRP-312-00** pour :

- tous les UC Quantum (140-CPU-651-50, 140-CPU-651-60, 140-CPU-652-60, 140-CPU-671-60 et 140-CPU-672-61).
- Les versions de micrologiciel de l'UC supérieures ou égales à 3.00.

Pour configurer l'adresse IP du coupleur maître **140-CRP-312-00**, cliquez avec le bouton droit de la souris sur le module dans le châssis local et sélectionnez **Propriétés**. La boîte de dialogue Propriétés du coupleur maître s'affiche et vous permet de sélectionner le **Segment de réseau** et l'**adresse IP**. Vous ne pouvez configurer qu'une seule adresse IP par module de coupleur maître de CRP.

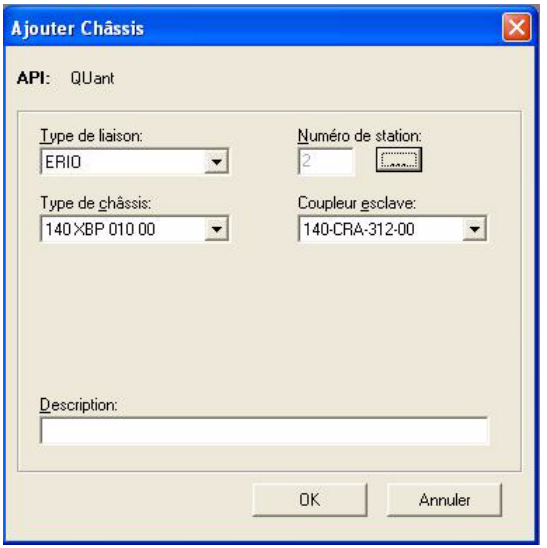
En cas de redondance d'UC Quantum, l'adresse IP suivante est attribuée à l'UC secondaire.

Pour les modes autonome et redondant, les 31 adresses IP suivantes sont réservées pour les stations distantes.

Configuration du coupleur esclave 140-CRA-312-00

Après la configuration du module de coupleur maître **140-CRA-312-00** dans le châssis local pour un API Quantum, créez un coupleur de type liaison ERIO.

La figure suivante illustre la création d'un nouveau châssis ERIO :



Lorsque le type de liaison **ERIO** est sélectionné, le coupleur esclave est automatiquement paramétré sur **140-CRA-312-00**. Le nombre total de stations autorisé est 31 (1... 31). Vous ne pouvez sélectionner que le numéro de station qui ne soit pas déjà utilisé.

Les types suivants de châssis sont autorisés pour les stations ERIO :

- 140 XBP 002 00
- 140 XBP 003 00
- 140 XBP 004 00
- 140 XBP 006 00
- 140 XBP 010 00
- 140 XBP 016 00

Les stations ERIO figurent dans l'arborescence **modèle topologique** sous le nœud API portant la convention de nommage **ERIO-Dropx** (x: numéro de station). Vous ne pouvez pas modifier le **type de liaison** ni le **coupleur esclave** d'une station ERIO dès lors qu'ils sont créés. Vous pouvez cependant modifier la taille de châssis et le **numéro de station** d'une station ERIO à tout moment.

Par défaut, toutes les stations ERIO ont le coupleur esclave **140-CRA-312-00** configuré dans le premier emplacement du châssis.

Les stations ERIO prennent en charge les catégories suivantes de modules :

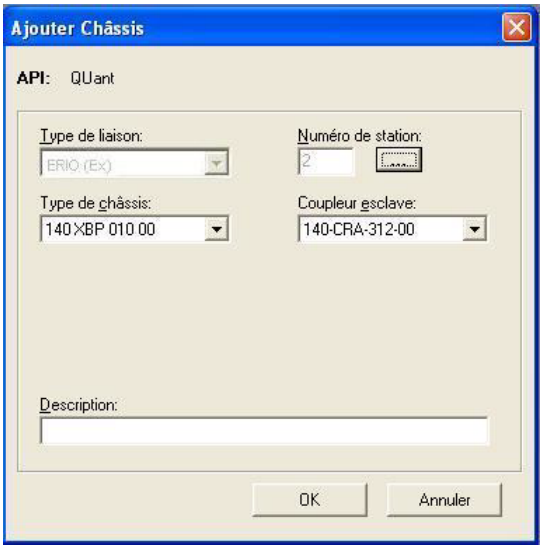
- Module d'alimentation
- Analogique
- TOR
- Extension
- Expert

Configuration d'un châssis d'extension pour stations

Vous pouvez créer des châssis d'extension pour toute station ERIO si un module d'extension est sélectionné dans le châssis primaire.

Le numéro de station d'un châssis d'extension reste le même car le numéro de station d'un châssis parent et le type de liaison sont configurés automatiquement au Quantum ERIO.

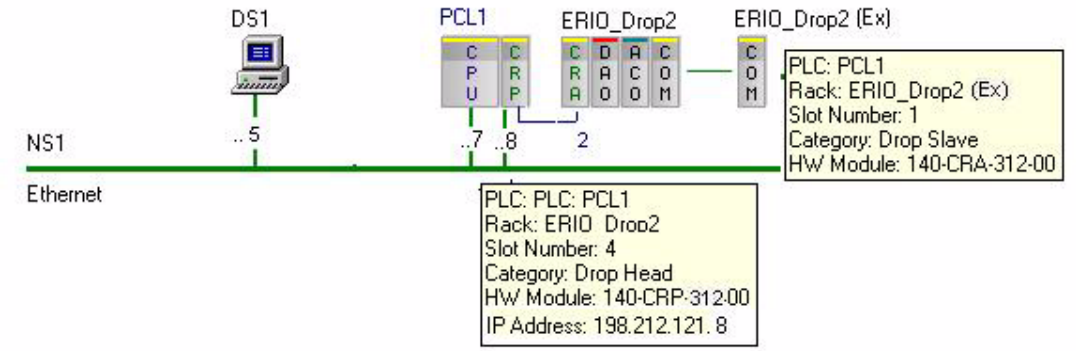
La figure suivante illustre la configuration d'un nouveau châssis d'extension ERIO :



Afficheur topologique UAG

L'afficheur topologique UAG présente les propriétés des modules CRP et CRA.

La figure suivante affiche le châssis ERIO dans l'affichage topologique d'UAG :



Copier/Coller d'un ERIO

La station QERIO prend en charge la caractéristique copier/coller, qui est autorisée dans les cas suivants :

- dans le même API ou entre différents API équipés de la version 3.0 ou ultérieure du micrologiciel de l'UC ;
- processeurs (140-CPU-651-50, 140-CPU-651-60, 140-CPU-652-60, 140-CPU-671-60 et 140-CPU-672-61) ; et
- le châssis local doté d'un coupleur maître CRP.

Le copier/coller des stations d'extension ERIO est autorisé dans un même API ou entre API (uniquement si un châssis d'extension est disponible sur un premier châssis et un coupleur maître ERIO sur un châssis local).

Exportation et importation d'un ERIO

Vous pouvez exporter l'API doté du coupleur maître 140-CRP-312-00 et du coupleur esclave 140-CRA-312-00, de même que l'ensemble des modules matériel et leurs points d'E/S configurés sur ses châssis primaires et d'extension.

UAG prend en charge la caractéristique d'importation de l'ensemble des modules QERIO.

NOTE : UAG ne prend pas en charge l'exportation et l'importation de QERIO en format CSV.

E/S M580 Ethernet

Introduction

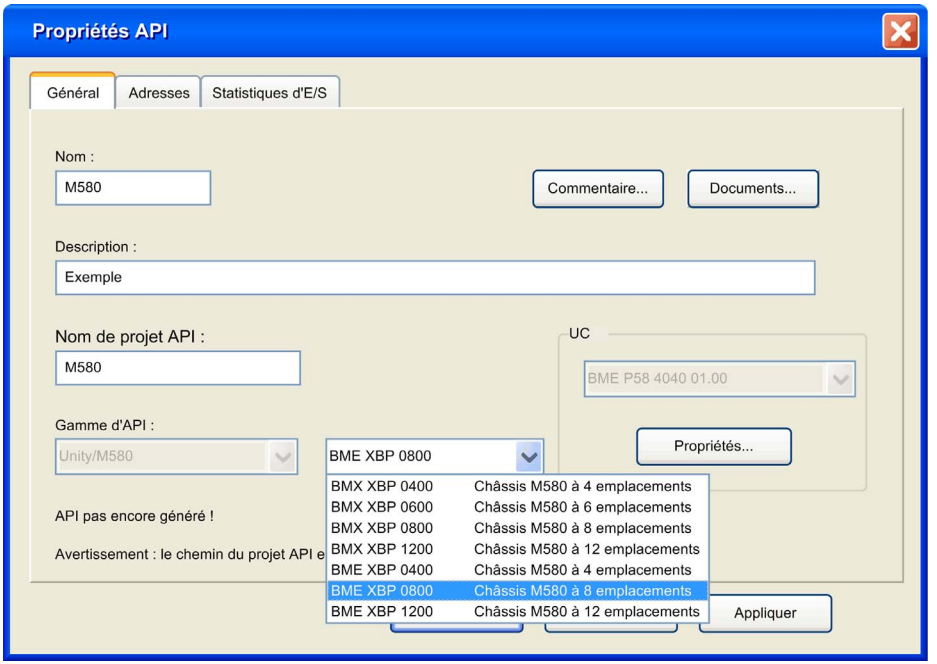
UAG prend en charge les API M580 pour la création de stations d'E/S Ethernet directement similaires à tous les châssis étendus ayant la version 3.3 SP4 ou une version ultérieure.

Configuration de châssis locaux

Les châssis suivants sont pris en charge pour un API M580 :

- Châssis BMX
  - BMX XBP 0400
  - BMX XBP 0600
  - BMX XBP 0800
  - BMX XBP 1200
- Châssis BME
  - BME XBP 0400
  - BME XBP 0800
  - BME XBP 1200

La figure ci-dessous affiche les châssis locaux pris en charge par un API M580 :



Configuration de stations locales

Les châssis suivants sont pris en charge pour une station locale :

- Châssis BMX
  - BMX XBP 0400
  - BMX XBP 0600
  - BMX XBP 0800
  - BMX XBP 1200
- Châssis BME
  - BME XBP 0400
  - BME XBP 0800
  - BME XBP 1200
- Châssis Premium
  - TSX RKY 4EX
  - TSX RKY 6EX
  - TSX RKY 8EX
  - TSX RKY 12EX

Configuration d'une tête de station

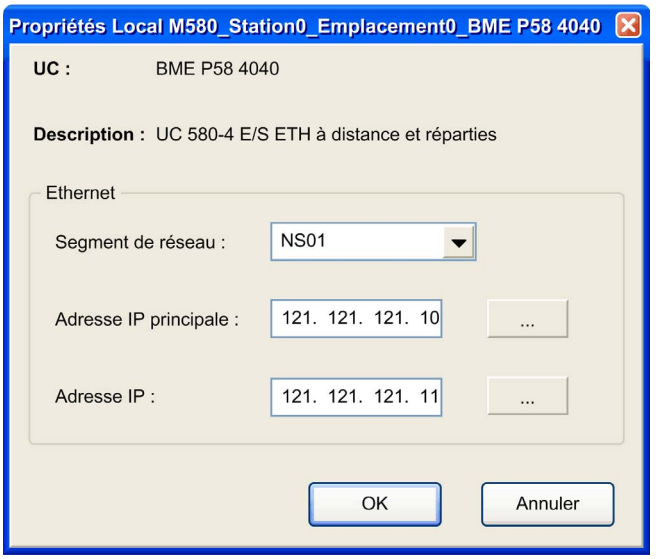
La gamme d'API M580 dispose d'une tête de station d'E/S Ethernet intégrée. Ainsi, vous pouvez créer des stations d'E/S Ethernet directement, sans avoir de tête externe configurée sur le châssis local.

Les UC suivants, de la gamme d'API M580, permettent de créer des stations d'E/S Ethernet M580 X80 :

- BME P58 2040 - Le nombre de stations pouvant être attribuées est de 8.
- BME P58 3040 - Le nombre de stations pouvant être attribuées est de 16.
- BME P58 4040 - Le nombre de stations pouvant être attribuées est de 16.

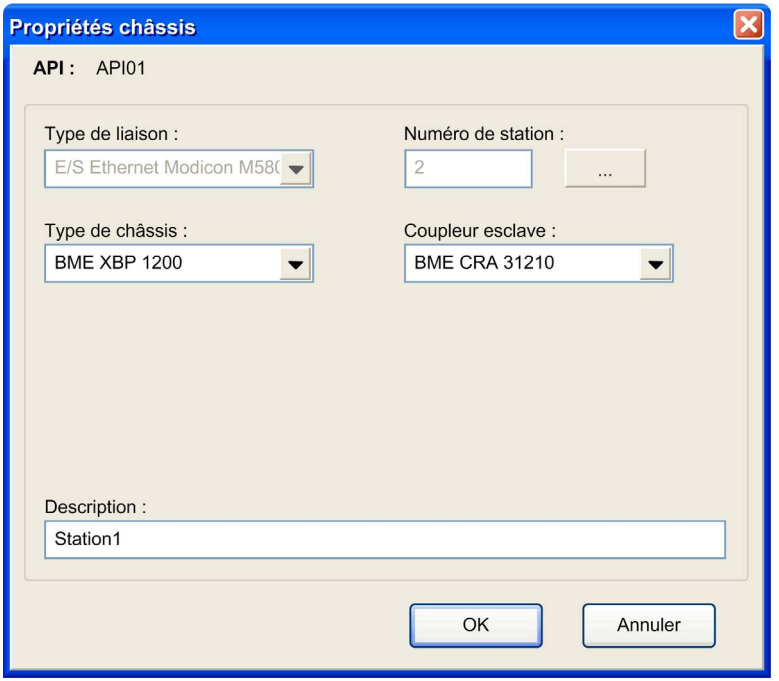
Lorsque vous créez un API M580, l'adresse IP attribuée à la tête de station réserve automatiquement les 8 ou 16 prochaines adresses IP, en fonction du type d'UC. Vous pouvez le voir dans l'affichage topologique.

La boîte de dialogue **Propriétés** suivante de l'API vous permet de sélectionner le **Segment de réseau** et l'**Adresse IP** pour le nouvel API sur M580 :



Configuration d’un coupleur esclave

La figure suivante illustre la création d’un nouveau châssis d’E/S Ethernet Modicon M580 :



Les types suivants de châssis sont autorisés pour les stations d’E/S Ethernet Modicon M580 :

- Châssis BMX
  - BMX XBP 0400
  - BMX XBP 0600
  - BMX XBP 0800
  - BMX XBP 1200
- Châssis BME
  - BME XBP 0400
  - BME XBP 0800
  - BME XBP 1200

Par exemple, si vous créez l’API M580 à l’aide de l’UC BME P58 3040, alors un nouveau type de liaison d’**E/S Ethernet Modicon M580** devient disponible pour créer un châssis. Ce châssis peut configurer un communicateur de fin de station **BME CRA 312 10**, **BMX CRA 312 00**, ou **BMX CRA 312 10**.

Les stations sont créées en fonction des coupleurs esclaves sélectionnés (BMX CRA 312 00, BMX CRA 312 10 ou BME CRA 312 10). Après la création d’un châssis d’E/S Ethernet Modicon, le communicateur de fin de station BMX ou BME occupe le numéro d’emplacement 0, par défaut.

Par défaut, les deux premiers emplacements sont affectés au module d'alimentation des stations locales créées sous l'API M580. Ce module peut changer avec des versions différentes, mais il n'est pas possible de le déplacer et de le changer de sa position attribuée.

Les stations d'**E/S Ethernet Modicon M580** figurent dans l'arborescence **Modèle topologique** sous le nœud API portant la convention de nommage **EIO Modicon_Dropx** (x : numéro de station). Vous ne pouvez pas modifier le **Type de liaison** d'un châssis d'E/S Ethernet Modicon une fois créé. Toutefois, vous pouvez à tout moment modifier le **Coupleur esclave**, le **Type de châssis** et le **Numéro de station**.

La station d'E/S Ethernet M580 prend en charge les catégories suivantes de modules :
- Module d'alimentation
- Communication
- Numérique
- Analogique
- Expert

NOTE : Vous pouvez configurer la plupart des modules d'E/S étant pris en charge pour un API M340.

La station d'extension Premium M580 prend en charge les catégories suivantes de modules :
- Module d'alimentation
- Numérique
- Analogique
- Expert

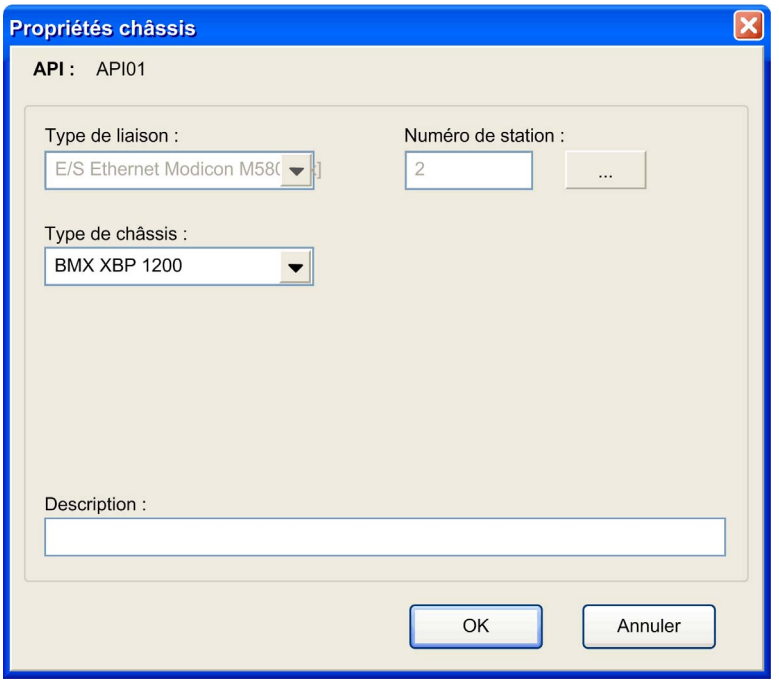
NOTE : L'architecture de l'API M580 ne prend pas en charge les modules de communication sur des stations locales d'extension Premium. C'est pourquoi sa configuration n'est pas disponible dans UAG.

Configuration d'un châssis d'extension pour stations

Vous pouvez créer des châssis d'extension pour n'importe quelle station XBus et d'E/S Ethernet du M580.

Le numéro de station d'un châssis d'extension reste le même car le numéro de station d'un châssis parent et le type de liaison sont configurés automatiquement à la station d'E/S Ethernet Modicon M580 oi XBus, en fonction du type de liaison de la station.

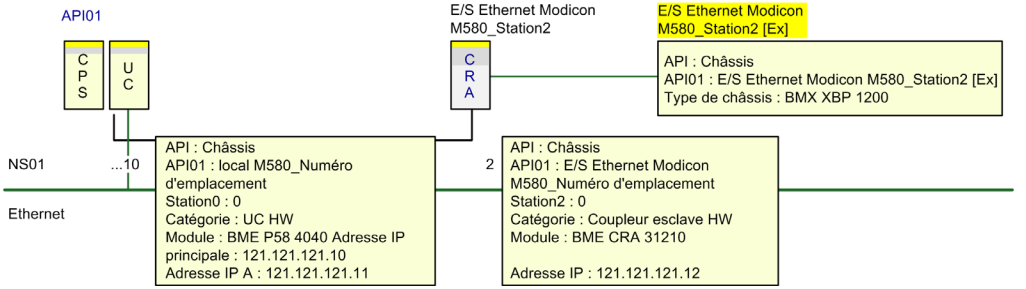
La figure suivante illustre la configuration d'un nouveau châssis d'extension d'E/S Ethernet Modicon M580 :



Afficheur topologique UAG

L'affichage topologique d'UAG présente l'architecture d'un API M580 et de ses modules configurés.

La figure suivante présente le châssis d'E/S Ethernet Modicon M580 dans l'affichage topologique d'UAG :



Copier/Coller d'une station d'E/S Ethernet

La station d'E/S Ethernet prend en charge la fonctionnalité Copier/Coller, au sein d'un même API ou entre différents API, à condition que l'API soit un M580.

Une vérification du nom de l'API et des abonnés réseau est effectuée pour valider leur unicité. En cas de problème entre des abonnés réseau et des noms de projet d'API, vous pouvez procéder aux changements nécessaires dans la fenêtre de collage de l'API avant de coller l'API.

Exportation et importation d'une station d'E/S Ethernet

Vous pouvez exporter l'API ainsi que tous les modules matériels.

UAG prend en charge la fonctionnalité d'importation pour la gamme d'API M580 et ses modules matériels.

NOTE : UAG ne prend pas en charge l'exportation et l'importation d'une station d'E/S Ethernet M580 au format CSV.

Sous-chapitre 5.4
Groupe IHM

Introduction

Cette partie décrit le groupe IHM et ses propriétés.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Introduction	215
Propriétés de l'IHM	216
Propriétés d'une zone de conduite	217
Suppression d'une IHM	218
Images et groupes d'images	219

Introduction

Introduction

On définit dans le groupe IHM plusieurs IHM. Une IHM contient tous les écrans relatifs aux éléments physiques fonctionnels et à leurs modules de commande.

- Il est possible de définir plusieurs zones de conduite pour chaque IHM.
- Chaque élément physique fonctionnel est lié à une IHM.
- Le lien entre un élément physique fonctionnel et une IHM se définit indirectement en affectant un élément physique fonctionnel à une zone de conduite.

Qu'est-ce qu'une IHM ?

Une IHM offre une représentation du procédé commandé à l'opérateur et aux ingénieurs d'exploitation. Une IHM contient tous les écrans relatifs aux éléments physiques fonctionnels et à leurs modules de commande.

Qu'est-ce qu'une zone de conduite ?

Les zones de conduite sont créées afin de permettre à différents opérateurs de surveiller différentes parties du procédé. Pour ce faire, l'ensemble de l'application IHM est segmentée en différentes zones de conduite.

A chaque zone de conduite correspond le niveau d'accès de l'opérateur. Une zone de conduite fait donc partie de l'application IHM complète et c'est de cette partie qu'est responsable l'opérateur. Cet opérateur peut commander les modules de cette zone de conduite (marche, arrêt, changement de consignes, etc.). L'opérateur ne recevra que les alarmes de la ou les zones de conduite dont il a la gestion. Unity Application Generator ne définit pas quel opérateur est responsable de la zone de conduite. Cela doit être configuré dans le système IHM.

Propriétés de l'IHM

Présentation

Unity Application Generator génère un écran par élément physique fonctionnel dans l'application IHM. Tous les modules de commande de chaque élément physique fonctionnel fait partie de la même IHM et sont générés dans le même écran d'élément physique fonctionnel.

Propriétés d'une IHM

Propriétés d'une IHM :

Propriété	Description	Options	Commentaire
`Nom d'IHM`	Nom unique de l'IHM	-	Ce nom doit respecter les conventions de nommage définies lors de la personnalisation.
`Type d'IHM`	Définit le type d'IHM que Unity Application Generator générera pour	• Monitor Pro • iFIX • IHM générique	Ce paramètre a une incidence prépondérante sur Unity Application Generator. La fonctionnalité des différentes IHM varie sensiblement. Les options IHM proposées dans d'autres boîtes de dialogue dépendent donc du type d'IHM sélectionné ici.
`Commentaire...`	-	-	Le champ commentaire est un champ texte libre. Il peut servir à documenter la conception du procédé.
`Documents...`	-	-	Il est possible d'affecter un nombre quelconque de fichiers de type différent à chaque élément. Les types de fichiers disponibles sont définis lors de la personnalisation.

Propriétés d’une zone de conduite

Présentation

Les zones de conduite sont rattachées à une seule IHM. Une zone de conduite réunit les écrans relatifs à une partie logique du procédé qui sera commandée par un opérateur. Les zones de conduite constituent des parties logiques de l’IHM et structurent ainsi cette application. Une zone de conduite est donc une partie de l’application IHM complète, qui relève de la responsabilité d’un opérateur. Car cet opérateur peut commander les modules de commande de cette zone de conduite (marche, arrêt, changement de points de consigne, etc.). L’opérateur ne recevra que les alarmes de la ou les zones de conduite dont il a la gestion. Unity Application Generator ne définit pas l’opérateur responsable de telle ou telle zone de conduite. Cela doit être configuré dans le système IHM.

Propriétés d’une zone de conduite

Propriétés d’une zone de conduite :

Propriété	Description	Options	Commentaire
`Nom de zone de conduite`	Nom de la zone de conduite		Ce nom doit respecter les conventions de nommage définies lors de la personnalisation.
Description de la `zone de conduite`	Zone de texte		
`Commentaire`	Le champ commentaire est un champ texte libre. Il peut servir à documenter la conception du procédé.		

Suppression d’une IHM

Description de la fonction de suppression

Si vous travaillez avec iFIX, observez les règles suivantes :

Si vous supprimez une IHM, vous devez générer votre application iFIX tout de suite après, avant d’apporter de quelconques modifications à votre projet UAG.

Images et groupes d’images

Objets fournis

UAG fournit 2 types d’objets dans le modèle topologique, à savoir :

- **GroupeImages**
- **Image**

Vous pouvez utiliser l’objet **GroupeImages** pour un objet IHM sélectionné et vous pouvez l’imbriquer de manière hiérarchique. Les objets **Image** peuvent être utilisés pour un **GroupeImages** sélectionné.

NOTE : Il n’y a pas de limite imposée aux nombres de niveaux d’un **GroupeImages** ou au nombre d’images d’un **GroupeImages**.

Affectation des objets

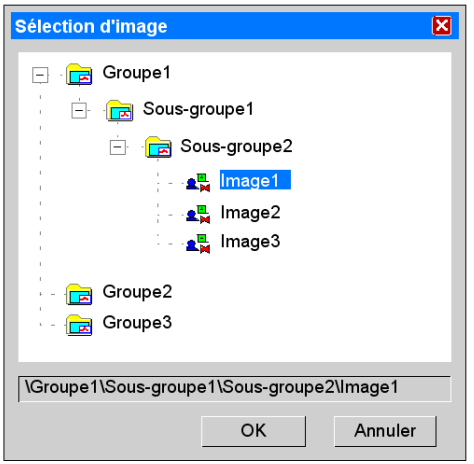
Vous trouverez, dans les éléments physiques fonctionnels, des propriétés à affecter à un objet image existant. Plusieurs éléments physiques fonctionnels peuvent être affectés à la même image. L’analyseur affiche un message si l’élément physique fonctionnel n’a pas d’image affectée.

Vous pouvez sélectionner l’IHM dans la boîte de dialogue des propriétés de l’élément physique fonctionnel. Après avoir sélectionné l’IHM, la zone de liste Zone de conduite est renseignée. Vous pouvez ouvrir un navigateur pour sélectionner les images. Dans cette boîte de dialogue, la hiérarchie des groupes d’images et les images qu’ils contiennent apparaissent dans une arborescence. Si vous avez sélectionné une image et avez fermé la boîte de dialogue, le nom de l’image apparaît avec les noms GroupeImages parents, dans la zone de texte en lecture seule.

Disponibilité des GroupesImages et des Images

Vous trouverez les groupes d'images et les images dans une grille qui s'ouvre à partir de l'objet IHM. Les types d'objets sont uniquement disponibles pour les IHM de type IHM générique ou Monitor Pro. Vous pouvez créer des groupes d'images et des images dans cette grille. Après avoir précisé un nom et donné une description facultative, sélectionnez le groupe d'images parent. Le groupe d'images parent peut rester sans affectation dans le seul cas où il est créé au niveau racine. Le nom d'un groupe d'images et d'une image doit être unique au sein du projet. Les noms, les descriptions et les relations parentales peuvent être modifiés, même si l'image est déjà affectée à un élément physique fonctionnel.

L'image suivante illustre la boîte de dialogue d'un groupe d'images :



Sous-chapitre 5.5
Le groupe Serveur de données

Introduction

Cette partie décrit les serveurs de données et leurs propriétés.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Description générale	222
Propriétés du serveur de données	223
Onglet Général	225
Onglet Réseaux (serveur principal)	226
Onglet Réseaux (serveur secondaire)	228
Serveurs de données redondants	230
Onglet Générateur de serveur de données et d'IHM	234
Suppression d'un serveur de données	236

Description générale

Présentation

Le serveur de données effectue l'échange de données entre un ou plusieurs API et les IHM.

Un serveur de données peut être utilisé par plusieurs IHM.

Un API n'est affecté qu'à un seul serveur de données (la scrutation n'est effectuée que par un seul serveur de données).

Vous ne pouvez ouvrir qu'un seul canal de communication entre l'API et une IHM.

Relation avec d'autres éléments

Le serveur de données est relié à d'autres éléments du modèle topologique, aux IHM et aux API.

- L'affectation d'un API au serveur de données est réalisée par le canal de communication associé à l'API.
- L'affectation de l'IHM au serveur de données n'est pas effectuée directement, mais par le biais d'éléments du modèle physique.
  - Chaque élément physique fonctionnel est géré à l'intérieur d'une zone de conduite.
  - Chaque élément physique fonctionnel est donc relié à une IHM.
  - Chaque élément physique fonctionnel est commandé dans un API.
  - L'API fait l'objet d'une scrutation par un serveur de données, par le biais d'un canal de communication.

  Le serveur de données est donc affecté à une IHM au travers de la relation de l'élément physique fonctionnel.

Serveurs de données redondants

Pour garantir la disponibilité des serveurs de données, vous pouvez utiliser simultanément deux options.

- Configurer deux connexions de réseau pour un serveur de données.
- Configurer deux serveurs de données (principal/secondaire).

Pour de plus amples informations, consultez les sections ci-après.

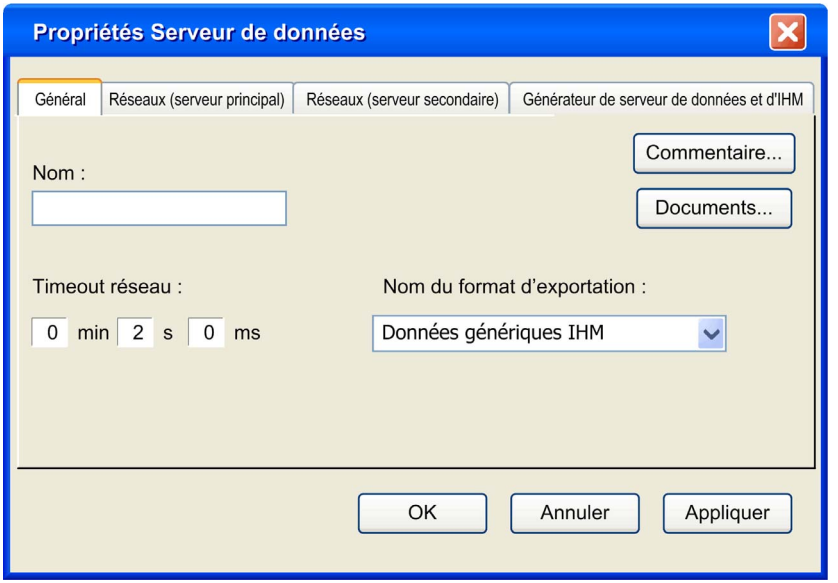
Propriétés du serveur de données

Accès aux propriétés

Pour accéder à la boîte de dialogue **Propriétés du serveur de données**, procédez comme suit.

Etape	Action
1	Dans le **modèle topologique**, sélectionnez soit le nœud **Serveurs de données** pour ajouter un nouveau serveur de données, soit un serveur de donnée déjà configuré, situé au-dessous de ce nœud, pour modifier une configuration de serveur de données.
2	Cliquez avec le bouton droit de la souris et sélectionnez **Nouveau serveur de données** ou **Propriétés** dans le menu contextuel. **Résultat** : La boîte de dialogue **Nouveau serveur de données** ou **Propriétés Serveur de données** s'affiche.

Structure générale

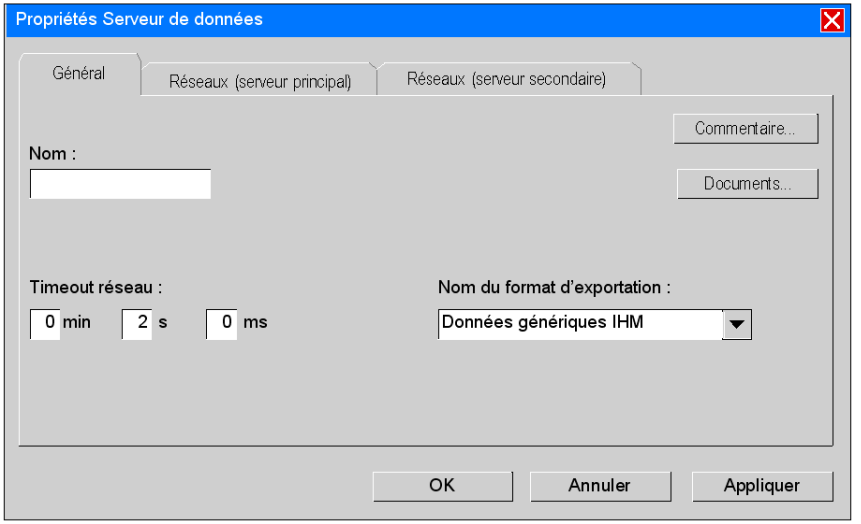


Elément	Description
Onglet **Général**	définit les propriétés générales d'un serveur de données Voir Onglet Général *(voir page 225)*.
Onglet **Réseaux (serveur principal)**	définit les propriétés réseau du serveur principal Voir Onglet Réseaux (serveur principal) *(voir page 226)*

Elément	Description
Onglet **Réseaux (serveur secondaire)**	définit les propriétés réseau du serveur secondaire, s'il est activé Voir Onglet Réseaux (serveur secondaire) *(voir page 228)*
onglet **Générateur de serveur de données et d'IHM**	définit les noms des grappes et des différents serveurs (alarmes, tendances, rapports) Voir Onglet Générateur de serveur de données et d'IHM *(voir page 234)* Cet onglet n'est disponible qu'avec le nouveau générateur Vijeo Citect. Veuillez consulter le manuel utilisateur avancé, à la section *Configuration du générateur (voir Unity Application Generator (UAG) 3.3 SP4, a SoCollaborative software, Manuel utilisateur avancé)*.

Onglet Général

Représentation

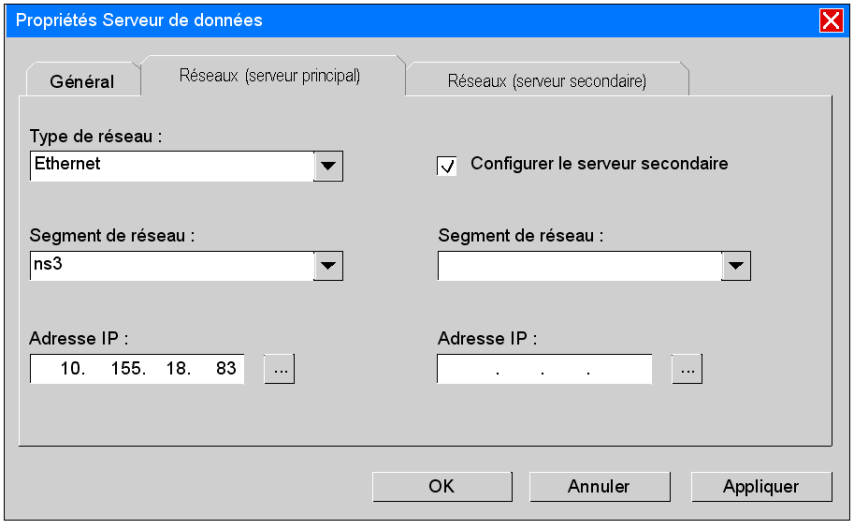


Les propriétés du serveur de données sont principalement axées sur le fonctionnement en réseau. Aucune autre définition n'est nécessaire concernant d'autres éléments.

Elément	Description
Nom	Nom unique du serveur de données Pour Monitor Pro, UAG génère l'application serveur dans le sous-répertoire *DATASERVERNAME* du **chemin de l'application serveur de données**, défini en choisissant **Vue** →**Options**. Pour iFIX, UAG génère la base de données des tags sous la forme *DATASERVERNAME.PDB*. **Remarque :** Ce nom doit respecter les conventions de nommage définies lors de la personnalisation.
Commentaire	Champ de texte libre (permet de documenter la conception du procédé).
Documents	Il est possible d'affecter à chaque élément un nombre quelconque de fichiers de différents types de document. Les types de fichiers à affecter sont définis lors de la personnalisation.
Nom du format d'exportation	Zone de liste permettant de sélectionner un nom associé à un fichier de feuille de style, qui définit le mode d'exportation des données pour ce serveur de données Les noms de formats sont définis lors de la personnalisation.
Timeout réseau	Timeout d'état de santé de la communication

Onglet Réseaux (serveur principal)

Représentation



Elément	Description
Type de réseau	Zone de liste permettant de sélectionner les types de réseaux suivants ● Ethernet ● Modbus Plus **Remarque :** Pour Monitor Pro, disponible uniquement avec OFS. Pour iFIX, non disponible.
Segment de réseau	Zone de liste permettant de sélectionner les Segments de réseau auxquels le serveur de données doit être connecté **Remarque :** Les segments de réseau doivent être définis auparavant dans le nœud **Segments de réseau** du **modèle topologique**.
Segment de réseau redondant	Grâce à une seconde zone de liste située juste à côté de la première, vous pouvez sélectionner un segment de réseau redondant pour le serveur principal.
Adresse IP ou **adresse réseau**	Zone de liste permettant de sélectionner ou de saisir une adresse valide **Remarque :** L'adresse doit être unique pour un segment donné.

Elément	Description
Adresse IP ou **adresse réseau** redondante	Seconde zone de liste, située juste à côté de la première, permettant de sélectionner ou de saisir une adresse valide pour le **segment de réseau** redondant
Configurer le serveur secondaire	Cette case à cocher permet d'activer l'onglet **Réseaux (serveur secondaire)** Vous pouvez y configurer un serveur secondaire supplémentaire (serveur redondant).

Serveur secondaire

Pour de plus amples informations sur la configuration du serveur secondaire (redondant), consultez la section *Onglet Réseaux (serveur secondaire), page 228*.

Onglet Réseaux (serveur secondaire)

Généralités

Si vous avez activé la case à cocher **Configurer le serveur secondaire** (voir la section *Onglet Réseaux (serveur principal), page 226*), la configuration du serveur secondaire est obligatoire.

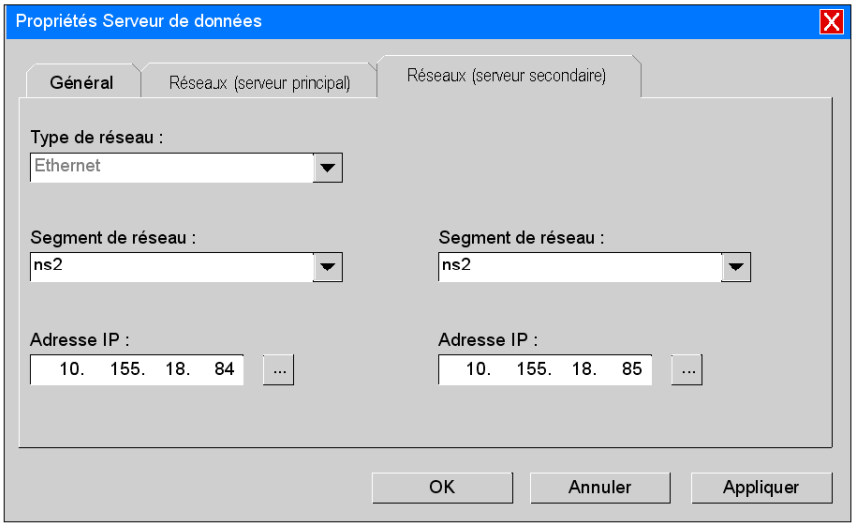
Si vous configurez le serveur secondaire, puis revenez à l'onglet **Réseaux (serveur principal)** et désactivez la case à cocher **Configurer le serveur secondaire**, l'onglet du serveur secondaire est désactivé, mais les informations sont mémorisées.

- Si vous cliquez alors sur l'option **Appliquer** ou **OK**, les informations concernant le serveur secondaire sont perdues.
- Si vous décidez de ne pas cliquer sur l'option **Appliquer** ou **OK**, mais que vous activez à nouveau la case à cocher, l'onglet du serveur secondaire est activé et les informations mémorisée s'affichent.

NOTE : La configuration d'un serveur secondaire n'est générée que pour Vijeo Citect.

Pour d'autres systèmes SCADA (par exemple, Monitor Pro, iFIX), les serveurs de données redondants ne sont pas définis dans UAG, mais doivent l'être dans le système SCADA.

Représentation



Elément	Description
Type de réseau	Cette zone de liste est désactivée pour le serveur secondaire **Remarque :** le **Type de réseau** du serveur secondaire est identique à celui du serveur principal.

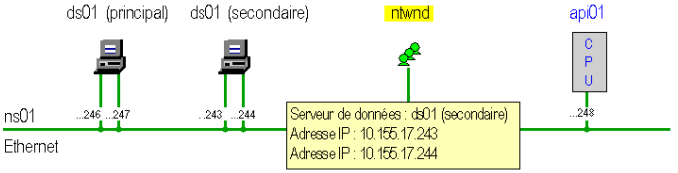
Elément	Description
Segment de réseau	Zone de liste permettant de sélectionner les segments de réseau auxquels le serveur de données doit être connecté
Segment de réseau redondant	Grâce à une seconde zone de liste située juste à côté de la première, vous pouvez sélectionner un segment de réseau redondant pour le serveur secondaire.
Adresse IP ou **adresse réseau**	Zone de liste permettant de sélectionner ou de saisir une adresse valide **Remarque :** L'adresse doit être unique pour un segment donné.
Adresse IP ou **adresse réseau** redondante	Seconde zone de liste, située juste à côté de la première, permettant de sélectionner ou de saisir une adresse valide pour le **segment de réseau** redondant

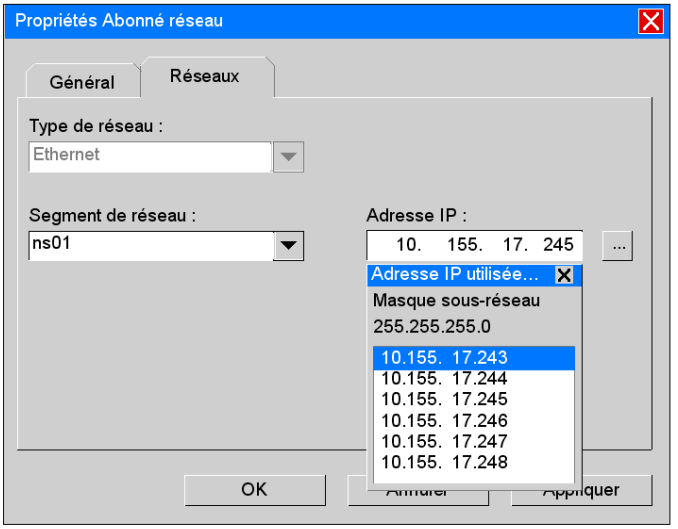
Serveurs de données redondants

Adresses de serveur secondaire

Si vous avez configuré un serveur de données principal et un serveur de données secondaire, les adresses du serveur secondaire apparaissent également dans

- l'infobulle de l'**afficheur topologique** de ce segment de réseau ;
- l'onglet **Réseaux** de la boîte de dialogue **Propriétés Abonné réseau**

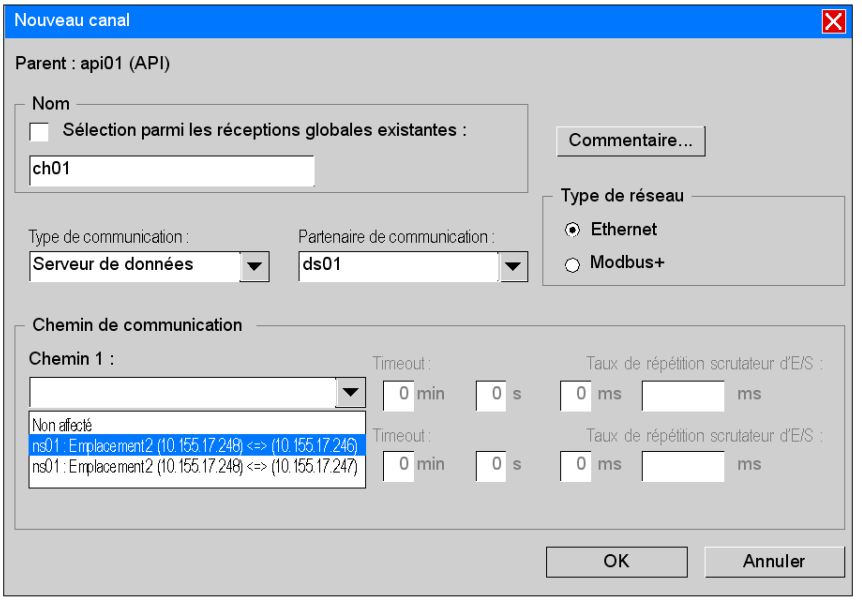




Canal uniquement avec serveur principal

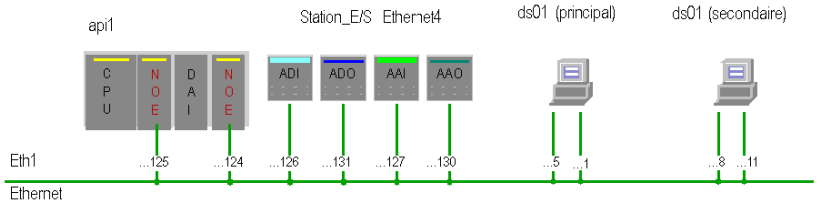
Le canal d'un API au serveur de données ne peut être défini qu'en utilisant les adresses du serveur principal.

La zone de liste **Chemin de communication** de la boîte de dialogue **Nouveau canal** ne contient que les adresses IP du serveur principal.



Afficheur topologique

Dans le menu principal, cliquez sur l'option **Vue** →**Afficheur topologique**.



Les serveurs de données redondants apparaissent dans l'afficheur topologique.

- **ds01 (principal)** est le serveur de données principal.
- **ds01 (secondaire)** est le serveur de données secondaire.
- Chaque serveur de données possède des connexions redondantes configurées sur le même segment de réseau **Eth1**.

Si vous ouvrez la boîte de dialogue des propriétés à partir des deux symboles, elle fait référence au même serveur de données.

Mais selon le symbole sur lequel vous cliquez, l'onglet correspondant (Principal.../Secondaire...) s'affiche.

Une infobulle affiche alors les différentes adresses configurées.

Onglet Générateur de serveur de données et d'IHM

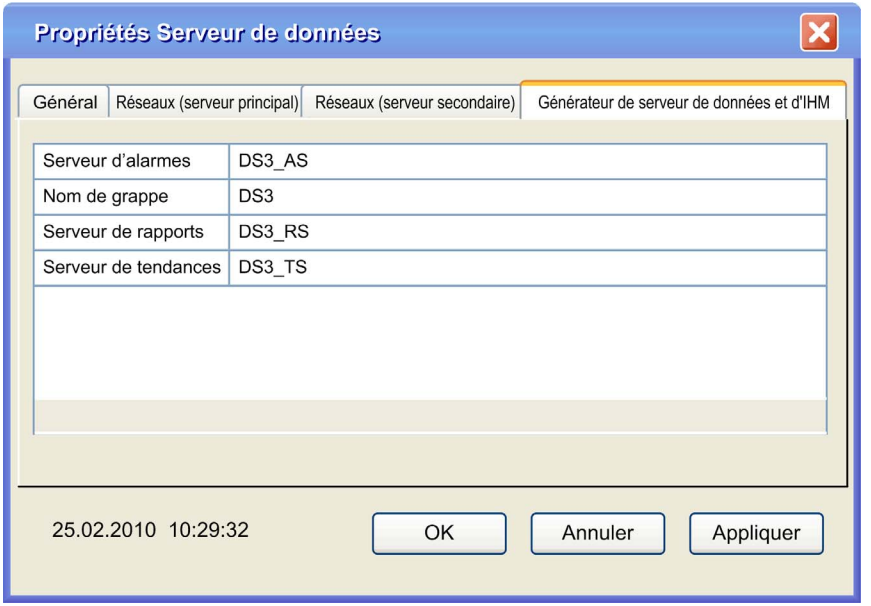
Généralités

Il existe des propriétés de générateur évoluées supplémentaires pour les objets serveur de données figurant dans les projets dans lesquels le générateur Vijeo Citect est utilisé.

Pour créer plusieurs grappes dans Vijeo Citect, vous devez préciser le nom des grappes dans UAG à l'aide des propriétés étendues propres au générateur se rapportant au serveur de données (voir image ci-dessous).

Les propriétés ont leur valeur par défaut.

Représentation



Elément	Description
Serveur d'alarmes	nom du serveur d'alarmes créé sur la même machine que celle utilisée pour le serveur d'E/S généré pour le serveur de données. La valeur par défaut est <Nom Serveur Données>_AS.
Nom de grappe	nom de la grappe du serveur de données La valeur par défaut est <Nom Serveur Données>

Elément	Description
Serveur de rapports	nom du serveur de rapports créé sur la même machine que celle utilisée pour le serveur d'E/S généré pour le serveur de données. La valeur par défaut est <Nom Serveur Données>_RS.
Serveur de tendances	nom du serveur de tendances créé sur la même machine que celle utilisée pour le serveur d'E/S généré pour le serveur de données. La valeur par défaut est <Nom Serveur Données>_TS.

Nom de grappe

Grâce à une propriété fournie par le générateur, vous pouvez spécifier un nom de grappe pour chaque serveur de données.

La saisie du même nom de grappe pour différents serveurs de données permet de regrouper ces derniers en une seule grappe.

Serveurs d'alarmes/de rapports/de tendances

D'autres propriétés propres au générateur sont prévues pour permettre à l'utilisateur de spécifier s'il faut créer, ou non, des serveurs d'alarmes, de tendances ou de rapports sur la même machine que celle utilisée pour le serveur d'E/S généré pour le serveur de données.

Vous êtes autorisé à spécifier plusieurs noms de serveur en les qualifiant d'un nom de grappe, car dans Vijeo Citect il est possible d'avoir par exemple, sur la même machine, plusieurs serveurs d'alarmes appartenant à différentes grappes.

Mais seule une paire de serveurs redondants est autorisée pour chaque grappe.

Vous pouvez par exemple entrer des noms tels que `SrvAlarm1`, `GrappeB.SrvAlarm2`, ce qui veut dire que `SrvAlarm1` sera utilisé avec la grappe spécifiée dans la propriété **Nom de grappe** du serveur de données tandis que `SrvAlarm2` est le serveur d'alarmes de la `GrappeB`.

Modification du nom du serveur de données

Si vous modifiez le nom du serveur de données à l'onglet **Général**, l'ancien nom du serveur de données sera remplacé par le nouveau nom dans les valeurs des propriétés **Serveur d'alarmes**, **Nom de grappe**, **Serveur de rapports** et **Serveur de tendances**.

Par exemple, si le serveur de données a pour ancien nom **DS3**, si la valeur de la propriété **Serveur d'alarmes** est **DS3_AS** et que vous donnez au serveur de données le nom **DS4**, la propriété de générateur deviendra automatiquement **DS4_AS**.

Suppression d’un serveur de données

Description de la fonction de suppression

Si vous travaillez avec iFIX, observez les règles suivantes :

Si vous supprimez un serveur de données, vous devez générer votre application iFIX tout de suite après, avant d’apporter de quelconques modifications à votre projet UAG.

Sous-chapitre 5.6
Le groupe Abonnés réseau

Introduction

Cette partie décrit les abonnés réseau et leurs propriétés.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Description d'un partenaire réseau	238
Propriétés d'un partenaire réseau	240
Description des autres abonnés	241
Propriétés des autres abonnés	242

Description d'un partenaire réseau

Qu'appelle-t-on des partenaires réseau ?

Un automatisme utilise plusieurs composants matériels. Les composants pouvant être connectés à d'autres composants via le réseau mais qui ne sont ni du type API ni du type IHM ni Serveur de données, ne peuvent être générés directement par Unity Application Generator. C'est notamment le cas des programmateurs de poche Magelis. Pour ces appareils appelés des partenaires réseau, Unity Application Generator peut exporter un fichier CSV contenant les variables avec les informations nécessaires pour la scrutation par le partenaire réseau (adresses, type de données, valeur initiale, etc.).

Reportez-vous également au Manuel avancé UAG, variables de partenaire réseau *(voir Unity Application Generator (UAG) 3.3 SP4, a SoCollaborative software, Manuel utilisateur avancé)*.

Conditions

Pour introduire des partenaires réseau, les conditions suivantes doivent être respectées :

- Les partenaires réseau doivent avoir été définis dans Unity Application Generator.
- Toutes les variables communiquées aux partenaires réseau doivent être définies (dans l'élément physique fonctionnel ou le module de commande) comme les variables API_RES.
- Un canal de communication doit être défini entre l'API et le partenaire réseau.
- Dans l'outil de programmation du partenaire réseau, il doit exister une option d'importation de liste de variables sous forme de fichier CSV.

Quels sont les éléments générés pour les partenaires réseau ?

Pour l'API :

- Unity Application Generator crée des variables localisées de sorte que le partenaire réseau puisse y accéder.

Pour le partenaire réseau :

- Unity Application Generator crée une liste de variables au format CSV pour traitement ultérieur, par exemple pour importation dans le partenaire réseau, reportez-vous également au Manuel avancé UAG, Génération pour les partenaires réseau *(voir Unity Application Generator (UAG) 3.3 SP4, a SoCollaborative software, Manuel utilisateur avancé)*.

Exportation Magelis

UAG permet de générer des fichiers CSV à importer dans les terminaux Magelis grâce au bouton `Exportation Magelis` de la boîte de dialogue des propriétés des `Abonnés de réseau` (pour le logiciel XBT-L1000 uniquement).

Le tableau suivant présente les différences existant entre l'exportation traditionnelle et l'exportation vers Magelis.

	Exportation traditionnelle	**Exportation Magelis**
Fichiers API	Crée un fichier CSV unique pour tous les API.	Crée un fichier par API.
Format du fichier	`plc;networktype;networkpath; var;datatype;984addr; initial;alarmtext`	`N;var;datatype;984addr; initial;alarmtext`

Propriétés d’un partenaire réseau

Plage de mémoire des partenaires réseau

La plage générale de mémoire pour tous les partenaires réseau est définie dans les propriétés de l’API. Les plages de mémoire de chaque partenaire réseau sont définies dans les propriétés du canal. Chaque partenaire réseau utilise une plage 0x/%M et une plage 4x/%MW dans l’API.

Propriétés

Propriétés d’un partenaire réseau :

Propriété	Commentaire
`Type`	Partenaire réseau (type de l’abonné réseau).
`Nom`	Nom unique du partenaire réseau. Ce nom doit respecter les conventions de nommage définies lors de la personnalisation.
`Description`	Vous avez la possibilité de donner une brève description du partenaire réseau dans le champ Description.
`Fichier matriciel (bitmap)`	Pour chaque partenaire réseau, vous pouvez choisir un fichier matriciel, qui s’affiche dans l’afficheur topologique. Les formats matriciels possibles sont les suivants : ● .bmp ● .jpg ● .gif ● .ico En l’absence de sélection de fichier matriciel, un symbole par défaut apparaît dans l’afficheur topologique.
`Type de réseau`	Modbus Plus ou Ethernet
`Segment de réseau`	Définit le segment auquel le partenaire réseau est connecté.
`Adresse réseau` ou `Adresse IP`	Adresse de réseau Modbus Plus ou Ethernet
`Commentaire...`	Le commentaire est un champ texte libre. Il peut servir à documenter la conception du procédé.
`Documents...`	Il est possible d’affecter un nombre quelconque de fichiers de type différent à chaque élément. Les types de fichiers disponibles sont définis lors de la personnalisation.
`Var. export`	Génération d’un fichier .CSV contenant les variables affectées au partenaire réseau actuel.

Description des autres abonnés

Qu'entend-on par autres abonnés ?

Un automatisme utilise plusieurs composants matériels. Les composants pouvant être connectés à d'autres composants par le réseau mais qui ne sont ni du type API ni du type IHM ni des serveurs de données ni des partenaires réseau , ne peuvent être pas générés par Unity Application Generator. Pour ces composants appelés Autres abonnés, Unity Application Generator peut réserver une adresse réseau pour empêcher l'affectation multiple des adresses réseau.

Conditions

Pour introduire d'Autres abonnés, les conditions suivantes doivent être respectées :

- Les autres abonnés doivent avoir été définis dans Unity Application Generator.

Quels sont les éléments générés pour les autres abonnés ?

Rien. Seule une adresse réseau est réservée.

Propriétés des autres abonnés

Propriétés

Propriétés d'un autre abonné :

Propriété	Commentaire
`Type`	Autre abonné (type de l'abonné réseau).
`Nom`	Nom unique de l'autre abonné. Ce nom doit respecter les conventions de nommage définies lors de la personnalisation.
`Description`	Vous avez la possibilité de donner une brève description de l'autre abonné dans le champ Description.
`Fichier matriciel (bitmap)`	Pour chaque autre abonné, vous pouvez choisir un fichier matriciel, qui s'affiche dans l'afficheur topologique. Les formats matriciels possibles sont les suivants : ● .bmp ● .jpg ● .gif ● .ico En l'absence de sélection de fichier matriciel, un symbole par défaut apparaît dans l'afficheur topologique.
`Type de réseau`	Modbus Plus ou Ethernet
`Segment de réseau`	Définit le segment auquel l'autre abonné est connecté.
`Adresse réseau` ou `Adresse IP`	Adresse de réseau Modbus Plus ou Ethernet
`Commentaire...`	Le commentaire est un champ texte libre. Il peut servir à documenter la conception du procédé.
`Documents...`	Il est possible d'affecter un nombre quelconque de fichiers de type différent à chaque élément. Les types de fichiers disponibles sont définis lors de la personnalisation.

Partie II
Travailler avec Unity Application Generator

Introduction

Cette partie vous permettra de travailler de manière correcte et efficace avec Unity Application Generator. Pour élaborer une application, suivez les étapes de la méthode opératoire et consultez les informations détaillées nécessaires pour réaliser les étapes dans les autres parties de ce manuel.

Contenu de cette partie

Cette partie contient les chapitres suivants :

Chapitre	Titre du chapitre	Page
6	Règles de travail avec Unity Application Generator	245
7	Manipulation de l'outil et caractéristiques pour un travail efficace	251
8	Schéma d'élaboration d'une application	317
9	Rubriques spécifiques aux bus de terrain	349
10	Intégration Advantys	371

Chapitre 6
Règles de travail avec Unity Application Generator

Introduction

Ce chapitre vous informe sur les règles, les directives et les messages de sécurité à prendre en compte lorsque vous utilisez Unity Application Generator.

⚠ AVERTISSEMENT
FONCTIONNEMENT IMPREVU DE L'EQUIPEMENT Lisez attentivement ces règles et respectez-les lorsque vous travaillez avec Unity Application Generator. Dans le cas contraire, vous risquez d'obtenir des configurations de projet erronées ou un code non valide. **Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.**

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Règles générales pour la configuration et la génération de projet	246
Ouverture des éditeurs SCoD et de personnalisation	247
Règles concernant l'API et Unity Pro	248
Règles concernant les IHM	250

Règles générales pour la configuration et la génération de projet

Définition du procédé en une seule fois

NOTE : Servez-vous de Unity Application Generator pour définir le procédé et pour effectuer ensuite les modifications de configuration. Cela permet de s'assurer de la cohérence des définitions.

Dans les environnements classiques d'automatisation, dans lesquels la définition de la logique d'automate, des écrans IHM et du réseau est effectuée par des outils distincts, de nombreuses définitions doivent être répétées deux voire trois fois. Dans UAG, ces définitions ne sont effectuées qu'une seule fois et elles sont donc cohérentes. Toutes les modifications apportées aux procédés, fixées avec UAG, sont définies uniquement avec UAG et nulle part ailleurs. Par exemple, toutes les variables localisées doivent être définies dans UAG et ne doivent pas être modifiées dans Unity Pro.

⚠ ATTENTION

APPLICATION IMPREVISIBLE

N'utilisez pas d'outil comme Unity Pro pour modifier des segments de l'application générée par UAG. Pour assurer une maintenance correcte des liaisons de communication et des propriétés, toute modification apportée à ces segments doit être effectuée dans l'environnement UAG.

Le non-respect de ces instructions peut provoquer des blessures ou des dommages matériels.

Nouvelle génération ou génération incrémentale

NOTE : Faites attention lorsque vous choisissez la nouvelle génération. Si vous avez déjà commencé à renseigner la logique après une précédente génération, effectuez une génération incrémentale. Sinon vos changements manuels seront perdus.

Une partie significative de la logique d'API et des écrans IHM est générée par UAG. Il est néanmoins nécessaire d'ajouter de la logique d'API et de compléter les écrans IHM ultérieurement. Même si la logique et les écrans sont modifiés à l'aide des outils de programmation API et IHM, il est possible d'effectuer des modifications sur toutes les parties du procédé défini dans UAG et de reconstruire l'application de façon incrémentale **sans** détériorer les modifications apportées en dehors de UAG.

Equipements et modules de commande

Lorsqu'un équipement ou module de commande est généré, il n'est plus possible de modifier l'API.

Ouverture des éditeurs SCoD et de personnalisation

Introduction

Dans UAG, l'utilisateur peut ouvrir les éditeurs SCoD et de personnalisation au sein de l'interface utilisateur de UAG. Pour ce faire, deux chemins (**Vue →Editeur de personnalisation** et **Vue → Editeur SCoD**) et deux boutons de barre d'outils sont disponibles.

Ouverture en lecture seule

Les éditeurs SCoD ou de personnalisation ouverts dans UAG sont en lecture seule. Cela signifie que l'utilisateur peut consulter les données, mais qu'il ne peut pas les modifier. Il est néanmoins possible d'enregistrer ces données dans un nouveau fichier.

Lorsque l'éditeur de personnalisation est lancé dans UAG, il affiche la personnalisation du projet en cours. Si aucun projet n'est ouvert, l'éditeur de personnalisation s'ouvre et n'affiche aucune personnalisation.

Lorsque l'éditeur SCoD est lancé dans UAG, il affiche le SCoD actuellement sélectionné en mode d'affichage FFB. Si une variable de module de commande est sélectionnée dans UAG, l'éditeur SCoD ouvre le SCoD correspondant et affiche la variable sélectionnée. Si aucun projet n'est ouvert dans UAG, ou qu'aucun SCoD ou aucune variable de module de commande n'est sélectionné(e), l'éditeur SCoD s'ouvre et n'affiche pas de bibliothèque.

Les éditeurs SCoD et de personnalisation permettent à l'utilisateur d'ouvrir les bases de données (bibliothèques ou bases personnalisées) en lecture seule et ce même s'il travaille de manière autonome. Une option (case à cocher) supplémentaire est ajoutée à la boîte de dialogue standard d'ouverture des fichiers. Lorsque la bibliothèque de la base de données de personnalisation est déjà ouverte en accès exclusif par Unity Application Generator, la base de données s'ouvre automatiquement en lecture seule.

Règles concernant l'API et Unity Pro

Présentation

D'une manière générale, tout ce qui peut être défini dans UAG doit être défini dans UAG et nulle part ailleurs. UAG permet à l'utilisateur de définir toutes les informations nécessaires concernant la mémoire de l'API, par exemple, les variables, les E/S, etc.

En cas de modification de ces définitions par Unity Pro, UAG écrasera les changements lors de la génération suivante du code pour Unity Pro (si la case **Générer entièrement à neuf** est cochée dans l'onglet **API** de la boîte de dialogue **Options**).

Souvenez-vous...

Grâce à la génération incrémentale, UAG offre une manière simple et rapide de mettre à jour le projet Unity Pro sur la base des informations modifiées par Unity Application Generator. En vous servant de cette fonctionnalité sans apporter manuellement de modifications dans Unity Pro, vous vérifiez que la conception du procédé - telle que définie dans Unity Application Generator - est toujours synchronisée avec le programme et la configuration API.

Règles concernant la logique générée

Appliquez les règles suivantes :

- Ne pas créer manuellement de logique d'API si elle peut l'être par Unity Application Generator.
- Ne pas changer les valeurs littérales connectées aux entrées des blocs fonctions, leurs valeurs sont définies dans Unity Application Generator.
- Ne pas supprimer les variables des blocs fonctions générés par Unity Application Generator.
- Si vous changez des valeurs dans Unity Pro, mettez-les à jour dans UAG.

Règles concernant les variables

Appliquez les règles suivantes :

- Ne pas déclarer de variables localisées dans Unity Pro, toutes les variables localisées doivent être déclarées dans Unity Application Generator afin de vérifier la cohérence de la cartographie mémoire générée par UAG.
- Pour la logique supplémentaire, il est possible de créer dans Unity Pro des variables non localisées, mais aussi de les définir dans UAG, ce dernier autorisant l'ajout de variables libres soit aux éléments physiques fonctionnels, soit aux modules de commande.

Règles concernant les noms

Appliquez les règles suivantes :

- Ne pas changer dans Unity Pro les noms des objets ayant été générés par Unity Application Generator, par exemple le nom d'une variable, le nom d'instance d'une fonction de module de commande ou le nom d'une section.
  Si vous changez le nom, UAG ne sera plus capable de trouver les objets dans le projet Unity Pro et ne pourra donc plus les faire évoluer conformément aux modifications de la conception du procédé.

Modules déconseillés

Les modules TSX PAY 262 et TSX PAY 282 ne seront pas gérés correctement par Unity Pro. Un fonctionnement imprévisible peut survenir en cas d'utilisation de ces modules dans UAG.

Règles concernant les IHM

Présentation

Vous trouverez ci-après des règles de travail avec Unity Application Generator associé à une IHM donnée.

Règles pour Vijeo Citect/ Monitor Pro/ iFIX

Respectez les règles suivantes lorsque vous associez une IHM Vijeo Citect / Monitor Pro / iFIX à Unity Application Generator :

- Si une nouvelle zone de conduite est ajoutée dans Unity Application Generator, ajoutez la zone d'alarme et les zones de sécurité correspondantes dans iFIX.
- Si les droits d'accès sont changés dans la personnalisation de Unity Application Generator, adaptez les zones de sécurité existantes dans iFIX.
- Ne modifiez pas dans Vijeo Citect / Monitor Pro / iFIX ce qui a été généré par Unity Application Generator, par exemple :
  - Ne renommez pas les images.
  - Ne modifiez pas les propriétés des symboles IHM (sauf le réglage des couleurs).
  - Ne supprimez pas d'étiquettes.

Chapitre 7
Manipulation de l’outil et caractéristiques pour un travail efficace

Introduction

Unity Application Generator est conçu dans l’objectif global d’améliorer l’efficacité de la configuration d’un système de commande de procédé.

Pour satisfaire à cet objectif, l’interface utilisateur a été conçue de manière à ce que vous puissiez l’utiliser intuitivement si vous avez l’habitude d’autres applications Windows.

Ce chapitre traite des concepts de base de l’interface utilisateur et des caractéristiques pour améliorer votre productivité au moment de configurer votre projet.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
7.1	Concepts généraux de l’interface utilisateur	252
7.2	Editeur graphique de forçages	277
7.3	Afficheur de variables	300

Sous-chapitre 7.1
Concepts généraux de l’interface utilisateur

Introduction

Cette section concerne les concepts généraux de l’interface utilisateur.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Concepts de l’interface utilisateur	253
Utilisation des tables (listes)	255
Glisser-déposer des objets, modules et variables	257
Recherche d’objets à l’aide de critères de recherche	259
Utilisation de la liste d’instruments	261
Copier et coller dans les modèles physique et topologique	264
Etablissement d’une définition de forçage	266
Utilisation de l’afficheur topologique	269
Comment visualiser l’état de la génération	272
Comment échanger les noms	273
Fenêtre - Raccourcis clavier	276

Concepts de l'interface utilisateur

Introduction

Unity Application Generator a pour but principal d'accroître l'efficacité de l'ingénieur procédé et de l'ingénieur automaticien dans l'élaboration d'une application d'automatisation. Cette efficacité n'est pas seulement le fruit du concept entièrement nouveau de Unity Application Generator en tant qu'outil de conception de procédé mais c'est également l'objectif principal de la conception de l'interface utilisateur.

Similarités avec Windows

La manipulation de Unity Application Generator est semblable à celle des autres logiciels tels que Word, Excel et autres.

Les principes généraux suivants sont particuliers à Unity Application Generator.

- **Propriétés orientées objet** : Les propriétés sont accessibles depuis l'objet.
- **Menus contextuels** : Le menu principal ne comporte que quelques options. La plupart des options se trouvent dans les menus contextuels, accessibles par un clic droit de la souris.

Une navigation performante

Les possibilités de navigation dans Unity Application Generator sont les suivantes :

- **A partir du modèle physique et du modèle topologique** : Menus contextuels sur tous les objets pour naviguer vers des tableaux/listes et des boîtes de dialogue de définition des propriétés.
- **À partir des tableaux et des listes** : Menus contextuels sur toutes les lignes pour naviguer vers d'autres tableaux/listes et des boîtes de dialogue de définition des propriétés.
- **À partir du Rapport d'analyse** : Navigation vers les objets erronés par double-clic sur l'entrée.
- **À partir du rapport de recherche :** Navigation vers l'un des objets trouvés par double-clic sur l'entrée.
- **À partir de l'Afficheur topologique** : Navigation vers les boîtes de dialogue des propriétés depuis la vue topologique.

Pas de commande Enregistrer

Unity Application Generator enregistre automatiquement les modifications apportées à la base de données du projet sans commande explicite de **Sauvegarde**. Cela permet de s'assurer que les informations du projet modifié sont conservées, même si le PC devient inopérant.

Contrôle de toutes les entrées

Lors de la définition d'une application avec Unity Application Generator, le système vérifie immédiatement, et avec toute la rigueur possible, toutes les entrées d'utilisateur pour s'assurer qu'elles sont valides. Cela diminue d'autant le temps de mise au point de l'application.

De plus, partout où cela est possible, vous trouverez des zones de listes afin de faciliter la saisie et d'éviter des entrées erronées. La plupart des limites servant à contrôler les entrées sont paramétrées lors du processus de personnalisation, lequel s'effectue à l'aide d'un outil spécial, en l'occurrence l'éditeur de personnalisation de Unity Application Generator.

Affectation de valeurs par défaut

Des valeurs par défaut sont affectées autant que faire se peut. Même si parfois une entrée utilisateur est nécessaire, il n'est pas obligatoire d'attribuer des valeurs à un objet dès sa saisie. Il y a en principe trois options d'affectation :

- Une valeur
- Aucune/zéro/non utilisé
  Cela signifie en général que ce champ n'est pas utilisé.
- Non affecté
  Cela signifie qu'une valeur est obligatoire mais pas encore définie, par ex. l'ingénieur procédé a commencé à définir un élément mais il sera terminé ultérieurement par l'ingénieur automaticien. Lors de l'analyse du projet, toutes les valeurs non affectées sont mentionnées dans le rapport.

Actualiser par l'utilisateur

Le projet affiché en cours de conception est actualisé de manière active par l'utilisateur à l'aide de la fonction Actualiser (en appuyant sur la touche F5).

Utilisation des tables (listes)

Tables

Les tables servent à saisir et à afficher les listes d'un groupe d'objets avec leurs attributs.

Exemples

- Liste d'objets : Eléments physiques fonctionnels, modules de commande, variables, modules matériels...
- Liste d'instruments

Caractéristiques

- Edition directe des champs
- Copier et coller
- Navigation vers des boîtes de dialogue de propriétés ou d'autres tables

Menus contextuels

Les menus contextuels apparaissent en cliquant avec le bouton droit sur une ligne ou dans la boîte de dialogue de la table.

Il existe un menu contextuel par exemple :

- Pour chaque ligne de la table comportant des éléments tels que **Supprimer** (pour supprimer la ligne) ou**Propriétés** (pour accéder à la boîte de dialogue des propriétés de l'objet).
- Pour la boîte de dialogue de la table avec des éléments tels que **Nouveau** (pour ajouter une nouvelle ligne) ou **Coller** pour coller un objet copié dans la table.

Ajouter nouvelles lignes

Une nouvelle ligne est ajoutée par l'option **Nouveau** du menu contextuel. Avec Nouveau, vous accédez à une boîte de dialogue.

Nouveau... Boîtes de dialogue

Dans la boîte de dialogue **Nouveau...**, vous définissez les propriétés du ou des objets à créer.

Appliquer enregistre l'information saisie et **OK** enregistre l'information et ferme la boîte de dialogue.

Créer des variables libres

Dans la boîte de dialogue **Nouvelle variable**, il est possible de créer en une seule fois une ou plusieurs variables libres. Suivez ces étapes :

Si vous voulez...	Alors ...
Créer une variable libre ou Créer plusieurs variables libres équivalentes	**1.** Cliquez avec le bouton droit n'importe où dans la grille des variables de l'afficheur de variables. **2.** Choisissez **Nouveau** dans le menu contextuel. Veuillez consulter la section *Coller/Nouveau, page 305*. **3.** Entrez le `Nom`, la `Description` et le `Type de connexion`, le `Type de données` et l'`E/S` de la ou des variables. **4.** **En option** : Saisissez le `Numéro de départ` et le `Nombre` de variables équivalentes à créer. **5.** Appuyez sur **Appliquer**. **Résultat** : la ou les nouvelles variables apparaissent dans la grille des variables. Dans le cas de plusieurs variables, les noms sont numérotés de manière consécutive. **6.** Si nécessaire, entrez les autres propriétés en vous déplaçant parmi les onglets (Alarme, Commande, Affichage). **7.** Validez par **OK**. Les propriétés sont enregistrées comme vous les avez saisies pour toutes les variables créées. La boîte de dialogue se ferme.

Enregistrer une ligne

Lorsque vous saisissez une information dans une ligne d'une table, les changements sont mémorisés dans la base de données dès que vous passez à une autre ligne. Tant que vous travaillez sur les différents champs d'une même ligne, les données **ne sont pas** enregistrées.

Sélectionner et supprimer des lignes

Le tableau suivant vous explique comment sélectionner et supprimer des lignes d'une table :

Si vous voulez...	Alors...
Sélectionner une ligne	Cliquez dans la colonne la plus à gauche de ligne.
Sélectionnez un groupe de lignes	• Cliquez sur la colonne la plus à gauche de la première ligne et faites glisser la souris sur plusieurs lignes sans relâcher le bouton de la souris. **ou bien** : • Cliquez sur la colonne la plus à gauche de la première ligne, appuyez sur la touche **MAJ** tout en cliquant sur la dernière ligne à sélectionner.
Sélectionner plusieurs lignes non consécutives	Cliquez sur la colonne la plus à gauche de chaque ligne à sélectionner tout en appuyant sur la touche **Ctrl**.
Supprimer plusieurs lignes sélectionnées	Utilisez la touche **Suppr**.

Glisser-déposer des objets, modules et variables

Fonctionnalité d'édition glisser-déposer

La fonction de glisser-déposer facilite la configuration d'un projet à la fois pour l'ingénieur projet et pour l'ingénieur automaticien.

Restrictions d'utilisation de l'opération glisser-déposer

Vous pouvez activer une alerte de procédé en cas d'utilisation d'une opération glisser-déposer. Pour activer l'alerte de procédé, ouvrez la boîte de dialogue **Options**.

Ouvrez **Vue** →**Options..** →**Général**. Sélectionnez la case à cocher **Confirmations** (**avant de déplacer un objet vers un autre parent de l'arborescence physique**) pour activer l'alerte de procédé.

NOTE : La case à cocher est désactivée par défaut.

Glisser-déposer dans l'arborescence du modèle physique

Vous pouvez, dans l'arborescence du modèle physique, déplacer des objets d'une position à une autre par glisser-déposer. Les positions cibles valides/non valides sont indiquées lorsque vous déplacez l'objet avec la souris.

Que se passe-t-il lorsqu'un objet est déplacé par glisser-déposer ?

- Tous les noms de variable sont automatiquement modifiés en respect des conventions de nommage.
- Les sections Unity Pro, les caractéristiques d'IHM et autres, sont automatiquement déplacées et adaptées.
- Les modifications sont commentées dans le fichier Historique des changements.

NOTE :

- Après avoir déplacé un objet au sein de l'arborescence du modèle physique, il est nécessaire d'effectuer une génération de code afin de mettre ce dernier à jour.
- Le déplacement d'une unité n'influence pas l'ordre d'exécution dans Unity Pro.

Déplacement de modules matériels

Pour les API de Unity Pro :

Il est possible de déplacer les modules matériels au sein d'un même châssis et d'un châssis à l'autre par glisser-déposer dans une table de châssis ou d'une table de châssis à l'autre (au sein du même API).

NOTE : Si vous copiez des modules de communication ou analogiques dans Unity Pro, les modules situés aux anciens emplacements sont supprimés et de nouveaux modules sont créés au nouvel emplacement. Cela entraîne la suppression de toutes les données (canaux, E/S, etc.) affectées à ces modules.

Il en va de même si vous changez un numéro de station RIO dans Unity Pro. L'intégralité du châssis avec tous les modules est supprimée et un nouveau châssis est créé.

Affectation de variables

Dans l'afficheur des variables (reportez-vous à *Afficheur de variables, page 300*), vous affectez des variables à des types de connexions d'API (ES_API, API, API_IHM, API_RES) par une opération de glisser-déposer sur les cibles suivantes :

- Variable de type E/S_API vers modules E/S
- Variable de tous types vers canaux API-API
- Variable de type API_RES vers canaux de partenaires réseau
- Variables vers l'éditeur graphique de forçages (reportez-vous à *Editeur graphique de forçages, page 277*)
- Variables des liaisons

Recherche d'objets à l'aide de critères de recherche

Fonction Trouver

Unity Application Generator est capable de rechercher des objets répondant à certains critères. La recherche s'applique à un certain type d'objet (module de commande, variable, API...). Un ou plusieurs critères de recherche sont combinés par les fonctions AND et OR jouant le rôle de filtres. Les critères sont basés sur les attributs d'objets. Dans les chaînes de caractères, il est possible d'utiliser des caractères génériques.

Exemple : Vous pouvez rechercher les variables d'alarme possédant une priorité d'alarme `haute`.

Constituants d'une recherche

Une recherche se base sur la définition du

- **Type d'objet**

Un critère de recherche simple est constitué des éléments suivants :

- **Attribut** : Attribut du type d'objet sélectionné. Unity Application Generator propose une liste d'attributs.
- **Opérateur** : Opérateur de comparaison avec le critère. Unity Application Generator propose une liste d'opérateurs correspondant à la nature de l'attribut.
- **Critère** : Valeur de l'attribut recherché. Selon la nature de l'attribut, Unity Application Generator propose une liste ou un champ libre pour saisir un nombre ou une chaîne de caractères.

Plusieurs critères de recherche sont combinés par des opérateurs logiques

- **And/Or** : Opérateurs de combinaison de plusieurs critères de recherche. Le premier critère de recherche ne nécessite pas de AND ni de OR. Dans une recherche avec opérateur AND et OR, l'opérateur AND possède la plus haute priorité.

Opérateur LIKE

L'opérateur LIKE peut servir si le critère est un texte. Il est possible d'utiliser des caractères génériques pour définir des modèles de recherche.

NOTE : Dans la chaîne ou le modèle de texte à rechercher, les caractères spéciaux `|` et `"` **ne sont pas** autorisés.

Exemple d'utilisation de caractères génériques pour rechercher une plage de valeurs : `LIKE Va*`

Mais : Exemple de définition de l'intégralité de la valeur : `=Vanne`

Caractères génériques

Les caractères génériques utilisés dans les modèles de recherche d'expressions LIKE sont définies comme suit :

Caractère générique	Type d'analogie	Exemple	Analogie	Pas d'analogie
*	Caractères multiples	a*a	Aa, aBa, aBBBa	ABC
		ab	w	AZb, bac
		ab*	Abcdefg, abc	Cab, aab
[*]	Caractère spécial	a[*]a	A*a	Aaa
?	Caractère unique	a?a	aaa, a3a, aBa	ABBBa
#	Chiffre unique	a#a	a0a, a1a, a2a	Aaa, a10a
[...-...]	Plage de caractères	[a-z]	f, p, j	2, _
[!...-...]	En dehors d'une plage	[!a-z]	9, _	b, a
	Pas un chiffre	[!0-9]	A, a, &, ~	0, 1, 9

Procédure de recherche

Pour trouver des objets correspondant à votre critère de recherche, suivez les étapes :

Etape	Action
1	Ouvrez la boîte de dialogue Trouver avec la commande **Vue** →**Trouver**.
2	Choisissez le type d'objet que vous recherchez dans la liste déroulante.
3	Saisissez votre premier critère de recherche en renseignant les champs **Attribut**, **Opérateur**, **Critère**.
4	Ajoutez une nouvelle ligne à la table des critères de recherche (clic droit dans la table et clic gauche sur **Nouveau**).
5	Répétez les étapes 3 et 4 jusqu'à ce que vous ayez saisi tous vos critères de recherche.
6	Combinez les critères de recherche par AND ou OR en renseignant la colonne `And/Or`.
7	Lancez la recherche en appuyant sur **Trouver**.

Résultat

Tous les objets satisfaisant aux critères de recherche s'affichent dans la fenêtre des messages.

NOTE : A partir de la fenêtre des messages, vous pouvez naviguer directement vers certains objets par un double-clic sur l'entrée souhaitée.

Utilisation de la liste d'instruments

Liste d'instruments

La liste d'instruments est une liste brute des instruments (modules de commande). C'est un dispositif permettant à l'ingénieur procédé de saisir les instruments **avant** de définir l'ensemble de la hiérarchie du modèle physique. Après avoir défini la hiérarchie jusqu'au niveau de l'élément physique fonctionnel, les instruments sont simplement déplacés et déposés dans le modèle physique et deviennent des modules de commande.

Voir également : *Instruments, page 81*

NOTE : L'utilisation de la liste d'instruments constitue donc un autre moyen de créer les modules de commande dans l'arborescence du modèle physique. Par comparaison avec la méthode normale, l'utilisation d'une liste d'instruments peut améliorer votre productivité.

Fonctionnalités avantageuses

La liste d'instruments vous permet d'effectuer les actions suivantes :

- Créer très simplement de multiples objets, par exemple 100 moteurs équivalents en une seule fois.
- Glisser et déposer simplement le ou les instruments sélectionnés sur le nœud d'élément physique fonctionnel cible dans le modèle physique
- Importer des instruments depuis le schéma PID (tuyauterie et instrumentation)

Que peut-on définir pour un instrument ?

A l'aide de la liste d'instruments, il est possible de définir des modules de commande avec toutes leurs propriétés sauf la propriété `API`.

Pour chaque instrument, vous pouvez définir les propriétés suivantes :

- Nom
- Description
- Type de module de commande
- Commentaires
- Documents...
- Propriétés spécifiques au type de module de commande
- Variables

NOTE : Il n'est pas possible de définir des commandes de forçage pour un instrument. Cela n'est possible qu'après avoir déplacé l'instrument vers le modèle physique.

Utilisation

Le tableau ci-dessous vous explique comment utiliser la liste d'instruments :

Si vous voulez...	Alors...
Importer des instruments depuis le schéma PID	Voir *Exemple : Importation d'instruments / de la hiérarchie du modèle physique, page 408*
Ouvrir la liste d'instruments	Appuyez sur le bouton Liste d'instruments ou utilisez le menu contextuel du **site**.
Créer un nouvel instrument ou Créer plusieurs instruments équivalents	**1.** Cliquez avec le bouton droit n'importe où dans la boîte de dialogue. **2.** Choisissez **Nouveau** dans le menu contextuel. **3.** Saisissez le `Type`, le `Nom` et la `Description` du ou des instruments. **4.** En option : Saisissez le `Numéro de départ` et le `Nombre` d'instruments équivalents à créer. **5.** Appuyez sur **Appliquer**. Résultat : Les onglets de propriété correspondant au type de module de commande sélectionné apparaissent. Le ou les nouveaux instruments apparaissent dans la liste d'instruments. **6.** Saisissez les propriétés en vous déplaçant parmi les onglets. **7.** Validez par **OK**.
Ouvrir les propriétés d'un instrument	Cliquez avec le bouton droit n'importe où sur la ligne puis choisissez **Propriétés** dans le menu contextuel.
Ouvrir les variables d'un instrument	Cliquez avec le bouton droit n'importe où sur la ligne puis choisissez **Ouvrir variables** dans le menu contextuel.
Copier, coller, supprimer un instrument	Cliquez avec le bouton droit n'importe où sur la ligne puis choisissez l'option désirée dans le menu contextuel.
Déplacer ou copier un ou plusieurs instruments vers le modèle physique	**1.** Sélectionnez le ou les instruments à déplacer. **2.** Pour déplacer : Faites glisser le ou les instruments sélectionnés vers le nœud d'élément physique fonctionnel cible du modèle physique. Pour copier : Appuyez sur la touche **Ctrl** tout en faisant glisser le ou les instruments sélectionnés vers le nœud d'élément physique fonctionnel cible du modèle physique. **Résultat** : Avant de coller, Unity Application Generator vérifie si les noms sont uniques et si le nombre maximum de 32 modules de commande dans un même élément physique fonctionnel est dépassé ou non. Si le collage est autorisé, les instruments sont intégrés au modèle physique.

Restrictions

La liste d'instruments fait l'objet des restrictions suivantes :

- **Type de module de commande** :
  - Il est possible de créer un instrument sans définir de type de module de commande (***type_module_commande*=**`Non_affecté`). Vous pouvez alors faire glisser l'instrument sans type vers le modèle physique sans restriction. Dans le modèle physique, il deviendra un module de commande sans type.
  - Après avoir affecté le type de module de commande d'un nouvel instrument avec **Appliquer** dans la boîte de dialogue **Nouveau**, vous **ne pouvez plus** changer le type. Si vous voulez changer le type de module de commande, supprimez l'instrument en question et créez-en un nouveau portant le type de module de commande correct.
- **Edition des propriétés** : Il est impossible d'éditer les propriétés des instruments multiples. Cette opération est uniquement réalisable dans la boîte de dialogue **Nouveau** lors de la création de plusieurs instruments équivalents.
- **Nombre maximum de modules de commande par élément physique fonctionnel** : Il est impossible de faire glisser un module de commande depuis la liste d'instruments vers le nœud d'élément physique fonctionnel cible du modèle physique si l'élément physique fonctionnel contient déjà le nombre maximum de modules de commande. Vous devez dans ce cas supprimer un ou plusieurs modules de commande dans l'élément physique fonctionnel ou créer un nouvel élément physique fonctionnel.
- **Unicité des noms** : Il est impossible de glisser des modules de commande du même nom dans un élément physique fonctionnel. Le caractère unique des noms d'un élément physique fonctionnel est vérifié avant de terminer l'opération glisser-déposer. Modifiez les noms avant le glissement.

Copier et coller dans les modèles physique et topologique

Fonction d'édition Copier et coller

Bien souvent, le concepteur créé des structures similaires dans le modèle physique et dans le modèle typologique du procédé. Pour une conception de procédé efficace avec Unity Application Generator, vous pouvez copier et coller des objets des modèles physique et topologique. Il est également possible de copier et coller des objets dans un même projet ou entre plusieurs projets.

Copier et coller dans le modèle physique

Vous pouvez copier des objets du modèle physique. Par exemple, il est possible de copier une zone complète avec l'ensemble de la hiérarchie.

Selon les types d'objets copiés, l'option de menu Coller est disponible uniquement pour les objets pouvant contenir les types d'objets copiés. Un élément physique fonctionnel copié peut être collé uniquement vers un autre élément physique d'un niveau physique supérieur dans la hiérarchie.

Lorsque vous essayez de coller un objet copié, Unity Application Generator ouvre une fenêtre comportant la liste des éléments contenus dans l'objet.

Lorsqu'une coche apparaît dans la première colonne d'un élément listé, cela indique que le nom de cet élément n'est pas déjà utilisé.

Chaque nom devant être unique, tous les éléments qui ne possèdent pas de coche dans la première colonne doivent être renommés pour pouvoir être copiés. Unity Application Generator ne colle le nouvel objet que lorsque tous les éléments listés sont cochés.

NOTE : Lors d'une copie, les références au modèle topologique sont supprimées. Cela signifie, par exemple, que les éléments physiques fonctionnels ou les modules de commande ne sont pas affectées à un API ou à une IHM ou que les variables E/S_API ne sont plus cartographiées au niveau des modules E/S (il est impossible de cartographier plusieurs variables dans le même point d'E/S d'un même module E/S). Ces références doivent être ajoutées manuellement après l'opération de copier-coller. Pour plus d'efficacité, vous devez utiliser l'analyseur.

Copier et coller dans le modèle topologique

Dans le modèle topologique, vous pouvez copier et coller des API et des châssis.

Lorsque vous essayez de coller un objet copié, Unity Application Generator ouvre une fenêtre comportant la liste des éléments contenus dans l'objet.

Lorsqu'une coche apparaît dans la première colonne d'un élément listé, cela indique que le nom de cet élément n'est pas déjà utilisé.

Tous les éléments qui ne possèdent pas de coche dans la première colonne doivent être renommés pour pouvoir être copiés et les adresses réseau doivent également être affectées pour la copie. Unity Application Generator ne colle le nouvel objet que lorsque tous les éléments listés sont cochés.

NOTE : Lors d'une copie, les références au modèle physique sont supprimées. Cela signifie, par exemple, que l'élément physique fonctionnel ou les modules de commande ne sont pas affectées à un API ou à une IHM ou que les variables E/S_API ne sont plus cartographiées au niveau des modules E/S (il est impossible de cartographier plusieurs variables dans le même point d'E/S d'un même module E/S). Ces références doivent être ajoutées manuellement après l'opération de copier-coller. Pour plus d'efficacité, vous devez utiliser l'analyseur.

Copier et coller entre plusieurs projets

Il est possible de copier et de coller des objets d'un projet à un autre si la personnalisation est identique. Dans ce cas de figure, les mêmes principes que ceux décrits ci-dessus s'appliquent. Cela implique souvent moins d'effort étant donné que les objets copiés peuvent garder le même nom si celui-ci n'existe pas déjà dans le projet cible.

NOTE : Unity Application Generator permet d'ouvrir un seul projet à la fois. Si vous avez besoin d'ouvrir deux projets en même temps pour copier et coller des objets, lancez deux fois Unity Application Generator sur votre PC.

Etablissement d'une définition de forçage

Présentation

La procédure suivante explique étape par étape comment établir une définition de forçage pour un module de commande.

Reportez-vous à : *Forçages des modules de commande, page 82* et Manuel avancé de UAG, code généré : éléments physiques fonctionnels, modules de commande et forçages *(voir Unity Application Generator (UAG) 3.3 SP4, a SoCollaborative software, Manuel utilisateur avancé)*.

Editeur graphique de forçages

Pour une description détaillée, reportez-vous à la section *Editeur graphique de forçages, page 277*.

Procédure d'établissement d'une définition de forçage

Pour établir une définition de forçage d'un module, effectuez les étapes suivantes :

Etape	Action
1	Dans le **modèle physique**, cliquez avec le bouton droit de la souris sur le module de commande à forcer.
2	Sélectionnez **Ouvrir forçage** dans le menu contextuel. **Résultat** : l'éditeur graphique de forçage de ce module de commande s'affiche.
3	Dans la liste **Entrées**, sélectionnez la broche de forçage souhaitée.
4	Dans la barre d'outils graphiques, sélectionnez les blocs fonctions (FFB) nécessaires pour construire votre logique de forçage.
5	Reliez les sorties et les entrées des blocs fonctions au moyen de liaisons.
6	Associez des expressions littérales ou des variables aux broches d'entrée des FFB. **Remarque** : ● Pour associer des variables aux broches d'entrée, vous pouvez utiliser soit une sélection de données, soit effectuer une opération de glissement de variables entre l'afficheur de variables et l'éditeur de forçage graphique. ● les variables appartenant à un autre API peuvent être configurées comme variables de condition. Afin que la génération s'effectue sans problème, il faut que cette variable soit communiquée à l'API du forçage par un canal. ● Si vous devez utiliser des fonctions telles que SIN, COS etc. voir ci-dessous.
7	Sélectionnez la broche de sortie du bloc fonction situé à la fin de votre logique, et utilisez la commande **Connecter à une broche de forçage** à partir du menu contextuel. **Remarque :** le type de données de la broche de sortie doit être BOOL.
8	Ajoutez un **commentaire** à votre définition de forçage.
9	Cliquez sur le bouton **Appliquer**. **Résultat :** la logique est analysée et les erreurs détectées sont affichées dans l'onglet **Messages**.

Etape	Action
10	Corrigez les erreurs détectées et cliquez à nouveau sur le bouton **Appliquer**. **Résultat :** si aucune erreur n'est détectée, la logique analysée est affichée dans l'onglet **Définition du forçage**. **Remarque :** l'onglet **Définition du forçage** est en lecture seule.
11	Fermez l'éditeur graphique de forçages.

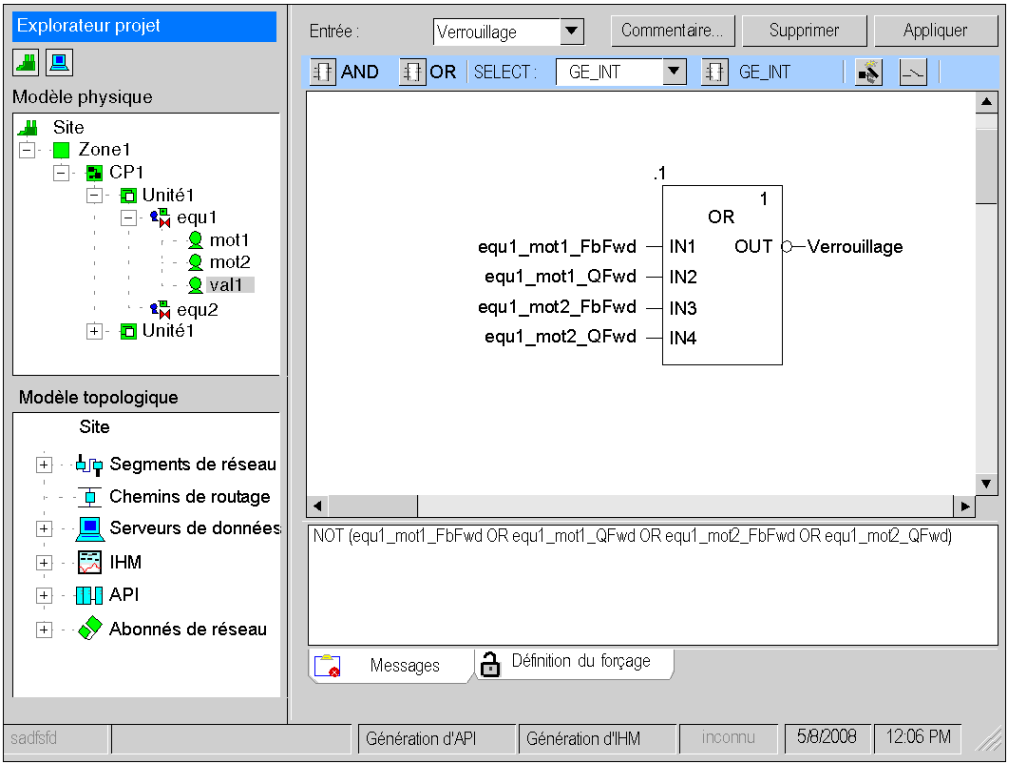
Fonctions telles que SIN ou COS dans une définition de forçage

Vous pouvez dans certains cas avoir besoin de fonctions complexes telles que SIN ou COS dans votre définition du forçage. Vous devez configurer ces fonctions directement dans Unity Pro car Unity Application Generator n'autorise que les blocs fonctions pour les opérateurs logiques et de comparaison. La solution consiste à définir une variable libre comme paramètre fictif pour la fonction complexe et à ajouter la logique complexe après génération directement sous Unity Pro :

Etape	Action	Exemple
1	Définissez une variable libre dans l'élément physique fonctionnel. **Remarque** : cette variable libre doit être le paramètre fictif pour la fonction complexe dans la condition de forçage.	Vous voulez utiliser la variable ***angle*** comme argument de la fonction SIN, SIN(***angle***) dans une condition de forçage. Définissez une variable libre portant le nom SINANGLE et qui sera le paramètre fictif pour SIN(***angle***).
2	Déposez cette variable libre sur une broche de bloc fonction pour laquelle la fonction complexe est nécessaire.	Définissez la condition de forçage numéro 1 : `SINANGLE=1`
3	Après génération par Unity Pro, ajoutez votre programme complexe à la section Unity Pro correspondante et connectez-la par la broche de variable libre du bloc fonction généré.	Intégrez un bloc fonction SIN à la section correspondante. Définissez ***angle*** comme entrée et connectez la sortie sur la variable libre SINANGLE.

Exemple : définition du forçage

Les captures d'écran illustrent une définition de forçage dont la syntaxe est correcte.



Utilisation de l’afficheur topologique

Afficheur topologique

L’afficheur topologique fait partie de Unity Application Generator. L’afficheur topologique permet de représenter le modèle topologique graphiquement dans un écran distinct sous Unity Application Generator. La vue topologique est créée à partir du contenu du modèle topologique.

L’afficheur topologique vous offre une vue d’ensemble de l’état actuel de la configuration matérielle de l’automatisme de votre projet à tout instant.

Caractéristiques

Comparée à l’arborescence du modèle topologique, la vue topologique affiche plus d’informations sur les relations entre les composants matériels ainsi que plus de détails tels que par exemple les emplacements libres dans un API sous forme graphique.

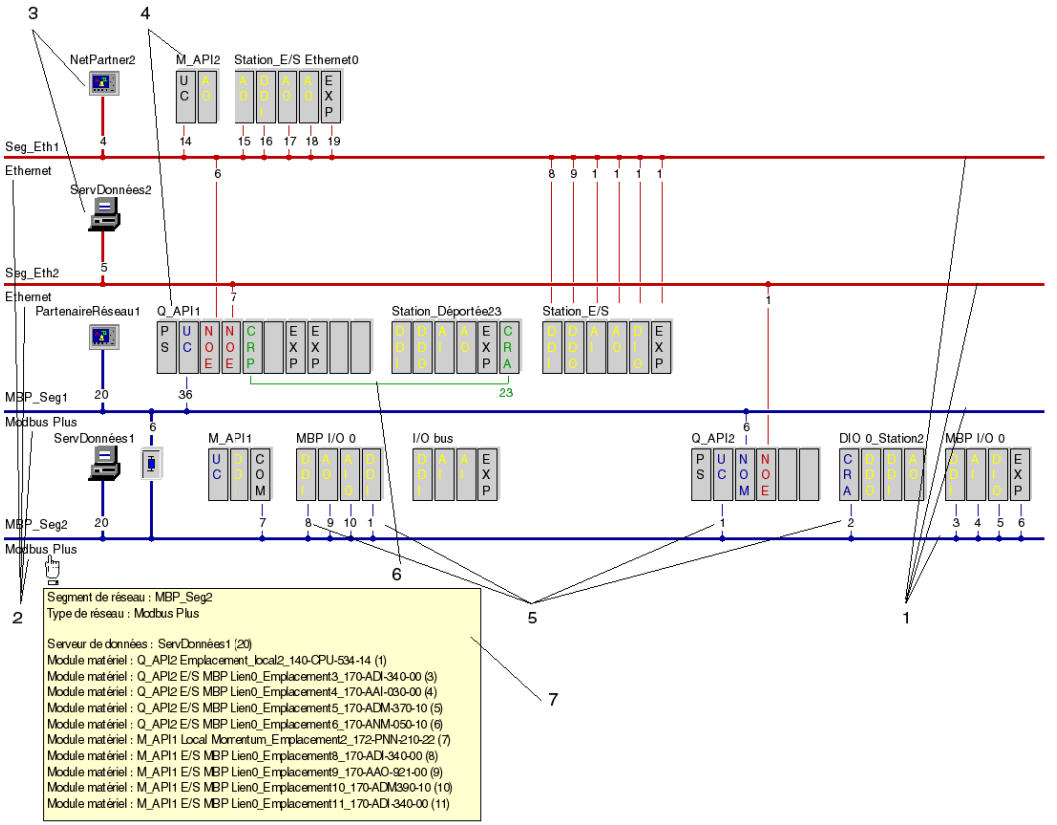
Les fonctionnalités suivantes seront très intéressantes pour votre travail :

- Vous pouvez enregistrer la vue topologique sous forme d’image bitmap et l’inclure dans un fichier de rapport Word.
- La vue topologique est synchronisée avec le modèle topologique.
- Vous pouvez utiliser la vue topologique pour naviguer vers les boîtes de dialogue de propriétés.
- Après avoir changé quelque chose dans la configuration, la vue topologique est actualisée dès que la fenêtre de l’afficheur topologique est réactivée.
- La fenêtre de l’afficheur topologique peut être ouverte en même temps que d’autres boîtes de dialogue.

NOTE : L’afficheur topologique n’est **pas** un éditeur graphique. Il n’est pas possible de créer de nouveaux objets tels que des API, des modules matériels, etc. ou de les supprimer. Ceci doit être effectué à l’aide des menus de l’arborescence du modèle topologique.

Exemple de vue topologique

Vue topologique d'un exemple de projet :



1 La vue topologique s'appuie sur les segments de réseau qui sont représentés en lignes horizontales dans différentes couleurs. La couleur indique le type de protocole (Modbus Plus ou Ethernet).

2 Les segments de réseau sont identifiés avec le nom et le type de protocole.

3 Les API, serveurs de données, abonnés réseau ou les chemins de routage se connectent sur les segments de réseau.

4 Un API est constitué de châssis et de modules matériels. Les châssis sont représentés sous forme de case montrant l'emplacement avec les modules matériels et, en mode d'affichage complet, les emplacements vides restants. Les modules matériels sont identifiés par une abréviation correspondant à la catégorie de module. Les modules de communication se connectent sur les segments de réseau correspondants.

5 Numéro de liaison : le numéro de liaison est l'adresse Modbus Plus, l'adresse Ethernet ou le chemin de communication (pour les objets du chemin de routage). Seul le dernier octet des adresses Ethernet est affiché. Pour les chemins de communication, seul le premier octet est affiché. En déplaçant la souris sur l'adresse abrégée, l'adresse s'affiche dans son intégralité.

6 Les châssis déportés sont connectés par un trait fin entre la tête CRP et la station CRA. La connexion à un module CRA est repérée par le numéro du châssis.

7 Une fenêtre contenant des informations sur l'objet s'ouvre lorsque l'on déplace la souris sur celui-ci.

Utilisation

Le tableau ci-dessous vous explique comment utiliser l'afficheur topologique :

Si vous voulez...	Alors...
Passer l'affichage à ● complet : affichage de tous les emplacements des API ● compact : affichage des seuls emplacements utilisés ● communication : affichage seulement de la configuration matérielle de la communication	Servez-vous du menu contextuel en cliquant n'importe où dans la fenêtre de l'afficheur topologique.
Voir plus de détails sur un objet	Déplacez la souris sur l'objet symbolisé. **Résultat** : une fenêtre contenant des informations détaillées sur l'objet s'ouvre à côté du pointeur de la souris. Remarque : si vous cliquez une fois sur l'objet symbolisé, cet objet est sélectionné dans l'arborescence du modèle topologique.
Déplacer un objet invisible vers une zone visible de la vue topologique	Sélectionnez l'objet désiré dans l'arborescence du modèle topologique. Remarque : l'objet sélectionné passe dans la zone visible de l'affichage et son nom s'affiche sur fond jaune.
Edition d'un objet	Clic droit sur le symbole de l'objet ou sur son nom ; le menu contextuel s'ouvre alors. Double clic sur le symbole de l'objet ou sur son nom ; la boîte de dialogue des propriétés s'ouvre alors. **Remarque** : si vous saisissez maintenant de nouvelles données dans la boîte de dialogue des propriétés, celles-ci seront automatiquement appliquées dans l'affichage topologique.
Enregistrer la vue topologique comme bitmap	Cliquez avec le bouton droit n'importe où dans la vue topologique, sélectionnez **Créer une bitmap...** puis saisissez le nom du fichier cible de l'image bitmap.
Intégrer la vue topologique dans un fichier rapport MS Word	Dans la boîte de dialogue **Rapport**, sélectionnez la vue topologique.

Comment visualiser l'état de la génération

Présentation

Vous pouvez visualiser à tout instant les objets qui ont été modifiés après la dernière génération et qui doivent être régénérés. Cette possibilité est nécessaire pour conserver la cohérence du projet, par exemple si vous avez généré des API individuels à des fins de test ou pour d'autres raisons.

Visualisation de l'état de la génération

Possibilités de visualiser des informations relatives à l'état de la génération :

Si vous souhaitez savoir ...	**Alors ...**
Comment le projet a été généré (nouveau ou incrémental) ?	Visualisez l'historique des changements du projet avec **Fichier** →**Historique des changements**
Quels sont les objets devant encore être générés pour l'API ?	Ouvrez la liste des objets à générer avec **Fichier** →**Propriétés** →**Objets à générer** →**API**
Quelles sont les IHM devant être générées ?	Ouvrez la liste des IHM à générer avec **Fichier** → **Propriétés** →**Objets à générer** →**IHM**
Quels sont les serveurs de données devant être générés ?	Ouvrez la liste des serveurs de données à générer avec **Fichier** →**Propriétés** →**Objets à générer** →**Serveurs de données**

Comment échanger les noms

Échange des noms dans un modèle physique

Vous ne pouvez pas échanger en 1 étape, les noms de 2 éléments physiques fonctionnels, modules de commande et variables libres. Pour échanger les noms de 2 éléments physiques fonctionnels, procédez comme suit :

Etape	Action
1	Créez 2 éléments physiques fonctionnels, par exemple , **EQ1** et **EQ2** dans le modèle physique. Cliquez sur **Générer** →**API**. **Résultat : EQ1** et **EQ2** sont générés dans Unity Pro.
2	Donnez au premier élément physique fonctionnel un nom provisoire, par exemple **EQ1** devient **EQ1x**. Cliquez sur **Générer** →**API**. **Résultat : EQ1x** est généré dans Unity Pro en remplacement de **EQ1**.
3	Donnez au second élément physique fonctionnel le nom précédent du premier module, c'est-à-dire passez de **EQ2** à **EQ1**. Cliquez sur **Générer** →**API**. **Résultat : EQ1** est généré dans Unity Pro en remplacement de **EQ2**.
4	Donnez au premier élément physique fonctionnel le nom précédent du second module, c'est-à-dire passez de **EQ1x** à **EQ2**. Cliquez sur **Générer** →**API**. **Résultat : EQ2** est généré dans Unity Pro en remplacement de **EQ1x**.

Vous pouvez échanger de la même manière les noms de 2 modules de commande et variables libres.

NOTE : La suppression et la création de variables portant le même nom ne sont pas autorisées pour la génération incrémentielle d'IHM.

Exemple :

Étape	Action
1	Créez une variable portant le nom de **VAR1** dans le module de commande. Cliquez sur **Générer** →**IHM**. **Résultat : VAR1** est générée dans Vijeo Citect.
2	Copiez et collez la variable **VAR1** en tant que **VAR2** dans le même module de commande.
3	Supprimez la variable **VAR1** et renommez la variable **VAR2 VAR1**. Cliquez sur **Générer** →**IHM**. **Résultat : VAR1** n'est pas générée dans Vijeo Citect, car une variable portant ce nom est déjà générée.

Échange des noms dans un modèle topologique

Vous ne pouvez pas échanger en 1 étape, les noms de 2 segments de réseau, serveurs de données, IHM, API, zones de conduite, images, groupes d'images et points de raccordement réseau. Pour échanger les noms de 2 IHM, procédez comme suit :

Etape	Action
1	Créez 2 IHM, par exemple, **IHMA** et **IHMB** dans le modèle topologique. Cliquez sur **Générer →IHM**. **Résultat : IHMA** et **IHMB** sont générés dans Vijeo Citect.
2	Donnez au premier IHM un nom provisoire, par exemple **IHMA** devient **IHMAx**, et cliquez sur **Générer →IHM**. **Résultat : IHMAx** est généré dans Vijeo Citect en remplacement de **IHMA**.
3	Donnez au second IHM le nom précédent du premier IHM, c'est-à-dire passez de **IHMB** à **IHMA**, et cliquez sur **Générer →IHM**. **Résultat : IHMA** est généré dans Vijeo Citect en remplacement de **IHMB**.
4	Donnez au premier IHM le nom précédent du second IHM, c'est-à-dire passez de **IHMAx** à **IHMB**, et cliquez sur **Générer →IHM**. **Résultat : IHMB** est généré dans Vijeo Citect en remplacement de **IHMAx**.

Vous pouvez échanger de la même manière les noms de 2 segments de réseau, serveurs de données, API, zones de conduite, images, groupes d'images et points de raccordement réseau.

Vous ne pouvez pas échanger en 1 étape 2 modules matériels (emplacements et numéros de station). Pour échanger les numéros de station déportée dans un API, procédez comme suit :

Etape	Action
1	Créer un API, par exemple, **140-CPU-651-50**. Attribuez le segment de réseau et l'adresse IP.
2	Ouvrez le châssis et saisissez le **Coupleur maître**, par exemple **140-CRP-93x-00**.
3	Créez le premier châssis avec **Type de liaison**, **Numéro de station**, **Type de châssis**, et **Coupleur esclave**, par exemple **Station_déportée 3** avec le module **1-140CRA-93x-00**. Cliquez sur **Générer →API**. **Résultat :** Dans Unity Pro, sous **RIO bus**, **3 : la station Quantum d'E/S déportée** est générée.
4	Créez le second châssis avec **Type de liaison**, **Numéro de station**, **Type de châssis**, et **Coupleur esclave**, par exemple **Station_déportée 4** avec le module **1-140CRA-93x-00**. Cliquez sur **Générer →API**. **Résultat :** Dans Unity Pro, sous **RIO bus**, **4 : la station Quantum d'E/S déportée** est générée.

Etape	Action
5	Attribuez le chiffre 2 au **Numéro de station** du premier châssis, autrement dit passez de **Station_déportée 3** à **Station_déportée 2**. Cliquez sur **Générer** →**API**. **Résultat :** Dans Unity Pro, sous **RIO bus**, **2 : la station Quantum d'E/S déportée** est générée en remplacement de **3 : station Quantum d'E/S déportée**.
6	Attribuez le chiffre 3 au **Numéro de station** du second châssis, autrement dit passez de **Station_déportée 4** à **Station_déportée 3**. Cliquez sur **Générer** →**API**. **Résultat :** Dans Unity Pro, sous **RIO bus**, **3 : la station Quantum d'E/S déportée** est générée en remplacement de **4 : station Quantum d'E/S déportée**.
7	Attribuez le chiffre 4 au **Numéro de station** du premier châssis, autrement dit passez de **Station_déportée 2** à **Station_déportée 4**. Cliquez sur **Générer** →**API**. **Résultat :** Dans Unity Pro, sous **RIO bus**, **4 : la station Quantum d'E/S déportée** est générée en remplacement de **2 : station Quantum d'E/S déportée**.

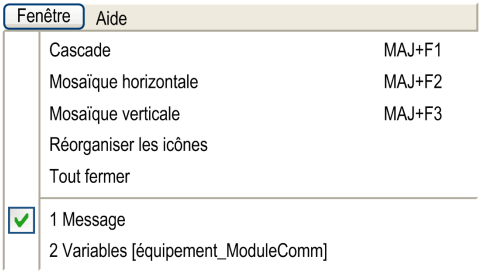
L'échange est possible pour tous les types de châssis et de modules matériels d'un API.

Fenêtre - Raccourcis clavier

Raccourcis clavier

Sur UAG V3.3 SP4, le menu **Fenêtres** est mis à jour à l'aide de raccourcis clavier qui permettent de réorganiser plusieurs fenêtres ouvertes. L'utilisateur peut ouvrir les fenêtres en cascade et toutes les disposer en mosaïque, horizontalement et verticalement.

La figure ci-dessous montre le menu **Fenêtres** avec les raccourcis clavier :



Ce qui suit sont des exemples d'utilisation des raccourcis clavier pour réorganiser les fenêtres ouvertes :

Raccourcis clavier	Exemple
MAJ + F1	Ouvre toutes les fenêtres ouvertes en cascade
MAJ + F2	Ouvre toutes les fenêtres ouvertes en mosaïque, horizontalement
MAJ + F3	Ouvre toutes les fenêtres ouvertes en mosaïque, verticalement

Sous-chapitre 7.2
Editeur graphique de forçages

Introduction

Cette section décrit l'éditeur graphique de forçages.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Description générale	278
Blocs fonctions (FFB)	283
Liaisons	287
Barre d'outils graphiques	289
Zone de dessin	293
Analyse logique et syntaxique de la logique de forçage	297
Enregistrement et récupération de la logique de forçage	299

Description générale

Présentation

Pour les modules de commande comportant des entrées de forçage (par exemple les moteurs, vannes, commandes d'axe ou pompes), il est possible de définir des commandes de forçage dans le modèle physique.

Un module de commande peut posséder une ou plusieurs entrées de forçage. Ces entrées sont définies comme broches de forçage dans les SCoD.

L'éditeur graphique de forçages facilite la conception de la logique des broches de forçage, de manière graphique par

- le positionnement de blocs fonctions (FFB) dans une zone de dessin ;
- la connexion d'entrées et de sorties des blocs fonctions au moyen de liaisons ;
- l'association d'expressions littérales ou de variables aux broches d'entrée des FFB ;
- l'inversion des broches des FFB.

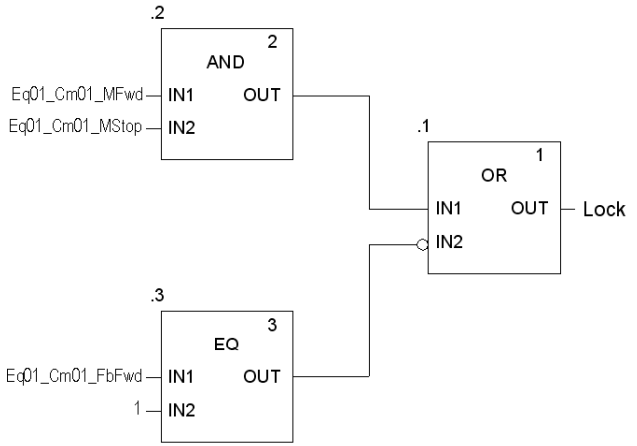
L'éditeur graphique de forçages analyse la logique dessinée, affiche les messages des erreurs détectées et génère une définition de forçage si la logique est correcte.

Exemple de définition de forçage :

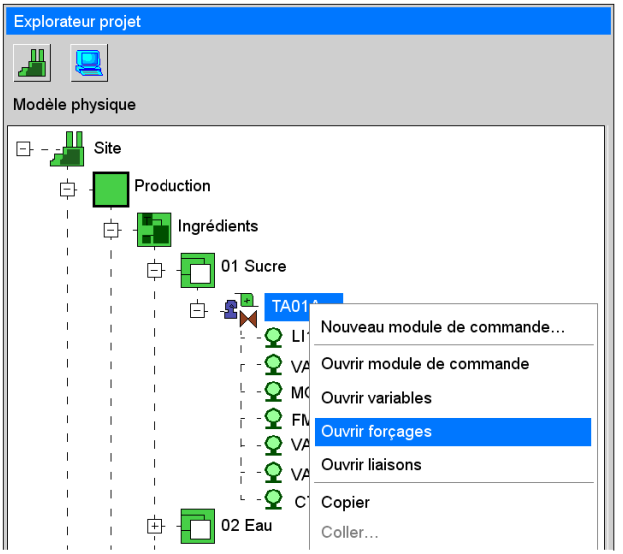
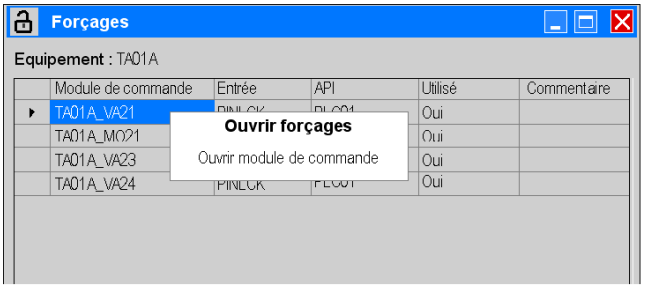
```
(((Eq01_cm01_MFwd=1)AND(Eq01_cm01_MStop=1))OR((Eq01_cm01_FbFwd=1)AND(Eq
01_cm01_Ack=1))).
```

Exemple

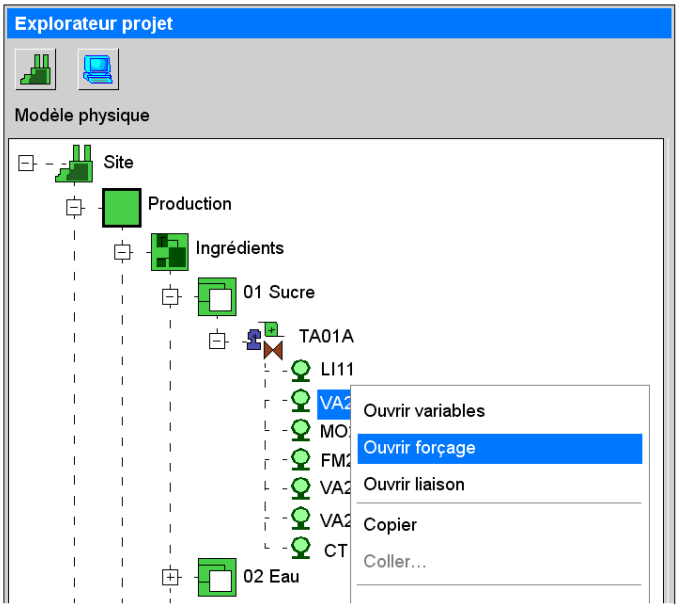
Logique de forçage dessinée pour la broche `Verrouillage`.



Accès au moyen d'une liste de forçages

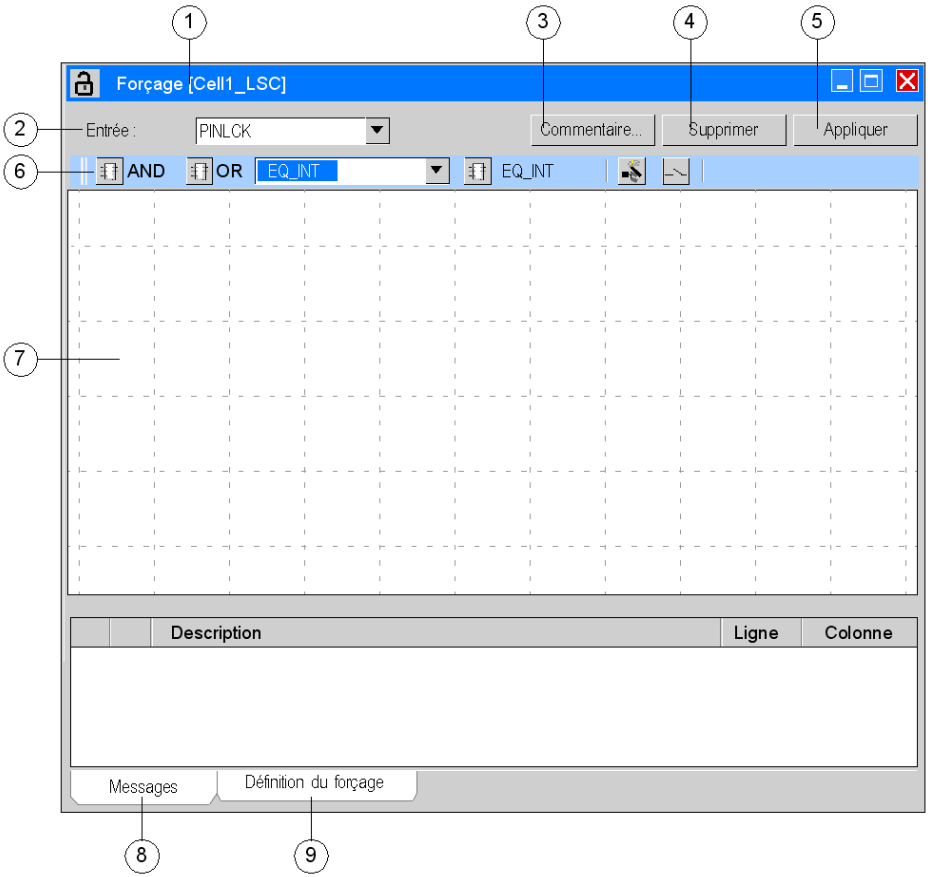
Etape	Action
1	Dans le **modèle physique**, sélectionnez un élément physique situé au-dessus du niveau des modules de commande. 
2	Cliquez avec le bouton droit de la souris et sélectionnez **Ouvrir forçages** dans le menu contextuel, ou utilisez le bouton de la barre d'outils. **Résultat :** La liste des **forçages** associés s'affiche. 
3	Sélectionnez le forçage que vous souhaitez modifier, double-cliquez ou cliquez avec le bouton droit et sélectionnez **Ouvrir forçages** dans le menu contextuel. **Résultat :** L'éditeur graphique de forçages s'affiche.

Accès au moyen d'un module de commande

Etape	Action
1	Dans le **modèle physique**, sélectionnez un module de commande comportant une entrée de forçage. 
2	Cliquez avec le bouton droit de la souris et sélectionnez **Ouvrir forçages** dans le menu contextuel, ou utilisez le bouton de la barre d'outils. **Résultat :** L'éditeur graphique de forçages s'affiche.

Structure générale

Editeur graphique de forçages



Elément	Nom	Description
1	Légende	Affiche le nom complet du module de commande entre crochets
2	**Entrée**	Répertorie l'ensemble des broches de forçage du module de commande
3	Bouton **Commentaire**	Ouvre une boîte de dialogue destinée à commenter la logique de forçage
4	Bouton **Supprimer**	Supprime la logique de forçage associée à la broche sélectionnée

Elément	Nom	Description
5	Bouton **Appliquer**	Valide et associe la logique associée à la broche sélectionnée
6	Barre d'outils graphiques	Permet de ● sélectionner les blocs fonctions AND et OR ● sélectionner d'autres blocs fonctions au moyen d'une liste d'opérateurs ● sélectionner le dernier bloc fonction choisi dans cette liste d'opérateurs ● ouvrir une boîte de dialogue de sélection de données ● réaliser une connexion avec la broche de forçage
7	Zone de dessin	Zone destinée à la création de votre logique de forçage Si la logique excède en dimensions la zone de dessin visible, des barres d'ascenseur s'affichent.
8	Onglet **Messages**	Répertorie les messages des erreurs détectées, identifiés dans la logique de forçage une fois que vous avez cliqué sur le bouton **Appliquer**
9	Onglet **Définition du forçage**	Affiche la définition de forçage analysée, dérivée de la logique conçue graphiquement

Blocs fonctions (FFB)

Présentation

L'éditeur graphique de forçages propose les éléments graphiques suivants :

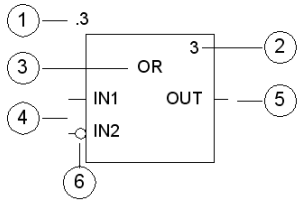
- blocs fonctions (FFB)
- liaisons (reportez-vous à la section *Liaisons, page 287*)

Blocs fonctions, généralités

Les blocs fonctions comportent :

- des entrées du côté gauche
  Les entrées d'un bloc fonction possèdent des types de données prédéfinis. Vous pouvez associer des valeurs littérales ou des variables à des broches d'entrée. Vous pouvez connecter une liaison à une entrée, si le type de données correspond.
- une sortie du côté droit
  Le type de données de la sortie d'un bloc fonction pris en charge par l'éditeur graphique de forçages sera BOOL. Vous pouvez connecter une liaison à une sortie ou connecter la sortie à la broche de forçage du module de commande correspondant.

Représentation d'un bloc fonction



Elément	Nom/exemple	Description
1	.3	Nom d'instance (numéro d'identification du bloc fonction dans la liste des blocs fonctions placés dans la zone de dessin) Le nom d'instance dépend de la séquence selon laquelle le bloc fonction est créé dans l'éditeur graphique de forçages. Le nom d'instance est utilisé dans l'onglet de messages pour indiquer les erreurs détectées éventuelles associées au bloc fonction.
2	3	Ordre d'exécution du bloc fonction
3	OR	Type de bloc fonction
4	IN1, IN2	Entrées
5	OUT	Sortie
6	petit cercle sur la broche	Broche inversée

Broche inversée

Un petit cercle sur une broche indique qu'elle est inversée. Cliquez avec le bouton droit sur la broche et sélectionnez **Inverser la broche** dans le menu contextuel.

Vous ne pouvez inverser que des broches dont le type de données est BOOL.

Blocs fonctions proposés

L'éditeur graphique de forçages propose les blocs fonctions suivants :

- Blocs fonctions logiques
  - AND
  - OR
  - XOR
- Blocs fonctions de comparaison
  - EQ_*
  - GT_*
  - GE_*
  - LT_*
  - LE_*
  - NE_*

  (* remplace le type de données pris en charge)

Type de données pris en charge

L'éditeur graphique de forçages prend en charge les types de données suivants :

- Pour les blocs fonctions logiques
  - BOOL
- Pour les blocs fonctions de comparaison
  - BOOL
  - BYTE
  - DATE
  - DINT
  - DT
  - DWORD
  - INT
  - REAL
  - STRING
  - TIME
  - TOD
  - UDINT
  - UINT
  - WORD

Redimensionnement d’un bloc fonction

Il est possible de redimensionner les blocs fonctions logiques (AND, OR et XOR) pour disposer de plus de deux broches d’entrée.

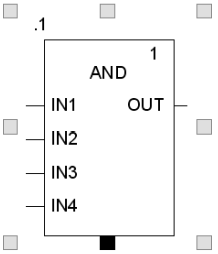
Le redimensionnement permet également de réduire le nombre de broches d’entrée d’un bloc fonction.

Le nombre maximal de broches d’entrée peut être porté à 32.

Le nombre minimal de broches d’entrée est de 2.

Le redimensionnement ne supprime pas les broches connectées. Les broches d’entrées qui sont connectées à des variables, à des valeurs littérales ou à des liaisons ne peuvent pas être supprimées par le redimensionnement. La réduction n’est donc autorisée que dans le cas de broches non reliées.

Pour redimensionner un bloc fonction, sélectionnez-le, puis déplacez verticalement le rectangle plein situé en bas du bloc. Vous ne pouvez pas modifier la largeur d’un bloc fonction.



NOTE : Il est impossible de redimensionner des blocs fonctions de comparaison et ces blocs fonctions ne prennent en charge que deux broches d’entrée par défaut.

Menu contextuel d’un bloc fonction

Cliquez avec le bouton droit de la souris sur un bloc fonction pour ouvrir son menu contextuel.

Le menu contextuel d’un bloc fonction propose les options suivantes :

- Couper
- Copier
- Coller
- Supprimer
- Remplacer bloc fonction

Menu contextuel d’une broche

Cliquez avec le bouton droit de la souris sur une broche d’un bloc fonction pour ouvrir son menu contextuel.

Le menu contextuel d’une broche propose les options suivantes :

- Couper
- Copier
- Coller
- Supprimer
- Inverser la broche
- Connecter à une broche de forçage
- Sélection de données

Les opérations telles que **Couper**, **Copier**, **Coller** et **Supprimer** n’ont aucune influence sur la broche, mais agissent sur la variable ou la valeur littérale associée à cette broche.

Ces options de menu sont désactivées si la broche n’est pas associée à une variable ou à une valeur littérale.

La fonction **Connecter à une broche de forçage** n’est disponible que pour la broche de sortie d’un bloc fonction. Elle associe la broche de forçage du module de commande à la broche de sortie du bloc fonction.

Liaisons

Présentation

L'éditeur graphique de forçages propose les éléments graphiques suivants :

- blocs fonctions (FFB), reportez-vous à la section *Blocs fonctions (FFB), page 283*
- des liaisons

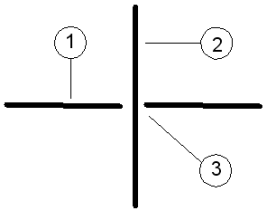
Liaisons, généralités

- Les liaisons sont des connexions entre des broches de blocs fonctions
- Les liaisons sont constituées de segments de lignes verticaux et horizontaux
- Les entrées et les sorties connectées doivent posséder des types de données correspondants
- Il est impossible de relier deux broches d'un bloc fonction en utilisant une liaison
- Il est impossible de connecter des broches de sortie en utilisant une liaison
- Une liaison permet de connecter la sortie d'un bloc fonction à l'entrée d'un autre bloc fonction
- Des liaisons peuvent faire appel à différents types de données, mais sont représentées de manière identique

NOTE : Il est impossible de connecter une liaison à une broche qui est déjà reliée à une variable ou à une valeur littérale.

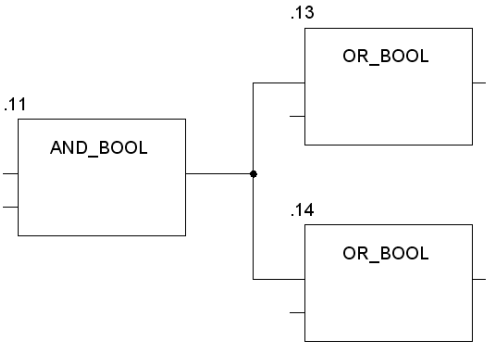
Représentation

Des liaisons qui ne sont pas connectées et qui se croisent sont affichées de la manière suivante. Le croisement d'une liaison avec une autre remplace la liaison du dessous par une séparation de liaison.



Elément	Description
1	Première liaison dessinée
2	Deuxième liaison dessinée
3	Séparation de liaison

Les liaisons issues d'une origine commune et qui se séparent ultérieurement comportent un point au niveau de leur séparation.



Mise en place d'une liaison

Pour mettre en place une liaison, procédez de la manière suivante.

Etape	Action
1	Cliquez sur la broche de sortie du bloc fonction d'origine.
2	Faites glisser le pointeur de la souris vers la broche d'entrée du bloc fonction de destination. **Résultat :** Une liaison est mise en place entre les deux broches. **Remarque :** il est également possible d'effectuer la procédure inverse.

Edition d'une liaison

Pour sélectionner une liaison, cliquez avec le bouton gauche sur celle-ci. Une liaison sélectionnée s'affcihe en rouge.

Pour supprimer une liaison sélectionnée, utilisez la touche **Suppr** ou sélectionnez **Supprimer** dans le menu contextuel.

NOTE : Il est impossible de déplacer des segments de liaison, donc de modifier la forme d'une liaison.

Barre d’outils graphiques

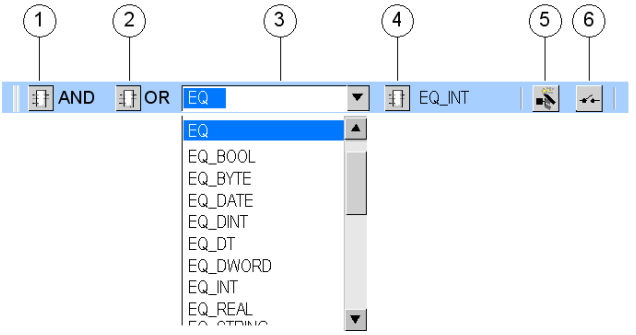
Présentation

La barre d’outils graphiques vous permet de :

- sélectionner les blocs fonctions AND et OR
- sélectionner d’autres blocs fonctions (FFB) au moyen d’une liste d’opérateurs
- sélectionner le dernier bloc fonction choisi dans cette liste d’opérateurs
- ouvrir une boîte de dialogue de sélection de données
- réaliser une connexion avec la broche de forçage

Représentation

Barre d’outils graphiques



Elément	Nom	Description
1	Bouton de bloc fonction AND	Sélectionne un bloc fonction AND
2	Bouton de bloc fonction OR	Sélectionne un bloc fonction OR
3	Sélection de bloc fonction	Zone de liste permettant de sélectionner tous les blocs fonctions disponibles
4	Dernier bouton de bloc fonction sélectionné	Sélectionne le dernier bloc fonction choisi
5	Sélection de données	Ouvre une boîte de dialogue de sélection de données
6	Connexion à un forçage	Connecte la broche de sortie sélectionnée à la broche de forçage du module de commande

Mise en place de blocs fonctions dans la zone de dessin

Pour mettre en place un bloc fonction, sélectionnez-le dans la barre d'outils graphiques et cliquez sur un espace libre dans la zone de dessin.

Vous ne pouvez pas recouvrir ou recouper un objet par un autre (bloc fonction ou liaison).

S'il n'y a pas suffisamment d'espace pour mettre en place un objet, le pointeur de la souris affiche un signe d'interdiction .

Boutons de blocs fonctions AND et OR

Les blocs fonctions `AND` et `OR` sont fréquemment utilisés. La barre d'outils graphiques comporte deux boutons qui permettent la sélection de ces blocs fonctions.

Zone de liste de sélection de blocs fonctions

Cette zone de liste permet de sélectionner tous les blocs fonctions disponibles

Les blocs fonctions sont répertoriés par ordre alphabétique.

Pour présélectionner les blocs fonctions répertoriés, vous pouvez, par exemple, saisir `EQ` dans la zone de liste. Tous les blocs fonctions contenant `EQ` dans leur nom apparaîtront dans la liste (`EQ`, `EQ_BOOL`, `EQ_BYTE` etc.).

Si vous saisissez le nom d'un bloc fonction qui n'est pas pris en charge, il s'affiche en rouge.

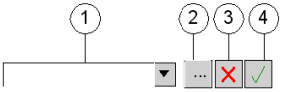
Dernier bouton de bloc fonction sélectionné

Le nom du dernier bloc fonction choisi apparaît juste à côté de ce bouton. Cliquez sur le bouton pour sélectionner le dernier bloc fonction choisi, puis cliquez sur un espace libre dans la zone de dessin pour mettre en place le bloc fonction.

Boîte de dialogue de sélection de données

Sélectionnez une broche d'entrée de bloc fonction et cliquez sur le bouton , ou double-cliquez sur une broche d'entrée de bloc fonction.

La boîte de dialogue de sélection de données apparaît alors à proximité de la broche sélectionnée.



Elément	Nom	Description
1	Champ de saisie, zone de liste	Champ permettant de saisir un nom de variable ou une valeur littérale à associer à la broche sélectionnée Si vous saisissez un nom de variable ou une valeur littérale non valides (par exemple, un type de données erroné) et que vous confirmez, un message correspondant s'affiche. La zone de liste ouverte au moyen de la flèche descendante affiche les noms de variables ou les valeurs littérales précédemment saisies pour l'ensemble des broches de cette session d'éditeur graphique de forçages, dans l'ordre d'utilisation la plus récente. Une nouvelle opération de fermeture et d'ouverture de l'éditeur graphique de forçage efface la liste.
2	Bouton d'afficheur de variables	Ouvre l'afficheur de variables Dans le filtre de l'afficheur de variables, le **Type de données**, nécessaire pour la broche sélectionnée, est prédéfini. Toutes les variables appropriées sont répertoriées et vous pouvez sélectionner celle que vous souhaitez, en confirmant par **OK**. Vous pouvez modifier les options **Domaine** et **Filtre**. Pour des informations plus détaillées, consultez la section *Afficheur de variables, page 300*.
3	Fermer la sélection de données	Ferme la sélection de données sans associer la variable ou la valeur littérale sélectionnée
4	Confirmer la sélection	Vérifie la validité du nom de la variable ou de la valeur littérale (type de données) et associe une variable ou une valeur littérale à la broche sélectionnée Si le nom de la variable (ou la valeur littérale) est valide, celle-ci est associée à la broche sélectionnée. Dans le cas contraire, un message approprié s'affiche.

Connexion à un forçage

Sélectionnez une broche de sortie de bloc fonction et cliquez sur le bouton , ou cliquez avec le bouton droit sur une broche de sortie de bloc fonction et sélectionnez l'option **Connecter à une broche de forçage** dans le menu contextuel.

La broche de sortie sélectionnée est connectée à la broche de forçage du module de commande.

Si la broche de forçage est déjà reliée à une broche de sortie et que vous connectez la broche de forçage à une autre broche de sortie, l'opération est effectuée sans nécessité d'une confirmation.

Zone de dessin

Présentation

La zone de dessin permet de réaliser les opérations suivantes sur les éléments graphiques (blocs fonctions, liaisons) :

- placer ;
- remplacer ;
- sélectionner ;
- déplacer ;
- copier ;
- couper ;
- coller ;
- supprimer ;
- glisser-déplacer ;

Menu contextuel

Cliquez avec le bouton droit de la souris dans un espace libre de la zone de dessin pour ouvrir son menu contextuel.

Le menu contextuel de la zone de dessin propose les éléments suivants :

- Coller
- Zoom
  - 25 %
  - 50 %
  - 75 %
  - 100 %
  - 200 %
  - Agrandissement
  - Réduction

Placer des objets

Vous pouvez placer des blocs fonctions et des liaisons dans la zone de dessin.

- Blocs fonctions
  Pour mettre en place un bloc fonction, sélectionnez-le dans la barre d'outils et cliquez sur un espace libre dans la zone de dessin.
  Vous ne pouvez pas recouvrir ou superposer un objet à un autre (bloc fonction ou liaison).
  S'il n'y a pas suffisamment d'espace pour mettre en place un objet, le pointeur de la souris affiche un signe d'interdiction .
  Pour des informations plus détaillées, consultez la section *Blocs fonctions (FFB), page 283*.
- Liaison
  Pour placer une liaison dans la zone de dessin, cliquez sur la broche de sortie du bloc fonction d'origine et faites glisser le pointeur de la souris vers la broche d'entrée du bloc fonction de destination. Les broches d'entrée et de sortie connectées doivent avoir des types de données correspondants.
  Pour des informations plus détaillées, consultez la section *Liaisons, page 287*.

Remplacer un bloc fonction

Pour remplacer un bloc fonction, sélectionnez **Remplacer bloc fonction** dans le menu contextuel.

Dans la boîte de dialogue **Sélectionner un bloc fonction** qui s'affiche, vous pouvez sélectionner le bloc fonction souhaité.

- Si le bloc fonction à remplacer est déjà lié, les types de données d'entrée et de sortie du nouveau bloc fonction doivent correspondre.
- Si le bloc fonction à remplacer n'est pas encore lié, il peut être remplacé par n'importe quel bloc fonction.

Sélectionner plusieurs objets

Pour sélectionner plusieurs objets, vous pouvez maintenir enfoncée la touche **Ctrl** et effectuer une sélection objet par objet, ou maintenir le bouton de gauche de la souris enfoncé et tracer un rectangle autour des objets à sélectionner.

Déplacer des objets

Pour déplacer un bloc fonction, cliquez avec le bouton de gauche sur le bloc fonction sélectionné, maintenez le bouton de la souris enfoncé, puis déplacez le bloc fonction vers la position souhaitée. Gardez à l'esprit les informations concernant la mise en place des objets, données ci-dessus.

NOTE : Les liaisons ne peuvent être ni déplacées, ni remises en forme. Elles sont automatiquement remises en forme.

Couper, copier et coller

Pour couper ou copier un objet dans le presse-papier, sélectionnez l'objet, puis sélectionnez **Couper** ou **Copier** dans le menu contextuel.

Grâce au menu contextuel, vous pouvez **Coller** l'objet dans un espace libre de la zone de dessin. Gardez à l'esprit les informations concernant la mise en place des objets, données ci-dessus.

Si vous coupez/copiez et collez un bloc fonction unique comportant une liaison associée, cette liaison sera perdue.

NOTE : Les opérations couper, copier et coller peuvent s'appliquer à des liaisons uniquement si on y inclut les blocs fonctions d'origine et de destination (sélection multiple)

Suppression d'objets

Pour supprimer un objet (bloc fonction, liaison), sélectionnez l'objet, puis appuyez sur la touche **Suppr** ou sélectionnez **Supprimer** dans le menu contextuel.

NOTE : Les variables, les valeurs littérales ou les liaisons associées sont supprimées en même temps que l'objet.

La suppression d'une variable de la zone de dessin ne supprime pas cette variable du projet.

Fonctionnalité glisser-déposer

Pour associer des variables à des broches, outre la sélection de données, vous pouvez utiliser la fonctionnalité glisser-déposer.

- Vous pouvez faire glisser des variables d'une broche à une autre dans la zone de dessin.
- Vous pouvez faire glisser des variables depuis l'afficheur de variables vers la zone de dessin (veuillez consulter la section *Afficheur de variables, page 300*).

NOTE :

- Si une broche est déjà associée à une variable ou à une valeur littérale, l'affectation d'une nouvelle variable ou d'une valeur littérale à l'aide de la fonctionnalité glisser-déposer remplace l'association précédente.
- Il est impossible d'associer des variables de **type de connexion** IHM à une broche de bloc fonction.

Fonctionnement du clavier

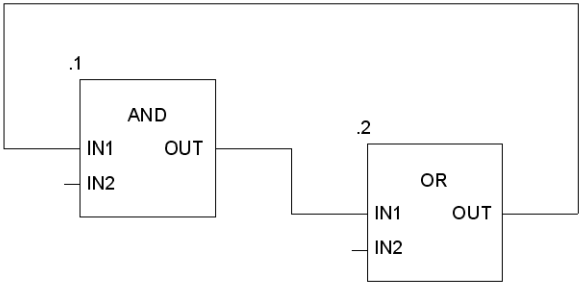
Touche	Description
Flèches (gauche, droite, haut, bas)	Déplace le bloc fonction sélectionné de 5 pixels dans la direction correspondante
Page suiv./page préc.	Fait défiler d'une hauteur de fenêtre dans la direction correspondante
Ctrl + page suiv./page préc.	Fait défiler d'une largeur de fenêtre dans la direction correspondante
Suppr	Supprime l'objet sélectionné
F1	Affiche l'aide contextuelle
Echap	Désélectionne tout objet sélectionné
Ctrl + I	Sélection de données
Ctrl + L	Connexion à une broche de forçage

Analyse logique et syntaxique de la logique de forçage

Analyse de la logique de forçage

Si vous cliquez sur le bouton **Appliquer** de l'éditeur graphique de forçages, la logique dessinée est analysée, en fonction des critères suivants.

- Aucune boucle ne doit être présente dans la logique.
  Exemple de boucle.



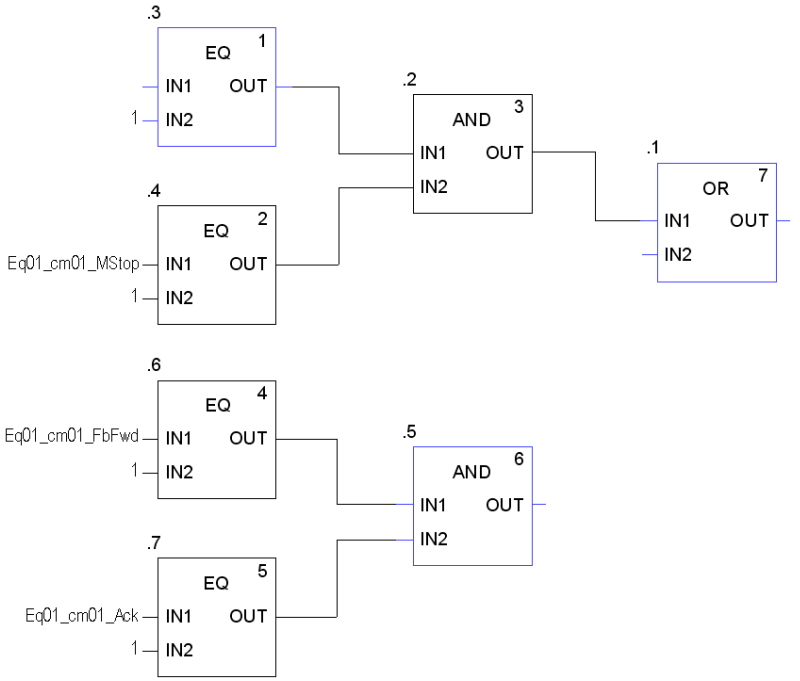
- Au moins deux broches d'entrée des blocs fonctions logiques placés dans la zone de dessin doivent être associées à une variable, à une valeur littérale ou à une liaison.
- Les deux broches d'entrée des blocs fonctions de comparaison placés dans la zone de dessin doivent être associées à une variable, à une valeur littérale ou à une liaison.
- Toutes les broches de sortie de tous les blocs fonctions doivent être connectées à des broches d'entrée d'autres blocs fonctions, à l'exception de la broche de sortie qui sera reliée à la broche de forçage du module de commande.

Onglet Messages

Les erreurs résultant d'un non-respect de ces critères sont répertoriées dans l'onglet **Messages**.

		Description du message	Ligne	Colonne
⊗	1	Broche d'entrée IN2 du FB .1 non connectée	3	8
⊗	2	Broche d'entrée IN1 du FB .3 non connectée	2	3
⊗	3	Broche de sortie Out du FB .1 non connectée	3	8
⊗	4	Broche de sortie Out du FB .5 non connectée	4	5

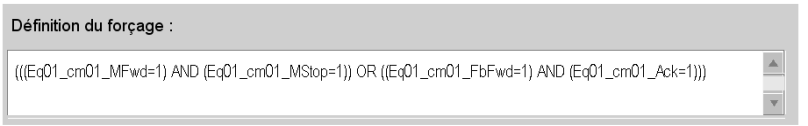
Si les broches d'entrée ou de sortie de blocs fonctions ne sont pas connectées, ces blocs fonctions sont affichés en bleu.



Analyse syntaxique de la logique de forçage

Remédiez aux erreurs détectées au cours de l'analyse de la logique de forçage, puis cliquez à nouveau sur le bouton **Appliquer**.

si aucune erreur n'est détectée, la logique analysée est affichée dans l'onglet **Définition du forçage**.



NOTE : L'onglet **Définition du forçage** est en lecture seule. Vous ne pouvez pas modifier manuellement la logique analysée.

Enregistrement et récupération de la logique de forçage

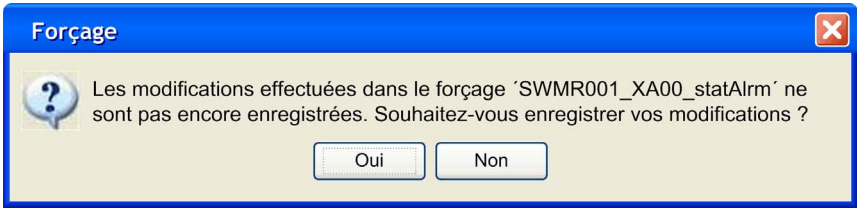
Présentation

L'éditeur graphique de forçages enregistre et récupère la logique de forçage lors de sa fermeture ou de son ouverture.

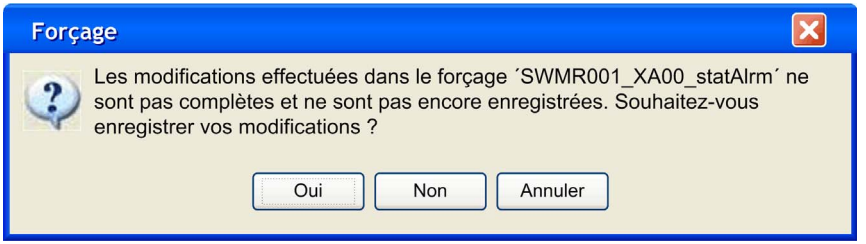
Fermeture sans le bouton Appliquer

Si vous décidez de ne pas cliquer sur le bouton **Appliquer** avant de fermer l'éditeur graphique des forçages :

- Le message suivant s'affiche si le forçage est achevé.



- Le message suivant s'affiche si le forçage est inachevé.



La description de la réponse **Oui** ou **Non** est identique pour les deux boîtes de dialogue.

Bouton	Description
Oui	Si vous cliquez sur **Oui**, le forçage sera enregistré qu'il soit achevé ou non, et la fenêtre des forçages sera fermée.
Non	Si vous cliquez sur **Non**, aucune modification ne sera apportée et la logique précédente sera conservée. La fenêtre des forçages sera fermée.
Annuler	Si vous cliquez sur **Annuler**, la fenêtre des forçages restera ouverte pour vous permettre de corriger la définition incomplète.

Sous-chapitre 7.3
Afficheur de variables

Introduction

Cette section décrit l'afficheur de variables.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Description générale	301
Zone d'informations	303
Filtre	307
Grille des variables	311

Description générale

Présentation

- L'afficheur de variables répertorie l'ensemble des variables disponibles dans un projet UAG. Les variables peuvent être affichées en fonction de différentes sélections de domaine et différents critères de filtres.
- La sélection du domaine permet d'afficher les variables d'un objet spécifique, ou de visualiser toutes les variables du projet.
- Grâce aux critères de filtres, les variables affichées peuvent être limitées, en se basant sur des propriétés telles que le nom ou le type de données.
- Il est possible d'ouvrir simultanément plusieurs fenêtres d'afficheur de variables.
- Vous pouvez faire glisser des variables de l'afficheur de variables à d'autres fenêtres, si nécessaire (par exemple, l'éditeur graphique de forçages).

Accès au moyen d'un élément physique ou d'API

Etape	Action
1	• Dans le **modèle physique**, sélectionnez un élément physique (de **Site** vers **Modules de commande**) ou • dans le **modèle topologique**, sélectionnez le nœud des **API** ou l'un des API configurés situés sous ce nœud ou • dans l'arborescence **Bibliothèques**, sélectionnez le nœud d'un **Type de modèle**.
2	Cliquez sur le bouton droit et sélectionnez **Ouvrir variables** dans le menu contextuel. **Résultat :** L'afficheur de variables s'ouvre. Le **domaine** est l'élément physique sélectionné (par exemple, le site) et le filtre reste vide ou sera défini selon l'API ou les API correspondants.

Accès par double-clic

Dans le **modèle physique**, double-cliquez sur l'un des modules de contrôle pour ouvrir l'afficheur de variables. Le **domaine** est le module de commande sélectionné et toutes les variables disponibles pour ce module de commande sont répertoriées.

Accès par la barre d'outils

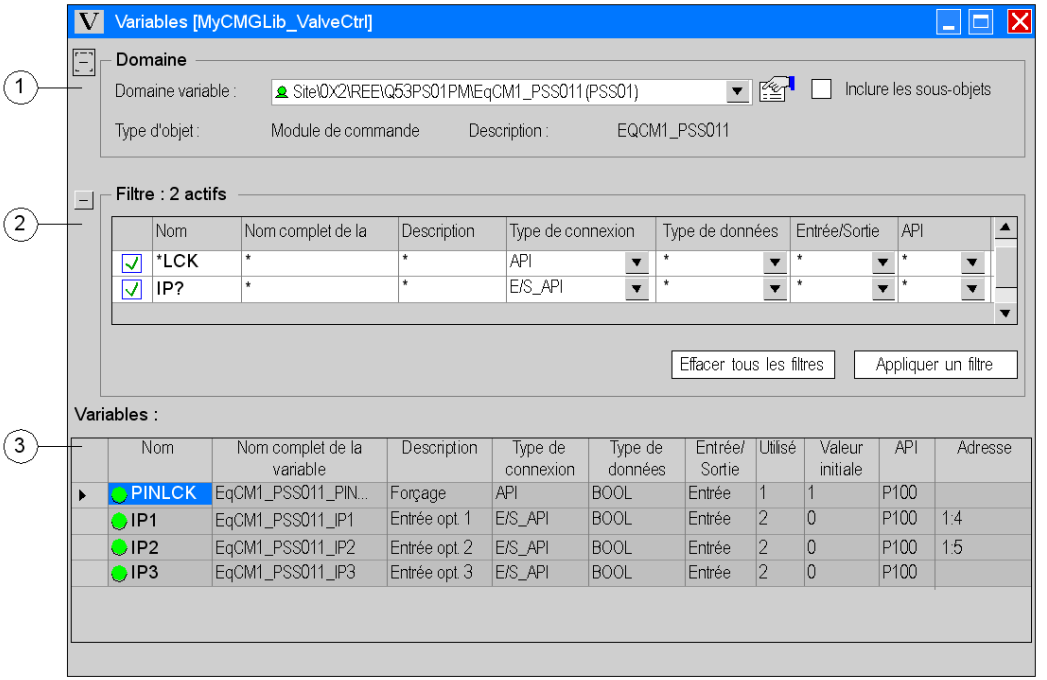
Si vous accédez à l'afficheur de variables en utilisant le bouton de la barre d'outils, le **domaine** sera défini par l'objet sélectionné dans l'arborescence du modèle physique.

Un seul afficheur de variables par domaine

Pour un même domaine, il n'est possible de lancer qu'une seule instance de l'afficheur de variables.

Pour différents domaines, il est possible de lancer plusieurs instances de l'afficheur de variables.

Structure générale



Elément	Nom	Description
1	**Domaine**	Zone d'informations avec : ● Sélection du **domaine de variable** ● Case à cocher **Inclure les sous-objets** ● **Type d'objet** de l'objet dans le domaine de variable ● **Description** de l'objet dans le domaine de variable
2	**Filtre**	Grille de filtre avec champs pour conditions de filtres
3	**Variables**	Grille des variables

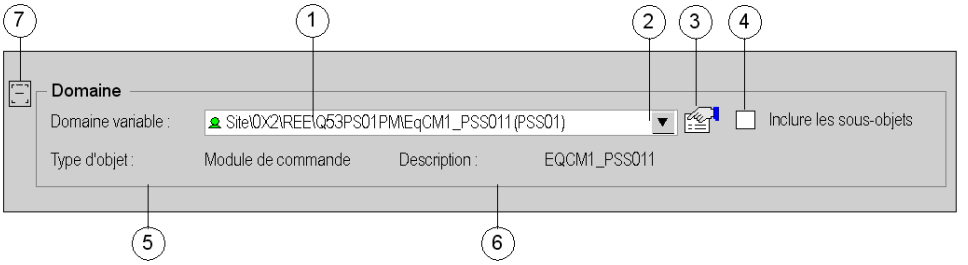
Zone d'informations

Présentation

La zone d'informations contient :

- le domaine de variable actuel
- une zone de liste permettant de modifier le domaine de variable
- un bouton destiné à ouvrir la boîte de dialogue de propriétés de l'élément de domaine de variable
- une case à cocher permettant d'inclure des variables à partir de sous-objets de l'élément de domaine de variable
- le type d'objet de l'élément de domaine de variable
- la description de l'élément de domaine de variable

Représentation



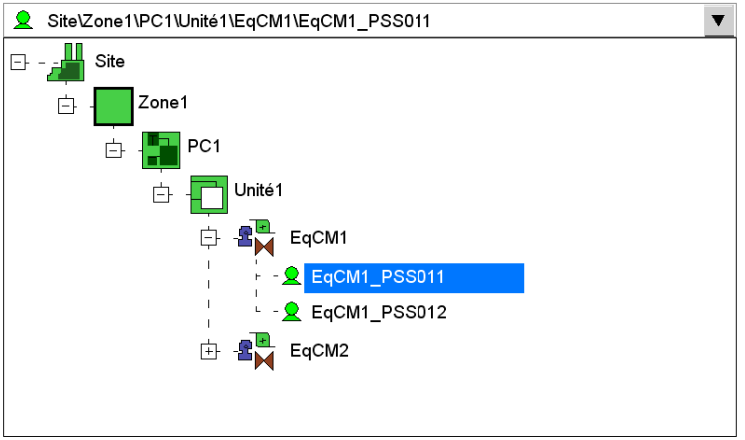
Elément	Nom	Description
1	**Domaine de variable**	Affiche le domaine de variable actuel avec son chemin d'accès complet dans le modèle physique Le domaine de variable apparaît également dans la légende de la zone d'informations, mais sans son chemin d'accès.
2	Zone de liste	Zone de liste permettant de sélectionner un nouveau domaine de variable (Voir le paragraphe ci-après).
3	Bouton **Propriétés**	Ouvre la boîte de dialogue Propriétés de l'élément de domaine de variable actuel, par exemple la boîte de dialogue Propriétés d'un module de commande
4	Case à cocher **Inclure les sous-objets**	Permet de répertorier également l'ensemble des variables des sous-éléments du domaine de variable (Voir le paragraphe ci-après).
5	**Type d'objet**	Type de l'élément sélectionné comme domaine de variable (par exemple, module de commande)

Elément	Nom	Description
6	**Description**	Description de l'élément sélectionné comme domaine de variable
7	Bouton **+ (-)**	Développe ou réduit la zone d'informations ● Si elle est développée, toutes les informations sont affichées dans la zone ● Si elle est réduite, seule la première ligne de la zone d'informations apparaît

Modification du domaine de variable

Cliquez sur la flèche descendante de la zone de liste pour ouvrir une structure d'arborescence et visualiser l'ensemble des éléments du **modèle physique**.

Exemple



Utilisez cette structure d'arborescence pour sélectionner l'un des domaines de variable souhaité.

Vous pouvez également modifier le domaine de variable en faisant glisser dans l'afficheur de variables un élément quelconque du **modèle physique** ou du **modèle topologique**, pour lequel le balayage des variables est autorisé.

Si vous modifiez le domaine de variable,
- l'afficheur de variables procède à un nouveau chargement des données ;
- la coche de la case **Inclure les sous-objets** est effacée.

NOTE : Les conditions de filtre ne sont pas affectées par la modification du domaine de variable.

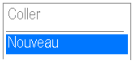
Inclure les sous-objets

La case à cocher **Inclure les sous objets** permet également de répertorier dans la grille de variables celles qui appartiennent aux sous-éléments du domaine de variable.

- Si la case n'est **pas** cochée, seules les variables appartenant à l'élément du champ de domaine de variable apparaissent.
  Par défaut, la case n'est pas cochée lorsqu'une nouvelle instance de l'afficheur de variables s'affiche.
- Si la case **est** cochée, toutes les variables appartenant aux sous-éléments de l'élément de domaine de variable sont également affichées.
  Les filtres définis (le cas échéant) sont également pris en compte lors de l'établissement de la liste des variables.

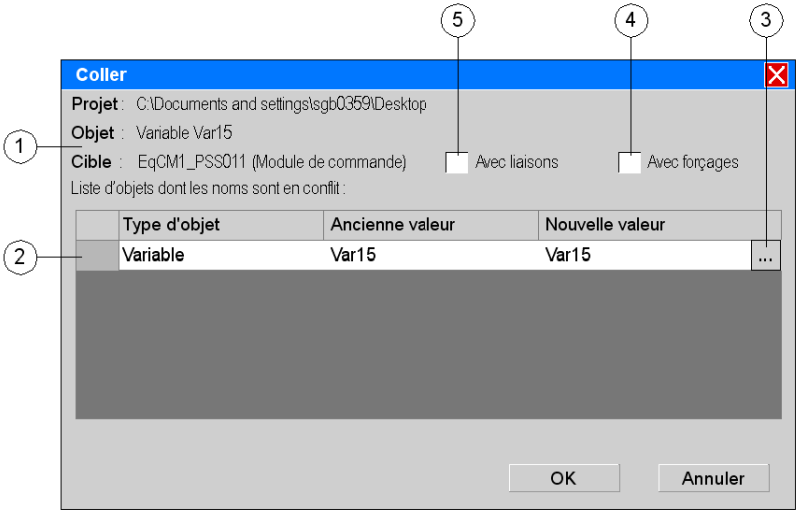
Coller/Nouveau

Cliquez avec le bouton droit de la souris à un endroit quelconque de la zone d'informations pour afficher son menu contextuel.



- Cliquez sur **Nouveau** dans le menu contextuel de la zone d'informations pour ouvrir la boîte de dialogue **Nouvelle variable**, qui permet de créer une nouvelle variable libre.
  La nouvelle variable créée est ajoutée à l'élément sélectionné dans le champ de domaine de variable.
  L'option **Nouveau** du menu contextuel n'est activée que si l'élément du domaine de variable est un module de commande ou un élément physique susceptible de contenir des variables.
- L'option **Coller** du menu contextuel n'est activée que si le presse-papier contient une variable valide (par exemple, copiée à partir d'une instance de l'afficheur de variables) et si l'élément de domaine de variable est un module de commande ou un élément physique susceptible de contenir des variables.
  Cliquez sur **Coller** pour ouvrir la boîte de dialogue **Coller**, qui permet, entre autres, de modifier le nom de la nouvelle variable (voir le paragraphe ci-après).
  La nouvelle variable collée est ajoutée à l'élément sélectionné dans le champ de domaine de variable.

Boîte de dialogue Coller



Elément	Nom	Description
1	Informations	Affiche les informations suivantes • **Projet** Nom du projet UAG actuel, avec son chemin d'accès complet • **Objet** Nom de la variable • **Cible** Elément cible (par exemple, module de commande) de l'opération de collage
2	**Liste d'objets dont les noms sont en conflit**	Indique le **Type d'objet** ainsi que son **Ancienne valeur** et sa **Nouvelle valeur**, avec un bouton **...** pour saisir le nouveau nom de la variable
3	**...Bouton**	Ouvre la boîte de dialogue **Nom** Cette boîte de dialogue vous permet de saisir une **Nouvelle valeur** pour la nouvelle variable.
4	Case à cocher **Avec forçages**	Permet de coller la variable avec ses forçages
5	Case à cocher **Avec liaisons**	Permet de coller la variable avec ses liaisons

Filtre

Présentation

Les variables du **domaine de variable** sont répertoriées dans la grille des **variables** en fonction des conditions de **filtre**.

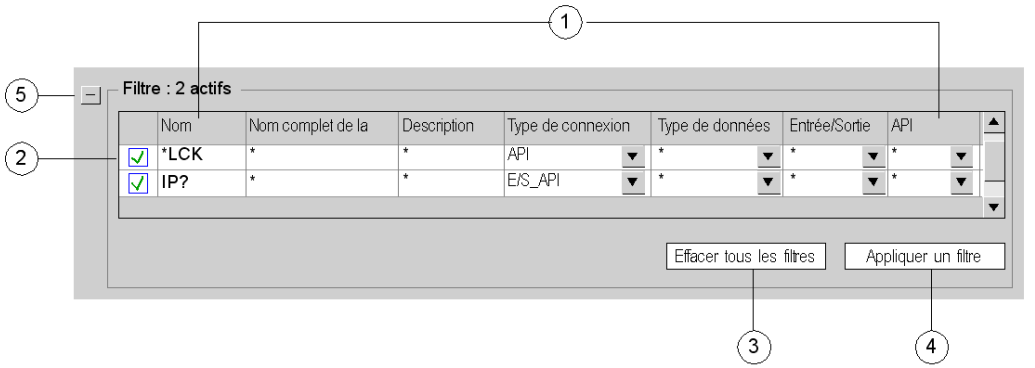
Le **filtre** permet de :

- combiner des conditions de filtre dans une ligne de filtres (par une opération AND) ;
- combiner des lignes de conditions de filtre (par une opération OR) ;
- activer/désactiver des lignes de conditions de filtre.

Les filtres définis ne sont pas enregistrés lors de la fermeture de l'afficheur de variables.

NOTE : Bien qu'elles apparaissent dans les colonnes de la grille des variables, vous ne pouvez pas utiliser les propriétés de variables **Utilisée**, **Valeur initiale** et **Adresse** comme conditions de filtre.

Représentation



Elément	Nom	Description
1	Conditions de filtre	Toutes les conditions de filtre d'une ligne sont combinées par un opérateur AND.
2	Ligne de condition de filtre	Toutes les lignes de condition de filtre activées sont combinées par un opérateur OR. Pour activer/désactiver une ligne, utilisez la case à cocher située devant cette ligne.
3	Bouton **Appliquer filtre**	Affiche la liste de toutes les variables correspondant au filtre défini dans la grille des variables
4	Bouton **Effacer tous les filtres**	Efface toutes les lignes de filtre, sauf une Seule une ligne de filtre comportant des caractères génériques reste affichée.

Elément	Nom	Description
5	Bouton **+ (-)**	Développe ou réduit le filtre ● S'il est développé, toutes les lignes de condition de filtre et les boutons sont affichés. ● S'il est réduit, seuls les numéros des filtres actifs et des filtres appliqués sont affichés.

Conditions de filtre

Toutes les conditions de filtre d'une ligne sont combinées par un opérateur AND.

- **Nom**
  Nom de la variable, texte avec caractères génériques (voir le paragraphe ci-après)
- **Nom complet de la variable**
  Nom complet de la variable, texte avec caractères génériques (voir le paragraphe ci-après)
  Le nom complet de la variable est constitué conformément aux **règles de création de noms** de l'éditeur de **personnalisation UAG**.
- **Description**
  Description de la variable, texte avec caractères génériques (voir le paragraphe ci-après)
- **Type de connexion**
  Une zone de liste permet de sélectionner le type de connexion parmi tous les types de connexion disponibles.
- **Type de données**
  Une zone de liste permet de sélectionner le type de données parmi tous les types de données disponibles dans le projet.
- **Entrée/Sortie**
  Une zone de liste permet de sélectionner le type d'entrée/sortie parmi tous les types d'entrée/sortie disponibles.
- **API**
  Une zone de liste vous permet de sélectionner tout API défini dans le **modèle topologique**.

Ligne de condition de filtre

Toutes les lignes de condition de filtre activées sont combinées par un opérateur OR.

Pour activer/désactiver une ligne, utilisez la case à cocher située devant cette ligne.

Nouveau filtre

A l'ouverture de l'afficheur de variables, une ligne de condition de filtre s'affiche.

Pour ajouter une nouvelle ligne, utilisez l'option **Nouveau filtre** du menu contextuel.

Pour toutes les conditions de filtre d'une nouvelle ligne, des caractères génériques s'affichent et la case à cocher située devant la ligne est activée.

Supprimer

Pour supprimer une ligne de condition de filtre, sélectionnez la ligne correspondante et utilisez l'option **Supprimer** du menu contextuel.

NOTE : Aucune demande de confirmation de la suppression d'une ligne n'est faite.

Appliquer des filtres

Cliquez sur le bouton **Appliquer des filtres** pour afficher la liste de toutes les variables correspondant aux conditions de filtre.

Si vous tentez d'appliquer un filtre qui comporte des lignes de filtre vierges (autrement dit, toutes les conditions de filtre sont vierges ou ne contiennent que des caractères génériques), un message approprié s'affiche. Il vous appartient alors de confirmer ce filtre.

Effacer tous les filtres

Cliquez sur le bouton **Effacer tous les filtres** pour effacer toutes les conditions de filtre et toutes les lignes, sauf une.

Seule une ligne de filtre reste affichée et comporte des caractères génériques correspondant à toutes les conditions de filtre.

Caractères génériques

Vous pouvez utiliser deux types de caractères génériques :

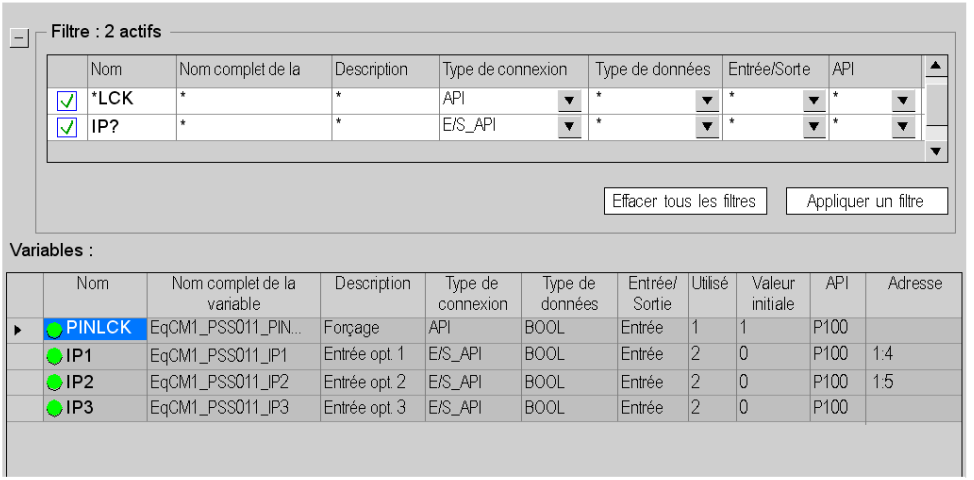
- ***** comme caractère générique pour plusieurs caractères
- **?** comme caractère générique pour un seul caractère

Exemple :

Utilisation de la condition de filtre **Nom** avec caractères génériques

Texte saisi	Résultat
*	Variables portant un nom quelconque
*LCK	Variable portant un nom se terminant par LCK
LCK?	Variables portant un nom contenant LCK dans la position des trois premiers caractères et comportant exactement quatre caractères comportant

Des conditions complexes de filtre peuvent être définies en indiquant des conditions dans plusieurs champs et en ajoutant plusieurs lignes de filtre.



Grille des variables

Présentation

La grille des variables répertorie l'ensemble des variables selon

- le **domaine de variable** actuel ;
- la valeur de la case à cocher **Inclure les sous-objets** (cochée/non cochée) ;
- les conditions de filtre appliquées.

Représentation

Variables :

		Nom	Nom complet de la variable	Description	Connexion Type	Type de données	Entrée/ Sortie	Utilisé	Valeur initiale	API	Adresse
▶		RDY	EqCM1_PSS011_RDY	Prêt	API	BOOL	Entrée	5	1	P100	1:1
		MS	EqCM1_PSS011_MS	Etat moteur	E/S_API ✓	BOOL	Entrée	2	0	P100	1:2
		LR	EqCM1_PSS011_LR	Comm. distant	E/S_API ✓	BOOL	Entrée	2	0	P100	1:3
		IP1	EqCM1_PSS011_IP1	Entrée opt. 1	E/S_API ✓	BOOL	Entrée	2	0	P100	1:4
		IP2	EqCM1_PSS011_IP2	Entrée opt. 2	E/S_API	BOOL	Entrée	2	0	P100	1:5
		IP3	EqCM1_PSS011_IP3	Entrée opt. 3	E/S_API	BOOL	Entrée	2	0	P100	
	+	DDT	EqCM1_PSS011_DDT	Variable structurée	API_IHM	DDT_PSSR	Entrée	2	0	P100	
		PRR	EqCM1_PSS011_PRR	Dem. exécution progr.	API	BOOL	Entrée	1	0	P100	
		PSR	EqCM1_PSS011_PSR	Dem. arrêt progr.	API	BOOL	Entrée	1	0	P100	
		MSTRT	EqCM1_PSS011_MS...	Démarr. manuel	API_IHM	BOOL	Entrée	1	0	P100	
		MODE	EqCM1_PSS011_MO...	Mode manuel/auto	API_IHM	BOOL	Entrée/	1	0	P100	
		IP4	EqCM1_PSS011_IP4	Entrée opt.	API_RES ✓	BOOL	Entrée/	1	0	P100	
		Var1	EqCM1_PSS011_Var1	Variable libre	API	BOOL	Entrée	5	1	P100	
		Var2	EqCM1_PSS011_Var2	Var. libre var2	API	BOOL	Entrée	2	0	P100	

Les différents types de variables sont présentés selon différentes manières.

Type de variable	Présentation	Le nom de la variable et la description de la variable ...
Variables SCoD	Couleur grise en arrière-plan	... ne peuvent pas être modifiés
Variables libres	Couleur blanche en arrière-plan	... peuvent être modifiés directement dans la grille des variables

Il est impossible de modifier le **nom** et la **description** de variable de sous-éléments de variables structurées.

Pour modifier d'autres propriétés de variable, ouvrez la boîte de dialogue **Propriétés de variable** par un double clic sur la variable.

Autres informations concernant les variables

Variable	Présentation
Variables structurées	Bouton **+** situé face à la colonne **Nom** voir variable `DDT` dans le tableau ci-dessus Le bouton **+** permet d'afficher les variables enfants d'une variable structurée.
Variables avec type de connexion **E/S_API**, si elles sont mises en correspondance avec un **point d'E/S**	Marque ✓ dans la colonne **Type de connexion** voir variable `MS` dans le tableau ci-dessus
Variables avec type de connexion **API_RES**, si elles sont mises en correspondance avec un canal **API-PartenaireReseau**	Marque ✓ dans la colonne **Type de connexion** voir variable `IP4` dans le tableau ci-dessus
Variables avec propriété **Inversée** fixée à Vrai	Symbole I dans la colonne **Type de connexion** voir variable `IP3` dans le tableau ci-dessus
Variables avec une propriété **Variable utilisée** positionnée à Faux	Texte présent et barré dans **Type de connexion**

Ordre d'affichage par défaut

L'ordre d'affichage par défaut des variables est le suivant :

- Si la case **Inclure les sous-objets** est cochée, les variables d'un objet sont affichées avant les variables des sous-objets.
  Cette règle se répète pour chaque sous-objet, ce qui signifie que les variables des sous-objets sont affichées avant les variables d'autres sous-objets de même niveau.
- Les variables SCoD sont affichées avant les variables libres (en séquence, selon la configuration définie dans l'éditeur SCoD).

NOTE : Le tri des variables par intitulés de colonnes ne tient pas compte de l'ordre d'affichage hiérarchique des variables.

Tri des variables

Pour trier les variables, cliquez sur un intitulé de colonne de la grille des variables (**Nom**, **Nom complet de variable** etc.).

L'ordre d'affichage des variables enfants d'une variable, qui utilise un type de données structurées, n'est pas affecté par le tri.

Valeur Utilisée des variables structurées

La valeur affichée dans la colonne **Utilisée** des variables structurées est calculée de la manière suivante :

- Pour les variables parents
  Seules sont prises en compte les utilisations de l'intégralité de la variable (toutes les variables enfants doivent être utilisées). Ne sont pas prises en compte les utilisations dans lesquelles seules quelques variables enfants sont exploitées.
- Pour les variables enfants
  Toutes les utilisations des variables enfants sont prises en compte.

Pour plus d'informations, reportez-vous à la section *Onglet Utilisations des propriétés de variables, page 114*.

Fonctionnalité glisser-déposer

Vous pouvez faire glisser des variables de la grille des variables à une boîte de dialogue d'utilisation des variables :

- Tableau **Liaisons**
- Editeur graphique de forçages
- Tableau **Points d'E/S**.
- Tableau **Canaux**

Propriétés

Pour ouvrir la boîte de dialogue **Propriétés** d'une variable

- Double-cliquez sur la variable souhaitée dans la grille des variables
- Sélectionnez la variable souhaitée dans la grille des variables et sélectionnez **Propriétés** dans le menu contextuel

Menu contextuel de la grille des variables

Cliquez avec le bouton droit de la souris sur une ligne de la grille des variables pour ouvrir un menu contextuel comportant les options suivantes :

- **Copier**
  Copie la variable sélectionnée dans le presse-papier
- **Coller**
  L'option **Coller** du menu contextuel n'est activée que si le presse-papier contient une variable valide (par exemple, copiée à partir d'une instance de l'afficheur de variables) et si l'élément de domaine de variable est un module de commande ou un élément physique susceptible de contenir des variables.
  Cliquez sur **Coller**pour ouvrir la boîte de dialogue Coller qui permet, entre autres, de modifier le nom de la nouvelle variable (reportez-vous à la Boîte de dialogue Coller *(voir page 306)*).
  La nouvelle variable collée est ajoutée à l'élément sélectionné dans le champ de domaine de variable.

- **Nouveau**
  L'option **Nouveau** du menu contextuel n'est activée que si l'élément du domaine de variable est un module de commande ou un élément physique susceptible de contenir des variables.
  Si elle est activée, l'option **Nouveau** ouvre la boîte de dialogue **Nouvelle variable**.
- **Supprimer**
  Supprime la variable sélectionnée
  L'option **Supprimer** n'est disponible que pour les variables libres.
- **Propriétés**
  Ouvre la boîte de dialogue **Propriétés** de la variable sélectionnée

Personnaliser les colonnes

Cliquez avec le bouton droit de la souris sur l'intitulé d'une colonne de la grille des variables pour ouvrir la boîte de dialogue **Configuration des colonnes**.

Elément	Description
Colonnes	Activez/désactivez la case à cocher située devant une colonne pour afficher/masquer la colonne correspondante de la grille des variables. La colonne **Nom** de la grille des variables est obligatoire et ne peut pas être désactivée.

Elément	Description
Bouton **Afficher**	Sélectionnez une colonne et cliquez sur le bouton **Afficher** pour afficher la colonne correspondante de la grille des variables.
Bouton **Masquer**	Sélectionnez une colonne et cliquez sur le bouton **Masquer** pour masquer la colonne correspondante de la grille des variables.
Bouton **Monter**	Pour modifier l'ordre des colonnes, sélectionnez une colonne et cliquez sur le bouton **Monter**. Il est impossible de modifier la position de la colonne **Nom** (première colonne).
Bouton **Descendre**	Pour modifier l'ordre des colonnes, sélectionnez une colonne et cliquez sur le bouton **Descendre**. Il est impossible de modifier la position de la colonne **Nom** (première colonne).
Bouton **Par défaut**	Active les cases à cocher des colonnes par défaut et rétablit l'ordre par défaut
Bouton **Annuler**	Ferme la boîte de dialogue sans modification
Bouton **OK**	Confirme les modifications de disposition de la grille des variables

Chapitre 8
Schéma d’élaboration d’une application

Introduction

Le schéma opératoire décrit ci-dessous est valable pour tous les projets développés sur un même site. Unity Application Generator gère également le développement multi-site. Dans ce cas, le projet général est scindé en projets partiels, lesquels peuvent être développés de la même manière que celle décrite dans ce chapitre.

Pour de plus amples informations concernant le développement de projet multi-site, voir : *Gestion du développement de projets distribués – Fusion de projets, page 385*

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Schéma global d’élaboration d’une application	318
Définition de la personnalisation	320
Définition du modèle physique	322
Définition du modèle topologique	323
Définition du modèle physique	326
Génération d’applications API et IHM	327
Génération d’un fichier d’importation pour les partenaires réseau	329
Assistants de maintenance	330
Documentation de l’application ou d’objets individuels (Rapport)	347

Schéma global d'élaboration d'une application

Introduction

L'établissement d'une application avec Unity Application Generator se décompose en plusieurs phases :

- Phase de préparation
- Développement de l'application
- Test et documentation

Phase de préparation

La première phase sert à adapter Unity Application Generator aux exigences de la société de l'utilisateur et au procédé particulier. Il offre une large gamme d'options de personnalisation.

Phase	Description
1	Personnalisation d'Unity Application Generator de manière à l'adapter aux directives d'un projet particulier, comme par exemple : ● Convention de nommage ● Liste des outils se rapportant au projet ● Types d'API et de modules API autorisés ● ...

Développement de l'application

Le développement d'application est une procédure en plusieurs étapes :

Phase	Description
1	Définition du modèle physique : ● Eléments physiques (ex. : zones, cellules procédé, unités, équipements) ● Modules de commande ● Saisie des paramètres des modules
2	Définition de votre modèle topologique : ● Réseau ● IHM ● Serveur de données ● API
3	Renseignement du modèle physique avec des informations de commande, comme par exemple : ● quel élément physique est commandé par quel API, ● quel élément physique relève de quelle zone de conduite, ● ...
4	Analyse de l'application : ● Informations manquantes ● Cohérence des informations
5	Génération des applications API et IHM

Phase	Description
6	Renseignement des applications API et IHM à l'aide des outils standard de programmation d'API et d'IHM tels que Unity Pro, Vijeo Citect, Monitor Pro, iFIX, ArchestrA.
7	Génération de fichiers d'import pour les partenaires réseau

Test et documentation

La dernière phase couvre le test et le débogage des défauts ainsi que la documentation du projet :

Phase	Description
1	Test de votre application.
2	Réalisation de toutes les modifications nécessaires à la conception de votre procédé dans Unity Application Generator et le cas échéant, faire une génération incrémentale.
3	Documentation de votre application à l'aide du générateur de rapports.

Définition de la personnalisation

Présentation

Ce chapitre vous présente les premières étapes d'un projet.

Etapes avec l'éditeur de personnalisation

Les étapes de la personnalisation (pour plus d'informations sur le fonctionnement de la personnalisation de Unity Application, voir aussi Manuel avancé UAG, Personnalisation et maintenance de projet *(voir Unity Application Generator (UAG) 3.3 SP4, a SoCollaborative software, Manuel utilisateur avancé)*) :

Etape	Action
1	Définir les bibliothèques à utiliser dans votre projet.
2	Définir les outils servant à documenter votre procédé, par exemple DAO, fichiers Word, etc.
3	Définir le modèle physique.
4	Définir la longueur de vos commentaires.
5	Définir les conventions de nommage des différents éléments de Unity Application Generator.
6	Définir les règles de création des noms.
7	Définir les droits d'accès.
8	Définir les groupes d'alarmes.
9	Définir les priorités d'alarme.
10	Définir les noms d'archive.
11	Définir les trames de communication.

Etapes au sein de Unity Application Generator

Dans Unity Application Generator, procédez ainsi :

Etape	Action
1	Ouvrir un nouveau projet et sélectionner un fichier de personnalisation (**Fichier** →**Nouveau**).
2	Enregistrer le nouveau projet dans le répertoire souhaité (**Fichier** →**Enregistrer sous ...**). **NOTE :** Si vous utilisez **Fichier** →**Enregistrer sous...** pour un projet existant, il est possible que vous deviez adapter les paramètres de votre chemin dans le projet, par exemple, pour les applications générées ou votre configuration Profibus ou Advantys.
3	Définir le chemin de l'application API (**Vue** →**Options** →**API** →**Chemin d'accès au projet API**). **NOTE :** ● Le chemin d'accès au projet API est le chemin utilisé pour les projets Unity Pro qui seront chacun générés dans leur propre sous-répertoire. ● Vous pouvez saisir le **chemin d'accès au projet API** en utilisant une variable d'environnement qui correspond à l'un des chemins définis dans l'éditeur de personnalisation, par exemple, `%PROJECTDIR%\PLC`. Pour plus d'informations, veuillez consulter le *Manuel utilisateur avancé UAG, Options de personnalisation*.

Définition du modèle physique

Présentation

Ce chapitre vous donne un aperçu de ce qu'il faut faire pour définir le modèle physique.

NOTE : Pour travailler efficacement, servez-vous de la possibilité qui vous est offerte d'importer des données depuis d'autres outils, tels qu'AutoCAD. Unity Application Generator sait importer tout ou partie du modèle physique sous forme de fichier CSV ou de fichier XML.

Voir également :

- *Modèle physique, page 59*
- *Interfaces avec d'autres outils (fonctions Importer et Exporter), page 397*

Etapes de la définition du modèle physique

Suivez les étapes ci-dessous pour définir le modèle physique :

Etape	Action
1	Saisissez les instruments (futurs modules de commande) dans la liste d'instruments et définissez les propriétés. **Remarque** : il est possible d'importer des instruments sous forme de fichier CSV ou XML depuis le schéma de tuyauterie et d'instrumentation (PID) vers la liste d'instruments.
2	Définissez les éléments physiques (ex. : zones, cellules procédé, unités, équipements), décrivez et commentez chacun d'eux, puis affectez les documents significatifs pour ces éléments physiques. Les définitions relatives aux API et IHM sont données une fois le modèle physique construit.
3	Créez des modules de commande dans les éléments physiques ou glissez des instruments depuis la liste d'instruments vers les éléments physiques (ce faisant, les instruments deviennent des modules de commande et disparaissent de la liste d'instruments). Renseignez les propriétés du module. **Remarque** : il est possible de configurer des modules de commande sans spécifier de type (Module de commande sans type) si vous n'êtes pas certain de la mise en œuvre nécessaire pour le module de commande.
4	Ajoutez des lignes de séparation afin de regrouper les modules de commande, si nécessaire (voir *Lignes de séparation, page 68*).

Définition du modèle topologique

Présentation

Ce chapitre vous donne un aperçu de ce qu'il faut faire pour définir le modèle topologique. Il faut tout d'abord définir le réseau. Les informations issues de cette définition sont nécessaires pour d'autres éléments du modèle topologique.

Reportez-vous à : *Modèle topologique, page 117*

Etapes de définition du réseau

Suivez les étapes ci-dessous pour définir le réseau :

Etape	Action
1	Pour Modbus Plus, définissez le segment de réseau (nom et type de réseau). Pour Ethernet, définissez le masque de sous-réseau et la passerelle par défaut.
2	Pour Modbus Plus, définissez les chemins de routage (émetteur/récepteur, chemin).

Etapes de définition des serveurs de données

Suivez les étapes ci-dessous pour définir les serveurs de données :

Etape	Action
1	Définissez les serveurs de données, commentez-les tous et affectez les documents significatifs pour ces serveurs de données.
2	Définissez le timeout du serveur de données.
3	Définissez la connexion au réseau (segment, adresse).

Etapes de définition des IHM

Suivez les étapes ci-dessous pour définir les IHM :

Etape	Action
1	Définissez les IHM, commentez-les toutes et affectez les documents significatifs pour ces IHM.
2	Définissez les zones de conduite de chaque IHM et commentez-les.
3	Affectez les éléments physiques fonctionnels aux zones de conduite.

Etapes de définition des API

Suivez les étapes ci-dessous pour définir les API :

Etape	Action
1	Définissez les API, commentez-les tous et affectez les documents significatifs pour ces API. Définissez ensuite le nom du projet API (un projet pour chaque API). De plus pour Quantum, définissez la dimension du châssis local. Les API Quantum, Premium, M340 et M580 sont disponibles dans Unity Pro.
2	Affectez l'élément physique fonctionnel aux API. Dans les propriétés de l'API, des statistiques d'E/S indiquent le nombre de points d'E/S nécessaires.
3	Définissez les autres châssis pour chaque API, leurs type de liaison, numéro de station et taille de châssis.
4	Définissez les modules matériels au sein de chaque châssis.
5	Définissez les plages d'adresses pour chaque API. **Remarque :** vérifiez les adresses en fin de définition du procédé afin d'optimiser l'utilisation de la mémoire de l'API.
6	Pour les modules de communication et les UC, définissez leur connexion au segment de réseau et à l'adresse sur le réseau.

Etapes de définition des abonnés réseau

Suivez les étapes ci-dessous pour définir les abonnés réseau :

Etape	Action
1	Définissez les abonnés réseau, commentez-les tous et affectez les documents significatifs pour l'abonné réseau.
2	Définissez la connexion au réseau (segment, adresses).

Etapes de définition des canaux de communication

Suivez les étapes ci-dessous pour définir les canaux de communication de chaque API :

Etape	Action
1	Sélectionnez un API.
2	Définissez les canaux de communication et commentez-les. Chaque partenaire réseau de l'API nécessite un canal de communication. ● Nom de canal ● Type de communication (API, Serveur de données, Partenaire réseau) et partenaire réseau ● Chemin de communication, timeout et taux de répétition du scrutateur d'E/S (uniquement pour les canaux Ethernet API-API) ● Plages d'adresses pour les registres 0x et 4x dans le cas des partenaires réseau **Remarque :** pour les serveurs de données, les statistiques (nombre de variables disponibles, nombre de variables utilisées, pourcentage de variables utilisées) des variables IHM peuvent être affichées à l'aide de l'option du menu **Ouvrir la communication IHM**. Si une trame de communication est utilisée à plus de 100 %, cette trame est recalculée au prochain passage de cartographie mémoire. Pour éviter ce traitement, il convient d'augmenter le nombre de variables IHM dans les propriétés de l'API.
3	Répétez l'étape 1 jusqu'à ce que vous ayez défini les canaux de communication ainsi que les partenaires de communication pour chaque API.

Etapes de génération des API

Suivez les étapes ci-dessous pour générer les API :

Etape	Action dans Unity Application Generator
1	Ouvrez le projet en accès exclusif.
2	Générez des API.
3	Sélectionnez un API dans le modèle topologique et cliquez sur l'icône Unity Pro dans la barre d'outils pour ouvrir le projet dans Unity Pro.
4	Après chaque **génération d'API**, établissez la connexion entre le module de communication et le réseau dans Unity Pro.

Etapes de finalisation du projet

Suivez les étapes ci-dessous pour finaliser le projet :

Etape	Actions dans Unity Pro
1	Créez la logique utilisateur du projet.
2	Analysez le projet à l'aide de **Construire** →**Analyser projet** (Unity Pro uniquement).
3	Corrigez le projet si nécessaire (Unity Pro uniquement). Si nécessaire, utilisez la fonction de correction.
4	Téléchargez le projet sur l'API.

Définition du modèle physique

Présentation

Cette partie vous explique comment faire pour parachever le modèle physique à la suite du modèle topologique.

Les étapes du renseignement du modèle physique

Les étapes du renseignement du modèle physique

Etape	Action
1	Affectez un API à chaque élément physique fonctionnel et définissez un nom de section Unity Pro pour le code généré de l'élément physique fonctionnel.
2	Affectez une IHM/zone de conduite à chaque élément physique fonctionnel.
3	Définissez des variables libres pour tous les éléments physiques fonctionnels (si besoin est).
4	Affectez un type aux modules de commande sans type ; définissez les propriétés correspondant au type.
5	Définissez des variables libres pour les modules de commande (si besoin est).
6	Afin d'obtenir une vue d'ensemble des points d'E/S disponibles et obligatoires, consultez les statistiques d'E/S dans les propriétés API. Ajoutez d'autres modules E/S si nécessaire.
7	Affectez des variables du type E/S_API aux modules d'E/S (à glisser depuis l'afficheur des variables et à déposer sur le tableau des points d'E/S d'un module d'E/S).
8	Définissez les variables à échanger, par exemple, entre plusieurs API (à glisser depuis l'afficheur de variables vers la table de communication API). **Remarque** : Unity Application Generator assure automatiquement la communication avec l'IHM par l'intermédiaire des serveurs de données.
9	Définissez les variables de type **API_RES** à échanger entre les API et les partenaires réseau (à glisser depuis le tableau des variables vers la table de communication des partenaires réseau). **Remarque** : chaque variable de type **API_RES** ne peut être utilisée qu'une seule fois pour cette action, sinon l'adresse est ambiguë.
10	En option : pour obtenir une vue d'ensemble des adresses calculées, vous pouvez lancer séparément la cartographie mémoire.

Génération d'applications API et IHM

Présentation

Ce chapitre vous donne un aperçu de ce qu'il faut faire pour générer et tester les applications API et IHM.

Une application ne sera jamais entièrement correcte ni terminée après la première génération de code. Il faudra donc procéder à des modifications de conception. Vous trouverez dans la suite la description des procédures à suivre pour la première génération ainsi que pour les générations de mises à jour.

Reportez-vous à :

- Manuel avancé UAG, Présentation du code généré et des principes de génération *(voir Unity Application Generator (UAG) 3.3 SP4, a SoCollaborative software, Manuel utilisateur avancé)*

Les étapes de génération de nouvelles applications API et IHM

Voici les étapes à suivre pour générer les applications API et IHM :

Etape	Action
1	Lancer l'analyseur. Celui-ci vérifiera toute l'application. Le résultat est ensuite affiché dans la fenêtre des messages.
2	Lire le rapport d'analyse et remédier à tous les problèmes identifiés. Chaque ligne du rapport est un lien hypertexte vers la source du problème, laquelle s'ouvre par un double clic sur la ligne.
3	Répéter les étapes 1 et 2 jusqu'à ce que tous les problèmes identifiés soient solutionnés.
4	Vérifier les plages de mémoire utilisées dans les propriétés de l'API et définir les emplacements de réserve.
5	Ouvrir le projet à l'aide de `Ouvrir sans partage` **Remarque** : ouvrir sans partage interdit à d'autres utilisateurs de travailler sur le même projet. Cela s'avère nécessaire si vous devez effectuer une génération de code.
6	Générer l'application Unity Pro.
7	Préparer et générer l'application IHM. Reportez-vous à : Manuel avancé UAG, IHM et leur configuration *(voir Unity Application Generator (UAG) 3.3 SP4, a SoCollaborative software, Manuel utilisateur avancé)*.
8	Renseigner les applications API et IHM à l'aide des outils standard adéquats tels que Unity Pro et Vijeo Citect, Monitor Pro, iFIX ou ArchestrA.

Les étapes de mise à jour de l'application

La mise à jour de l'application passe par les étapes suivantes :

Etape	Action
1	Identifier les changements devant être apportés à la conception du projet dans Unity Application Generator, par exemple : ● Définition des nouvelles variables ● Ajout de modules à l'automate ● Equipements ou modules de commande supplémentaires ● ...
2	Effectuer tous les changements à la conception dans Unity Application Generator. **Règle** : tout ce qui peut être fait dans Unity Application Generator doit être fait par Unity Application Generator.
3	Lancer l'analyseur. Celui-ci vérifiera toute l'application. **Résultat** : le rapport d'analyse s'affiche ensuite dans une fenêtre de messages.
4	Lire le rapport d'analyse et remédier à tous les problèmes identifiés. **Remarque** : chaque ligne du rapport est un lien hypertexte vers la source du problème, laquelle s'ouvre par un double clic sur la ligne.
5	Répéter les étapes 3 et 4 jusqu'à ce que tous les problèmes identifiés soient solutionnés.
6	Ouvrir le projet à l'aide de `Ouvrir sans partage` **Remarque** : ouvrir sans partage interdit à d'autres utilisateurs de travailler sur le même projet. Cela s'avère nécessaire si vous devez effectuer une génération de code.
7	Générer l'application Unity Pro. **Remarque** : par défaut la génération est effectuée de manière incrémentale. Dans la boîte de dialogue **Options** vous pouvez basculer entre génération incrémentale et nouvelle génération.
8	Préparer et générer l'application IHM (ce qui est effectué de manière incrémentale par défaut). Reportez-vous à : Manuel avancé UAG, IHM et leur configuration *(voir Unity Application Generator (UAG) 3.3 SP4, a SoCollaborative software, Manuel utilisateur avancé)*.
9	Renseigner les applications API et IHM à l'aide des outils standard adéquats tels que Unity Pro et Monitor Pro ou iFIX.

Génération d’un fichier d’importation pour les partenaires réseau

Présentation

Pour la communication des partenaires réseau avec l’application d’automatisme, Unity Application Generator génère un fichier CSV, qui peut être importé dans les partenaires réseau.

Voir aussi Manuel avancé UAG, Génération pour les partenaires réseau *(voir Unity Application Generator (UAG) 3.3 SP4, a SoCollaborative software, Manuel utilisateur avancé)*.

Etape à suivre pour générer un fichier d’importation pour les partenaires réseau

Etape à suivre pour générer un fichier CSV à importer dans les partenaires réseau

Etape	Action
1	Pour chaque partenaire réseau, générer un fichier CSV (**Partenaire réseau** → **Propriétés** →**Exporter variables)**

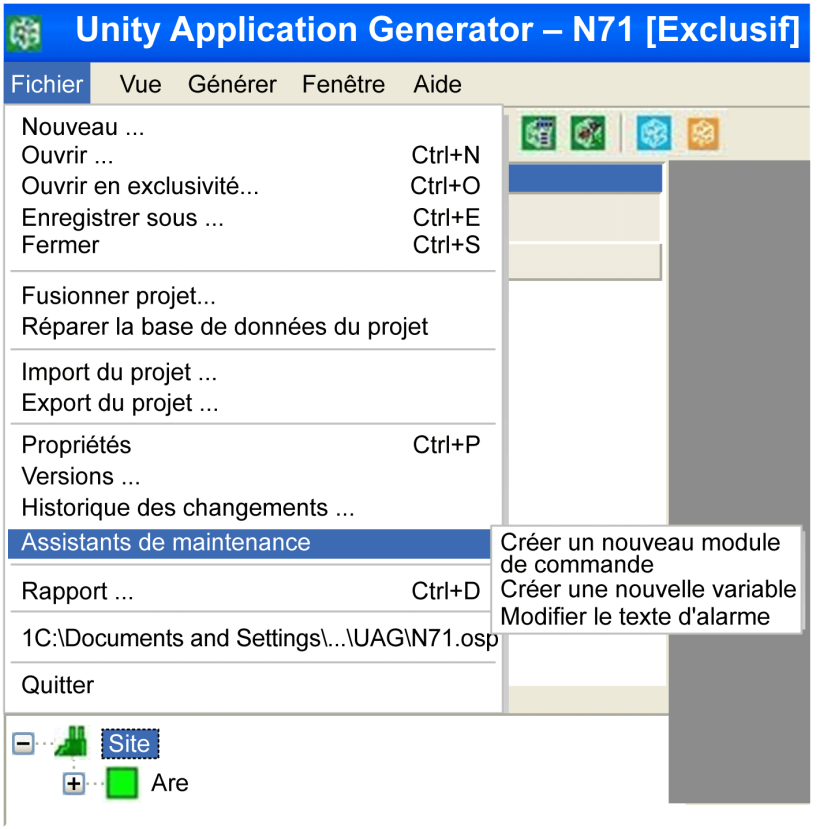
Assistants de maintenance

Introduction

La fonction Assistant de maintenance aide les techniciens chargés de l'entretien de l'application en cours d'exécution. Cette fonctionnalité est active uniquement lorsqu'un projet UAG est ouvert en mode exclusif.

Cette fonctionnalité comprend les options suivantes :

- **Créer un nouveau module de commande**
- **Créer une nouvelle variable**
- **Modifier le texte d'alarme**



Les règles suivantes sont applicables aux **Assistants de maintenance** :

- Seule la génération incrémentale API et IHM est autorisée pour les assistants. La génération complète n'est pas possible.
- Lorsque l'utilisateur travaille sur ces assistants, UAG est désactivé en arrière-plan.
- Il ne permet ni la modification ni la suppression des modules de commande et des variables des assistants.
- **Assistants de maintenance** est désactivé uniquement en mode lecture seule d'UAG.

L'**historique des changements** est mis à jour respectivement pour tous les nouveaux modules de commande, les nouvelles variables et le texte d'alarme modifié avec ses valeurs.

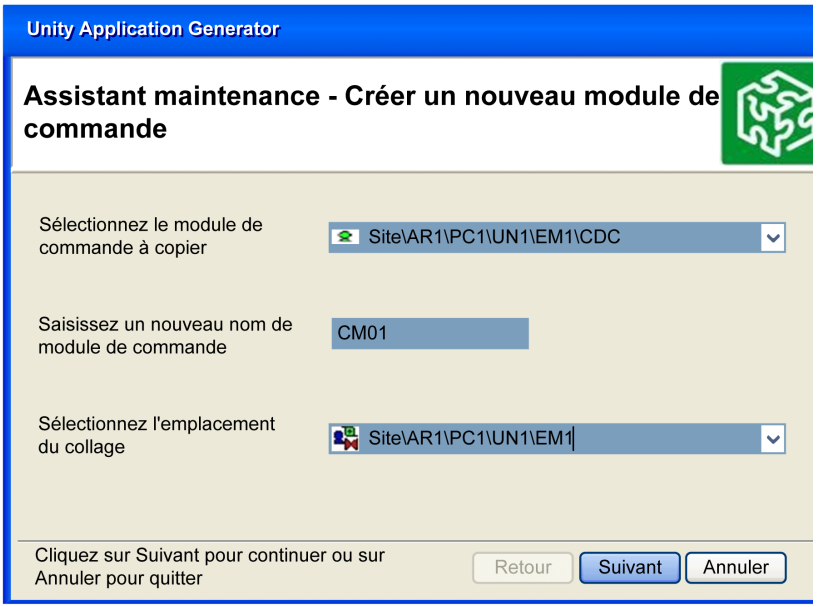
Création d'un nouveau module de commande

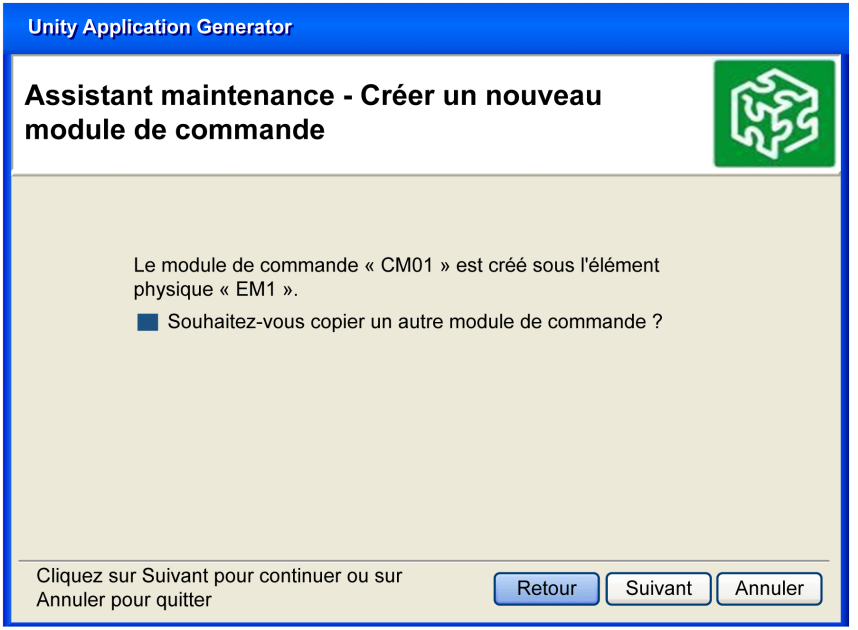
Vous pouvez créer un nouveau module de commande en cliquant sur **Fichier** →**Assistants de maintenance** →**Créer un nouveau module de commande**. Les règles suivantes sont applicables à l'option Créer un nouveau module de commande dans **Assistants de maintenance** :

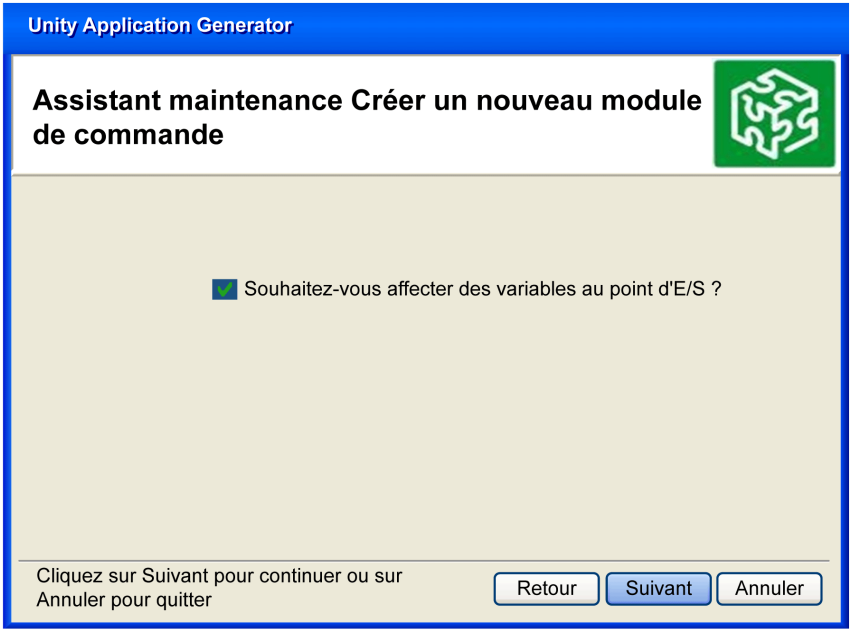
- Le nouveau nom du module de commande est validé selon les règles de nommage et la convention de nommage définies lors de la personnalisation du projet.
- Toutes les propriétés configurées du module de commande source sont affectées au nouveau module lors de l'opération copier-coller du module de commande.
- Vous pouvez créer un nombre N de modules de commande à partir de l'assistant.
- Seules les variables de type E/S_API sont autorisées pour l'affectation d'E/S depuis l'assistant.
- Si les variables n'appartiennent pas au même API lors de l'attribution de la variable à un point d'E/S, l'opération glisser-déposer de la variable n'est pas autorisée.

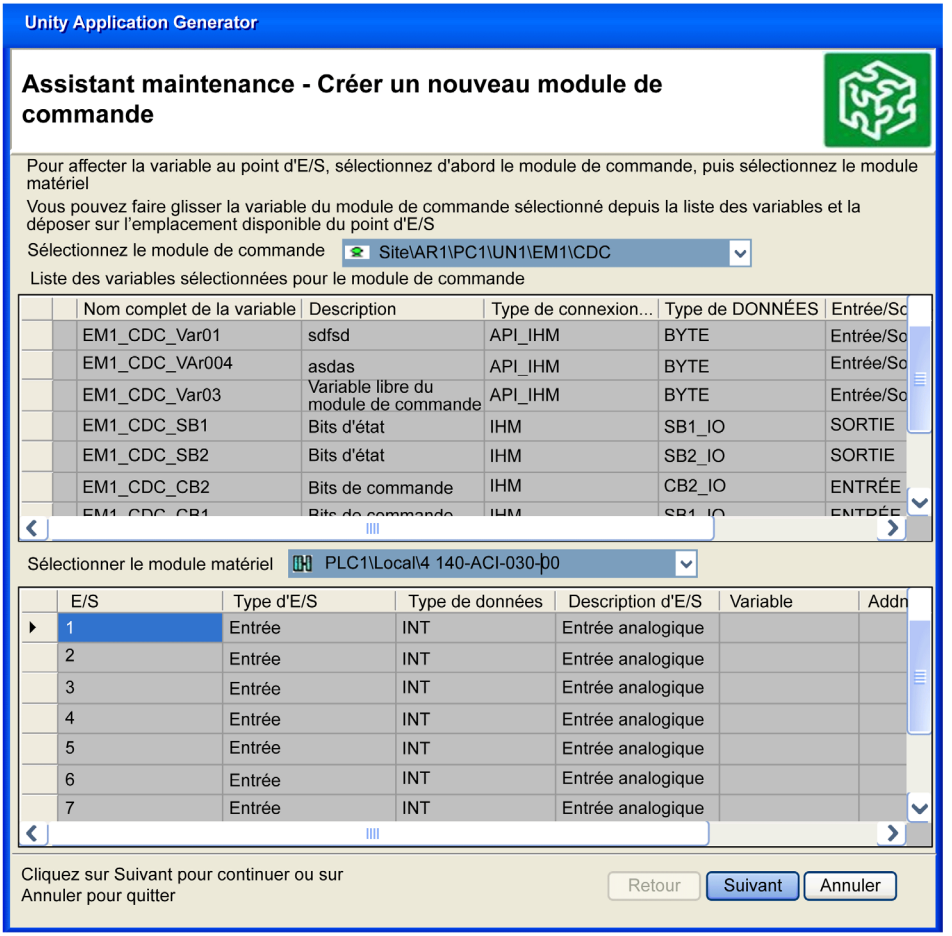
NOTE : Si vous cliquez sur **Annuler** après la création du module de commande, les données créées sont enregistrées.

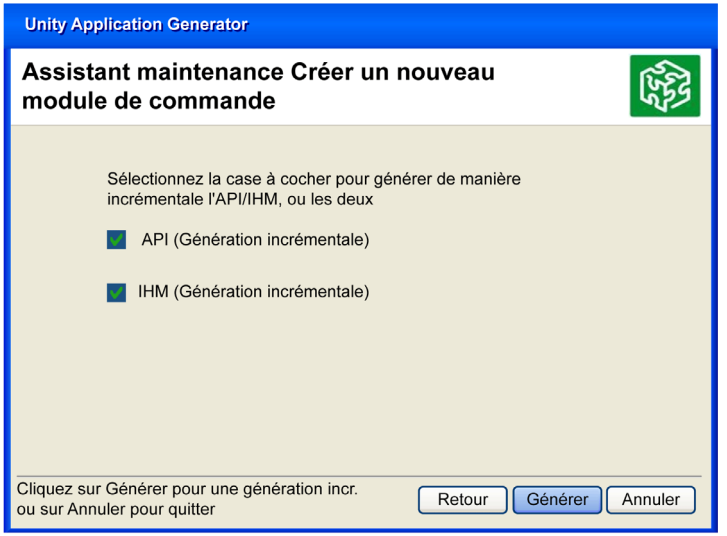
Si vous cliquez sur **Annuler** avant la création du module de commande, l'assistant se ferme.

Etape	Action
1	Cliquez sur **Fichier →Assistants de maintenance →Créer un nouveau module de commande**. **Résultat :** La boîte de dialogue **Assistant de maintenance - Créer un nouveau module de commande** s'affiche. 
2	Sélectionnez un module de commande à copier dans la liste de modèle physique.
3	Saisissez le nom du nouveau module de commande dans le champ **Saisir un nouveau nom de module de commande**.
4	Sélectionnez la destination vers laquelle coller dans le champ **Sélectionner la destination de collage**.

Etape	Action
5	Cliquez sur **Suivant** pour continuer. **Résultat :L'** Assistant de maintenance - Module de commande est créé avec le nouveau nom donné et affiche le nom de la boîte de dialogue **Assistant de maintenance - Créer un nouveau module de commande**. 
6	Sélectionnez l'option **Souhaitez-vous copier un autre module de commande** pour créer un autre module de commande. **NOTE :** Si vous sélectionnez cette option, la boîte de dialogue **Créer un nouveau module de commande** s'affiche.
7	Cliquez sur **Suivant** pour continuer.

Etape	Action
8	Pour affecter une variable à un point d'E/S, sélectionnez l'option **Souhaitez-vous affecter des variables à un point d'E/S ?**. **NOTE :** Si l'option n'est pas sélectionnée, cela provoque la génération d'un API et d'une IHM. 
9	Cliquez sur **Suivant** pour continuer.

Etape	Action
10	Sélectionnez un module matériel dans la liste **Sélectionner un module matériel**. 
11	Sélectionnez une variable de module de commande dans la liste des variables du module de commande.
12	Glissez-déposez la variable sélectionnée dans la liste sélectionnée dans le module matériel.

Etape	Action
13	Cliquez sur **Suivant** pour continuer. **Résultat :** La boîte de dialogue**Assistants de maintenance - Créer un nouveau module de commande** s'actualise. 
14	Sélectionnez **API**, **IHM** ou les deux options.
15	Cliquez sur **Générer** pour la génération incrémentale.

Création d'une nouvelle variable

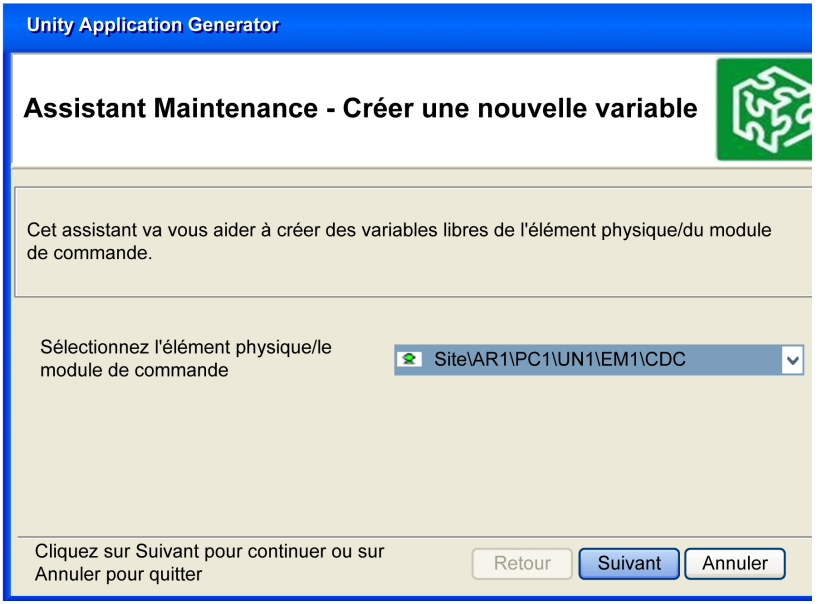
Vous pouvez créer une nouvelle variable dans le module dans Élément physique/Module de commande en cliquant sur **Fichier →Assistants de maintenance →Créer une nouvelle variable** et configurer les propriétés de la variable. Cette option permet à l'utilisateur de créer une nouvelle variable sous l'élément physique ou le module de commande à l'intérieur du projet existant. Les variables créées à partir d'**Assistants de maintenance** sont traitées comme des variables de UAG libres.

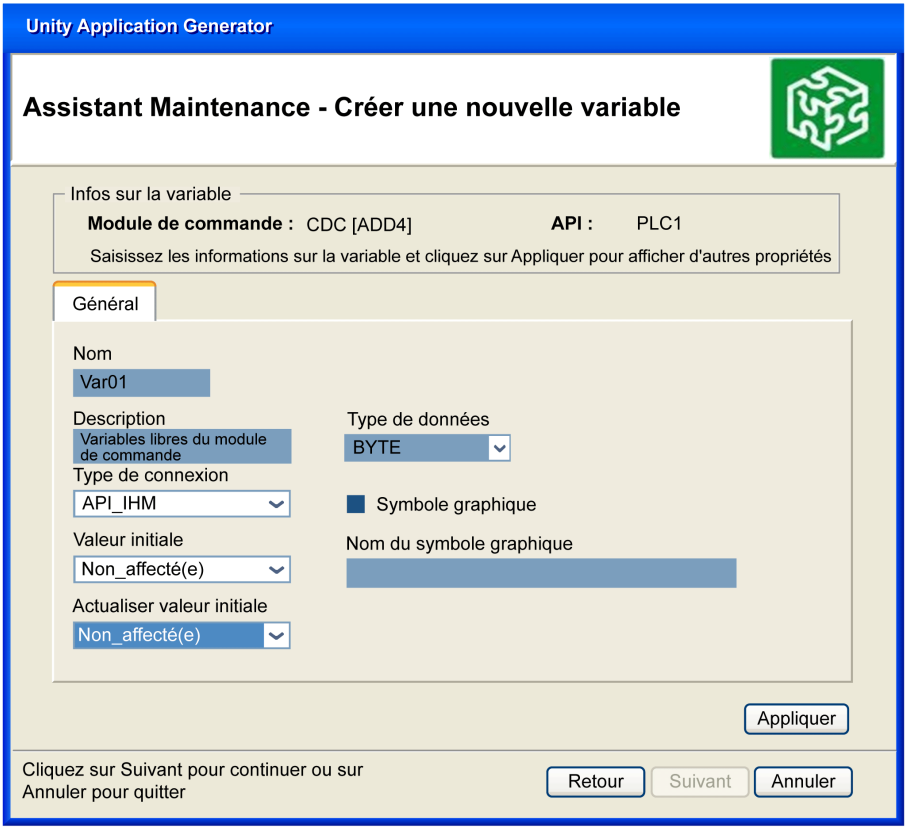
Les règles suivantes s'appliquent à l'option **Créer une nouvelle variable** dans **Assistants de maintenance** :

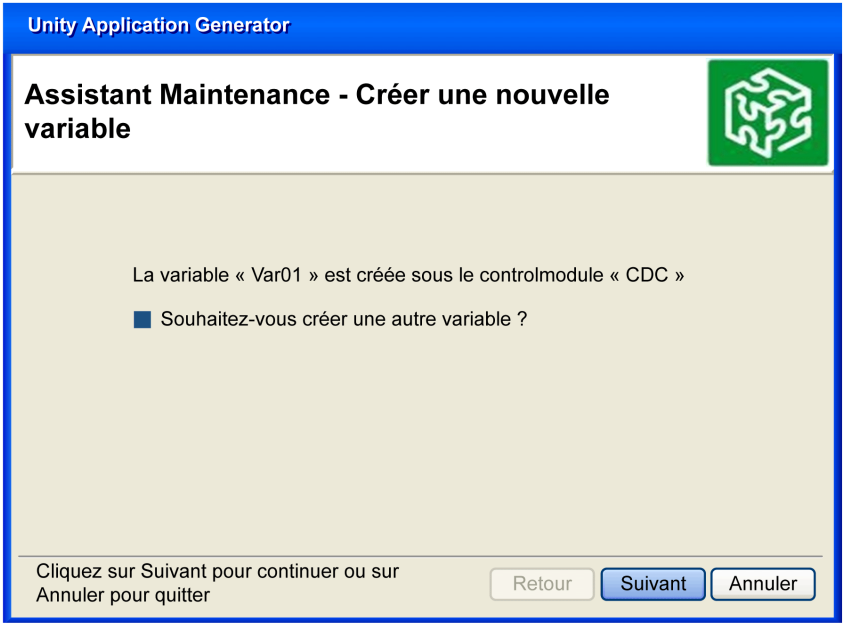
- l'utilisateur peut créer une variable au niveau de l'élément physique et du module de commande du modèle physique en fonction du niveau du conteneur défini dans la personnalisation du projet.

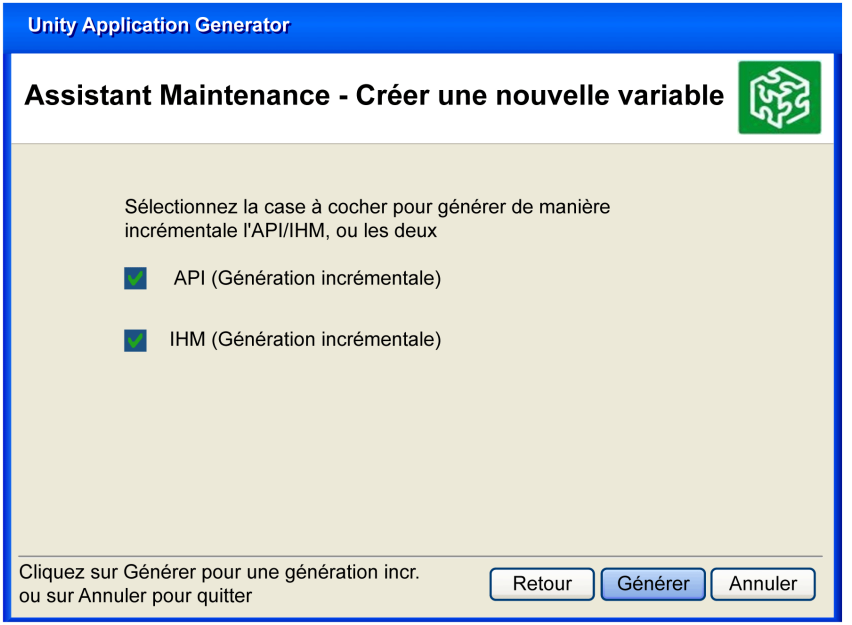
NOTE : Si vous cliquez sur **Annuler** après la création d'une nouvelle variable, les données créées sont enregistrées.

Si vous cliquez sur **Annuler** avant de créer une variable, l'assistant se ferme.

Etape	Action
1	Sélectionnez un élément physique ou un module de commande dans la liste du modèle physique. 
2	Cliquez sur **Suivant** pour continuer.

Etape	Action
3	Saisissez le **Nom** et la **Description** d'une variable dans les champs respectifs. 
4	Sélectionnez les détails des variables tels que **Type de connexion**, **Type de données**, et ainsi de suite à partir des listes respectives.
5	Cliquez sur **Appliquer** pour créer une nouvelle variable. D'autres propriétés de la variable à configurer s'affichent.
6	Cliquez sur **Suivant** pour continuer.

Etape	Action
7	Pour créer une autre variable, sélectionnez l'option **Souhaitez-vous créer une autre variable ?**. 
8	Cliquez sur **Suivant** pour continuer.

Etape	Action
9	Sélectionnez **API**, **IHM** ou les deux options. 
10	Cliquez sur **Générer** pour la génération incrémentale.

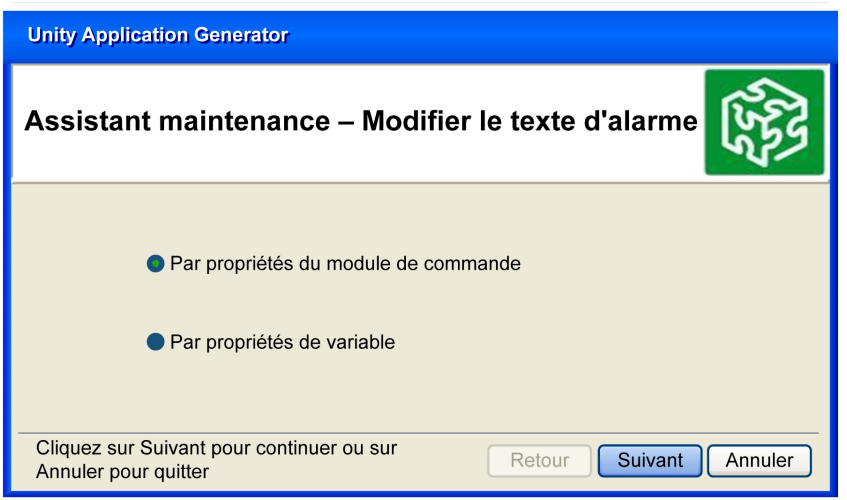
Modification du texte d'alarme

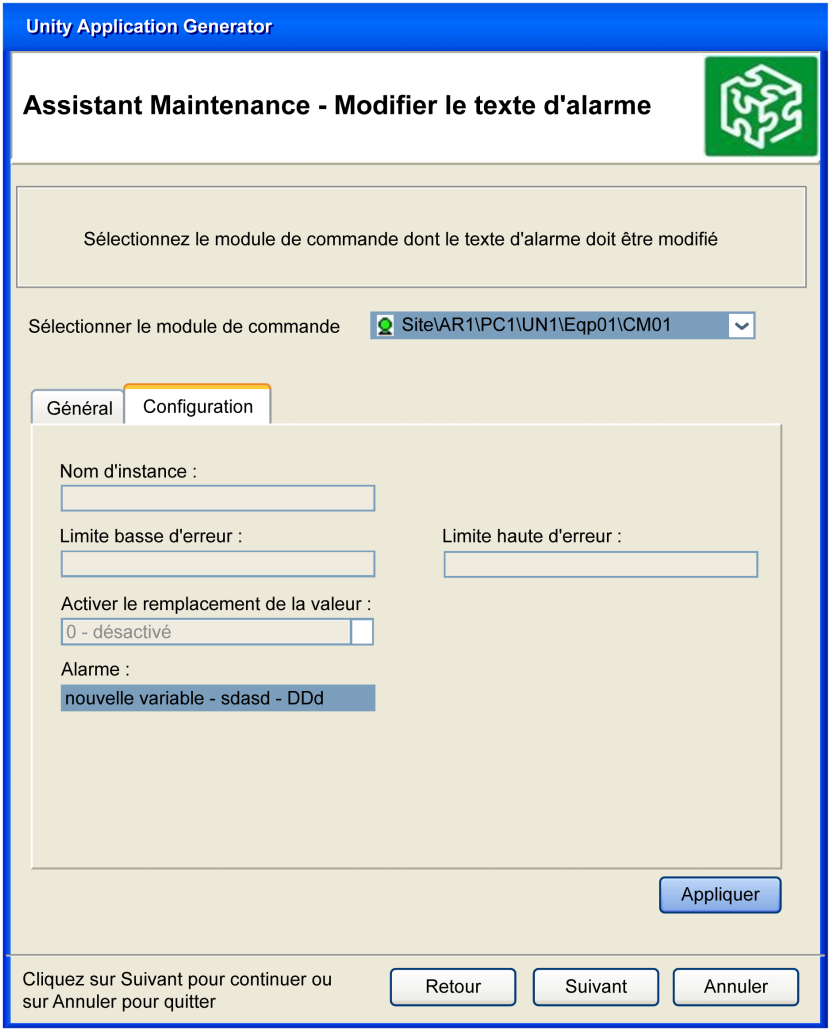
L'option **Par propriétés de variable** ou **Par propriétés du module de commande** permet de modifier le texte d'alarme du module/de la variable de commande en cliquant sur **Fichier** → **Assistants de maintenance** →**Modifier le texte d'alarme**. Les propriétés du texte d'alarme doivent être définies dans ScoD en tant que propriétés du module de commande.

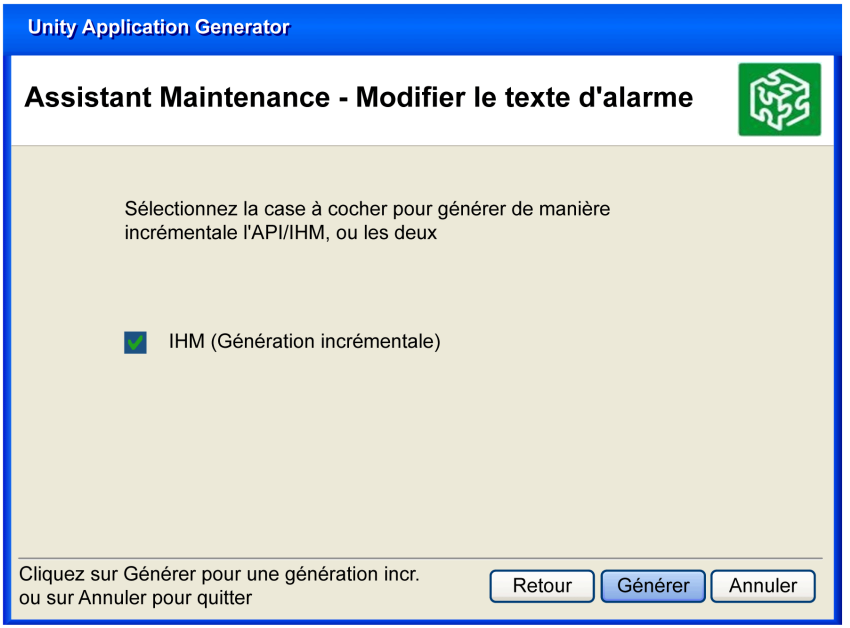
NOTE : Si vous cliquez sur **Annuler** après la modification du texte d'alarme, les données modifiées sont enregistrées.

Si vous cliquez sur **Annuler** avant de modifier le texte d'alarme, l'assistant se ferme.

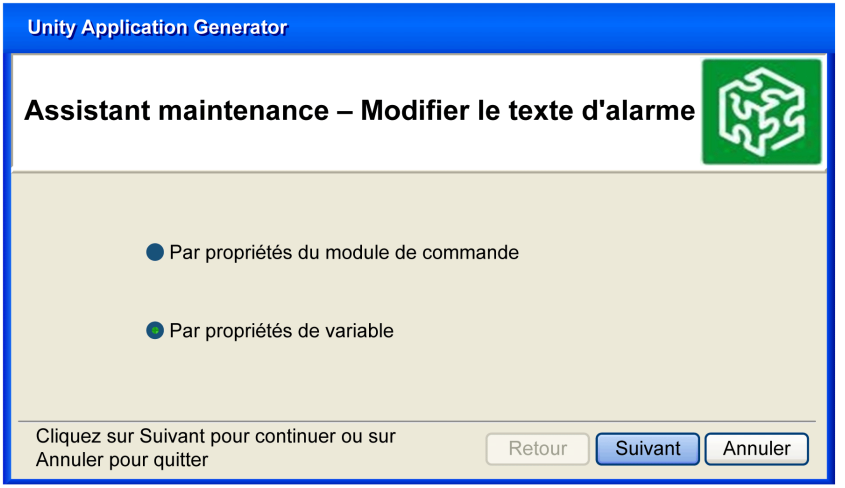
Les étapes suivantes expliquent comment modifier le texte d'alarme à l'aide de l'option **Par propriétés du module de commande** :

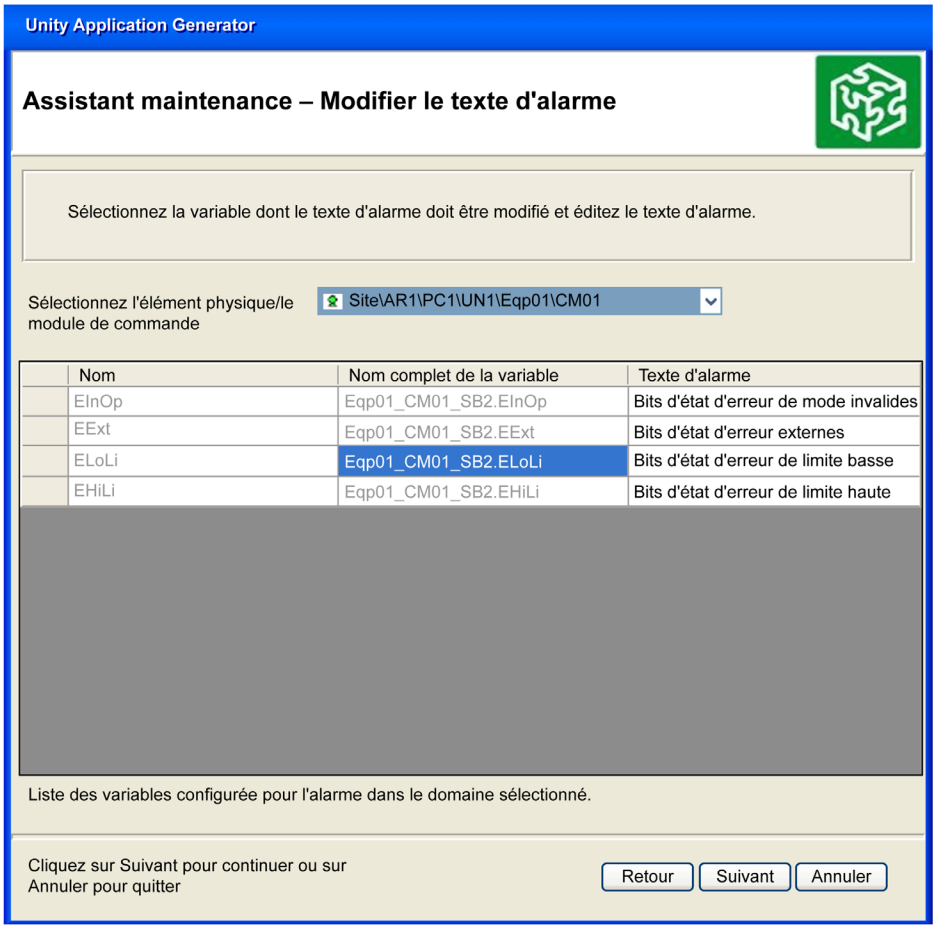
Etape	Action
1	Sélectionnez l'option **Par propriétés du module de commande**. 
2	Cliquez sur **Suivant** pour continuer.

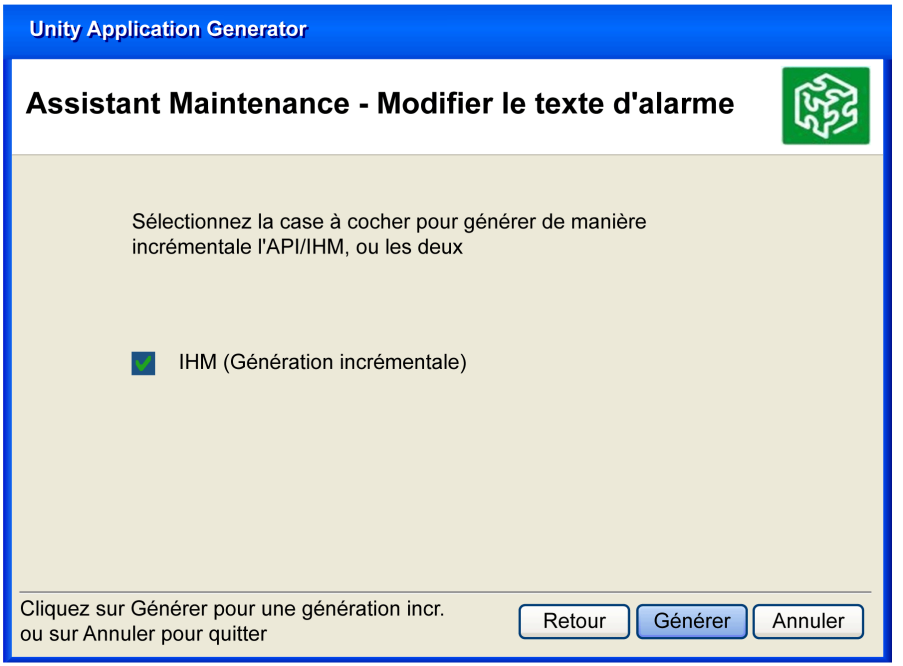
Etape	Action
3	Sélectionnez un module de commande dont le texte d'alarme doit être modifié dans la liste**Sélectionner le module de commande**. 
4	Modifiez le texte d'alarme.
5	Cliquez sur **Appliquer** pour enregistrer la propriété modifiée et sur **Suivant** pour continuer.

Etape	Action
6	Sélectionnez l'option **IHM**. 
7	Cliquez sur **Générer** pour la génération incrémentale.

Les étapes suivantes expliquent comment modifier le texte d'alarme à l'aide de l'option **Par propriétés de variable** :

Etape	Action
1	Sélectionnez l'option **Par propriétés de variable**. 
2	Cliquez sur **Suivant** pour continuer.

Etape	Action
3	Modifiez le texte d’alarme de la variable. 
4	Cliquez sur **Suivant** pour continuer.

Etape	Action
5	Sélectionnez l'option **IHM**. 
6	Cliquez sur **Générer** pour la génération incrémentale.

Documentation de l’application ou d’objets individuels (Rapport)

Présentation

Vous pouvez, à l’aide de la fonction **Rapport**, générer différents rapports pour documenter votre projet, par exemple la configuration du réseau, le modèle physique complet ou les variables d’E/S. Vous avez en outre la possibilité d’imprimer un rapport sur des objets individuels.

Unity Application Generator offre plusieurs propriétés de document (DOCPROPERTY) (par ex. Version UAG, nom de projet UAG etc.) qui peuvent être utilisées dans les fichiers modèles de Microsoft Word (.dot). Vous trouverez la liste finale des propriétés du document dans les remarques de version ReleaseNotes.txt. Vous trouverez un exemple de fichier de modèle (example.dot) dans le répertoire "Doc".

NOTE :

- La documentation est générée sous forme de document Microsoft Word. Il faut pour cela que Microsoft Word soit installé sur le PC sur lequel la documentation doit être générée.
- Dans des projets importants, le rapport du modèle topologique pourrait être erroné. La fonctionnalité de UAG n’est pas affectée.

Etapes à suivre pour la documentation d’objets individuels

Pour générer la documentation pour des objets individuels, suivez les étapes ci-dessous :

Etape	Action
1	Cliquez le bouton droit de la souris sur l’objet sur lequel établir un rapport.
2	Sélectionnez **Rapport** dans le menu contextuel. Résultat : La boîte de dialogue Rapport s’ouvre.
3	Définissez un fichier de modèle Word et une destination pour le fichier de rapport généré.
4	Définissez les sous-objets à inclure dans le rapport (case à cocher).
5	Lancez le générateur de rapport (bouton **Rapport**).

Etapes à suivre pour la documentation de l’application

Pour générer la documentation ou l’ensemble de l’application, suivez les étapes ci-dessous :

Etape	Action
1	Ouvrez la boîte de dialogue de générateur de rapport (**Fichier** →**Rapport**).
2	Définissez un fichier de modèle Word et une destination pour le fichier de rapport généré.
3	Sélectionnez les chapitres à générer.
4	La sélection du chapitre **Vue d’ensemble topologique** permet d’imprimer la **Vue d’ensemble topologique** sur une ou plusieurs pages (boutons d’option).
5	Lancez le générateur de rapport (bouton **Rapport**).

Chapitre 9
Rubriques spécifiques aux bus de terrain

Présentation

Ce chapitre donne des informations sur les sujets particuliers aux bus de terrain.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
9.1	Informations générales sur les bus de terrain	350
9.2	Importation de la configuration de bus de terrain	359
9.3	Configuration de bus de terrain pour Quantum et Premium	368

Sous-chapitre 9.1
Informations générales sur les bus de terrain

Introduction

La section suivante fournit des informations générales sur les bus de terrain pris en charge.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Informations générales sur les bus de terrain	351
Modules de bus de terrain dans le modèle topologique	353
Description de l'interface utilisateur	354
Boîte de dialogue Esclave Profibus	357

Informations générales sur les bus de terrain

Introduction

Unity Application Generator gère la configuration des bus de terrain. La technologie de bus de terrain suivante sera gérée :

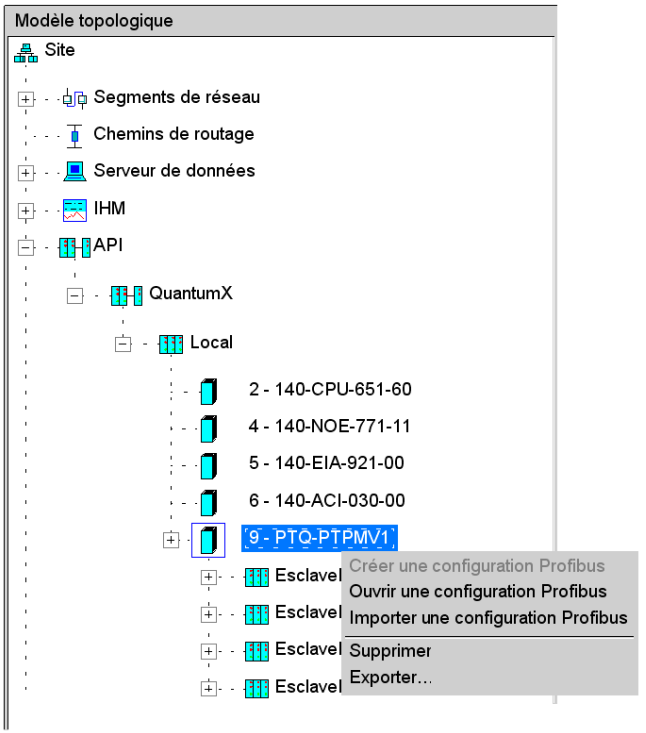
- Profibus

Vous pouvez définir des configurations et gérer tous les paramètres dans UAG. Vous avez besoin d'un outil tiers de chez Prosoft ou Hilscher pour obtenir la configuration.

La liste suivante décrit les outils nécessaires

- Prosoft Configuration Builder (fourni sur votre DVD d'installation de UAG), voir les paragraphes ci-après
- Hilscher Sycon 2.10, voir les paragraphes ci-après

La figure suivante illustre un exemple de l'intégration Profibus dans le modèle topologique de UAG.



NOTE : Si une configuration Profibus existe déjà, l'option de menu **Créer une configuration Profibus** n'est pas active.

Les informations suivantes donnent uniquement les étapes générales permettant d'obtenir une configuration Profibus. Pour plus d'informations, reportez-vous à la section Import of Fieldbus Configuration *(voir page 361)*.

ProSoft Configuration Builder

Vous devez utiliser le logiciel ProSoft Configuration Builder pour configurer le module Quantum Profibus.

Hilscher SyCon

Vous devez utiliser le logiciel Hilscher SyCon pour configurer le module Premium Profibus.

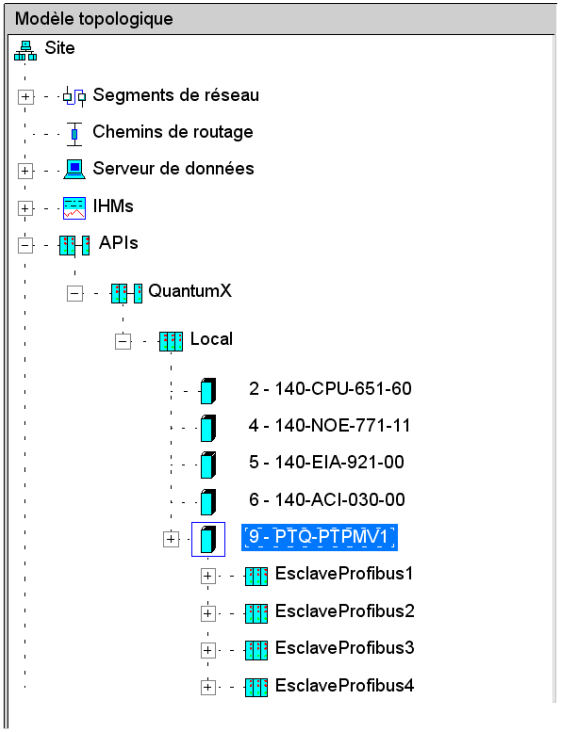
Modules de bus de terrain dans le modèle topologique

Introduction

Les esclaves Profibus sont représentés directement au-dessous du module maître Profibus.

NOTE : Les esclaves Profibus sont modélisés sous la forme de châssis déportés dans UAG.

La figure suivante illustre la représentation d'un module maître Profibus et des esclaves Profibus connectés (modules E/S) dans le modèle topologique :



NOTE : Si vous supprimez le module maître Profibus, les esclaves Profibus associés seront aussi supprimés.

Description de l’interface utilisateur

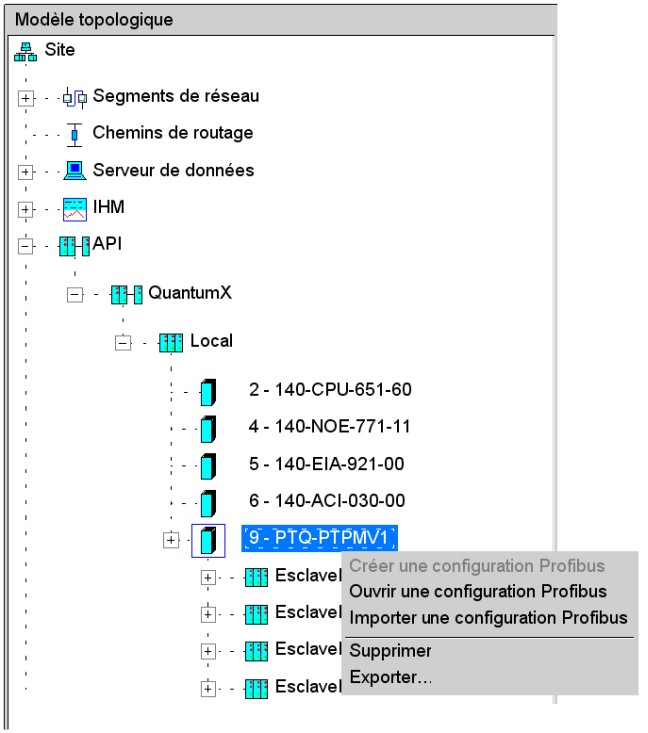
Introduction

Les informations suivantes décrivent l’interface utilisateur de la connexion de bus de terrain de UAG.

Actions de l’interface utilisateur

La configuration du bus de terrain de UAG vous offre différentes actions de paramétrage et de configuration de vos communications par le bus de terrain.

La figure suivante présente un exemple des actions de bus de terrain Profibus :



NOTE : Si une configuration Profibus existe déjà, l’option de menu **Créer une configuration Profibus** n’est pas active.

En ce qui concerne la configuration de bus de terrain, vous pouvez effectuer les opérations suivantes :

- **Créer une configuration Profibus**
  Vous pouvez sélectionner le module maître de bus de terrain dans le modèle topologique et choisir l'entrée **Créer une configuration Profibus** dans le menu contextuel du module.
  NOTE :
- Cette opération est uniquement réalisable si une configuration Profibus n'existe pas encore dans UAG pour ce module maître Profibus. Une configuration Profibus contenant uniquement le maître Profibus est créée. Le chemin d'accès au projet qui vient d'être créé est stocké avec le module maître de bus de terrain dans UAG. L'emplacement et le nom du projet sont automatiquement choisis par UAG.
- Il n'existe pas de dialogue de navigation permettant de sélectionner manuellement le chemin d'accès et le nom du projet.
- **Ouvrir une configuration Profibus**
  Vous pouvez sélectionner le module maître de bus de terrain dans le modèle topologique et choisir l'entrée **Ouvrir une configuration Profibus** dans le menu contextuel du module. Le configurateur de bus de terrain est démarré et la configuration de bus de terrain attribuée au module maître est ouverte.
- **Importer une configuration Profibus**
  Vous pouvez sélectionner le module maître de bus de terrain dans le modèle topologique et choisir l'entrée **Importer une configuration Profibus** dans le menu contextuel du module. La configuration de bus de terrain sera alors importée dans UAG.
  S'il n'existe pas de configuration de bus de terrain lorsque vous démarrez l'importation, UAG ouvre un dialogue de navigation vous invitant à sélectionner un projet existant contenant la configuration de bus de terrain. Vous pouvez sélectionner le même projet pour plusieurs maîtres de bus de terrain.
  Remarque : une erreur est détectée si le fichier projet sélectionné contient plusieurs configurations.
  - **Vérification de l'importation**
    UAG vérifie dans l'opération **Analyse** si la dernière configuration de bus de terrain d'un module maître a déjà fait l'objet d'une importation. Si ce n'est pas le cas, une alerte est émise.
- **Supprimer**
  Vous pouvez sélectionner le module maître de bus de terrain dans le modèle topologique et choisir l'entrée **Supprimer** dans le menu contextuel du module. La configuration de bus de terrain sera alors supprimée.
  Remarque : si vous supprimez un module maître, tous les modules esclaves associés seront automatiquement supprimés.
- **Exporter**
  Vous pouvez sélectionner le module maître de bus de terrain dans le modèle topologique et choisir l'entrée **Exporter** dans le menu contextuel du module. La configuration de bus de terrain sera alors exportée dans un fichier d'export.
  Remarque : vous devez saisir un nom de fichier pour les données à exporter.

Astuces

- Si vous déplacez le module maître à un emplacement différent, les références des esclaves de bus de terrain à ce module maître seront automatiquement mises à jour.
- Vous ne pouvez pas supprimer ni ajouter un esclave de bus de terrain.
  Vous obtenez la configuration des esclaves de bus de terrain en utilisant les fichiers de configuration.
- Vous ne pouvez pas supprimer ni ajouter un module esclave de bus de terrain ni le déplacer à un autre emplacement.
  Vous obtenez la configuration des esclaves de bus de terrain en utilisant les fichiers de configuration.

Archivage de la configuration de bus de terrain

Vous devez inclure les fichiers de configuration de bus de terrain appropriés dans l'archivage du projet UAG au moment où vous archivez l'API contenant le maître de bus de terrain.

Remarque : l'archive doit inclure les informations d'exportation.

Boîte de dialogue Esclave Profibus

Introduction

Vous pouvez changer les modules d'un esclave Profibus dans le dialogue **Ouvrir châssis** de la vue topologique.

Si vous ouvrez un châssis, le dialogue vous montre tous les modules disponibles de l'esclave Profibus particulier. Les modules disponibles seront personnalisés dans l'éditeur de personnalisation.

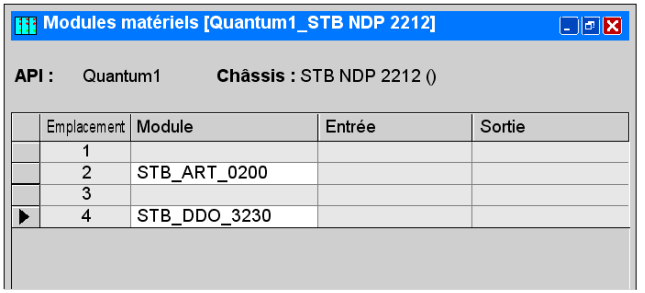
Ouvrez le châssis

Ouvrez le dialogue pour modifier les modules des esclaves Profibus, en procédant comme suit :

Etape	Action
1	Sélectionnez un esclave Profibus dans la vue topologique et cliquez le bouton droit de la souris. **Résultat :** Une boîte de dialogue contextuelle apparaît.
2	Appuyez sur **Ouvrir châssis**. **Résultat :** La boîte de dialogue **Modules matériels** correspondant au module sélectionné, apparaît

Boîte de dialogue Modules matériels et boîte de dialogue Points d'E/S

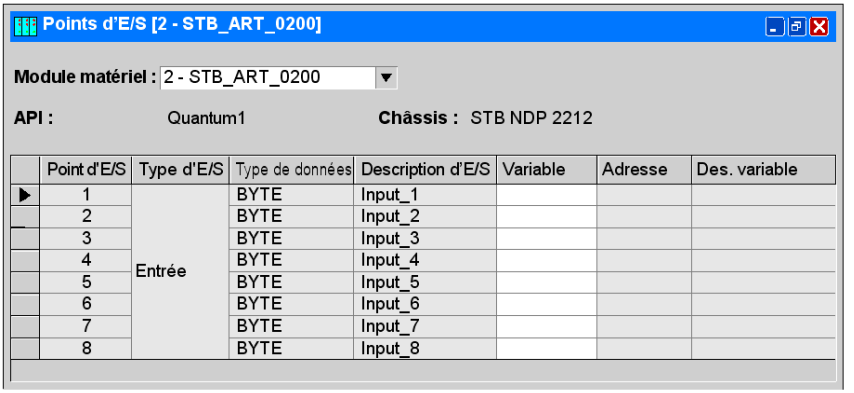
La figure ci-dessous illustre la boîte de dialogue **Modules matériels** :



Vous pouvez relier les variables aux points d'E/S dans le tableau **Points d'E/S**.

Ouvrez le tableau de liaison des variables en double-cliquant sur le module particulier dans la boîte de dialogue **Modules matériels**.

La figure ci-dessous illustre un exemple du tableau **Points d'E/S** :



	Point d'E/S	Type d'E/S	Type de données	Description d'E/S	Variable	Adresse	Des. variable
▶	1	Entrée	BYTE	Input_1			
	2		BYTE	Input_2			
	3		BYTE	Input_3			
	4		BYTE	Input_4			
	5		BYTE	Input_5			
	6		BYTE	Input_6			
	7		BYTE	Input_7			
	8		BYTE	Input_8			

Lien des variables

Pour lier les variables, procédez comme suit :

Etape	Action
1	Fermez tous les dialogues dans le volet de droite de UAG. **Résultat :** Le volet de droite est vide.
2	Ouvrez le tableau **Points d'E/S**. **Résultat :** Le tableau **Points d'E/S** apparaît sur le volet de droite.
3	Ouvrez l'afficheur de variables du module de commande (voir la section *Afficheur de variables, page 300*). **Résultat :** L'afficheur de variables apparaît.
4	Appuyez sur **Fenêtre** →**Mosaïque horizontale** dans la barre de menus de UAG. **Résultat :** L'afficheur de variables et le tableau **Points d'E/S** s'affichent.
5	Faites glisser les variables nécessaires depuis l'afficheur de variables vers le tableau **Points d'E/S**. **Résultat :** Les variables seront affichées dans le tableau **Points d'E/S**.

Sous-chapitre 9.2
Importation de la configuration de bus de terrain

Introduction

La section suivante donne des informations et indique les procédures concernant l'importation de la configuration des différents bus de terrain.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Catalogue de modules Profibus	360
Importation de la configuration de bus de terrain	361
Génération d'une application API	367

Catalogue de modules Profibus

Introduction

Si vous utilisez un module Profibus, vous devez définir les adresses des différents modules. UAG installe un fichier qui contient toutes les définitions des modules Profibus proposés par Schneider Electric.

Le nom du fichier est **Profibus.osm**.

⚠ AVERTISSEMENT

PARAMETRES IMPREVUS DANS L'IMAGE D'E/S DE L'API

Vérifiez manuellement que les modules et les configurations utilisés sont corrects.

Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

Caractéristiques du catalogue de modules Profibus

Le catalogue de modules est utilisé pour définir les points d'E/S pour les modules Profibus. Pendant l'importation des modules, voici ce qui se passe :

Phase	Description
1	UAG lit les données du module à partir de la personnalisation de UAG.
2	Si le module n'existe pas, UAG lit les données à partir du catalogue de modules **Profibus.osm**.
3	Dans ce cas, UAG génère une configuration standard. Dans cette configuration, chaque octet d'E/S et chaque point d'E/S a le type de données `BYTE` **Remarque :** vous ne pouvez pas utiliser cette configuration pour votre projet. C'est simplement un paramètre fictif pour le projet.

Importation de la configuration de bus de terrain

Introduction

Vous pouvez importer la configuration de bus de terrain dans votre projet UAG.

La création d'une configuration de bus de terrain s'effectue à l'aide d'un outil tiers de chez Prosoft ou Hilscher. Les noms des programmes sont :

- Prosoft Configuration Builder (fourni sur votre DVD d'installation de UAG, pour Quantum)

et

- Hilscher Sycon 2.10 (pour Premium)

⚠ AVERTISSEMENT

AFFECTATION DES E/S INCORRECTE.

Si les affectations des E/S de Profibus dans les outils tiers sont différentes de celles de UAG, des valeurs non valides apparaissent dans les paramètres de l'image d'E/S de l'API. Vérifiez les deux affectations d'E/S.

Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

Vérifications au cours de l'importation

Au cours de l'importation de la configuration Profibus, UAG vérifie les adresses d'entrée et de sortie par rapport aux plages d'adresses de bus terrain d'API définies dans UAG (reportez-vous à la section Propriétés d'API *(voir page 151)*).

Si les adresses d'entrée et de sortie se trouvent hors des plages d'adresses de bus de terrain définies dans UAG, l'importation est refusée et des messages correspondants sont affichés.

Deux messages sont possibles :

- "Les adresses d'entrée du module ne se trouvent pas dans la zone d'adresses de bus de terrain d'API. Adresses d'entrée = [%Début, %Fin]. Bus de terrain = [%Début, %Fin].
- "Les adresses de sortie du module ne se trouvent pas dans la zone d'adresses de bus de terrain d'API. Adresses de sortie = [%Début, %Fin]. Bus de terrain = [%Début, %Fin].

Règles d'importation

Si la configuration de bus de terrain a déjà été importée, vous devez conserver les affectations de variable (variables SCoD affectées aux points d'E/S) autant que possible. Vous ne pouvez le faire que si les éléments de la configuration de bus de terrain qui viennent d'être importés correspondent aux éléments existants. Néanmoins, puisqu'en général la configuration de bus de terrain ne comporte pas d'identifications uniques permettant une correspondance exacte, l'importation ne peut pas toujours conserver les affectations des variables. Observez les règles d'importation suivantes :

NOTE : Les actions de la liste suivante conduisent à l'annulation de l'affectation des variables :

- Modification des adresses d'esclave
- Déplacement du module maître Profibus vers un autre emplacement
- Modification des points d'E/S à l'intérieur de l'éditeur de personnalisation

Le tableau ci-dessous décrit les différentes étapes de la création d'une connexion de bus de terrain dans UAG.

L'exemple illustre uniquement la configuration de la partie bus de terrain, non pas celle de l'intégralité du projet Unity Application Generator.

⚠ AVERTISSEMENT
PARAMETRES IMPREVUS DANS L'API Si les configurations d'E/S sont différentes dans UAG et l'API, vérifiez que la configuration de bus de terrain est valide. **Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.**

Etapes de la génération d'une configuration Profibus pour Quantum

S'il n'existe pas de configuration Profibus, procédez comme suit pour configurer un API Quantum avec une communication Profibus :

Etape	Action
1	Ouvrez un châssis Quantum disponible à l'endroit où vous souhaitez ajouter les communications Profibus et ajoutez le module Quantum Profibus PTQ-PDP-MV-1. **Résultat :** Le module sera affiché dans le modèle topologique du châssis Quantum que vous avez sélectionné.
2	Sélectionnez le module Profibus par un clic droit de la souris. **Résultat :** Un menu contextuel s'affiche.

Etape	Action
3	Sélectionnez **Créer une configuration Profibus**. **Résultat :** UAG génère un fichier de configuration Profibus (*PbConfig.pb*) dans un sous-répertoire particulier. Le nom du sous-répertoire est *votre_répertoire_de_projet_UAG\ProfibusConfiguration_xx_Slotyy_zz*. ● *xx* représente le nom de l'API ● *yy* représente le numéro d'emplacement du module maître Profibus ● *zz* représente une identification interne de UAG Si plusieurs modules Profibus ont été configurés, vous pouvez identifier le répertoire le plus récent par l'identification la plus haute. **Remarque :** si vous modifiez le nom d'API dans votre projet UAG, le nom du répertoire ne changera pas.
4	Sélectionnez à nouveau le module Profibus par un clic droit de la souris. **Résultat :** Un menu contextuel s'affiche.
5	Sélectionnez **Ouvrir une configuration Profibus**. **Remarque :** si ProSoft Configuration Builder est déjà démarré, un message d'erreur détectée s'affiche. Toutes les instances de ProSoft Configuration Builder doivent être fermées avant de sélectionner **Ouvrir une configuration Profibus**. **Résultat :** Le ProSoft Configuration Builder s'ouvre.
6	Créez une configuration de module maître Profibus avec le ProSoft Configuration Builder, reportez-vous à la rubrique Steps for Generating with ProSoft Configuration Builder *(voir page 364)*. **Résultat :** Une configuration existe maintenant.
7	Enregistrez la configuration Profibus et fermez Prosoft Configuration Builder.
8	Sélectionnez le module Profibus par un clic droit de la souris dans la vue topologique UAG. **Résultat :** Un menu contextuel s'affiche.
9	Sélectionnez **Importer une configuration Profibus**. **Résultat :** Toutes les informations sur la configuration Profibus seront importées et affichées dans le modèle topologique.

Etapes de la génération avec ProSoft Configuration Builder

Pour générer une configuration Profibus avec ProSoft Configuration Builder, procédez comme suit :

Etape	Action
1	Sélectionnez l'élément **PTQ Profibus Master DPV1** dans l'arborescence de ProSoft Configuration Builder et appuyez sur le bouton droit de la souris. **Résultat :** Un menu contextuel s'affiche.
2	Appuyez sur **Configurer**. **Résultat :** Une boîte de dialogue apparaît.
3	Editez les paramètres suivants du module maître **PTQ Profibus Master DPV1** et appuyez sur **OK**. **Liste des paramètres à éditer :** ● **Numéro d'emplacement** Donnez comme numéro d'emplacement le numéro d'emplacement de la configuration du maître UAG Profibus, voir Steps to Generate a Profibus Configuration for Quantum *(voir page 362)*, étape 3. ● **Registre de début d'entrée/sortie** Définissez des zones d'adresses égales aux zones d'adresses **Bus de terrain** de l'API dans UAG. **Remarque :** si vous utilisez plusieurs modules maîtres Profibus dans un API, vous devez vérifier que les zones d'adresses ne se chevauchent pas. ● **Taille des données d'entrée/sortie** Définissez des valeurs supérieures ou égales à la taille des données cycliques de votre configuration Profibus. **Remarque :** vous pouvez réserver de l'espace aux extensions à venir de votre configuration Profibus. ● **Echange d'octet d'entrée/sortie** Définissez sur `NO` ● **Registre de début de redondance d'UC** Réglez sur `0` **Résultat :** Les paramètres ont été modifiés.
4	Configurez votre réseau Profibus. Consultez la documentation utilisateur de ProSoft Configuration Builder.
5	Enregistrez le fichier de configuration ProSoft Configuration Builder Profibus.

Etapes de la génération d'une configuration Profibus pour Premium

Pour configurer un API Premium avec une communication Profibus, procédez comme suit :

Etape	Action
1	Ouvrez un châssis Premium disponible à l'endroit où vous souhaitez ajouter les communications Profibus et ajoutez le module Premium Profibus. **Résultat :** Le module sera affiché dans le modèle topologique du châssis Premium que vous avez sélectionné.
2	Sélectionnez le module Profibus par un clic droit de la souris. **Résultat :** Un menu contextuel s'affiche.
3	Sélectionnez **Créer une configuration Profibus**. **Résultat :** UAG génère un fichier de configuration Profibus (*PbConfig.pb*) dans un sous-répertoire particulier. Le nom du sous-répertoire est *votre_répertoire_de_projet_UAG\ProfibusConfiguration_xx_Slotyy_zz*. ● *xx* représente le nom de l'API ● *yy* représente le numéro d'emplacement du module maître Profibus ● *zz* représente une identification interne de UAG Si plusieurs modules Profibus ont été configurés, vous pouvez identifier le répertoire le plus récent par l'identification la plus haute. **Remarque :** si vous modifiez le nom d'API dans votre projet UAG, le nom du répertoire ne changera pas.
4	Sélectionnez à nouveau le module Profibus par un clic droit de la souris. **Résultat :** Un menu contextuel s'affiche.
5	Sélectionnez **Ouvrir une configuration Profibus**. **Résultat :** Le logiciel Hilscher SyCon s'ouvre.
6	Configurez votre réseau Profibus. Consultez la documentation utilisateur du logiciel Hilscher SyCon.
7	Créez un fichier d'export pour la configuration Profibus avec le logiciel Hilscher SyCon (**Fichier** →**Export** →**ASCII**). **Résultat :** Un fichier *config.cnf* existe maintenant.
8	Sélectionnez le module Profibus par un clic droit de la souris dans la vue topologique UAG. **Résultat :** Un menu contextuel s'affiche.

Etape	Action
9	Sélectionnez **Importer une configuration Profibus**. **Résultat :** Toutes les informations sur la configuration Profibus seront importées et affichées dans le modèle topologique.

Importation d'une configuration Profibus existante

Vous pouvez importer dans UAG une configuration Profibus déjà existante pour Quantum ou Premium, plutôt que d'en créer une nouvelle. Pour cela, procédez comme suit :

Etape	Action
1	Ouvrez un châssis Quantum disponible à l'endroit où vous souhaitez ajouter les communications Profibus et ajoutez le module Quantum ou Premium Profibus. **Résultat :** Le module sera affiché dans le modèle topologique du châssis Quantum que vous avez sélectionné.
2	Sélectionnez le module Profibus par un clic droit de la souris. **Résultat :** Un menu contextuel s'affiche.
3	Sélectionnez **Importer une configuration Profibus**. **Résultat :** Un dialogue de navigation apparaît.
4	Sélectionnez le fichier de configuration Profibus et appuyez sur **OK**. **Résultat :** La configuration Profibus est importée. **NOTE :** ● Après l'importation, la configuration Profibus est liée au module maître. Si vous appuyez à nouveau sur **Importer une configuration Profibus**, aucun dialogue de navigation ne s'ouvrira. Dans ce cas, c'est ProSoft Configuration Builder ou Hilscher SyCon qui s'ouvre. ● La configuration Profibus doit obéir aux mêmes règles que celles décrites à la rubrique Steps to Generate a Profibus Configuration for Quantum *(voir page 362)* ou Steps to Generate a Profibus Configuration for Premium *(voir page 365)*. ● Le fichier d'exportation doit exister pour que l'importation se déroule normalement.

Génération d'une application API

Introduction

UAG génère automatiquement toutes les données nécessaires aux communications par bus de terrain dans l'API.

Sections pour la conversion des données

UAG crée automatiquement les sections API pour la conversion des données entre la représentation pour le bus de terrain et la représentation pour l'API.

Les sections à créer sont des sections FBD.

Génération du code module par module

La génération du code s'effectue module par module.

UAG vérifie tous les modules pour savoir s'il s'agit de modules définis par l'utilisateur.

La conversion s'effectue de 2 façons différentes selon le résultat de cette vérification.

Sans blocs fonctions définis par l'utilisateur

S'il n'existe pas de blocs fonctions définis par l'utilisateur, les blocs fonctions installés avec UAG sont utilisés (par ex. BYTE_AS_WORD, BYTE_AS_REAL, REAL_AS_BYTE etc.).

Pendant la génération du code, UAG gère les différences suivantes entre la représentation pour le bus de terrain et la représentation pour l'API :

- Ordre des octets
  Les octets dans la représentation pour le bus de terrain peuvent avoir un ordre différent (format Motorola ou Intel).
- Décalage des données
  Dans la représentation pour le bus de terrain, une valeur analogique risque de ne pas suivre l'alignement des mots, et commencer par exemple au milieu d'un mot.

Avec des blocs fonctions définis par l'utilisateur

Vous pouvez définir des blocs fonctions de conversion pour les données d'entrée et de sortie d'un module Profibus dans l'éditeur **Personnalisation de UAG**. Consultez le *manuel utilisateur avancé de UAG*, section "Modules définis par l'utilisateur *(voir Unity Application Generator (UAG) 3.3 SP4, a SoCollaborative software, Manuel utilisateur avancé)*".

Dans ce cas, UAG utilise ces blocs fonctions de conversion.

Les variables, affectées aux points d'E/S du module, sont connectées aux broches des blocs fonctions de conversion définis par l'utilisateur.

Sous-chapitre 9.3
Configuration de bus de terrain pour Quantum et Premium

Introduction

La section suivante fournit des informations sur la configuration des bus de terrain pour les automates Quantum et Premium.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Unity Pro Quantum	369
Unity Pro Premium	370

Unity Pro Quantum

Communication Profibus

UAG importe le fichier *.XSY* qui est exporté de ProSoft dans l'application API. Ce fichier contient :

- 2 variables pour l'échange des données entre l'API et le module maître Profibus ;
- les définitions de type de données de ces variables.

NOTE : Ces variables ne sont pas utilisées par UAG, car UAG utilise directement les adresses d'API pour avoir accès aux données E/S cycliques. Les variables sont toujours importées dans l'application API pour vérifier que l'application API travaille avec les définitions correctes pour l'accès aux données de diagnostic et les communications acycliques.

⚠ AVERTISSEMENT

PAS DE COMMUNICATION D'E/S OU COMMUNICATION VERS UN POINT D'E/S INCORRECT.

La modification manuelle de la configuration du bus de terrain dans Unity Pro ne modifie pas la configuration du bus de terrain dans UAG. Ceci entraîne une incompatibilité dans l'échange de données du bus de terrain. Ne modifiez pas manuellement la configuration du bus de terrain dans Unity Pro.

Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

Unity Pro Premium

Communication Profibus

Unity Pro vous permet d'importer et d'exporter l'intégralité de la configuration Profibus en utilisant la description XML de la configuration matérielle.

UAG ne génère pas les données de l'intégralité de la configuration Profibus, mais uniquement la référence au fichier de configuration *.CNF* à importer. De cette manière, on s'assure que l'application API utilise le même fichier d'importation que UAG. Après la génération, vous devez importer manuellement le fichier dans Unity Pro afin d'assurer la cohérence entre le projet UAG et l'application API.

⚠ AVERTISSEMENT
PAS DE COMMUNICATION D'E/S OU COMMUNICATION VERS UN POINT D'E/S INCORRECT. La modification manuelle de la configuration du bus de terrain dans Unity Pro ne modifie pas la configuration du bus de terrain dans UAG. Ceci entraîne une incompatibilité dans l'échange de données du bus de terrain. Ne modifiez pas manuellement la configuration du bus de terrain dans Unity Pro. **Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.**

Chapitre 10
Intégration Advantys

Introduction

Ce chapitre décrit l'intégration du système d'E/S distribuées Advantys STB.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Intégration des systèmes d'E/S distribuées Advantys STB	372
Importation de la configuration Advantys	373
Catalogue des modules Advantys	375
Affectation des entrées/sorties des modules Advantys	376
Génération du scrutateur d'E/S et de la diffusion des E/S pour le logiciel de programmation d'API	379
Génération de la logique FBD	380
Fonctionnalités améliorées	381

Intégration des systèmes d'E/S distribuées Advantys STB

Introduction

Le système d'E/S distribuées Advantys STB est un système modulaire ouvert, conçu pour l'industrie des machines avec une filière d'évolution vers l'industrie process. Sa configuration s'effectue avec le logiciel de configuration Advantys. UAG est capable d'importer la configuration matérielle Advantys et de l'afficher dans le modèle topologique.

UAG est capable de gérer les points d'E/S des modules E/S Advantys de la manière indiquée comme s'il s'agissait d'un module E/S Ethernet déjà implanté.

La configuration Advantys est intégrée au rapport, à l'exportation, à la recherche et à l'afficheur topologique comme pour tous les autres modules matériels.

La configuration Advantys s'effectue avec le logiciel de configuration Advantys. L'utilisateur configure la topologie matérielle et attribue le comportement et la valeur en cas de timeout (appelés respectivement **mode repli** et **valeur de repli**).

⚠ ATTENTION
AFFECTATION DES E/S IMPREVUE Vérifiez que la configuration des E/S Advantys STB dans UAG est conforme à la dernière version de la configuration E/S Advantys STB et à la description matérielle. Dans le cas contraire, l'affectation des E/S pourrait être incorrecte. **Le non-respect de ces instructions peut provoquer des blessures ou des dommages matériels.**

Importation de la configuration Advantys

Introduction

Pour l'importation Advantys, sélectionnez l'API avec la connexion d'îlot Advantys. Cliquez avec le bouton droit de la souris sur le nom de l'API. Accédez à l'option de menu **Nouveau châssis**. Une boîte de dialogue s'ouvre. Saisissez les paramètres suivants.

- **Type de châssis**
  Indiquez ici le type d'adaptateur utilisé par la connexion. Le choix entre Modbus Plus et Ethernet doit concorder avec l'adaptateur de bus Advantys.
  Les types de lien suivants peuvent être utilisés :
  - **Périphérique MBP 0**
  - **Périphérique MBP 1**
  - **Périphérique MBP 2**
  - **Périphérique Ethernet**
- **Adresse MBP (Modbus Plus) / emplacement du module de communication (Ethernet)**
  Vous devez indiquer ici l'adresse MBP pour Modbus Plus ou l'emplacement de l'adaptateur pour Ethernet.
- **Type de châssis**
  - Advantys STB
- **Fichier de configuration**
  Le chemin d'accès complet et le nom de fichier du fichier de configuration Advantys.
- **Nom du périphérique**
  Le nom de l'îlot Advantys. Le nom par défaut est le nom qui est indiqué dans le fichier de configuration Advantys. Vous pouvez le modifier et entrer un autre nom.
- **Description**
  Entrez ici une description par défaut de tous les châssis générés. La description peut être modifiée ultérieurement dans la boîte de dialogue des propriétés de châssis.

Après avoir défini tous les paramètres, confirmez en cliquant sur **OK**.

UAG importe la configuration Advantys et l'enregistre dans le fichier de projet UAG.

Adresse Modbus Plus pour NIM

Vous devez définir l'adresse du lien Advantys MBP dans la saisie du module NIM dans le gestionnaire du bus de terrain, conformément à la description suivante :

- Lien Advantys MBP = adresse PLC MBP - 1

Comportement à l'importation

L'importation peut uniquement être incrémentale.

Les règles suivantes s'appliquent à l'affectation des E/S des modules :

Si un module est...	Alors l'affectation des E/S de ce module...
amené à un autre emplacement,	est laissée intacte.
supprimé,	est supprimée ; toutes les informations de module seront perdues.
ajouté,	sera nouvellement créée.

NOTE : Si vous souhaitez mettre un nouveau module ou un module existant en un emplacement où un autre module existe déjà, vous devez d'abord supprimer ce dernier module.

Représentation

Après l'importation, la configuration Advantys sera affichée sous la forme d'un châssis doté de plusieurs châssis d'extension. Le châssis et ses châssis d'extension représentent un îlot Advantys. Un châssis représente un rail Advantys. Le nom de l'îlot sera intégré au nom du châssis.

Le fichier de configuration Advantys peut s'interpréter comme étant un modèle de configuration. Vous pouvez importer le même fichier d'îlot à plusieurs reprises tant que les noms des îlots, indiqués dans la boîte de dialogue d'importation, sont différents.

Restrictions

La configuration Advantys ne peut pas être directement créée dans UAG ; elle est uniquement affichée pour les besoins de la documentation, de l'affectation des E/S et de la génération d'API.

NOTE : Il est impossible de supprimer un châssis Advantys individuel ; vous devez supprimer un îlot complet. Toutes les modifications apportées à la configuration doivent être effectuées dans le fichier de configuration Advantys et être importées dans UAG.

Catalogue des modules Advantys

Présentation

UAG ne stocke aucune information sur les modules Advantys. Le nom des modules et la structure des points d'E/S sont extraits du fichier d'îlot Advantys au cours de l'importation. UAG peut donc gérer les nouveaux modules Advantys à partir du moment où ils sont utilisables dans le logiciel Advantys. Aucune mise à jour UAG n'est nécessaire. UAG ne connaît pas la catégorie des modules ni leur description. Ces informations ne se trouvent pas dans le fichier d'îlot Advantys et ne seront donc pas affichées dans UAG pour Advantys.

Affectation des entrées/sorties des modules Advantys

Introduction

Les points d'E/S des modules Advantys peuvent être beaucoup plus complexes que les points d'E/S des modules Quantum, Premium ou M340.

C'est pourquoi UAG est en mesure de gérer des points d'E/S structurés. Un point d'E/S structuré accepte des variables booléennes et des variables mot selon un panachage quelconque et les variables booléennes peuvent être affectées à des variables mot.

Le nombre de bits affectés à un mot est variable. Un point d'E/S structuré représente l'E/S complète d'un module.

Exemple d'affectation d'E/S

Les exemples suivants illustrent l'affectation des E/S de différents modules STB dans UAG.

Type de module matériel	ID E/S	Nb de mot	Nb de bits	Description
STB EHC 3020	Entrée	0	8	Données E/S
STB EHC 3020	Entrée	1	4	Etat E/S
STB EHC 3020	Entrée	2	6	Etat compteur
STB EHC 3020	Entrée	3	5	Etat comparaison
STB EHC 3020	Entrée	4	-1	Compteur d'entrée : Valeur actuelle
STB EHC 3020	Entrée	5	-1	Compteur d'entrée : Valeur capturée
STB EHC 3020	Sortie	0	4	Données de sortie
STB EHC 3020	Sortie	1	3	Validation d'entrée
STB EHC 3020	Sortie	2	5	Direct
STB EHC 3020	Sortie	3	-1	Seuil de sortie : Limite supérieure
STB EHC 3020	Sortie	4	-1	Seuil de sortie : Limite inférieure

L’exemple ci-dessus est représenté dans UAG dans la boîte de dialogue suivante :

Points d'E/S [11 - STB EHC 3020]

Module matériel : Advantys_Rail1_Slot11_STB EHC3

API : UQuan002 **Châssis :** Advantys_Rail1

	E/S	Type d'E/S	Type de données	Description d'E/S	Variable	Adresse	Description var...
	1.1	INPUT		Données E/S			
	1.2						
	1.3						
	1.4						
	1.5						
	1.6						
	1.7						
	1.8						
	2.1			Etat E/S			
	2.2						
	2.3						
	2.4			Etat compteur			
	3.1						
	3.2						
	3.3						
	3.4						
	3.5						
	3.6						
	4.1			Etat comparaison			
	4.2						
	4.3						
	4.4						
	4.5						
	5			Compteur d'entrée : C...			
	6			Compteur d'entrée : C...			
	1.1	OUTPUT		Données de sortie			
	1.2						
	1.3						
	1.4						
	2.1			Validation d'entrée			
	2.2						
	2.3						
	3.1			Direct			
	3.2						
	3.3						
	3.4						
	3.5						
	4			Seuil de sortie...			
	5			Seuil de sortie...			

Cartographie mémoire

La cartographie mémoire attribue des adresses aux modules matériels et aux points d'E/S Advantys. Les variables affectées à un point d'E/S du module E/S Advantys n'ont pas d'adresse.

Pour chaque îlot Advantys Ethernet, UAG réserve 235 mots. Le découpage des mots est le suivant :

- 35 mots
  Pour les données de diagnostic
- 100 mots
  Pour les données d'entrée
- 100 mots
  Pour les données de sortie

Pour chaque îlot Advantys Modbus Plus, UAG réserve 264 mots. Le découpage des mots est le suivant :

- 132 mots
  Pour les données d'entrée (32 mots peuvent être lus par la diffusion des E/S et 100 mots par le MSTR)
- 132 mots
  Pour les données de sortie (32 mots peuvent être écrits par la diffusion des E/S et 100 mots par le MSTR)

Génération du scrutateur d'E/S et de la diffusion des E/S pour le logiciel de programmation d'API

Introduction

Pour pouvoir lire ou écrire les points d'E/S Advantys, il faut générer des entrées de communication dans le logiciel de programmation d'API. Selon le module de communication utilisé dans la communication Advantys, UAG doit générer une entrée de scrutateur d'E/S (Ethernet) ou des entrées de diffusion des E/S (Modbus Plus).

Pour chaque module Ethernet, UAG génère deux entrées de scrutateur d'E/S. La première entrée (35 mots) lit les données de diagnostic de l'îlot. La seconde lit et écrit les données d'E/S. La longueur du bloc de lecture et d'écriture du scrutateur d'E/S est constante (100 mots chacun).

Pour chaque module Modbus Plus qui est en communication avec un API Quantum ou Momentum, UAG génère une entrée globale de diffusion des E/S pour le bloc de lecture (32 mots) et une entrée spécifique de diffusion des E/S pour le bloc d'écriture (32 mots).

Pour chaque module Modbus Plus qui est en communication avec un API Premium, UAG génère uniquement une entrée spécifique de diffusion des E/S pour le bloc d'écriture (32 mots).

NOTE : L'entrée globale de diffusion des E/S pour Premium n'est pas prise en charge par UAG et l'utilisateur doit la configurer manuellement.

Génération de la logique FBD

Introduction

UAG génère une section spéciale pour chaque rail Advantys. Dans cette section spéciale, UAG génère l'affectation de l'adresse d'un module matériel à la variable.

Les modules Advantys stockent les entrées et les sorties numériques dans des variables mot. Dans le logiciel de programmation d'API, ces informations sont affectées à des variables booléennes. Cette opération est effectuée par une logique FBD générée par UAG.

UAG génère une section pour chaque îlot Advantys. Les sections générées sont regroupées dans le module fonctionnel de configuration Advantys. UAG génère un FFB pour chaque mot de la table des points d'E/S affecté au module matériel E/S numérique.

La convention de nommage suivante est utilisée :

NomModuleFonctionnel = ConfigurationPériphériqueConfigurable

NomSection = CDv_<NomPériphérique>_<Numéro>

NomFBI = <NomPériphérique>_<NuméroExtensionChâssis>_<NuméroEmplacement>_<IndicateurEntrée/Sortie> (**_E** pour l'entrée, **_S** pour la sortie)

Les types de FFB suivants sont utilisés :

Pour Unity Pro :

Entrée TOR : **INT_TO_BIT**

Sortie TOR : **BIT_TO_INT**

Fonctionnalités améliorées

Introduction

Les fonctionnalités générales suivantes de UAG concernant la configuration matérielle ont été améliorées pour Advantys.

Rapport

La fonction rapport a dû être améliorée pour pouvoir gérer la configuration Advantys. Le rapport de catégorie et de description des modules sera supprimé pour Advantys.

Importer/exporter

Les fonctions Importer/exporter ne sont pas gérées pour Advantys. Etant donné que les modules Advantys ne sont pas connus de UAG, leur validité ne peut pas être vérifiée par UAG. Pour éviter tout conflit, l'importation est interdite.

Afficheur topologique

La configuration Advantys est affichée dans la vue topologique. Etant donné que les modules Advantys ne sont pas connus de UAG, tous les modules Advantys ont un bitmap unique. Le code couleur utilisé par les autres modules pour visualiser le type de module ne peut pas être utilisé pour les modules Advantys. De plus, l'abréviation du nom de module ou de son type n'est pas visible à l'intérieur de l'objet bitmap.

Rechercher

Vous pouvez effectuer une recherche de modules matériels sur la base du nom de module. Outre la sélection d'un nom de module dans une zone de liste modifiable, vous pouvez entrer une chaîne de texte libre qui sera interprétée comme un nom de module. C'est obligatoire car les modules Advantys ne sont pas connus du catalogue de matériel UAG.

Partie III
Gestion de projet

Introduction

Cette partie explique comment gérer un nouveau projet en utilisant Unity Application Generator ainsi que les tâches que l'administrateur de projet doit accomplir.

Les caractéristiques indiquées ci-dessous offrent des possibilités très intéressantes pour assurer une gestion de projet moderne :

- Fusion de projets
- Interfaces ouvertes (importation/exportation via des fichiers CSV ou XML)
- Gestion IHM
- Outil de personnalisation (éditeur de personnalisation de Unity Application Generator)

Contenu de cette partie

Cette partie contient les chapitres suivants :

Chapitre	Titre du chapitre	Page
11	Gestion du développement de projets distribués – Fusion de projets	385
12	Interfaces avec d'autres outils (fonctions Importer et Exporter)	397
13	Types de données dérivés	415

Chapitre 11
Gestion du développement de projets distribués – Fusion de projets

Introduction

Ce chapitre traite de l'ingénierie participative (travaux d'ingénierie déportés, multi-site).

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Présentation	386
Génération de code pour des API individuels	387
Schéma du développement de projet distribué	388
Conditions générales préalables à la fusion de projets	391
Fusion des modèles physiques	393
Fusion des modèles topologiques	394

Présentation

Gestion du développement de projets distribués – fusion de projets

Unity Application Generator prend en charge l'ingénierie collective (travaux d'ingénierie déportés, multi-site) grâce aux fonctionnalités suivantes :

- Fusion de projet : un projet d'automatisation est divisé en plusieurs projets partiels développés par des groupes d'étude indépendants. Il est possible de fusionner facilement des projets partiels en un projet global unique.
- Génération de code pour des API individuels

Caractéristiques de la fonctionnalité de fusion

La fonction de fusion présente les caractéristiques suivantes :

- Tous les objets du projet à fusionner seront créés dans le projet global exactement tels qu'ils sont définis dans le projet à fusionner.
- L'une des tâches de développement consiste à renseigner la logique d'API générée et les applications IHM. C'est pourquoi la fusion de projet comprend les projets Unity Pro existants, les serveurs de données et les IHM appartenant au projet fusionné.
- Afin d'être en mesure de fusionner deux projets développés séparément, plusieurs conditions préalables doivent être réunies. Avant que la fusion n'ait lieu, les deux projets seront analysés pour déterminer s'ils remplissent ces conditions. Si l'analyseur a détecté des erreurs, la fusion est annulée et l'utilisateur doit d'abord corriger ces erreurs.
- Lorsque l'utilisateur choisit de fusionner deux projets, Unity Application Generator demande d'enregistrer le projet en cours comme version d'archive afin de conserver une sauvegarde du projet existant en cas de résultats non désirés.

Génération de code pour des API individuels

Génération de code pour des API individuels

Il est possible de sélectionner un API particulier et de générer le code de cet API à l'aide d'une option du menu contextuel.

Cela est particulièrement utile dans le cas d'un développement collectif, dans le cadre duquel différents ingénieurs ou groupes sont responsables de différents API. Cette caractéristique peut également s'avérer utile pour un ingénieur automaticien dans son travail de génération des API, l'un après l'autre.

Que génère-t-on ?

Si vous effectuez la génération pour un API individuel, seules les variables de cet API sont affectées aux adresses mémoire, aucune autre variable n'est concernée.

Règle

Vérifier que les projets API sont générés de façon cohérente et qu'ils sont chargés correctement sur les API.

Voir : *Comment visualiser l'état de la génération, page 272*

⚠ AVERTISSEMENT

FONCTIONNEMENT IMPREVU DE L'APPAREIL

Attention lors de la génération d'API individuels. Vérifier que les projets et les configurations API sont générés de façon cohérente et qu'ils sont chargés sur tous les API intervenant dans l'application.

Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

Schéma du développement de projet distribué

Phase 1 : Scinder le travail

Il faut commencer par scinder le travail des ingénieurs procédé en différents groupes, comme décrit dans les étapes suivantes :

Etape	Action
1	Définir la personnalisation.
2	Créer un nouveau projet Unity Application Generator.
3	Définir la configuration du réseau par des segments réseau et des chemins de routage.
4	S'il n'est prévu qu'une seule application IHM pour l'intégralité du projet, définissez-la maintenant. S'il doit y avoir d'autres applications IHM, laissez chaque groupe d'étude définir son application IHM ultérieurement.
5	Définir la hiérarchie du modèle physique jusqu'au niveau de responsabilité de chaque groupe.
6	Définir clairement quel groupe est responsable de quelle partie de l'automatisation de l'usine.
7	Créer une copie du projet pour chaque groupe, chaque copie (projet partiel) ayant un nom différent.

Directives à suivre pour le partage

Ne pas scinder les objets suivants :

- Eléments physiques fonctionnels
- API
- IHM
- Serveurs de données

  Remarque : même si l'ensemble du système est conçu pour fonctionner avec un seul serveur de données, il est toujours nécessaire de définir un serveur de données dans chaque projet partiel (car il n'est pas possible de fusionner deux fichiers iFIX PDB). Après la fusion, les serveurs de données doivent être fusionnés manuellement dans le projet global, voir phase 4, étape 3.

Phase 2 : Travail indépendant des groupes

Chaque groupe travaille maintenant indépendamment afin de réaliser sa part du travail dans la conception du procédé :

Etape	Action
1	Définir les éléments suivants pour les parties de projet relevant de leur responsabilité : ● les éléments physiques fonctionnels, les modules de commande et les variables avec différent noms ● les API avec des noms différents et des configurations d'API différentes ● le lien des objets du modèle physique avec les objets du modèle topologique ● les serveurs de données avec des noms différents ● les partenaires réseau avec des noms différents ● les IHM (si elles n'ont pas été définies globalement en phase 1) ● la communication entre les API (dans le cadre des API définis par le groupe)
2	Générer les API pour chaque projet partiel. **Remarque** : il est possible de générer du code pour des API individuels, reportez-vous à la section *Génération de code pour des API individuels, page 387*.
3	Compléter les applications Unity Pro manuellement autant que souhaité dans chaque groupe de projet. **Remarque** : les changements manuels effectués sous Unity Pro ne seront **pas** perdus. Ils seront récupérés dans le projet global lors de la fusion des parties de projet.
4	Copier les projets Unity Pro appartenant aux projets partiels dans le chemin d'accès au projet API du projet global.
5	Dans le cas de l'utilisation d'IHM iFIX : **1.** Générer pour IHM iFIX pour tous les projets partiels. **2.** Copier les fichiers iFIX GRF de tous les projets partiels dans le répertoire local DYNAMICS/PIC. **3.** Copier les fichiers iFIX PDB de tous les projets partiels dans le répertoire local DYNAMICS/PDB.

Phase 3 : Fusion de projet

Lorsque chaque groupe a terminé son travail, les projets peuvent être fusionnés.

Etape	Action
1	Ouvrir l'un des projets en accès exclusif.
2	Lancer la fusion des projets avec **Fichier** →**Fusionner** et sélectionner le projet à fusionner. **Résultat** : Une analyse des projets sera effectuée tant pour leur cohérence que pour l'unicité des noms et des adresses. Si des erreurs sont détectées, les projets ne seront pas fusionnés et les erreurs détectées devront être corrigées. Si aucune erreur n'est détectée, les projets fusionneront. **Remarque** : les versions archivées et l'historique des changements des projets à fusionner **ne seront pas** copiés vers le projet global.
3	Répéter l'étape 2 pour tous les projets partiels.

Règles à suivre après la fusion de projet

Une fois les projets fusionnés, respectez les règles suivantes :

- Après la fusion des projets, les projets partiels ne doivent plus être utilisés ou modifiés.
- Toutes les autres modifications sont à effectuer **uniquement** dans le projet global.

NOTE : Il n'est pas possible de répéter la fusion car le projet global contient déjà les objets des projets à fusionner (par exemple, les éléments physiques fonctionnels ou les API) ce qui engendrerait la détection d'erreurs lors de l'analyse des projets à fusionner.

Phase 4 : Terminer le projet global

Il faut, après fusion, compléter le projet global :

Etape	Action
1	Définir la communication entre les API ayant appartenu aux différents projets partiels.
2	Définir le forçage entre les modules de commande ayant appartenu aux différents projets partiels.
3	Si vous aviez créé un serveur de données pour chaque projet partiel, mais que le projet global ne doit avoir qu'un seul serveur de données, fusionner les serveurs de données comme suit : **1.** Supprimer tous les canaux existants du ou des serveur(s) de données obsolètes. **2.** Créer de nouveaux canaux entre les API et le serveur de données finalement souhaité. **3.** Supprimer le ou les serveur(s) de données obsolète(s). **Remarque** : aucun travail manuel supplémentaire n'est nécessaire (tel que recréer les tableaux de communication) et aucune donnée n'est perdue.
4	Générer le projet de manière incrémentale.

Conditions générales préalables à la fusion de projets

Présentation

Lors de la fusion de deux projets, Unity Application Generator vérifie certaines conditions générales avant que la fusion ne soit possible. Pour fusionner deux projets, ouvrez le projet principal et fusionnez un autre projet dans le projet ouvert (en cours). Si vous voulez fusionner d'autres projets, fusionnez-les l'un après l'autre dans le projet en cours.

A la fusion de deux projets, les restrictions suivantes s'appliquent :

Version logicielle

Les projets à fusionner doivent avoir été créés avec la même version de Unity Application Generator.

Marque d'IHM

La marque d'IHM doit être la même dans les deux projets.

Bibliothèques de SCoD

- Les bibliothèques de SCoD utilisées dans les deux projets doivent être les mêmes.
- Les bibliothèques de SCoD et les versions de SCoD des deux projets doivent être les mêmes.

Bibliothèques de modèles

- Les bibliothèques de modèles utilisées dans les deux projets doivent être les mêmes.
- Les bibliothèques de modèles et les versions de types de modèles des deux projets doivent être les mêmes.

Ouverture des projets

- Le projet en cours doit être ouvert sans partage.
- Le projet à fusionner doit actuellement être ouvert par un utilisateur.
- Le code de tous les objets générés doit être cohérent. Cela signifie que les listes d'objets à générer doivent être vides. Reportez-vous à *Comment visualiser l'état de la génération, page 272*.

Personnalisation

Il n'est pas nécessaire que les deux projets aient la même personnalisation. C'est pourquoi après une fusion réussie, il peut s'avérer nécessaire de procéder à une analyse de la personnalisation du projet. Unity Application Generator détecte ce cas de figure et propose à l'utilisateur une analyse de personnalisation.

Conditions préalables à la personnalisation

Il existe une arborescence physique pour chaque projet complet, (A,B). Lors de la fusion des projets, appliquez les règles suivantes :

- La définition du modèle physique dans la personnalisation doit être la même pour les deux projets. Jusqu'à la version 2.3 d'UAG, cela était automatiquement le cas. Etant donné qu'un seul et même projet est issu de la fusion, les définitions d'arborescence ne doivent pas être différentes.
- La règle d'unicité associée à la personnalisation du projet A s'applique à tous les niveaux.
  - Si un niveau doit être unique et qu'il possède un nom identique dans A et B, il convient de conserver l'élément physique de A (l'élément B et ses éléments enfants ne sont pas fusionnés).
  - S'il n'est pas nécessaire qu'un niveau soit unique et si ce niveau porte le même nom dans A et B, l'élément physique de B est fusionné avec A, si l'élément n'existe pas au niveau de ce point de l'arborescence physique A. Sinon, l'élément physique de A doit être conservé (l'élément B et ses éléments enfants ne sont pas fusionnés).

Adresses sur le réseau

Les adresses réseau des abonnés réseau du projet à fusionner doivent être différentes des adresses du projet en cours.

Fusion des modèles physiques

Présentation

Tous les objets sont identifiés par leur nom. Ainsi, si deux objets sont détectés avec le même nom, ils sont considérés comme étant un seul et même objet, du moins du point de vue de la fonction fusion.

NOTE : Il peut exister des objets identiques dans les deux projets (identiques par leur nom). La fusion conservera dans ce cas les objets existants dans le projet en cours et **ne vérifiera pas** les incohérences entre les propriétés des objets dans les deux projets.

Fusion d'éléments et règles

Pour les éléments du modèle physique, certaines règles doivent être respectées pour une fusion correcte :

Elément	Règles	Procédure de fusion
Elément physique structurel	Les hiérarchies peuvent être identiques, se recouper partiellement ou être totalement différentes dans les deux projets. Si elles se recoupent partiellement, ces objets doivent être identiques.	La fusion veillera à combiner la hiérarchie des deux projets de façon appropriée.
Elément physique fonctionnel	Les noms d'éléments physiques fonctionnels dans les deux projets doivent être différents.	L'élément physique fonctionnel du projet à fusionner sera créé à l'identique dans le projet global.
des modules de commande	Les noms des modules de commande doivent être uniques au sein d'un même élément physique fonctionnel. Il peut y avoir deux modules de commande ou plus, portant le même nom, dans les deux projets tant qu'ils appartiennent à des éléments physiques fonctionnels différents.	Les modules de commande du projet à fusionner seront créés à l'identique dans le projet global.
Variables	Les noms des variables doivent être uniques au sein d'un même élément physique fonctionnel ou module de commande. Il peut y avoir deux variables ou plus, portant le même nom, dans les deux projets tant qu'ils appartiennent à des éléments physiques fonctionnels ou des modules de commande différents.	Les variables du projet à fusionner seront créés à l'identique dans le projet global.
Liste d'instruments	Il est possible d'avoir deux modules de commande ou plus de même nom dans les listes d'instruments des deux projets, car les noms des modules de commande de la liste d'instruments ne doivent pas nécessairement être uniques.	Les modules de commande dans la liste d'instruments du projet à fusionner seront créés à l'identique dans le projet global.

Fusion des modèles topologiques

Présentation

Tous les objets sont identifiés par leur nom. Ainsi, si deux objets sont détectés avec le même nom, ils sont considérés comme étant un seul et même objet, du moins du point de vue de la fonction fusion.

NOTE : Il peut exister des objets identiques dans les deux projets (identiques par leur nom). La fusion conservera dans ce cas les objets existants dans le projet en cours et **ne vérifiera pas** les incohérences entre les propriétés des objets dans les deux projets.

Fusion d'éléments et règles

Pour les éléments du modèle topologique, certaines règles doivent être respectées pour une fusion correcte :

Elément	Règles	Procédure de fusion
Segments de réseau	Les même segments de réseau peuvent exister dans les deux projets. Si des segments de réseau de même nom existent dans les deux projets, ils doivent être identiques.	Si des segments de réseau existent dans le projet à fusionner et non dans le projet courant, ils seront créés dans le projet global.
Chemins de routage	Les mêmes chemins de réseau global peuvent exister dans les deux projets. Pour un nouveau chemin de routage, les adresses sur le réseau doivent être uniques.	Si des chemins de routage existent dans le projet à fusionner et non dans le projet courant, ils seront créés dans le projet global.
API	• Les API dans les deux projets doivent être différents. • Les projets Unity Pro doivent être générés avant la fusion. • Les projets Unity Pro appartenant au projet à fusionner doivent être copiés manuellement vers le chemin du projet API du projet global.	Les API du projet à fusionner seront créés à l'identique dans le projet courant. La fusion veillera à créer toute information d'administration permettant à l'utilisateur de continuer à générer de manière incrémentale les API dans le projet global.
IHM	• Les IHM des deux projets peuvent être identiques. • iFIX : les écrans IHM (fichiers GRF) doivent être générés avant la fusion et les fichiers iFIX GRF doivent être copiés manuellement dans le répertoire local DYNAMICS\PIC.	Si des IHM existent dans le projet à fusionner et non dans le projet courant, elles seront créées dans le projet global. La fusion veillera à créer toute information d'administration permettant à l'utilisateur de continuer à générer de manière incrémentale les IHM dans le projet global.

Elément	Règles	Procédure de fusion
Zones de conduite	Les zones de conduite dans les deux projets peuvent être les mêmes tant qu'elles sont rattachées à la même IHM.	
Serveurs de données	● Les serveurs de données dans les deux projets doivent être différents. ● iFIX : les serveurs de données (bases de données de variables = fichiers PDB) doivent être générés (en tant que partie d'IHM) avant la fusion et les fichiers iFIX PDB doivent être copiés manuellement dans le répertoire local DYNAMICS\PDB.	Les serveurs de données du projet à fusionner seront créés à l'identique dans le projet global.
Partenaires réseau	Les partenaires réseau dans les deux projets doivent être différents.	Les partenaires réseau du projet à fusionner seront créés à l'identique dans le projet global.
Canaux	Les canaux dans les deux projets doivent être différents puisque tous les canaux appartiennent à des API.	Les canaux du projet à fusionner seront créés à l'identique dans le projet global.

Chapitre 12
Interfaces avec d'autres outils (fonctions Importer et Exporter)

Introduction

Unity Application Generator offre pour l'importation et l'exportation de données vers et depuis sa base de données les 2 formats de fichiers suivants :

- fichiers CSV
- fichiers XML

Ce concept d'interfaces ouvertes facilite le développement de projets en autorisant l'échange de données avec d'autres outils tels qu'AutoCAD.

Veuillez également vous reporter aux sections :

- Format des fichiers CSV d'importation et d'exportation *(voir page 445)*
- Format des fichiers XML d'importation et d'exportation. *(voir page 483)*

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
12.1	Informations générales sur l'importation/exportation	398
12.2	Exemples d'importation/Exportation CSV	407
12.3	Importer/exporter XML	413

Sous-chapitre 12.1
Informations générales sur l’importation/exportation

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Concept d’interfaces ouvertes	399
Exportation d’un projet UAG	401
Importer dans un projet UAG	403
Mode d’importation/exportation de fichiers CSV/XML	404
Importation incrémentielle	406

Concept d'interfaces ouvertes

Interfaces ouvertes

Unity Application Generator propose des interfaces ouvertes pour importer et exporter des donnés depuis et vers d'autres applications.

Il est possible d'importer et d'exporter des objets à la fois pour le modèle physique et pour le modèle topologique.

Les données sont échangées soit par des fichiers CSV (**C**omma **S**eparated **V**alue) soit par des fichiers XML (E**X**tensible **M**arkup **L**anguage).

Les deux formats spécifiques se trouvent dans Unity Application Generator.

Lors du processus d'importation, Unity Application Generator vérifie les données avant de les incorporer dans la base de données. Les données en erreur sont ignorées et l'utilisateur est informé des erreurs détectées par un message.

Exemples

Exemples d'importation :
- Importation d'instruments depuis le schéma de tuyauterie et d'instrumentation, créé par exemple sous AutoCAD
- Importation de valeurs initiales de variables existantes à des fins de réglage

Exemples d'exportation :
- Exportation d'un API pour créer des schémas de câblage sous AutoCAD
- Exportation des variables pour un partenaire réseau

Fichier CSV

Un fichier CSV (comma separated value) est un fichier ASCII contenant un certain nombre de lignes. Chaque ligne d'un fichier CSV contient un certain nombre de champs séparés par un séparateur, par exemple une virgule, un point-virgule ou une tabulation.

Format des fichiers CSV

Les fichiers CSV d'importation et d'exportation depuis et vers Unity Application Generator se définissent comme suit :
- Le premier champ de chaque ligne contient un mot clé spécifiant la nature de l'objet contenu dans la ligne.
- Les champs suivants de la ligne contiennent les attributs de chaque objet individuel. L'ordre des valeurs est fixe pour chaque mot clé.
- Le séparateur est au choix de l'utilisateur.

Le format des fichiers CSV est le même en importation et exportation.

Pour obtenir plus d'informations, consultez la sectionFormat des fichiers CSV d'importation et d'exportation *(voir page 445)*.

Fichier XML

UAG prend en charge un format XML hiérarchique.

Pour obtenir des détails sur le format et la structure des fichiers XML, veuillez consulter la section Format des fichiers XML d'importation et d'exportation *(voir page 483)*.

Exportation d'un projet UAG

Présentation

UAG prend en charge l'exportation de données dans 2 formats :
- format CSV (compatibilité ascendante avec la version 3.0 et ultérieure de UAG)
- format XML

Limitations d'utilisation

Lorsqu'un fichier CSV est importé, étant donné que le fichier CSV sera converti en XML pour le traitement, les numéros de ligne du message affichés dans la fenêtre de résultat correspondent au numéro de ligne du fichier XML.

Exportation de données

Vous pouvez exporter des données à partir :
- d'une arborescence du **modèle topologique**
- d'une arborescence du **modèle physique**
- des **bibliothèques de modèles** de l'arborescence des **bibliothèques** (exportation XML uniquement)

Pour exporter des données, cliquez avec le bouton droit de la souris sur les arborescences susmentionnées puis sélectionnez **Exporter** dans le menu contextuel.

La boîte de dialogue **Sélectionner un fichier d'exportation** s'affiche avec **Fichiers d'exportation (*.xml)** comme valeur par défaut pour **Enregistrer sous le type**.

Pour **Enregistrer sous le type**, vous avez le choix entre :
- **Fichiers d'exportation (*.xml)**
- **Fichiers d'exportation (*.txt)**
- **Tous les fichiers (*.*)**

Exportation de parties/la totalité du projet

Si vous ne souhaitez exporter que des parties d'un projet, sélectionnez l'objet concerné et utilisez **Exporter** dans le menu contextuel. L'objet sélectionné est exporté, y compris les objets qu'il contient.

Les bibliothèques de modèles peuvent être uniquement exportées en utilisant le menu contextuel.

NOTE : Si vous devez exporter la totalité du projet, utilisez **Fichier** →**Exporter Projet**.

Format XML avec une hiérarchie

Si vous sélectionnez **Fichiers d'exportation (*.xml)**, le fichier XML exporté inclut la hiérarchie du parent.

Par exemple, si vous exportez un module de commande (**CM_01**), qui est un sous-élément de **Zone_01** →**Cellule procédé_01** →**Unité_01** →**Equipement_01**, le fichier XML contiendra des données jusqu'au niveau de zone en complément des données du module de commande exporté.

Historique des changements

Chaque opération d'exportation est consignée dans l'**historique des changements**.

Colonne de propriété	Description
Qui	initiateur de l'importation
Quand	date et heure de l'importation
Type d'objet	type de l'objet importé, par exemple `Projet`
Nom de l'objet	reste vide
Changements	`Importation commencée` ou `Importation terminée`
Attribut	reste vide

Importer dans un projet UAG

Présentation

UAG prend en charge l'importation de données dans 2 formats :
- format CSV (compatibilité ascendante avec UAG 3.0 et UAG 3.1)
- format XML

Importation de données

Vous pouvez importer des données vers :
- l'arborescence d'un **modèle topologique**
- l'arborescence d'un **modèle physique**
- les **bibliothèques de modèles** de l'arborescence des **bibliothèques** (importation XML uniquement).
  Vous pouvez uniquement importer une bibliothèque de modèles dans sa totalité.

Pour importer des données, utilisez **Fichier** →**Import du projet**. Le projet doit être ouvert sans partage.

La boîte de dialogue **Sélectionner un fichier d'importation** s'affiche avec **Fichiers d'importation (*.xml)** comme valeur par défaut pour **Type de fichiers**.

Pour **Type de fichiers**, vous avez le choix entre :
- **Fichiers d'importation (*.xml)**
- **Fichiers d'importation (*.txt)**
- **Tous les fichiers (*.*)**

Historique des changements

Chaque opération d'importation est consignée dans l'**historique des changements**.

Mode d'importation/exportation de fichiers CSV/XML

NOTE : L'importation/exportation entre différentes versions de UAG n'est pas prise en charge.

Procédure d'importation

Procédez comme suit pour importer un fichier CSV/XML :

Etape	Action
1	Préparez un fichier CSV/XML correspondant aux formats décrits dans les sections Format des fichiers CSV d'importation et d'exportation *(voir page 445)* et Format des fichiers XML d'importation et d'exportation *(voir page 483)*. **Remarque** : les données d'importation doivent répondre aux conventions de nommage de la personnalisation de votre projet cible.
2	Ouvrez un projet cible Unity Application Generator.
3	Réglez le séparateur dans Unity Application Generator pour qu'il corresponde au séparateur de votre fichier CSV (**Vue** →**Options**) (fichiers CSV uniquement).
4	Dans les options de l'analyseur (**Vue** →**Options** →**Analyse projet**), précisez si les alertes doivent être affichées et s'il faut arrêter l'importation après la détection d'un certain nombre de messages ou d'alertes de non-conformité.
5	Lancez l'importation à l'aide de **Fichier** →**Import du projet**.

Résultat

En cours de l'importation, la fenêtre des messages sert à afficher l'activité d'importation en cours.

La syntaxe et la sémantique du fichier sont analysées. En cas de non-conformité logique et syntaxique, un message est écrit dans la fenêtre des messages.

Unity Application Generator ignore les lignes de programmation erronées et continue de fonctionner.

Si une ligne de programmation est correcte, elle est interprétée et les différents objets sont créés, si nécessaire.

La fonction **Importer** utilise les réglages des options de l'analyseur pour savoir s'il faut afficher tous les alertes ou arrêter après détection d'un certain nombre de messages ou d'alertes de non-conformité.

Les objets existants peuvent être modifiés par une importation.

- Les données, à l'exception des noms d'objets, modifiables au moyen de l'interface utilisateur UAG, peuvent être modifiés également par l'importation.
- Si un objet peut être sélectionné à l'aide d'une zone de liste à partir de l'interface utilisateur UAG (API, zone de conduite, image), s'il s'agit d'un attribut obligatoire dans le format de fichier d'exportation et si la valeur est absente du fichier exporté, la valeur sera considérée comme `non affectée` lors de la création ou de la mise à jour de l'objet.
  Si la valeur est non valide pour ce type d'objet, cet objet ne sera pas créé ou mis à jour par une importation et un message sera affiché.

Procédure d'exportation

Pour exporter des parties du modèle physique ou du modèle topologique en un fichier CSV ou XML, procédez comme suit :

Etape	Action
1	Réglez le séparateur dans Unity Application Generator au choix, pour le fichier CSV cible (**Vue** →**Options**). (fichiers CSV uniquement).
2	Ouvrir le menu contextuel de l'objet à exporter par un clic droit dans l'arborescence du modèle et sélectionner **Exporter**.
3	Suivre les instructions données dans la boîte de dialogue.

Résultat

Unity Application Generator génère un fichier CSV/XML contenant l'objet sélectionné et les objets qu'il contient. Les données du fichier CSV/XML correspondent au format décrit dans les sections Format des fichiers CSV d'importation et d'exportation *(voir page 445)* ou Format des fichiers XML d'importation et d'exportation *(voir page 483)*.

NOTE : Si vous devez exporter l'intégralité du projet (modèle physique et topologique), sélectionnez **Fichier** →**Exporter projet**.

Importation incrémentielle

Présentation

UAG prend en charge l'importation incrémentielle.

En général, vous pouvez modifier des données qui peuvent être également changées à partir de l'interface de l'utilisateur à l'aide de la fonction d'importation.

NOTE : L'importation incrémentielle des configurations Advantys et Profibus n'est pas possible avec UAG 3.3 SP4.

Règles

Les règles suivantes s'appliquent à l'importation incrémentielle :

- Les données qui peuvent être modifiées à partir de l'interface utilisateur peuvent être également changées à l'aide de la fonction d'importation.
  Par exemple, l'attribut `Type de réseau` de `Segment de réseau` ne peut pas être modifié.
- Si aucun des champs n'est modifié pour un élément spécifique, le format CSV affiche un message alors que le format XML ignore l'objet.
- Si l'attribut `Nom` est modifié, l'importation au format XML le traite comme un nouvel objet.
- Si des attributs autres que `Nom` sont modifiés, l'importation modifie l'objet.

Changement de nom et suppression (format CSV)

Au format CSV, le changement de nom et la suppression d'objet sont possibles en utilisant les mots-clés `DELETE` et `RENAME`.

Veuillez consulter la section Suppression et changement de nom : format de fichier CSV *(voir page 479)*.

Changement de nom et suppression (format XML)

Au format XML, le changement de nom et la suppression d'objet sont possibles en utilisant l'attribut `importAction` associé à

- `rename` et `newName` ou avec
- `delete`.

Veuillez consulter la section Règles de suppression et de changement de nom d'objets *(voir page 501)*.

Sous-chapitre 12.2
Exemples d’importation/Exportation CSV

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Exemple : Importation d’instruments / de la hiérarchie du modèle physique	408
Exemple : importer des valeurs initiales de variables	411

Exemple : Importation d'instruments / de la hiérarchie du modèle physique

Introduction

La fonctionnalité d'importation fournit une interface ouverte permettant de renseigner les bases de données de projet de Unity Application Generator depuis une source externe d'une manière simple mais très efficace. Il est possible à l'aide d'un fichier CSV (comma separated values) de créer des instruments, voire l'ensemble de la hiérarchie du modèle physique.

Reportez-vous à : *Instruments, page 81* et *Instruments : format de fichier CSV, page 469*.

Importation depuis le schéma de tuyauterie et d'instrumentation (PID)

La fonction d'importation est particulièrement utile pour les clients commençant la conception par le schéma de tuyauterie et d'instrumentation, lequel est créé à l'aide d'un autre outil tel qu'AutoCAD avec les extensions Rebis. Si vous avez déjà défini tous les instruments (modules de commande) à l'aide de ce genre d'outils, vous pouvez vous dispenser d'une nouvelle saisie des mêmes informations dans le projet Unity Application Generator en vous servant de la fonction d'importation.

Ligne du fichier CSV

Pour importer des instruments ou la hiérarchie du modèle physique, chaque ligne du fichier CSV possède les entrées suivantes :

`INSTRUMENT`;***controlmodule_name***;***controlmodule_type***;***library_name***;***controlmodule_description***;***controlmodule_comment***;***full_path***;

NOTE : Ce format simplifié n'est disponible qu'à l'importation. Le format d'exportation des objets de la hiérarchie du modèle physique fait l'objet d'une description dans *Les éléments du modèle physique : format de fichier CSV, page 449*.

Séparateur

Le séparateur par défaut est un point-virgule ; cela peut être modifié. Le caractère séparateur est défini dans la boîte de dialogue **Vue** →**Options** dans Unity Application Generator.

NOTE : N'utilisez pas de caractère séparateur pouvant être utilisé ailleurs dans votre projet.

Options

Vous avez la possibilité d'importer des instruments sans la hiérarchie du modèle physique ou avec l'ensemble de la hiérarchie jusqu'au niveau zone :

Si vous voulez...	Alors...
N'importer que des instruments (sans la hiérarchie du modèle physique)	Omettre l'entrée ***full_path***
Importer des instruments en tant que modules de commande avec la totalité de la hiérarchie du modèle physique.	Spécifier l'entrée ***full_path***

Description des entrées

Les entrées d'une ligne du fichier CSV d'importation d'instruments ont la signification ci-dessous. Les champs obligatoires sont repérés par une étoile, *. Si les champs non obligatoires ne sont pas définis, ceux-ci restent vides (;;).

Entrée	Description et vérification lors de l'importation	Exemple
* `INSTRUMENT`	La première entrée doit exactement correspondre à la chaîne de caractères `INSTRUMENT` pour identifier ce que contient la ligne.	
* ***controlmodule_name***	Cette entrée contient le nom du module de commande à créer. Un contrôle est effectué sur le nom pour vérifier sa conformité avec la convention de nommage utilisée dans la personnalisation de projet. Si le module de commande doit être créé dans un équipement, l'unicité du nom du module de commande est vérifiée dans le cadre de l'équipement parent.	`moteur1`
* ***controlmodule_type***	Cet élément contient le nom du type de module de commande du module de commande à créer. Le nom est vérifié pour s'assurer qu'il correspond exactement à celui de l'un des types de modules de commande de la bibliothèque des types de modules de commande spécifiée dans ***controlmodule_type_library***.	MSR01
* ***library_name***	Cette entrée contient le nom de la bibliothèque de types de modules de commande, laquelle contient le type de module spécifié. Il est vérifié si la personnalisation de projet contient la bibliothèque.	`IATBasic10`

Entrée	Description et vérification lors de l'importation	Exemple
* ***controlmodule_description***	Si le texte est plus long que la longueur de description admissible, le texte est tronqué à la longueur maximum.	`Moteur du convoyeur n° 1`
controlmodule_comment	Cette entrée contient le commentaire du module de commande à créer. Si le texte est plus long que la longueur de commentaire admissible, le texte est tronqué à la longueur maximum. **Remarque** : la séquence de caractères spéciale `\n` peut être utilisée pour inclure des retours à la ligne dans le commentaire. La séquence de caractères spéciale \t peut être utilisée pour inclure des tabulations dans le commentaire.	`Moteur du convoyeur n° 1\nLe moteur peut fonctionner en mode manuel ou auto`
full_path	Cette entrée définit la hiérarchie du modèle physique dans lequel le module de commande doit être créé. **Si elle n'est pas renseignée**, le module de commande est créé dans la liste d'instruments et **non** au sein d'un élément physique du modèle physique. **Si elle est renseignée**, ● les objets seront créés s'ils n'existent pas encore, et ● le module de commande sera créé dans la hiérarchie du modèle physique	`\Niveau1\Niveau2\Niveau3\Niveau4`

Erreurs détectées

Lors de la procédure d'importation, Unity Application Generator vérifie si les entrées ont un nom unique et si elles sont conformes à la personnalisation. Si une erreur est détectée lors de ces vérifications, le traitement de la ligne est abandonné et le reste de la ligne est ignoré. Un message d'erreur détectée s'affiche dans la fenêtre des messages.

Exemple : importer des valeurs initiales de variables

Introduction

L'importation d'un fichier CSV permet de modifier les valeurs initiales de variables existantes. Les paramètres PID doivent souvent être affinés lors de la phase de mise en service afin d'optimiser les boucles. Pour redémarrer le procédé avec les paramètres de boucle optimisés, ces valeurs doivent être mémorisées en tant que valeurs initiales des variables adéquates.

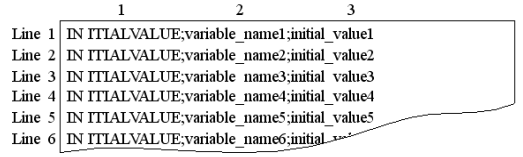
Ligne du fichier CSV

Pour importer les valeurs initiales des variables, chaque ligne du fichier CSV possède les entrées suivantes :

INITIALVALUE;***variable_full_name*** ;***variable_parent_type*** ;***variable_path*** ;***initial_value***

Exemple : valeurs initiales d'un fichier CSV

Voici un exemple de fichier CSV d'importation de valeurs initiales de variables :



NOTE : Le séparateur par défaut est un point-virgule ; cela peut être modifié. Le caractère séparateur est défini dans la boîte de dialogue **Vue** →**Options** dans Unity Application Generator pour permettre une bonne lecture du fichier CSV.

Description des entrées

Les entrées d'une ligne du fichier CSV d'importation de valeurs initiales ont la signification suivante :

Entrée	Description et vérification lors de l'importation	Exemple
`INITIALVALUE`	La première entrée doit exactement correspondre à la chaîne de caractères `INITIALVALUE` pour identifier ce que contient la ligne.	-
variable_full_name	Cette entrée contient le nom complet de la variable dont la valeur initiale doit être modifiée. Le nom est vérifié pour s'assurer qu'il s'agit bien du nom d'une variable existante.	`Boiler1_Motor1_FTR`
variable_parent_type	`CONTROLMODULE` et `PHYSICALELEMENT`	`CONTROLMODULE`
variable_path	Cette entrée contient le chemin d'accès à l'élément physique fonctionnel (ou au module de commande) de la variable dont la valeur initiale doit être modifiée.	`\Zone1\Cellule1\Unité1\Chaudière1`
initial_value	Cette entrée contient la valeur initiale de la variable. Un contrôle est effectué sur la validité de la valeur initiale conformément au type de donnée de la variable.	`3.1415`

Erreurs détectées

Lors du processus d'importation, Unity Application Generator contrôle les entrées comme décrit dans le tableau ci-dessus. Si une erreur est détectée lors de ces vérifications, le traitement de la ligne est abandonné et le reste de la ligne est ignoré. Un message d'erreur détectée s'affiche dans la fenêtre des messages.

Sous-chapitre 12.3
Importer/exporter XML

Exportation depuis l’éditeur SCoD

Présentation

Veuillez consulter *Exportation depuis l’éditeur SCoD* dans le *Manuel utilisateur de l’éditeur SCoD*.

Chapitre 13
Types de données dérivés

Introduction

Ce chapitre explique l'utilisation possible des types de données dérivés, DDT.

NOTE : Les types de données dérivés (DDT) pour les communications API-IHM ne peuvent pas être utilisés avec iFIX.

Pour Monitor Pro, vous devez utiliser le serveur OPC pour transférer les données.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
13.1	Notions fondamentales sur les types de données dérivés	416
13.2	DDT et l'éditeur SCoD	417
13.3	DDT et l'éditeur de personnalisation	430
13.4	DDT et l'éditeur UAG	433
13.5	Les DDT et les Interfaces Homme/Machine (IHM)	438

Sous-chapitre 13.1
Notions fondamentales sur les types de données dérivés

Gestion des types de données dérivés (DDT)

Introduction

UAG gère les types de données structurés et les tableaux. On les appelle types de données dérivés (DDT).

Vous pouvez importer la définition du DDT dans l'éditeur SCoD à partir de Unity Pro. Après l'importation, vous pouvez créer des variables avec ces types de données. Vous pouvez affecter à la broche d'un SCoD une variable ayant un type de donnée dérivé. Il est également possible d'affecter un élément de la variable DDT à la broche.

Dans votre projet UAG, vous pouvez créer des variables libres en utilisant des types de données structurés et des tableaux. Toutes les variables de type **DDT** peuvent être transmises entre API et entre API et serveurs de données, en fonction du type de connexion défini.

La cartographie mémoire réserve l'espace mémoire nécessaire aux API et aux serveurs de données. UAG génère les instances de variable dans le projet API et dans l'application IHM.

NOTE : Les types de données dérivés (DDT) pour les communications API-IHM ne peuvent pas être utilisés avec iFIX.

Pour Monitor Pro, vous devez utiliser le serveur OPC pour transférer les données.

Sous-chapitre 13.2
DDT et l’éditeur SCoD

Introduction

Cette partie vous explique comment utiliser les types de données dérivés (DDT) dans l’éditeur SCoD.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Importation de DDT dans l’éditeur SCoD	418
Utilisation des DDT dans l’éditeur SCoD	420
Définition de l’héritage avec des variables DDT	424
Connexion de variables DDT à des broches	426
Exemple d’importation de DDT pour l’éditeur SCoD	427
Règles d’affectation en mémoire	429

Importation de DDT dans l'éditeur SCoD

Introduction

L'éditeur SCoD permet d'importer des définitions de DDT à partir de fichiers d'échange XML (*.x*) de Unity Pro.

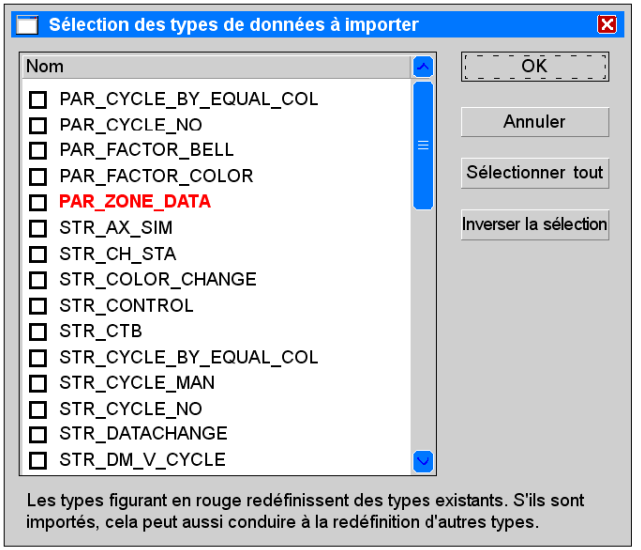
Importation de DDT

Vous pouvez importer automatiquement les DDT avec l'importation des FFB, si les définitions des DDT sont contenues dans le même fichier que celles des FFB. Mais il est aussi possible d'importer les DDT indépendamment des FFB. L'éditeur SCoD ne vérifie pas le nombre de niveaux dans les DTT imbriqués.

Pour un DDT, le nom, le commentaire et tous les éléments enfants accompagnés de leur nom, de leur type et de leur commentaire seront importés.

Lorsque le fichier a été lu, une boîte de dialogue de sélection s'affiche pour vous inviter à sélectionner les noms des DDT à importer. Vous pouvez alors affecter les définitions de DDT (incluant les éléments enfants) à importer. Lorsqu'un DDT n'est pas affecté, seul son nom DDT est importé et le DDT est créé en tant que DDT non défini.

Le schéma suivant illustre la boîte de dialogue **Sélection des types de données à importer**.



DDT existants

Si le nom d'un DDT existe déjà dans l'éditeur SCoD, vous avez la possibilité d'écraser la définition existante ou de la conserver si ce type n'est pas utilisé dans la bibliothèque. Sinon, la nouvelle définition de DDT ne sera pas importée. Si cela survient lors de l'importation d'un FFB, le FFB ne sera pas importé non plus.

Avant la génération pour l'API, UAG vérifie si les définitions de DDT du projet UAG et celles du projet Unity correspondent bien. Unity Pro étant la source de la définition des DDT, UAG ne génère pas la définition des DDT dans le système API. Le DDT doit être disponible dans le projet Unity Pro (global ou local).

Types de données de Unity Pro

Vous pouvez utiliser les types de données dérivés de Unity Pro. Pour utiliser les types de données, vous devez importer un fichier inclus dans le DVD de UAG. Vous trouverez le fichier sur votre disque dur après l'installation de UAG (*<UAG Install Dir>\DB\UnityProLibset.dty*).

Vous pouvez importer ce fichier avec **Fichier →Importer types de données**. Sélectionnez les types de données nécessaires dans la boîte de dialogue qui s'affiche.

Voici des exemples de types de données : **BoolArr32**, **INTARR8**, **WordArr5**.

Utilisation des DDT dans l'éditeur SCoD

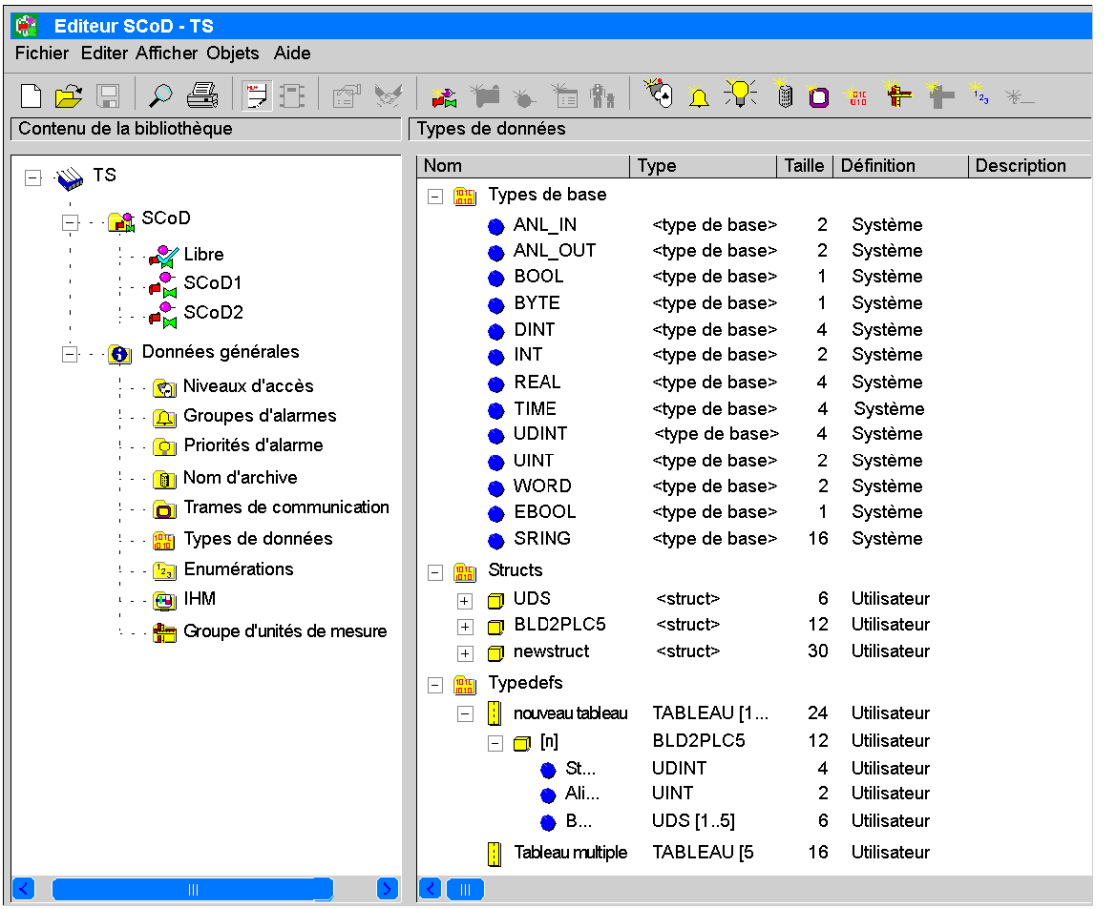
Introduction

Vous pouvez utiliser les types de données dérivés après les avoir importés depuis Unity Pro.

Affichage des DDT

Vous pouvez trouver des DDT et leurs éléments sous **Données générales** →**Types de données** dans l'arborescence de l'éditeur SCoD.

Le schéma suivant présente un exemple de la vue **Types de données** :



Utilisation des DDT

Les DDT peuvent servir à créer des variables et des broches. Dans la liste des variables d'un SCoD, vous pouvez ouvrir la boîte de dialogue des propriétés d'une variable DDT ou de ses éléments enfants.

Vous pouvez créer des variables DDT avec les types de connexion suivants :

- **API**
- **API_IHM**
  Ne concerne pas iFIX.
  Pour tous les autres IHM, uniquement si les DDT sont entièrement définis.
- **IHM**
  Ne concerne pas iFIX.

Les éléments enfants suivants des DDT sont en lecture seule et ont les mêmes valeurs que la variable elle-même (Nom, Type de connexion et Type d'E/S) ou ont les valeurs définies dans le DDT (Description, Type de données et Ordre).

- **Nom de propriété**
- **Description**
- **Type de connexion**
- **Type de données**
- **Type d'E/S**
- **Ordre**

Tous les éléments enfants d'une variable DDT, dotés d'un type de donnée élémentaire, possèdent toutes les propriétés associées comme dans le cas d'une variable élémentaire normale (par ex. **Valeur initiale**, **Alarme**, **Archive** et autres). Il est ainsi possible d'utiliser ces éléments enfants et leurs propriétés dans la page des propriétés d'un SCoD. Les propriétés des éléments peuvent également être utilisées comme propriété source ou cible dans les relations d'héritage. De plus, il est possible de lier un élément d'une variable DDT à une broche ayant le même type de donnée.

Vue des éléments et des propriétés d'une variable

Vous pouvez modifier la vue de la variable dans la boîte de dialogue **Propriétés de variable**. Vous pouvez basculer entre

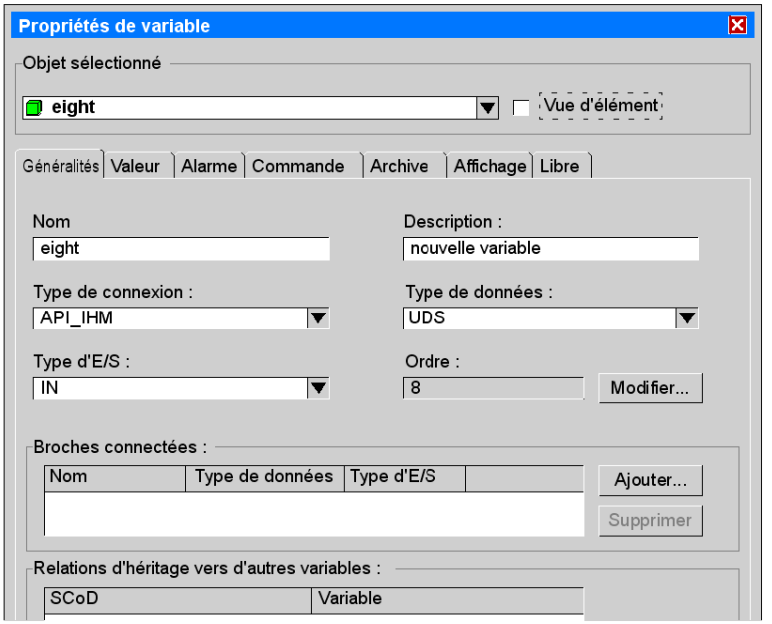
- la vue des variables (de niveau supérieur)

et

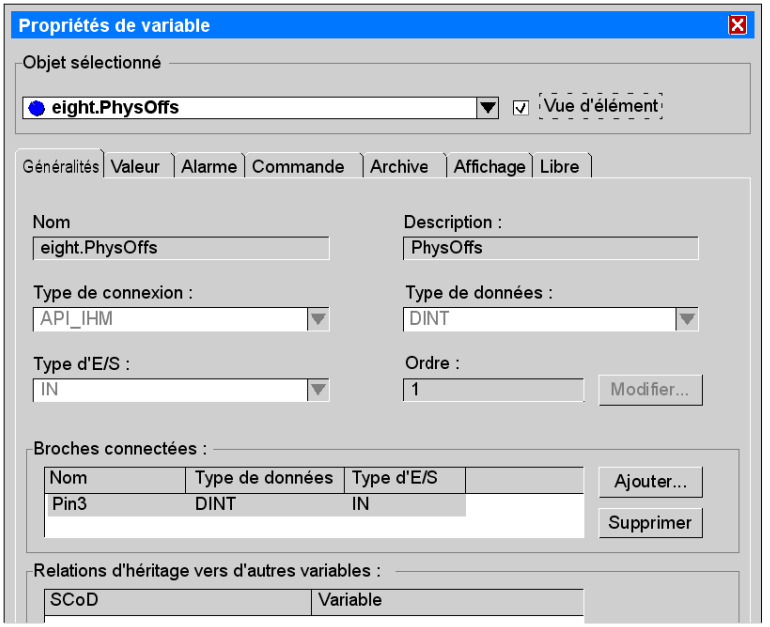
- celle des éléments de la variable DDT sélectionnée.

Vous pouvez utiliser la case à cocher **Vue d'élément** pour afficher différemment les variables.

Le schéma suivant illustre l'affichage des **Propriétés de variable** sans la **Vue d'élément**.



Le schéma suivant illustre l'affichage des **Propriétés de variable** avec la **Vue d'élément**.



Si vous activez la vue d'élément, tous les éléments de la variable actuellement sélectionnée, qui ont un type de donnée système (**BOOL**, **INT**, etc.) vont s'afficher dans la zone de liste modifiable, dans l'ordre défini par le type de donnée, le premier élément étant sélectionné. Si vous désactivez l'option, l'affichage revient aux variables de niveau supérieur, et présente la variable actuellement sélectionnée.

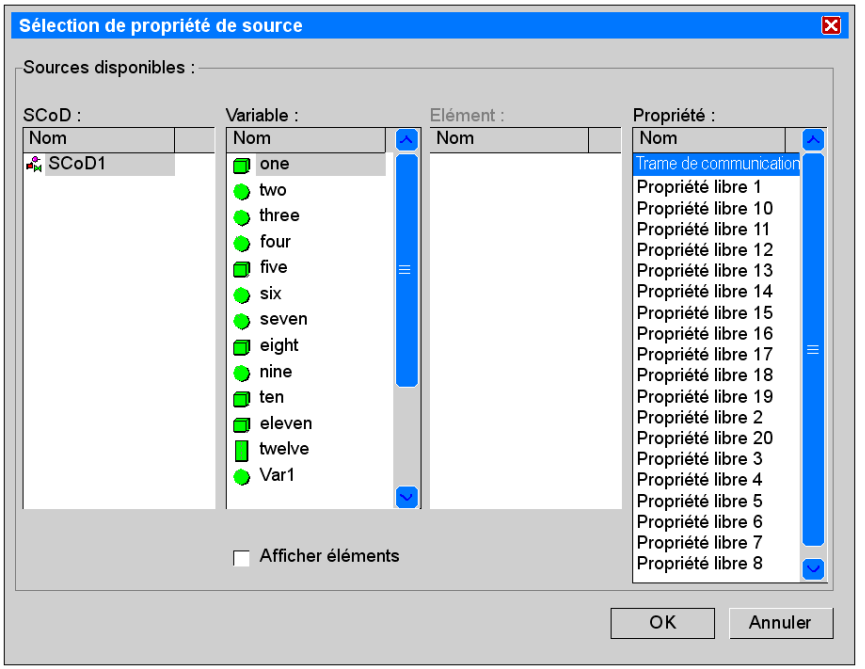
Définition de l'héritage avec des variables DDT

Introduction

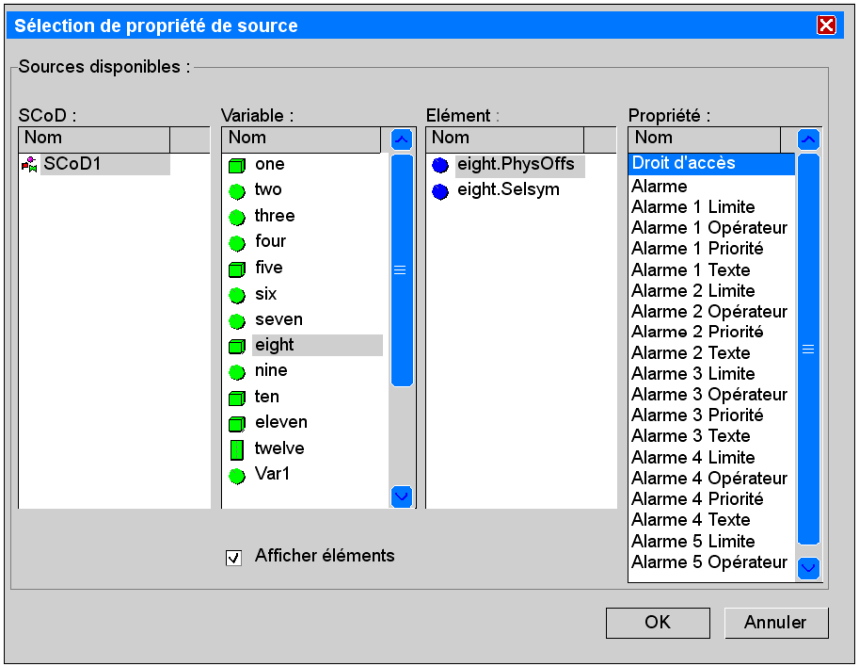
Vous avez la possibilité d'hériter des propriétés de DDT. Pour l'héritage, utilisez la boîte de dialogue **Sélection de propriété de source**. Cette boîte de dialogue contient la zone de liste **Propriété**. Les éléments du DDT figureront dans cette zone de liste. Les éléments apparaîtront si l'option **Afficher éléments** est cochée.

NOTE : Seules la **Trame de communication** et les **Propriétés libres** peuvent être héritées d'une variable DDT.

La figure suivante présente la boîte de dialogue **Héritage →Propriété de variable**. La case à cocher **Afficher éléments** n'est pas sélectionnée.



La figure suivante présente la boîte de dialogue **Héritage** →**Propriété de variable**. La case à cocher **Afficher éléments** est sélectionnée.

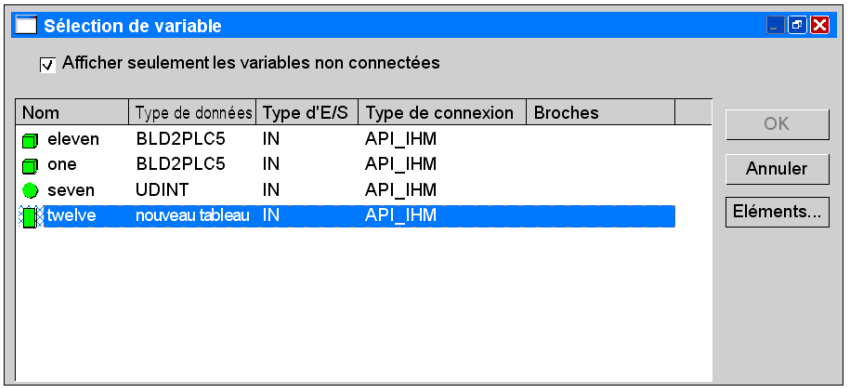


Connexion de variables DDT à des broches

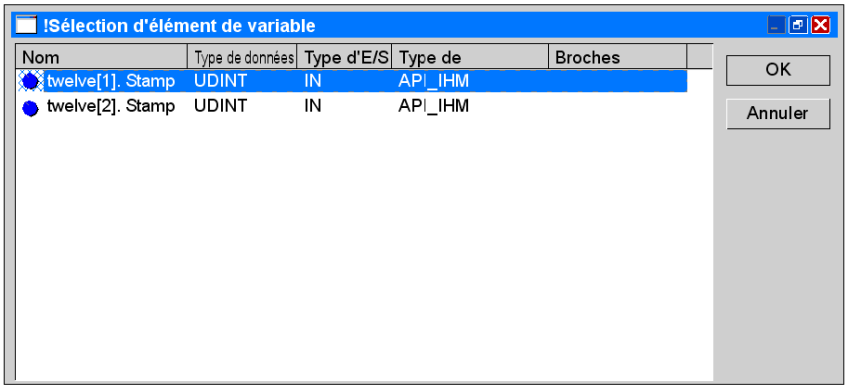
Introduction

Si une broche connectée a DDT comme type de donnée, la boîte de dialogue **Sélection de variable** affiche la liste de toutes les variables DDT. Le bouton **OK** est désactivé jusqu'à ce qu'une variable ou un élément du type de donnée correct soit sélectionné. Le bouton **Eléments...** ouvre une boîte de dialogue permettant de sélectionner un des éléments de la variable actuellement sélectionnée.

La figure suivante illustre la boîte de dialogue **Sélection de variable**.



La figure suivante illustre la boîte de dialogue **Sélection de variable** →**Eléments...**.



Exemple d'importation de DDT pour l'éditeur SCoD

Introduction

Exécutez les étapes suivantes pour importer une structure de données dans l'éditeur SCoD. Une fois l'importation réalisée, vous avez la possibilité d'utiliser la structure.

Vous pouvez importer des types de données dérivés (DDT) à partir de fichiers aux formats suivants :

- avec Unity Pro
  - *XD**
  - *XML*

Possibilités d'importation

Vous avez la possibilité d'importer les DDT de 2 façons différentes :

- Importation du type de donnée dérivé à partir du fichier d'export DDT de Unity Pro
- Importation du type de donnée dérivé à partir du fichier d'export DFB de Unity Pro

NOTE : Si vous supprimez le DFB de votre fichier de bibliothèque SCoD (*.*OSL*), le DDT n'est pas automatiquement supprimé mais les données sont transférées uniquement si elles sont utilisées.

Importation d'un DDT

Les étapes suivantes décrivent l'importation de DDT dans une bibliothèque SCoD Monitor Pro.

Etape	Action
1	Générez et exportez le type de donnée dérivé (DDT) du projet Unity Pro. Enregistrez le fichier d'export dans le répertoire de projet.
2	Ouvrez l'éditeur SCoD.
3	Ouvrez la bibliothèque SCoD pour Monitor Pro (*IATBasic10Monitor.osl*).
4	Enregistrez la nouvelle bibliothèque SCoD avec le nom utilisateur choisi.
5	Importez le fichier d'export DFB ou DDT dans la bibliothèque SCoD. Pour cela, sélectionnez **Fichier →Importer types de données...** ou **Fichier →Importer DFB...**
6	Sélectionnez le fichier dans le répertoire de projet.
7	Une boîte de dialogue de sélection des types de données dérivés (DDT) ou des blocs fonctions dérivés (DFB) s'affiche. Sélectionnez l'élément à importer. Appuyez sur OK après la sélection. **Remarque :** les types de données peuvent être importés en tant qu'éléments non définis. Dans ce cas, seul le nom du DDT sera importé.
8	Après une importation réussie des variables ou des blocs fonctions, les DDT seront disponibles.

Longueur des noms des variables DDT

L'importation des types de données et des blocs fonctions dérivés fait l'objet de certaines restrictions sur la longueur des noms de variables :

- Monitor Pro
  Le nom des variables ne peut pas dépasser 32 caractères. Si les variables ont plus de 32 caractères, elles ne pourront pas être transférées dans Monitor Pro.
- Vijeo Citect avec OFS
  Le nom des variables ne peut pas dépasser 55 caractères.
- Vijeo Citect avec pilote standard
  Le nom des variables ne peut pas dépasser 79 caractères.

DDT et Monitor Pro / IHM générique

NOTE : Les types de données dérivés (DDT) pour les communications API-IHM peuvent être uniquement utilisés avec Monitor Pro et une IHM générique.

Pour Monitor Pro, vous devez utiliser le serveur OPC pour transférer les données.

Consultez également la section *Les DDT et les Interfaces Homme/Machine (IHM), page 438*

Règles d’affectation en mémoire

Règles d’affectation en mémoire

Les règles d’affectation en mémoire dans Unity Pro sont les suivantes :

Si un tableau ou une structure contient uniquement des éléments de type **BOOL** ou **BYTE**, il n’y a pas d’alignement particulier. Sinon, il pourrait y avoir un octet de remplissage entre un élément **BOOL** ou **BYTE** et un élément d’un autre type quelconque car les autres types doivent commencer à une adresse d’octet paire (de même que les DDT qui ont des éléments autres que **BOOL** ou **BYTE**).

Consommation de mémoire pour les EDT dans Unity Pro :

- 8 bits :
  BOOL, **EBOOL**, **BYTE**
- 16 bits :
  INT, **UINT**, **WORD**
- 32 bits :
  DINT, **UDINT**, **TIME**, **DATE**, **TOD** (TIME_OF_DAY), **REAL**, **DWORD**
- 64 bits :
  DT (DATE_AND_TIME)
- 128 bits :
  STRING
- 8 bits*n :
  STRING[n]

Sous-chapitre 13.3
DDT et l’éditeur de personnalisation

Introduction

Cette partie vous explique comment utiliser les types de données dérivés (DDT) dans l’éditeur de personnalisation.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Importation de DDT dans l’éditeur de personnalisation	431
Utilisation des DDT dans l’éditeur de personnalisation	432

Importation de DDT dans l'éditeur de personnalisation

Introduction

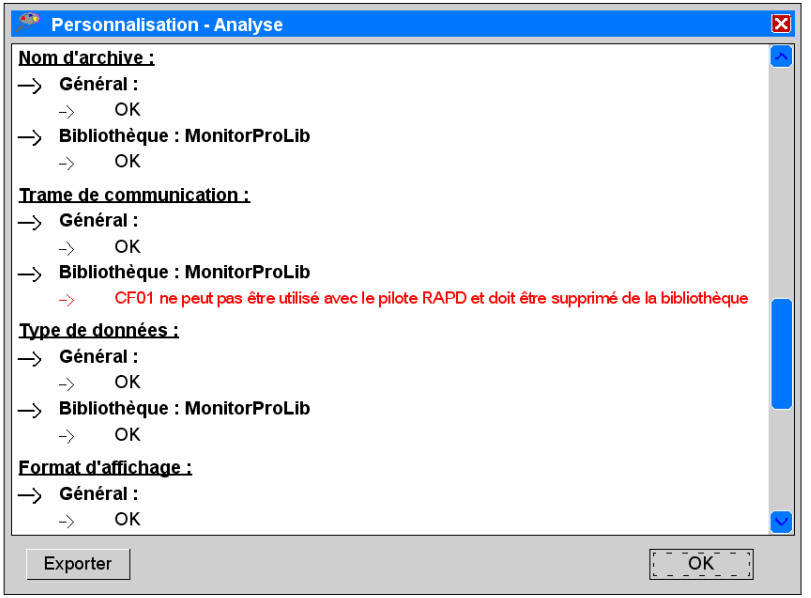
L'éditeur de personnalisation importe les noms de DDT depuis les bibliothèques SCoD sélectionnées si les DDT sont utilisés dans un SCoD validé. Dans ce cas, seuls les noms de DDT seront importés. Si un nom de DDT existe déjà dans l'éditeur de personnalisation, les définitions de DDT dans les bibliothèques seront comparées. S'il y a une différence, l'utilisateur sera invité à mettre à jour manuellement la bibliothèque SCoD pour assurer sa cohérence. Tant qu'il existe des définitions de DDT différentes, la personnalisation n'est pas valable et il est impossible d'ouvrir le projet UAG correspondant.

Analyse de projets avec les DDT

Vous pouvez utiliser les structures de données au sein des variables pour votre communication API-IHM après avoir effectué l'importation dans l'éditeur SCoD. Pour la communication des variables, vous devez utiliser une trame de communication dont le type de donnée est **Any**.

Si vous utilisez une bibliothèque SCoD qui contient ce genre de variables, un message d'erreur détectée est affiché dans l'éditeur de personnalisation après l'analyse du projet.

La figure suivante illustre le message qui apparaît dans la fenêtre Personnalisation – Analyse :



NOTE : Ne supprimez pas la trame de communication ni la variable de la bibliothèque. Changez le pilote Monitor Pro sous **Général** →**Monitor Pro** dans l'arborescence de gauche. Remplacez le pilote par **OPC (OFS, gérant à la fois Modbus Plus et TCP/IP)**.

Utilisation des DDT dans l'éditeur de personnalisation

Introduction

Le nom des DDT est indiqué dans la liste des types de données. La liste comporte une colonne qui indique les noms des bibliothèques qui contiennent la définition des DDT.

Vous avez la possibilité de sélectionner les types de données qui peuvent être utilisés pour créer des variables libres dans le projet UAG. Vous pouvez annuler la sélection d'un type de donnée uniquement si le type en question n'est pas utilisé dans le projet UAG ni pour un élément enfant d'un DDT utilisé dans le projet.

Il est impossible d'annuler la sélection d'une bibliothèque SCoD dans l'éditeur de personnalisation si une définition de DDT issue de cette bibliothèque est utilisée dans le projet UAG et s'il n'y a pas d'autre bibliothèque ayant la même définition. Si une bibliothèque SCoD doit être supprimée de la personnalisation, l'éditeur de personnalisation vérifiera qu'il n'existe pas de définition DDT provenant exclusivement de cette bibliothèque. Vous serez averti que le DDT sera également supprimé si vous poursuivez la suppression.

Si un DDT a été supprimé dans l'éditeur SCoD, l'éditeur de personnalisation supprimera également le DDT, s'il n'y a pas d'autre bibliothèque ayant la même définition de DDT. L'analyseur de personnalisation du projet UAG, qui démarre automatiquement lorsque le projet est en cours d'ouverture et que la personnalisation a été modifiée, supprimera le DDT dans le projet UAG. L'utilisateur de UAG recevra un message et pourra annuler l'ouverture du projet UAG.

Sous-chapitre 13.4
DDT et l’éditeur UAG

Introduction

Cette partie vous explique comment utiliser les types de données dérivés (DDT) dans l’éditeur UAG.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Utilisation des variables DDT dans l’éditeur UAG	434
Diffusion des DDT entre les API	436
Affectation des variables DDT à des points d’E/S	437

Utilisation des variables DDT dans l'éditeur UAG

Introduction

Vous pouvez utiliser les types de données dérivés dans votre projet UAG et vous pouvez créer des variables libres avec les DDT.

Lorsqu'un module de commande est créé, les variables de module de commande avec les DDT seront automatiquement créées par UAG en fonction de la définition présente dans la bibliothèque.

Les variables DDT apparaîtront avec leurs éléments enfants dans la grille des variables.

Les éléments enfants ayant des types de données élémentaires peuvent être utilisés comme des variables normales dans la définition de liaisons et dans l'éditeur graphique de forçages.

Grille des variables

Les variables DDT et leurs éléments enfants figureront avec les autres variables dans la grille des variables de l'afficheur de variables (voir *Afficheur de variables, page 300*).

Les éléments enfants, s'ils existent, peuvent être rendus visibles ou non pour chaque variable ou élément DDT.

Les noms d'éléments sont placés en retrait en fonction de leur niveau d'imbrication.

Exemple : `Variables [Eq1_SCoD1]`

Nom	Description	Connexion	Type de données	Entrée/Sortie	Adresse	Valeur initiale
▶ − one	nouvelle variable	API_IHM	BLD2PLC	Entrée		
Stamp	Stamp	API_IHM	UDINT	Entrée		
Alive	Alive	API_IHM	UNIT	Entrée		
BTM	BTM	API_IHM	UNIT	Entrée		
two	nouvelle variable	API_IHM	DINT	Entrée		
three	nouvelle variable	API_IHM	DINT	Entrée		
four	nouvelle variable	API_IHM	BOOL	Entrée		
+ five	nouvelle variable	API_IHM	struct2	Entrée		
+ six	nouvelle variable	API	Incognito	Entrée		
seven	nouvelle variable	API_IHM	UDINT	Entrée		
eight	nouvelle variable	API_IHM	UDINT	Entrée		
nine	nouvelle variable	API_IHM	BOOL	Entrée		
+ ten	nouvelle variable	API_IHM	newstruct	Entrée		
− eleven	nouvelle variable	API_IHM	BLD2PLC	Entrée		
Stamp	Stamp	API_IHM	UDINT	Entrée		
Alive	Alive	API_IHM	UINT	Entrée		
BTM	BTM	API_IHM	UNIT	Entrée		
twelve	nouvelle variable	API_IHM	UINT	Entrée		
Var1	nouvelle variable	API_IHM	BOOL	Entrée		
Var2	nouvelle variable	API_IHM	BOOL	Entrée		
Var3	nouvelle variable	API_IHM	BOOL	Entrée		

Une variable structurée, de niveau supérieur et imbriquée, est précédée d'un signe plus.

En cliquant sur le signe plus, vous faites apparaître tous les éléments enfants de cette structure tandis que les structures imbriquées ne sont pas automatiquement développées.

Les variables de structure développées sont précédées d'un signe moins, utilisé pour réduire la structure.

La colonne nom affiche le nom de l'élément.

Le nom complet des variables est affiché dans une infobulle lorsque vous déplacez la souris au-dessus d'une ligne contenant l'élément d'une structure ou dans la colonne **NomCompletVariable** si elle est configurée (voir la section *Personnaliser les colonnes, page 314*).

Diffusion des DDT entre les API

Introduction

Vous devez transmettre les variables entre les API utilisés, y compris celles de type de donnée dérivé. C'est pourquoi il est possible de faire glisser un ensemble de variables DDT ou des éléments enfants sélectionnés jusque vers la table de communication API-API. La taille de la variable ou de l'élément sera vérifiée en la comparant à la taille maximale d'un paquet de communication, admise pour le type de communication spécifique. L'opération glisser est uniquement autorisée si la taille est correcte.

Variables supplémentaires

Si un DDT contient des types de données élémentaires qui ne peuvent pas être mis en mémoire Bit ou Word de l'API (par ex. **STRING**, **TIME_OF_DAY**, etc.), UAG vous avertit que ces éléments ne sont pas ajoutés à la table de communication.

Affichage des variables

Pour une variable ou un élément DDT, tous les éléments enfants de type de donnée élémentaire sont affichés dans la table. La variable DDT elle-même ne figurera pas en tant qu'élément séparé de la table. Les éléments DDT apparaîtront optimisés pour la communication, par exemple les éléments sont classés par type de donnée et non pas en fonction de leur ordre dans le DDT. S'il reste de la place entre une zone Bit et une zone Word de la table, il est possible d'amener par une opération glisser d'autres variables ou éléments dans ces espaces.

Si vous souhaitez supprimer de la table un élément d'une variable DDT, il vous sera demandé si vous souhaitez également supprimer ou non tous les autres éléments appartenant au parent.

Comportement en cas de timeout

Pour chaque élément de la table, vous pouvez définir le comportement et la valeur par défaut en cas de timeout.

Section de communication Unity Pro

UAG crée des sections de communication dans le programme API pour compacter et décompacter les éléments en variables de DTT approprié. L'utilisateur doit vérifier que tous les types de données sont disponibles dans les outils de programmation d'API. Avant de procéder à la génération pour un API Unity, UAG vérifie que les DDT sont cohérents entre les projets API et UAG.

Affectation des variables DDT à des points d'E/S

Introduction

UAG n'autorise pas l'affectation des variables DDT ou de ses éléments enfants à un point d'E/S. Seuls les types de données **ANL_IN** et **ANL_OUT** sont autorisés.

NOTE : UAG ne gère pas les variables **IODDT**.

Sous-chapitre 13.5
Les DDT et les Interfaces Homme/Machine (IHM)

Introduction

Cette partie vous explique comment utiliser les types de données dérivés (DDT) dans les systèmes IHM.

NOTE : Les types de données dérivés (DDT) pour les communications API-IHM ne peuvent pas être utilisés avec iFIX.

Pour Monitor Pro, vous devez utiliser le serveur OPC pour transférer les données.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Transfert de variables DDT vers Vijeo Citect	439
Transfert de variables DDT vers Monitor Pro	440
Transfert de variables DDT vers une IHM générique	442

Transfert de variables DDT vers Vijeo Citect

Introduction

Pour avoir des informations sur le transfert de variables DDT vers Vijeo Citect, veuillez consulter *Types de variables complexes* dans le *Manuel utilisateur avancé UAG*, au chapitre *IHM pris en charge et leur configuration/Vijeo Citect et Unity Application Generator/Génération de variables*.

Transfert de variables DDT vers Monitor Pro

Introduction

Les informations données ici concernent le transfert de types de données dérivés vers Monitor Pro.

NOTE : Vous devez utiliser le serveur OPC pour communiquer les DDT à Monitor Pro.

Pilote OPC/Pilote RAPD

Dans l'**Editeur de personnalisation**, il existe une option permettant de choisir entre le pilote OPC et le pilote RAPD. La sélection du pilote RAPD permet de prendre en charge les projets existants.

NOTE : Si le projet a été généré auparavant et que l'option a changé, le projet de serveur de données doit faire l'objet d'une génération complètement nouvelle.

Variables DDT

Si la gestion OPC n'est pas activée et si des variables DDT sont utilisées dans le projet UAG, l'analyseur et le générateur serveur de données adressent une alerte à l'utilisateur pour lui indiquer que la variable ne peut pas être transférée.

Tâches ODX, ECI, MBTCP et IOX

Si la gestion OPC est activée, le générateur serveur de données générera la configuration pour les tâches **ODX** et **ECI** dans l'application serveur Monitor Pro ; sinon, la configuration pour les tâches **MBTCP** et **IOX** sera générée normalement.

NOTE : La tâche ECI de Monitor Pro ne fonctionne pas si une table d'information ECI pour une trame de communication des blocs d'écriture contient uniquement le tag déclencheur de bloc d'écriture mais pas de variable à communiquer. Le tag déclencheur de bloc d'écriture est activé si la fonction lecture/écriture est définie sur `WrTrigg`. Si OPC est sélectionné dans l'éditeur de personnalisation comme pilote pour Monitor Pro, assurez-vous que les trames de communication des blocs d'écriture sans variables sont supprimées de l'application serveur Monitor Pro. Pour cela, vérifiez tous les canaux du serveur de données dans UAG et vérifiez si une trame de bloc d'écriture contient un 0 dans le champ **Utilisé**, vérifiez que les champs **Adresse** et **Longueur** sont vides.

Tags pour une variable DDT

Une variable DDT de type **API_IHM** dans UAG résultera en plusieurs tags (ou blocs de données) dans Monitor Pro, un par élément enfant ayant le type de données élémentaires.

Si, dans le cas d'une variable tableau, les éléments de tableau sont du type de données élémentaires et que les dimensions du tableau commencent par un indice zéro, un tableau de tags sera généré dans Monitor Pro

NOTE : Monitor Pro ne gère pas les valeurs initiales et les descriptions des éléments individuels des tableaux de tags.

Longueur maximale des noms de tag

Chaque niveau d'une structure sera séparé par le caractère de soulignement ("_") dans le nom complet du tag dans Monitor Pro. Du fait que les noms de tag dans Monitor Pro ont une longueur maximale de 32 caractères, UAG créera des messages d'erreurs détectées pendant la procédure de génération du serveur de données ou de l'analyseur si des noms de tag résultants sont plus longs, et les tags concernés ne seront pas créés dans Monitor Pro.

Trames de communication du type de données ANY

Chaque variable DDT de type **API_IHM** sera affectée à une trame de communication dans UAG. Tous les éléments enfants appartiendront à la même trame de communication que celle de leur variable parent.

Les trames de communication de type de données ANY doivent être utilisées pour les variables DDT.

Les variables de type de données élémentaires peuvent aussi être ajoutées à une trame de type de données ANY.

La trame peut être définie dans l'**éditeur SCoD** ou dans l'**éditeur de personnalisation**.

Une trame de communication de type de données **ANY** ne sera pas générée dans l'application serveur si la gestion OPC n'est pas activée. Dans ce cas, l'analyseur et le générateur serveur de données émettront des messages d'erreurs détectées.

Si la prise en charge OPC est activée, il existe une autre option de traitement des variables **API_IHM** en variables localisées ou non.

Si la gestion OPC n'est pas activée, les variables **API_IHM** seront affectées à des adresses de mémoire localisées de l'automate comme dans les versions précédentes.

Pilotes de communication MBPLUS et TCPIP

Les pilotes de communication pris en charge sont MBPLUS et TCPIP. Ainsi, il sera possible d'affecter un serveur de données à un réseau MBPlus, si la gestion OPC est activée.

Si la gestion OPC est désactivée et qu'un serveur de données est encore affecté à un réseau MBPlus, l'analyseur et le générateur du serveur de données émettront un message d'erreur détectée et rien ne sera généré pour ce serveur de données.

Feuille de style fournie

Une feuille de style (*OFSDeviceConfig.xsl*) est fournie et peut être utilisée pour définir un format d'exportation pour un serveur de données, afin que UAG génère automatiquement un fichier d'importation de registre (fichier **.reg*) pour la configuration des alias d'équipements dans OFS, le serveur OPC de Schneider Electric.

L'outil de configuration d'OFS ne peut pas être utilisé pour modifier ces paramètres de configuration car il ne peut pas importer les données du registre vers sa base de données de configuration.

Transfert de variables DDT vers une IHM générique

Introduction

Un fichier XML est généré pour transférer les variables DDT vers une IHM générique.

Ce fichier XML utilisé pour l'IHM générique contient toutes les variables **API_IHM** qui seront transférées entre un API et un serveur de données.

Il contient de plus tous les éléments enfants d'une DDT qui ont un type de donnée élémentaire, avec les attributs de toutes les propriétés de variable comme pour les variables normales.

Annexes

Les annexes fournissent des informations sur les fichiers CSV et XML destinés à être importés et exportés.

Contenu de cette annexe

Cette annexe contient les chapitres suivants :

Chapitre	Titre du chapitre	Page
A	Format des fichiers CSV d'importation et d'exportation	445
B	Format des fichiers XML d'importation et d'exportation	483

Annexe A
Format des fichiers CSV d’importation et d’exportation

Introduction

Ce chapitre donne le format des fichiers CSV pour les processus d’importation et d’exportation.

Contenu de ce chapitre

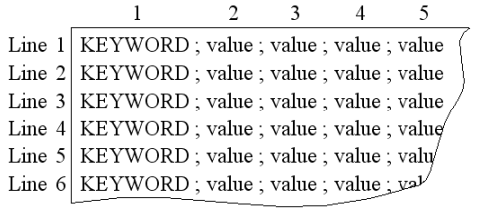
Ce chapitre contient les sujets suivants :

Sujet	Page
Format général	446
Les éléments du modèle physique : format de fichier CSV	449
Eléments du modèle topologique : format de fichier CSV	456
Instruments : format de fichier CSV	469
Valeurs initiales : format de fichier CSV	470
Format d’exportation des bus de terrain	471
Bibliothèque des types de modèles : format de fichier CSV	475
Suppression et changement de nom : Format de fichier CSV	479
Bibliothèque SCoD : format de fichier CSV	482

Format général

Structure générale d'un fichier CSV

Le schéma suivant illustre la structure générale d'un fichier CSV utilisé par Unity Application Generator en importation ou exportation :



Remarques générales

Reportez-vous aux remarques suivantes :

- le **premier élément** est le mot clé permettant d'identifier l'objet. Il doit correspondre exactement à la chaîne présente dans le tableau et être écrit en lettres majuscules. Les mots clés sont les mêmes pour toutes les versions linguistiques de Unity Application Generator.
- Les **références de document**, si elles existent dans l'objet, se trouvent au bout de la ligne. Chaque référence est constituée de deux champs : le fichier spécifique et les paramètres de l'outil spécifique. Le nombre de références de document n'est pas limité.
- Les champs comportant du **texte libre**, par exemple un commentaire, peuvent contenir des **sauts de ligne** et des **tabulations**. Un saut de ligne doit être défini par `\n`, une tabulation par `\t`.
- Les numéros de ce tableau ne sont donnés que pour améliorer l'aperçu, ils ne font **pas** partie du fichier CSV.
- *: Les **champs obligatoires** sont indiqués par une étoile. Les champs non obligatoires doivent rester vides s'ils ne sont pas renseignés (`;;`).
- Le choix du **séparateur** s'effectue dans la section Options. Evitez d'utiliser le caractère de séparation n'importe où, par exemple dans un texte libre, sinon les champs ne seront pas identifiés correctement.

Séparateur

Le caractère séparateur peut être une tabulation, une virgule, un point-virgule ou tout autre caractère. Le caractère séparateur est défini dans la boîte de dialogue **Vue** →**Options** dans Unity Application Generator.

NOTE : N'utilisez pas de caractère séparateur pouvant être utilisé ailleurs dans votre projet. UAG considèrerait ce caractère comme un séparateur d'importation/exportation.

Options de l'analyseur pour l'importation

Les options d'analyse du projet servent également à l'importation pour signaler des alertes ou arrêter le système après la détection d'un certain nombre d'erreurs ou d'alertes :

- `Afficher avertissement`
- `Arrêter après n erreurs`
- `Arrêter après n avertissements`

Mot-clé

Le mot-clé constitue la première entrée d'une ligne. Il définit le type d'objet décrit dans le reste de la ligne. Il doit être orthographié exactement comme dans le tableau suivant :

Mot-clé	Signification
`INSTRUMENT`	Importation d'instruments et/ou d'objets du modèle physique
`INITIALVALUE`	Importation de valeurs initiales de variables (à des fins de réglage)
`"userdefined physical level name"`	Importation/exportation d'un élément physique
`CONTROLMODULE`	Importation/exportation d'un module de commande
`VARIABLE`	Importation/exportation d'une variable
`ALARM`	Importation/exportation d'une alarme
`BITFIELD`	Exportation d'un champ d'élément binaire
`API`	Importation/exportation d'un API
`RACK`	Importation/exportation d'un châssis
`CONFIGURATION_RACK`	Importation/exportation d'un châssis Advantys
`HWMODULE`	Importation/exportation d'un module matériel
`CONFIGURATION_MODULE`	Importation/exportation d'un module matériel Advantys
`IOPOINT`	Importation/exportation d'un point d'E/S
`CONFIGURATION_IOPOINT`	Importation/exportation d'un point d'E/S Advantys
`FBSLAVE`	Importation/exportation d'un esclave de bus de terrain
`FBMODULE`	Importation/exportation d'un module de bus de terrain
`FBIOPOINT`	Importation/exportation d'un point d'E/S de bus de terrain
`IHM`	Importation/exportation d'une IHM
`CONTROLDOMAIN`	Importation/exportation d'une zone de conduite
`DATASERVER`	Importation/exportation d'un serveur de données
`NETWORKNODE`	Importation/exportation d'un partenaire réseau ou d'un autre abonné
`NETWORKSEGMENT`	Importation/exportation d'un segment de réseau

Mot-clé	Signification
ROUTINGPATH	Importation/exportation d'un chemin de routage
CHANNEL	Importation/exportation d'un canal
COMMUNICATIONPATH	Importation/exportation d'un chemin de communication
NETCOMMUNICATION	Importation/exportation d'une communication NET
PLCCOMMUNICATION	Importation/exportation d'une communication API
HMICOMMUNICATION	Importation/exportation d'une communication IHM
NETCOMMUNICATIONTABLE	Importation/exportation d'une table de communication NET
PLCCOMMUNICATIONTABLE	Importation/exportation d'une table de communication API
LINK	Importation/exportation d'une liaison
INTERLOCK	Importation/exportation d'un forçage
INTERLOCK_CONDITION	Importation/exportation d'une condition de forçage
TEMPLATETYPE	Importation/exportation d'un type de modèle
HMISYMBOL	Importation/exportation d'un symbole IHM
TEMPLATETYPEPROPERTY	Importation/exportation d'une propriété de type de modèle
DELETE	Suppression d'objets
RENAME	Changement de nom d'objets

Pour RENOMMER et SUPPRIMER, veuillez vous reporter à la section Suppression et changement de nom : format de fichier CSV *(voir page 479)*.

Sauts de ligne ou tabulations dans un texte libre

Les champs contenant du texte libre, par exemple le commentaire d'un module de commande, peuvent contenir des sauts de ligne et des tabulations.

- Un **saut de ligne** doit être défini par `\n` dans le fichier CSV.
- Une **tabulation** doit être définie par `\t` dans le fichier CSV.

Ligne de commentaire

Il est possible d'insérer des lignes de commentaire dans un fichier CSV.

Une **ligne de commentaire** doit être définie par `#` comme premier caractère de la ligne.

Ordre dans les fichiers à exporter

L'ordre des éléments dans le fichier à exporter (fichier cible) correspond aux arborescences du modèle physique et du modèle topologique. Les lignes sont générées les unes à la suite des autres c'est-à-dire qu'après la ligne de l'objet parent suivent les lignes de tous les objets enfants qu'il contient.

Les éléments du modèle physique : format de fichier CSV

Références des documents

Pour importer dans UAG un fichier préalablement exporté, contenant des attributs `document_ref`, vous devez tout d'abord supprimer les attributs `document_ref` du fichier exporté.

Attributs non pris en charge par le format CSV

Les attributs suivants ne sont pas pris en charge par le format CSV (uniquement pris en charge par le format XML) :

- valeurs initiales
- énumérations des propriétés librement définies
- attribut d'information
  Des attributs d'information tels que `fullName`, `variableNamePrefix`, `variableNameSuffix` ne sont pas pris en charge par le format CSV.
  Ils ne sont exportés qu'à titre d'information, en utilisant le format XML.

Elément physique structurel

La ligne d'un fichier CSV d'importation/exportation d'éléments physiques structurels contient les éléments suivants, délimités par un séparateur (* = champs obligatoires mais pouvant être vierges).

1	**** level_name***	2	**** name***	3	**** path***
4	**** description***	5	**** comment***	6	***document_ref_1***
7	***document_parameters_1***	...	***document_ref_i***	...	***document_parameters_i***
...	***document_ref_n***	...	***document_parameters_n***		

Exemples :

```
Area;Zone1;;zone1;
ProcessCell;Cell1;\Zone1;cell1;
Unit;Unit1;\Zone1\Cell1;unit1;
```

Elément physique fonctionnel

La ligne d'un fichier CSV d'importation/exportation d'éléments physiques fonctionnels contient les éléments suivants, délimités par un séparateur (* = champs obligatoires mais pouvant être vierges).

1	***level_name**	2	***name**	3	***path**
4	* ***description**	5	* ***plc_name**	6	* ***control_domain_name**
7	* ***section_name**	8	* ***picture_name**	9	* ***comment**
10	***template_type**	11	***template_library**	12	***hmi_symbol_configuration_type**
13	***hmiSymbol_name**	14	***hmiSymbol_type**	15	***document_ref_1**
16	***document_parameters_1**	17	***document_ref_i**	18	***document_parameters_i**
19	***document_ref_n**	20	***document_parameters_n**	-	-

Exemple :

```
Equipment;Equipment2;\Zone1\Cell1\Unit1;Description of
Equipment2;PLC1;ZConduite1;Equipment2;;
```

Module de commande

La ligne d'un fichier CSV d'importation/exportation d'un module de commande contient les éléments suivants, délimités par un séparateur (* = champs obligatoires mais pouvant être vierges).

1	* CONTROLMODULE	2	* ***name**	3	***full_path**
4	* ***SCod_type_name**	5	***library_name**	6	* ***description**
7	* ***plc_name**	8	* ***device_master_name**	9	* ***comment**
10	* ***separator_line_information**	11	* ***hmi_symbol_type**	12	* ***hmi_symbol_name**
13	**template_type_control_module_name**	14	***document_ref_1**	15	***document_parameters_1**
16	***document_ref_i**	17	***document_parameters_i**	18	***document_ref_n**

Liaison

La ligne d'un fichier CSV d'importation/exportation d'une liaison contient les éléments suivants, délimités par un séparateur (* = champs obligatoires).

1	* `LINK`	2	* **controlmodule_full_name**	3	* **controlmodule_pin_name**
4	* **variable_parent_type**	5	* **variable_parent_full_path**	6	* **variable_full_name**
7	* **description**	8	* **controlmodule_full_path**	-	-

- ***LINK*** : mot-clé de l'élément d'importation/exportation.
- ***controlmodule_full_name*** : nom complet de l'instance du module de commande (par exemple, `FunctionalPhysicalElement_controlmodule`).
- **controlmodule_pin_name** : nom de la broche connectée (32 caractères au maximum). Le nom doit exister dans la définition des broches de SCoD pour ce SCoD (type de module de commande).
- **variable_parent_type** : soit CONTROLMODULE, soit PHYSICALELEMENT
- **variable_parent_full_path** : chemin d'accès complet du module de commande ou de l'élément physique fonctionnel (par exemple, `\Zone1\CelluleProcédé1\Unité1\Equipment1\ModuledeCommande`).
- **variable_full_name** : nom complet de la variable connectée (par exemple, `FunctionalPhysicalElement_controlmodule_variable`).
  La longueur des noms de
  - `FunctionalPhysicalElement`,
  - `controlmodule` et
  - `variable`

  dépendent de la personnalisation.
- **description** : description de la liaison, 255 caractères maximum.
- **controlmodule_full_path** : chemin d'accès complet de l'instance de module de commande (par exemple, `\Zone1\CelluleProcédé1\Unité1\Equipment1\moduledecommande`).

NOTE : Une liaison est valide si et seulement si toutes les conditions suivantes sont remplies :

- la broche est du type E/S **Entrée**, **Sortie** ou **Entrée/Sortie** et n'est pas connectée,
- le type de connexion de la variable n'est pas **IHM** et
- la broche et la variable ont le même type de données.

Forçage

La ligne d'un fichier CSV d'importation/exportation d'un forçage contient les éléments suivants, délimités par un séparateur (* = champs obligatoires).

1	* `INTERLOCK`	2	* **controlmodule_name**	3	* **controlmodule_pin_name**
4	* **interlock_definition**	5	* **comment**	6	* **controlmodule_full_path**

- ***INTERLOCK*** : mot-clé de l'élément d'importation/exportation.
- ***controlmodule_name*** : nom complet de l'instance du module de commande (par exemple, `FunctionalPhysicalElement_controlmodule`).
- **controlmodule_pin_name** : nom de la broche connectée (32 caractères au maximum). Le nom doit exister dans la définition des broches de SCoD pour ce SCoD (type de module de commande).
- **interlock_definition** : une `interlock_definition` est constituée des éléments suivants :
  - `interlock_conditions` référencées par `$n`, où `n` est le numéro séquentiel de l'une des lignes `interlock_condition` immédiatement au-dessous de la ligne `INTERLOCK` (en commençant par 1),
  - des opérateurs logiques combinant les `interlock_conditions` (**and**, **or**, **xor**, **not**),
  - des parenthèses pour définir la priorité des opérateurs.

  Voici un exemple d'une `interlock_definition` : `($1 or not $2) and $3`.
  64 000 caractères max. Toutes les `interlock_conditions` référencées par une `interlock_definition` (`$n`) doivent exister.
- **comment** : 64 000 caractères max.
- **controlmodule_full_path** : chemin d'accès complet de l'instance de module de commande (par exemple, `\Zone1\CelluleProcédé1\Unité1\Equipment1\moduledecommande`).

Condition de forçage

Une `condition de forçage` est une comparaison logique entre des valeurs de variables (provenant d'un autre module de commande ou élément physique fonctionnel) ou entre une valeur de variable et une constante (littérale). Le résultat d'une telle condition est du type de donnée `Booléen`.

La ligne d'un fichier CSV d'importation/exportation d'une condition de forçage contient les éléments suivants, délimités par un séparateur (* = champs obligatoires).

1	* `INTERLOCK_CONDITION`	2	**variable_parent_type**	3	**variable_parent_full_path**
4	**variable_full_name**	5	**condition_operator**	6	**variable_parent_type** (variable de condition)
7	**variable_parent_full_path** (variable de condition)	8	**variable_full_name** (variable de condition)	9	**condition_literal**
10	**description**	11	**interlock_literal**	-	-

- ***INTERLOCK_CONDITION*** : mot-clé de l'élément d'importation/exportation.
- **variable_parent_type** : soit CONTROLMODULE, soit PHYSICALELEMENT

- **variable_parent_full_path** : chemin d'accès complet du module de commande ou de l'élément physique fonctionnel (par exemple, `\Zone1\CelluleProcédé1\Unité1\Equipment1\ModuledeCommande`).
- **variable_full_name** : nom complet de la variable défini dans n'importe quel module de commande ou élément physique fonctionnel à comparer à une autre variable ou à une valeur littérale (par exemple, `FunctionalPhysicalElement_controlmodule_variable`). Le nom complet de la variable correspond aux règles de création de variables définies dans la personnalisation.
- **condition_operator** : l'un des opérateurs suivants : **=**, **<>**, **>**, **<**, **=>**, **=<**. L'opérateur dépend du type de données.
- **variable_full_name** (variable de condition) : nom complet de la variable pour comparaison avec `variable_full_name` ci-dessus.
- **condition_literal** : valeur littérale pour comparaison avec `variable_full_name`. 64 caractères max.
- **description** : description de 64 caractères maximum.

Variable

La ligne d'un fichier CSV d'importation/exportation d'une variable contient les éléments suivants, délimités par un séparateur (* = champs obligatoires mais pouvant être vierges).

1	* `VARIABLE`	2	* ***name***	3	* ***parent_type*** Les valeurs autorisées sont `FUNCTIONAL_PHYSICAL_ELEMENT_NAME` ou `CONTROLMODULE`.
4	* ***controlmodule_path*** ou * **FunctionalPhysicalElement_path**	5	* ***controlmodule_name***	6	* ***description***
7	* ***data_type_name***	8	* ***connection_type_name***	9	* ***in_out_type_name***
10	***address_segment***	11	***address_offset***	12	***initial_value***
13	***outputFailure_behavior***	14	***outputFailure_value***	15	***command_name***
16	***inverse***	17	***variable_used***	18	***graphical_symbol***
19	***graphical_symbol_name***	20	***display_format***	21	***measurement_unit***
22	***scaling_min***	23	***scaling_max***	24	***value_boundary_min***
25	***value_boundary_max***	26	***access_level***	27	***state0_text***
28	***state1_text***	29	***alarm_value***	30	***alarm_1_priority***
31	***alarm_1_operator_id***	32	**alarm_1_limit**	33	***alarm_1_text***
34	***alarm_2_priority***	35	***alarm_2_operator_id***	36	**alarm_2_limit**
37	***alarm_2_text***	38	***alarm_3_priority***	39	***alarm_3_operator_id***
40	**alarm_3_limit**	41	***alarm_3_text***	42	***alarm_4_priority***
43	***alarm_4_operator_id***	44	**alarm_4_limit**	45	***alarm_4_text***

46	***alarm_5_priority***	47	***alarm_5_operator_id***	48	**alarm_5_limit**
49	***alarm_5_text***	50	***alarm_6_priority***	51	***alarm_6_operator_id***
52	**alarm_6_limit**	53	***alarm_6_text***	54	***alarm_7_priority***
55	***alarm_7_operator_id***	56	**alarm_7_limit**	57	***alarm_7_text***
58	***alarm_8_priority***	59	***alarm_8_operator_id***	60	**alarm_8_limit**
61	***alarm_8_text***	62	**archive**	63	**archive_file**
64	**communication_frame**	65	**get_free_property_1**	66	***get_free_property_2***
67	***get_free_property_3***	68	***get_free_property_4***	69	***get_free_property_5***
70	***get_free_property_6***	71	***get_free_property_7***	72	***get_free_property_8***
73	***get_free_property_9***	74	***get_free_property_10***	75	***get_free_property_11***
76	***get_free_property_12***	77	***get_free_property_13***	78	***get_free_property_14***
79	***get_free_property_15***	80	***get_free_property_16***	81	***get_free_property_17***
82	***get_free_property_18***	83	***get_free_property_19***	84	***get_free_property_20***
85	***alarm_group_name***	86	***initial_value***	87	***SCoD_variable_id***
88	***deadband_mode***	89	***deadband_value***	90	***raw_min***
91	***raw_max***	92	***archive_sample_rate***	-	**-**

NOTE :

- Veuillez également consulter la section `ALARME` ci-dessous.
- Les informations concernant l'utilisation des énumérations système (reportez-vous au *Manuel utilisateur de l'éditeur SCoD -> Définition d'un SCoD -> Enumérations système, Infos projet*) ne figurent pas dans le fichier CSV. En d'autres termes, une variable créée par importation utilisera les énumérations système définies dans l'éditeur SCoD (le cas échéant).

Alarme

La ligne d'un fichier CSV d'importation/exportation d'alarmes contient les éléments suivants, délimités par un séparateur (* = champs obligatoires).

1	* ALARM	2	* **variable_full_name**	3	* **head_variable_parent type** FUNCTIONAL_PHYSICAL_ELEMENT ou CONTROLMODULE
4	* **head_variable_parent_path**	5	* **alarm_number**	6	**alarm_priority**
7	**alarm_operator**	8	**alarm_limit**	9	**alarm_text**
10	**alarm_timer_mode**	11	**alarm_timer_variable_parent_path**	12	**alarm_timer_variable_full_name**

NOTE :

- Pour des raisons de compatibilité, VARIABLE détient encore les éléments d'alarme ***alarm_1* jusqu'à alarm_8***. Mais si un fichier CSV contient des lignes incluant le mot-clé ALARM, ces lignes sont prioritaires lors de l'importation sur les lignes incluant le mot-clé VARIABLE. Si plus de 8 alarmes sont définies, ces informations sont uniquement présentes à la ligne ALARM.
- Une ligne ALARM contenant des données invalides est omise et la ligne ALARM suivante valide se verra attribuer le numéro d'alarme normalement affecté à la ligne ALARM incluant des données invalides.

Champ d'élément binaire

La ligne d'un fichier CSV d'exportation d'un champ d'élément binaire contient les éléments suivants, délimités par un séparateur (* = champs obligatoires).

1	* BITFIELD	2	**variable_parent_type** FUNCTIONAL_PHYSICAL_ELEMENT_NAME ou CONTROLMODULE	3	**variable_parent_full_path**
4	**variable_full_name**	5	**variable_parent_type** FUNCTIONAL_PHYSICAL_ELEMENT_NAME ou CONTROLMODULE	6	**variable_parent_full_path**
7	**variable_full_name**	-	-	-	-

Eléments du modèle topologique : format de fichier CSV

Références des documents

Pour importer dans UAG un fichier préalablement exporté, contenant des attributs `document_ref`, vous devez tout d'abord supprimer les attributs `document_ref` du fichier exporté.

API

La ligne d'un fichier CSV d'importation/exportation d'un API contient les éléments suivants, délimités par un séparateur (* = champ obligatoire).

1	* `API`	2	***plc_name***	3	***description***
4	***plc_project_name***	5	***plc_family***	6	***local_rack_part_number***
7	***comment***	8	***battery_coil***	9	***timer_register***
10	***time_of_day***	11	***hwmodules_0x_start***	12	***hwmodules_0x_end***
13	***hwmodules_1x_start***	14	***hwmodules_1x_end***	15	***hwmodules_3x_start***
16	***hwmodules_3x_end***	17	***hwmodules_4x_start***	18	***hwmodules_4x_end***
19	***plc_communication_4x_start***	20	***plc_communication_4x_end***	21	***hmi_0x_start***
22	***hmi_0x_end***	23	***hmi_4x_start***	24	***hmi_4x_end***
25	***netpartner_0x_start***	26	***netpartner_0x_end***	27	***netpartner_4x_start***
28	***netpartner_4x_end***	29	***reserve_0x_start***	30	***reserve_0x_end***
31	***reserve_1x_start***	32	***reserve_1x_end***	33	***reserve_3x_start***
34	***reserve_3x_end***	35	***reserve_4x_start***	36	***reserve_4x_end***
37	***fieldbus_3x_start***	38	***fieldbus_3x_end***	39	***fieldbus_4x_start***
40	***fieldbus_4x_end***	41	***document_ref_1***	42	***document_parameters_1t***
43	***document_ref_i***	44	***document_parameters_i***	45	***document_ref_nt***
46	***document_parameters_n***				

Description des entrées :

- ***plc_family*** : `Unity/Quantum`, `Unity/Premium` ou `Unity/M340`.
- ***local_rack_part_number*** : reportez-vous au tableau ci-dessous.
- ***battery_coil*** : adresse en mémoire d'état du contrôle de batterie (vide pour Unity Pro).
- ***timer_register*** : adresse en mémoire d'état du registre tempo (vide pour Unity Pro).
- ***time_of_day*** : adresse en mémoire d'état de l'horloge calendaire (premier registre) (vide pour Unity Pro).
- ***hwmodule_nx_start*** ...***hwmodule_nx_end*** :
  Quantum/Momentum : plages d'adresses de mémoire d'état (0x=%M, 1x=%I, 3x=%IW, 4x=%MW) pour les modules.
  Premium et M340 : vide car utilisation d'adresses topologiques.

- ***plc_communication_4x_start*** ... ***plc_communication_4x_end*** : plage d'adresses de mémoire d'état (4x=%MW) pour la communication API-API.
- ***hmi_nx_start*** ... ***hmi_nx_end*** : plage d'adresses de la mémoire d'état (0x=%M, 4x=%MW) pour les variables IHM.
- ***netpartner_nx_start*** ... ***netpartner_nx_end*** : plage d'adresses de la mémoire d'état (0x=%M, 4x=%MW) pour les variables de partenaire réseau.
- ***reserve_nx_start*** ... ***reserve_nx_end*** :
  plages des adresses de mémoire d'état réservées (0x=%M, 1x=%I, 3x=%IW, 4x=%MW).
  Premium et M340 : ***reserve_1x_start*** , ***reserve_1x_end***, ***reserve_3x_start*** et ***reserve_4x_end*** sont vides.
- ***fieldbus_nx_start*** ... ***fieldbus_nx_end*** :
  plages des adresses de mémoire d'état de bus de terrain (3x=%IW, 4x=%MW).
  Pour les API Unity Quantum uniquement (Profibus)

Tableau de ***local_rack_part_number*** :

local_rack_part_number	Taille	Description
140 XBP 002 00	2	Châssis Quantum à 2 emplacements
140 XBP 003 00	3	Châssis Quantum à 3 emplacements
140 XBP 004 00	4	Châssis Quantum à 4 emplacements
140 XBP 006 00	6	Châssis Quantum à 6 emplacements
140 XBP 010 00	10	Châssis Quantum à 10 emplacements
140 XBP 016 00	16	Châssis Quantum à 16 emplacements
TSX RKY 6	6	Châssis non extensible à 6 positions
TSX RKY 8	8	Châssis non extensible à 8 positions
TSX RKY 12	12	Châssis non extensible à 12 positions
TSX RKY 6E	6	Châssis extensible à 6 positions
TSX RKY 8E	8	Châssis extensible à 8 positions
TSX RKY 12E	12	Châssis extensible à 12 positions
TSX RKY 4EX	4	Châssis extensible à 4 positions
TSX RKY 6EX	6	Châssis extensible à 6 positions
TSX RKY 8EX	8	Châssis extensible à 8 positions
TSX RKY 12EX	12	Châssis extensible à 12 positions
BMX XBP 0400	4	Châssis non extensible M340 à 4 emplacements
BMX XBP 0600	6	Châssis non extensible M340 à 6 emplacements
BMX XBP 0800	8	Châssis non extensible M340 à 8 emplacements
BMX XBP 1200	12	Châssis non extensible M340 à 12 emplacements

Châssis

La ligne d'un fichier CSV d'importation/exportation d'un châssis contient les éléments suivants, délimités par un séparateur (* = champ obligatoire).

1	`* RACK`	2	***part_number***	3	***description***
4	***drop_number***	5	***plc_name***	6	***link_type***
7	***extension_rack_number***	8	***networkSegment***	9	***networkAddress***
10	***rack name***	11	***IsShared***	-	**-**

Description des entrées :

- ***part_number*** : reportez-vous au tableau ci-dessous.
- ***drop_number*** : numéro du châssis (entier). Pour un châssis local, il vaut 1.
  Plages : déporté `2 à 32`, SYMAX : `2 à 32`, E/S 800 : `2 à 32`, distribuées : `2 à 64` (=adresse Modbus Plus), générique : `1 à 9999`, bus E/S : `2`, E/S Modbus Plus : numéro de station inutile, E/S Ethernet : `1 à 16` (= numéro d'emplacement du NOE ou M1E (si M1E, `0`).
- ***plc_name*** : nom de l'API parent.
- ***link_type*** : type de connexion avec l'API. Les valeurs valides sont les suivantes : `Local`, `Local Momentum`, `XBus`, `Remote`, `DIO0`, `DIO1`, `DIO2`, `SYMAX`, `800IO`, `Generic`, `IOBUS`, `MBPIOLink0`, `MBPIOLink1`, `MBPIOLink2`, `ETHIO`.
- ***extension_rack_number*** : numéro du châssis d'extension. Chaîne vide pour le châssis local et numéro pour les châssis d'extension.

Pour la prise en charge d'Advantys, le mot-clé et les éléments suivants sont utilisés :

1	`* CONFIGURATION_RACK`	2	***part_number***	3	***description***
4	***drop_number***	5	***plc_name***	6	***link_type***
7	***extension_rack_number***	8	***rack_name***	9	***configuration_file_path***

Description des entrées supplémentaires :

- ***rack_name*** : nom du châssis. Obligatoire pour Advantys. Ne doit pas être vide.
- ***configuration_file_path*** : chemin d'accès au fichier XML de configuration dans lequel sont stockées toutes les informations concernant Advantys. Obligatoire pour Advantys. Ne doit pas être vide.

Tableau de ***part_number*** :

part_number	Taille	Description
140 XBP 002 00	2	Châssis Quantum à 2 emplacements
140 XBP 003 00	3	Châssis Quantum à 3 emplacements
140 XBP 004 00	4	Châssis Quantum à 4 emplacements
140 XBP 006 00	6	Châssis Quantum à 6 emplacements
140 XBP 010 00	10	Châssis Quantum à 10 emplacements
140 XBP 016 00	16	Châssis Quantum à 16 emplacements
TSX RKY 6	6	Châssis non extensible à 6 positions
TSX RKY 8	8	Châssis non extensible à 8 positions
TSX RKY 12	12	Châssis non extensible à 12 positions
TSX RKY 6E	6	Châssis extensible à 6 positions
TSX RKY 8E	8	Châssis extensible à 8 positions
TSX RKY 12E	12	Châssis extensible à 12 positions
TSX RKY 4EX	4	Châssis extensible à 4 positions
TSX RKY 6EX	6	Châssis extensible à 6 positions
TSX RKY 8EX	8	Châssis extensible à 8 positions
TSX RKY 12EX	12	Châssis extensible à 12 positions
8030-RRK-100	5	Châssis SY/MAX à 5 emplacements
8030-RRK-200	9	Châssis SY/MAX à 9 emplacements
8030-RRK-300	18	Châssis SY/MAX à 18 emplacements
174 CEV 300 20	16	Passerelle Modbus vers Ethernet, 16 esclaves max.
TSX ETG 100	32	Passerelle Modbus vers Ethernet, 32 esclaves max.
TSX ETG 1000	32	Passerelle Modbus vers Ethernet, serveur Web, 32 esclaves max.
EGX 100 MG	32	Passerelle Modbus vers Ethernet, 32 esclaves max.
EGX 400 MG	64	Passerelle Modbus vers Ethernet, serveur Web, 64 esclaves max.
Generic Gateway	64	Passerelle Modbus vers Ethernet, 64 esclaves max.
AS-H810-100	4	Châssis E/S 800 à 4 emplacements
AS-H810-208	4	Châssis E/S 800 à 4 emplacements
AS-H810-209	4	Châssis E/S 800 à 4 emplacements
AS-H819-100	7	Châssis E/S 800 à 7 emplacements
AS-H819-103	7	Châssis E/S 800 à 7 emplacements
AS-H819-209	7	Châssis E/S 800 à 7 emplacements
AS-H827-100	11	Châssis E/S 800 à 11 emplacements

part_number	Taille	Description
AS-H827-103	11	Châssis E/S 800 à 11 emplacements
AS-H827-209	11	Châssis E/S 800 à 11 emplacements
800 I/O (55)	55	Châssis E/S 800 à 55 emplacements
BMX XBP 0400	4	Châssis M340 à 4 emplacements
BMX XBP 0600	6	Châssis M340 à 6 emplacements
BMX XBP 0800	8	Châssis M340 à 8 emplacements
BMX XBP 1200	12	Châssis M340 à 12 emplacements
8030-RRK-100	5	Châssis SY/MAX à 5 emplacements
8030-RRK-200	9	Châssis SY/MAX à 9 emplacements
8030-RRK-300	18	Châssis SY/MAX à 18 emplacements
AS-H810-100	4	Châssis E/S 800 à 4 emplacements
AS-H810-208	4	Châssis E/S 800 à 4 emplacements
AS-H810-209	4	Châssis E/S 800 à 4 emplacements
AS-H819-100	7	Châssis E/S 800 à 7 emplacements
AS-H819-103	7	Châssis E/S 800 à 7 emplacements
AS-H819-209	7	Châssis E/S 800 à 7 emplacements
AS-H827-100	11	Châssis E/S 800 à 11 emplacements
AS-H827-103	11	Châssis E/S 800 à 11 emplacements
AS-H827-209	11	Châssis E/S 800 à 11 emplacements
Local Momentum	2	Châssis local Momentum
I/O Bus (128)	128	Châssis bus d'E/S à 128 emplacements
I/O Bus (44)	44	Châssis bus d'E/S à 44 emplacements
Ethernet I/O (128)	128	Châssis d'E/S Ethernet (Quantum)
Ethernet I/O (64)	64	Châssis d'E/S Ethernet (Momentum)
5240-MNET-PDPMV1	1	Profibus jusqu'à la passerelle Ethernet
MBP I/O	64	Châssis d'E/S ModbusPlus
Generic	255	Châssis générique
Advantys STB	255	Châssis Advantys
BMX XBP 0400	4	Châssis non extensible M340 à 4 emplacements
BMX XBP 0600	6	Châssis non extensible M340 à 6 emplacements
BMX XBP 0800	8	Châssis non extensible M340 à 8 emplacements
BMX XBP 1200	12	Châssis non extensible M340 à 12 emplacements

Module matériel

La ligne d'un fichier CSV d'importation/exportation d'un module matériel contient les éléments suivants, délimités par un séparateur (* = champ obligatoire).

1	* HWMODULE	2	* ***hwmodule_name***	3	* ***plc_name***
4	* ***link_type***	5	* ***drop_number***	6	***extension_rack_number***
7	***slot_number***	8	***network_segment_1***	9	***network_address_1***
10	***network_segment_2***	11	***network_address_2***	12	***0x_address***
13	***1x_address***	14	***3x_address***	15	***4x_address***
16	***timeout_behavior***	17	***link_number***	18	***config_file***
19	***last_import_time_stamp***	20	**original_type_HW_module**	21	**IsMonitoredEthernetModule**
19	***iOScannerRepetitiveRate***	20	***rack_name***	-	**-**

Description des entrées :

- ***hwmodule_name*** : Nom d'un module matériel valide, par exemple `CPS-111-00` (module d'alimentation) ou `CPU-113-03`, etc.
  hwmodule_name est exporté comme, par exemple, `140-CPU-311-10:2.60`, où `2.60` représente la version de l'UC qui peut être utilisée pour l'importation. Pour une compatibilité descendante, dans les versions antérieures où la version de l'UC n'est pas présente, la version par défaut de Unity Pro de l'UC est utilisée.
- ***plc_name*** : API dans lequel se trouve le module matériel.
- ***link_type*** : type de châssis. Les valeurs valides sont les suivantes : `Local`, `Local Momentum`, `XBus`, `Remote`, `DIO0`, `DIO1`, `DIO2`, `SYMAX`, `800IO`, `Generic`, `IOBUS`, `MBPIOLink0`, `MBPIOLink1`, `MBPIOLink2`, `ETHIO`.
- ***drop_number*** : numéro de station du châssis correspondant.
- ***slot_number*** : numéro de l'emplacement du module matériel.
- ***network_segment_1/_2*** : nom du segment de réseau. Ce champ n'est valide que pour les modules UC ou de communication, sinon il doit être vide.
- ***network_address_1/_2*** : adresse réseau du module matériel. Ce champ n'est valide que pour les modules UC ou de communication, sinon il doit être vide.
- ***0x_address*** : adresse 0x de début de mémoire d'état du module. Ce champ peut être vide. L'adresse est exportée, mais ignorée pendant l'importation (recalculée par Unity Application Generator).
- ***1x_address*** : adresse 1x de début de mémoire d'état du module. Ce champ peut être vide. L'adresse est exportée, mais ignorée pendant l'importation (recalculée par Unity Application Generator).
- ***3x_address*** : adresse 3x de début de mémoire d'état du module. Ce champ peut être vide. L'adresse est exportée, mais ignorée pendant l'importation (recalculée par Unity Application Generator).
- ***4x_address*** : adresse 4x de début de mémoire d'état du module. Ce champ peut être vide. L'adresse est exportée, mais ignorée pendant l'importation (recalculée par Unity Application Generator).

- ***timeout_behavior*** : 1 des 3 chaînes suivantes : `Disabled (Non activé), LastValue (Dernière valeur),UserDefined (Défini par l'utilisateur).` Ce champ n'est valide que pour les modules de sortie TOR, sinon il doit être vide.
- ***link_number*** : numéro de liaison (uniquement pour 170-BNO-6x1-00 ou vide).
- ***config_file*** : fichier de configuration (si le module matériel est maître de bus de terrain ou est vide).
- ***last_import_time_stamp*** : horodateur de dernière importation (si le module matériel est maître de bus de terrain ou est vide).
- ***original_type_HW_module*** : type d'origine du module matériel (si le module matériel est maître de bus de terrain ou est vide).
- ***iOScannerRepetitiveRate*** : durée de scrutation pour le scrutateur d'E/S.
- ***IsMonitoredEthernetModule*** : "1" si le module matériel est contrôlé par le module Ethernet ou est vide.
- ***rack_name*** : nom du châssis dans lequel se trouve le module matériel.

NOTE : Pour MEG, tous les champs sont obligatoires. Pour d'autres châssis, * indique les champs obligatoires.

Pour la prise en charge d'Advantys, le mot-clé et les éléments suivants sont utilisés :

1	* `CONFIGURATION_MODULE`	2	* ***hwmodule_name***	3	* ***plc_name***
4	**** drop_number***	5	**** rack_name***	6	**** configuration_file_path***
7	****slot_number***	8	**** extension_rack_number***	9	***network_segment***
10	***network_address***	11	***link_type***	-	**-**

Description des entrées supplémentaires :
- ***rack_name*** : nom du châssis. Obligatoire pour Advantys. Ne doit pas être vide.
- ***configuration_file_path*** : chemin d'accès au fichier XML de configuration dans lequel sont stockées toutes les informations concernant Advantys. Obligatoire pour Advantys. Ne doit pas être vide.

Point d'E/S

La ligne d'un fichier CSV d'importation/exportation d'un point d'E/S contient les éléments suivants, délimités par un séparateur (* = champ obligatoire).

1	`* IOPOINT`	2	* ***variable_parent_type***	3	* ***variable_full_path***
4	* ***variable_full_name***	5	* ***plc_name***	6	* ***link_type***
7	* ***rack_number***	8	***extension_rack_number***	9	***slot_number***
10	***io_point***	11	***rack_name***	-	**-**

Description des entrées :

- ***variable_full_name*** : nom complet (généré) de la variable affectée au point d'E/S ; par exemple `EquipmentName_DeviceName_VariableName`.
- ***plc_name*** : API contenant le module matériel.
- ***link_type*** : type de châssis. Les valeurs valides sont les suivantes : `Local`, `Remote`, `DIO0`, `DIO1`, `DIO2`, `SYMAX`, `800IO`, `Generic`, `ETHIO`, `MBPIOLink0`, `MBPIOLink1`, `MBPIOLink2`, `Local Momentum`, `IOBUS`. Correspond au châssis contenant le point d'E/S.
- ***rack_number*** : numéro de station du châssis, valeur entière.
- ***slot_number*** : numéro de l'emplacement du module matériel, valeur entière.
- ***io_point*** : position dans le module d'E/S ; valeur entière. Dans un module panaché avec connexions d'ENTREE et de SORTIE, les connexions d'entrée sont comptées les premières, par exemple sur un module à 4 connexions d'entrée, la première connexion de sortie est 5.

NOTE : Pour MEG, tous les champs sont obligatoires. Pour d'autres châssis, * indique les champs obligatoires. Dans le module matériel, io_point peut être vide ou non.

Pour la prise en charge d'Advantys, le mot-clé et les éléments suivants sont utilisés :

1	`* CONFIGURATION_IOPOINT`	2	**** variable_parent_type***	3	**** variable_fullpath***
4	**** full_variable_name***	5	**** plc_name***	6	**** link_type***
7	**** rack_number***	8	**** extension_rack_number***	9	****slot_number***
10	***Position du point d'E/S dans le module matériel***	11	**** description***	12	**** inout_name***
13	**** datatype_display_name***	14	**** byteorWord_name***	15	**** bit_number***
16	**** rack_name***	17	**** configuration_file_path***	-	-

Description des entrées supplémentaires :

- ***Position du point d'E/S dans le module matériel*** : facultatif pour le mappage des variables ES_API avec un point d'E/S sans position de point d'E/S.
- ***description*** : description du point d'E/S. Obligatoire pour Advantys. Ne doit pas être vide.
- ***inout_name*** : nom d'entrée/sortie du point d'E/S. Obligatoire pour Advantys. Ne doit pas être vide.
- ***byteorWord_name*** : nom d'ordre d'octets du point d'E/S. Obligatoire pour Advantys. Ne doit pas être vide.
- ***datatype_display_name*** : nom de type de données du point d'E/S. Obligatoire pour Advantys. Ne doit pas être vide.
- ***bit_number*** : numéro de bit du point d'E/S. Obligatoire pour Advantys. Ne doit pas être vide.
- ***rack_name*** : nom du châssis. Obligatoire pour Advantys. Ne doit pas être vide.
- ***configuration_file_path*** : chemin d'accès au fichier XML de configuration dans lequel sont stockées toutes les informations concernant Advantys. Obligatoire pour Advantys. Ne doit pas être vide.

IHM

La ligne d'un fichier CSV d'importation/exportation d'une IHM contient les éléments suivants, délimités par un séparateur (* = champ obligatoire).

1	* `IHM`	2	***name***	3	**** hmi_type***
4	***comment***	5	***document_ref_1***	6	***document_parameters_1***
...	***document_ref_i***	...	***document_parameters_i***	...	***document_ref_n***
...	***document_parameters_n***				

hmi_type : type d'IHM. Les valeurs valides sont `iFIX`, `Generic HMI` ou `Monitor Pro`.

Zone de conduite

La ligne d'un fichier CSV d'importation/exportation d'une zone de conduite contient les éléments suivants, délimités par un séparateur (* = champ obligatoire).

1	`* CONTROLDOMAIN`	2	**** name***	3	**** hmi_name***
4	**** description***	5	***comment***		

Description des entrées :

- ***hmi_name*** : nom de l'IHM parent.

Groupe d'images

La ligne d'un fichier CSV d'importation/exportation d'un groupe d'images contient les éléments suivants, délimités par un séparateur (* = champ obligatoire).

1	* PICTUREGROUP	2	* ***hmi_name***	3	***parent_picturegroup***
4	* ***name***	5	* ***description***		

Description des entrées :

- **parent_picturegroup** : nom du groupe d'images parent. Si cette valeur est vide, il s'agira d'un groupe racine.

Image

La ligne d'un fichier CSV d'importation/exportation d'une image contient les éléments suivants, délimités par un séparateur (* = champ obligatoire).

1	* PICTURE	2	* ***hmi_name***	3	***parent_picturegroup***
4	* ***name***	5	* ***description***		

Description des entrées :

- **parent_picturegroup** : nom du groupe d'images parent. Si cette valeur est vide, il s'agira d'un groupe racine.

Serveur de données

La ligne d'un fichier CSV d'importation/exportation d'un serveur de données contient les éléments suivants, délimités par un séparateur (* = champ obligatoire).

1	* DATASERVER	2	* ***name***	3	***export_format_name***
4	***primary_network1_segment***	5	**primary_network1_address**	6	***timeout_min***
7	***timeout_sec***	8	***timeout_ms***	9	**primary_network2_segment**
10	**primary_network2_address**	11	***comment***	12	**secondary_network1_segment**
13	**secondary_network1_address**	14	**secondary_network2_segment**	15	**secondary_network2_address**
16	***document_ref_1***	17	***document_parameters_1***	18	***document_ref_i***
19	***document_parameters_i***	20	***document_ref_n***	21	***document_parameters_n***

Description des entrées :

- ***export_format_name*** : nom du format d'exportation.
- **primary_network2_segment** : segment pour carte réseau redondante en option dans le PC serveur de données.
- **primary_network2_address** : adresse pour carte réseau redondante en option dans le PC serveur de données.

Abonné réseau

La ligne d'un fichier CSV d'importation/exportation d'un abonné réseau contient les éléments suivants, délimités par un séparateur (* = champ obligatoire).

1	* NETWORKNODE	2	* ***name***	3	* ***network_node_type***
4	***description***	5	***network_segment***	6	***network_address***
7	***timeout_min***	8	***timeout_sec***	9	***timeout_ms***
10	***export_file***	11	***bitmap_file***	12	***comment***
13	***document_ref_1***	14	***document_parameters_1***	...	***document_ref_i***
...	***document_parameters_i***	...	***document_ref_n***	...	***document_parameters_n***

Description des entrées :

- ***network_node_type*** : les chaînes valides sont `NetPartner` et `OtherNode`.

Segment de réseau

La ligne d'un fichier CSV d'importation/exportation d'un segment de réseau contient les éléments suivants, délimités par un séparateur (* = champ obligatoire).

1	* NETWORKSEGMENT	2	* ***name***	3	* ***network_type***
4	***subnetmask***	5	***default_gateway***		

Description des entrées :

- ***network_type*** : Les chaînes valides sont `ModbusPlus` et `Ethernet`.
- ***subnetmask*** : adresse IP correcte. Seulement pour le type de réseau Ethernet ; sinon cette entrée est vide.
- ***default_gateway*** : adresse IP correcte. Seulement pour le type de réseau Ethernet ; sinon cette entrée est vide.

Chemin de routage

La ligne d'un fichier CSV d'importation/exportation d'un chemin de routage contient les éléments suivants, délimités par un séparateur (* = champ obligatoire).

1	* ROUTINGPATH	2	* ***senders_segment_name***	3	* ***receivers_segment_name***
4	* ***network_path***				

network_path : de une à quatre adresses Modbus+ séparées par un point, par exemple : `2.12.11.3`, `3.4` ou `1`.

NOTE : La ligne commençant par `ROUTINGPATH` n'est valide que pour les segments de réseau Modbus Plus.

Canal

La ligne d'un fichier CSV d'importation/exportation d'un canal contient les éléments suivants, délimités par un séparateur (* = champ obligatoire).

1	* CHANNEL	2	***channel_name**	3	***plc_name**
4	* ***communication_partner**	5	***comment**	-	-

Chemin de communication

La ligne d'un fichier CSV d'importation/exportation d'un chemin de communication contient les éléments suivants, délimités par un séparateur (* = champ obligatoire).

1	* COMMUNICATIONPATH	2	***channel_name**	3	***plc_name**
4	* ***plc_network_segment**	5	* ***plc_network_address**	6	* ***partner_network_segment**
7	* ***partner_network_address**	8	**timeout_minute**	9	**timeout_second**
10	**timeout_milli_second**	11	**iOScannerRepetitiveRate**	12	**sender_network_segment**
13	**receiver_network_segment**	14	**global_network_path**	-	-

Communication réseau

La ligne d'un fichier CSV d'importation/exportation d'une communication réseau contient les éléments suivants, délimités par un séparateur (* = champ obligatoire).

1	* NETCOMMUNICATION	2	***plc_name**	3	***channel_name**
4	* ***bit_or_word_type**	5	* ***start_address**	6	* ***length**
7	***redundant_start_address**	8	***redundant_length**	-	-

Communication API

La ligne d'un fichier CSV d'importation/exportation d'une communication API contient les éléments suivants, délimités par un séparateur (* = champ obligatoire).

1	* PLCCOMMUNICATION	2	***plc_name**	3	***channel_name**
4	* ***bit_or_word_type**	5	* ***start_address**	6	* ***length**
7	* ***redundant_start_address**	8	* ***redundant_length**	9	***plc_communication_type**

Communication IHM

La ligne d'un fichier CSV d'importation/exportation d'une communication IHM contient les éléments suivants, délimités par un séparateur (* = champ obligatoire).

1	* HMICOMMUNICATION	2	***plc_name***	3	***channel_name***
4	* ***communication_type***	5	* ***communication_frame***	6	* ***data_type***
7	* ***frame_type***	8	***start_address***	9	***length***

Table de communication

La ligne d'un fichier CSV d'importation/exportation d'une table de communication contient les éléments suivants, délimités par un séparateur (* = champ obligatoire).

1	* NETCOMMUNICATIONTABLE ou PLCCOMMUNICATIONTABLE	2	***plc_name***	3	***channel_name***
4	* ***bit_or_word_type***	5	* ***bit_or_word_position***	6	* ***variable_parent_path***
7	***variable_full_name***	-	-	-	-

Instruments : format de fichier CSV

Ordre des données sur une ligne

Chaque ligne du fichier CSV destiné à l'importation d'instruments représente un instrument.

La ligne contient les éléments suivants, séparés par un séparateur (* = champ obligatoire).

1	`* INSTRUMENT`	2	* ***controlmodule_name***	3	* ***controlmodule_type***
4	* ***library_name***	5	* ***controlmodule_description***	6	***controlmodule_comment***
7	***full_path***	8	***hmi_symbol_type***	9	***hmi_symbol_name***

L'étoile, *, indique les champs obligatoires. Les champs non obligatoires doivent rester vides s'ils ne sont pas renseignés (`;;`).

NOTE :

- La première entrée doit exactement correspondre à la chaîne de caractères `INSTRUMENT`, en majuscules, pour identifier ce que contient la ligne. Ce mot-clé est le même pour toutes les versions linguistiques de Unity Application Generator.
- L'élément ***full_path*** sera ignoré à l'importation et peut donc rester vierge. L'objet sera toujours importé vers la liste des instruments UAG.
- Les éléments ***hmi_symbol_type*** et ***hmi_symbol_name*** sont facultatifs.
- Les numéros de ce tableau ne sont donnés que pour améliorer l'aperçu, ils ne font **pas** partie du fichier CSV.

Exemple de ligne

```
INSTRUMENT;OPM1;OPMode_P10;UAGProcessLibVCV2.0;OPM;OP mode
selection;;VijeoCitect Genie;lib_basic.OpMode_P10
```

Valeurs initiales : format de fichier CSV

Ordre des données sur une ligne

Chaque ligne d'un fichier CSV d'importation de valeurs initiales représente une valeur initiale d'une variable.

La ligne contient les éléments suivants, séparés par un séparateur (* = champs obligatoires).

1	* `INITIALVALUE`	2	***variable_full_name***	3	* ***variable_parent_type***
4	***variable_path***	5	* ***initial_value***	-	-

Description des entrées :

- ***variable_full_name*** : cette entrée contient le nom complet de la variable pour laquelle la valeur initiale doit être changée, par exemple `NomEquipement_NomModule_NomVariable`. Un contrôle est effectué sur le nom pour savoir s'il s'agit du nom d'une variable existante dans le projet Unity Application Generator.
- ***variable_parent_type*** : soit CONTROLMODULE, soit PHYSICALELEMENT
- ***initial_value*** : cette entrée contient la valeur initiale de la variable. Un contrôle est effectué sur la validité de la valeur en fonction du type de donnée de la variable.

NOTE :

- La première entrée doit exactement correspondre à la chaîne de caractères `INITIALVALUE`, en majuscules, pour identifier ce que contient la ligne. Ce mot-clé est le même pour toutes les versions linguistiques de Unity Application Generator.
- Les numéros de ce tableau ne sont donnés que pour améliorer l'aperçu, ils ne font **pas** partie du fichier CSV.

Exemple de ligne

```
INITIALVALUE;Boiler1_Motor1_FTR;CONTROLMODULE;\Zone1\Cellule1\Unité1\Chaudière1\Moteur1;0
```

Format d'exportation des bus de terrain

Introduction

Dans UAG, les fichiers CSV utilisés pour l'importation et l'exportation des modules matériels, des esclaves de bus de terrain, des modules de bus de terrain et des points E/S des bus de terrain, doivent avoir les formats suivants.

Format d'exportation pour les modules matériels

La ligne d'un fichier CSV d'importation/exportation d'un module matériel contient les éléments suivants, délimités par un séparateur (* = champ obligatoire).

1	* `HWMODULE`	2	* ***hwmodule_name***	3	* ***plc_name***
4	* ***link_type***	5	* ***drop_number***	6	***extension_rack_number***
7	* ***slot_number***	8	***mbp_link***	9	***network_segment_1***
10	***network_address_1***	11	***network_segment_2***	12	***network_address_2***
13	***0x_address***	14	***1x_address***	15	***3x_address***
16	***4x_address***	17	***timeout_behavior***	18	***bus_children***
19	***configuration_file***	20	***last_import_timestamp***	21	***original_type***

Description des entrées :

- ***HWMODULE*** : mot-clé.
- ***hwmodule_name*** : nom d'un module matériel valide, par exemple `TSX PBY 100` (module Profibus-DP pour automate Premium). 32 caractères.
- ***plc_name*** : API dans lequel se trouve le module matériel. 8 caractères.
- ***link_type*** : type de châssis. Les valeurs valides sont les suivantes : `Local`, `Local Momentum`, `XBus`, `Remote`, `DIO0`, `DIO1`, `DIO2`, `SYMAX`, `800IO`, `Generic`, `IOBUS`, `MBPIOLink0`, `MBPIOLink1`, `MBPIOLink2`, `ETHIO`.
- ***drop_number*** : numéro de station du châssis correspondant.
- ***extension_rack_number*** : numéro du châssis d'extension. Chaîne vide pour le châssis local et numéro pour les châssis d'extension.
- ***slot_number*** : numéro de l'emplacement du module matériel.
- ***mbp_link*** : type de liaison du module Modbus Plus. Ne sert que pour le NOM-2xx-00, sinon vide. Les valeurs valides sont `1` et `2`.
- ***network_segment_1/_2*** : nom du segment de réseau. Ce champ n'est valide que pour les modules CPS ou de communication, sinon il doit être vide. 32 caractères.
- ***network_address_1/_2*** : adresse sur le réseau du module matériel. Ce champ n'est valide que pour les modules CPS ou de communication, sinon il doit être vide. 64 caractères.
- ***0x_address*** : adresse 0x de début de mémoire d'état du module. Ce champ peut être vide. L'adresse est exportée, mais ignorée pendant l'importation (recalculée par Unity Application Generator).

- ***1x_address*** : adresse 1x de début de mémoire d'état du module. Ce champ peut être vide. L'adresse est exportée, mais ignorée pendant l'importation (recalculée par Unity Application Generator).
- ***3x_address*** : adresse 3x de début de mémoire d'état du module. Ce champ peut être vide. L'adresse est exportée, mais ignorée pendant l'importation (recalculée par Unity Application Generator).
- ***4x_address*** : adresse 4x de début de mémoire d'état du module. Ce champ peut être vide. L'adresse est exportée, mais ignorée pendant l'importation (recalculée par Unity Application Generator).
- ***timeout_behavior*** : 1 des 3 chaînes suivantes : `Non activé`, `Dernière valeur`, `Défini par l'utilisateur`. Ce champ n'est valide que pour les modules de sortie TOR, sinon il doit être vide.
- ***bus_children*** : seulement pour le module Momentum 170-BNO-6x1-00, nombre d'enfants bus local/déporté ; sinon, il doit être vide.
- ***configuration_file*** : chemin d'accès absolu du fichier à partir duquel la configuration de bus de terrain doit être importée. Seulement pour les modules maîtres de bus de terrain. 255 caractères.
- ***last_import_timestamp*** : date et heure (dans UTC) de la dernière importation de la configuration de bus de terrain (format : aaaa-mm-jj hh:mm:ss). Seulement pour les modules maîtres de bus de terrain.
- ***original_type*** : type d'origine importé à partir de la configuration de bus de terrain. Seulement pour les modules maîtres de bus de terrain. 32 caractères.

Format d'exportation pour les esclaves de bus de terrain

La ligne d'un fichier CSV d'importation/exportation d'un esclave de bus de terrain contient les éléments suivants, délimités par un séparateur (* = champ obligatoire).

1	`* FBSLAVE`	2	* ***rack_name***	3	* ***rack__type_name***
4	* ***drop_number*** ou ***master_gateway_number***	5	* ***plc_name***	6	* ***network_segment***
7	* ***network_address***				

Description des entrées :

- ***FBSLAVE*** : mot-clé.
- ***rack_name*** : nom d'un esclave de bus de terrain valide. 50 caractères.
- ***rack__type_name*** : nom du type d'esclave défini dans la personnalisation ou dans le catalogue de modules. 50 caractères.
- ***drop_number*** : numéro de station (si un esclave est présent sous le module maître PTQ) ou
- ***master_gateway_number*** : nom de la passerelle maître (si un esclave est présent sous la passerelle maître de bus de terrain)
- ***plc_name*** : nom de l'API parent du module maître de bus de terrain. 8 caractères.

- ***network_segment*** : nom du segment de réseau du module maître de bus de terrain (celui exporté avec le module maître). Il peut s'agir soit de ***network_segment_1***, soit de ***network_segment_2***, mais il doit avoir un bus de terrain de type réseau. 32 caractères.
- ***network_address*** : adresse de l'esclave dans le segment de réseau ci-dessus. 64 caractères.

Format d'exportation pour les modules de bus de terrain

La ligne d'un fichier CSV d'importation/exportation d'un module de bus de terrain contient les éléments suivants, délimités par un séparateur (* = champ obligatoire).

1	`* FBMODULE`	2	* ***hwmodule_name***	3	* ***plc_name***
4	* ***drop_number*** ou ***master_gateway_number***	5	* ***rack_name***	6	* ***network_segment_parent***
7	* ***network_address_parent***	8	* ***slot_number***	9	* ***address_pattern_input***
10	* ***address_pattern_output***	11	* ***addressing_scheme***	12	* ***num_bytes_input***
13	* ***num_bytes_output***	14	* ***byte_offset_input***	15	* ***byte_offset_output***
16	* ***conversion_fb_input***	17	* ***conversion_fb_output***	-	-

Description des entrées :

- ***FBMODULE*** : mot-clé.
- ***hwmodule_name*** : nom d'un module de bus de terrain valide. 32 caractères.
- ***plc_name*** : nom de l'API parent du module maître de bus de terrain. 8 caractères.
- ***drop_number*** : numéro de station (si un esclave est présent sous le module maître PTQ) ou
- ***master_gateway_number*** : nom de la passerelle maître (au cas où elle serait présente sous la passerelle maître MNET)
- ***rack_name*** : nom d'un esclave de bus de terrain du module. 50 caractères.
- ***network_segment_parent*** : nom du segment de réseau du module maître de bus de terrain. 32 caractères.
- ***network_address_parent*** : adresse de l'esclave dans le segment de réseau ci-dessus. 64 caractères.
- ***slot_number*** : numéro de l'emplacement du module à l'intérieur de l'esclave de bus de terrain.
- ***address_pattern_input*** : celle définie par la configuration du bus de terrain. 100 caractères.
- ***address_pattern_ouput*** : celle définie par la configuration du bus de terrain. 100 caractères.
- ***addressing_scheme*** : soit `bytewise` (par octet), soit `wordwise` (par mot), selon les impératifs de la configuration du bus de terrain.
- ***num_bytes_input*** : nombre d'octets d'entrée.
- ***num_bytes_output*** : nombre d'octets de sortie.
- ***byte_offset_input*** : décalage d'adresse du premier octet d'entrée.
- ***byte_offset_output*** : décalage d'adresse du premier octet de sortie.
- ***conversion_fb_input*** : nom du bloc fonction de conversion (FB) pour les points d'entrée, ou chaîne vide. 32 caractères.
- ***conversion_fb_output*** : nom du FB de conversion pour les points de sortie, ou chaîne vide. 32 caractères.

Format d'exportation pour les points d'E/S de bus de terrain

La ligne d'un fichier CSV d'importation/exportation d'un point d'E/S de bus de terrain contient les éléments suivants, délimités par un séparateur (* = champ obligatoire).

1	`* FBIOPOINT`	2	***variable_parent_type***	3	***variable_full_path***
4	***variable_full_name***	5	* ***plc_name***	6	* ***rack_number***
7	* ***rack_name***	8	* ***network_segment***	9	* ***network_address***
10	* ***slot_number***	11	* ***inout_type***	12	***position_in_HW_module***
13	***description***	14	***data_type***	15	***byte_order_id***
16	***byte_number*** ou ***word_number***	17	***bit_number***	-	-

Description des entrées :

- ***FBIOPOINT*** : mot-clé.
- ***variable_full_name*** : nom complet (généré) de la variable affectée au point d'E/S, par exemple `EquipmentName_DeviceName_VariableName`.
- ***plc_name*** : nom de l'API parent du module maître de bus de terrain. 8 caractères.
- ***rack_number*** : numéro d'emplacement du module maître de bus de terrain.
- ***rack_name*** : nom d'un esclave de bus de terrain du module. 50 caractères.
- ***network_segment*** : nom du segment de réseau du module maître de bus de terrain. 32 caractères.
- ***network_address*** : adresse de l'esclave dans le segment de réseau ci-dessus. 64 caractères.
- ***slot_number*** : numéro de l'emplacement du module à l'intérieur de l'esclave de bus de terrain.
- ***inout_type*** : soit `IN`, soit `OUT`.
- ***position_in_HW_module*** : position dans le module E/S . Dans un module panaché avec connexions d'ENTREE et de SORTIE, les connexions d'entrée sont comptées les premières, par exemple sur un module à 4 connexions d'entrée, la première connexion de sortie est 5.
- ***description*** : nom unique du point d'E/S. 64 caractères.
- ***data_type*** : type de données du point d'E/S. Soit 1 des types de données du système (`BYTE`, `BOOL`, etc.), soit un type de chaîne de longueur fixe (par exemple `String[10]`, selon la définition dans une des bibliothèques SCoD utilisées.
- ***byte_order_id*** : soit `Intel`, soit `Motorola` si le type de données est l'un des types suivants : `INT`, `UINT`, `DINT`, `UDINT`, `WORD`, `DWORD`. Vide pour tous les autres types de données ou si un bloc fonction est utilisé pour la conversion entre la représentation pour le bus de terrain et les variables d'API.
- ***byte_number*** : nombre binaire commençant par un 1 du point d'E/S à l'intérieur de l'espace adresse du module. Vide si un bloc fonction est utilisé pour la conversion entre la représentation pour le bus de terrain et les variables d'API.
- ***bit_number*** : nombre de bits 1 à l'intérieur de l'octet indiqué par ***byte_number*** si le type de données est `BOOL`. -1 pour tous les autre types de données. Vide si un bloc fonction est utilisé pour la conversion entre la représentation pour le bus de terrain et les variables d'API.

Bibliothèque des types de modèles : format de fichier CSV

Remarques générales

Le format des modules de commande, des variables, des liaisons et des forçages de la bibliothèque des types de modèles est le même que celui des modules de commande, des liaisons et des forçages du modèle physique.

Références des documents

Pour importer dans UAG un fichier préalablement exporté, contenant des attributs `document_ref`, vous devez tout d'abord supprimer les attributs `document_ref` du fichier exporté.

Type de modèle

La ligne d'un fichier CSV d'importation/exportation d'un type de modèle contient les éléments suivants, délimités par un séparateur (* = champs obligatoires).

1	* `TEMPLATETYPE`	2	* **name**	3	* **description**
4	* **hmi_symbol_config_type**	5	**help_file**	-	-

Symbole IHM

La ligne d'un fichier CSV d'importation/exportation d'un symbole IHM contient les éléments suivants, délimités par un séparateur (* = champ obligatoire).

1	* `HMISYMBOL`	2	* **name**	3	* **template_type_name**
4	**is_default**	5	**hmi_symbol_type**	-	-

Module de commande des types de modèles

La ligne d'un fichier CSV d'importation/exportation d'un module de commande des types de modèles contient les éléments suivants, délimités par un séparateur (* = champs obligatoires mais peuvent être vierges).

1	* CONTROLMODULE	2	* ***name***	3	**** full_path***
4	* ***SCoD_type_name***	5	***library_name***	6	* ***description***
7	* ***plc_name***	8	* ***device_master_name***	9	* ***comment***
10	* ***separator_line_information***	11	* ***hmi_symbol_type***	12	* ***hmi_symbol_name***
13	***document_ref_1***	14	***document_parameters_1***	15	***document_ref_i***
16	***document_parameters_i***	17	***document_ref_n***	18	***document_parameters_n***

- ***plc_name*** : Cette chaîne doit être vide. Il n'existe pas d'API pour un module de commande des types de modèles. Même si ce champ contient une valeur dans un fichier CSV, il ne sera pas importé.

Variable de type de modèle

La ligne d'un fichier CSV d'importation/exportation d'une variable de type de modèle contient les éléments suivants, délimités par un séparateur (* = champs obligatoires mais peuvent être vierges).

1	* VARIABLE	2	* ***name***	3	* ***parent_type*** Les valeurs autorisées sont TEMPLATETYPE (par exemple Equipement) ou CONTROLMODULE
4	* **head_variable_parent_path**	5	* ***parent_name***	6	* ***description***
7	* ***data_type***	8	* ***connection_type_name***	9	* ***in_out_type_name***
10	***address_segment***	11	***address_offset***	12	***initial_value***
13	***outputFailure_behavior***	14	***outputFailure_value***	15	***command_name***
16	***inverse***	17	***variable_used***	18	***graphical_symbol***
19	***graphical_symbol_name***	20	***display_format***	21	***measurement_unit***
22	***scaling_min***	23	***scaling_max***	24	***value_boundary_min***
25	***value_boundary_max***	26	***access_level***	27	***state0_text***
28	***state1_text***	29	***alarm_value***	30	***alarm_priority***
31	***alarm_operator_id***	32	**alarm_limit**	33	***alarm_text***
34	**archive**	35	**archive_file**	36	**communication_frame**
37	**get_free_property**	38	***alarm_group_name***	39	***update_initial_value***
40	***SCoD_variable_id***	41	***deadband_mode***	42	***deadband_value***
43	***raw_min***	44	***raw_max***	45	***archive_sample_rate***

- ***head_variable_parent_path*** : chemin parent de tête de la variable, c.-à-d. jusqu'à TEMPLATETYPE/CONTROLMODULE.

Liaison de type de modèle

La ligne d'un fichier CSV d'importation/exportation d'une liaison de type de modèle contient les éléments suivants, délimités par un séparateur (* = champs obligatoires).

1	* LINK	2	* **controlmodule_full_name**	3	* **controlmodule_pin_name**
4	* **variable_parent_type**	5	* **variable_full_path**	6	* **variable_full_name**
7	* **description**	8	* **controlmodule_full_path**	-	-

- **variable_parent_type** : TEMPLATETYPE/CONTROLMODULE
- **controlmodule_full_path** : chemin du module de commande possédant des liaisons (par ex. TempLibrary1\TempType1\T_CM1). Le nom de la bibliothèque ainsi que les éléments parents sont extraits du chemin.

Forçage de type de modèle

La ligne d'un fichier CSV d'importation/exportation d'un forçage de type de modèle contient les éléments suivants, délimités par un séparateur (* = champs obligatoires).

1	* INTERLOCK	2	* **controlmodule_name**	3	* **controlmodule_pin_name**
4	* **interlock_definition**	5	* **comment**	6	* **controlmodule_full_path**

Propriété de type de modèle

La ligne d'un fichier CSV d'importation/exportation d'une propriété de type de modèle contient les éléments suivants, délimités par un séparateur (* = champs obligatoires).

1	* TEMPLATETYPEPROPERTY	2	* **property_name**	3	* **variable_full_name**
4	* **variable_full_path**	5	* **variable_property_name**	6	* **template_type_name**
7	* **tab_name**	8	* **tab_order**	9	* **tab_position**
10	**controlmodule_name**	11	**controlmodule_property_name**	-	-

Suppression et changement de nom : Format de fichier CSV

Introduction

Au format CSV, le changement de nom et la suppression d'objet sont possibles en utilisant les mots-clés `DELETE` et `RENAME`.

Veuillez consulter Importation incrémentielle *(voir page 406)*.

Supprimer

La ligne d'un fichier CSV de suppression d'un objet contient les éléments suivants, délimités par un séparateur (* = champs obligatoires).

1	* `DELETE`	2	* **instance_name**	3	* **object_type**
4	**parent_path**	5	**parent_type**	6	**variable_full_name**
7	**pin_name**	8	**pin_variable_parent_path**	9	**plc_name**
10	**drop_number**	11	**link_type**	12	**extension_number**
13	**slot_number**	14	**network_node_type**	15	**hardware_module_type**
16	**sender_network_segment**	17	**receiver_network_segment**	18	**network_path**

Exemple :

```
DELETE;Var1;VARIABLE;Ar01\Pc01\Un01\Eq01\Cm1;CONTROMODULE;Eq01_Cm1_Var1
;
```

Renommer

La ligne d'un fichier CSV de changement de nom d'un objet contient les éléments suivants, délimités par un séparateur (* = champs obligatoires).

1	* `RENAME`	2	* **instance_name**	3	* **object_type**
4	**parent_path**	5	**parent_type**	6	**variable_full_name**
7	* **new_name**	8	**pin_name**	9	**plc_name**
10	**drop_number**	11	**link_type**	12	**extension_number**
13	**slot_number**	14	**network_node_type**	15	**-**

Eléments requis

Eléments requis pour le changement de nom et la suppression

Type d'objet	Requis pour le changement de nom	Requis pour la suppression
Elément physique	instance_name, object_type, parent_path (jusqu'à parent), new_name	instance_name, object_type, parent_path (jusqu'à parent)
Module de commande	instance_name, object_type, parent_path (jusqu'à parent), new_name	instance_name, object_type, parent_path (jusqu'à parent)
Variable	instance_name, object_type, parent_path (jusqu'à parent), parent_type, variable_full_name, new_name	instance_name, object_type, parent_path (jusqu'à parent), parent_type, variable_full_name
Instrument	instance_name, object_type, new_name	instance_name, object_type
Liaison	s/o	object_type, parent_path, variable_full_name, pin_name, pin_variable_parent_path
Forçage	s/o	object_type, parent_path, pin_name
Segment de réseau	instance_name, object_type, new_name	instance_name, object_type
IHM	instance_name, object_type, new_name	instance_name, object_type
Serveur de données	instance_name, object_type, new_name	instance_name, object_type
Zone de conduite	instance_name, object_type, parent_path (jusqu'à parent), new_name	instance_name, object_type, parent_path (jusqu'à parent)
Groupe d'images	instance_name, object_type, parent_path (jusqu'à parent c.-à-d. hmi_name), new_name	instance_name, object_type, parent_path (jusqu'à parent c.-à-d. hmi_name)
Image	instance_name, object_type, parent_path (jusqu'à parent), new_name	instance_name, object_type, parent_path (jusqu'à parent)
API	instance_name, object_type, new_name	instance_name, object_type

Type d'objet	Requis pour le changement de nom	Requis pour la suppression
Châssis	instance_name, object_type, new_name, plc_name, drop_number, link_type, extension_number **NOTE :** Le changement de nom est uniquement possible pour Advantys et MEG.	instance_name (MEG* et Advantys*), object_type, plc_name, drop_number, link_type, extension_number
Module matériel	s/o	object_type, plc_name, drop_number (châssis), link_type (châssis), extension_number (châssis), slot_number, network_node_type (nom de châssis pour MEG), hardware_module_type
Abonné réseau (NetPartner)	instance_name, object_type, new_name, network_node_type.	instance_name, object_type, network_node_type.
Canal	instance_name, object_type, parent_path (jusqu'à parent), new_name	instance_name, object_type, parent_path (jusqu'à parent)
Chemin de routage	s/o	object_type, sender_network_segment, receiver_network_segment, network_path
Type de modèle	instance_name, object_type, new_name	instance_name, object_type
Propriété de type de modèle	instance_name, object_type, parent_path (c.-à-d. le nom du type de modèle), new_name	instance_name, object_type, parent_path (c.-à-d. le nom du type de modèle)
s/o = sans objet * = les informations sur les éléments requis ne doivent pas rester vides. Les éléments non requis présents avant les éléments requis au format de ligne CSV peuvent être vides.		

Bibliothèque SCoD : format de fichier CSV

Introduction

L’exportation de SCoD ou de bibliothèques SCoD depuis l’éditeur SCoD n’est pas prise en charge par le format CSV.

UAG prend en charge l’exportation de SCoD depuis l’éditeur SCoD uniquement au format XML.

Annexe B
Format des fichiers XML d’importation et d’exportation

Introduction

Ce chapitre donne le format des fichiers XML pour les processus d’importation et d’exportation.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
B.1	Informations générales sur le format XML	484
B.2	Format XML pour un projet UAG	485
B.3	Format XML pour des bibliothèques de modèles	502
B.4	Format XML pour l’éditeur SCoD	503

Sous-chapitre B.1
Informations générales sur le format XML

Informations générales sur le format XML

Présentation

UAG prend en charge un format XML hiérarchique.

Les détails du format XML et de sa structure sont expliqués dans les sections suivantes.

Définition formelle du format sous schéma XML

Pour obtenir une définition formelle du format sous schéma XML, veuillez consulter les fichiers suivants du répertoire d'installation de UAG :

- *ProjectExchangeFile.xsd* (pour les projets UAG)
- *ControlModuleTypeExchangeFile.xsd* (pour les bibliothèques SCoD)
- *TemplateLibraryExchangeFile.xsd* (pour les bibliothèques de modèles)

Création, modification, changement de nom et suppression d'objets via Importer

En général, vous pouvez créer, modifier, renommer ou supprimer des objets qui peuvent être également créés, modifiés, renommés ou supprimés à partir de l'interface utilisateur.

Les règles de suppression et de changement de nom d'objets sont décrites au paragrapheRègles de suppression et de changement de nom d'objets *(voir page 501)*.

NOTE : La fonction d'importation pour les bibliothèques SCoD n'est pas disponible avec UAG 3.3 SP4.

Attributs en option

NOTE : Les attributs en option qui sont vides ne sont pas exportés.

Options de l'analyseur pour l'importation

Les options d'analyse du projet servent également à l'importation pour signaler des alertes ou arrêter le système après la détection d'un certain nombre d'erreurs ou d'alertes :

- `Afficher avertissement`
- `Arrêter après n erreurs`
- `Arrêter après n avertissements`

Sous-chapitre B.2
Format XML pour un projet UAG

Principales sections d'un fichier XML

Le format XML propose 3 sections principales :
- En-tête
- Modèle physique
- Modèle topologique
- Liste d'instruments

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Format XML des données d'en-tête	486
Format XML des éléments physiques	487
Format XML des éléments topologiques	492
Format XML des listes d'instruments	500
Règles de suppression et de changement de nom d'objets	501

Format XML des données d'en-tête

Présentation

Les données d'en-tête fournissent des informations sur l'origine du fichier XML comprenant :

- nom de la société
- nom du produit
- version du produit
- version de schéma
- date et heure
- contenu
- fichier du projet

Format des données d'en-tête

```
<?xml version="1.0" encoding="utf-8" ?>
- <ProjectExchangeFile>
    <fileHeader company="Schneider Electric" product="Unity Application Generator"
      productVersion="3.2" schemaVersion="1.0" dateTime="09/24/2008 10:13:31 +0530"
      content="Project source file" />
    <contentHeader projectFile="D:\snapshot\AC_BA_UAG_3.1\UAG\dev\Run\chcheck2.osp" />
  + <topologicalModel>
  + <physicalModel>
  + <instrumentList>
  </ProjectExchangeFile>
```

Format XML des éléments physiques

Présentation

L'élément `physicalModel` est un élément enfant de l'élément racine contenant des éléments pour les éléments physiques.

Les informations sur les références de documents sont placées comme un élément séparé `documents` sous chaque niveau physique ainsi que sous chaque module de commande.

L'attribut `importAction` est disponible pour la plupart des éléments afin de réaliser le changement de nom et la suppression (voir le fichier schéma XML installé *ProjectExchangeFile.xsd* dans le répertoire d'installation UAG pour plus d'informations).

NOTE : Les attributs `fullName`, `variableNamePrefix`, `variableNameSuffix` sont exportés à titre d'information et sont ignorés lors de l'importation.

Eléments physiques et modules de commande

L'élément `physicalElements` est un élément enfant de `physicalModel` ou de `physicalElement`.

L'élément `controlModules` est un élément enfant de `physicalElement` (ou de `instrumentList`, voir ci-dessous).

Les éléments `physicalElements` et `controlModules` ont la structure suivante :

```
- <physicalModel>
  - <physicalElements>
    - <physicalElement name="ar01" levelName="Area" importAction="" description="area">
        <comment>some text</comment>
      - <documents>
          <document type="exe" name="C:\FolderExist.exe" parameter="Folder file" />
        </documents>
      - <physicalElements>
        - <physicalElement name="pc01" levelName="ProcessCell" importAction="" description="process cell">
            <comment>some text</comment>
          - <documents>
              <document type="exe" name="C:\FolderExist.exe" parameter="Folder file" />
            </documents>
          - <physicalElements>
            - <physicalElement name="pc01" levelName="Unit" importAction="" description="unit">
                <comment>some text</comment>
              - <documents>
                  <document type="exe" name="C:\FolderExist.exe" parameter="Folder file" />
                </documents>
              - <physicalElements>
                - <physicalElement name="Eq01" fullName="Eq01" levelName="Equipment" importAction=""
                    variableNamePrefix="EQ01_" variableNameSuffix="" hmi="hmi1" description="eq m" plc="PLC1"
                    controlDomain="cd1" section="Eq01" picture="pic1" templateType="" templateLibrary=""
                    hmiSymbolConfiguration="" hmiSymbolName="" hmiSymbolType="">
                    <comment>some text</comment>
                  + <documents>
                  - <controlModules>
                    - <controlModule name="cm3" fullName="Eq01_cm3" importAction="" type="Free"
                        library="UAGBasic_Citect" description="cm3" plc="PLC1" variableNamePrefix="Eq01_cm3_"
                        variableNameSuffix="" controlModuleMaster="" separatorLine="1" hmiSymbolType="MonitorPro"
                        hmiSymbolName="qwert" templateTypeControlModule="">
                        <comment>some text</comment>
                      + <documents>
                      + <variables>
                      + <interlocks>
                      + <links>
                      </controlModule>
                    </controlModules>
                  </physicalElement>
                </physicalElements>
              </physicalElement>
            </physicalElements>
          </physicalElement>
        </physicalElements>
      </physicalElement>
    </physicalElements>
  </physicalModel>
```

Variables

L'élément `variables` est un élément enfant de `physicalElement`, `controlModule`ou `variable`.

Les attributs de l'élément variable `fullName`, `addressSegment`, `addressOffset` et `addressBitOffset` sont exportés à titre d'information et sont ignorés lors de l'importation.

NOTE : Les DDT et leurs éléments DDT sont représentés de façon hiérarchique lorsqu'un élément `variable` contient un élément `variables` contenant les éléments variables pour le sous-niveau suivant.

L'attribut `name` des éléments DDT commence par un `"."` pour des éléments de structure ou le nom est mis entre crochets pour des éléments de tableau.

Exemple :

`<variable name=".StructMember"` ou

`<variable name="[1]"`

L'attribut `fullName` qui est présent est similaire à d'autres variables. Les variables représentant des éléments DDT ne peuvent pas être renommées ou supprimées (de même pour les variables SCoD).

NOTE : Les énumérations utilisées pour `InitialValue` et `FreeProperty` sont uniquement exportées au format XML, en aucun cas au format CSV.

L'élément `variables` possède la structure suivante :

```
- <variables>
  - <variable name="Var1" fullName="Eq01_cm3_var1" importAction="" connectionType="PLC"
      dataType="bool" inOut="IN" description="swd">
    - <variableProperties>
        <variableProperty name="InitialValue" value="" enumeration="1" />
        <variableProperty name="OutputFailureBehavior" value="Last value" />
        <variableProperty name="VariableUsed" value="True" />
        <variableProperty name="GraphicalSymbol" value="False" />
        <variableProperty name="UpdateInitialValue" value="False" />
        <variableProperty name="DeadBandMode" value="Percentage" />
        <variableProperty name="Command" value="Logic" />
        <variableProperty name="Inverse" value="True" />
        <variableProperty name="AlarmPriority" index="1" value="1" />
        <variableProperty name="AlarmOperator" index="1" value="<" />
        <variableProperty name="AlarmLimit" index="1" value="95" />
        <variableProperty name="AlarmText" index="1" value="alarm occurred" />
        <variableProperty name="AlarmPriority" index="2" value="1" />
        <variableProperty name="AlarmOperator" index="2" value="<" />
        <variableProperty name="AlarmLimit" index="2" value="95" />
        <variableProperty name="AlarmText" index="2" value="alarm occurred" />
        <variableProperty name="FreeProperty" index="19" value="" enumeration="1" />
        <variableProperty name="FreeProperty" index="20" value="free" />
      </variableProperties>
    </variable>
  </variables>
```

Forçages

L'élément `interlocks` est un élément enfant de `controlModule`.

L'élément `interlocks` possède la structure suivante :

```
- <interlocks>
  - <interlock importAction="" pin="Lock" interlockDefinition="$1">
      <comment>some text</comment>
    - <interlockConditions>
        <interlockCondition index="1" operand1="Eq01_cm3_var1" operand1ParentPath="ar01\pc01\un01\Eq01
          \cm2" operand1Type="Variable" operator="=" operand2="Eq01_cm3_var1" operand2ParentPath="ar01
          \pc01\un01\Eq01\cm2" operand2Type="Variable" description="" />
      </interlockConditions>
    </interlock>
  </interlocks>
```

Liaisons

L'élément `links` est un élément enfant de `controlModule`.

L'élément `links` possède la structure suivante :

```
- <links>
    <link importAction="" pin="Err" variableParentPath="ar01\pc01\un01\Eq02\cm3"
      variableFullName="Eq02_cm3_Err" description="Lock - Description" />
  </links>
```

Concernant la broche de liaison DDT :

1. Si la broche de liaison DDT n'est pas connectée, et qu'aucun des éléments DDT n'est connecté, la liaison est exportée en tant que liaison normale (identique à la structure précédente).
2. Si la broche de liaison DDT est connectée avec, ne serait-ce qu'un seul élément DDT connecté, la liaison est exportée conformément à la structure suivante.

```
<link pin="Input_Link_Data" variableParentPath="" variableFullName="">
  <links>
    <link pin="Bool1" variableParentPath="" variableFullName="" />
    <link pin="Bool2" variableParentPath=
    "\Area\PCell\Unit\EQ01\NormalCM" variableFullName=
    "EQ01_NormalCM_Auto" description="Auto mode active - Normal CM -
    EQ01" />
    <link pin="Real1" variableParentPath="" variableFullName="" />
    <link pin="Byte1" variableParentPath="" variableFullName="" />
    <link pin="Int1" variableParentPath="" variableFullName="" />
  </links>
</link>
```

Format XML des éléments topologiques

Présentation

Les données des éléments topologiques fournissent des informations sur :

- `networkSegments`
- `routingPaths`
- `dataServers`
- `hmis`
- `plcs`
- `networkNodes`

Ces éléments sont des éléments enfants directs de l'élément `topologicalModel` du fichier XML.

L'attribut `importAction` est disponible pour la plupart des éléments afin de réaliser le changement de nom et la suppression (voir le fichier schéma XML installé *ProjectExchangeFile.xsd* dans le répertoire d'installation UAG pour plus d'informations).

Segments de réseau

L'élément `networkSegments` est un élément enfant de `topologicalModel`.

L'élément `networkSegments` possède la structure suivante.

```
- <networkSegments>
    <networkSegment name="eth1" importAction="" networkType="Ethernet" subnetmask="255.255.255.0"
      defaultGateway="111.1.11.1" />
  </networkSegments>
```

Chemins de routage

L'élément `routingPaths` est un élément enfant de `topologicalModel`.

L'élément `routingPaths` possède la structure suivante.

```
- <routingPaths>
    <routingPath importAction="" senderNetworkSegment="mbp1" receiverNetworkSegment="mbp2"
      path="2" />
  </routingPaths>
```

Serveurs de données

L'élément `dataServers` est un élément enfant de `topologicalModel`.

L'élément `dataServers` possède la structure suivante.

```
- <dataServers>
  - <dataServer name="ds3" timeoutMin="0" timeoutSec="1" timeoutMs="0">
    - <extendedProperties>
        <extendedProperty name="Alarm Server" value="DS3_AS" />
        <extendedProperty name="Cluster Name" value="DS3" />
        <extendedProperty name="Report Server" value="DS3_RS" />
        <extendedProperty name="Trend Server" value="DS3_TS" />
      </extendedProperties>
    - <primaryServer>
      - <networkConnections>
          <networkConnection index="1" networkSegment="eth1" address="155.155.155.1" />
          <networkConnection index="2" networkSegment="eth1" address="155.155.155.2" />
        </networkConnections>
      </primaryServer>
    - <secondaryServer>
      - <networkConnections>
          <networkConnection index="1" networkSegment="eth1" address="155.155.155.3" />
          <networkConnection index="2" networkSegment="eth1" address="155.155.155.4" />
        </networkConnections>
      </secondaryServer>
      <comment>some text</comment>
    - <documents>
        <document type="exe" name="C:\FolderExists.exe" parameter="Folder file" />
      </documents>
    </dataServer>
  </dataServers>
```

IHM, zone de conduite, groupe d'images et image

L'élément `hmis` est un élément enfant de `topologicalModel`.

L'élément `hmis` possède la structure suivante.

```
- <hmis>
  - <hmi name="hmi1" importAction="" type="VijeoCitect">
    - <documents>
        <document type="exe" name="C:\FolderExist.exe" parameter="Folder file" />
      </documents>
      <comment>some text</comment>
    - <controlDomains>
      - <controlDomain name="cd1" importAction="" description="cd">
          <comment>some text</comment>
        </controlDomain>
      </controlDomains>
    - <pictureGroups>
      - <pictureGroup name="picg1" importAction="" description="group1">
        - <pictures>
            <picture name="pic1" importAction="" description="picture1" />
            <picture name="pic2" importAction="" description="picture2" />
          </pictures>
        - <pictureGroups>
            <pictureGroup name="picg2" importAction="" description="group2" />
          </pictureGroups>
        </pictureGroup>
      </pictureGroups>
    </hmi>
  </hmis>
```

API, châssis, module matériel et point d'E/S

L'élément `plcs` est un élément enfant de `topologicalModel`.

L'élément `plcs` possède la structure suivante.

```
- <plcs>
  - <plc name="m3401" importAction="" description="sadasd" projectName="m3401"
      plcFamily="Unity/M340">
      <comment>some text</comment>
    + <addressRanges>
    + <documents>
    - <racks>
      - <rack name="" importAction="" type="BMX XBP 0800" description="rack" dropNumber="0"
          partNumber="" linkType="LocalM340" extensionRackNumber="1" configurationFilePath=""
          isShared="" fieldbusMasterGateway="">
        - <hwModules>
          - <hwModule type="XYZ" moduleCategory="" slotNumber="" mbplink="" timeoutBehavior=""
              busChildren="" ioScannerRepetitiveRate="" version="">
            - <addressRanges>
                <addressRange type="ReadWriteAnalog" usage="inOut" startAddress="218"
                  length="8" />
              </addressRanges>
            - <networkConnections>
                <networkConnection index="1" networkSegment="eth1" address="111.0.0.35" />
                <networkConnection index="2" networkSegment="eth1" address="111.0.0.36" />
              </networkConnections>
            - <ioPoints>
                <ioPoint type="IN" position="2" variableParentPath="ar01\pc01\un01\Eq02\cm3"
                  variableFullName="Eq02_cm3_MFwd" />
              </ioPoints>
            </hwModule>
          </hwModules>
        </rack>
      + <rack name="Device_Test(1)" importAction="" partNumber="Advantys STB" desciption="descr"
          dropNumber="3" linkType="DeviceMBP0" extensionRackNumber="" configurationFilePath="">
      + <rack name="" importAction="" desciption="descr" dropNumber="3" linkType="FBSlave">
      </racks>
    - <channels>
      + <channel name="m3401_ds3" importAction="" communicationType="DataServer"
          communicationPartner="ds3">
      + <channel name="ch_pcl" importAction="" communicationType="PLC" communicationPartner="plc2">
      + <channel name="ch_pcl" importAction="" communicationType="NetPartner"
          communicationPartner="netp2">
      </channels>
    </plc>
  </plcs>
```

L'élément `addressRanges` est un élément enfant de l'élément `plcs`.

L'élément `addressRanges` possède la structure suivante :

```
- <addressRanges>
    <addressRange type="ReadWriteDigital" usage="HWModules" startAddress="-1" length="1"
      actualLength="0" />
    <addressRange type="ReadWriteDigital" usage="HMI" startAddress="1" length="1024"
      actualLength="0" />
    <addressRange type="ReadWriteDigital" usage="NetPartner" startAddress="1025" length="1024"
      actualLength="0" />
    <addressRange type="ReadWriteDigital" usage="Reserve" startAddress="2049" length="1024"
      actualLength="1024" />
    <addressRange type="ReadDigital" usage="HWModules" startAddress="-1" length="1"
      actualLength="0" />
    <addressRange type="ReadDigital" usage="Reserve" startAddress="-1" length="1"
      actualLength="1" />
    <addressRange type="ReadAnalog" usage="HWModules" startAddress="-1" length="1"
      actualLength="0" />
    <addressRange type="ReadAnalog" usage="Reserve" startAddress="-1" length="1"
      actualLength="1" />
    <addressRange type="ReadWriteAnalog" usage="HWModules" startAddress="-1" length="1"
      actualLength="0" />
    <addressRange type="ReadWriteAnalog" usage="HMI" startAddress="2" length="1016"
      actualLength="0" />
    <addressRange type="ReadWriteAnalog" usage="NetPartner" startAddress="1024" length="1024"
      actualLength="0" />
    <addressRange type="ReadWriteAnalog" usage="PLCCommunication" startAddress="2048"
      length="1024" actualLength="0" />
    <addressRange type="ReadWriteAnalog" usage="Reserve" startAddress="3073" length="1024"
      actualLength="1024" />
  </addressRanges>
```

Châssis, module matériel et point d'E/S pour Advantys et Profibus

Le format de Advantys et Profibus est similaire aux `rack`, `hwModule` et `ioPoint` normaux, si ce n'est qu'ils fournissent des informations supplémentaires.

Advantys et Profibus sont identifiés à partir de l'attribut `linkType`.

NOTE : Les informations de Advantys et Profibus ne peuvent pas être importées dans UAG 3.3 SP4.

Au cours de l'importation, seuls des points d'E/S de modules existants sont mis à jour, par exemple l'attribution variable d'un point d'E/S.

Lors de l'importation, si une configuration Profibus ou Advantys n'est pas trouvée dans le projet UAG et si l'information est trouvée dans le fichier exporté, un message s'affiche dans la fenêtre des messages de sortie.

```
- <rack name="Device_Test(1)" importAction="" partNumber="Advantys STB" desciption="descr"
    dropNumber="3" linkType="DeviceMBP0" extensionRackNumber="" configurationFilePath="">
  - <hwModules>
    - <hwModule type="" moduleCategory="" slotNumber="">
      - <networkConnections>
          <networkConnection index="1" networkSegment="eth1" address="111.0.0.37" />
        </networkConnections>
      - <ioPoints>
          <ioPoint type="IN" position="2" variableParentPath="ar01\pc01\un01\Eq02\cm3"
            variableFullName="Eq02_cm3_MFwd" desciption="" dataType="" bitPosition="" />
        </ioPoints>
      </hwModule>
    </hwModules>
  </rack>
- <rack name="" importAction="" desciption="descr" dropNumber="3" linkType="FBSlave">
  - <hwModules>
    - <hwModule type="" moduleCategory="" slotNumber="" addressPatternInput=""
        addressPatternOutput="" addressingScheme="" bytesInput="" bytesOutput=""
        byteOffsetInput="" byteOffsetOutput="" fbInput="" fbOutput="">
      - <networkConnections>
          <networkConnection index="1" networkSegment="eth1" address="111.0.0.37" />
        </networkConnections>
      - <ioPoints>
          <ioPoint type="IN" position="2" variableParentPath="ar01\pc01\un01\Eq02\cm3"
            variableFullName="Eq02_cm3_MFwd" desciption="" byteOrder="" dataType=""
            bitPosition="" />
        </ioPoints>
      </hwModule>
    </hwModules>
  </rack>
```

Canal

L'élément `channels` est un élément enfant de `plc`.

Les éléments de canal des différents types de communication (`DataServer`, `PLC` et `NetPartner`) ont la structure suivante :

```
- <channels>
  - <channel name="m3401_ds3" importAction="" communicationType="DataServer" communicationPartner="ds3">
    - <communicationPaths>
        <communicationPath index="1" plcNetworkSegment="eth1" plcNetworkAddress="10.180.39.10"
          partnerNetworkSegment="eth2" partnerNetworkAddress="10.180.38.104"
          connectionTimeout="0:2.0" />
        <communicationPath index="2" plcNetworkSegment="eth1" plcNetworkAddress="10.180.39.10"
          partnerNetworkSegment="eth2" partnerNetworkAddress="10.180.38.21" connectionTimeout="0:2.0" />
      </communicationPaths>
    - <hmiCommunicationFrames>
        <hmiCommunicationFrame type="ReadWriteDigital" name="BOOLRD" dataType="BOOL" frameType="RD"
          startAddress="5000" length="1000" />
        <hmiCommunicationFrame type="ReadWriteAnalog" name="R1RD" dataType="REAL" frameType="RD"
          startAddress="2000" length="1000" />
      </hmiCommunicationFrames>
    </channel>
  - <channel name="ch_pcl" importAction="" communicationType="PLC" communicationPartner="plc2">
    - <communicationPaths>
        <communicationPath index="1" plcNetworkSegment="eth1" plcNetworkAddress="10.180.39.10"
          partnerNetworkSegment="eth2" partnerNetworkAddress="10.180.38.104"
          connectionTimeout="0:2.0" />
        <communicationPath index="2" plcNetworkSegment="eth1" plcNetworkAddress="10.180.39.10"
          partnerNetworkSegment="eth2" partnerNetworkAddress="10.180.38.21" connectionTimeout="0:2.0" />
      </communicationPaths>
    - <communicationTables>
      - <communicationTable type="Word" startAddress="4097" length="32" redundantStartAddress=""
          redundantLength="" plcCommunicationType="">
        - <communicationVariables>
            <communicationVariable wordPosition="3" bitPosition="1" variableParentPath="ar01\pc01\un01
              \Eq02\cm3" variableFullName="Eq02_cm3_MFwd" failureValue="" failureBehavior="" />
          </communicationVariables>
        </communicationTable>
      </communicationTables>
  </channel>
- <channel name="ch_pcl" importAction="" communicationType="NetPartner" communicationPartner="netp2">
  - <communicationPaths>
      <communicationPath index="1" plcNetworkSegment="eth1" plcNetworkAddress="10.180.39.10"
        partnerNetworkSegment="eth2" partnerNetworkAddress="10.180.38.104" connectionTimeout="0:2.0"
        globalNetworkPath="" />
      <communicationPath index="2" plcNetworkSegment="eth1" plcNetworkAddress="10.180.39.10"
        partnerNetworkSegment="eth2" partnerNetworkAddress="10.180.38.21" connectionTimeout="0:2.0"
        globalNetworkPath="" />
    </communicationPaths>
  - <communicationTables>
    - <communicationTable type="Bit" startAddress="4097" length="32">
      - <communicationVariables>
          <communicationVariable bitPosition="1" variableParentPath="ar01\pc01\un01\Eq02\cm3"
            variableFullName="Eq02_cm3_MFwd" />
        </communicationVariables>
      </communicationTable>
    </communicationTables>
  </channel>
</channels>
```

Abonné réseau

L'élément `networkNodes` est un élément enfant de `topologicalModel`.

L'élément `networkNodes` possède la structure suivante :

```
- <networkNodes>
  - <networkNode name="test" importAction="" type="NetPartner" description="" timeoutMin="" timeoutSec="" timeoutMs=""
      exportFile="" bitmapFile="">
    - <networkConnections>
        <networkConnection networkSegment="eth1" address="111.0.0.45" index="1" />
      </networkConnections>
      <comment>some text</comment>
    - <documents>
        <document type="exe" name="C:\FolderExist.exe" parameter="Folder file" />
      </documents>
    </networkNode>
  </networkNodes>
```

Format XML des listes d'instruments

Présentation

L'élément `instrumentList` est un élément enfant de l'élément `root`.

L'élément `instrumentList` possède la structure suivante :

```
- <instrumentList>
  - <controlModules>
    - <controlModule name="cm3" importAction="" type="Free" library="UAGBasic_Citect" description="cm3"
      hmiSymbolType="MonitorPro" hmiSymbolName="qwert">
      <comment>some text</comment>
     - <documents>
        <document type="exe" name="C:\FolderExist.exe" parameter="Folder file" />
      </documents>
     - <variables>
      - <variable name="Var1" importAction="" connectionType="PLC" dataType="bool" inOut="IN" description="swd">
       - <variableProperties>
          <variableProperty name="InitialValue" value="" enumeration="1" />
          <variableProperty name="OutputFailureBehavior" value="Last value" />
          <variableProperty name="VariableUsed" value="True" />
          <variableProperty name="GraphicalSymbol" value="False" />
          <variableProperty name="UpdateInitialValue" value="False" />
          <variableProperty name="DeadBandMode" value="Percentage" />
          <variableProperty name="Command" value="Logic" />
          <variableProperty name="Inverse" value="True" />
          <variableProperty name="AlarmPriority" index="1" value="1" />
          <variableProperty name="AlarmOperator" index="1" value="<" />
          <variableProperty name="AlarmLimit" index="1" value="95" />
          <variableProperty name="AlarmText" index="1" value="alarm occurred" />
          <variableProperty name="AlarmPriority" index="2" value="1" />
          <variableProperty name="AlarmOperator" index="2" value="<" />
          <variableProperty name="AlarmLimit" index="2" value="95" />
          <variableProperty name="AlarmText" index="2" value="alarm occurred" />
          <variableProperty name="FreeProperty" index="19" value="" enumeration="1" />
          <variableProperty name="FreeProperty" index="20" value="free" />
        </variableProperties>
      </variable>
     </variables>
    </controlModule>
   </controlModules>
  </instrumentList>
```

Règles de suppression et de changement de nom d'objets

Règles

Les règles suivantes s'appliquent pour supprimer et renommer des objets :

- L'attribut `importAction` doit être présent pour renommer et supprimer une opération.
- L'attribut `newName` doit être présent pour renommer une opération.
- Si l'attribut `importAction` conserve la valeur `delete`, l'objet est supprimé.
  Les objets qui peuvent être supprimés de l'interface utilisateur peuvent également être supprimés en utilisant le format XML d'exportation.
- Si l'attribut `importAction` conserve la valeur `rename`, et si l'attribut `newName` a une valeur, l'objet est renommé dans la mesure où le nouveau nom respecte la convention de nommage.
  Les objets qui peuvent être renommés à partir de l'interface utilisateur peuvent également être renommés en utilisant le format XML d'exportation.
- Si `newName` est présent et la valeur de `importAction` est `delete`, l'objet est supprimé. La valeur de `newName` est ignorée.
- Si `newName` est vide même lorsque `importAction` est `rename`, un message s'affiche alors et l'ancien nom est conservé.
- L'attribut `importAction` s'applique uniquement aux éléments qui existent déjà dans le projet. Il n'est pas appliqué à la création d'un élément.
- Si `importAction` d'un objet est `rename` et qu'il est fait référence de cet objet dans le même fichier d'importation, le nouveau nom est alors retenu pour son traitement.
  Par exemple, si l'importation d'un équipement avec un API affecté est incluse dans le même fichier d'importation, l'équipement doit inclure le nouveau nom d'API dans son attribut `PLC`.

NOTE : Pour modifier les valeurs d'autres attributs, à l'exception de `name`, l'attribut `importAction` n'est pas nécessaire et la valeur peut être directement modifiée. Pour créer de nouveaux objets, il suffit également de fournir un élément ayant un nom nouveau.

Sous-chapitre B.3
Format XML pour des bibliothèques de modèles

Format XML des bibliothèques de modèles

Bibliothèques de modèles

L'élément `templateLibrary` possède la structure suivante :

```
- <TemplateLibraryExchangeFile>
    <fileHeader company="Schneider Electric SAS" product="Unity Application Generator" productVersion="V3.2 Version 3.2.24.23 - Large Version" schemaVersion="1.0"
      dateTime="06/08/2009 14:31:50 02:00:00" />
    <contentHeader projectFile="MyTemplateLib" />
  - <templateLibrary>
    - <templateTypes>
      - <templateType name="eq2motors1valve" description="simple sample" hmiSymbolConfiguration="PictureTemplate">
        - <hmiSymbols>
            <hmiSymbol name="picture_template1" default="true" />
            <hmiSymbol name="picture_template2" default="false" />
          </hmiSymbols>
        + <variables>
        + <controlModules>
        - <templateTypeProperties>
            <templateTypeProperty variableFullName="eq2motors1valve_mot1_Instname" name="mot1-Instance Name"
              variableParentPath="MyTemplateLib\eq2motors1valve\mot1" variableProperty="InitialValue" tabName="Configuration" tabOrder="1" positionInTab="1"
              controlModuleName="mot1" controlModulePropertyName="Instance Name" />
            <templateTypeProperty variableFullName="eq2motors1valve_mot1_PAR.FbEn" name="mot1-Feedback Supervision"
              variableParentPath="MyTemplateLib\eq2motors1valve\mot1" variableProperty="InitialValue" tabName="Configuration" tabOrder="1" positionInTab="2"
              controlModuleName="mot1" controlModulePropertyName="Feedback Supervision" />
            <templateTypeProperty variableFullName="eq2motors1valve_mot2_Instname" name="mot2-Instance Name"
              variableParentPath="MyTemplateLib\eq2motors1valve\mot2" variableProperty="InitialValue" tabName="Configuration" tabOrder="1" positionInTab="3"
              controlModuleName="mot2" controlModulePropertyName="Instance Name" />
            <templateTypeProperty variableFullName="eq2motors1valve_mot2_PAR.FbEn" name="mot2-Feedback Supervision"
              variableParentPath="MyTemplateLib\eq2motors1valve\mot2" variableProperty="InitialValue" tabName="Configuration" tabOrder="1" positionInTab="4"
              controlModuleName="mot2" controlModulePropertyName="Feedback Supervision" />
            <templateTypeProperty variableFullName="eq2motors1valve_val1_Instname" name="val1-Instance Name"
              variableParentPath="MyTemplateLib\eq2motors1valve\val1" variableProperty="InitialValue" tabName="Configuration" tabOrder="1" positionInTab="5"
              controlModuleName="val1" controlModulePropertyName="Instance Name" />
            <templateTypeProperty variableFullName="eq2motors1valve_val1_PAR.FbEn" name="val1-Feedback Supervision"
              variableParentPath="MyTemplateLib\eq2motors1valve\val1" variableProperty="InitialValue" tabName="Configuration" tabOrder="1" positionInTab="6"
              controlModuleName="val1" controlModulePropertyName="Feedback Supervision" />
            <templateTypeProperty variableFullName="eq2motors1valve_int1" name="int1 - Initial Value" variableParentPath="MyTemplateLib\eq2motors1valve"
              variableProperty="InitialValue" tabName="Configuration" tabOrder="1" positionInTab="7" />
          </templateTypeProperties>
        </templateType>
      </templateTypes>
    </templateLibrary>
  </TemplateLibraryExchangeFile>
```

Sous-chapitre B.4
Format XML pour l'éditeur SCoD

Présentation

Cette section donne des informations sur le format XML pour l'éditeur SCoD.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Format XML des données d'en-tête (SCoD)	504
Format XML du type de module de commande (SCoD)	505
Format XML des variables de type de module de commande (SCoD)	506
Format XML des propriétés du module de commande, des FFBPins et des propriétés héritées (SCoD)	507
Format XML des données générales (SCoD)	508

Format XML des données d'en-tête (SCoD)

Présentation

Les données d'en-tête fournissent des informations sur l'origine du fichier XML comprenant :

- nom de la société
- nom du produit
- version du produit
- version de schéma
- date et heure
- contenu
- fichier de bibliothèque

Format des données d'en-tête

```
<?xml version="1.0" encoding="utf-8" ?>
- <ControlModuleTypeExchangeFile>
    <fileHeader company="Schneider Electric" product="Unity Application Generator" productVersion="3.2"
      SchemaVersion="1.0" dateTime="09/24/2008 10:13:31 +0530" content="Control Module Type Library source file" />
    <contentHeader file="C:\Documents\UAGTraining10Monitor.osl" />
```

Format XML du type de module de commande (SCoD)

Présentation

L'élément `controlModuleType` a pour éléments enfants directs `hmiSymbols`, `controlModuleTypeVariables`, `FFBPins`, `controlModuleProperties` et `inheritedProperties`.

L'élément `controlModuleType` possède la structure suivante :

```
- <controlModuleTypeLibrary name="IATBasic10Monitor" version="15.03.2006 16:20:31" description="V 1.0">
  - <controlModuleTypes>
    - <controlModuleType name="AIA01T" description="V 1.0" FFB="AIA01T" helpFile="AIA01.pdf"
       isCommitted="True" hmiSymbolAccess="Visible" version="15.03.2006 16:19:04">
       <icon />
      - <hmiSymbols>
         <hmiSymbol name="xxxx" type="Vijeo Citect Genie" default="1" />
       </hmiSymbols>
      - <controlModuleTypeVariables>
        - <controlModuleTypeVariable name="THHD" connectionType="PLC_HMI" dataType="REAL"
           description="Threshold HH & LL deviation" inOut="IN" bitfieldVariable="">
          + <variableProperties>
          </controlModuleTypeVariable>
       </controlModuleTypeVariables>
      + <FFBPins>
      + <controlModuleProperties>
      + <inheritedProperties>
      </controlModuleType>
    </controlModuleTypes>
  - <generalData>
    + <accessLevels>
    + <alarmGroups>
    + <alarmPriorities>
    + <archiveNames>
    + <communicationFrames>
    + <dataTypes>
    + <enumerations>
      <hmi type="VijeoCitect" />
    + <measurementUnitGroups>
    </generalData>
  </controlModuleTypeLibrary>
</ControlModuleTypeExchangeFile>
```

Format XML des variables de type de module de commande (SCoD)

Présentation

L'élément `controlModuleTypeVariables` possède la structure suivante :

```
- <controlModuleTypeVariables>
  - <controlModuleTypeVariable name="THHD" connectionType="PLC_HMI" dataType="REAL"
      description="Threshold HH & LL deviation" inOut="IN" bitfieldVariable="">
    - <variableProperties>
        <variableProperty name="ScalingMin" value="0" access="Visible" enumeration="SomeINTs" />
        <variableProperty name="ScalingMax" value="100" access="Visible" enumeration="SomeINTs" />
        <variableProperty name="AlarmPriority" index="1" value="1" access="Changeable"
          enumeration="" />
        <variableProperty name="AlarmOperator" index="1" value="<" access="Changeable"
          enumeration="" />
        <variableProperty name="AlarmLimit" index="1" value="95" access="Changeable" enumeration="" />
        <variableProperty name="AlarmText" index="1" value="alarm occurred" access="Changeable"
          enumeration="" />
        <variableProperty name="AlarmPriority" index="2" value="1" access="Changeable"
          enumeration="" />
        <variableProperty name="AlarmOperator" index="2" value="<" access="Changeable"
          enumeration="" />
        <variableProperty name="AlarmLimit" index="2" value="95" access="Changeable" enumeration="" />
        <variableProperty name="AlarmText" index="2" value="alarm occurred" access="Changeable"
          enumeration="" />
        <variableProperty name="FreeProperty" index="19" value="1" access="None"
          enumeration="<System UIDs>" />
        <variableProperty name="FreeProperty" index="20" value="2" access="None"
          enumeration="<System UIDs>" />
      </variableProperties>
    </controlModuleTypeVariable>
  </controlModuleTypeVariables>
```

NOTE : La structure ci-dessus ne montre que quelques propriétés. Le format reste le même pour d'autres propriétés.

Format XML des propriétés du module de commande, des FFBPins et des propriétés héritées (SCoD)

Présentation

Les éléments `FFBPins`, `controlModuleProperties` et `inheritedProperties` ont la structure suivante :

```
- <FFBPins>
    <FFBPin name="PVX" description="" dataType="REAL" inOut="IN" position="2" pinUsage="LiteralValue"
      literalValue="1.0" variableName="" variableProperty="" />
    <FFBPin name="PVL" description="" dataType="REAL" inOut="IN" position="3"
      pinUsage="LiteralVariableProperty" literalValue="" variableName="PVL" variableProperty="InitialValue" />
    <FFBPin name="PVH" description="" dataType="REAL" inOut="IN" position="4" pinUsage="Variable"
      literalValue="" variableName="PVH" variableProperty="" />
  </FFBPins>
- <controlModuleProperties>
    <controlModuleProperty variableName="HHEN" variableProperty="CommandID" displayText="HHEN
      Command" tabName="Enable" position="2" />
    <controlModuleProperty variableName="LATCH" variableProperty="InitialValue" displayText="Latch Enable"
      tabName="Enable" position="7" />
  </controlModuleProperties>
- <inheritedProperties>
  - <inheritedProperty>
      <source controlModuleTypeName="AIA01T" variableName="SP" variableProperty="DisplayFormatID" />
      <target controlModuleTypeName="AIA01T" variableName="TL" variableProperty="DisplayFormatID" />
    </inheritedProperty>
  </inheritedProperties>
```

Format XML des données générales (SCoD)

Présentation

L'élément `generalData` a la structure suivante :

```
- <generalData>
  - <accessLevels>
      <accessLevel name="1" desciption="level1" />
    </accessLevels>
  - <alarmGroups>
      <alarmGroup name="alarms" desciption="alarms" />
    </alarmGroups>
  - <alarmPriorities>
      <alarmPriority name="1" desciption="low" />
    </alarmPriorities>
  - <archiveNames>
      <archiveName name="archive1" desciption="archive1" />
    </archiveNames>
  - <communicationFrames>
      <communicationFrame name="frame1" dataType="BOOL" frameType="RD" />
    </communicationFrames>
  - <dataTypes>
    - <dataType name="CB1_MATH" type="<struct>" size="2" definition="" description="">
        <dataType index="1" name="Bit00" type="BOOL" size="1" description="" />
        <dataType index="2" name="Bit01" type="BOOL" size="1" description="" />
      </dataType>
      <dataType name="ArrayINT5" type="INT" dimensions="1..5" size="10" description="" />
    </dataTypes>
  - <enumerations>
    - <enumeration name="OnOff" type="BOOL" description="">
      - <enumerationValues>
          <enumerationValue value="1" description="On" />
          <enumerationValue value="0" description="Off" />
        </enumerationValues>
      </enumeration>
    </enumerations>
    <hmi type="VijeoCitect" />
  - <measurementUnitGroups>
    - <measurementUnitGroup name="measurementUnitGroup1">
        <measurementUnit name="Mu1" description="test" />
      </measurementUnitGroup>
    </measurementUnitGroups>
  </generalData>
```

Glossaire

A

Abonné réseau

Objet du modèle topologique, constitué de partenaires de réseau et des autres abonnés.

ActiveX

ActiveX est une spécification ouverte conçue pour régir l'interaction entre composants logiciels. Les objets ActiveX sont des composants logiciels préétablis, réutilisables qui possèdent des capacités interactives. Un ActiveX est un concept de programmation associant des données (propriétés) à des fonctions (méthodes) pour utiliser les données. Les objets graphiques dans l'IHM représentant les modules de commande de Unity Application Generator sont élaborés comme des composants ActiveX.

Administrateur

Une ou plusieurs personnes ayant la responsabilité de personnaliser Unity Application Generator selon les exigences de la société, par ex. les conventions de nommage, les outils de documentation. L'administrateur est responsable de l'administration de l'utilisateur / du groupe d'utilisateurs.

Administrateur système

Une ou plusieurs personnes chargées de personnaliser Unity Application Generator en fonction des besoins du projet.

Afficheur topologique

Représentation graphique du modèle topologique dans un écran distinct de Unity Application Generator.

Analyseur

Partie de Unity Application Generator vérifiant la validité formelle de toute la conception.

API

Automate Programmable Industriel, utilisé pour automatiser et commander un procédé.

Autre abonné

Abonné du réseau communiquant avec l'API mais n'étant ni un API Schneider, ni une IHM ni un serveur de données ou un partenaire réseau. Les PC des opérateurs ou les pilotes intelligents en sont des exemples. Il peut communiquer par Modbus Plus ou Ethernet avec d'autres périphériques, mais la communication avec l'abonné n'est pas définie dans Unity Application Generator. Il sert uniquement à réserver une adresse réseau et à afficher l'abonné dans l'afficheur topologique.

B

Bloc fonction (instance) (FB)

Unité d'organisation de programme fournissant des valeurs pour ses sorties et ses variable(s) interne(s) selon les algorithmes définis dans sa description de type de bloc fonction, s'il est exécuté comme une instance particulière. Toutes les valeurs des sorties et des variables internes d'une instance particulière de bloc fonction sont conservées entre deux appels de ce bloc fonction. Ainsi, plusieurs appels de la même instance de bloc fonction avec les mêmes arguments (valeurs des paramètres d'entrée) ne conduisent pas nécessairement aux mêmes valeurs de sortie.

Chaque instance de bloc fonction est symbolisée graphiquement par un bloc rectangulaire. Le nom du type de bloc fonction est centré en haut, à l'intérieur du rectangle. Le nom de l'instance de bloc fonction est également en haut, mais en dehors du rectangle. Il est généré automatiquement lors de l'établissement d'une instance, et ne doit pas être changé par l'utilisateur (si le FB a été généré par Unity Application Generator). Les entrées se trouvent sur la gauche et les sorties sur la droite du bloc. Les noms des paramètres d'entrée/sortie formels s'affichent dans le rectangle sur les points d'entrée/sortie correspondants.

C

Canal

Un canal désigne les éléments suivants :

- les objets du système d'automatisme avec lesquels l'API communique ;
- le chemin de communication sur lequel aura lieu la communication ;
- le chemin alternatif sur lequel aura lieu la communication ;
- les variables qui doivent être échangées.

La communication s'effectue par un API. Cela signifie que l'un des partenaires de communication d'un canal est un API.

Cartographie mémoire

Partie de Unity Application Generator. Elle génère toutes les adresses de variables, définitions d'E/S, paramètres de communication etc. de manière cohérente, à la fois pour l'API et l'IHM. Il est possible de lancer la cartographie mémoire lors de la phase de conception afin de vérifier l'adressage.

Cellule procédé

Groupement logique d'équipements comportant tous les équipements nécessaires à la production d'un ou de plusieurs lots. Elle définit l'étendue du contrôle logique d'un ensemble d'équipements de procédé au sein d'une zone, par exemple (défini dans S88).

Chemin de routage

Définit les ponts et leurs adresses interconnectant différents segments de réseau. (Dans les versions précédentes un chemin de routage était appelé chemin d'accès sur le réseau global.)

Chemins d'accès sur le réseau global

Ancien terme pour Chemins de routage

Commande

(Commande de procédé) Il s'agit de l'activité de conduite comprenant les fonctions de commande qui sont requises pour obtenir un contrôle séquentiel, de régulation et tout ou rien, ainsi que celle de la collecte et de l'affichage des données (défini dans S88).

Commande de procédé

Il s'agit de l'activité de conduite comprenant les fonctions de commande qui sont requises pour obtenir un contrôle séquentiel, de régulation et tout ou rien, ainsi que celle de la collecte et de l'affichage des données (défini dans S88).

E

Elément physique

Terme générique utilisé pour des objets, comme une zone, une cellule procédé, une unité et un équipement selon la définition utilisée dans le modèle physique de la norme ISA S88.01.

Il existe 2 types d'éléments physiques dans Unity Application Generator :

- Les éléments physiques structurels
  Ils structurent les installations d'automatisation d'un site et peuvent contenir d'autres éléments physiques.
- Les éléments physiques fonctionnels
  Il s'agit d'entités fonctionnelles selon la définition des équipements dans le modèle physique de la norme ISA S88.01.
  Contrairement à la norme ISA S88.01, ils sont utilisables à chaque niveau hiérarchique du modèle physique dans Unity Application Generator.
  Les éléments physiques fonctionnels peuvent contenir d'autres éléments physiques, modules de commande et variables.

Equipement

Groupe fonctionnel de modules de commande pouvant effectuer un certain nombre d'activités de traitement spécifique mineures.

En principe, un équipement est centré autour d'un poste de procédé (une cuve de pesage, un appareil de chauffage industriel, un laveur, etc.) ou d'activités de traitement, par ex. dosage et pesage.

F

FBD

Schéma fonctionnel (Function Bloc Diagram)

Une ou plusieurs sections contenant des réseaux représentés graphiquement et consistant en des fonctions, des blocs fonctions et des liaisons.

Fichier CSV

Un fichier CSV (comma separated value) est un fichier ASCII contenant un certain nombre de lignes. Chaque ligne d'un fichier CSV contient un certain nombre de champs séparés par un séparateur, par exemple une virgule, un point-virgule ou une tabulation. Les fichiers CSV peuvent servir à échanger des données entre différentes applications logicielles. Unity Application Generator permet d'importer et d'exporter des fichiers CSV.

Fichier XML

XML (Extensible Markup Language) est une spécification universelle permettant de créer des langages de balisage personnalisés.

XML est recommandé par le World Wide Web Consortium (W3C). C'est une norme ouverte gratuite. La recommandation spécifie la grammaire lexicale et les exigences d'analyse syntaxique (www.w3.org).

Les fichiers XML peuvent servir à échanger des données entre différentes applications logicielles. Unity Application Generator permet d'importer et d'exporter des fichiers XML.

I

IHM

IHM = Interface Homme Machine. Egalement appelé système superviseur. Grâce à l'IHM, l'opérateur contrôle le procédé de production.

Ingénieur automaticien

Spécialiste de tous les aspects d'un automatisme tel qu'API, IHM et mise en réseau, plutôt que du procédé proprement dit.

Ingénieur procédé

Spécialiste de tous les aspects du procédé devant être automatisé plutôt que de l'automatisme proprement dit.

Instruments

Capteurs et actionneurs servant à l'API pour la mesure et la modification des caractéristiques physiques et chimiques du procédé.

L

Liste d'instruments
Liste de tous les instruments utilisés dans un projet d'automatisme.

Lot
- Matériau en cours de fabrication ou ayant été fabriqué par une simple exécution d'un traitement par lots.
- Entité représentant la production d'un matériau en un point quelconque du procédé.

Le terme "lot" a deux sens : il couvre à la fois le matériau fabriqué par le procédé et pendant celui-ci, et l'entité représentant la production de ce matériau. Le terme "lot" est utilisé comme contraction abstraite des mots "la production d'un lot".

M

Modèle physique
Définit la hiérarchie des éléments physiques utilisés dans le procédé.

Modèle topologique
Dans Unity Application Generator, le modèle topologique définit l'architecture de l'automatisme. Il comporte la définition des composants matériels de l'API, les IHM et les réseaux.

Module de commande
Un ou plusieurs capteurs et actionneurs avec l'équipement de traitement associé, fonctionnant d'un point de vue de la commande comme une seule entité, par ex. un moteur, une vanne, un niveau, un témoin d'alarme.

Un module de commande est dérivé d'un type de module de commande (SCoD).

Module de commande libre
Module de commande qui n'est pas d'un type générique. Il ne définit aucune logique d'API ni aucune fonctionnalité prédéfinie pour l'IHM. Il est possible d'affecter des variables libres aux modules de commande libres. Ces modules sont généralement utilisés pour les composants matériels de procédé pour lesquels il n'existe aucun type de module de commande adapté.

Module de commande sans type

Module de commande auquel aucun type n'est affecté. Il ne s'agit ni d'un type de module de commande, ni d'un module de commande libre.

Avec des modules de commande sans type, l'ingénieur procédé a la possibilité de configurer un module de commande dans l'arborescence du modèle physique même s'il n'est pas certain de sa mise en œuvre spécifique.

L'ingénieur procédé peut affecter un module de commande sans type à un API. Tant que le type reste indéfini, rien n'est généré mais l'analyseur imprime un message. Il est toujours possible d'affecter un type à n'importe quel stade du développement du projet. Une fois le type défini, il est possible de définir les propriétés et variables de ce module de commande. Lorsque le type a été attribué, il n'est plus possible de revenir en arrière et de le désigner "module de commande sans type".

Module générique

Module matériel défini par l'utilisateur et non connu par son nom dans Unity Pro. L'administrateur les ajoute à la liste de modules dans la personnalisation. Il doit spécifier le nombre d'entrées et de sorties, le nombre de registres 0x (%M), 1x (%I), 3x (%IW) et 4x (%MW) utilisés ainsi que le type de châssis possible = générique. Les modules génériques ne peuvent être configurés que dans des châssis génériques et ne sont PAS générés. Ils ne peuvent pas être configurés dans des châssis Quantum/Premium et ne sont pas saisis dans l'affectation des E/S Unity Pro. Les spécifications effectuées ne sont utilisées que pour réserver des adresses de mémoire d'état ainsi que pour les adresses des variables d'E/S.

N

Nœud IHM

Dans iFIX, le terme "Nœud IHM" sert à désigner une IHM. Il correspond au terme IHM dans Unity Application Generator.

O

Objets de procédé

Elément d'équipement du procédé qui est représenté dans l'API et le superviseur (IHM) avec tous ses attributs.

OLE

Object linking and embedding = liaison et incorporation d'objets

Opérateur

Une ou plusieurs personnes responsables de l'utilisation du système exécutable d'automatisation du procédé afin de commander celui-ci.

P

Partenaire réseau

Abonné du réseau communiquant avec l'API mais n'étant ni un API Schneider, ni une IHM ni un serveur de données. Il s'agit par exemple des terminaux portables Magelis, des API d'autres fabricants, des capteurs ou actionneurs intelligents. Il peut communiquer par Modbus Plus ou Ethernet avec d'autres périphériques, c'est pourquoi la communication avec le partenaire réseau est définie dans Unity Application Generator.

PID

Pipework and Instrument Drawing = schéma de tuyauterie et d'instrumentation

Procédé

Séquence d'activités chimiques, physiques ou biologiques destinées à la transformation, au transport ou au stockage de matériau ou d'énergie (défini dans S88).

Procédure

Stratégie d'exécution d'un procédé.

R

Réseau

Interconnexion d'unités le long d'un chemin d'accès de données partagé, ces unités communiquant entre elles au moyen d'un protocole commun.

S

SCADA

Supervisory **C**ontrol **a**nd **D**ata **A**cquisition (télésurveillance et acquisition de données).

Dans iFIX, un système IHM est appelé système SCADA, un serveur de données est appelé hôte SCADA.

Segment de réseau

Les réseaux sont divisés en segments. Cela réduit la charge des données à transférer par segment puisque seules les données significatives pour les modules du segment concerné sont transmises sur un segment de réseau.

Serveur de données

Serveur de toutes les données relatives à l'IHM (par ex. les variables).

Site

Unité logique de structuration du procédé. Elément occupant le niveau le plus élevé du modèle physique de Unity Application Generator.

Superviseur (IHM)

Permet à un opérateur de gérer la marche du procédé à l'aide de l'automatisme.

T

Type de module de commande

Types génériques de module de commande testés et validés pouvant être réutilisés aussi souvent que nécessaire.

U

Unité

Ensemble de modules de commande et/ou d'équipements associés et d'autres équipements de procédé au sein desquels on peut conduire une ou plusieurs activités de traitement majeures (défini dans S88).

V

Variable

En général, paramètre à définition libre.

Unity Application Generator distingue deux principaux types de variable

- Variables de type de module de commande
- Variables libres

Variable de type de module de commande

Les variables de type de module de commande font référence à un type de module de commande. Elles sont générées automatiquement pour chaque instance d'un type de module de commande.

Variable libre

Les variables libres sont des variables définies manuellement pour :

- les éléments physiques fonctionnels
- les modules de commande.

Visualisation des alarmes

Schéma synoptique accessible par tous les messages d'alarme affectés à la visualisation des alarmes.

Z

Zone

Elément d'un site de production identifié par une segmentation physique, géographique ou logique au sein du site.

Zone de conduite

Groupement logique d'éléments de procédé commandés par un opérateur.

Sont affectés à la zone de conduite :

- les écrans d'opérateur
- les droits d'accès
- les alarmes

Index

0-9

21CFR11, *33*

A

Abonné
  abonnés Ethernet et règles d'adressage IP, *130*
Abonnés
  Réseau, *144*
Abonnés réseau, *144*, *237*
  étapes de définition des abonnés réseau, *324*
  groupe Abonnés de réseau du modèle topologique, *125*
accès, *42*
accès à l'Éditeur de sécurité, *42*
Acrobat Reader
  configuration logicielle minimale, *21*
actions
  actions de bus de terrain, *354*
  actions Profibus, *354*
Actualiser, *254*
Administrateur
  rôle, *37*
Adressage
  adressage dans un système de redondance d'UC, *133*, *137*
  règles d'adressage Ethernet, *129*
Adresse MBP
  Advantys, *373*
Advantys, *380*
  Adresse MBP pour NIM, *373*
  afficheur topologique, *381*
  Cartographie mémoire, *378*
  comportement à l'importation, *374*
  configuration, *372*
  E/S distribuées, *371*, *372*
  Ethernet, *373*
  exemple d'affectation d'E/S, *376*
  fonctionnalités améliorées, *381*
  génération incrémentale, *374*
  importation, *373*
  importer/exporter, *381*
  logique FBD, *380*
  périphérique MBP, *373*
  rapport, *381*
  rechercher, *381*
  restrictions, *374*
  restrictions à l'importation, *374*
  STB, *371*, *372*
affectation des DDT, *437*
Affectation des entrées/sorties des modules Advantys, *376*
affection des variables DDT à des points d'E/S, *437*
affichage
  actualiser par l'utilisateur, *254*
Affichage
  propriétés relatives à l'affichage, *106*
Afficheur de variables, *300*
  description générale, *301*
  filtre, *307*
  grille des variables, *311*
  zone d'informations, *303*
Afficheur des communications
  API-API, *165*
Afficheur des communications API-API, *165*
Afficheur topologique, *269*, *381*
  capture d'écran, *270*
alarme
  format de fichier CSV, *455*

Alarme
propriétés relatives aux alarmes, *102*
Analyse de la logique
forçage, *297*
Analyse syntaxique de la logique
forçage, *297*
And/Or
d'une recherche, fonction Trouver, *259*
any, *431*, *431*
API
canal, *156*
Copier et coller, *159*
étapes de définition des API, *324*
étapes de génération d'applications API et IHM, *327*
génération de code pour des API individuels, *387*
groupe API du modèle topologique, *125*
méthode de communication, *157*
propriétés, *151*
règles concernant l'API et Unity Pro, *248*
renommer, *154*
Application
création d'une application, *37*
application
étapes à suivre pour la documentation de l'application, *347*
schéma d'élaboration d'une application, *317*
Arborescence du modèle physique, *30*
arborescence du modèle physique
glisser-déposer dans l'arborescence du modèle physique, *257*
Arborescence du modèle topologique, *30*
ArchestrA
configuration logicielle minimale, *19*
Architecture du système
tâche de l'ingénieur automaticien, *38*
archiver
bus de terrain, *356*
Attribut
d'une recherche, fonction Trouver, *259*
Automate
automate principal et redondant, *132*
Automate principal, *132*
Automate redondant, *132*
Automatisme, *119*
automatisme
éléments d'un automatisme gérés par UAG, *120*
Autre abonné, *237*
définition et description, *241*
Autres abonnés
propriétés, *242*
avantages de Unity Application Generator, *30*

B

Barre d'outils graphiques
forçage, *289*
Base de données
base de données de procédé unique, *35*
Bibliothèque des types de modèle
format de fichier CSV, *475*
Bibliothèque SCoD
format de fichier CSV, *482*
bibliothèques
bibliothèques des types de module de commande, *74*
bibliothèques de modèles
répertoires, *24*
Bloc fonction
forçage, *283*
Boîte de dialogue Esclave
Profibus, *357*
Boîte de dialogue Nouvelle variable, *256*
boîte de dialogue Ouvrir lien(s), *84*, *84*
bus de terrain
actions, *354*
archiver, *356*
Bus de terrain
catalogue de modules, *360*
configuration, *177*, *351*
bus de terrain
Créer une configuration Profibus, *355*
exporter, *355*
Importer une configuration Profibus, *355*
modèle topologique, *353*
Ouvrir une configuration Profibus, *355*

Bus de terrain
Premium, *370*
Profibus, *177*, *351*
Quantum, *369*
bus de terrain
représentation, *353*
supprimer, *355*
Bus de terrain
Unity Pro, *369*, *370*
Bus de terrain (Profibus)
propriétés de l'API, *151*

C

Canal
Canal global, *160*
Canal spécifique, *160*
définition, *128*, *156*
étapes de définition des canaux de communication, *325*
Canal de communication
étapes de définition des canaux de communication, *325*
Canal global, *160*
paramétrage d'un canal global, *161*
Canal spécifique, *160*
paramétrage d'un canal Ethernet, *164*
paramétrage d'un canal spécifique (Modbus Plus), *161*
Caractère générique
caractères génériques utilisés dans une recherche, *260*
Caractère séparateur dans les fichiers CSV, *446*
Cartes processeur
cartes processeur Ethernet Momentum gérées, *130*
Cartographie mémoire
DDT, *416*
catalogue
Advantys, *375*
Catalogue de modules Profibus.osm, *360*
catalogue des modules Advantys, *375*
Cellule procédé
général, *64*
position dans la hiérarchie des modèles physiques, *33*
rapide définition, *62*
champ d'élément binaire
format de fichier CSV, *455*
Champ de texte libre d'un fichier CSV
sauts de ligne et tabulations, *448*
châssis, *169*
Châssis Ethernet amélioré, *167*
Châssis supplémentaires, *167*
Chemin de routage
description et propriétés, *146*
exemple de définition de chemin, *146*
Chemin du projet automate, *321*
Chemins de routage
groupe de l'arborescence du modèle topologique, *142*
Commande
propriétés relatives à la commande, *105*
Commande Enregistrer, *253*
Communication
comment définir la communication, *128*
communication API-API via Ethernet, *162*
communication API-API via Modbus Plus, *160*
définition de la communication, *156*
interruptions de communication (Ethernet), *163*
interruptions de communication (Modbus Plus), *161*
méthodes de communication, *157*
via Modbus, Modbus Plus et Ethernet, *126*
Communication API-API
via Ethernet, *162*
via Modbus Plus, *160*
Comportement à l'importation
Advantys, *374*
Comportement en cas de timeout
DDT, *436*
Conception du procédé
tâche de l'ingénieur procédé, *37*
Concepts généraux de l'interface utilisateur,

252
Condition de forçage
format de fichier CSV, *452*
Configuration
règles pour la configuration et la génération de projet, *246*
Configuration de projet
règles pour la configuration et la génération de projet, *246*
Configuration et procédure de mise à niveau, *17*
configuration logicielle minimale, *19*
Acrobat Reader, *21*
Advantys Configuration Software, *20*
ArchestrA, *19*
Hilscher Sycon, *20*
iFIX, *20*
Monitor Pro, *20*
MS Word, *21*
Prosoft Configuration Builder, *20*
Vijeo Citect, *19*
configuration matérielle minimale] , *18*
configuration minimale]
matériel, *18*
Configuration Profibus
Premium, *365*
Quantum, *362*
Contrôle d'accès au logiciel, *40*
accès par un utilisateur inconnu, *56*
contrôle d'accès au logiciel
Éditeur de sécurité, *42*
fonctions utilisateur, *45*
informations utilisateur, *43*
Contrôle d'accès au logiciel
mot de passe non valide, *56*
contrôle d'accès au logiciel
profils utilisateur, *49*
Contrôle d'accès au logiciel
protection de la base de données, *56*
contrôle d'accès au logiciel
règles, *54*
Contrôle d'accès au logiciel
reprise après détection d'une erreur, *56*
Convention de nommage
pour variables, *87*
Copier et coller
copier et coller dans les modèles physique et topologique, *264*
copier et coller entre projets, *264*
Copier et coller des API, *159*
Critère
d'une recherche, fonction Trouver, *259*

D

DDT, *440*
.x, *418*
affectation, *437*
affectation en mémoire, *429*
affichage des DDT, *436*
BOOL, *429*
BYTE, *429*
cartographie mémoire, *416*
comportement en cas de timeout, *436*
connexion de variables à des broches, *426*
DATE, *429*
DINT, *429*
DT (DATE_AND_TIME), *429*
DWORD, *429*
EBOOL, *429*
ECI, *440*
éditeur de personnalisation, *430*
éditeur SCoD, *417*
éditeur UAG, *433*
élément enfant, *421*
éléments, *426*
exemple, *427*
exemple d'importation, *427*
grille des variables, *434*
IHM générique, *442*
importation, *418*
importation de DDT dans l'éditeur de per-

sonnalisation, *431*
INT, *429*
IOX, *440*
MBTCP, *440*
Monitor Pro, *440*
ODX, *440*
REAL, *429*
règles d'affectation, *429*
règles de communication, *429*
section de communication Unity Pro, *436*
Serveur OPC, *440*
STRING, *429*
STRING[n], *429*
systèmes IHM, *438*
TIME, *429*
TOD (TIME_OF_DAY), *429*
type de connexion, *421*
type de donnée "any", *431*
UDINT, *429*
UINT, *429*
utilisation des DDT dans l'éditeur de personnalisation, *432*
utilisation des variables DDT dans l'éditeur UAG, *434*
variables supplémentaires, *436*
WORD, *429*
DDT existants, *419*
Décalage
entrée/sortie, *179*
Définition
élément physique fonctionnel, *66*
définition d'alarme, *108*
définition du forçage, *82*
fonctions complexes telles que SIN, COS dans une définition de forçage, *267*
Description générale
afficheur de variables, *301*
forçage, *278*
serveur de données, *222*
Développement de projet distribué
schéma du développement de projet distribué, *388*
développement de projets distribués, *385*
Diffusion des DDT entre les API, *436*
Diffusion des E/S, *157*, *379*
méthode de communication API-API, *160*
Directives
règles de travail avec Unity Application Generator, *245*
documentation
étapes à suivre pou la documentation de l'application ou d'objets individuels, *347*
droits d'accès, *57*

E

Editeur de personnalisation, *247*
éditeur de personnalisation
DDT, *430*
Editeur de personnalisation
lecture seule, *247*
éditeur de personnalisation
type de donnée dérivé, *430*
Editeur de sécurité, *40*
Éditeur de sécurité, *42*, *42*
Editeur de sécurité
accès par un utilisateur inconnu, *56*
Éditeur de sécurité
ajouter un nouveau profil utilisateur, *51*
ajouter un nouvel utilisateur, *48*
description des droits d'accès, *57*
Editeur de sécurité
description des paramètres de règle de sécurité, *55*
Éditeur de sécurité
droits d'accès, *57*
Editeur de sécurité
fichier journal, *41*
Éditeur de sécurité
fonctions utilisateur, *45*
Editeur de sécurité
gestion de la sécurité à UAG, *40*
informations utilisateur, *43*
Éditeur de sécurité
modifier un profil utilisateur, *52*
Editeur de sécurité
mot de passe non valide, *56*
onglet Informations utilisateur, *43*

Éditeur de sécurité
onglet Profils, *49*
Editeur de sécurité
Onglet Règles, *54*
Éditeur de sécurité
Onglet Utilisateurs, *45*
Editeur de sécurité
procédure de reprise après détection d'une erreur, *56*
profils utilisateur, *41*
Éditeur de sécurité
profils utilisateur, *49*
Editeur de sécurité
protection de la base de données, *56*
règles, *54*
super-utilisateur, *40*
utilisateur, *40*
Editeur SCoD, *247*
éditeur SCoD
DDT, *417*
Editeur SCoD
lecture seule, *247*
éditeur SCoD
type de donnée dérivé, *417*
éditeur UAG
DDT, *433*
type de donnée dérivé, *433*
Elément physique (fonctionnel)
définition, *66*
Elément physique fonctionnel
en relation avec l'API, l'IHM, les variables, *68*
caractéristiques, *68*
définition, *66*
élément physique fonctionnel
format de fichier CSV, *450*
Elément physique fonctionnel
propriétés, *70*
type de modèle, *72*
élément physique structurel
format de fichier CSV, *449*
éléments physiques
en général, *64*
Enregistrement de la logique
forçage, *299*
Enregistrer une ligne, *256*
entrée
contrôle de toutes les entrées, *254*
Entrée de forçage, *82*
Equipement, *246*
Équipement
général, *64*
Equipement
position dans la hiérarchie des modèles physiques, *33*
rapide définition, *62*
état de la génération
visualisation de l'état de la génération, *272*, *272*
Ethernet, *129*
Advantys, *373*
communication, *126*
communication API-API via Ethernet, *162*
méthode de communication, *157*
propriétés des segments de réseau Ethernet, *143*
Ethernet d'usine, *126*
Ethernet de bureau, *126*
Evénement, *157*
exportation d'un projet UAG, *401*
Exportation depuis l'éditeur SCoD
exportation XML, *413*
Exportation Magelis, *239*
Exporter
exemples d'exportation, *399*
interfaces avec d'autres outils, *397*
exporter
Profibus, *355*
exporter/importer
étapes d'importation/exportation de fichiers CSV/XML, *404*

F

FBD, *380*
Fenêtre des messages
résultat d'une recherche dans la fenêtre des messages, *260*

feuille de style
fournie, *441*
OFS, *441*
feuille de style fournie, *441*
FFB
forçage, *283*
fFormat XML
type de module de commande (SCoD), *505*
Fichier CSV
format de fichier CSV pour la bibliothèque des types de modèles, *475*
format de fichier CSV pour les instruments, *469*
format de fichier CSV pour les valeurs initiales, *470*
format du fichier CSV à renommer, *479*
format du fichier CSV à supprimer, *479*
fichier CSV
format du fichier CSV pour les éléments du modèle physique, *449*
Fichier CSV
définition, *399*
étapes d'importation/exportation de fichiers CSV, *404*
format, *399*
format de fichier CSV de suppression, *479*
format de fichier CSV pour la liaison de type de modèle, *477*
format de fichier CSV pour la propriété de type de modèle, *478*
format de fichier CSV pour le forçage de type de modèle, *478*
format de fichier CSV pour le module de commande des types de modèles, *476*
format de fichier CSV pour le symbole IHM, *475*
fichier CSV
format de fichier CSV pour les alarmes, *455*
format de fichier CSV pour les champs d'élément binaire, *455*
format de fichier CSV pour les conditions de forçage, *452*
format de fichier CSV pour les forçages, *452*
format de fichier CSV pour les liaisons, *451*
format de fichier CSV pour les modules de commande, *450*
Fichier CSV
format de fichier CSV pour un type de modèle, *475*
format de fichier CSV pour une bibliothèque SCoD, *482*
fichier CSV
format de fichier CSV pour une variable, *453*
Fichier CSV
format de fichier CSV pour une variable de type de modèle, *477*
format du fichier CSV à renommer, *479*
format du fichier CSV pour le modèle topologique, *456*
format général, *446*
fichier CSV
format pour élément physique fonctionnel, *450*
format pour élément physique structurel, *449*
Fichier CSV
importer/exporter, *398*
ligne de commentaire, *448*
liste des mots-clés, *447*
ordre des objets dans un fichier CSV à exporter, *448*
fichier CSV
pour importation dans un partenaire réseau, *157*
Fichier CSV
saut de ligne dans un champ de texte libre, *448*
tabulation dans un champ de texte libre, *448*
Fichier de configuration Advantys, *373*
Fichier journal, *41*

Fichier XML
définition, *400*
étapes d'importation/exportation de fichiers XML, *404*
format pour des bibliothèques de modèles, *502*
format pour SCoD, *503*
format pour un projet UAG, *485*
fichier XML
généralités sur le format, *484*
Fichier XML
importer/exporter, *398*
Fichier XML], *483*
fichiers de données, *24*
fichiers de schéma XML, *484*
fichiers pour schéma XML, *484*
Filtre
afficheur de variables, *307*
Finalisation
étapes de finalisation du projet, *325*
fonction
fonctions complexes telles que SIN, COS dans une définition de forçage, *267*
Fonctionnalités améliorées, *381*
forçage, *82*
Forçage
analyse de la logique, *297*
analyse syntaxique de la logique, *297*
barre d'outils, *289*
barre d'outils graphiques, *289*
bloc fonction, *283*
Copie, *83*
description générale, *278*
enregistrement de la logique, *299*
forçage
établissement d'une définition de forçage, *266*
Forçage
FFB, *283*
format de fichier CSV, *452*
liaison, *287*
récupération de la logique, *299*
représentation, *82*
forçage
zone de dessin, *293*
Forçage de type de modèle
format de fichier CSV, *478*
Format des fichiers CSV, *446*
Format des fichiers XML, *483*
Format XML
bibliothèques de modèles, *502*
données générales (SCoD), *508*
éléments physiques, *487*
format XML
éléments topologiques, *492*
Format XML
en-tête, *486*
en-tête (SCoD), *504*
FFBPins du module de commande (SCoD), *507*
listes d'instruments, *500*
propriétés du module de commande (SCoD), *507*
propriétés héritées du module de commande (SCoD), *507*
règles de suppression et de changement de nom, *501*
variables de type de module de commande (SCoD), *506*
Fréquence d'actualisation en lecture des variables API, *157*
fusion, *385*
Fusion
conditions préalables à la fusion de projets, *391*
fusion des modèles topologiques, *394*
schéma du développement de projet distribué, *388*

G

général
Cellule procédé, *64*
éléments physiques, *64*
Équipement, *64*
Unité, *64*
Zone, *64*
généralités sur le format XML
importer/exporter, *484*

Génération
étapes de génération d'applications API et IHM, *327*
génération
étapes de génération de fichiers d'importation pour les partenaires réseau, *329*
Génération
génération de code pour des API individuels, *387*
nouvelle génération ou génération incrémentale, *246*
règles pour la configuration et la génération de projet, *246*
Génération d'API
étapes de génération des API, *325*
Génération incrémentale, *374*
nouvelle génération ou génération incrémentale, *246*
Grille des variables
afficheur de variables, *311*
Groupe
groupe d'images, *219*
Groupe API, *148*
introduction, *149*
structure et captures d'écran, *149*
Groupe d'images, *219*

H

Hiérarchie
du modèle physique, *33*
Hilscher SyCon, *177*, *351*
Hilscher Sycon
configuration logicielle minimale, *20*

I

iFIX
configuration logicielle minimale, *20*
règles pour iFIX, *250*
IHM
définition, *215*
étapes de définition des IHM, *323*
étapes de génération d'applications API et IHM, *327*
groupe IHM, *214*
groupe IHM du modèle topologique, *125*
méthode de communication, *157*
propriétés, *216*
propriétés des variables relatives à l'IHM, *102*
règles concernant les IHM, *250*
IHM générique
DDT, *442*
types de données dérivés, *442*
Image et groupe d'images, *71*
Image et groupe d'images, *219*
Importation
configuration Advantys, *373*
importation
étapes de génération de fichiers d'importation pour les partenaires réseau, *329*
Importation
exemple : importation d'instruments / de la hiérarchie du modèle physique, *408*
exemple : importer des valeurs initiales de variables, *411*
interfaces avec d'autres outils, *397*
options de l'analyseur pour l'importation, *447*
importation de DDT, *418*
importation de DDT dans l'éditeur de personnalisation, *431*
importation incrémentielle, *406*, *406*
Importer
exemples d'importation, *399*
importer dans un projet UAG, *403*
Importer/exporter, *381*
bibliothèques de modèles de format, *502*
données générales au format XML (SCoD), *508*
en-tête au format XML, *486*
importer/exporter
étapes d'importation/exportation de fichiers CSV/XML, *404*

Importer/exporter
exemples CSV, *407*
exportation d'un projet UAG, *401*
exportation depuis l'éditeur SCoD, *413*
FFBPins du module de commande au format XML (SCoD), *507*
format éléments physiques, *487*
importer/exporter
format éléments topologiques, *492*
Importer/exporter
format listes d'instruments, *500*
format XML d'en-tête (SCoD), *504*
importer/exporter
généralités sur le format XML, *484*
importation incrémentielle, *406*
importer dans un projet UAG, *403*
Importer/exporter
informations générales, *398*
interfaces ouvertes, *399*
propriétés du module de commande au format XML (SCoD), *507*
propriétés héritées du module de commande au format XML (SCoD), *507*
règles de suppression et de changement de nom, *501*
type de module de commande (SCoD) au format XML, *505*
variables de type de module de commande au format XML (SCoD), *506*
XML, *413*
Informations sur les bus de terrain
généralités, *351*
ingénieur automaticien, *30*
Ingénieur automaticien
rôle, *37*
ingénieur procédé, *30*
Ingénieur procédé
rôle, *37*
installation
information, *22*
Instanciation
multiple, *41*
Instrument, *81*
format de fichier CSV pour les instruments, *469*
définition, *81*
importation d'instruments, *408*
propriétés, *81*
rôle, *81*
Interface
concept d'interfaces ouvertes pour l'importation et l'exportation, *399*
concepts de l'interface utilisateur, *253*
interfaces avec d'autres outils (importation et exportation), *397*
Interface utilisateur
concepts de l'interface utilisateur, *253*
concepts généraux, *252*
Introduction à UAG, *27*
IODDT, *437*
ISA S88.01
différences avec les conventions de nommage de l'ISA S88.01, *62*
modèle physique selon, *33*

L

le groupe Serveur de données, *221*
Lecture seule, *57*
Lecture/Écriture, *57*
Liaison
forçage, *287*
format de fichier CSV, *451*
Liaison de type de modèle
format de fichier CSV, *477*
liaisons entre les SCoD, *84*
Ligne de commentaire
définition d'une ligne de commentaire dans un fichier CSV, *448*
lignes de séparation, *68*
LIKE
Opérateur LIKE dans une recherche, *259*
liste
utilisation des tables (listes), *255*
Liste d'instruments
utilisation de la liste d'instruments, *261*
Liste de forçages
utilisation des tables (listes), *255*

Logiciel de configuration Advantys
configuration logicielle minimale, *20*
Logique générée
règles concernant la logique générée, *248*

M

M1E
cartes processeur Ethernet Momentum gérées, *130*
fonctions de communication, *162*
M340
modules Ethernet gérés, *131*
M580
E/S M580 Ethernet, *207*
modules Ethernet pris en charge, *131*
manipulation
manipulation de Unity Application Generator, *251*
Masque de sous-réseau, *143*
Menu contextuel, *255*
Mise à jour
étapes de mise à jour d'applications API et IHM, *328*
mise à niveau de projet vers UAG et Unity Pro, *25*
mise à niveau des projets, *25*
Modbus Plus
communication API-API via Modbus Plus, *160*
méthode de communication, *157*
Modbus, Modbus Plus
communication, *126*
Mode repli, *372*
Modèle physique, *59*
modèle physique
format de fichier CSV pour les éléments du modèle physique, *449*
Modèle physique
copier et coller dans les modèles physique et topologique, *264*
étapes de la définition du modèle physique, *322*, *326*
fusion des modèles physiques, *393*
hiérarchie, *33*, *60*
importation de la hiérarchie du modèle physique, *408*
présentation, *60*
rapide définition des éléments, *62*
selon l'ISA S88.01, *33*
structure, *60*
Modèle topologique, *117*
modèle topologique
bus de terrain, *353*
Modèle topologique
capture d'écran d'un exemple de modèle topologique, *124*
copier et coller dans les modèles physiques et topologique, *264*
étapes de définition du modèle topologique, *323*
format du fichier CSV pour le modèle topologique, *456*
fusion des modèles topologiques, *394*
représentation graphique du modèle topologique à l'aide de l'afficheur topologique, *269*
structure du modèle topologique, *124*
vue d'ensemble des éléments du modèle topologique, *124*
Module
cartes processeur Ethernet Momentum gérées, *130*
glisser-déposer des objets, modules et variables, *257*
module de redondance d'UC 140 CHS 110 00, *132*
modules Ethernet M340 gérés, *131*
modules Ethernet M580 pris en charge, *131*
modules Ethernet Premium gérés, *131*
modules Ethernet Quantum gérés, *130*

Module de commande, *246*
caractéristiques, *75*
définition, *74*
forçages des modules de commande, *82*
format de fichier CSV, *450*
Module de commande libre, définition, *78*
position dans la hiérarchie des modèles physiques, *33*
propriétés, *79*
rapide définition, *62*
représentation dans une IHM, *75*
représentation dans Unity, *75*
variables libres, *75*
Module de commande des types de modèles
format de fichier CSV, *476*
Module de commande intelligent (SCoD), *74*
Module de commande libre
définition, *78*
Module Ethernet
amélioré, *179*
Module Ethernet amélioré, *167*, *179*
module matériel, *169*
Module matériel
déplacement de modules matériels, *258*
Modules de commande intelligents (SCoD)
définition, *35*
Momentum
cartes processeur Ethernet gérées, *130*
Monitor Pro
configuration logicielle minimale, *20*
DDT, *440*
règles pour Monitor Pro, *250*
Mot-clé
liste des mots-clés dans les fichiers CSV pour l'importation et l'exportation, *447*
MS Word
configuration logicielle minimale, *21*
MSTR, *160*, *162*
Multiple instanciation, *41*

N

Navigation, *253*
utilisez la vue topologique pour naviguer vers les boîtes de dialogue de propriétés, *269*
NOE
fonctions de communication, *162*
modules de communication Ethernet Quantum gérés, *130*
NOM, *160*
Nom
règles concernant les noms, *249*
Nom de canal
renommer, *154*
nom du périphérique, *373*
Nombre
création de plusieurs instruments, *262*
création de plusieurs variables libres, *256*
Nouveau
option du menu contextuel des listes, *255*
Nouvelle génération
nouvelle génération ou génération incrémentale, *246*
Numéro de départ
création de plusieurs instruments, *262*
création de plusieurs variables libres, *256*
Numéro de liaison, *270*

O

Objet
glisser-déposer des objets, modules et variables, *257*
Objets générés
configuration du scrutateur d'E/S générée, *164*
OFS
feuille de style, *441*
Onglet Général
serveur de données, *225*
onglet Générateur de serveur de données et d'IHM
serveur de données, *234*
Onglet Redondance d'UC
propriétés d'API, *138*
Onglet Réseaux (serveur principal)
serveur de données, *226*
Onglet Réseaux (serveur secondaire)
serveur de données, *228*

Onglet Utilisations
propriétés des variables, *114*
OPC, *431*
Opérateur
d'une recherche, fonction Trouver, *259*
Options de l'analyseur pour l'importation, *447*

P

Paramètre, *157*
Paramètres de communication
châssis supplémentaires, *168*
Partenaire réseau, *237*
description et propriétés, *238*
partenaire réseau
étapes de génération de fichiers d'importation pour les partenaires réseau, *329*
Partenaire réseau
méthode de communication, *157*
propriétés, *240*
Passerelle
Modbus/Ethernet, *127*, *182*
Profibus/Ethernet, *127*, *188*
Passerelle Modbus/Ethernet, *127*, *182*, *182*
Passerelle par défaut, *143*
Passerelle Profibus/Ethernet, *127*, *188*, *188*
Périphérique MBP
Advantys, *373*
personnalisation
définition de la personnalisation, *320*
Personnalisation
tâche de l'administrateur, *37*
PID
importation d'instruments à partir du schéma PID, *408*
Pilote MBT, *20*
Plages d'adresses, *151*
PNN
fonctions de communication, *162*
Pont, *126*
définition d'un pont entre segments Ethernet, *127*
définition d'un pont entre segments Modbus, *128*
Premium
configuration Profibus, *365*
modules Ethernet gérés, *131*
Principes, *27*
Processus de fusion
fusion des modèles physiques, *393*
Profibus, *177*, *351*
actions, *354*
boîte de dialogue Esclave, *357*
propriétés de l'API, *151*
Profil
utilisateur, *41*, *41*
Profil utilisateur, *41*
Profil utilisateur par défaut, *41*
profils, *49*
profils utilisateur, *49*, *49*
Profils utilisateur préconfigurés, *41*
Propriété de type de modèle
format de fichier CSV, *478*
Propriétés, *102*
propriétés
Elément physique fonctionnel, *70*
Module de commande, *79*
propriétés d'un instrument, *81*
Propriétés d'un partenaire réseau, *240*
propriétés d'une zone de conduite, *217*
Propriétés
propriétés de l'API, *151*
propriétés de l'IHM, *216*
propriétés
propriétés des autres abonnés, *242*
Propriétés des chemins de routage, *146*
Propriétés
propriétés des segments de réseau, *143*
propriétés des variables, *90*
propriétés relatives à l'affichage, *106*
propriétés relatives à l'alarme, *102*
propriétés relatives à la commande, *105*
propriétés
serveur de données, *223*
Propriétés d'API
onglet Redondance d'UC, *138*
Propriétés de l'API
bus de terrain (Profibus), *151*

propriétés des variables
liées à l'API , *99*
Propriétés des variables
onglet Utilisations, *114*
propriétés des variables liées à l'API, *99*
Propriétés des variables relatives à l'IHM, *102*
ProSoft Configuration Builder, *177*, *351*, *352*, *352*
Prosoft Configuration Builder
configuration logicielle minimale, *20*
ProSoft Configuration Builder
étapes de la génération d'une configuration, *364*

Q

Quantum
configuration Profibus, *362*
Ethernet Quantum distant d'E/S (QERIO), *203*
modules Ethernet gérés, *130*

R

RAPD, *431*, *440*
Rapport, *381*
rapport
étapes à suivre pour la documentation de l'application ou d'objets individuels, *347*
Rapport
intègrez la vue topologique dans un fichier rapport Word, *269*
Recherche
constituants d'une recherche, *259*
recherche d'objets à l'aide de critères de recherche, *259*
Rechercher, *381*
Récupération de la logique
forçage, *299*
Redondance
configuration de redondance d'UC, *132*, *135*
possibilités de configuration de la redondance à l'aide de Unity Application Generator, *123*
redondance de canal pour la communication API-API via Ethernet, *163*
redondance de canal pour la communication API-API via Modbus Plus, *161*
Redondance d'UC, *132*, *132*, *135*
affectation de registres dans un système de redondance d'UC, *133*
Définition et exemple, *132*, *135*
fonctionnement, *132*, *136*
Premium, *135*
redondance d'UC et adressage, *137*
redondance d'UC et communication, *133*
registre de commande, *134*
Restrictions, *133*, *137*
zones de mémoire spécifiques, *138*
Redondant
serveur de données, *230*
Registre
affectation de registres dans un système de redondance d'UC, *133*
registre de commande de la redondance d'UC, *134*
registres réservés, *151*
Registre d'état (redondance d'UC), *134*
Registre de commande
registre de commande de la redondance d'UC, *134*
Registres de transfert inverse (redondance d'UC), *134*
Registres réservés, *151*
Règle
règles concernant l'API et Unity Pro, *248*
règles concernant les IHM, *250*
règles de travail avec Unity Application Generator, *245*
règles pour la configuration et la génération de projet, *246*
remarques sur la version
configuration logicielle minimale, *19*
Renommer
API, *154*
format de fichier CSV, *479*
nom de canal, *154*

Renommer/supprimer
format du fichier CSV, *479*
répertoires
bibliothèques de modèles, *24*
bibliothèques SCoD, *24*
feuilles de style XML, *24*
spécifiques, *24*
Réseau
étapes de définition du réseau, *128*, *323*
groupe Réseau du modèle topologique, *125*
réseaux gérés par Unity Application Generator, *122*
Routeur, *126*

S

Saut de ligne
définition d'un saut de ligne dans un champ de texte libre d'un fichier CSV, *448*
schéma opératoire
schéma d'élaboration d'une application, *317*
Schéma opératoire
schéma du développement de projet distribué, *388*
SCoD, *74*, *76*
définition, *74*
liaisons entre les SCoD, *84*
Scrutateur d'E/S, *157*, *379*
configuration du scrutateur d'E/S générée, *164*
méthode de communication API-API, *162*
sécurité d'accès au logiciel
droits d'accès, *57*
Segment
exigences relatives à un segment Ethernet, *129*
segments de réseau Ethernet local, *129*
Segment de réseau
définition, *126*, *143*
description et propriétés, *143*
segments de réseau Ethernet local, *129*
types de segments gérés, *126*
Segments de réseau
groupe de l'arborescence du modèle topologique, *142*
Sélection parmi les réceptions globales existantes, *161*
Serveur de données, *221*
description générale, *222*
étapes de définition des serveurs de données, *323*
groupe Serveur de données du modèle topologique, *125*
onglet Général, *225*
serveur de données
onglet Générateur de serveur de données et d'IHM, *234*
Serveur de données
onglet Réseaux (serveur principal), *226*
onglet Réseaux (serveur secondaire), *228*
serveur de données
propriétés, *223*
Serveur de données
redondant, *230*
relation avec d'autres éléments, *222*
Serveur OPC, *440*
Site
position dans la hiérarchie des modèles physiques, *33*
rapide définition, *62*
sortie de UAG, *31*
SPANG Power Electronics, *179*
Statistiques d'E/S, *155*
STB, *371*, *372*
Super-utilisateur, *40*
Supprimer
format de fichier CSV, *479*
supprimer
Profibus, *355*
Supprimer/renommer
format du fichier CSV, *479*
SyCon, *177*, *351*
Symbole IHMI
format de fichier CSV, *475*
Synoptique, *157*

système d'exploitation
configuration logicielle minimale, *19*
Systèmes IHM
DDT, *438*
type de donnée dérivé, *438*

T

Table
utilisation des tables (listes), *255*
Table du scrutateur d'E/S, *181*
Tabulation
définition d'une tabulation dans un champ de texte libre d'un fichier CSV, *448*
Taux de répétition
scrutateur d'E/S, *169*, *182*
Taux de répétition de scrutateur d'E/S, *169*, *182*
Topologie
figure illustrant la topologie d'un automatisme type, *120*
topologie
topologie d'un automatisme, *118*
Topologie
topologie d'un automatisme, *119*
topologie d'un automatisme, *118*
Topologie d'un automatisme, *119*
Topologie d'un automatisme
figure illustrant la topologie d'un automatisme type, *120*
trame de communication, *431*
any, *431*
transfert de variables DDT vers Monitor Pro, *440*
Transfert de variables DDT vers Vijeo Citect, *439*
Trouver
recherche d'objets à l'aide de critères de recherche, *259*
TSX PAY 262, *249*
TSX PAY 282, *249*
Type de châssis, *373*
Type de connexion
des variables, *88*
liste des types de connexions d'une variable, *90*
type de donnée "any", *431*
type de donnée dérivé
éditeur de personnalisation, *430*
éditeur SCoD, *417*
éditeur UAG, *433*
Type de donnée dérivé
systèmes IHM, *438*
Type de données
des variables, *88*
liste des types de données d'une variable, *90*
Type de liaison, *373*
Type de modèle
bibliothèque de modèles, *72*
élément physique fonctionnel, *72*
format de fichier CSV, *475*
généralités, *72*
Type de module de commande, *76*
bibliothèques, *74*
définition, *74*
propriétés des variables de type de module de commande, *90*
que définit-on dans un type de module de commande, *76*
Types de données
Types de données supplémentaires de Unity Pro, *419*
types de données dérivés
IHM générique, *442*

U

Unité
général, *64*
position dans la hiérarchie des modèles physiques, *33*
rapide définition, *62*
Unity Pro
configuration logicielle minimale, *19*
étapes de génération d'applications API et IHM, *327*
règles concernant l'API et Unity Pro, *248*
Types de données supplémentaires, *419*
Utilisateur, *40*

V

Valeur de repli, *372*
Valeur initiale
des variables Unity Pro, *88*
format de fichier CSV pour les valeurs initiales, *470*
Valeurs initiales
exemple : importer des valeurs initiales de variables, *411*
Valeurs par défaut, *254*
variable, *86*
Variable
affectation de variables, *258*
conventions de nommage, *87*
créer des variables libres, *256*
variable
définition d'alarme, *108*
Variable
exemple : importer des valeurs initiales de variables, *411*
variable
format de fichier CSV, *453*
Variable
glisser-déposer des objets, modules et variables, *257*
importance des variables, *87*
propriétés des variables de type de module de commande, *90*
propriétés des variables libres, *90*
propriétés des variables relatives à l'IHM, *102*
propriétés générales, *90*
règles concernant les variables, *248*
types de connexions, *88*
types de données, *88*
valeurs initiales, *88*
Variable du type de modèle
format de fichier CSV, *477*
Variable libre
créer des variables libres, *256*
d'un module de commande, *75*
propriétés des variables libres, *90*
Variables DDT
transfert vers Vijeo Citect, *439*
Variables de niveau supérieur, *422*
Variables supplémentaires
DDT, *436*
Vérification de l'état de santé
de la communication API-API via Ethernet, *163*
de la communication API-API via Modbus Plus, *161*
versions antérieures de UAG, *24*
Vijeo Citect
configuration logicielle minimale, *19*
Transfert de variables DDT vers, *439*
Vue des éléments, *422*

X

XSL
feuille de style, *441*

Z

Zone
général, *64*
position dans la hiérarchie des modèles physiques, *33*
rapide définition, *62*
Zone d'informations
afficheur de variables, *303*
Zone de conduite
définition, *215*
propriétés, *217*
zone de dessin
forçage, *293*
Zones de mémoire spécifiques
redondance d'UC, *138*